LA-6261-M
Manual
C.3

UC-34

Issued: December 1976

YAQUI User's Manual for Fireball Calculations

by

J. L. Norton
H. M. Ruppel

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

An Affirmative Action/Equal Opportunity Employer

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

This work was supported by the Defense Nuclear Agency.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22161
Price: Printed Copy $9.25 Microfiche $3.00

CONTENTS

I. DESCRIPTION OF CODE 1

A. Initial Zoning 1
B. Fireball Initialization 2
C. Particle Setup 4
D. Particle Movement 5
E. Turbulence 6
F. Rezone Techniques 12
G. Mixed Equation of State 12
H. System Improvements 13

II. DESCRIPTION OF INPUT 13

III. RESULTS OF A SAMPLE CALCULATION 17

A. *Initial Conditions* 17
B. Turbulence Seeding Conditions 25
C. Turbulence Equilibrium Conditions 27
D. Moderately Late-Time Conditions (Eight Torus-Formation Times) 30

IV. FLOW DIAGRAM 45

V. CODE LISTING 47

A. LTSS-Dependent Code 47
B. CROS-Dependent Code 61
C. KRONOS-Dependent Code 66
D. KRONOS/CROS-Dependent Code 69
E. System-Independent Code 76

REFERENCES 297

YAQUI USER'S MANUAL FOR

FIREBALL CALCULATIONS

by

J. L. Norton and H. M. Ruppel

ABSTRACT

Recent modifications and additions made to the YAQUI code are described. This code, which was written to simulate nuclear explosions in the atmosphere, has been improved to include the effects of turbulence. In addition, it now allows input data to be obtained by direct interpolation in the one-dimensional results of early time radiation codes. This new version also makes most of the input free-format (namelist) and is composed of modules for easier modification and isolation of computer system dependence.

I. DESCRIPTION OF CODE

The YAQUI code is a combination of two fluid-dynamical techniques, ALE and ICE. ALE is an acronym for Arbitrary-Lagrangian-Eulerian; using this method the finite difference mesh points can be moved with the fluid (Lagrangian), held stationary (Eulerian), or moved by some prescribed rezone algorithm (Arbitrary). ICE stands for Implicit Continuous-fluid Eulerian; because the hydrodynamic equations are solved implicitly, the technique can be used to solve for flow at all speeds. The analyses of these techniques were presented by Harlow and Amsden[1] and by Hirt et al.,[2] and an initial version of the code was described by Amsden and Hirt.[3]

This section will examine the differences between the initial version of the code and the modified version used to do the calculations in this report.

A. Initial Zoning

An initial YAQUI zoning setup is shown in Fig. 1. The initial grid is determined by the parameters DR, DZ, IBAR, JBAR, and YB. First, a uniform grid of IBAR zones in the r-direction and JBAR zones in the z-direction with the bottom at z = YB is generated. The zones are all DR cm wide by DZ cm high. Then the parameters FREZXR, FREZYB, and FREZYT are examined. If any one of them is larger than 1, it is assumed that a region of nonuniform zoning will surround the uniform region. The parameters that describe the nonuniform region are IUNF, JUNF, JCEN, and REZYO. IUNF is the number of zones starting at the left that are to remain uniform. To the right of this region, the zone widths are related by

$$DR(I+1) = DR(I)*FREZXR ,$$

where I increases to the right. Likewise, JUNF is the number of zones in the z-direction that will remain uniform. Referring to Fig. 1, one can see that the region of uniform zoning will be centered on the line JCEN zones above the bottom of the grid with half the uniform zones above the top of the JCENth zone and half below. The heights of the zones above the region of uniform zoning will be related by the equation

$$DZ(J+1) = DZ(J)*FREZYT ,$$

where J increases toward the top. The heights of the zones below the region of uniform zoning will be related by the equation

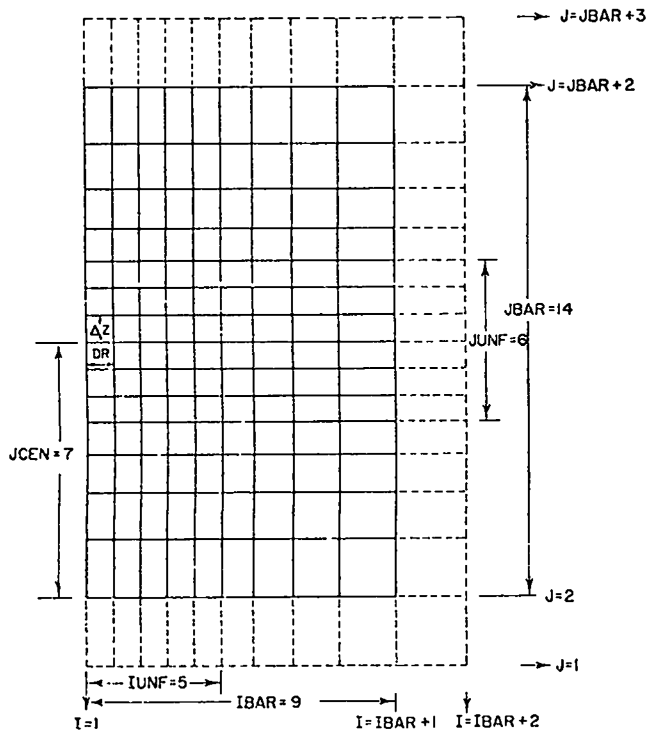


Fig. 1. Sample initial zoning setup.

DZ(J+1) = DZ(J)*FREZYB .

The top of the JCENth zone is redefined to be at the point z = REZYO. Because FREZYB will move the bottom of the mesh, it is necessary to recalculate YB based on JUNF, JCEN, REZYO, and FREZYB. In defining grids for problems with the ground at the bottom, care must be taken to calculate FREZYB so that the bottom of the J = 2 zone coincides with y = 0.

The grid is surrounded on three sides by layers of ficticious cells, shown as dotted lines in Fig. 1. These allow a variety of boundary conditions to be applied. The bottom layer of fictitious cells causes the first row of real zones to be at J = 2 rather than at J = 1. This fact can be confusing if one is not aware of the convention. The grid lines are known as I and J lines; I = 1 is at the extreme left; I = IBAR + 2 is at the extreme right of the dummy column; J = 1 is at the bottom of the bottom ficticious cells; and J = JBAR + 3 is at the top of the top ficticious cells. The zone whose lower left corner is at the intersection of the lines I and J is known as zone (I,J).

B. Fireball Initialization

The initial version of the code read in data that had been interpolated from one-dimensional spherical form to a two-dimensional grid with velocities centered at cell edges as in MAC. However, in YAQUI, velocities appear at vertices; therefore, a second interpolation was required. To avoid this unnecessary step and the resultant smoothing, the code was modified to permit inputting data in one-dimensional form. The data is then interpolated directly onto the YAQUI grid.

The interpolation technique is as follows. The 1D data is input one zone per card. Card i contains the radius of the outer boundary of zone i (r_i), the velocity of the outer boundary ($\dot{r}_i$), and the specific internal energy and density of the zone (e_i and ρ_i, respectively) in the format 4E15.0. Cards are read until a blank card or an end-of-file is encountered. The inner radius of the first zone is assumed to be zero (r_o = 0) as is the velocity at the center ($\dot{r}_o$ = 0). The density and energy are cell-centered quantities. A sample 1D setup is shown in Fig. 2. The 1D data is superimposed on the 2D YAQUI grid as shown in Fig. 3.

To carry out the interpolation, one divides each spherical shell into subzones using both radial and angular segments. Consider a portion of a 1D zone as shown in Fig. 4. Of course, in practice

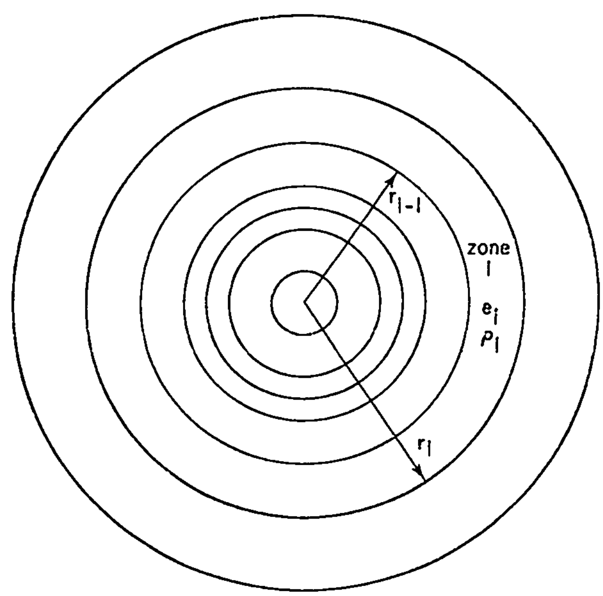


Fig. 2. Sample 1D fireball initial data configuration.

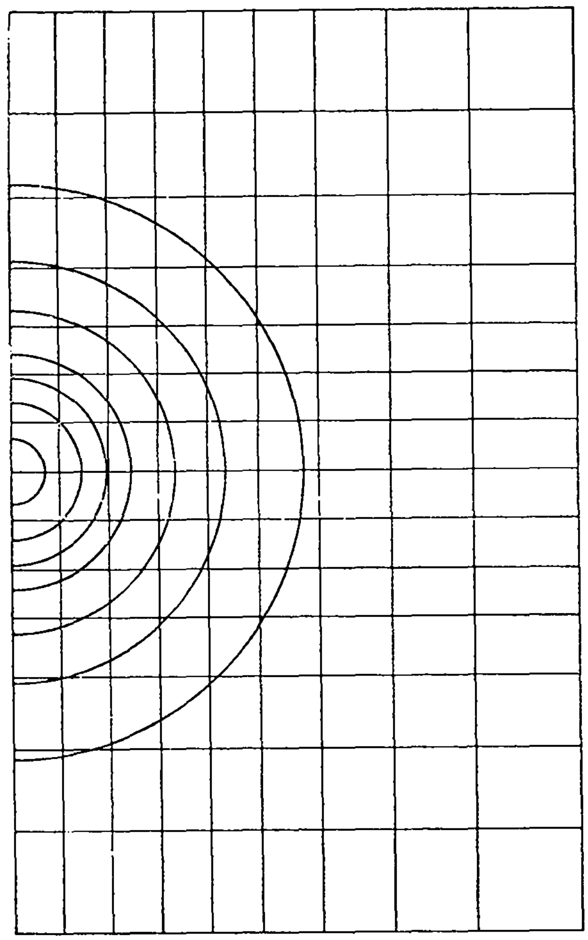


Fig. 3. Superposition of 1D fireball data on the 2D YAQUI grid.

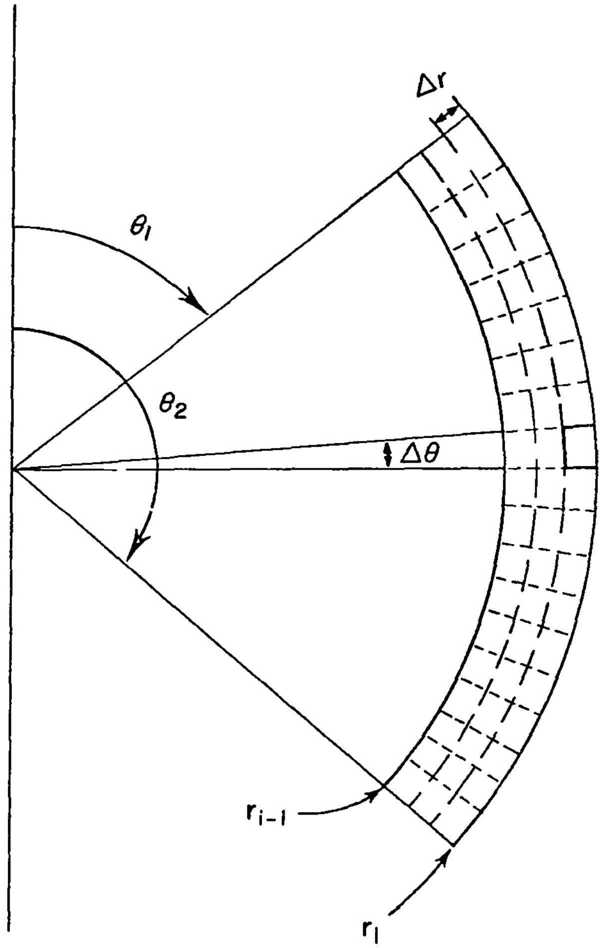


Fig. 4. Division of 1D fireball data into subzones for interpolation purposes.

θ_1 is 0° and θ_2 is 180°. $\Delta\theta$ is determined from 180°/NTH, where NTH is an input quantity; Δr is determined from $\frac{r_i - r_{i-1}}{\text{NRAD}}$, where NRAD is also an input quantity. NTH and NRAD are held constant for all zones. The center of a subzone is defined as shown in Fig. 5. The interpolation procedure, then, is as follows. Consider a single subzone of a spherical shell. Find into which YAQUI zone the <u>center</u> of the subzone falls. Assign <u>all</u> the mass, momentum, and internal energy of the subzone to the YAQUI zone. One can estimate the accuracy of the procedure by accumulating the volumes of the subzones that are assigned to an individual YAQUI zone. When all of the subzones have been assigned, the volume of the YAQUI zone and the sum of the volumes of the subzones assigned to the YAQUI zone should be nearly the same except for those YAQUI zones only partially within the outermost spherical shell.

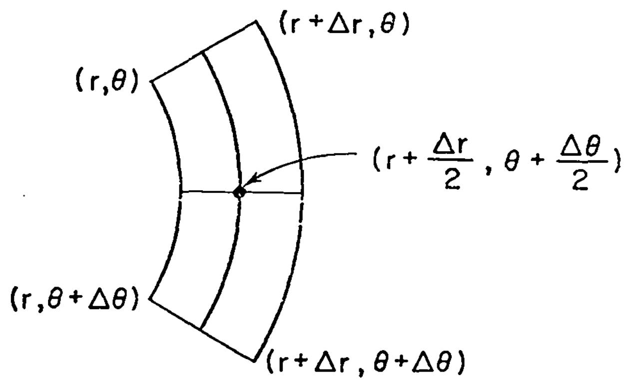


Fig. 5. A typical 1D Interpolation subzone.

To handle the latter, the sum of the subzone volumes is also used as a flag. If the volume error [(sum of subzone volumes - YAQUI zone volume)/YAQUI zone volume] is greater than 1%, then the five nearest cells on the top, bottom, and right are examined to see if any are within the fireball. (See Fig. 6.) If the YAQUI zone falls only partially within the fireball, at least one of the five neighbors will not be within the fireball at all. If all of the five neighbors are fireball cells, then there is an internal volume inaccuracy and the interpolation must be made finer (Δr and $\Delta\theta$ must be decreased). If a nonfireball neighbor is found, then the specific internal energy and density of this neighbor are used to establish the mass and internal energy of the part of the YAQUI zone that is not within the fireball.

To understand this better, consider the example shown in Fig. 6. Let r_{max} be the outer radius of the last 1D zone. The horizontally crosshatched portion of zone (i,j) is within the fireball and the vertically crosshatched portion is outside of the fireball. The relative volume error will certainly be greater than 1%. Of the five neighbors shown, cell (i+1,j-1) is completely outside of the fireball; its specific internal energy and density would be used to establish the internal energy and mass of the vertically crosshatched portion.

Once the internal energy, mass, and momentum components are established for all YAQUI zones that fall completely or partially within the fireball data, the specific internal energy is determined by dividing the zone internal energy by the zone mass. The zone density is determined by dividing the zone mass by the zone volume. The velocities are vertex quantities and are determined as follows. The cells containing a particular vertex are examined to see if all are fireball cells, i.e., if the vertex lies within the fireball. If any are not, then the vertex velocity is set to zero. If all are fireball cells, then the vertex takes $\frac{1}{N}$ of the momentum of each neighbor (for central cells, N = 4; for boundary cells, N = 2).

C. Particle Setup

The particle setup has two options: a rectangular particle region and a circular particle region.

Rectangular Region

The lower left-hand corner of the particle region is at (XC,YC) and the upper right-hand corner is at (XD,YD). The actual location of the particles is determined by superimposing a uniform grid of zones, which are DRPAR wide by DZPAR high starting at (XC,YC) and placing a particle at the center of each zone if the particle fails within the retangular region. See Fig. 7.

Circular Region

A circular particle region is more useful for fireball problems. This option is initiated by setting YD = 0. The circular region's center is at (0,YC), its radius is XD (XC is not used). Particles are placed as in the rectangular region except that the uniform grid starts at (0,YC-XD) and only particles that fall within the circle are used.

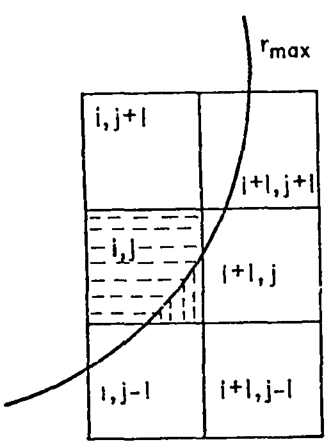


Fig. 6. Technique for handling YAQUI zones only partially within the fireball.

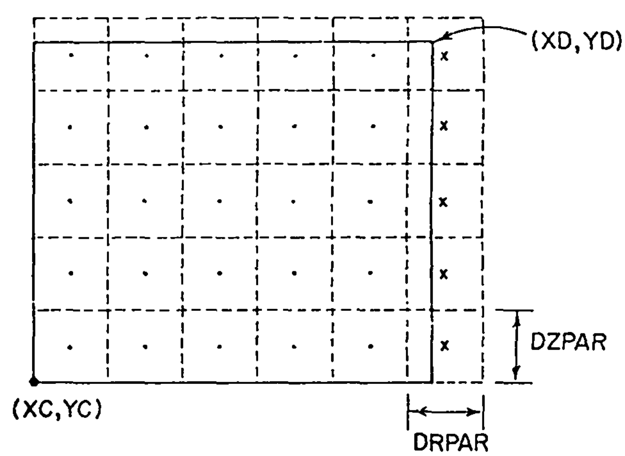


Fig. 7. Particle generation scheme.

D. Particle Movement

Calculating with nonrectangular cells poses some additional problems in the movement of marker particles. The approach used in the initial version of YAQUI was to define a grid of particle cells with constant Δx and Δy to overlay the calculational mesh. Masses and momenta were obtained by linear interpolation in the values assigned to the calculational grid for each vertex of the particle grid. The final step was to interpolate in these particle grid values to find the particle velocity with which it was to be moved. In addition to requiring two interpolations each cycle, this method tends to break down when a disparity exists between the size of calculation and particle cells. If the variable resolution causes large cells in one region of space and small cells in another, accuracy will be severely restricted. For such a situation there may be many fluid cells for one particle cell, or vice-versa. The former will cause smoothing of the velocity field as applied to the particles, the latter, to an uncertain determination.

To take advantage of the greater resolution that variable cells allow, we dispensed with the concept of an overlay grid, and interpolated directly in the fluid field. For this approach, two points must be considered: you must know in which cell a marker particle lies, and you must implement a reasonable scheme for interpolation in a skewed mesh. Consider first the method of searching for the values with which to do the interpolation. This is done by drawing vectors in succession from the particle to each of the vertices of a calculational cell. If these proceed in order and the angles between successive vectors are less than π for each of the four vectors, the particle lies within the cell. This may be seen more clearly in Fig. 8. In Fig. 8(a) the particle lies inside the cell. In Fig. 8(b), taking the vectors in order, the vector to vertex 4 lies between the vectors to vertices 2 and 3, and hence the particle lies outside the cell. In Fig. 8(c) the angle between the vectors to vertices 4 and 3 is greater than π, and the particle lies outside the cell.

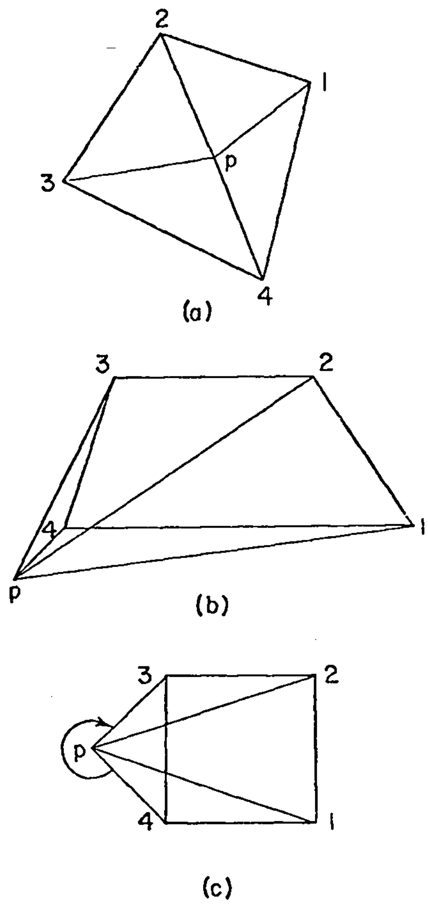


Fig. 8. Determination of particle positions.

The numbering of vertices is arbitary and may be clockwise or counter-clockwise, but must be consecutive. The rationale for this seemingly complicated criterion is that it avoids precise knowledge of any angles and hence requires no reference to trigonometric subroutines. Improved methods for which this is also true have since been found, one of which we will mention later. However, in the current version of the code, the above approach is used. To increase efficiency a one-dimensional array holds the number of the cell containing a particle at the beginning of a calculational cycle.

If a particle is no longer in the cell in which it began the cycle, an efficient search is undertaken to find the new location. For skewed cells, the particle can move more than one cell away, and hence determining an optimum trajectory along which to search is useful. This minimizes the number of tests that must be made.

Two criteria are imposed in selecting the interpolation scheme once the appropriate cell has been identified:

1. The interpolated velocity must be bounded by the vertex values — that is, its value must lie between the smallest and largest of the four velocities.

2. The interpolated values must be continuous across cell boundaries.

An efficient method is to map the (x,y) space to the logical (η,θ) space by the transformation

$$\vec{x} = (1-\eta)\ (1-\theta)\vec{x}_1 + (1-\eta)\theta\ \vec{x}_2 + \eta\theta\vec{x}_3 + (1-\theta)\eta\ \vec{x}_4. \quad (1)$$

The vectors $\vec{x}_1,\ldots,\vec{x}_4$ are drawn from any convenient origin to the vertices of the cell. The vector $\vec{x}$ points to the position for which interpolated velocities are desired. From the knowledge of the coordinates $\vec{x}_1,\ldots,\vec{x}_4$, and $\vec{x}$, we can invert Eq. (1) to obtain the values of η and θ corresponding to the point (x,y). In the (η,θ) space we then do a bilinear interpolation, using for any scalar the form

$$s(\eta,\theta) = (1-\eta)\ (1-\theta)s_1 + (1-\eta)\theta s_2 + \eta\theta s_3 + (1-\theta)\eta s_4. \quad (2)$$

We search for the new location of a particle by drawing a line from the center of the cell in which the particle began the cycle to its present location. This defines a trajectory that may pass through several cells and along which we perform our test for particle in cell. Although this is somewhat cumbersome, it requires testing very few cells, usually just one or two. An improved approach used by Pracht[4] eliminates the separate testing and automatically defines the direction of search. If the particle lies in the cell in question, the values of η and θ obtained from Eq. (1) will both lie between 0 and 1. If either does not, the cell indicated by the values of η and θ is examined. For example, if $\eta = 1.2$, increase i by one and recalculate η and θ. This approach is neater and more efficient. In future versions of the particle transport, we plan to incorporate such a scheme into YAQUI.

An additional component of particle displacement is required when the effects of turbulence are being calculated. This is described in the next section.

E. Turbulence

A significant new feature in the present version of YAQUI is the addition of a one equation transport model for turbulence. The modified equations, which we now solve are

the mass equation

$$\frac{\partial\rho}{\partial t} + \nabla\cdot\rho\vec{u} = \nabla\cdot\sigma\nabla\rho\ , \quad (3)$$

the momentum equation

$$\frac{\partial\rho\vec{u}}{\partial t} + \frac{\partial}{\partial t}\ \overline{\rho'u'} + \nabla\cdot(\rho\vec{u}\ \vec{u}) = \nabla\cdot\vec{\vec{\Pi}} + \rho\vec{g}\ , \quad (4)$$

the internal energy equation

$$\frac{\partial\rho I}{\partial t} = -\ p\nabla\cdot\vec{u} + \frac{2\rho\sigma q}{\beta s^2} + \nabla\sigma\cdot\nabla\rho I\ , \quad (5)$$

and the turbulence energy equation

$$\frac{\partial\rho q}{\partial t} = -\frac{\sigma}{\rho}\nabla\rho\cdot\nabla p + \rho\sigma\Gamma - \left(\frac{2}{3}\nabla\cdot\vec{u} + \frac{2\sigma}{\beta s^2}\right)\rho q$$

$$-\frac{2}{3}\sigma\rho(\nabla\cdot\vec{u})^2 + \nabla\cdot\sigma\nabla\rho q\ . \quad (6)$$

The turbulence viscosity is given by

$$\sigma = \beta s\sqrt{2q}\ . \quad (7)$$

For conciseness we have used the following definitions:

$$\Pi_{ij} = (\mu+\rho\sigma)e_{ij} + \sigma H_{ij} + \delta_{ij}\left[\left(\lambda-\frac{2}{3}\rho\sigma\right)\frac{\partial u_k}{\partial x_k}\right.$$

$$\left.-\left(p+\frac{2}{3}\rho q\right)\right]\ , \quad (8)$$

$$H_{ij} = u_i\frac{\partial\rho}{\partial x_j} + u_j\frac{\partial\rho}{\partial x_i}\ , \quad (9)$$

$$\Gamma = e_{ij}\frac{\partial u_i}{\partial x_j}\ ,\ \text{and} \quad (10)$$

$$e_{ij} = \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\ . \quad (11)$$

The parameters of the model are β and the scale, s, Values are obtained phenomenologically from fits to experiment. For β we use the constant, $\beta = 0.02$. For the scale, constant values over the mesh and the more complicated phenomenological form

$$s = 0.14\ d(z)\left(2 - \frac{v(r,z)}{v(o,z)}\right) \qquad v(r,z) \geq 0 \qquad (12a)$$

$$= 0.28\ d(z) \qquad v(r,z) < 0 \qquad (12b)$$

have been used. The function d(z) is the radial distance to the point at which the velocity changes sign, and v(o,z) is the axial component of the velocity on the axis, r = 0. In a skewed mesh Eq. (12) requires considerable logic and interpolation. Because s is only crudely known, we generally approximate the above form by

$$s = 0.14d\left(2 - \frac{v(r,z)}{v(o,z)_{max}}\right) \quad ; \qquad (13)$$

in this simplified relation, d is a constant, usually taken to be 1, and $v(o,z)_{max}$ is the maximum value of v along the axis. This expression may overestimate the scale for fireballs and cut down on the rate of decay of the turbulence energy. Erring in this direction gives us an upper bound on the effect of the turbulence. In the latest version of the code, we simply incorporate a constant scale throughout the mesh, though provision exists in the code for a more general treatment.

In differencing the turbulence additions, we can take advantage of much of the existing framework of the basic YAQUI program. Because small time-level inconsistencies in the turbulence equations are insignificant, economy is a major criterion in establishing the order in which the several additions are included. For example, looking at the structure of the modified stress tensor, we are led to replace $\mu \rightarrow \mu + \rho\sigma$, $\lambda \rightarrow \lambda - \frac{2}{3}\rho\sigma$ and $p \rightarrow p + \frac{2}{3}\rho q$. Similarly H_{ij} is calculated from ρ and $\vec{u}$ at the old time level; this allows the stress tensor to be handled explicitly in Phase I simplifying the addition. The term $\frac{\partial}{\partial t}\overline{\rho' u_i'}$ in the momentum equation is modeled, using the flux approximation, as $-\frac{\partial}{\partial t}\sigma\frac{\partial\rho}{\partial x_i}$; this requires saving the two components of $\sigma\frac{\partial\rho}{\partial x_i}$ for each cell from the previous time step. Clearly such an approach centers the time derivative about $t - \frac{\delta t}{2}$ and not at the same time level as Π_{ij}. For reasonable time variations, the inconsistency should not be important.

Three more diffusion terms must be differenced: one in the mass equation, one in the internal energy equation, and one in the turbulence energy equation. Each of these is added explicitly in Phase I with derivatives from time level n.

The transport equation for the turbulence energy q is mainly handled explicitly. However, we can include some advanced time information by writing Eq. (6) in the form

$$(\rho q)^{n+1}\left[1 + \frac{2}{3}\nabla\cdot\vec{u} + \frac{2\sigma}{\beta s^2}\right]^n = (\rho q)^n - \sigma\delta t\left[\frac{\nabla\rho\cdot\nabla p}{\rho} - \rho\Gamma + \frac{2}{3}\rho(\nabla\cdot\vec{u})^2\right]^n + (\nabla\cdot\sigma\nabla\rho q)^n. \qquad (14)$$

In cylindrical geometry Γ takes the form

$$\Gamma = 2\left[\left(\frac{\partial u}{\partial r}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2 + \frac{1}{2}\left(\frac{\partial u}{\partial z} + \frac{\partial v}{\partial r}\right)^2 + \left(\frac{u}{r}\right)^2\right] \quad . \qquad (15)$$

All the terms on the right-hand side of Eq. (14) are evaluated at time level n as cell-centered quantities. For most of the terms this proceeds naturally; but for the buoyancy creation, $\frac{\sigma}{\rho}\nabla\rho\cdot\nabla p$, a small reach is necessary. In the notation of Fig. 9 we could evaluate the derivatives directly at the

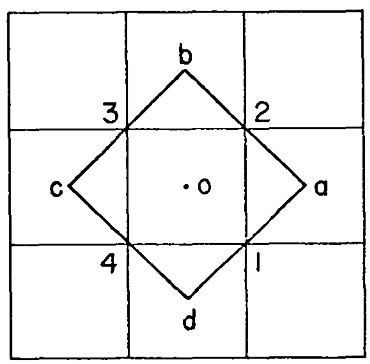


Fig. 9. Quantities involved in YAQUI derivative evaluation.

vertices 1-4 and then average to obtain $(\nabla\rho\cdot\nabla p)_o$. This uses information at nine cells and is unnecessarily smoothing. For this reason we use cells a,b,c, and d and obtain the derivatives at the center of cell o in terms of these. For example,

$$\frac{\partial\rho}{\partial r} = \frac{1}{2A}\left\{(\rho_a-\rho_c)\ (z_b-z_d) + (\rho_b-\rho_d)\ (z_c-z_a)\right\} , \qquad (16)$$

where A is the area of the quadrilateral abcd. We are particularly concerned about smoothing out derivatives because the sensitivity of the turbulence energy and the level of turbulence that can be supported depend on the size of gradients of various quantities. Derivatives of vertex quantities are calculated at cell centers by the equations

$$\frac{\partial u}{\partial r} = \frac{1}{2A}\left\{(z_2-z_4)\ (u_1-u_3) + (z_3-z_1)\ (u_2-u_4)\right\} \qquad (17a)$$

and

$$\frac{\partial u}{\partial z} = \frac{1}{2A}\left\{(r_2-r_4)\ (u_3-u_1) + (r_1-r_3)\ (u_2-u_4)\right\} , \qquad (17b)$$

where A is the area of the cell.

If we expand the derivatives about the center of the cell, we find that the error is proportional to second and higher derivatives. In a similar fashion we can obtain derivatives of cell-centered quantities at cell centers (as discussed above) by appropriate choice of configurations. The necessity to optimize the calculation of gradients implies that high-resolution calculations are important in evaluating turbulence models.

Similar problems arise for the Laplacian or diffusion-like terms. Again, several approaches are possible and in this case optimization is much less crucial, because the level of turbulence is only indirectly affected. The criteria applied to select an approach for differencing such terms were: that an equation of the form

$$\frac{\partial c}{\partial t} + \vec{u}\cdot\nabla c = \nabla\cdot\sigma\nabla c$$

could not lead to negative c anywhere in the mesh, and that the difference form of $\nabla\cdot\sigma\nabla c$ should reduce to the expected form for the case of a uniform rectangular mesh. For example, if c represents the concentration of a chemical species,

$$\frac{1}{V_o}\sum_{j=1}^{4}\frac{c_j-c_o}{\frac{1}{2}(A_j+A_o)}\,d_{jo}^2\langle r_{jo}\rangle\langle\sigma_{jo}\rangle = (\nabla\sigma\nabla c)_o , \qquad (18)$$

where c_j is the concentration in cell j, d_{jo} is the length of the cell side between cells o and j, $\langle r_{jo}\rangle$ is the distance between the centers of cells o and j, A_j is the area of cell j, V_o is the volume of cell o, and $\langle\sigma_{jo}\rangle$ is $\frac{\sigma_j+\sigma_o}{2}$. See Fig. 10 for examples. In Cartesian coordinates for a uniform rectangular mesh with sides δx, δy, and constant σ, this reduces to

$$\sigma\left(\frac{c_1+c_3-2c_o}{\delta x^2} + \frac{c_2+c_4-2c_o}{\delta y^2}\right) \qquad (19)$$

as we would wish. The vertices of this quadrilateral are at the center of their respective cells.

In the finite difference approach with finite time steps it is possible for the turbulence energy to become negative in certain cells. If, for example, $\delta t\,\frac{\sigma}{\rho}\nabla\rho\cdot\nabla p$ is positive and greater than ρq in a given cell, more energy will be subtracted from the cell than it contains. One could reduce the time step, but this would never really solve the problem. The assumption we make is that since, numerically, turbulence can go negative in regions in which it is decaying, if $q < 0$, we set q to zero. Experience has shown for the fireball calculations that less than 1% of the turbulence energy is lost by this procedure.

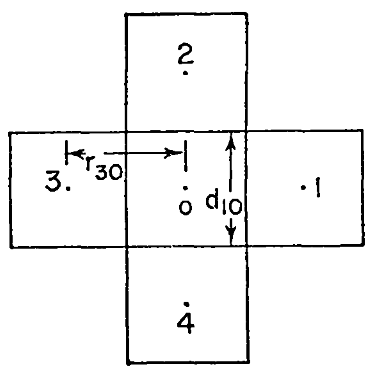


Fig. 10. Definition of quantities used in differencing of turbulence part of the concentration equation.

As far as convection is concerned, the quantity ρq is convected and, because the difference equations for the convection conserve the convected quantity identically, the turbulence energy is conserved in Phase III. The total energy, however, is not. In the initial code version[3] an equation for the total energy is written, and this is the quantity that is convected. Momentum is also convected, and hence conserved; the internal energy is obtained by subtracting the kinetic energy from the total energy. This places all the uncertainty in the internal energy, which if it is a small part of the total energy, can lead to large fractional errors in the internal energy.

With the introduction of the equation for turbulence energy, it has seemed convenient to deal directly with the internal energy. In the code the equation for the internal energy follows the pressure iteration, allowing us to calculate the pdV work with a time-advanced pressure. The quantities that are convected in Phase III, then, are the internal energy, the turbulence energy, and the momentum, which are individually conserved. In general, this means that the kinetic energy and, thus, the total energy, $E = \rho I + \rho u^2 + \rho q$, will not be conserved. This lack of energy conservation does not seem to be significant and is ignored in actual calculations.

A small change has been made in the energy equation related to the smoothing of the velocity field for computational stability. If alternate mesh vertices are not coupled in some way (see Ref. 3), an instability arises. The approach we elect is to couple the alternate nodes only when a local minimum or maximum in the velocity field occurs. We apply a restoring acceleration to vertex 4 of the form

$$\frac{1}{a_{nc}\,\delta t}\left[\frac{1}{4}\left(\vec{u}_1+\vec{u}_3+\vec{u}_6+\vec{u}_8\right) - \vec{u}_4\right] \qquad (20)$$

(See Fig. 2 of Ref. 3).

In Ref. 3 this is applied to each vertex each cycle to control the instability. In our version we apply the restoring force only to those components of $\vec{u}$ at vertex 4 that have values larger than or smaller than any of the neighbors (1,3,6, and 8). That is, a local maximum or minimum in the r-component of $\vec{u}$ is smoothed by a restoring acceleration in the r-direction and similarly for the z-component. This is less diffusive than the approach that applies it everywhere each cycle. The intent is that when no instability threatens, this will not smooth gradients.

This node coupler clearly reduces the kinetic energy of the system, acting like a viscous dissipation. If this loss is ignored, it will lend to a gradual diminishing of the system's energy. On the other hand, to include the energy in the internal energy equation as viscous heating really has no basis in physical reality. However, in our version, we choose the latter option and include the energy removed by the node coupler in the equation for the internal energy. In the original version of the code the other choice was made.

One can see by examining the model equations for turbulence that no mechanism has been built in to initiate the turbulence spontaneously. That is if there is no turbulence present, i.e. $q = 0$, none can be created or can grow. For this reason, the turbulence must be seeded initially and allowed to equilibrate with the mean flow through the creation and decay terms in the equation for the turbulence energy. Several alternative seedings have been tried and found to lead to the same turbulent configurations after a fairly short time. A very reasonable approach is to seed the turbulence proportional to the vorticity of the mean flow field. This is done after the field has been established, that is, shortly before torus formation time. This timing is not crucial. It can be seeded earlier and find its way to a similar level and distribution in a short time. We miss any high-intensity early time turbulence, likely initated by Taylor instabilities as the device case and the very hot debris decelerate. We assume that the fluctuations decay in a few seconds because nothing appears to be present to support them. The equilibrium turbulence we calculate really only has meaning at later times.

Because the measured properties of the fireball, rate of rise and radial expansion, depend on the positions of the marker particles, it was felt that their motion should be coupled directly into the turbulence. This is done by adding a random turbulent diffusion velocity to the particle motion by the following technique.

The diffusion of mass as a function of time from a point source of unit mass at position $\vec{r}_o$ is described by the diffusion equation

$$\frac{\partial \rho}{\partial t} = \lambda \nabla^2 \rho + \delta(\vec{r}-\vec{r}_o) , \tag{21}$$

where ρ is the density and λ is a constant diffusion coefficient. Define the quantities $\delta\vec{r} = \vec{r}-\vec{r}_o$, $\delta x = x-x_o$, $\delta y = y-y_o$, and $\delta z = z-z_o$. If $\rho(\delta\vec{r},t)$ is written as $\rho(\delta\vec{r},t) = X(\delta x,t)\ Y(\delta y,t)\ Z(\delta z,t)$, it can be shown that

$$X(\delta x,t) = \frac{1}{\sqrt{4\pi\lambda t}}\, e^{-(\delta x)^2/4\lambda t} , \tag{22}$$

with similar expressions for Y and Z.

In calculating the additional particle motion due to turbulence, consider the particle at time $t_o = 0$ to be a massless point at $\vec{r}_o$ and use Eq. (22) as a probability distribution function for determining the position of the particle at time $t = t_o+\delta t$. The turbulence viscosity then serves as the diffusion coefficient λ. In cylindrical coordinates one should solve Eq. (21) in a cylindrical basis

$$\rho(\delta\vec{r}) = R(\delta r)\ Z(\delta z) .$$

However, R cannot be determined in closed form and for small δt and δx, with $\delta x > \frac{\lambda\delta t}{x_o}$, R reduces to the form of X in Eq. (22). Thus, Eq. (22) can be used with confidence in cylindrical geometry as long as it is applied in its region of validity. Furthermore, δx must be small compared with a cell dimension so that the turbulence viscosity being used as the diffusion coefficient remains constant in the region and during the time considered.

The general Gaussian (normal) distribution is

$$f(y) = \frac{1}{\sqrt{2\pi}\,\sigma}\, e^{-y^2/2\sigma^2} , \tag{23}$$

where σ is the standard deviation. Equation (22) can be put in this form by setting

$$\sigma = \sqrt{2\lambda\delta t} \quad \text{and} \tag{24}$$

$$y = \delta x , \tag{24a}$$

where we have replaced t by δt.

The standard probability distribution function (random number generator) available on most computers is the uniform distribution $p(x)$,

$$\begin{aligned} p(x) &= 1 , \quad 0 \le x \le 1 \\ &= 0 , \quad \text{elsewhere.} \end{aligned} \tag{25}$$

What we need is the distribution given by Eq. (23) with σ given by Eq. (24). To accomplish this Eq. (25) is mapped on to Eq. (23) (see Fig. 11) with the expression

$$p(x)\ dx = 2\ f(y)\ dy, \tag{26}$$

because the probability of finding x in dx is equal to the sum of the probabilities of finding both y and -y in their respective dy's. Therefore,

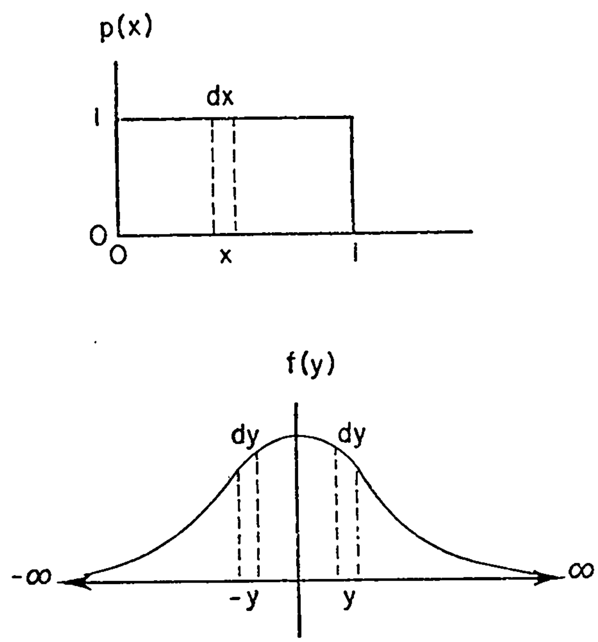


Fig. 11. (a) Uniform distribution, (b) Gaussian distribution.

$$\int_0^x p(x') \, dx' = 2\int_0^y f(y') \, dy' \quad ,$$

$$\text{or} \int_0^x dx' = \frac{2}{\sqrt{2\pi}\,\sigma} \int_0^y e^{-y'^2/2\sigma^2} \, dy' \; .$$

With the variable transformation $z' = \frac{y'}{\sigma\sqrt{2}}$,

$$x = \frac{2}{\sqrt{\pi}} \int_0^{y/\sigma\sqrt{2}} e^{-z'^2} \, dz' \; .$$

By definition, the standard error function is

$$\mathrm{erf}(t) \equiv \frac{2}{\sqrt{\pi}} \int_0^t e^{-t'^2} \, dt' \; ;$$

therefore,

$$x = \mathrm{erf}\left(\frac{y}{\sigma\sqrt{2}}\right) \; , \tag{27}$$

and from Eq. (24),

$$y = \sqrt{4\lambda\delta t} \; \mathrm{erf}^{-1}(x) \; . \tag{28}$$

The above is the mapping from Eq. (25) to Eq. (23) that is desired.

In principle, then, one determines a random number using Eq. (25) and maps it onto Eq. (23) using Eq. (28). One can see that the y in Eq. (28) is the required particle displacement due to turbulent diffusion if he remembers that the diffusion coefficient λ in Eq. (21) is replaced by the local turbulence viscosity.

One problem remains, namely, making the procedure efficient enough computationally so that one can afford to use it. The major difficulty is the calculation of inverse error functions. To do this without undue cost the following interpolation procedure was established.

The error function is monotonic increasing on the interval [0,1]; erf(0) = 0; and $|\mathrm{erf}\,(y)|$ approaches 1 asymptotically (see Fig. 12). Care must be taken in using $\mathrm{erf}^{-1}(x)$ for $x \sim 1$, for in this region $\mathrm{erf}^{-1}(x)$ is very large which could lead to an abnormally large particle displacement. To

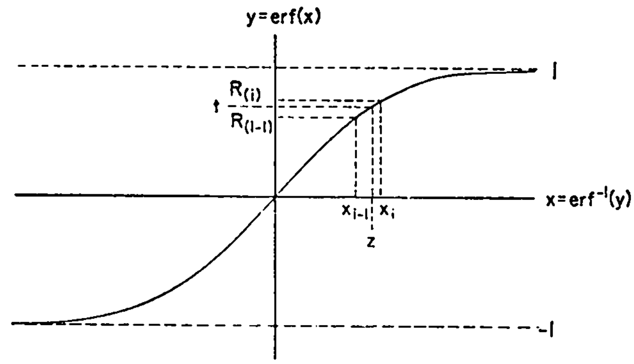


Fig. 12. Standard error function.

handle this, one selects a number WMAXEF and distributes NERFV points equally spaced on the interval [0, WMAXEF] with point 1 located at 0 and point NERFV at WMAXEF. Thus, the equal spacing DX will be

$$\mathrm{DX} = \frac{\mathrm{WMAXEF}}{\mathrm{NERFV-1}} \; , \tag{29}$$

and the position of point i, denoted by x_i, will be

$$x_i = (i-1)\mathrm{DX} \; . \tag{30}$$

Now, let $R(i) \equiv \mathrm{erf}(x_i)$. Then,

$$x_i = \mathrm{erf}^{-1}\,(R(i)) \tag{31}$$

(see Fig. 12). Our problem is to find $\mathrm{erf}^{-1}(t) \equiv z$ given an arbitrary t. To do this, we see that $t = \mathrm{erf}(z)$. Find an i such that

$$R(i-1) < t \leq R(i) \quad \text{or} \quad \mathrm{erf}(x_{i-1}) < t \leq \mathrm{erf}(x_i) \; .$$

Therefore, $x_{i-1} < \mathrm{erf}^{-1}(t) \leq x_i$. One linearly interpolates to get

$$\mathrm{erf}^{-1}(t) \cong x_{i-1} + \frac{(t-R(i-1))}{(R(i) - R(i-1))} (x_i - x_{i-1}) \; ,$$

or

$$\mathrm{erf}^{-1}(t) \cong \left[(i-2) + \frac{(t-R(i-1))}{(R(i) - R(i-1))}\right] \mathrm{DX} \tag{32}$$

using Eq. (30). One selects another random number t if t > R(NERFV). Because R(1) = 0 and R(NERFV) = WMAXEF using Eq. (28), y is limited to the range

$$0 \leq y \leq \text{WMAXEF} * \sqrt{4\lambda\delta t} = \text{WMAXEF} * \sqrt{2}\,\sigma \ .$$

Thus, y is limited to WMAXEF * $\sqrt{2}$ standard deviations.

Because Eq. (22) is invalid near the axis of symmetry, another cutoff parameter RMINEF has been added. If the r-coordinate of the particle is $\leq$ RMINEF, the turbulent diffusion effect is not applied. In most cases, RMINEF can be considerably less than the Δr's of the zones on the axis.

The current version of the code can be run easily with turbulence by-passed because the turbulence must be seeded initially. Most of the coding related to turbulence is excluded and no efficiency is lost if this version is used for turbulence-free calculations.

For a discussion of the output relevant to turbulence, see the sample calculation in Sec. III.

F. REZONE TECHNIQUES

The convection phase is appended to a Lagrangian calculation in a way that provides maximum flexibility for the continuous rezone. This is achieved by including the convective fluxes as functions of difference velocities,

$$\vec{u}_d = \vec{u}_{fluid} - \vec{u}_g \ ,$$

where $\vec{u}_g$ is the grid velocity with which the mesh is moved in a given calculation cycle. For an Eulerian calculation $\vec{u}_g = 0$; for a Lagrangian calculation $\vec{u}_g = \vec{u}_{fluid}$, the difference velocity vanishes, and there is no fluxing. In general the presciption to determine $\vec{u}_g$ is at the discretion of the user. A general form that we have found useful and incorporated in the present version of the code is to write

$$\vec{u}_g = \vec{u}_{fluid} + \frac{f}{\delta t} \ (\langle\vec{x}\rangle - \vec{x})$$

for each vertex. That is, the grid velocity is composed of two components: the fluid velocity and a term to relax the mesh such that each vertex is at the average position of its nearest neighbors. The latter term prevents the mesh from distorting excessively. A typical value for f might be 0.05, which would relax the mesh in approximately 20 calculational cycles if there were no fluid motion. Many variations of the relaxation component are possible. The important point is to run as near to Lagrangian as possible to minimize the smoothing, but still to maintain some degree of regularity in the mesh.

G. MIXED EQUATION OF STATE

To model more accurately the atmospheric detonation of a Mylar balloon filled with methane, it was necessary to incorporate two equations of state in one problem, one for the combustion products and one for ambient air. The method was to divide the cells into two groups at t = 0, those inside the fireball and those outside. This division was done on the basis of the specific internal energy (e); those cells with $e \geq 10^{10}$ erg/g were considered inside the fireball and all others, outside.

Define the concentration c_i as the ratio of the mass of constituent i in a given cell to the total mass of the cell. Obviously, if there are n constituents, $\sum_i^n c_i = 1$ in a given cell. For a problem like ours with only two constituents (exploded methane and ambient air), $c_{meth} + c_{air} = 1$ so that one only needs to keep track of c_{meth}. Here we use c_{meth} to refer to the concentration of combustion products and γ_{meth} to refer to the effective γ for the combustion products.

Initially, c_{meth} is defined to be 1 in all fireball zones and 0 elsewhere. As the problem proceeds, the concentration is convected into the ambient region using

$$\frac{\partial c_i}{\partial t} + \vec{u} \cdot \nabla c_i = 0 \ .$$

Note that this is just the Lagrangian form of the continuity equation.

To determine the effective γ in a given zone, we write

$$\gamma_{eff} = c_{meth}\,\gamma_{meth} + (1-c_{meth})\gamma_{air} \ .$$

The pressure is then obtained from

$$p = (\gamma_{eff}-1)\rho e \ .$$

H. SYSTEM IMPROVEMENTS

Because YAQUI is a rather large computer code, because many modifications have been made to it in the course of investigating various techniques, and because the Los Alamos Scientific Laboratory began the switch from batch to time-shared computing in the course of this research, considerable effort went into making YAQUI a convenient and flexible research tool. The various efforts involved are summarized below.

1. The original code was written in large blocks, which made it difficult to modify and also severely taxed the compilers and loaders. Thus, the code was divided into numerous smaller subroutines to avoid these problems. The concept of modularity was followed as closely as possible.
2. Because of rapidly shifting operating systems at LASL, it was important to make the code as system-independent as possible so that it could be easily switched from old to new systems. The code will now run on any of the three systems available at LASL, in either batch or time-sharing mode. It could be brought up on any other system with minimal difficulty, assuming sufficient small- and large-core storage were available.
3. Fixed format input is prone to error, whereas NAMELIST input tends to be system dependent. Thus, a NAMELIST input package was written that is system-independent for the most part, and can be easily modified for other systems.
4. Error checking within the code is meticulous. The code never assumes anything on the part of the user but monitors for errors, particularly in the setup phase, as though the user were completely unfamiliar with the code.
5. Because YAQUI runs often take several hours of CDC-7600 time, a problem may have to be run in several smaller pieces. For this reason, a flexible restart procedure was developed to allow the user to restart a problem at any point and change input parameters without having to actually modify the code itself.
6. The original YAQUI was very well documented externally in LASL report LA-5100.[3] However, a code that is under heavy development is much easier to modify if it is carefully annotated internally. Work has been proceeding on this and is largely completed.

II. DESCRIPTION OF INPUT

The input to the code, except for the problem title and the 1D fireball initialization data, is in NAMELIST form. The basic rules are:

- Each input record begins with a $ in card column (cc) 2 followed immediately by the namelist name.
- Input values are of the form

  NAME = NUMBER,

where blanks may not occur within NAME or an individual number but are ignored elsewhere. NUMBER may be a single constant or a series of comma-delimited constants. Multipliers of the form N*NUMBER are allowed but not grouping with parentheses (e.g., N*(N1, N2, N3) is illegal).

- Continuation cards are legal. NAMELIST variable names and constants may not be split across card boundaries but hollerith fields may if they end in cc80 of one card and begin again in cc1 of the next card.
- An input record is terminated by a $ anywhere on the card except for cc1 and 2.
- Variables are stored without regard to type. If one has I = 5., I will contain a floating point 5. rather than an integer 5; likewise X = 2 will cause an integer 2 to be stored in X which will most likely be interpreted as a floating point 0.
- If a "P" is punched in cc1 of the first card of a namelist record, the entire record will be printed as part of the code output.

For most information about the namelist conventions, see the internal documentation at the beginning of the routine NAMLST in the listing of the code in Sec. V.

From here on, each input record will be described, the namelist name given, and the variable names listed and discussed.

Record 1: Namelist name - START

Namelist variables

NAME	TYPE (units)	Possible values	Default
RESTRT	LOGICAL	.TRUE., .FALSE.	.TRUE.
FILM	LOGICAL	.TRUE., .FALSE.	.TRUE.
PAPER	LOGICAL	.TRUE., .FALSE.	.FALSE.
WRAPUP	REAL (s)	0.≤WRAPUP<∞	20.0

(a) RESTRT - .TRUE. if the run will be a pickup from a previous dump tape. Otherwise it is an initial setup.

(b) FILM - .TRUE. if all output will go to film.

(c) PAPER - .TRUE. if all output with the exception of plots will go to paper.

(d) WRAPUP - time to allow for the last cycle dumps, plots, printouts, and general termination procedure. If the time limit is TLIM, then the calculation will be stopped and termination begun as soon as the run time exceeds TLIM-WRAPUP.

Example: $CARDN RESTRT = .TRUE., FILM = .TRUE., PAPER = .FALSE., WRAPUP = 30.$

The next input depends on whether RESTRT is .TRUE. or .FALSE. Input based on RESTRT = .FALSE. (problem generation) will be considered first.

* * * INITIAL PROBLEM SETUP * * *

Record 2: Problem title, cc2-80 on one data card.

Record 3: Namelist name - CARDN

Namelist variables

NAME	TYPE (units)	Possible values	Default
AO	Real (none)	0.≤AO≤1.	0.1
AOFAC	Real (none)	0.<AOFAC≤.5	0.2
AOM	Real (none)	0.≤AOM≤1.	1.0
ANC	Real (none)	0.≤ANC≤.2	0.05
BO	Real (none)	0., 2.	0.0
CYL	Real (none)	0., 1.	1.0
DR	Real (cm)	0.<DR<∞	Must be specified
DT	Real (s)	0.<DT<∞	10^{-3}
DTØ	Real (s)	0.<DTØ<∞	1.0
DTØC	Real (s)	0.<DTØC<∞	10^{30}
DZ	Real (cm)	0.<DZ<∞	Must be specified
EPS	Real (none)	0.<EPS<1.	10^{-5}
FREZXR	Real (none)	1.≤FREZXR<∞	1.0
FREZYB	Real (none)	1.≤FREZYB<∞	1.0
FREZYT	Real (none)	1.≤FREZYT<∞	1.0
GR	Real (cm/s^2)	-∞<GR<∞	0.0
GRDVEL	Real (none)	0., 1., 2.	2.0
GZ	Real (cm/s^2)	-∞<GZ<∞	-980.0
GZP	Real (cm/s^2)	-∞<GZP<∞	0.0
IBAR	Integer (none)	1≤IBAR <Storage	Must be specified
IEOF	Integer (none)	0, 1	0
IST	Integer (none)	-1, 1≤IST≤ storage	-1
IUNF	Integer (none)	0≤IUNF≤IBAR	Must be specified
JBAR	Integer (none)	1≤JBAR<storage	Must be specified
JCEN	Integer (none)	1≤JCEN≤JBAR	Must be specified
JDUMP	Integer (none)	1≤JDUMP<∞	999999
JUNF	Integer (none)	0≤JUNF≤JBAR	Must be specified
KXI	Integer (none)	-1, 0, 1	-1
LAM	Real (see Ref. 3)	0.≤LAM<∞	0.6
MU	Real	0.≤MU<∞	0.0
NCLST	Integer (none)	1≤NCLST<∞	999999
NCQ	Integer (none)	-∞<NCQ<∞	-1
ØM	Real (none)	0.<ØM<2.	1.0
QLEVEL	Real (none)	.02	0.02
REXRØN	Real (gm/cm^3)	0.<REZRØN<∞	0.001
REZSIE	Real (cm^2/s^2)	0.<REZSIE<∞	$2x10^{10}$
REZYO	Real (cm)	YB<REZYO<∞	0.0
RMINEF	Real (cm)	0.≤RMINEF<∞	50.0
T	Real (s)	0.≤T<∞	0.0
TQ	Real (s)	0.≤TQ<∞	0.0
TSTRTD	Real (s)	0.≤TSTRTD<∞	1.0
TUQI	Real ($cm^2 \cdot s$)	0.<TUQI<∞	0.0
TUSI	Real (cm)	0.<TUSI≤fireball radius	0.0
TWFIN	Real (s)	0.<TWFIN<∞	1.E30
WMAXEF	Real (none)	0.<WMAXEF<∞	2.0
YB	Real (cm)	0.≤YB<∞	0.0
ZØRIG	Real (none)	0.≤ZØRIG<∞	6.0

AO - Amount of donor cell in momentum fluxing (0. is centered differencing; 1. is full donor cell).

AOFAC - Stability condition $u\Delta t/\Delta x<AOFAC$.

AOM - Amount of donor cell in mass fluxing (see AO).

ANC - Amount of node coupler (see Ref. 2).

BO - Interpolated donor cell coefficients; allows for partial cancellation of truncation errors in the convection terms.

CYL - Geometry-type switch. If CYL = 1., the calculation is done in cylindrical geometry; if CYL = 0., it is done in slab geometry.

DR - Initial value of the width of the zones in the region of uniform zoning (see Sec. I.A).

DT - Initial time step.

DTO, DTOC - Two arrays that determine when plots and long prints (edits) will occur as a function of problem time. Edits will generally occur every DTO(I) seconds in the interval DTØC(I-1) $<$ T $\leqslant$ DTØC(I). DTØC(0) $\equiv$ 0.

DZ - Same as DR except zone height (see Sec. I.A).

EPS - Convergence criterion in the pressure iteration (see Ref. 2).

FREZXR - Geometric ratio of zone Δr's in the right region of nonuniform zoning (see Sec. I.A).

FREZYB - Geometric ratio of zone Δz's in the bottom region of nonuniform zoning (see Sec. I.A).

FREZYT - Geometric ratio of zone Δz's in the top region of nonuniform zoning (see Sec. I.A).

GR - Body force acceleration in the radial direction.

GRDVEL - Type of rezone. GRDVEL = 0. is Eulerian, = 1. is Lagrangian, and =2. causes the rezone subroutine to be called.

GZ - Body force acceleration in the axial direction (usually gravity).

GZP - Particle acceleration in the axial direction (not applied unless particles have mass.)

IBAR - Number of real zones in the radial direction (see Sec. I.A).

IEOF - Input of the record 3 section is terminated by inputting a record with IEOF = 1.

IST - Number of particles whose positions are to be plotted as a function of time. If IST $\leqslant$ 0, no particles are followed.

IUNF - Number of zones in the radial direction in the region of uniform zoning (see Sec. I.A).

JBAR - Number of real zones in the axial direction (see Sec. I.A).

JCEN - Number of real zones from bottom of the problem to center of region of uniform zoning (see Sec. I.A).

JDUMP - Frequency of dumps based on cycles. Dumps will occur when the cycle number is an even multiple of JDUMP.

JUNF - Number of zones in the axial direction in region of uniform zoning; must be an even number because JUNF/2 zones will occur above and below the point defined by JCEN (see Sec. I.A).

KXI - Viscosity flag (see Ref. 3).

LAM - Viscosity parameter (see Ref. 3).

MU - Viscosity parameter (see Ref. 3.)

NCLST - Cycle number after which to terminate the run.

NCQ - Cycle number after which to seed the turbulence; if NCQ $<$ 0, the turbulence is disabled; if NCQ = 0, seeding will occur based on problem time instead of cycle number, i.e. when T=TQ (see TQ).

ØM - Relaxation parameter in the pressure iteration.

QLEVEL - Phenomenological turbulence viscosity parameter relating to specific turbulence energy (see Ref. 5).

REZRØN - Initial density of the ambient atmosphere at y = REZYO; the density of the atmosphere above and below REZYO is determined by the condition that the entire atmosphere initially be in hydrostatic equilibrium.

REZSIE - Specific internal energy of the entire ambient atmosphere.

REZYO - Center of y-coordinate of the region of uniform zoning (see Sec. I.A).

RMINEF - Particles with r$\leqslant$RMINEF are not subject to turbulent diffusion.

T - Time at which the problem begins.

TQ - Time at which to seed the turbulence if NCQ = 0 (see NCQ).

TSTRTD - Time at which to start turbulent particle diffusion if the turbulence is on.

TUQI - Proportionality constant for seeding turbulence energy. Should be chosen such that turbulence energy is a few percent of kinetic energy in any cell.

TUSI - Turbulence scale (constant over mesh). (Code could be easily changed to allow scale variation throughout mesh).

TWFIN - Time at which to terminate the run.

WMAXEF - If the turbulence and particle turbulent diffusion are on, a particle can be moved no more than WMAXEF*SQRT (4.*SIGMA*DT) because of turbulent diffusion in any one cycle (see Sec. E).

YB - Bottom of the problem mesh if the zoning is entirely uniform. If the zoning is nonuniform, YB will be calculated internally and need not be specified.

ZORIG - Number of fireball radii away from the fireball that the right problem boundary is kept.

Example:
```
$CARDN    DR = 200. $
$CARDN    DZ = 200. $
$CARDN    IBAR = 30 $
$CARDN    IUNF = 5 $
$CARDN    JBAR = 45 $
$CARDM    JCEN = 15 $
$CARDN    JUNF = 10 $
$CARDN    REZYO = 4300. $
$CARDN    REZRØN = 1.E-3 $
$CARDN    REZSIE = 1.95E9 $
$CARDN    T = .0083 $
$CARDN    TWFIN = 30. $
$CARDN    DTØ = .1, 1., 5. $
$CARDN    DTØC = 1., 10., 30. $
$CARDN    FREZYB = 1.089359 $
$CARDN    FREZYT = 1.089359 $
$CARDN    FREZXR = 1.089359 $
$CARDN    IEOF = 1 $
```

Again assuming RESTRT = .FALSE., the next record will be to define marker particles.

Record 4: Namelist name - PARTN

Namelist variables

NAME	TYPE (units)	Possible values	Default
DRPAR	Real (cm)	0.<DRPAR<∞	Must be specified
DZPAR	Real (cm)	0.<DZPAR<∞	0.
XC	Real (cm)	0.≤XC<∞	0.
XD	Real (cm)	0.<XD<∞	Must be specified
YC	Real (cm)	0.≤YC<∞	0.
YD	Real (cm)	0.≤YD<∞	0.

(a) DRPAR - Spacing between particles in the radial direction. Particle definition cards are read until one is input with DRPAR = 0.

(b) DZPAR - Spacing between particles in the axial direction.

(c) XC - (See Sec. I.C)

(d) XD - (See Sec. I.C)

(e) YC - (See Sec. I.C)

(f) YD - (See Sec. I.C)

Example:
```
$PARTN DRPAR = 100., DZPAR = 100., YC = 4300.,
      XD = 1000., YD = 0., SC = 0. $
$PARTN DRPAR = 0. $
```

The final input will be the fireball initialization data. One namelist record is needed:

Record 5: Namelist name - FIRE

Namelist variables

NAME	TYPE (units)	Possible values	Default
FBFILE	Logical	.TRUE., .FALSE.,	.FALSE.
NRAD	Integer	1≤NRAD<∞	5
NTH	Integer	1≤NTH<∞	180

(a) FBFILE - If .TRUE., the fireball initialization input will be found on logical unit 3. Otherwise, the input will follow the $FIRE namelist card.

(b) NRAD - (See Sec. I.B).

(c) NTH - (See Sec. I.B).

Example: $FIRE FBFILE = .TRUE.$

For the form of the fireball initialization data, see the section Sec. I.B.

This completes the input for an initial setup. Restart dumps are written on each edit cycle as determined by DTO and DTOC or as specified by JDUMP and go out to logical unit 8.

* * * PROBLEM RESTART * * *

To restart, a restart dump tape must be present on logical unit 7. Input record 1 must have RESTRT = .TRUE. Next follows a namelist record telling from which dump to restart.

$RCYCLE INTCYC = N $

where N is either the cycle on the dump tape from which one wishes to restart or -1. In the latter case, restart occurs from the last dump on logical unit 7. Last, records of the type described under input record 3 may appear if one desires to override any of the parameter values in effect at the time the problem is being restarted. This section is terminated by $CARDN IEOF = 1 $.

Example:
```
$START RESTART = .TRUE., PAPER = .FALSE.,
      FILM = .TRUE., WRAPUP = 60. $
$RCYCLE    INTCYC = -1 $
$CARDN     TWFIN = 30. $
```

```
$CARDN      NCLST = 99999 $
$CARDN      IEØF = 1 $
```

III. RESULTS OF A SAMPLE CALCULATION

A careful comparison of a YAQUI calculation with experiment was made and has been reported in Ref. 6, Fig. 1(c). In that simulation the mixed equation-of-state, turbulence, and turbulent particle diffusion options were all used. To help the user to understand the output options and to provide a comparison calculation, we include detailed results of a sample calculation, patterned after the simulation in Ref. 6.

In Sec. A the input parameters are listed along with the detailed one-dimensional fireball input data. Also given are the initial marker particle configurations, the initial grid for the complete mesh and for a smaller region surrounding the fireball, and the initial velocities.

In Sec. B contour plots of the vorticity and the specific turbulence energy at t = .5 s (immediately following the seeding of the turbulence) are given. Note that the general shapes of the two contour plots are similar because the turbulence energy is seeded proportional to the vorticity.

In Sec. C the positions of the marker particles and contour plots of the specific internal energy and the specific turbulence energy are given at t = 3 s at which time all memory of the seeding is gone. Note that the regions of greatest specific internal energy closely coincide with the regions of greatest specific turbulence energy. Also note the toroidal form of these contour plots, remembering that the left side of the mesh is an axis of cylindrical symmetry.

In Sec. D complete graphical output at t = 10 s is given. This is a moderately late time because torus formation occurs at $\sim t = 1.25$ s.

A. Initial Conditions

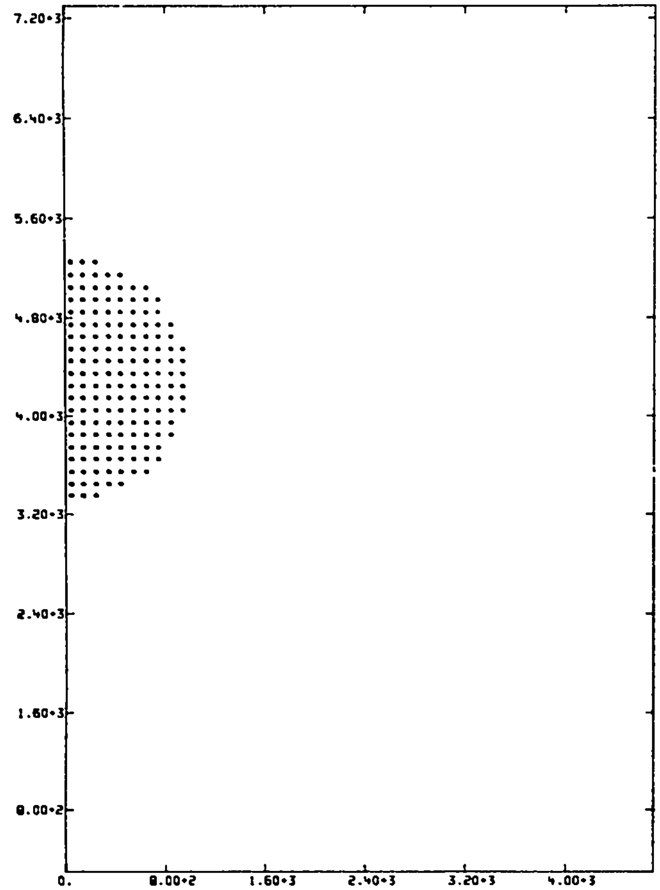
```
BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TPBCOR
THERE WILL BE TURBULENCE
QLEVEL= 2.00000E-02
  TUOI= 5.00000E+00
  TUSI= 1.00000E+03
   NCO=    0
    TO= 5.00000E-01
TSTRTD= 1.00000E+00
WMAXEF= 2.00000E+00
RMINEF= 5.00000E+01
  IBAR=  30
  JBAR=  45
  IUNF=   5
  JUNF=  10
  JCEN=  15
    DR= 2.00000E+02
    DZ= 2.00000E+02
   CYL= 1.00000E+00
GROVEL= 2.00000E+00
    AO= 1.00000E-01
   AOM= 1.00000E+00
    BO= 0.
   KXI= -1
    MU= 0.
   LAM= 6.00000E-01
    OM= 1.00000E+00
   EPS= 1.00000E-04
    GR= 0.
    GZ=-9.80000E+02
FREZXR= 1.08936E+00
FREZYT= 1.08936E+00
FREZYB= 1.08936E+00
 ZORIG= 6.00000E+00
    YB= 0.
 REZYO= 4.30000E+03
REZRON= 1.00000E-03
REZSIE= 1.95930E+09
   GZP=-9.80000E+02
     T= 8.30000E-03
    DT= 1.00000E-03
 NCLST=      1
 TWFIN= 3.00000E+01
 PAPER=0
  FILM=1
   ANC= 2.00000E-01
 AOFAC= 2.00000E-01
   IST=  10
 DTO(1-10)=   1.00000E-01   1.00000E+00   5.00000E+00   0.
              0.            0.            0.            0.
              0.            0.
DTOC(1-10)=   1.00000E+00   1.00000E+01   3.00000E+01   0.
              0.            0.            0.            0.
              0.            0.
PSPARTN DRPAR=100.,DZPAR=100.,YC=4300.,XD=1000.,YD=0.,XC=0. $
 DRPAR= 1.00000E+02  DZPAR= 1.00000E+02  XC= 0.
 YC= 4.30000E+03  XD= 1.00000E+03  YD= 0.
PSPARTN DRPAR=0.$
      150 PARTICLES GENERATED, WITH TOTAL MASS= 0.
PSFIRE FBFILE=.TRUE.  $

FIREBALL INPUT WILL BE FROM FILE 3
```

MESHMKR FIREBALL INPUT

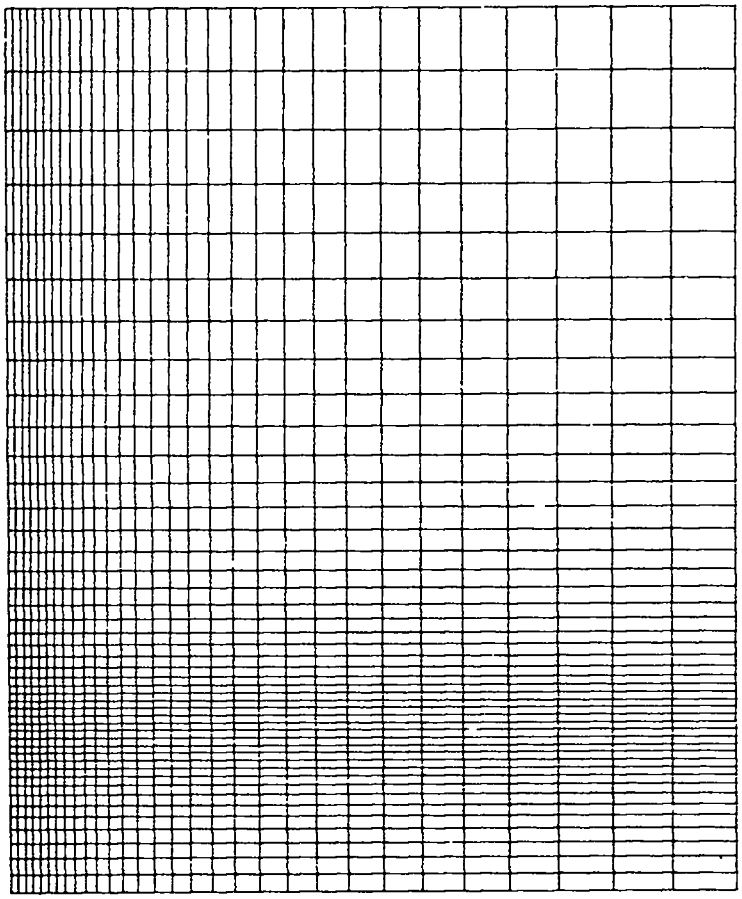
U	E	RHO	X	P	PRHO
3.7424474E+03	7.1150927E+10	1.9859535E-04	8.5614114E+00	1.5978265E+06	1.5978265E+06
7.4272781E+03	6.8662873E+10	2.0756046E-04	1.6904335E+01	1.6002742E+06	1.6002742E+06
1.1017913E+04	6.5330351E+10	2.2051980E-04	2.4969682E+01	1.6034271E+06	1.6034271E+06
1.4597835E+04	6.4165456E+10	2.2570703E-04	3.3018635E+01	1.6076463E+06	1.6076463E+06
1.8224794E+04	6.4177399E+10	2.2653017E-04	4.1125711E+01	1.6141628E+06	1.6141628E+06
2.1852264E+04	6.4367879E+10	2.2683715E-04	4.9262863E+01	1.6222339E+06	1.6222338E+06
2.5396177E+04	6.4492809E+10	2.2752965E-04	5.7399511E+01	1.6312426E+06	1.6312427E+06
2.8837839E+04	6.4526913E+10	2.2857219E-04	6.5519210E+01	1.6401362E+06	1.6401363E+06
3.2108888E+04	6.4551471E+10	2.2948266E-04	7.3625620E+01	1.6477522E+06	1.6477521E+06
3.5145431E+04	6.4572304E+10	2.3008191E-04	8.1727692E+01	1.6529134E+06	1.6529134E+06
3.7904370E+04	6.4562536E+10	2.3026145E-04	8.9836791E+01	1.6539651E+06	1.6539652E+06
4.0391666E+04	6.4549827E+10	2.2985509E-04	9.7968698E+01	1.6505087E+06	1.6505087E+06
4.2650180E+04	6.4516297E+10	2.2889916E-04	1.0613698E+02	1.6422730E+06	1.6422731E+06
4.4720316E+04	6.4461676E+10	2.2750392E-04	1.1435128E+02	1.6300991E+06	1.6300991E+06
4.6645869E+04	6.4399621E+10	2.2566496E-04	1.2262183E+02	1.6143934E+06	1.6143934E+06
4.8464773E+04	6.4338451E+10	2.2346126E-04	1.3095664E+02	1.5960185E+06	1.5960184E+06
5.0198018E+04	6.4273095E+10	2.2093046E-04	1.3936314E+02	1.5751189E+06	1.5751189E+06
5.1872022E+04	6.4181305E+10	2.1824896E-04	1.4784309E+02	1.5523834E+06	1.5523834E+06
5.3482969E+04	6.4085894E+10	2.1536178E-04	1.5640135E+02	1.5280966E+06	1.5280566E+06
5.5049004E+04	6.3961910E+10	2.1241419E-04	1.6503797E+02	1.5026514E+06	1.5026514E+06
5.6562111E+04	6.3847340E+10	2.0934358E-04	1.7375626E+02	1.4766793E+06	1.4766792E+06
5.8020348E+04	6.3743834E+10	2.0613250E-04	1.8256091E+02	1.4500901E+06	1.4300901E+06
5.9456217E+04	6.3651833E+10	2.0281039E-04	1.9145594E+02	1.4231047E+06	1.4231047E+06
6.0876553E+04	6.3539957E+10	1.9949170E-04	2.0044101E+02	1.3957321E+06	1.3957321E+06
6.2206853E+04	6.3420689E+10	1.9604025E-04	2.0952238E+02	1.3673251E+06	1.3673251E+06
6.3716637E+04	6.3299366E+10	1.9242442E-04	2.1870822E+02	1.3378075E+06	1.3378075E+06
6.5175153E+04	6.3154846E+10	1.8880368E-04	2.2799977E+02	1.3078363E+06	1.3078363E+06
6.6618302E+04	6.3017089E+10	1.8520866E-04	2.3739724E+02	1.2783904E+06	1.2783904E+06
6.7955210E+04	6.2877213E+10	1.8182404E-04	2.4689197E+02	1.2505825E+06	1.2505826E+06
6.9030814E+04	6.2764354E+10	1.7879916E-04	2.5646779E+02	1.2261441E+06	1.2261441E+06
6.9486811E+04	6.2702621E+10	1.7657756E-04	2.6608521E+02	1.2087481E+06	1.2087481E+06
6.7215425E+04	6.2734345E+10	1.7697051E-04	2.7561025E+02	1.2122939E+06	1.2122939E+06
6.4305665E+04	6.2947556E+10	1.8224201E-04	2.8480948E+02	1.2552554E+06	1.2552554E+06
6.0032417E+04	6.3260777E+10	1.9058565E-04	2.9358387E+02	1.3234276E+06	1.3234276E+06
5.7840194E+04	6.3445534E+10	1.9650938E-04	3.0209328E+02	1.3713999E+06	1.3713959E+06
5.5190905E+04	6.3680516E+10	2.0362195E-04	3.1031970E+02	1.4298534E+06	1.4298534E+06
5.3192895E+04	6.3895024E+10	2.0939162E-04	3.1834492E+02	1.4783551E+06	1.4783551E+06
5.2593687E+04	6.3963190E+10	2.1131643E-04	3.2632708E+02	1.4945345E+06	1.4945345E+06
5.3249521E+04	6.3951020E+10	2.1110238E-04	3.3434614E+02	1.4926047E+06	1.4926047E+06
5.1030180E+04	6.4054493E+10	2.1455134E-04	3.4226483E+02	1.5211647E+06	1.5211647E+06
5.0466455E+04	6.4165887E+10	2.1804764E-04	3.5008782E+02	1.5504324E+06	1.5504324E+06
5.0581713E+04	6.4178669E+10	2.1814622E-04	3.5793869E+02	1.5515396E+06	1.5515396E+06
4.9911185E+04	6.4211336E+10	2.1860081E-04	3.6580240E+02	1.5558857E+06	1.5558857E+06
4.8433146E+04	6.4357229E+10	2.2347513E-04	3.7352467E+02	1.5966836E+06	1.5966836E+06
4.8435327E+04	6.4422836E+10	2.2622082E-04	3.8118584E+02	1.6192717E+06	1.6192717E+06
4.8464106E+04	6.4355941E+10	2.2435782E-04	3.8894160E+02	1.6032665E+06	1.6032665E+06
4.7011089E+04	6.4466569E+10	2.2775749E-04	3.9661081E+02	1.6321559E+06	1.6321559E+06
4.7051374E+04	6.4576961E+10	2.3070950E-04	4.0421261E+02	1.6577911E+06	1.6577911E+06
4.6617931E+04	6.4588768E+10	2.3044152E-04	4.1185343E+02	1.6561355E+06	1.6561355E+06
4.6103550E+04	6.4635776E+10	2.3150648E-04	4.1948751E+02	1.6656348E+06	1.6656348E+06
4.5816445E+04	6.4727400E+10	2.3414791E-04	4.2706375E+02	1.6884806E+06	1.6884806E+06
4.5986287E+04	6.4717721E+10	2.3457330E-04	4.3465430E+02	1.6913946E+06	1.6913947E+06
4.5311040E+04	6.4753353E+10	2.3678506E-04	4.4220132E+02	1.7092774E+06	1.7092774E+06
4.4403097E+04	6.4803890E+10	2.3893740E-04	4.4970803E+02	1.7272210E+06	1.7272210E+06
4.4513807E+04	6.4818062E+10	2.3840572E-04	4.5725828E+02	1.7236436E+06	1.7236436E+06
4.3975257E+04	6.4856745E+10	2.3947666E-04	4.6480005E+02	1.7330244E+06	1.7330244E+06
4.4189199E+04	6.4935133E+10	2.4178134E-04	4.7229509E+02	1.7531010E+06	1.7531010E+06
4.4115429E+04	6.4970164E+10	2.4215048E-04	4.7980360E+02	1.7570601E+06	1.7570601E+06
4.3098205E+04	6.5021432E+10	2.4418424E-04	4.8727391E+02	1.7742509E+06	1.7742509E+06
4.3193302E+04	6.5021735E+10	2.4496571E-04	4.9474452E+02	1.7802213E+06	1.7802213E+06
4.2798233E+04	6.5018733E+10	2.4507802E-04	5.0223485E+02	1.7809781E+06	1.7809781E+06
4.2527414E+04	6.5105305E+10	2.4782966E-04	5.0966485E+02	1.8048870E+06	1.8048871E+06
4.2927887E+04	6.5112861E+10	2.4784693E-04	5.1711709E+02	1.8052738E+06	1.8052738E+06
4.1846138E+04	6.5158236E+10	2.4909695E-04	5.2455387E+02	1.8163692E+06	1.8163692E+06
4.1873625E+04	6.5227982E+10	2.5115615E-04	5.3195161E+02	1.8345182E+06	1.8345182E+06
4.1974613E+04	6.5208243E+10	2.5011419E-04	5.3940142E+02	1.8258585E+06	1.8258586E+06
4.1328358E+04	6.5270835E+10	2.5297217E-04	5.4678771E+02	1.8499209E+06	1.8499209E+06
4.1553763E+04	6.5245348E+10	2.5320570E-04	5.5418794E+02	1.8508328E+06	1.8508328E+06
4.0920658E+04	6.5274917E+10	2.5375363E-04	5.6159214E+02	1.8560602E+06	1.8560602E+06
4.0956392E+04	6.5353104E+10	2.5634045E-04	5.6894154E+02	1.8786608E+06	1.8786608E+06
4.0641076E+04	6.5335144E+10	2.5521024E-04	5.7634304E+02	1.8693426E+06	1.8693426E+06
4.0362453E+04	6.5400648E+10	2.5747003E-04	5.8369842E+02	1.8890239E+06	1.8890239E+06
4.0562110E+04	6.5413439E+10	2.5821800E-04	5.9105143E+02	1.8952354E+06	1.8952354E+06
3.9861654E+04	6.5453692E+10	2.5967806E-04	5.9838179E+02	1.9079145E+06	1.9079145E+06
3.9997421E+04	6.5501863E+10	2.6124194E-04	6.0568701E+02	1.9216991E+06	1.9216991E+06
3.9680666E+04	6.5503054E+10	2.6103773E-04	6.1301629E+02	1.9201654E+06	1.9201654E+06
3.9457735E+04	6.5552986E+10	2.6308612E-04	6.2030646E+02	1.9377829E+06	1.9377829E+06
3.9014056E+04	6.5516869E+10	2.6230650E-04	6.2763591E+02	1.9304576E+06	1.9304576E+06
3.8699303E+04	6.5578593E+10	2.6489805E-04	6.3491089E+02	1.9527217E+06	1.9527217E+06
3.8579594E+04	6.5622856E+10	2.6553749E-04	6.4218579E+02	1.9592838E+06	1.9592838E+06
3.8210083E+04	6.5714994E+10	2.6768740E-04	6.4941965E+02	1.9793262E+06	1.9793262E+06
3.8109627E+04	6.5750264E+10	2.6842126E-04	6.5665120E+02	1.9863244E+06	1.9863245E+06
3.7523874E+04	6.5771740E+10	2.7008194E-04	6.6385559E+02	2.0000209E+06	2.0000209E+06
3.7602140E+04	6.5755132E+10	2.7074826E-04	6.7105946E+02	2.0045789E+06	2.0045789E+06
3.6981140E+04	6.5783460E+10	2.7230005E-04	6.7823932E+02	2.0176998E+06	2.0176998E+06
3.7180971E+04	6.5787036E+10	2.7244616E-04	6.8543212E+02	2.0189702E+06	2.0189702E+06
3.6557407E+04	6.5836225E+10	2.7429801E-04	6.9259292E+02	2.0352341E+06	2.0352341E+06
3.6906060E+04	6.5872888E+10	2.7459565E-04	6.9976241E+02	2.0389416E+06	2.0389416E+06
3.6368277E+04	6.5919721E+10	2.7611735E-04	7.0690859E+02	2.0525859E+06	2.0525859E+06
3.6582275E+04	6.5934307E+10	2.7626351E-04	7.1406698E+02	2.0542831E+06	2.0542831E+06
3.6123990E+04	6.5959950E+10	2.7761017E-04	7.2120632E+02	2.0657759E+06	2.0657758E+06
3.6115551E+04	6.5964713E+10	2.7761170E-04	7.2836106E+02	2.0659707E+06	2.0659707E+06
3.5802443E+04	6.5995075E+10	2.7909724E-04	7.3549285E+02	2.0787447E+06	2.0787447E+06
3.6013123E+04	6.6006473E+10	2.7923670E-04	7.4263603E+02	2.0802752E+06	2.0802752E+06
3.5609793E+04	6.6045790E+10	2.8096965E-04	7.4974994E+02	2.0953533E+06	2.0953533E+06
3.5693239E+04	6.6046143E+10	2.8063880E-04	7.5688681E+02	2.0927784E+06	2.0927784E+06
3.5453286E+04	6.6093776E+10	2.8168492E-04	7.6401134E+02	2.1028242E+06	2.1028242E+06
3.5536530E+04	6.6087005E+10	2.8229404E-04	7.7113453E+02	2.1073302E+06	2.1073301E+06
3.5448353E+04	6.6108558E+10	2.8312804E-04	7.7825054E+02	2.1147091E+06	2.1147091E+06
3.5341065E+04	6.6143186E+10	2.8356259E-04	7.8536924E+02	2.1194773E+06	2.1194773E+06
3.5256959E+04	5.5309900E+09	9.9623870E-04	7.8768562E+02	2.1283584E+06	2.1283584E+06
3.5209144E+04	4.9053420E+09	1.1137507E-03	7.8979133E+02	2.1276443E+06	2.1276444E+06
3.5156540E+04	4.4972234E+09	1.2122925E-03	7.9175781E+02	2.1346623E+06	2.1346623E+06
3.5112162E+04	4.2609157E+09	1.2808223E-03	7.9364989E+02	2.1435098E+06	2.1435098E+06
3.5132002E+04	4.1209439E+09	1.3222266E-03	7.9551292E+02	2.1440803E+06	2.1440802E+06

3.5115795E+04	4.0395768E+09	1.3544449E-03	7.9736131E+02	2.1552814E+06	2.1552813E+06
3.4998455E+04	3.9827654E+09	1.3797260E-03	7.9920512E+02	2.1662647E+06	2.1662647E+06
3.4980834E+04	3.9315579E+09	1.3976571E-03	8.0105426E+02	2.1676758E+06	2.1676757E+06
3.4924712E+04	3.8864795E+09	1.4152509E-03	8.0290913E+02	2.1710943E+06	2.1710943E+06
3.4866744E+04	3.8463971E+09	1.4336627E-03	8.0476857E+02	2.1778156E+06	2.1778156E+06
3.4872806E+04	3.8071720E+09	1.4498187E-03	8.0663544E+02	2.1810343E+06	2.1810342E+06
3.4862563E+04	3.7716673E+09	1.4672379E-03	8.0850801E+02	2.1876866E+06	2.1876866E+06
3.4783107E+04	3.7398261E+09	1.4867516E-03	8.1038354E+02	2.1989978E+06	2.1989979E+06
3.4769346E+04	3.7058058E+09	1.5027809E-03	8.1226638E+02	2.2034832E+06	2.2034833E+05
3.4666535E+04	3.6728596E+09	1.5182728E-03	8.1415708E+02	2.2073736E+06	2.2073738E+06
3.4676816E+04	3.6431271E+09	1.5351190E-03	8.1605383E+02	2.2146745E+06	2.2146745E+06
3.4695951E+04	3.6143614E+09	1.5508051E-03	8.1795796E+02	2.2204886E+06	2.2204887E+06
3.4627353E+04	3.5875752E+09	1.5664984E-03	8.1986934E+02	2.2271302E+06	2.2271301E+06
3.4562078E+04	3.5642181E+09	1.5839971E-03	8.2178567E+02	2.2380425E+06	2.2380425E+06
3.4547129E+04	3.5377923E+09	1.5961118E-03	8.2371334E+02	2.2392271E+06	2.2392271E+06
3.4452210E+04	3.5149800E+09	1.6104773E-03	8.2564947E+02	2.2454942E+06	2.2454942E+06
3.4414207E+04	3.4931143E+09	1.6243718E-03	8.2759450E+02	2.2514340E+06	2.2514340E+06
3.4407332E+04	3.4716822E+09	1.6376464E-03	8.2954901E+02	2.2565507E+06	2.2565508E+06
3.4273689E+04	3.4512874E+09	1.6514498E-03	8.3151221E+02	2.2628176E+06	2.2628177E+06
3.4347673E+04	3.4319637E+09	1.6660401E-03	8.3348305E+02	2.2706127E+06	2.2706127E+06
3.4298110E+04	3.4124192E+09	1.6797741E-03	8.3546239E+02	2.2768864E+06	2.2768863E+06
3.4227273E+04	3.3957254E+09	1.6962041E-03	8.3744694E+02	2.2884185E+06	2.2884165E+06
3.4230127E+04	3.3768874E+09	1.7087326E-03	8.3944114E+02	2.2931063E+06	2.2931083E+06
3.4156343E+04	3.3581646E+09	1.7198944E-03	8.4144643E+02	2.2958637E+06	2.2958637E+06
3.4057786E+04	3.3430190E+09	1.7342046E-03	8.4345899E+02	2.3049913E+06	2.3049912E+06
3.4071515E+04	3.3247178E+09	1.7431560E-03	8.4548488E+02	2.3047679E+06	2.3047679E+06
3.3930328E+04	3.3103217E+09	1.7562606E-03	8.4751913E+02	2.3124840E+06	2.3124840E+06
3.3950430E+04	3.2954187E+09	1.7678914E-03	8.4956328E+02	2 3177797E+06	2.3177797E+06
3.3846926E+04	3.2801693E+09	1.7781635E-03	8.5161876E+02	2.3209315E+06	2.3209315E+06
3.3784672E+04	3.2680065E+09	1.7916204E-03	8.5368172E+02	2.3302033E+06	2.3302033E+06
3.3775590E+04	3.2527541E+09	1.7997060E-03	8.5575821E+02	2.3302694E+06	2.3302694E+06
3.3698677E+04	3.2404786E+09	1.8126121E-03	8.5784249E+02	2.3385065E+06	2.3385064E+06
3.3683397E+04	3.2296884E+09	1.8283332E-03	8.5993124E+02	2.3512732E+06	2.3512731E+06
3.3659585E+04	3.2163362E+09	1.8332907E-03	8.6203661E+02	2.3483203E+06	2.3483203E+06
3.3621172E+04	3.2124636E+09	1.8490261E-03	8.6414611E+02	2.3657470E+06	2.3657470E+06
3.3695086E+04	3.1997525E+09	1.8570781E-03	8.6626839E+02	2.3670495E+06	2.3670495E+06
3.3618831E+04	3.1921169E+09	1.8726053E-03	8.6839476E+02	2.3813877E+06	2.3813878E+06
3.3629543E+04	3.1841296E+09	1.8860084E-03	8.7052755E+02	2.3926864E+06	2.3926864E+06
3.3708649E+04	3.1737867E+09	1.8946077E-03	8.7267204E+02	2.3961195E+06	2.3961195E+06
3.3620982E+04	3.1706612E+09	1.9131760E-03	8.7481687E+02	2.4173211E+06	2.4173211E+06
3.3787943E+04	3.1587356E+09	1.9193687E-03	8.7697583E+02	2.4164093E+06	2.4164092E+06
3.3717361E+04	3.1578656E+09	1.9417212E-03	8.7913071E+02	2.4439052E+06	2.4439052E+06
3.3862282E+04	3.1488634E+09	1.9504843E-03	8.8129657E+02	2.4482309E+06	2.4482309E+06
3.3866510E+04	3.1454366E+09	1.9666511E-03	8.8346509E+02	2.4659499E+06	2.4659499E+06
3.3907936E+04	3.1414258E+09	1.9823665E-03	8.8563670E+02	2.4826188E+06	2.4826187E+06
3.4078515E+04	3.1333618E+09	1.9918356E-03	8.8781813E+02	2.4883422E+06	2.4883423E+06
3.4058596E+04	3.1355854E+09	2.0170949E-03	8.8999213E+02	2.5216114E+06	2.5216113E+06
3.4281901E+04	3.1272544E+09	2.0241597E-03	8.9217833E+02	2.5240010E+06	2.5240010E+06
3.4228645E+04	3.1288303E+09	2.0477889E-03	8.9435885E+02	2.5546981E+06	2.5546981E+06
3.4365292E+04	3.1217103E+09	2.0566002E-03	8.9654947E+02	2.5600958E+06	2.5600957E+06
3.4442066E+04	3.1202787E+09	2.0746491E-03	8.9874027E+02	2.5814284E+06	2.5814284E+06
3.4593747E+04	3.1202340E+09	2.0949301E-03	9.0092891E+02	2.6066277E+06	2.6066277E+06
3.4700485E+04	3.1154214E+09	2.1060612E-03	9.0312491E+02	2.6166042E+06	2.6166042E+06
3.4808810E+04	3.1192590E+09	2.1317003E-03	9.0531318E+02	2.6515849E+06	2.6515850E+06
3.4918872E+04	3.1137531E+09	2.1415689E-03	9.0750996E+02	2.6593540E+06	2.6593540E+06
3.4957665E+04	3.1151226E+09	2.1624276E-03	9.0970393E+02	2.6863877E+06	2.6863878E+06
3.5176825E+04	3.1124720E+09	2.1761624E-03	9.1190230E+02	2.7012459E+06	2.7012459E+06
3.5154021E+04	3.1126938E+09	2.1952162E-03	9.1409967E+02	2.7250833E+06	2.7250833E+06
3.5429866E+04	3.1118803E+09	2.2117616E-03	9.1629852E+02	2.7449346E+06	2.7449346E+06
3.5468387E+04	3.1129494E+09	2.2308588E-03	9.1849631E+02	2.7695470E+06	2.7695470E+06
3.5573108E+04	3.1141848E+09	2.2507700E-03	9.2069225E+02	2.7953290E+06	2.7953290E+06
3.5649490E+04	3.1095271E+09	2.2594536E-03	9.2289725E+02	2.8044427E+06	2.8020911E+06
3.5794816E+04	3.1132518E+09	2.2829034E-03	9.2509689E+02	2.8368151E+06	2.8344228E+06
3.5950417E+04	3.1114875E+09	2.2960763E-03	9.2730109E+02	2.8516419E+06	2.8492257E+06
3.5974976E+04	3.1154454E+09	2.3194394E-03	9.2950009E+02	2.8841831E+06	2.8817300E+06
3.6258650E+04	3.1128822E+09	2.3301621E-03	9.3170586E+02	2.8952379E+06	2.8927694E+06
3.6295192E+04	3.1196232E+09	2.3582757E-03	9.3390203E+02	2.9362663E+06	2.9337464E+06
3.6532198E+04	3.1187001E+09	2.3716634E-03	9.3610240E+02	2.9521055E+06	2.9495643E+06
3.6619990E+04	3.1221198E+09	2.3930645E-03	9.3829951E+02	2.9818866E+06	2.9793074E+06
3.6891669E+04	3.1263956E+09	2.4161084E-03	9.4049193E+02	3.0145649E+06	3.0119439E+06
3.6923684E+04	3.1279237E+09	2.4336871E-03	9.4268468E+02	3.0379302E+06	3.0352785E+06
3.7133722E+04	3.1311942E+09	2.4544207E-03	9.4487490E+02	3.0668931E+06	3.0542040E+06
3.7205328E+04	3.1316120E+09	2.4694429E-03	9.4706768E+02	3.0860672E+06	3.0833525E+06
3.7392724E+04	3.1372589E+09	2.4946243E-03	9.4925406E+02	3.1229372E+06	3.1201751E+06
3.7070561E+04	3.1326099E+09	2.4990925E-03	9.5145219E+02	3.1240914E+06	3.1213258E+06
3.5154104E+04	3.0880550E+09	2.4253868E-03	9.5373301E+02	2.9905668E+06	2.9879629E+06
3.0214252E+04	2.9400488E+09	2.1758195E-03	9.5629218E+02	2.5570943E+06	2.5570942E+06
2.2358707E+04	2.6744876E+09	1.8011065E-03	9.5940127E+02	1.9323934E+06	1.9323934E+06
1.3295307E+04	2.3712042E+09	1.4568163E-03	9.6326144E+02	1.3914247E+06	1.3914247E+06
5.5841253E+03	2.1348384E+09	1.2188572E-03	9.6788737E+02	1.0514459E+06	1.0514459E+06
1.3584421E+03	2.0100851E+09	1.0881251E-03	9.7307547E+02	8.8530622E+05	8.8530627E+05
2.0436807E+02	1.9688586E+09	1.0391039E-03	9.7851043E+02	8.2854402E+05	8.2854400E+05
2.5655060E+01	1.9606208E+09	1.0286106E-03	9.8400125E+02	8.1603626E+05	8.1603625E+05
2.9506340E+00	1.9594630E+09	1.0271156E-03	9.8950013E+02	8.1518015E+05	8.1518013E+05
1.4687131E-01	1.9593191E+09	1.0269294E-03	9.9500000E+02	8.1497412E+05	8.1497408E+05
0.	1.9593019E+09	1.0269072E-03	1.0005000E+03	8.1494950E+05	8.1494950E+05

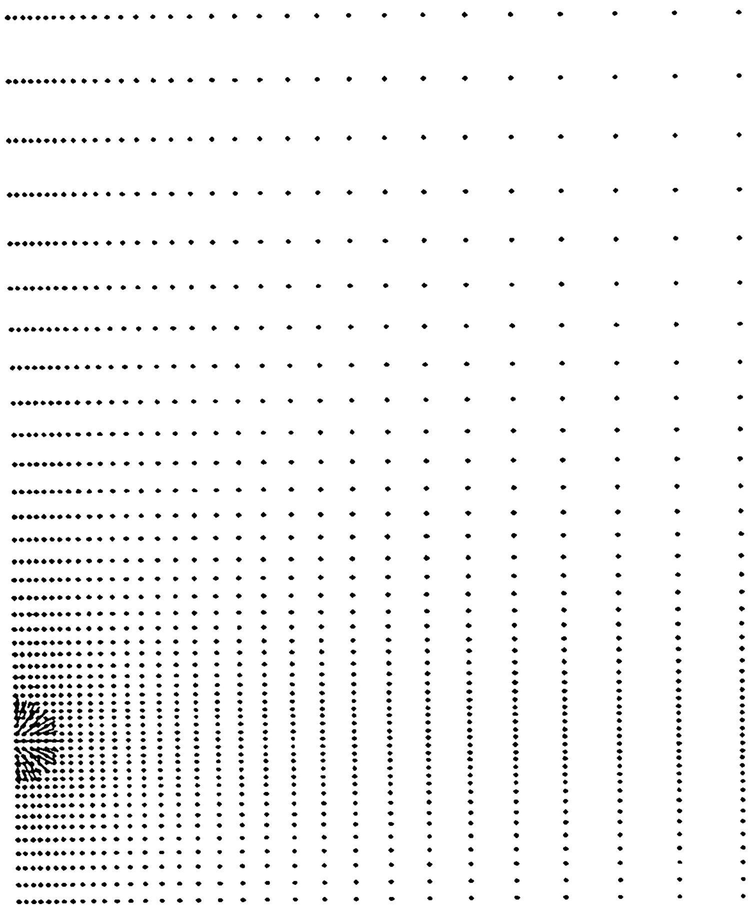
```
ALL FIREBALL INPUT DATA READ

CELL     1   12 HAS A RELATIVE VOLUME ERROR OF -4.03783E-02.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     2   12 HAS A RELATIVE VOLUME EPROR OF -2.57207E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CEL.     3   12 HAS A RELATIVE VOLUME ERROR OF -6.99058E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     4   13 HAS A RELATIVE VOLUME ERROR OF -4.72253E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     5   14 HAS A RELATIVE VOLUME ERROR OF -7.00923E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     5   15 HAS A RELATIVE VOLUME ERROR OF -2.60044E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     5   16 HAS A RELATIVE VOLUME ERROR OF -3.06375E-02.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     5   17 HAS A RELATIVE VOLUME ERROR OF -3.06375E-02.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     5   18 HAS A RELATIVE VOLUME ERROR OF -2.60044E-01.  IT WILL BE TREATED AS A FIREBALL BOUNCARY CELL
CELL     5   19 HAS A RELATIVE VOLUME ERRCR OF -7.00923E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     4   20 HAS A RELATIVE VOLUME ERROR OF -4.72253E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     1   21 HAS A RELATIVE VOLUME ERROR OF -4.03783E-02.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     2   21 HAS A RELATIVE VOLUME ERROR OF -2.57207E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
CELL     3   21 HAS A RELATIVE VOLUME ERROR OF -6.99058E-01.  IT WILL BE TREATED AS A FIREBALL BOUNDARY CELL
****** CYCLE     0. T= 8.30000E-03. DT= 1.00000E-04. CP= 2.25698E+01
                 GRINDS= 1.63643E-02. NUMIT=  0. CIRC= 3.10142E+02
                 DTV= 0.          . IDTV=  0. JDTV=  0
                 DTC= 0.          . IDTC=  0. JDTC=  0
                 TMAX= 6.54503E+10. ITM=  2. JTM= 14. XTMAX= 0.        YTMAX= 0.
                 TGMX= 2.47278E+08. ITG=  3. JTG= 18
                 PRIT= 1.000E+03. PTOP= 5.300E+03. PBOT= 3.300E+03. PDIAM= 2.000E+03. PAVHT= 4.300E+03
                 TOTAL INTERNAL ENERGY =  8.5132171E+18
                 TOTAL KINETIC ENERGY =  2.2074742E+14
                 TOTAL GRAV. POTENTIAL ENERGY =  4.6657320E+16
                 TOTAL RADIAL MOMENTUM =  1.1257216E+10
                 TOTAL AXIAL MOMENTUM =  2.1886826E-04
                 00000000000000000000     00000000000000000000
                 VMAX =  4.86465E+04 AT VERTEX     2   17
```



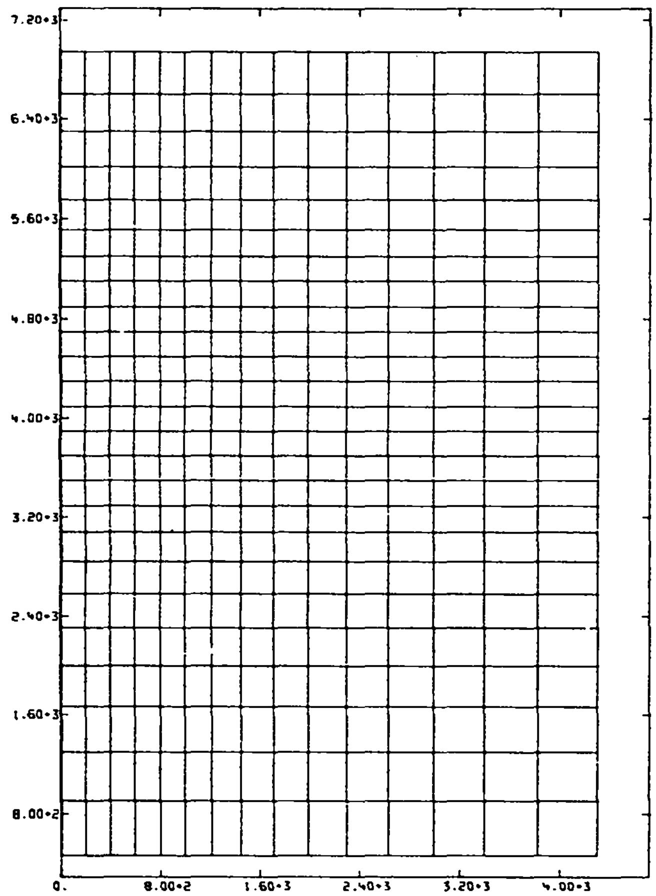

```
PARTICLES
        PXR= 1.92794E+04 PYB=-1.38538E-03 PYT= 2.35794E+04
   YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR          T= 8.30000E-03 CYCLE=    0
```



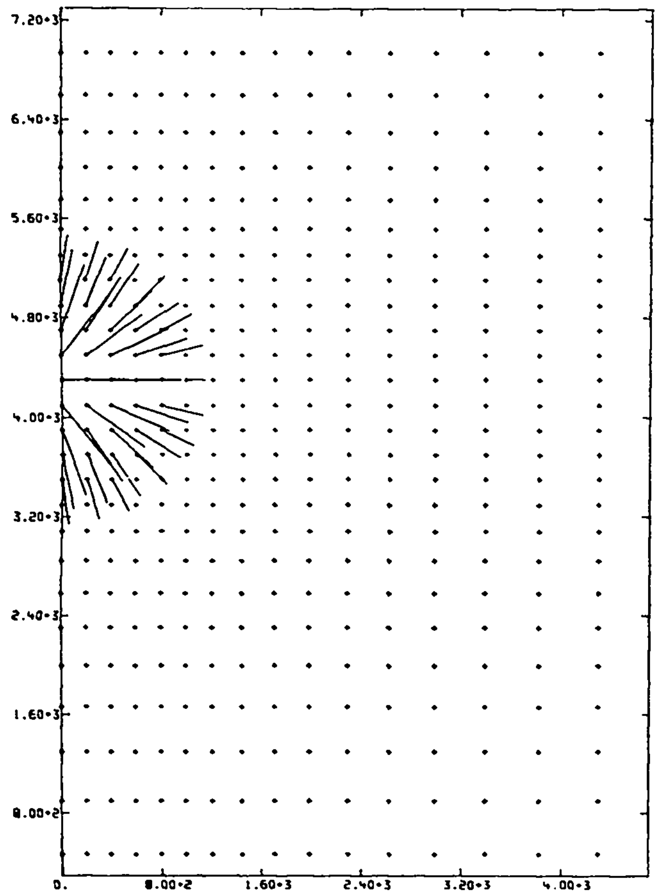
ALL ZONES
DRMIN= 2.00000E+02 DRMAX= 1.69944E+03 DZMIN= 2.00000E+02
DZMAX= 1.69944E+03 XR= 1.92794E+04 YB=-1.38538E-03 YT= 2.35794E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TRBCOR   T= 8.30000E-03 CYCLE= 0



FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 4.86465E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E-03 CYCLE= 0

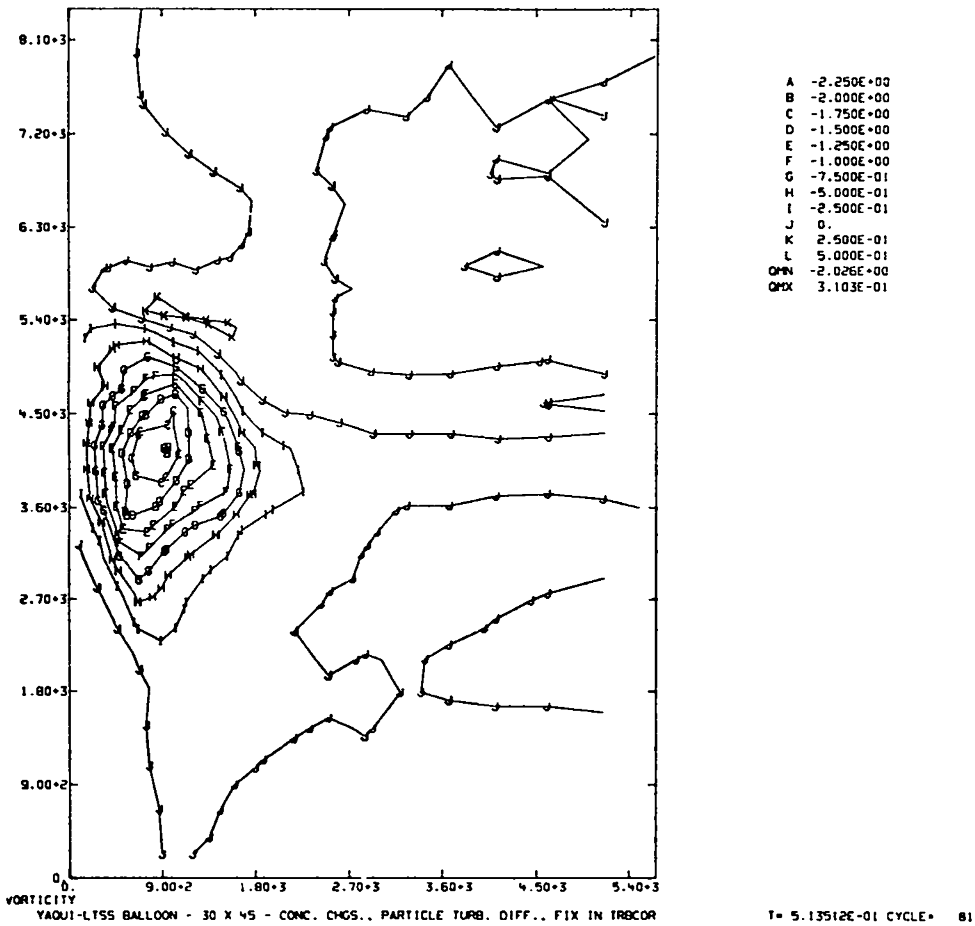


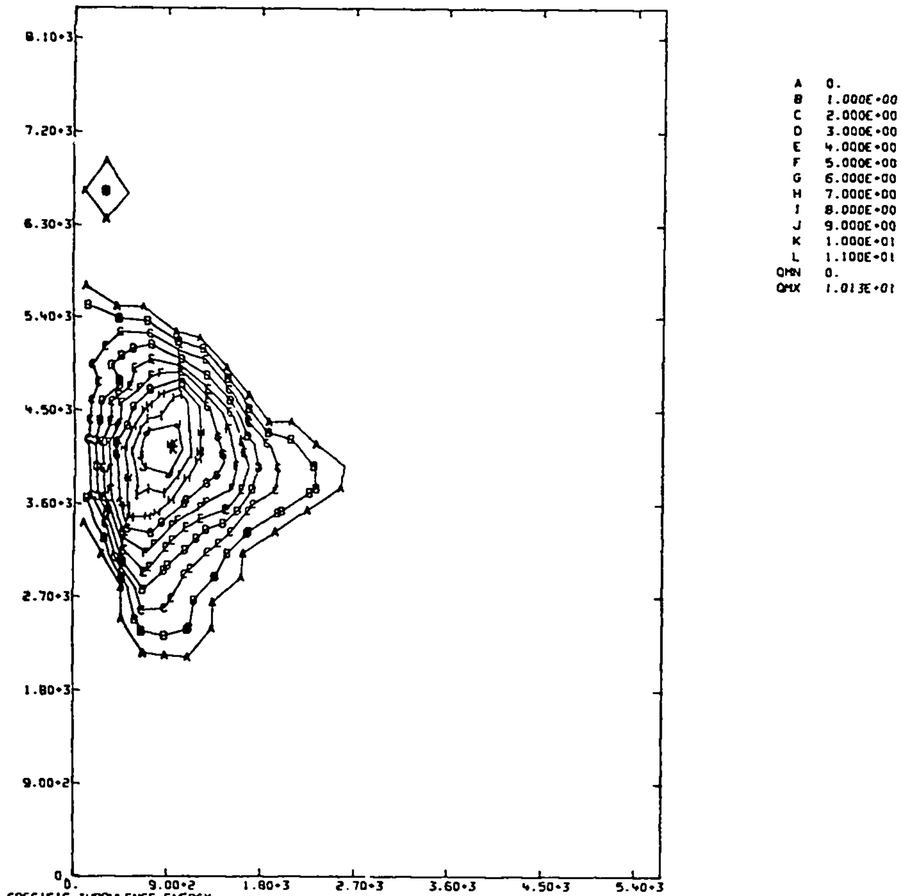
ZONES IN THE FIREBALL REGION
DRMIN= 2.00000E+02 DRMAX= 1.69944E+03 DZMIN= 2.00000E+02
DZMAX= 1.69944E+03 XR= 1.92794E+04 YB=-1.38538E-03 YT= 2.35794E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E-03 CYCLE= 0



FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 4.86465E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E-03 CYCLE= 0

B. Turbulence Seeding Conditions

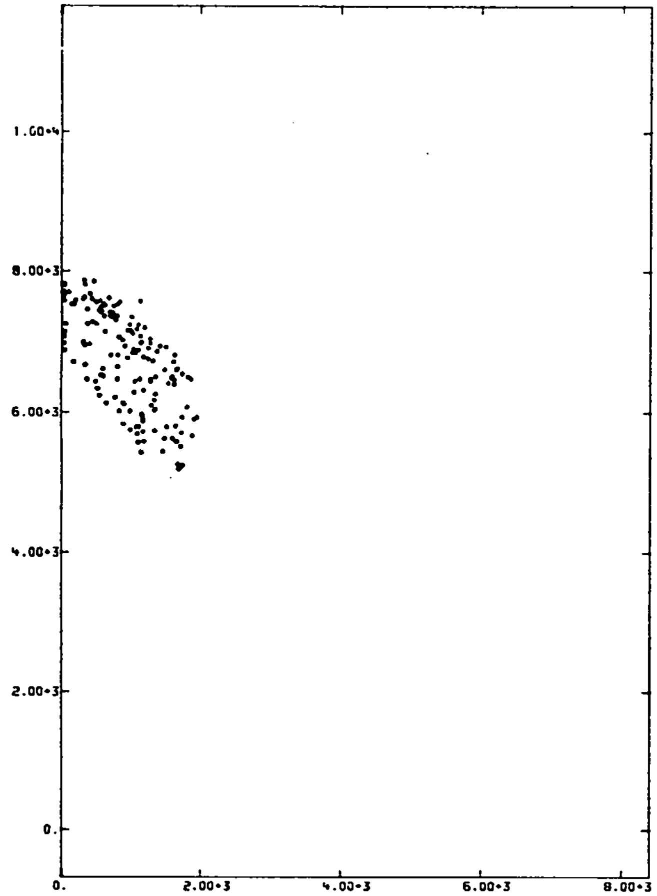




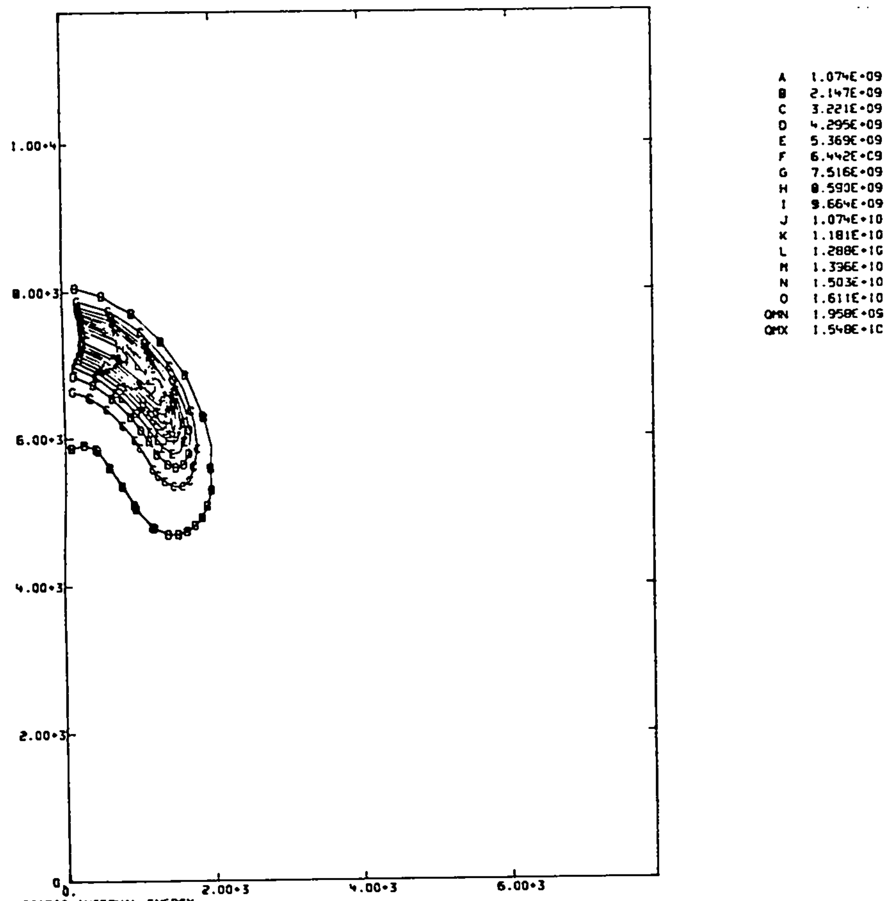
SPECIFIC TURBULENCE ENERGY

YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 5.13512E-01 CYCLE= 81

C. Turbulence Equilibrium Conditions

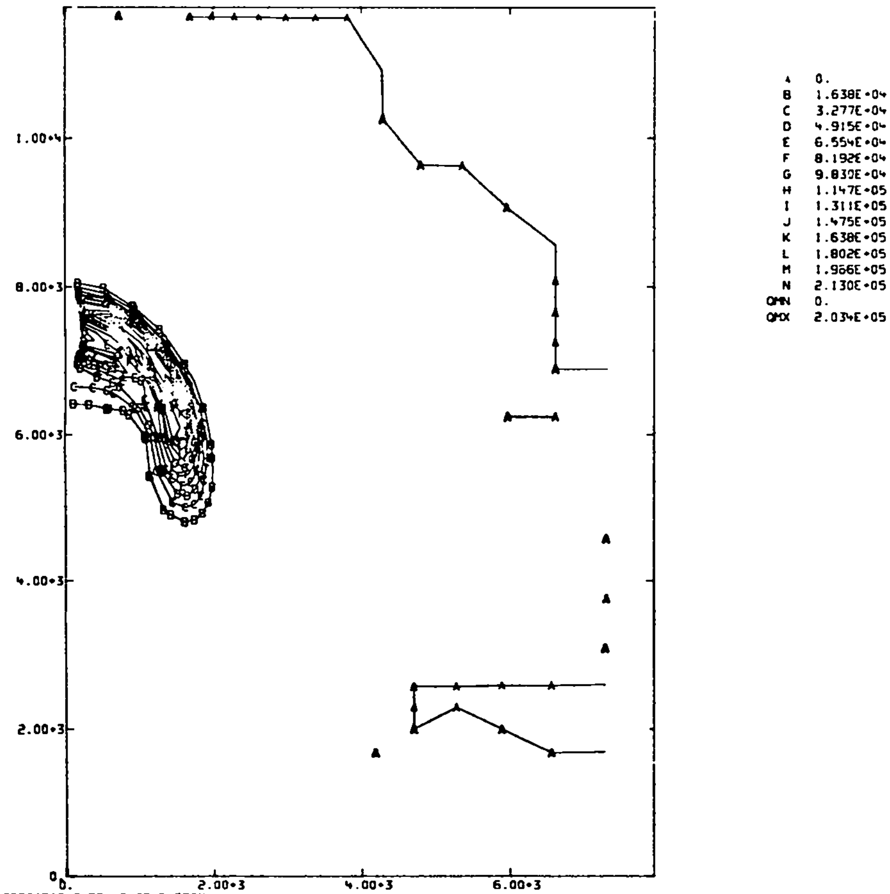


PARTICLES
PXR= 1.92794E+04 PYB= 0. PYT= 2.35794E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 3.00000E+00 CYCLE= 64



SPECIFIC INTERNAL ENERGY
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TPBCOR

T= 3.00000E+00 CYCLE= 464



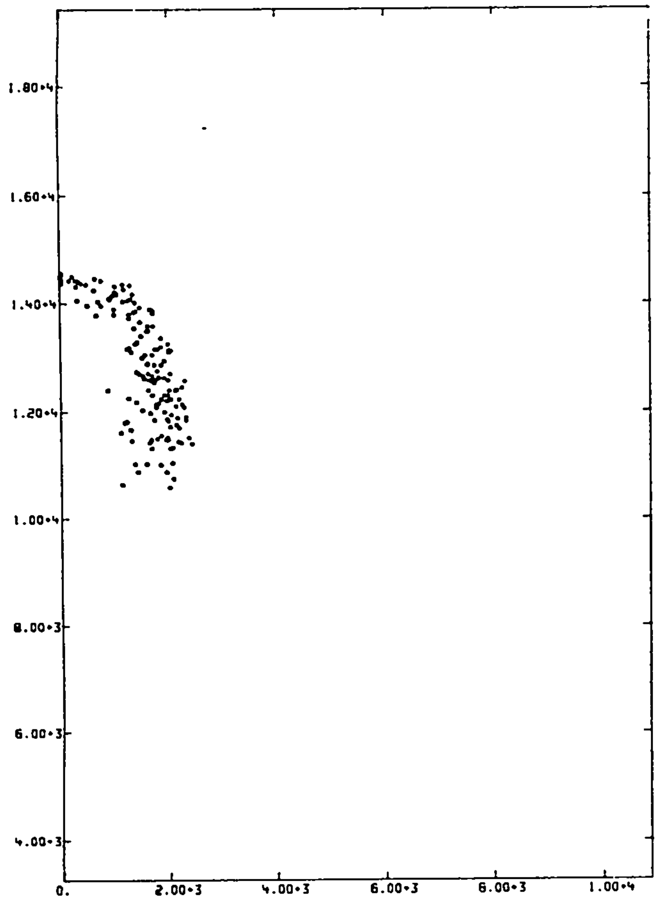
SPECIFIC TURBULENCE ENERGY

YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

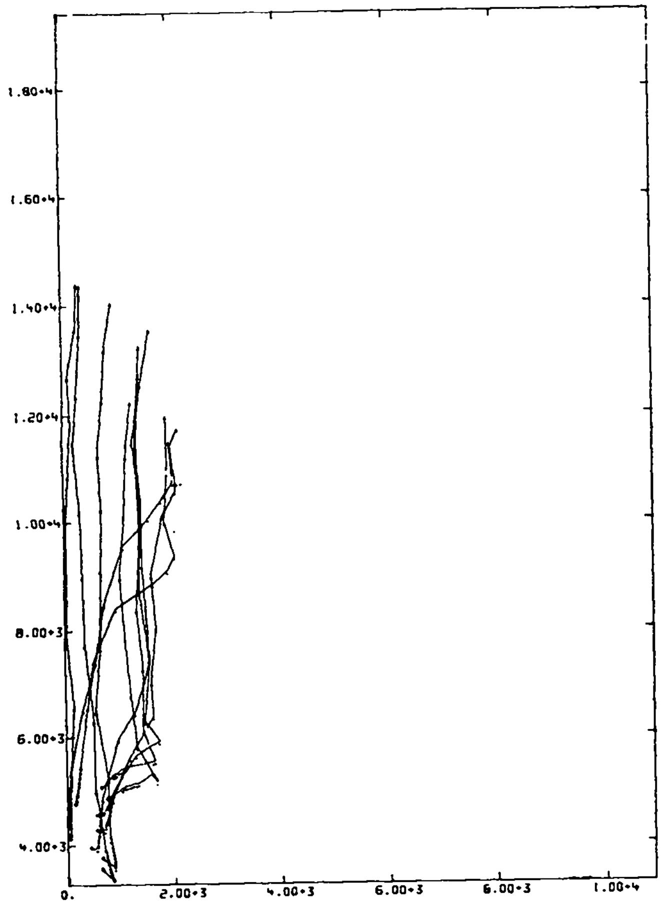
T= 3.00000E+00 CYCLE= 464

D. Moderately Late-Time Conditions (Eight Torus-Formation Times)

```
CYCLE  1546 FOUND
CYCLE  1803 FOUND
CYCLE  2050 FOUND
CYCLE  2289 FOUND
RESTARTING FROM CYCLE 2289
 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TPBCOR
T= 1.00000E+01

****** CYCLE  2289. T= 1.00000E+01, DT= 1.34180E-03, CP= 2.97676E+00
                GRINDS= 1.29761E-04, NUMIT=  0, CIRC= 1.09160E+07
                DTV= 4.25932E-03, IDTV=  7, JDTV= 22
                DTC= 1.10200E-01, IDTC=  1, JDTC= 23
                TMAX= 3.14928E+09, ITM=  2, JTM= 33, XTMAX= 0.        , YTMAX= 0.
                TGMX= 2.65171E+06, ITG=  1, JTG= 34
                PRIT= 2.436E+03, PTOP= 1.455E+04, PBOT= 1.058E+04, PDIAM= 4.871E+03, PAVHT= 1.257E+04
                TOTAL INTERNAL ENERGY =  1.0505706E+19
                TOTAL KINETIC ENERGY =  7.1391765E+12
                TOTAL GRAV. POTENTIAL ENERGY =  6.8986748E+16
                TOTAL RADIAL MOMENTUM =  4.6797458E+09
                TOTAL AXIAL MOMENTUM = -2.6862275E+10
                17104271067616720223     17147033011630461263
                VMAX =  1.12413E+03 AT VERTEX     5   16
```

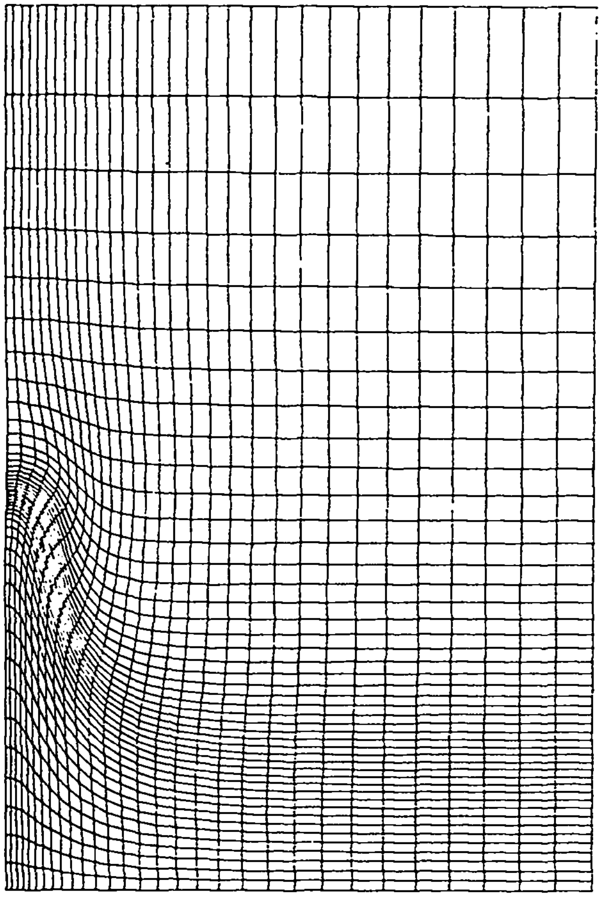



```
PARTICLES
         PXR= 1.92794E+04 PYB= 0.          PYT= 2.92102E+04
    YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TPBCOR          T= 1.00000E+01 CYCLE= 2289
```

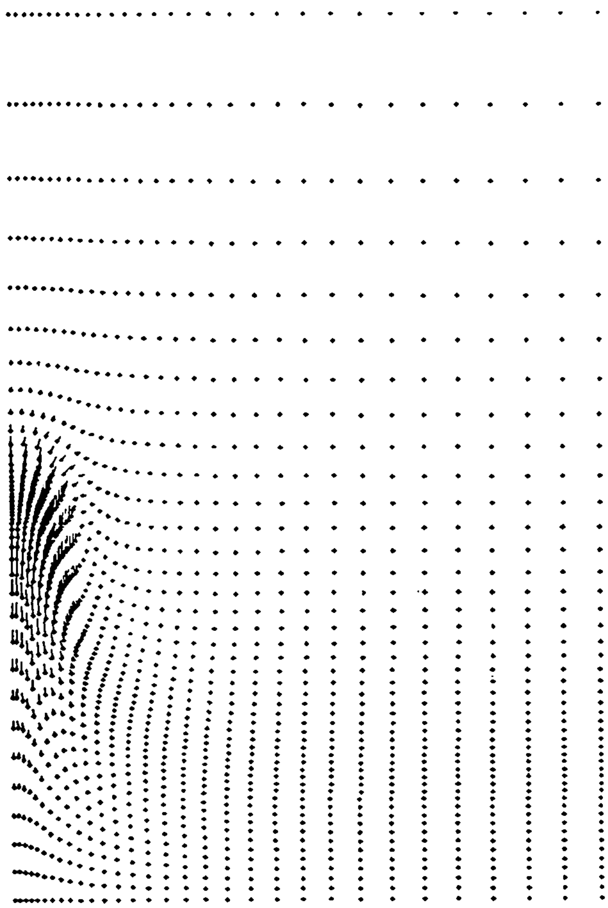


TIME-DEPENDENT PARTICLES
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TPBCOR

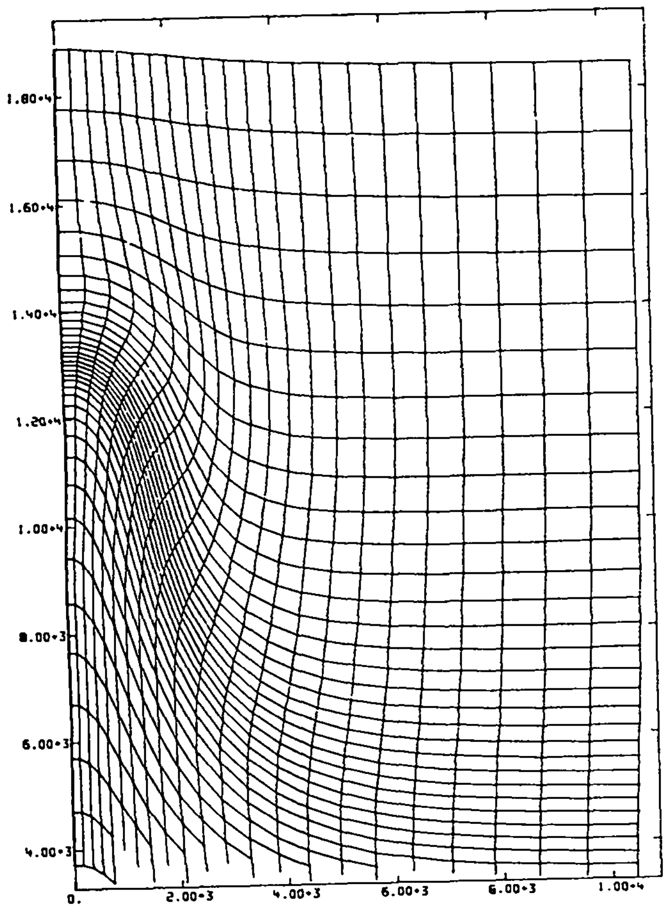
T= 1.00000E+01 CYCLE= 3299



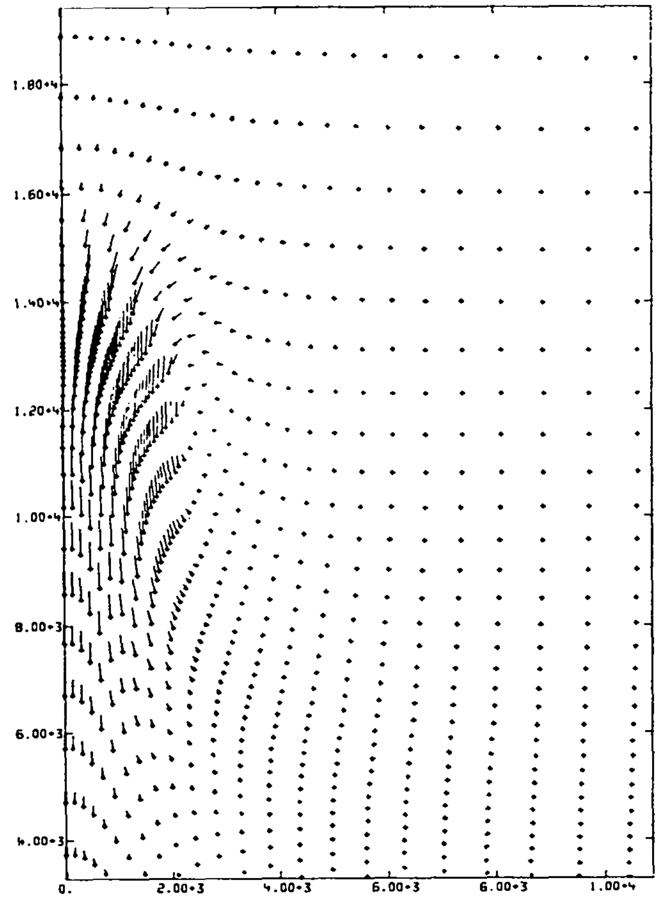
ALL ZONES
DRMIN= 1.59383E+02 DRMAX= 1.18653E+03 DZMIN= 8.03520E+01
DZMAX= 2.99070E+03 XR= 1.92794E+04 YB= 0. YT= 2.92102E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2299



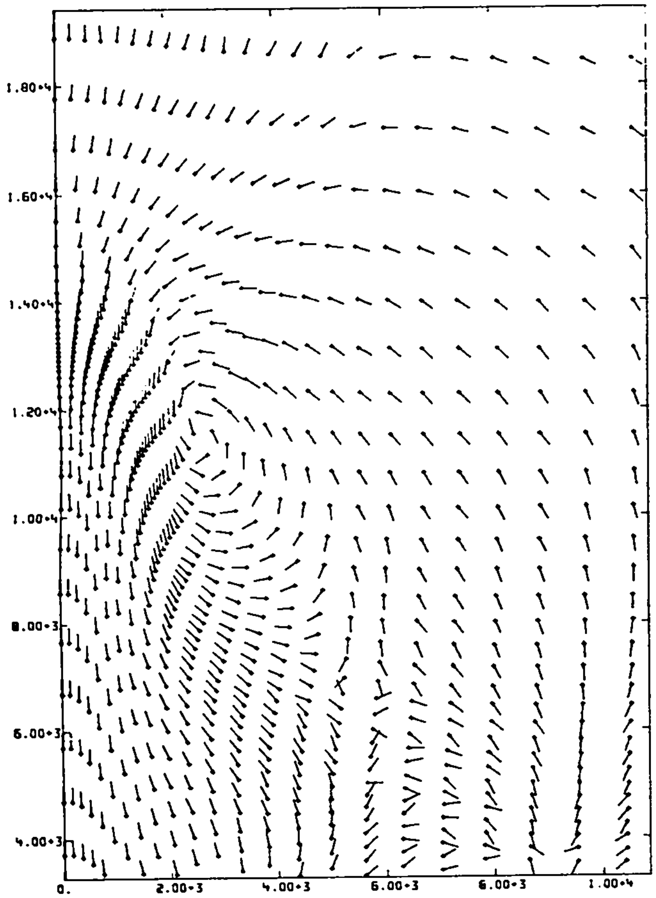
FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 1.12413E+03
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2299



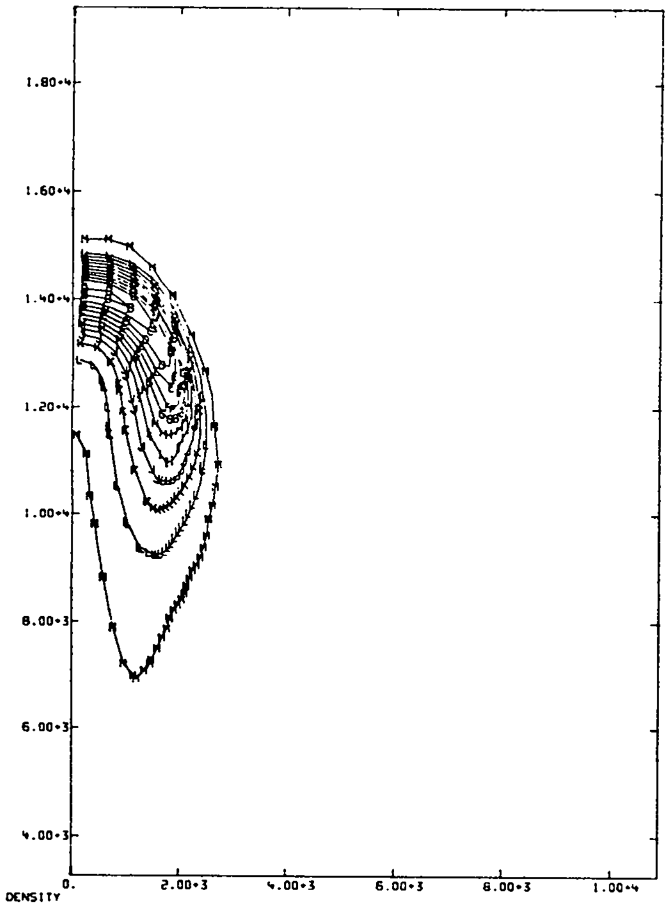
ZONES IN THE FIREBALL REGION
DRMIN= 1.59383E+02 DRMAX= 1.18653E+03 DZMIN= 8.03520E+01
DZMAX= 2.99070E+03 XR= 1.92794E+04 YB= 0. YT= 2.92102E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2299



FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 1.12413E-03
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR   T= 1.00000E-01 CYCLE= 2289



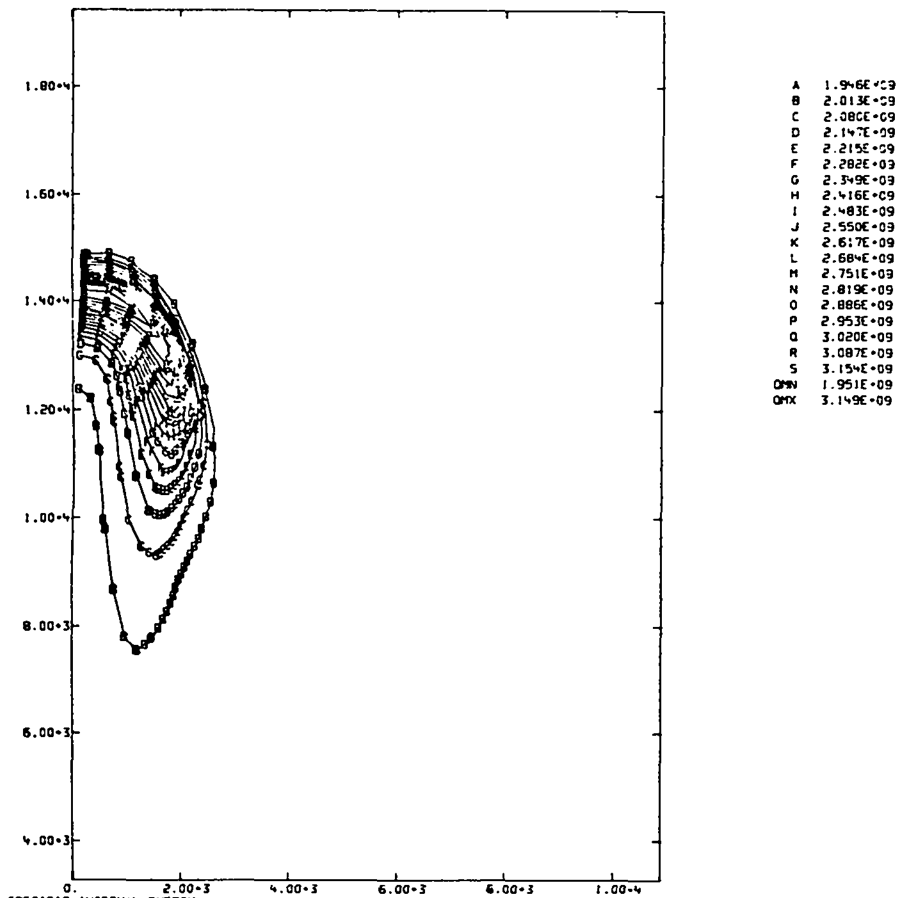
UNSCALED FLUID VELOCITY VECTORS
VMAX= 1.12413E+03
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2283



A	6.104E-04
B	6.409E-04
C	6.714E-04
D	7.019E-04
E	7.324E-04
F	7.629E-04
G	7.935E-04
H	8.240E-04
I	8.545E-04
J	8.850E-04
K	9.155E-04
L	9.460E-04
M	9.756E-04
N	1.007E-03
QMN	6.312E-04
QMX	1.036E-03

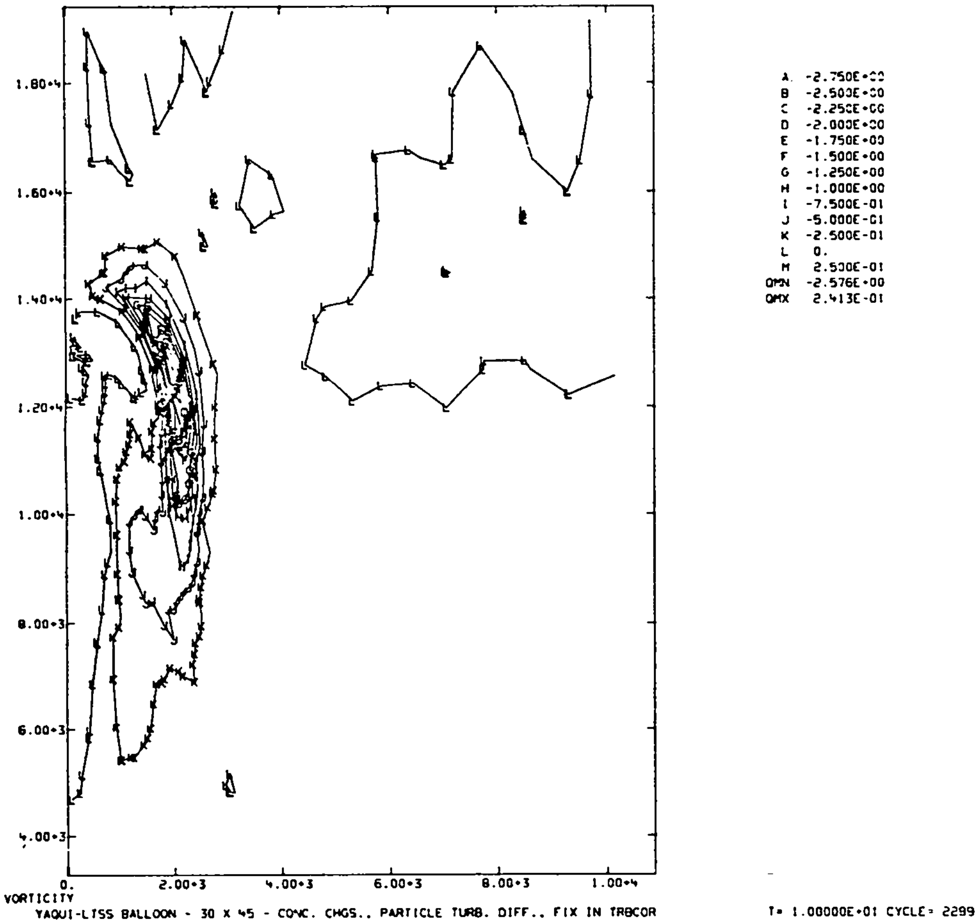
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

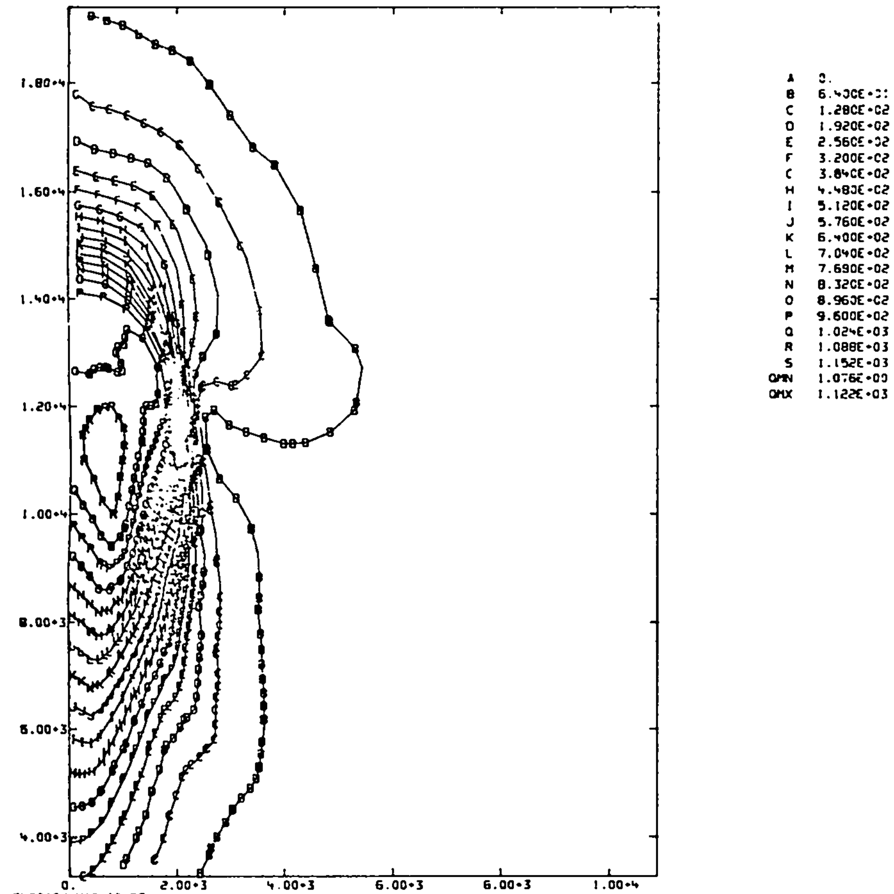
T= 1.00000E+01 CYCLE= 2283



SPECIFIC INTERNAL ENERGY
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

T= 1.00000E+01 CYCLE= 2289

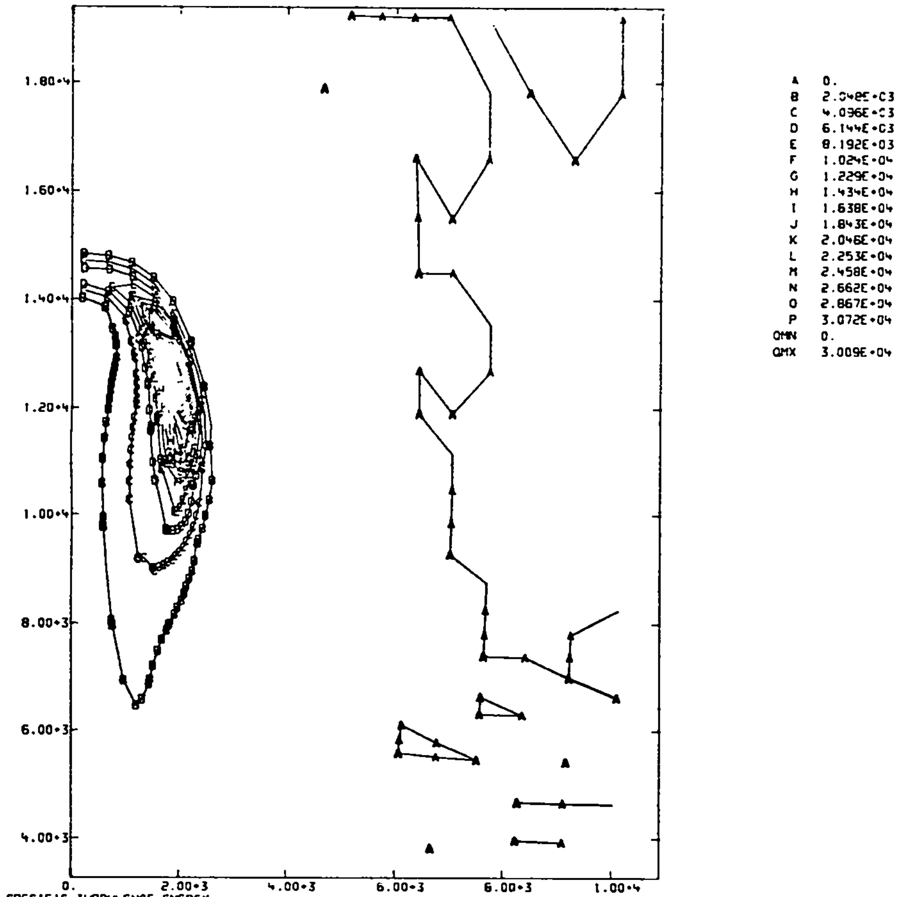

1.80+4
1.60+4
1.40+4
1.20+4
1.00+4
8.00+3
6.00+3
4.00+3
0.
2.00+3
4.00+3
6.00+3
8.00+3
1.00+4
A -2.750E+00
B -2.500E+00
C -2.250E+00
D -2.000E+00
E -1.750E+00
F -1.500E+00
G -1.250E+00
H -1.000E+00
I -7.500E-01
J -5.000E-01
K -2.500E-01
L 0.
M 2.500E-01
OMN -2.576E+00
OMX 2.413E-01
VORTICITY
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR
T= 1.00000E+01 CYCLE= 2299



VELOCITY MAGNITUDE

YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

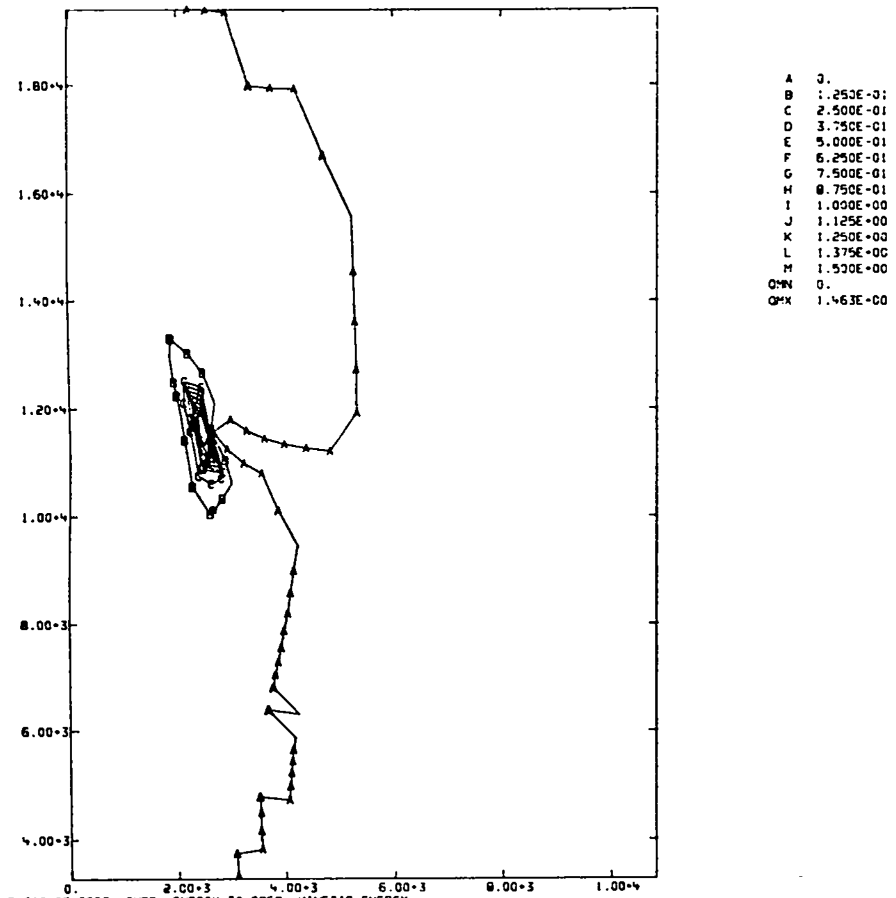
T= 1.00000E+01 CYCLE= 2289



SPECIFIC TURBULENCE ENERGY

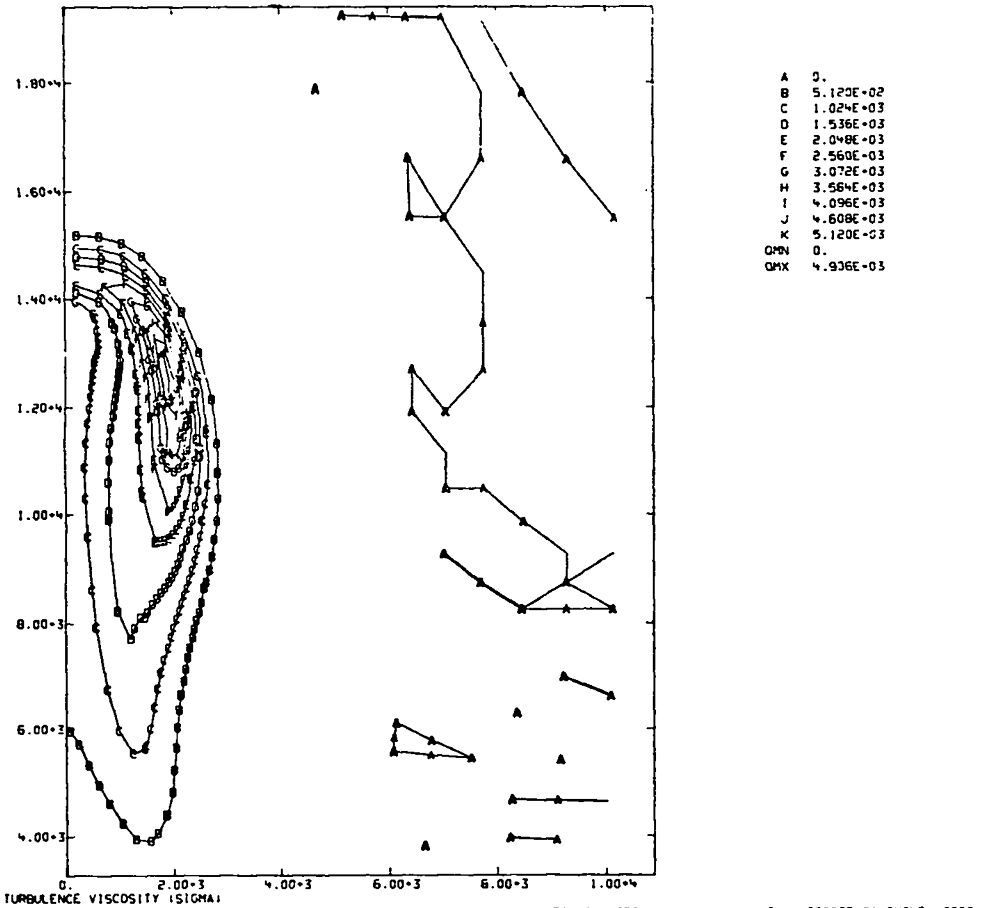
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

T= 1.00000E+01 CYCLE= 2289

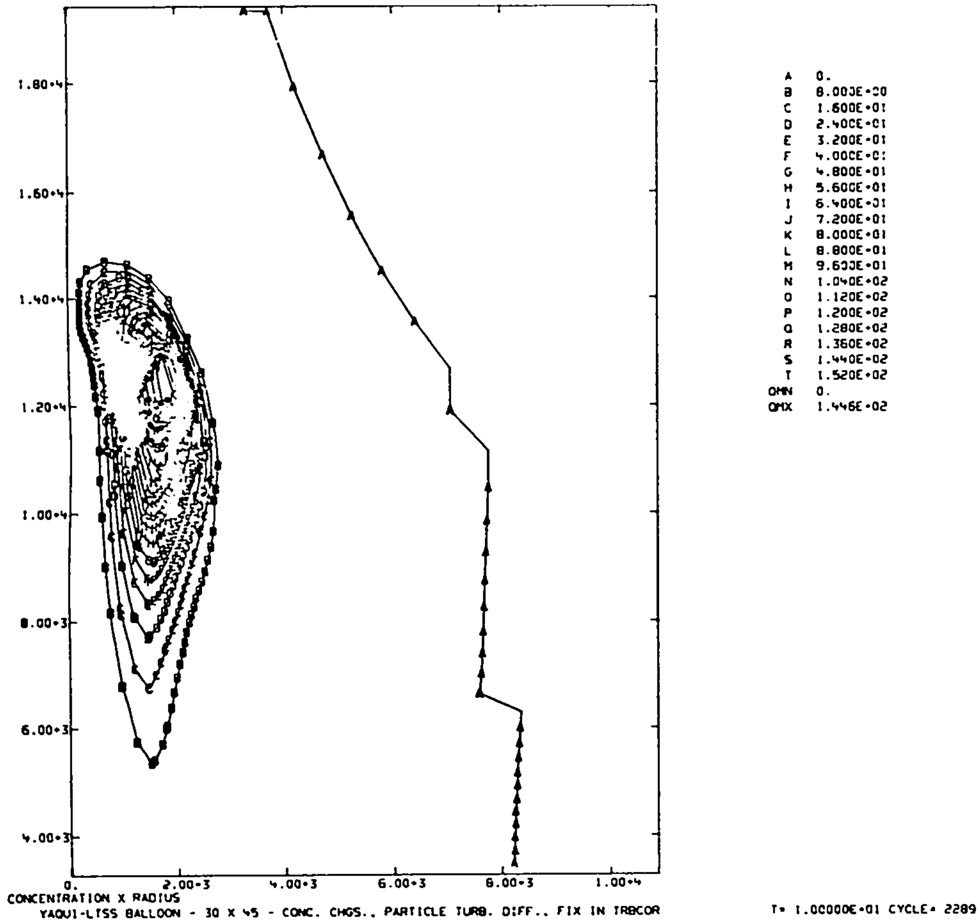


RATIO OF SPEC. TURB. ENERGY TO SPEC. KINETIC ENERGY
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

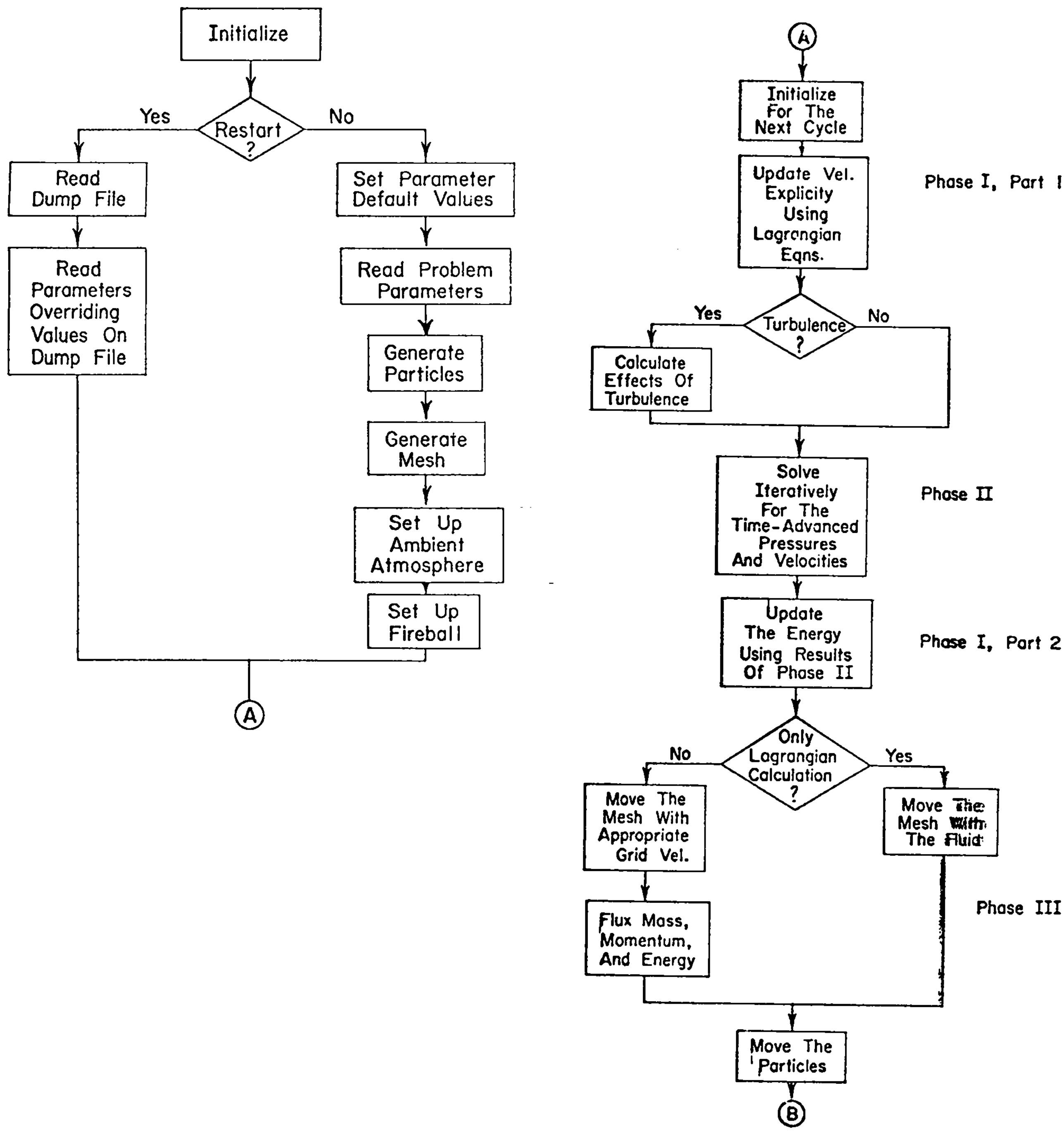
T= 1.00000E+01 CYCLE= 2289

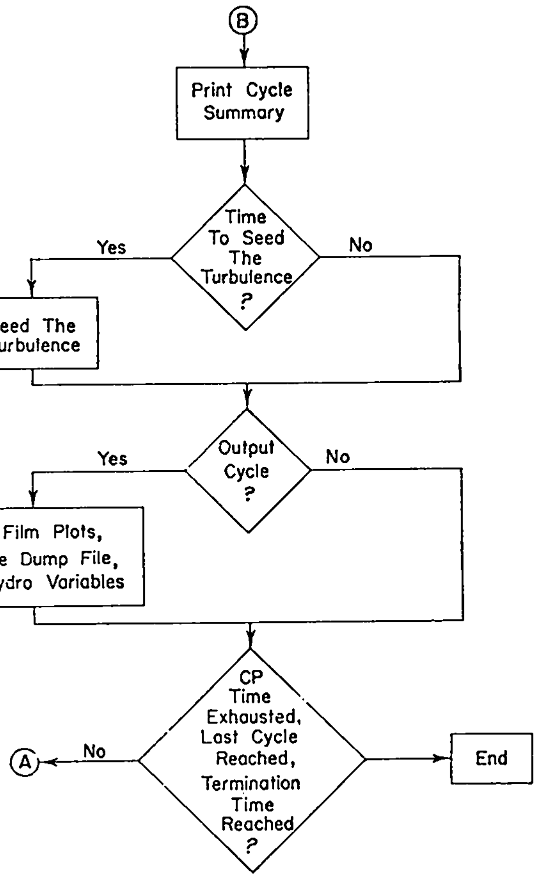


YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR   T= 1.00000E+01 CYCLE= 2289


1.80+4
1.60+4
1.40+4
1.20+4
1.00+4
8.00+3
6.00+3
4.00+3
0.
2.00+3
4.00+3
6.00+3
8.00+3
1.00+4
CONCENTRATION X RADIUS
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR
A 0.
B 8.000E+00
C 1.600E+01
D 2.400E+01
E 3.200E+01
F 4.000E+01
G 4.800E+01
H 5.600E+01
I 6.400E+01
J 7.200E+01
K 8.000E+01
L 8.800E+01
M 9.600E+01
N 1.040E+02
O 1.120E+02
P 1.200E+02
Q 1.280E+02
R 1.360E+02
S 1.440E+02
T 1.520E+02
QMN 0.
QMX 1.446E+02
T= 1.00000E+01 CYCLE= 2289

IV. FLOW DIAGRAM




B
Print Cycle Summary
Time To Seed The Turbulence ?
Yes
No
Seed The Turbulence
Output Cycle ?
Yes
No
Do Film Plots, Prepare Dump File, Print Hydro Variables
CP Time Exhausted, Last Cycle Reached, Termination Time Reached ?
No
A
End

V. CODE LISTING

A. LTSS-Dependent Code

```
      PROGRAM MAIN(TAPE1,TAPE5=TAPE1,OTAPE1,TAPE6=OTAPE1,TAPE63=100B,
     1 TAPE3)
C
C     LTSS MAIN ROUTINE FOR YAQUI
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL YAQUI
      END
```

```
      INTEGER AND
      FUNCTION AND(I,J)
C
C     LTSS ROUTINE TO TAKE THE BOOLEAN INTERSECTION OF TWO VARIABLES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      AND=I.INT.J
      RETURN
      END
```

```
      INTEGER ANDR
      FUNCTION ANDR(I,J)
C
C     LTSS INTERSECTION ROUTINE RETURNED IN A FLOATING-POINT VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      ANDR=I.INT.J
      RETURN
      END
```

```
      SUBROUTINE CLOSIT(I)
C
C     LTSS/7600 ROUTINE TO CLOSE A DISK FILE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL TPGEN(I,ITP)
```

```
      CALL ASSIGN(I,0,ITP,-2)
      RETURN
      END
```

```
      INTEGER COMP
      FUNCTION COMP(I)
C
C     LTSS ROUTINE TO TAKE THE COMPLEMENT OF A VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      COMP=.COMP.I
      RETURN
      END
```

```
      INTEGER COMPR
      FUNCTION COMPR(I)
C
C     LTSS COMPLEMENT ROUTINE RETURNED IN A FLOATING-POINT VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      COMPR=.COMP.I
      RETURN
      END
```

```
      SUBROUTINE ECRD(SCARR,IADDLC,NW,IERROR)
C
C     LTSS/7600 ROUTINE TO COPY DATA FROM LARGE CORE TO SMALL CORE
C
C     SCARR - SMALL CORE ARRAY INTO WHICH DATA IS TO BE COPIED
C     IADDLC - LARGE CORE ADDRESS FROM WHICH DATA IS TO BE COPIED
C     NW - NO. OF WORDS TO BE COPIED
C     IERROR - ERROR FLAG (DUMMY)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      LCM FWLCMC
      COMMON/FWLCMC/AA1(1)
```

```
      IERROR=0
      CALL BLOCKCOPY(AA1(IADDLC+2),SCARR,NW)
      RETURN
      END
```

```
      SUBROUTINE ECWR(SCARR,IADDLC,NW,IERROR)
C
C     LTSS/7600 ROUTINE TO COPY DATA FROM SMALL CORE TO LARGE CORE
C
C     SCARR - SMALL CORE ARRAY FROM WHICH DATA IS TO BE COPIED
C     IADDLC - LARGE CORE ADDRESS INTO WHICH DATA IS TO BE COPIED
C     NW - NO. OF WORDS TO BE COPIED
C     IERROR - ERROR FLAG (DUMMY)
C
C     WRITTEN BY J.L.NORTON,LASL T=3,1975
C
      LCM FWLCMC
      COMMON/FWLCMC/AA1(1)
      IERROR=0
      CALL BLOCKCOPY(SCARR,AA1(IADDLC+2),NW)
      RETURN
      END
```

```
      INTEGER GETIT
      FUNCTION GETIT(I)
C
C     LTSS ROUTINE TO ALLOW FETCHING CONTENTS OF A WORD GIVEN ITS
C         ABSOLUTE ADDRESS
C
C     WRITTEN BY J.L.NORTON,LASL T=3,1975
C
      ABSOLUTE IADDR(1)
      DIMENSION IADDR(1)
      GETIT=IADDR(I)
      RETURN
      END
```

```
      SUBROUTINE GETJOB(JOBID)
C
```

```
C       LTSS ROUTINE FOR RETURNING THE JOB ID
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1975
C
        JOBID=10HYAQUI-LTSS
        RETURN
        END
```

```
        SUBROUTINE GETJTL(TL)
C
C       LTSS ROUTINE FOR RETURNING THE JOB TIME LIMIT IN SECONDS
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1975
C
        CALL 00TIM(I,J,K,L)
        TL=FLOAT(J)*1.E-6
        RETURN
        END
```

```
        SUBROUTINE GETLCM(IFLLCM)
C
C       LTSS/7600 ROUTINE TO RETURN THE AMOUNT OF LCM AVAILABLE TO THE
C           USER
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1975
C
        COMMON/GOBCOM/IDUM(1)
        IFLLCM=IDUM(16)-IDUM(15)
        RETURN
        END
```

```
        SUBROUTINE GETTPE(IDUM)
C
C       LTSS ROUTINE TO DUMMY UP TAPE FETCHING AVAILABLE ON CROS/7600
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1975
C
        RETURN
        END
```

```
      SUBROUTINE LCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
C
C     LTSS/7600 ROUTINE TO READ OR WRITE LCM FROM OR TO A DISK FILE
C
C     FWA    - FIRST LCM ADDRESS
C     NWRDS  - NO. OF WORDS TO TRANSFER
C     IFILE  - LOGICAL UNIT NO. OF DISK FILE
C     IFLAG  - READ OR WRITE FLAG
C            - 0 - READ DISK
C            - 1 - WRITE DISK
C     IRET   - RETURN FLAG
C            - 0 - RETURN IMMEDIATELY AFTER ISSUING THE I/O REQUEST
C            - 1 - WAIT UNTIL I/O IS COMPLETED BEFORE RETURNING
C     IERROR - ERROR FLAG
C            - 0 - NO ERROR
C            - 1 - ERROR
C            - -1 - END-OF-FILE ON INPUT
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      LCM FWLCMC
      COMMON/FWLCMC/AA1(1)
      INTEGER FWA
C
C     CLEAR ERROR FLAG
C
      IERROR=0
C
C     SEE WHETHER REQUEST IS READ INTO OR WRITE FROM LCM
C
      IF(IFLAG.NE.0) GO TO 10
C
C     REQUEST IS WRITE LCM (READ DISK)
C
      BUFFER IN(IFILE,1)(AA1(FWA+2),AA1(FWA+NWRDS+1))
      GO TO 20
C
C     REQUEST IS READ LCM (WRITE DISK)
C
   10 CONTINUE
      BUFFER OUT(IFILE,1)(AA1(FWA+2),AA1(FWA+NWRDS+1))
C
C     SEE IF USER WISHES TO WAIT UNTIL I/O IS COMPLETE
C
   20 CONTINUE
      IF(IRET.EQ.0) RETURN
C
C     YES, WAIT FOR I/O TO COMPLETE
C
   30 IF(UNIT,IFILE) 30,40,50,60
C
C     I/O SUCCESSFULLY COMPLETED. ALL DONE.
C
   40 CONTINUE
```

```
      RETURN
C
C     END-OF-FILE. SET ERROR FLAG AND RETURN.
C
   50 CONTINUE
      IERROR=-1
      RETURN
C
C     I/O ERROR. SET ERROR FLAG AMD RETURN.
C
   60 CONTINUE
      IERROR=1
      RETURN
      END
```

```
      FUNCTION LOCF(I)
C
C     LTSS ROUTINE TO HANDLE LOCATION FUNCTION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      LOCF=.LOC.I
      RETURN
      END
```

```
      SUBROUTINE NCODE(NC,IFORM,INTAB,NIN,OUTTAB)
C
C     LTSS ROUTINE TO SIMULATE CDC ENCODE STATEMENT
C
C     ENCODE(NC,IFORM,OUTTAB) (INTAB(I),I=1,NIN)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      DIMENSION INTAB(NIN),OUTTAB(1),ITEMP(10),IFORM(1)
      DO 10 I=1,10
   10 ITEMP(I)=IFORM(I)
      REWIND 63
      NW=NC/10+1
      IF(MOD(NC,10).EQ.0) NW=NW-1
      WRITE(63,ITEMP)(INTAB(I),I=1,NIN)
      READ(63,20)(OUTTAB(I),I=1,NW)
      RETURN
C
   20 FORMAT(15A10)
      END
```

```
      SUBROUTINE OPENIT(IFILE,MODE)
C
C     LTSS/7600 ROUTINE TO OPEN A FILE
C
C     IFILE = LOGICAL UNIT NO. OF THE FILE
C     MODE  = TYPE OF FILE
C           = 0 = BCD
C           = 1 = BINARY
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      IF(MODE.EQ.0) RETURN
      LENGTH=1000000
      IOC=0
      CALL TPGEN(IFILE,ITP)
   10 CONTINUE
      CALL CREATE(ITP,LENGTH,IOC)
      IF(IOC.GE.0) GO TO 20
      LENGTH=FLOAT(LENGTH)*.9
      GO TO 10
   20 CONTINUE
      CALL ASSIGN(IFILE,0,ITP)
      RETURN
      END
```

```
      INTEGER OR
      FUNCTION OR(I,J)
C
C     LTSS ROUTINE TO TAKE THE BOOLEAN UNION OF TWO VARIABLES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      OR=I.UN.J
      RETURN
      END
```

```
      INTEGER ORR
      FUNCTION ORR(I,J)
C
C     LTSS FUNCTION TO RETURN THE UNION IN A FLOATING POINT VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
```

```
C
      ORR=I.UN.J
      RETURN
      END
```

```
      FUNCTION RNUMF(X)
C
C     LTSS/7600 ROUTINE TO RETURN RANDOM NOS. UNIFORMLY DISTRIBUTED
C         ON THE INTERVAL (0.,1.)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RNUMF=RNFL(X)
      RETURN
      END
```

```
      SUBROUTINE SCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
C
C     LTSS/7600 ROUTINE TO READ OR WRITE SCM FROM OR TO A DISK FILE
C
C     SEE LCBUFF FOR ARGUMENT DOCUMENTATION
C
C     FWA - BEGINNING OF SCM BLOCK TO BE WRITTEN
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      DIMENSION FWA(1)
      LCM FWLCMC
      COMMON/FWLCMC/AA1(1)
      COMMON/LCSCRC/ILSIZE,IFWASC
C
C     CLEAR ERROR FLAG
C
      IERROR=0
C
C     SEE IF THE LCM SCRATCH AREA IS LARGE ENOUGH TO HOLD THE
C         SCM BLOCK
C
      IF(NWRDS.GT.ILSIZE) CALL UNCLE(4,6HSCBUFF,25,
     1 25HNOT ENOUGH LCM FOR BUFFER)
C
C     YES.  SEE IF DISK IS TO BE READ OR WRITTEN.
C
      IF(IFLAG.NE.0) GO TO 60
C
```

```
C     READ DISK INTO THE LCM SCRATCH AREA
C
      BUFFER IN(IFILE,1)(AA1(IFWASC+2),AA1(IFWASC+NWRDS+1))
C
C     WAIT FOR I/O TO COMPLETE
C
   10 IF(UNIT,IFILE) 10,20,30,40
   20 CONTINUE
C
C     COPY LCM INTO SCM
C
      CALL BLOCKCOPY(AA1(IFWASC+2),FWA,NWRDS)
      RETURN
C
C     EOF ENCOUNTERED, SET THE ERROR FLAG,
C
   30 CONTINUE
      IERROR=-1
      RETURN
C
C     DISK ERROR OR INPUT RECORD WAS SHORTER THAN EXPECTED
C
   40 CONTINUE
C
C     GET THE LENGTH OF THE RECORD AND IF NOT ZERO,COPY IT TO SCM
C
      L=LENGTH(IFILE)
      IF(L.NE.0) CALL BLOCKCOPY(AA1(IFWASC+2),FWA,L)
C
C     SET THE ERROR FLAG
C
   50 CONTINUE
      IERROR=1
      RETURN
C
C     DISK IS TO BE WRITTEN
C
   60 CONTINUE
C
C     FIRST COPY SCM BLOCK TO LCM SCRATCH AREA
C
      CALL BLOCKCOPY(FWA,AA1(IFWASC+2),NWRDS)
C
C     WRITE LCM TO DISK
C
      BUFFER OUT(IFILE,1)(AA1(IFWASC+2),AA1(IFWASC+NWRDS+1))
C
C     WAIT FOR I/O TO COMPLETE IF SO REQUESTED
C
      IF(IRET.EQ.0) RETURN
   70 IF(UNIT,IFILE) 70,80,30,50
   80 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SDG(X)
C
C     LTSS/7600 ROUTINE TO FIND SINES AND COSINES
C         OF ARGUMENTS IN DEGREES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      DATA DEGRAD/.017453292519943/
      Y=X*DEGRAD
      SDG=SIN(Y)
      RETURN
      ENTRY CDG(X)
      Y=X*DEGRAD
      SDG=COS(Y)
      RETURN
      END
```

```
      INTEGER SHIFT
      FUNCTION SHIFT(IX,N)
C
C     LTSS ROUTINE TO PERFORM BIT SHIFTING
C
C     IX IS THE QUANTITY TO BE SHIFTED
C     N IS THE NO. OF BITS TO SHIFT. N POSITIVE MEANS LEFT END-AROUND
C         SHIFT AND N NEGATIVE MEANS RIGHT END-OFF SHIFT.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      IF(N.GE.0) GO TO 10
      NP=-N
      SHIFT=IX.SHR.NP
      RETURN
   10 CONTINUE
      SHIFT=IX.SHL.N
      RETURN
      END
```

```
      SUBROUTINE STORIT(IVAR,IADD)
C
C     LTSS ROUTINE TO ALLOW STORING INTO A WORD GIVEN ITS ABSOLUTE
C         ADDRESS
C
```

```
      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
      C
            ABSOLUTE IADDR(1)
            DIMENSION IADDR(1)
            IADDR(IADD)=IVAR
            RETURN
            END
```

```
            SUBROUTINE SYINIT
      C
      C     LTSS/7600 ROUTINE TO DO SYSTEM-DEPENDENT INITIALIZATION
      C
      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
      C
            LCM FWLCMC,YLC1,YLC2,YLC3,(FILMLB),LCMSCR
            COMMON/FWLCMC/FWLCM
            COMMON/YLC1/AA1(72300)/YLC2/AA2(5000)
            COMMON/YLC3/PAXY(10,200)/FILMLB/FLMBUF(4000)
            COMMON/LCMSCR/SCRTCH(1000)
      *     ----- BEGIN COMDECK PARAM      -----
            COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
           1 NLCP3,NLCP4,IFLMSZ
      *     ----- END COMDECK PARAM        -----
            COMMON/LCSCRC/ILSIZE,IFWASC
      C
      C     CHANGE THE DROPFILE NAME TO +YAQUIB
      C
            CALL CHANGE(7H+YAQUIB)
      C
      C     INITIALIZE THE FILM ROUTINES.  THE ARGUMENTS ARE AS FOLLOWS --
      C         FOR IDENT80 -
      C               (1) LOGICAL UNIT NO. OF THE FILM FILE
      C               (2) ARRAY IN LCM TO BE USED AS A FILM BUFFER
      C               (3) SIZE OF THE FILM BUFFER
      C               (4) SIZE OF DISK FILE FOR FILM
      C               (5) TYPE OF MICROFILM OUTPUT
      C                     - 35 FOR 35 MM
      C                     - 105 FOR MICROFICHE
      C               (6) NOT USED
      C         FOR HEAD80 -
      C               (1) LOGICAL UNIT NO. OF THE FILM FILE
      C               (2) NO. OF CHARACTERS IN THE HEADER (60 MAXIMUM)
      C               (3) THE HEADER ARRAY
      C               (4) FILM IDENTIFICATION ARRAY
      C         FOR KEEPFLM -
      C               (1) LOGICAL UNIT NO. OF THE FILM FILE NOT TO BE GIVEN
      C                         TO THE SYSTEM
      C
            CALL FILM80
            CALL IDENT80(12,FLMBUF,IFLMSZ,1000000,105,IDUM)
            CALL HEAD80(12,10,10HT3JLNYAQUI,9HBOX T3JLN)
```

```
      CALL KEEPFLM(12)
C
C     INITIALIZE THE SIZE OF THE LCM SCRATCH AREA
C
      ILSIZE=NLCP4
C
C     SET THE ADDRESS OF THE FIRST WORD OF THE SCRATCH AREA
C
      IFWASC=.LOC.SCRTCH(1)-.LOC.FWLCM
      RETURN
      END
```

```
      SUBROUTINE TPGEN(IFILE,ITP)
C
C     LTSS/7600 ROUTINE TO TAKE AN INTEGER LOGICAL UNIT NO. IFILE
C         AND CONVERT IT INTO A DISK FILE NAME OF THE FORM TAPEN
C         OR TAPENN WHICH IS RETURNED LEFT-JUSTIFIED IN ITP
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER OR,SHIFT,AND
      DIMENSION INUM(10)
      DATA INUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
C
C     INITIALIZE
C
      IFILEP=IFILE
      IC=-24
      ITP=4HTAPE
      MASK=77777777777B
C
C     SEE IF LOGICAL UNIT NO. IS ONE OR TWO DIGITS
C
      IF(IFILEP.LE.9) GO TO 10
C
C     TWO DIGITS. ISOLATE THE FIRST DIGIT.
C
      IT=IFILEP/10+1
      IF(MOD(IFILEP,10).EQ.0) IT=IT-1
C
C     PUT THE FIRST DIGIT INTO PLACE
C
      ITP=OR(AND(SHIFT(INUM(IT),IC),MASK),ITP)
C
C     MODIFY PARAMETERS FOR SECOND DIGIT
C
      IC=IC-6
      MASK=7777777777B
      IFILEP=IFILEP-10*(IT-1)
   10 CONTINUE
C
```

```
C      PUT THE LAST DIGIT INTO PLACE
C
       ITP=OR(AND(SHIFT(INUM(IFILEP+1),IC),MASK),ITP)
       RETURN
       END
```

```
       SUBROUTINE TRAP(IARG)
C
C      LTSS ROUTINE TO HANDLE ERROR INTERCEPTION
C
C      WRITTEN BY J.L.NORTON,LASL T-3,1975
C
       COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
       COMMON/OOERR/IERR(121)
       COMMON/GOBCOM/IG(1)
       ABSOLUTE TWO(2)
       INTEGER TWO,SHIFT
       LOGICAL FILM
       COMMON/IEDMP/IDUMP(16)
C
C      SET UP ABSOLUTE LOCATION 2 WITH A JUMP TO STATEMENT IRETA
C           AND SET THE EXCHANGE PACKAGE DUMP AREA TO IDUMP
C
       TWO=2.SHL.54.UN.(.LOC.IRETA.SHL.30).UN.(.LOC.IDUMP)
C
C      SET UP ORDERLIB ERROR INTERCEPTION
C
       DO 10 I=1,99
    10 CALL CONTROL(I,IORDL,IERO)
       RETURN
C
C      ORDERLIB ERROR HAS OCCURRED. LET THE USER KNOW.
C
IORDL  CONTINUE
       DO 20 IPX=6,IFD,6
    20 WRITE(IPX,50)(IERR(I),I=117,119)
       IDUMP(1)=SHIFT(IERR(118),36)
       CALL PDMPPK
       IFLS=IG(15)
       IF(FILM) GO TO 30
       GO TO 40
C
C      PROGRAM HAS ABORTED
C
IRETA  CONTINUE
C      DUMP THE EXCHANGE PACKAGE AND SMALL CORE ON EITHER SIDE
C           OF THE LOCATION OF THE ERROR
C
       CALL PABORT
C
C      DUMP ALL OF SMALL CORE TO FILM IF FILM IS ENABLED
```

```
C
      IF(.NOT.FILM) GO TO 40
C
C     PICK THE SMALL CORE FIELD LENGTH OUT OF THE EXCHANGE PACKAGE
C
      IFLS=(IDUMP(3).INT.(.COMP,7777777777B)).SHR.36
   30 CONTINUE
      IFLS=IFLS-1
      CALL DMP(0,IFLS,12)
   40 CONTINUE
C
C     TERMINATE BUT SAVE THE DROPFILE
C
      CALL EXIT(2)
      RETURN
C
   50 FORMAT(/1H ,14HORDERLIB ERROR,I3,21H OCCURRED AT LOCATION,O7,
     1 29H WHICH IS THE CALL TO ROUTINE,A10)
      END
```

```
      SUBROUTINE TTYTST(IFLAG)
C
C     LTSS ROUTINE TO SEE IF DROPFILE VARIABLE HAS BEEN CHANGED TO
C         SIGNAL JOB TERMINATION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      COMMON/OFFC/ISWTCH
      DATA ISWTCH/-1/
      IFLAG=0
      IF(ISWTCH.EQ.0) IFLAG=1
      RETURN
      END
```

B. CROS-Dependent Code

```
      PROGRAM MAIN(INP,OUT,FILM,FSET7,FSET8,FSET5=INP,FSET6=OUT,FSET12=
     1 FILM,FSET3,FSET59=OUT)
C
C     CROS/7600 MAIN ROUTINE FOR YAQUI
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL YAQUI
      END
```

```
      SUBROUTINE CLOSIT(I)
C
C     CROS/7600 ROUTINE FOR DESTROYING DISK FILES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL AFSREL(I,0,0,0)
      RETURN
      END
```

```
      SUBROUTINE GETJOB(JOBID)
C
C     CROS/7600 ROUTINE FOR RETURNING THE JOB ID
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL GETQ(4LKJBN,JOBID)
      RETURN
      END
```

```
      SUBROUTINE GETJTL(TL)
C
C     CROS/7600 ROUTINE TO RETURN THE JOB TIME LIMIT IN SECONDS
C
      CALL GETQ(4LKTLM,II)
      TL=27.5E-9*FLOAT(II)
      RETURN
      END
```

```
      SUBROUTINE GETLCM(ISIZE)
C
C     CROS/7600 ROUTINE TO RETURN THE AMOUNT OF LARGE CORE MEMORY
C         AVAILABLE TO THE JOB FOR DATA STORAGE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL GETQ(3LKMA,ISIZE)
      RETURN
      END
```

```
      SUBROUTINE GETTPE(TAPE)
C
C     CROS/7600 ROUTINE FOR CHECKING TAPE LABEL AND STAGING TAPE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER TAPE,AND
C
C     TAPE WAS READ. SEE IF IT WAS LEGAL.
C
      IF(AND(TAPE,77777000000000000000B).NE.3LXX0) CALL UNCLE(4,6HGETTPE
     1 ,20,20HTAPE NO. INPUT ERROR)
C
C     GO TO STAGE IN THE TAPE
C
      CALL STAGE(7,TAPE,IEFLAG)
C
C     SEE IF STAGE WAS SUCCESSFUL
C
      IF(IEFLAG.NE.0) CALL UNCLE(4,6HGETTPE,23,
     1 23HUNSUCCESSFUL TAPE STAGE)
C
C     YES. ALL DONE.
C
      RETURN
      END
```

```
      SUBROUTINE OPENIT(IFILE,MODE)
C
C     CROS/7600 ROUTINE TO OPEN A FILE
C
```

```
C     IFILE - LOGICAL UNIT NO. OF THE FILE
C     MODE  - TYPE OF FILE
C           - 0 - BCD
C           - 1 - BINARY
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL OPEN(IFILE,0,0,0,0,0,10000,0,0)
      RETURN
      END
```

```
      SUBROUTINE STAGE(IF,LABEL,IFLAG)
C
C     ROUTINE TO STAGE IN A TAPE WITH LABEL=XX0NNNNN INTO FILESET IF
C
C     WRITTEN BY J.L.NORTON,LASL TD-3,1973
C
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
      DIMENSION IMDISP(5)
      DATA IMDISP/4,0,0,0,0/
C
C     IFLAG IS RETURNED ZERO IF THE TAPE WAS SUCCESSFULLY STAGED. IT IS
C         RETURNED ONE IF FOUR STAGE ATTEMPTS FAILED.
C
      IFLAG=0
C
C     OPEN THE FILE TO A LARGE TRACK SECTOR LIMIT
C
      CALL OPEN(IF,0,0,0,0,0,10000,0,0)
C
C     ZERO OUT THE LAST TWO CHARACTERS OF THE TAPE LABEL
C
      IMDISP(5)=LABEL.AND.(.NOT.7777B)
C
C     INITIALIZE THE NO. OF STAGE ATTEMPTS
C
      ICNT=0
C
C     GET JOB CLASSIFICATION
C
      CALL GETQ(4LKCLA,JCLASS)
      KCLASS=1LG
      IF(JCLASS.NE.5) KCLASS=1LU
C
C     INITIATE THE STAGE
C
   10 CALL CREATE(IF,KCLASS,2LST,0,0,0,IMDISP,0,0,10000,0)
C
C     INCREMENT THE STAGE COUNT
C
      ICNT=ICNT+1
```

```
C
C     SEE IF THERE WERE ANY PARITY ERRORS
C
      CALL PARITY(IF,ICHECK)
      IF(ICHECK.NE.0) GO TO 30
C
C     NO. PRINT MESSAGE AND RETURN.
C
      DO 20 IPX=IPD,IFD,6
   20 WRITE(IPX,50) LABEL,IF,IF
      RETURN
   30 CONTINUE
C
C     YES. SEE IF THIS WAS THE FOURTH STAGE.
C
      IF(ICNT.LT.4) GO TO 40
C
C     YES. SET THE ERROR FLAG AND RETURN.
C
      IFLAG=1
      RETURN
   40 CONTINUE
C
C     NO. RELEASE THE FILE AND TRY AGAIN.
C
      CALL AFSREL(IF,0,0,0)
      GO TO 10
C
   50 FORMAT(1H0,5HTAPE ,A8,35H HAS BEEN SUCCESSFULLY STAGED INTO ,
     1 8HFILESET ,I2,2H (,A8,1H))
      END
```

```
      SUBROUTINE SYINIT
C
C     CROS/7600 ROUTINE TO PERFORM YAQUI SYSTEM INITIALIZATION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
C     CALL SETQ(4LKDPX,0)
      CALL OPEN(3LOUT,0,0,0,0,0,10000,0,0)
      CALL OPEN(4LFILM,0,0,0,0,0,10000,0,0)
      CALL OPEN(7,0,0,0,0,0,10000,0,0)
      CALL OPEN(3,0,0,0,0,0,10000,0,0)
      CALL MEMREQ(400000,1)
      RETURN
      END
```

```
      SUBROUTINE TRAP(RCOVER)
C
C     CROS/7600 ROUTINE TO INITIALIZE FOR RECOVERY FROM FATAL EXECUTION
C         ERRORS
C
C     RCOVER IS A SUBROUTINE TO CALL AFTER INTERCEPTING THE ERROR
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      DIMENSION IDUMP(16)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
      INTEGER SHIFT,AND,COMP
      LOGICAL FILM
      ASSIGN 10 TO IGOXIT
      CALL XIT(IGOXIT)
C
C     IABORT IS THE NO. OF ABORT TRAPS THAT HAVE OCCURRED
C
      IABORT=0
      RETURN
   10 CONTINUE
      CALL GETHPK(IDUMP)
      CALL PABORT(IDUMP)
      IF(.NOT.FILM) GO TO 20
      IFLS=SHIFT(AND(IDUMP(4),COMP(7777777777B)),-36)
      IFLS=IFLS-1
      CALL DMP(0,IFLS,12)
   20 CONTINUE
C
C     CODE HAS ABORTED. INCREMENT THE ABORT COUNT.
C
      IABORT=IABORT+1
C
C     CALL THE RECOVERY ROUTINE
C
      CALL RCOVER(IABORT)
      RETURN
      END
```

C. KRONOS-Dependent Code

```
      PROGRAM MAIN(INPUT,TAPE5=INPUT,YOUT,TAPE6=YOUT,TAPE59=YOUT,TAPE7,
     1 TAPE8,FILM,TAPE12=FILM,TAPE3)
C
C     KRONOS/6600 MAIN ROUTINE FOR YAQUI
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL YAQUI
      END
```

```
      SUBROUTINE CLOSIT(I)
C
C     KRONOS/6600 ROUTINE TO CLOSE FILES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RETURN
      END
```

```
      SUBROUTINE GETJOB(JOBID)
C
C     KRONOS/6600 ROUTINE TO RETURN THE JOB ID
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL GETJN(JOBID)
      RETURN
      END
```

```
      SUBROUTINE GETJTL(TL)
C
C     KRONOS/6600 ROUTINE TO RETURN THE JOB TIME LIMIT IN SECONDS
C         AS A FLOATING POINT NO.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL GETTL(ITL)
      TL=ITL
      RETURN
```

```
      END
```

```
      SUBROUTINE GETLCM(IFLLCM)
C
C     KRONOS/6600 ROUTINE TO RETURN THE AMOUNT OF LCM AVAILABLE TO
C         THE USER
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      IFLLCM=500000
      RETURN
      END
```

```
      SUBROUTINE GETTPE(IDUM)
C
C     KRONOS/6600 DUMMY ROUTINE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RETURN
      END
```

```
      SUBROUTINE OPENIT(IFILE,MODE)
C
C     KRONOS/6600 ROUTINE TO OPEN A FILE (DUMMY)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RETURN
      END
```

```
      SUBROUTINE SYINIT
C
```

```
C      KRONOS/6600 ROUTINE TO PERFORM ANY NECESSARY SYSTEM INITIALIZATION
C
C      WRITTEN BY J.L.NORTON,LASL T-3,1975
C
       RETURN
       END
```

```
                IDENT   TRAP



*
*      KRONOS/6600 ROUTINE TO INTERCEPT HARDWARE OR SOFTWARE ABORTS
*
*      REGISTERS AND SMALL CORE ARE DUMPED TO OUTPUT,A MESSAGE IS PUT
*          IN THE SYSTEM DAYFILE,AND THE EXTERNAL SUPPLIED IN THE CALL
*          TO TRAP (CALL TRAP(RCOVER)) IS CALLED WHENEVER AN ERROR
*          IS DETECTED
*
*      WRITTEN BY J.L.NORTON,LASL T-3,1975
*
                ENTRY   TRAP
 TRAP           DATA    0
                SX6     B1
                SA6     SAVA
                EREXIT RCOVER
                JP      TRAP
 RCOVER         SYSTEM DMP,R,0,0
                SYSTEM DMP,R,135000B,0
                MESSAGE MESS,3,R
                SA1     SAVA
                SB1     X1+1
                JP      B1
 MESS           DATA    17L*** JOB ABORT ***
                DATA    0
 SAVA           DATA    0
                END
```

D. KRONOS/CROS-Dependent Code

```
      FUNCTION AND(I,J)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO DO BOOLEAN INTERSECTION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER AND
      AND=I.AND.J
      RETURN
      END
```

```
      FUNCTION ANDR(I,J)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO BOOLEAN INTERSECTION
C         AND RETURN THE RESULT IN A FLOATING POINT VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER ANDR
      ANDR=I.AND.J
      RETURN
      END
```

```
      FUNCTION COMP(I)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO COMPLEMENT A WORD
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER COMP
      COMP=.NOT.I
      RETURN
      END
```

```
      FUNCTION COMPR(I)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO COMPLEMENT A WORD
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
```

```
C
      INTEGER COMPR
      COMPR=.NOT.I
      RETURN
      END
```

```
          IDENT   GETIT

*
*      CROS/7600 AND KRONOS/6600 FUNCTION GETIT(IADD) TO PICK UP THE
*          CONTENTS OF ABSOLUTE LOCATION IADD
*
          ENTRY   GETIT
          VFD     42/0HGETIT,18/1
 GETIT    DATA    0
          SA1     B1              PICK UP THE ADDRESS IN X1
          SA2     X1              PICK UP C(IADD)
          BX6     X2              PUT INTO X6 FOR FUNCTION
          JP      GETIT
          END
```

```
      SUBROUTINE LCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO READ OR WRITE LCM FROM
C         OR TO A DISK FILE
C
C     FWA    - FIRST LCM ADDRESS
C     NWRDS  - NO. OF WORDS TO TRANSFER
C     IFILE  - LOGICAL UNIT NO. OF DISK FILE
C     IFLAG  - READ OR WRITE FLAG
C            - 0 - READ DISK
C            - 1 - WRITE DISK
C     IRET   - RETURN FLAG
C            - 0 - RETURN IMMEDIATELY AFTER ISSUING THE I/O REQUEST
C            - 1 - WAIT UNTIL I/O IS COMPLETED BEFORE RETURNING
C              (NOT FUNCTIONAL ON CROS/7600 OR KRONOS/6600)
C     IERROR - ERROR FLAG
C            - 0 - NO ERROR
C            - 1 - ERROR
C              -1 - END-OF-FILE ON INPUT
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
```

```
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM        -----
      COMMON/YSC1/AASC(1)
      INTEGER FWA,FWAP
C
C     CLEAR ERROR FLAG
C
      IERROR=0
C
C     FWAP IS THE BEGINNING LCM ADDRESS OF THE PORTION OF LCM BEING
C          READ OR WRITTEN
C
      FWAP=FWA
C
C     NWRDSD IS THE TOTAL NO. OF WORDS OF LCM THAT HAVE BEEN READ
C          OR WRITTEN
C
      NWRDSD=0
C
C     LOOP FOR READING TO OR WRITING FROM LCM FROM OR TO DISK IN BLOCKS
C
   10 CONTINUE
C
C     NW IS THE NO. OF LCM WORDS TO TRANSFER THIS TIME. IT IS EQUAL
C          TO THE SIZE OF THE SCM BUFFER UNLESS THE NO. OF LCM WORDS
C          LEFT TO TRANSFER IS LESS THAN THE BUFFER SIZE. IN THE LATTER
C          CASE,NW IS JUST SET TO THE NO. OF REMAINING WORDS.
C
      NW=NSCP1
      NWTEST=NWRDSD+NW
      IF(NWTEST.GT.NWRDS) NW=NWRDS-NWRDSD
C
C     SEE WHETHER REQUEST IS READ INTO OR WRITE FROM LCM
C
      IF(IFLAG.NE.0) GO TO 80
C
C     REQUEST IS WRITE LCM (READ DISK)
C
      BUFFER IN(IFILE,1)(AASC(1),AASC(NW))
C
C     WAIT FOR I/O TO COMPLETE
C
   20 IF(UNIT,IFILE) 20,50,40,30
C
C     ERROR OCCURRED IN DISK TRANSFER (EITHER UNEXPECTED EOF,PARITY
C          ERROR,OR RECORD SHORTER THAN EXPECTED). SET THE ERROR FLAG
C          AND RETURN.
C
   30 CONTINUE
      IERROR=1
      RETURN
   40 CONTINUE
      IERROR=-1
      RETURN
   50 CONTINUE
      CALL ECWR(AASC,FWAP,NW,IDUM)
C
```

```
C      BLOCK TRANSFER COMPLETED. UPDATE NWRDSD AND SEE IF THERE IS
C          MORE DATA LEFT TO TRANSFER.
C
   60 CONTINUE
      NWRDSD=NWRDSD+NW
      IF(IFLAG.NE.0) GO TO 70
      N=LENGTH(IFILE)
      IF(N.NE.NW) GO TO 30
   70 CONTINUE
      IF(NWRDSD.GE.NWRDS) RETURN
      FWAP=FWAP+NW
      GO TO 10
C
C      REQUEST IS READ LCM (WRITE DISK)
C
   80 CONTINUE
      CALL ECRD(AASC,FWAP,NW,IDUM)
      BUFFER OUT(IFILE,1)(AASC(1),AASC(NW))
C
C      WAIT FOR I/O TO COMPLETE
C
   90 IF(UNIT,IFILE) 90,60,40,30
      END
```

```
      SUBROUTINE NCODE(NC,IFORM,INTAB,NIN,OUTTAB)
C
C      KRONOS/6600 AND CROS/7600 ROUTINE TO SIMULATE THE CDC ENCODE
C          STATEMENT
C
C      ENCODE(NC,IFORM,OUTTAB) (INTAB(I),I=1,NIN)
C
C      WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      DIMENSION INTAB(NIN)
      INTEGER OUTTAB
      ENCODE(NC,IFORM,OUTTAB) INTAB
      RETURN
      END
```

```
      FUNCTION OR(I,J)
C
C      KRONOS/6600 AND CROS/7600 ROUTINE TO DO BOOLEAN UNION
C
C      WRITTEN BY J.L.NORTON,LASL T-3,1975
C
```

```
      INTEGER OR
      OR=I.OR.J
      RETURN
      END
```

```
      FUNCTION ORR(I,J)
C
C     KRONOS/66ØØ AND CROS/76ØØ ROUTINE TO DO BOOLEAN UNION
C         AND RETURN THE RESULT IN A FLOATING POINT VARIABLE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER ORR
      ORR=I.OR.J
      RETURN
      END
```

```
      FUNCTION RNUMF(X)
C
C     KRONOS/66ØØ AND CROS/76ØØ ROUTINE TO RETURN RANDOM NOS. WITH
C         UNIFORM DISTRIBUTION ON THE INTERVAL (Ø.,1.)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RNUMF=RANF(X)
      RETURN
      END
```

```
      SUBROUTINE SCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
C
C     KRONOS/66ØØ AND CROS/76ØØ ROUTINE TO READ OR WRITE SCM FROM
C         OR TO A DISK FILE
C
C     SEE LCBUFF FOR ARGUMENT DOCUMENTATION
C
C     FWA - BEGINNING OF SCM BLOCK TO BE WRITTEN
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
```

```
      DIMENSION FWA(1)
      IERROR=0
      IF(IFLAG.NE.0) GO TO 10
      BUFFER IN(IFILE,1)(FWA(1),FWA(NWRDS))
      GO TO 20
   10 CONTINUE
      BUFFER OUT(IFILE,1)(FWA(1),FWA(NWRDS))
   20 CONTINUE
      IF(IRET.EQ.0) RETURN
   30 IF(UNIT,IFILE) 30,40,50,60
   40 CONTINUE
      IF(IFLAG.NE.0) RETURN
      N=LENGTH(IFILE)
      IF(N.NE.NWRDS) GO TO 60
      RETURN
   50 CONTINUE
      IERROR=-1
      RETURN
   60 CONTINUE
      IERROR=1
      RETURN
      END
```

```
          IDENT  SHIFT



          ENTRY  SHIFT
*
*         KRONOS/6600 AND CROS/7600 ROUTINE TO REPLACE THE LASL RUN
*             COMPILER IN-LINE SHIFT FUNCTION. THIS IS NECESSARY BECAUSE
*             OF THE NEED TO INCLUDE SHIFT IN AN INTEGER STATEMENT FOR
*             LTSS COMPATIBILITY. HOWEVER,DOING SO FORCES A CALL TO AN
*             EXTERNAL FUNCTION.
*
*         WRITTEN BY J.L.NORTON,LASL T-3,1975
*
          VFD    42/0HSHIFT,18/2
 SHIFT    DATA   0
          SA1    B2
          SA2    B1
          ZR     X1,ZERO
          PL     X1,LEFT
          BX3    -X1
          SB3    X3
          AX6    B3,X2
          JP     SHIFT
 LEFT     SB3    X1
          LX6    B3,X2
          JP     SHIFT
 ZERO     BX6    X2
```

```
              JP      SHIFT
              END
```

```
              IDENT   STORIT
*
*       CROS/7600 AND KRONOS/6600 SUBROUTINE STORIT(IWORD,IADD) WHICH
*           PUTS IWORD INTO ABSOLUTE SCM LOCATION IADD
*
              ENTRY   STORIT
              VFD     42/0HSTORIT,18/2
     STORIT   DATA    0
              SA1     B1                PICK UP IWORD IN X1
              SA2     B2                PICK UP IADD IN X2
              BX6     X1                STORE IWORD INTO IADD
              SA6     X2
              JP      STORIT
              END
```

```
      SUBROUTINE TTYTST(IFLAG)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO DUMMY UP CAPABILITY OF
C         LTSS TO MODIFY THE DROPFILE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      IFLAG=0
      RETURN
      END
```

```
      SUBROUTINE YAQUI
C
C     YAQUI IS A TWO-DIMENSIONAL FLUID DYNAMICS CODE THAT COMBINES
C         THE ICE (IMPLICIT CONTINUOUS EULERIAN) AND ALE (ARBITRARY
C         LAGRANGIAN-EULERIAN) METHODS,ALLOWING CALCULATION OF FLOWS
C         AT ALL SPEEDS.
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,HANS RUPPEL,AND C.W.HIRT,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
C     ******************************************************************
C     LCM IS SET UP IN THE FOLLOWING FASHION --
C         BLOCK 1,NLCP1 WORDS LONG,ARRAY DATA
C         BLOCK 2,NLCP2 WORDS LONG,PARTICLE POSITIONS AND MASSES
C         BLOCK 3,NLCP3 WORDS LONG,TIME-DEPENDENT PARTICLE POSITIONS
C         BLOCK 4,IFLMSZ WORDS LONG,FILM FILE BUFFER FOR LTSS
C         BLOCK 5,NLCP4 WORDS LONG,SCRATCH AREA
C     ******************************************************************
C
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTOS,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TPS,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
      COMMON/FTABC/FTAB(2)
      INTEGER AA1
      EXTERNAL YEXIT
      DATA FTAB/6,12/
      DATA NSCP1/9600/,ITABP/1000/,ITABXP/101/,ITABYP/151/
      DATA IPFB/200/,NP1/10/,NP2/200/,NLCP1/72300/,NLCP2/5000/
      DATA NLCP3/2000/,NLCP4/1000/,IFLMSZ/4000/
      DATA TP/0./
```

```
C
C     DO ANY NECESSARY SYSTEM INITIALIZATION
C
      CALL SYINIT
C
C     INITIALIZE THE ERROR RECOVERY ROUTINE
C
      CALL TRAP(YEXIT)
C
C     GET THE CP TIME AT JOB STARTUP
C
      CALL SECOND(TP)
      WRITE(59,30) TP
C
C     INITIALIZE
C
      CALL YINIT
C
C     CHECK TYPE OF RUN
C
      IF(RESTRT) GO TO 10
C
C     RUN IS A NEW PROBLEM, GO GENERATE IT.
C
      CALL YASET
      GO TO 20
C
C     RUN IS A PROBLEM RESTART, READ THE DUMP TAPE.
C
   10 CONTINUE
      CALL YARSRT
C
C     GET RID OF THE DUMP FILE
C
      CALL CLOSIT(7)
   20 CONTINUE
C
C     EXECUTE THE MAIN CODE
C
      CALL YAQUI2
C
C     RUN TERMINATION, EXIT.
C
      CALL UNCLE(2,5HYAQUI,15,15HRUN TERMINATION)
C
   30 FORMAT(1H ,29HBEGIN CODE EXECUTION AT CP = ,F10.4)
      END
```

```
      SUBROUTINE AIR
C
C     SEMI-PHYSICAL FIT TO THE EQUATION OF STATE OF AIR
```

```
C
C     DENSITIES FROM 10**2 TO 10**(-7) NORMAL DENSITY
C     PRESSURE=(GAMMA-1.)*RHO*E, WHERE GAMMA IS  A FUNCTON OF
C         DENSITY AND ENERGY
C     RHO=MATERIAL DENSITY
C     EJLN=MATERIAL SPECIFIC INTERNAL ENERGY
C     GMONE=GAMMA-1.
C     CONCJ=RELATIVE CONCENTRATION, SEE NOTE BELOW.
C
C     ALL UNITS ARE CGS FOR INPUT QUANTITIES
C
C     ORIGINALLY OBTAINED FROM THE AIR FORCE WEAPONS LAB
C     MODIFIED BY J.L.NORTON,LASL T-3,1974
C
      COMMON/EQNST/RHO,EJLN,GMONE,CONCJ
C
C     IN THIS VERSION OF THE CODE,TWO EQNS OF STATE ARE BEING USED,AIR
C         AND METHANE. THE FINAL VALUE OF GMONE IS DETERMINED BY
C         THE RELATION  GMONE(FINAL)=CONCJ*GMONE(METHANE)
C                                            +(1.-CONCJ)*GMONE(AIR).
C         THUS,CONCJ=1. YIELDS A METHANE GAMMA AND CONCJ=0.,AIR.
C         ANYTHING IN BETWEEN USES A LINEAR COMBINATION OF THE TWO.
C
C     CHECK TO MAKE SURE CONCJ DOES NOT EXCEED 1. IF IT DOES,SET IT TO 1
C
      IF(CONCJ.GT.1.) CONCJ=1.
C
C     IFLAG=1 SIGNALS AIR TO DO THE GAMMA LINEAR COMBINATION. IF
C         CONCJ=0.,THIS IS UNNECESSARY.
C
      IFLAG=0
C
C     IF CONCJ=0.,SKIP THE METHANE CALCULATION ENTIRELY
C
      IF(CONCJ.EQ.0.) GO TO 10
C
C     GO GET THE METHANE GAMMA
C
      CALL MTHANE
C
C     IF CONCJ=1.,WE ARE ALL DONE
C
      IF(CONCJ.EQ.1.) RETURN
C
C     TURN ON THE LINEAR COMBINATION FLAG,SAVE THE METHANE GMONE,AND
C         CONTINUE ON WITH THE AIR EOS CALCULATION
C
      IFLAG=1
      GMM=GMONE
C
C     CALCULATE GAMMA-1. FOR AIR
C
   10 CONTINUE
C
C     RHOZ IS THE NORMAL AIR DENSITY
C
      RHOZ=1.293E-3
```

```
C
C     RHOT IS THE COMPRESSION
C
      RHOT=RHO/RHOZ
C
C     MAKE E POSITIVE AND CONVERT TO SI UNITS (TJ/MG)
C
      E=1.E-10*ABS(EJLN)
C
C     THE ENERGY AT WHICH OXYGEN AND NITROGEN DISSOCIATE IS A
C         FUNCTION OF DENSITY
C
      E1=(8.5-E)/.975
C
C     THE FERMI-DIRAC FUNCTON IS ONLY COMPUTED WITHIN 5.*DELTA E OF
C         EACH TRANSITION, OTHERWISE IT IS ZERO OR ONE.
C
      IF(ABS(E1)-5.)50,20,20
   20 IF(E1) 40,40,30
   30 FO=EXP(-E/4.46)
      FON=0.
      WS=1.
      GO TO 60
   40 FO=0.
      FON=EXP(-E/6.63)
      WS=0.
      GO TO 60
   50 DE1=.975*RHOT**.05
      EE1=8.5+.357*ALOG10(RHOT)
      E1=(EE1-E)/DE1
      WS=1./(EXP(-E1)+1.)
      FO=EXP(-E/4.46)*WS
      FON=EXP(-E/6.63)*(1.-WS)
C
C     THE DENSITY DEPENDENCE ONLY OCCURS AVOVE E=1., AND IT IS OF
C          THE FORM (RHO/RHOZ**(CONSTANT*LOG(E))). THE CONSTANT
C          MAKES A TRANSITION FROM .048 TO .029 AS THE OXYGEN
C          DISSOCIATES AND THE DENSITY SPREAD BECOMES CONSTANT BEYOND
C          THE SECOND PEAK.
C
   60 IF(E-1.)70,70,80
   70 BETA=0.
      GO TO 90
   80 BETA=(.048*WS+.032*(1.-WS))*ALOG10(E)
   90 E2=(E-40.)/3.
      IF(ABS(E2)-5.)130,100,100
  100 IF(E2) 110,110,120
  110 FN=0.
      WS=0.
      GO TO 140
  120 FN=EXP(-E/25.5)
      WS=1.
      GO TO 140
  130 DE2=4.*RHOT**.085
      EE2=45.*RHOT**.0157
      E2=(E-EE2)/DE2
      WS=1./(EXP(-E2)+1.)
```

```
      FN=EXP(-E/25.5)*WS
  140 E3=(E-160.)/6.
      BETA=BETA*(1.-WS)+.045*WS
      IF(E3+5.)150,160,160
  150 FE=0.
      GO TO 170
  160 FE=1./(EXP(-E3)+1.)
  170 RHOFAC=RHOT**BETA
      GMONE=(.161+.255*FO+.280*FON+.137*FN+.050*FE)*RHOFAC
C
C     IF IFLAG IS ZERO,ONLY AIR IS DESIRED AND WE ARE ALL DONE
C
      IF(IFLAG.EQ.0) RETURN
C
C     CALCULATE THE GAMMA-1. LINEAR COMBINATION
C
      GMONE=CONCJ*GMM+(1.-CONCJ)*GMONE
      RETURN
      END
```

```
---------------------------------------------------------------------------------------------
```

```
      SUBROUTINE BC(IFLAG)
C
C     ROUTINE TO SET RIGID WALL BOUNDARY CONDITIONS IN YAQUI
C
C     IFLAG = INDEX INDICATING WHICH VELOCITY ARRAYS ARE TO BE SET
C           = 1 = UTIL,VTIL
C           = 2 = UL,VL
C           = 3 = UP,VP
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPO8,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1.IP2.ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
```

```
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFD2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZBIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ===== END COMDECK YAQSC      =====
*     ===== BEGIN COMDECK YAQEQ      =====
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUSVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ===== END COMDECK YAQEQ      =====
*     ===== END COMDECK YSTORE     =====
C     SET UP THE LOOP OVER CELLS
C
      CALL START
C
C     SET THE BOTTOM BOUNDARY VELOCITIES
C
      IJSV=IJ
      DO 40 I=1,IP1
      GO TO (10,20,30),IFLAG
   10 CONTINUE
      VTIL(IJ)=0.
      GO TO 40
   20 CONTINUE
      VL(IJ)=0.
      GO TO 40
   30 CONTINUE
      VP(IJ)=0.
   40 IJ=IJ+NQ
      IJ=IJSV
C
C     SET THE LEFT AND RIGHT BOUNDARIES
C
      DO 90 J=2,JP1
      GO TO (50,60,70),IFLAG
   50 CONTINUE
```

```
      UTIL(IJ)=0.
      UTIL(IJ+NQIB)=0.
      GO TO 80
   60 CONTINUE
      UL(IJ)=0.
      UL(IJ+NQIB)=0.
      GO TO 80
   70 CONTINUE
      UP(IJ)=0.
      UP(IJ+NQIB)=0.
   80 CONTINUE
      CALL LOOP
   90 CONTINUE
C
C     SET THE TOP BOUNDARY
C
      DO 130 I=1,IP1
      GO TO (100,110,120),IFLAG
  100 CONTINUE
      VTIL(IJ)=0.
      GO TO 130
  110 CONTINUE
      VL(IJ)=0.
      GO TO 130
  120 CONTINUE
      VP(IJ)=0.
  130 IJ=IJ+NQ
C
C     SET THE UPPER RIGHTHAND CORNER U
C
      IJ=IJ-NQ
      GO TO (140,150,160),IFLAG
  140 CONTINUE
      UTIL(IJ)=0.
      GO TO 170
  150 CONTINUE
      UL(IJ)=0.
      GO TO 170
  160 CONTINUE
      UP(IJ)=0.
  170 CONTINUE
      CALL DONE
      RETURN
      END
```

```
      SUBROUTINE CINIT
C
C     ROUTINE TO CALCULATE QUANTITIES FOR THE CYCLE PRINT AND
C         INITIALIZE FOR THE NEXT CYCLE
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
```

```
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ***** BEGIN COMDECK PARAM      *****
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ***** END COMDECK PARAM        *****
*     ***** BEGIN COMDECK YSTORE     *****
*     ***** BEGIN COMDECK YAQDIM     *****
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ***** END COMDECK YAQDIM       *****
*     ***** BEGIN COMDECK YAQSC      *****
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ***** END COMDECK YAQSC        *****
*     ***** BEGIN COMDECK YAQEQ      *****
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ***** END COMDECK YAQEQ        *****
```

```
*      ----- END COMDECK YSTORE     -----
       COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
C
C      INITIALIZE LOOP VARIABLES
C
       POTE=0.
       TK=0.
       TI=0.
       UMOM=0.
       VMOM=0.
       TMASSV=TMASS
       TMASS=0.
       TUGENG=0.
       CIRC=0.
       TMAX=0.
       TGMX=0.
       ITM=0
       ITG=0
       JTM=0
       JTG=0
C
C      LOOP OVER ALL REAL ZONES IN THE MESH
C
       CALL START
       DO 60 J=2,JP1
       DO 50 I=1,IBAR
       IPJ=IJ+NQ
       IPJP=IJP+NQ
C
C      TMAX IS THE MAXIMUM SIE OF ANY CELL IN THE MESH, IT IS FOUND
C          IN CELL ITM,JTM.
C
       IF(SIE(IJ).LE.TMAX) GO TO 10
       ITM=I
       JTM=J
       TMAX=SIE(IJ)
C
C      TGMX IS THE MAXIMUM SIE GRADIENT AND (ITG,JTG) IS THE CELL
C          IN WHICH IT OCCURS
C
    10 CONTINUE
       SAVA=(X(IPJ)-X(IJ))**2+(Y(IPJ)-Y(IJ))**2
       SAVB=(X(IJP)-X(IJ))**2+(Y(IJP)-Y(IJ))**2
       SAVA=ABS(SIE(IJ)-SIE(IPJ))/SQRT(SAVA)
       SAVB=ABS(SIE(IJ)-SIE(IJP))/SQRT(SAVB)
       SAV=AMAX1(SAVA,SAVB)
       IF(SAV.LT.TGMX) GO TO 20
       ITG=I
       JTG=J
       TGMX=SAV
    20 CONTINUE
C
C      CALCULATE THE CIRCULATION AROUND THE PROBLEM BOUNDARIES
C
       IF(I.EQ.1) CIRC=CIRC+0.5*(V(IJ)+V(IJP))*(Y(IJP)-Y(IJ))
       IF(I.EQ.IM1) CIRC=CIRC-0.5*(V(IJ)+V(IJP))*(Y(IJP)-Y(IJ))
       IF(J.EQ.3) CIRC=CIRC-0.5*(U(IJ)+U(IPJ))*(X(IPJ)-X(IJ))
```

```
      IF(J.EQ.JBAR) CIRC=CIRC+0.5*(U(IJ)+U(IPJ))*(X(IPJ)-X(IJ))
C
C     XMSENG IS THE MASS IN CELL I,J
C     TMASS IS THE TOTAL PROBLEM MASS
C
      XMSENG=RO(IJ)/RVOL(IJ)
      TMASS=TMASS+XMSENG
C
C     SPENGK IS THE SPECIFIC KINETIC ENERGY OF CELL I,J
C
      SPENGK=0.125*(U(IPJ)**2+U(IPJP)**2+U(IJP)**2+U(IJ)**2+V(IPJ)**2+V(
     1 IPJP)**2+V(IJP)**2+V(IJ)**2)
C
C     TK IS THE TOTAL KINETIC ENERGY IN THE PROBLEM
C
      TK=TK+SPENGK*XMSENG
C
C     TI IS THE TOTAL INTERNAL ENERGY IN THE PROBLEM
C
      TI=TI+XMSENG*SIE(IJ)
C
C     UMOM AND VMOM ARE THE RADIAL AND AXIAL TOTAL MOMENTA,
C         RESPECTIVELY
C
      UMOM=UMOM+0.25*XMSENG*(U(IPJ)+U(IPJP)+U(IJP)+U(IJ))
      VMOM=VMOM+0.25*XMSENG*(V(IPJ)+V(IPJP)+V(IJP)+V(IJ))
C
C     POTE IS THE TOTAL GRAVITATIONAL POTENTIAL ENERGY (WITH G=1.).
C         NOTE THAT VERTEX Y'S AND MASSES ARE USED.
C
      POTE=POTE+Y(IJ)/RM(IJ)
C
C     ADD THE J=JP2 VERTEX INTO THE POTENTIAL ENERGY
C
      IF(J.EQ.JP1) POTE=POTE+Y(IJP)/RM(IJP)
      IF(TURB) TUGENG=TUGENG+TUQ(IJ)*XMSENG
C
C     NEW PRESSURE MUST BE CALCULATED. P ARRAY CURRENTLY HOLDS
C         VELOCITIES DUE TO EQUIVALENCING OF P WITH UP.
C
C     PUT A LOWER BOUND OF ZERO ON THE SIE
C
      SIET=AMAX1(SIE(IJ),0.)
C
C     GO GET THE CELL GAMMA-1
C
      ROTMP=RO(IJ)
      ETMP=SIET
      CONCJ=CONC(IJ)
      CALL AIR
      GM1=GMONE
C
C     COMPUTE PRESSURE
C
      P(IJ)=GM1*ROTMP*SIET
C
C     SET PRESSURE OF BOTTOM FICTITIOUS ROW USING GAMMA OF CELLS ABOVE
```

```
      C
            IF(J.EQ.2) P(IJM)=GM1*RO(IJM)*SIE(IJM)
      C
      C     SET PRESSURE OF TOP FICTITIOUS ROW USING GAMMA OF CELLS BELOW
      C
            IF(J.EQ.JP1) P(IJP)=GM1*RO(IJP)*SIE(IJP)
      C
      C     SEE IF WE ARE PROCESSING THE RIGHTMOST REAL COLUMN
      C
            IF(I.NE.IBAR) GO TO 40
      C
      C     YES.
      C
      C     SET PRESSURE OF RIGHT FICTITIOUS COLUMN USING GAMMA OF CELLS
      C         TO THE LEFT
      C
            P(IPJ)=GM1*RO(IPJ)*SIE(IPJ)
      C
      C     SEE IF WE ARE PROCESSING THE BOTTOM REAL ROW
      C
            IF(J.NE.2) GO TO 30
      C
      C     YES.
      C
      C     SET PRESSURE OF BOTTOM CELL IN RIGHT FICTITIOUS COLUMN USING
      C         GAMMA OF CELL (IBAR,2)
      C
            IPJM=IJM+NQ
            P(IPJM)=GM1*RO(IPJM)*SIE(IPJM)
            GO TO 40
         30 CONTINUE
      C
      C     NO. SEE IF WE ARE PROCESSING THE TOP REAL ROW.
      C
            IF(J.NE.JP1) GO TO 40
      C
      C     YES.
      C
      C     SET PRESSURE OF UPPER CELL IN RIGHT FICTITIOUS COLUMN USING
      C         GAMMA OF CELL (IBAR,JP1)
      C
            P(IPJP)=GM1*RO(IPJP)*SIE(IPJP)
         40 CONTINUE
            IJ=IPJ
            IJP=IJP+NQ
            IJM=IJM+NQ
         50 CONTINUE
      C
      C     ADD THE I=IP1 VERTEX INTO THE POTENTIAL ENERGY
      C
            POTE=POTE+Y(IJ)/RM(IJ)
      C
      C     ADD THE (IP1,JP2) VERTEX INTO THE POTENTIAL ENERGY
      C
            IF(J.EQ.JP1) POTE=POTE+Y(IJP)/RM(IJP)
            CALL LOOP
         60 CONTINUE
```

```
      CALL DONE
      RETURN
      END
```

```
      SUBROUTINE CONTOR(ILOG,ITITLE,NWT)
C
C     ROUTINE FOR DOING CONTOUR PLOTS IN YAQUI
C
C     ILOG IS 0 IF LINEAR CONTOUR INCREMENTS ARE TO BE USED
C     ILOG IS 1 FOR LOGARITHMIC CONTOUR INCREMENTS
C     ITITLE IS THE HOLLERITH TITLE TO BE USED ON THE PLOT
C     NWT IS THE NO. OF COMPUTER WORDS IN THE TITLE
C
C     THE QUANTITY TO BE PLOTTED IS FOUND IN CQ
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
```

```
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ***** END COMDECK YAQSC      *****
*     ***** BEGIN COMDECK YAQEQ      *****
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ***** END COMDECK YAQEQ      *****
*     ***** END COMDECK YSTORE     *****
*     ***** BEGIN COMDECK ASTORE     *****
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*     ***** END COMDECK ASTORE     *****
*     ***** BEGIN COMDECK PCALL      *****
      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*     ***** END COMDECK PCALL      *****
      DIMENSION BCD(2),ITITLE(1)
      DIMENSION ILABEL(26),IPLBL(26)
      DATA ILABEL/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
     1 1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/
      DATA IPLBL/17,18,19,20,21,22,23,24,25,33,34,35,36,37,38,39,40,41,
     1 50,51,52,53,54,55,56,57/
      DATA BCD/1H /
C
C     FIND THE MINIMUM AND MAXIMUM (QMN AND QMX) OF THE QUANTITY
C         TO BE PLOTTED
C
      QMN=1.E30
      QMX=-QMN
      CALL START
      DO 20 J=2,JP1
      DO 10 I=1,IBAR
      QMN=AMIN1(CQ(IJ),QMN)
      QMX=AMAX1(CQ(IJ),QMX)
   10 IJ=IJ+NQ
      CALL LOOP
   20 CONTINUE
C
C     SET THE CONTOUR INCREMENT. THE BASIC ALGORITHM IS BASED ON
C         THE DESIRE TO HAVE THE CONTOUR INCREMENT AN INTEGRAL POWER
C         OF TWO AND THE CONTOURS THEMSELVES INTEGRAL MULTIPLES OF
```

```
C          THIS POWER OF TWO. THESE FACTS INSURE THAT TWO PLOTS
C          VERY NEARLY THE SAME WILL HAVE EXACTLY THE SAME CONTOUR
C          VALUES. FIRST MAKE AN INITIAL GUESS FOR THE CONTOUR
C          INCREMENT (DQ) SO THAT THERE ARE AT LEAST TEN CONTOURS.
C
      DQ=(QMX-QMN)/10.
C
C     IF THE INCREMENT IS ZERO OR NEGATIVE,BYPASS THE PLOT
C
      IF(DQ.LE.0.) RETURN
C
C     FIND A POWER OF TWO JUST LESS THAN DQ BUT AT LEAST 2**-33
C         (WHICH IS ABOUT 10**-10)
C
      TEST=2.**(-32)
   30 CONTINUE
      IF(TEST.GE.DQ) GO TO 40
      TEST=2.*TEST
      GO TO 30
   40 CONTINUE
C
C     SET THE CONTOUR INCREMENT TO IT
C
      DQ=.5*TEST
C
C     NOW DETERMINE THE CONTOUR VALUES. K WILL BE THE FINAL NO.
C         OF CONTOURS. THE FIRST CONTOUR IS THE NEAREST INTEGRAL
C         MULTIPLE OF DQ LESS THAN QMN.
C
   50 CONTINUE
      ITEST=QMN/DQ
      IF(ITEST.LT.0) ITEST=ITEST-1
      CON(1)=FLOAT(ITEST)*DQ
      DO 60 K=2,27
      IF(CON(K-1).GT.QMX) GO TO 70
      IF(K.EQ.27) GO TO 60
      CON(K)=CON(K-1)+DQ
   60 CONTINUE
C
C     TOO MANY CONTOURS. GO INCREASE DQ AND TRY AGAIN.
C
      GO TO 80
   70 CONTINUE
      K=K-1
C
C     IF THERE ARE MORE THAN 20 CONTOURS,INCREASE DQ AND GO CALCULATE
C         THE CONTOURS AGAIN.
C
      IF(K.LE.20) GO TO 90
   80 CONTINUE
      DQ=2.*DQ
      GO TO 50
   90 CONTINUE
C
C     ADVANCE THE FILM TO THE NEXT FRAME
C
      CALL ADV(1)
```

```
C
C      SET THE HEIGHT OF THE FIRST LINE OF THE CONTOUR LABELS
C
       CALL LINCNT(6)
C
C      LIST THE CONTOURS ON THE PLOT
C
       DO 100 KK=1,K
       WRITE(IFD,250) ILABEL(KK),CON(KK)
   100 CONTINUE
       WRITE(IFD,260) QMN,QMX
C
C      LABEL THE TYPE OF PLOT
C
       CALL LINCNT(58)
       WRITE(IFD,270)(ITITLE(I),I=1,NWT)
       WRITE(IFD,290) JNM,NAME,T,NCYC
C
C      PUT CONTOUR INFORMATION OUT TO CYCLE SUMMARY
C
       IPDX=6
       WRITE(IPDX,280)(ITITLE(I),I=1,NWT)
       WRITE(IPDX,300) QMN,QMX,CON(1),CON(K-1),DQ
C
C      LOOP TO DO THE ACTUAL PLOT
C
C      CALL START PULLS IN ROWS 1,2,AND 3 AND LEAVES IJ POINTING AT
C          ROW 2
C
       CALL START
       DO 240 J=2,JBAR
C
C      CALL LOOP AT THIS POINT SHIFTS THE ROWS IN SCM UP ONE SO THAT
C          IJ POINTS AT ROW J+1 INSTEAD OF ROW J
C
       CALL LOOP
       DO 230 I=1,IM1
       IPJ=IJ+NQ
       IPJM=IJM+NQ
C
C      N=0 SIGNIFIES THAT XCO AND YCO HAVE NOT BEEN COMPUTED FOR THIS
C          I AND J
C
       N=0
C
C      LOOP OVER ALL OF THE CONTOUR VALUES
C
       DO 220 KK=1,K
C
C      SET FLAGS.
C
C      CONSIDER THE FOUR CELLS (I,J),(I+1,J),(I,J+1), AND (I+1,J+1)
C          AND LET THEM BE DENOTED CELLS 1,2,3,AND 4,RESPECTIVELY.
C          THEN,KN IS ZERO IF THE QUANTITY TO BE PLOTTED IN CELL N
C          ISGREATER THAN THE CURRENT CONTOUR VALUE, AND 1 IF
C          THE QUANTITY TO BE PLOTTED IS LESS THAN OR EQUAL TO
C          THE CURRENT CONTOUR VALUE.
```

```
      C
      C     NOTE THAT WHAT IS BEING CONSIDERED IS THE QUADRILATERAL FORMED BY
      C         CONNECTING THE CENTERS OF THE FOUR CELLS IN THE ORDER
      C         1-2-4-3-1. LET THIS BE KNOWN AS THE CURRENT CONTOUR AREA.
      C
            K1=0
            K2=0
            K3=0
            K4=0
            IF(CQ(IJM).LE.CON(KK)) K1=1
            IF(CQ(IPJM).LE.CON(KK)) K2=1
            IF(CQ(IJ).LE.CON(KK)) K3=1
            IF(CQ(IPJ).LE.CON(KK)) K4=1
      C
      C     IF ALL THE FLAGS ARE 1 OR 0, THEN THE CURRENT CONTOUR IS EITHER
      C         GREATER THAN OR LESS THAN OR EQUAL TO THE PLOT QUANTITIES
      C         AT ALL FOUR CORNERS OF THE CURRENT CONTOUR AREA. THUS,THE
      C         CURRENT CONTOUR DOES NOT CROSS THE CURRENT CONTOUR AREA.
      C         GO ON TO THE NEXT CONTOUR.
      C
            IF(K1*K2*K3*K4.NE.0.OR.K1+K2+K3+K4.EQ.0) GO TO 220
      C
      C     CALCULATE THE CENTERS OF CELLS 1,2,3,AND 4 (THE VERTICES OF THE
      C         CURRENT CONTOUR AREA) IF THIS HAS NOT ALREADY BEEN DONE.
      C
            IF(N.GT.0) GO TO 130
            IJB=IJM
            IJA=IJ
            DO 120 JJ=1,2
            DO 110 II=1,2
            IPJB=IJB+NQ
            IPJA=IJA+NQ
            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
        110 IJB=IPJB
            IJB=IJ
        120 IJA=IJP
        130 CONTINUE
      C
      C     DETERMINE WHICH SIDES OF THE CURRENT CONTOUR AREA THE CURRENT
      C         CONTOUR CROSSES.
      C     LL COUNTS HOW MANY SIDES THAT HAVE BEEN FOUND TO BE CROSSED
      C
            LL=0
      C
      C     SEE IF THE LEFT SIDE IS CROSSED
      C
            IF(K1+K3.NE.1) GO TO 140
      C
      C     YES. IC1 AND IC2 ARE THE VERTEX NUMBERS OF THE CURRENT CONTOUR
      C         AREA THAT BOUND THE SIDE CROSSED. IJ1 AND IJ2 ARE THE INDICES
      C         OF THE MESH CELLS CONTAINING THESE VERTICES.
      C
            IC1=1
            IC2=3
```

```
      IJ1=IJM
      IJ2=IJ
C
C     GO SET UP PLOT COORDINATES FOR THIS CROSSING
C
      GO TO 170
C
C     SEE IF THE BOTTOM IS CROSSED BY THE CONTOUR
C
  140 IF(K1+K2.NE.1) GO TO 150
C
C     YES
C
      IC1=1
      IC2=2
      IJ1=IJM
      IJ2=IPJM
      GO TO 170
C
C     SEE IF RIGHT IS CROSSED
C
  150 IF(K2+K4.NE.1) GO TO 160
      IC1=2
      IC2=4
      IJ1=IPJM
      IJ2=IPJ
      GO TO 170
C
C     SEE IF TOP IS CROSSED
C
  160 IF(K3+K4.NE.1) GO TO 220
      IC1=3
      IC2=4
      IJ1=IJ
      IJ2=IPJ
C
C     INCREMENT THE NO. OF SIDES CROSSED
C
  170 LL=LL+1
C
C     CONTOUR LIES BETWEEN CQ(IJ1) AND CQ(IJ2). CALCULATE HOW FAR
C         ALONG THE LINE CONNECTING VERTICES IC1 AND IC2 THE
C         INTERSECTION SHOULD ACTUALLY LIE.
C
      XX=(CON(KK)-CQ(IJ1))/(CQ(IJ2)-CQ(IJ1))
C
C     IX1(LL) AND IY1(LL) ARE THE RASTER COORDINATES OF THE POINT OF
C         INTERSECTION OF THE CONTOUR AND THE CONTOUR AREA SIDE
C         FOR INTERSECTION LL
C
      IX1(LL)=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XL)*XCONV
      IY1(LL)=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YLB)*YCONV
C
C     IF FOR SOME REASON THE INTERSECTION POINT LIES OUTSIDE OF THE
C         PLOTTING RECTANGLE,IGNORE THE INTERSECTION
C
      IF(IY1(LL).GT.IYB.OR.IY1(LL).LT.IYT) GO TO 210
```

```
      IF(IX1(LL).LT.IXL.OR.(IX1(LL).GT.IXR)) GO TO 210
C
C     IF ONLY ONE INTERSECTION HAS BEEN LOCATED,GO BACK AND LOOK
C         FOR ANOTHER
C
      IF(LL.GE.2) GO TO 200
  180 CONTINUE
C
C     IF IC1=2,RIGHT SIDE WAS INTERSECTED LAST. GO BACK AND GET
C         THE TOP SIDE. IF IC1=3,TOP SIDE WAS INTERSECTED LAST.
C         ALL SIDES HAVE BEEN CHECKED. GO ON TO NEXT CONTOUR.
C
      GO TO (190,160,220),IC1
C
C     IC1=1. EITHER LEFT OR BOTTOM WAS LAST INTERSECTED. IF BOTTOM
C         (IC2=2), GO BACK AND CHECK THE RIGHT. IF LEFT (IC2=3),
C         GO BACK AND CHECK THE BOTTOM.
C
  190 CONTINUE
      IF(IC2.EQ.2) GO TO 150
      GO TO 140
  200 CONTINUE
C
C     TWO SIDES HAVE BEEN CROSSED. CONNECT THE POINTS OF
C         INTERSECTION.
C
      CALL DRV(IX1,IY1,IX2,IY2)
C
C     PLOT THE LABEL ON THE FIRST INTERSECTION POINT
C
      CALL PLT(IX1,IY1,IPLBL(KK))
  210 CONTINUE
C
C     START ON A NEW INTERSECTION PAIR IF BOTTOM WAS LAST SIDE
C         CHECKED. OTHERWISE,THERE ARE NOT TWO POSSIBLE INTERSECTIONS
C         LEFT SO WE ARE DONE.
C
      LL=0
      IF(IJ2.EQ.IPJM) GO TO 180
  220 CONTINUE
      IJM=IPJM
      IJ=IPJ
  230 IJP=IJP+NQ
  240 CONTINUE
C
C     DRAW THE PLOT FRAME AND LABEL IT
C
      CALL TICBOX
      RETURN
C
  250 FORMAT(100XA1,2X1PE10.3)
  260 FORMAT(98X3HQMN,2X1PE10.3/98X3HQMX,2XE10.3)
  270 FORMAT(1H ,8A10)
  280 FORMAT(1H ,16HCONTOUR PLOT OF ,6A10)
  290 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
  300 FORMAT(5H MIN=1PE12.5,5H MAX=E12.5,3H L=E12.5,3H H=E12.5,/1H ,16X
     1 4H DQ=E12.5)
```

```
          END
```

```
          SUBROUTINE DEFINE
    C
    C     ROUTINE TO INITIALIZE PROBLEM INPUT VARIABLES
    C
    C     WRITTEN BY J.L.NORTON,LASL T=3,1974
    C
    *     ===== BEGIN COMDECK PARAM      =====
          COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
         1 NLCP3,NLCP4,IFLMSZ
    *     ===== END COMDECK PARAM      =====
    *     ===== BEGIN COMDECK YAQSC      =====
          LOGICAL RESTRT,FILM,PAPER,TURB
          REAL LAM,MU
    C     COMMON/YSC1/AASC(NSCP1)
          COMMON/YSC1/AASC(9600)
          COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
         1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
         2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
         3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
         4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
          COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
         1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
         2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
         3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
          COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
         1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
         2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
         3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
         4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
         5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
         6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
          COMMON/YSC2/ZZ
    C     COMMON/YSC4/ITAB(ITABP)
          COMMON/YSC4/ITAB(1000)
          COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
    *     ===== END COMDECK YAQSC      =====
    C     MU AND LAM ARE THE VISCOSITY COEFFICIENTS
    C
          MU=0.
          LAM=.6
    C
    C     OM IS THE PHASE TWO ITERATION RELAXATION PARAMETER OMEGA
    C
          OM=1.
    C
    C     EPS IS THE PRESSURE ITERATION CONVERGENCE CRITERION
    C
          EPS=1.E-5
    C
```

```
      C     GR IS THE RADIAL COMPONENT OF THE BODY FORCE FELT BY THE ENTIRE
      C         PROBLEM
      C
            GR=0.
      C
      C     GZ IS THE AXIAL COMPONENT OF THE BODY FORCE FELT BY THE ENTIRE
      C         PROBLEM (USUALLY GRAVITY IN THE -Z DIRECTION)
      C
            GZ=-980.
      C
      C     FREZXR IS THE RATIO OF ZONE DR-S AS ONE MOVES AWAY FROM
      C         THE REGION OF CONSTANT ZONING TOWARD THE RIGHT BOUNDARY
      C
            FREZXR=1.
      C
      C     FREZYT IS THE RATIO OF ZONE DZ-S AS ONE MOVES AWAY FROM
      C         THE REGION OF CONSTANT ZONING TOWARD THE TOP BOUNDARY
      C
            FREZYT=1.
      C
      C     FREZYB IS THE RATIO OF ZONE DZ-S AS ONE MOVES AWAY FROM
      C         THE REGION OF CONSTANT ZONING TOWARD THE BOTTOM BOUNDARY
      C
            FREZYB=1.
      C
      C     YB IS THE ACTUAL BOTTOM OF THE GRID
      C
            YB=0.
      C
      C     REZY0 IS THE CENTER OF THE REGION OF UNIFORM ZONING, EITHER YB OR
      C         REZY0 MAY BE SPECIFIED.
      C
            REZY0=0.
      C
      C     REZRON IS THE INITIAL DENSITY OF THE AMBIENT ATMOSPHERE AT Z=REZY0
      C
            REZRON=.001
      C                                                  4
      C     REZSIE IS THE SPECIFIC INTERNAL ENERGY OF THE AMBIENT ATMOSPHERE
      C
            REZSIE=2.E10
      C
      C     IBAR IS THE NUMBER OF REAL ZONES IN THE RADIAL DIRECTION
      C
            IBAR=0
      C
      C     JBAR IS THE NUMBER OF REAL ZONES IN THE AXIAL DIRECTION
      C
            JBAR=0
      C
      C     IUNF IS THE NUMBER OF UNIFORM ZONES IN THE RADIAL DIRECTION
      C
            IUNF=0
      C
      C     JUNF IS THE NUMBER OF UNIFORM ZONES IN THE AXIAL DIRECTION
      C
            JUNF=0
```

```
C
C      JCEN IS THE J VALUE ABOVE AND BELOW WHICH THERE ARE AN EQUAL
C          NUMBER OF UNIFORM ZONES
C
       JCEN=Ø
C
C      DR IS THE UNIFORM RADIAL MESH SPACING
C
       DR=Ø.
C
C      DZ IS THE UNIFORM AXIAL MESH SPACING
C
       DZ=Ø.
C
C      CYL IS THE GEOMETRY INDICATOR. CYL=1. FOR CYLINDRICAL GEOMETRY
C                                         =Ø. FOR PLANE (SLAB) GEOMETRY
C
       CYL=1.
C
C      GRDVEL IS THE REZONE INDICATOR. GRDVEL=Ø. FOR EULERIAN
C                                             =1. FOR LAGRANGIAN
C                                             =2. FOR ALE
C
       GRDVEL=2.
C
C      ANC IS THE NODE COUPLER CONSTANT
C
       ANC=.Ø5
C
C      AØFAC IS A FACTOR USED IN CALCULATING THE COURANT TIMESTEP
C
       AØFAC=.2
C
C      AØ,AØM,AND BØ ARE REZONE COEFFICIENTS WHICH DETERMINE THE TYPE OF
C          DIFFERENCING USED IN CALCULATING THE FLUXING TERMS. AØ APPLIES
C          TO THE MOMENTUM EQN,AØM TO THE MASS AND ENERGY EQNS,AND
C          BØ TO ALL THREE. SOME EXAMPLES OF THEIR SIGNIFICANCE ARE
C               AØ=Ø.,BØ=Ø.,   CENTERED DIFFERENCING
C               AØ=1.,BØ=Ø.,   FULL DONOR CELL
C               AØ=Ø.,BØ=2.,   INTERPOLATED DONOR CELL.
C          NOTE THAT THE EQNS ARE UNSTABLE IN THE FIRST CASE.
C
       AØ=.1
       AØM=1.
       BØ=Ø.
C
C      KXI IS A PARAMETER GOVERNING THE TREATMENT OF VISCOSITY.
C               KXI=1 ALLOWS MU AND LAM TO BE TREATED AS NUMERICAL
C                         VISCOSITY COEFFICIENTS
C               KXI=Ø ALLOWS MU AND LAM TO BE TREATED AS TRUE PHYSICAL
C                         VISCOSITY COEFFICIENTS
C               KXI=-1 CAUSES TO CODE TO CALCULATE ITS OWN NUMERICAL
C                         VISCOSITY COEFFICIENTS BASED ONLY UPON THE
C                         RESTRICTION THAT THE RATIO OF THE MU AND LAM
C                         COMPUTED INTERNALLY BY THE CODE IS EQUAL TO
C                         THE RATIO OF THE MU AND LAM READ IN
C
```

```
      KXI=-1
C
C     GZP IS THE BODY FORCE FELT BY THE PARTICLES, IT IS NOT NECESSARILY
C         EQUAL TO GZ,
C
      GZP=-980.
C
C     ZORIG IS THE NO, OF FIREBALL RADII TO HOLD THE OUTER BOUNDARY
C         AWAY FROM THE FIREBALL
C
      ZORIG=6.
C
C     T IS THE PROBLEM START TIME
C
      T=0.
C
C     DT IS THE INITIAL TIMESTEP
C
      DT=.001
C
C     NCLST IS THE CYCLE NO, AT WHICH TO HALT THE RUN
C
      NCLST=99999
C
C     TWFIN IS THE REAL TIME AT WHICH TO TERMINATE THE RUN
C
      TWFIN=1.E30
C
C     DTO AND DTOC ARE OUTPUT FREQUENCY CONTROL ARRAYS, IF T.LE.DTOC(1),
C         EDITS AND PLOTS OCCUR EVERY DTO(1) PROBLEM SECONDS, IF
C         DTOC(I-1).LT.T.LE.DTOC(I),THEN OUTPUT OCCURS EVERY DTO(I)
C         PROBLEM SECONDS, A MAXIMUM OF 10 DTO,DTOC PAIRS MAY BE
C         SPECIFIED,
C
      DTO(1)=1.
      DTOC(1)=1.E30
      DO 10 II=2,10
      DTO(II)=0.
   10 DTOC(II)=0.
C
C     NCQ IS THE CYCLE ON WHICH TURBULENCE IS TO BE SEEDED, IF NCQ IS
C         LESS THAN 0,THERE IS NO TURBULENCE,
C
      NCQ=-1
C
C     IF TURB IS .FALSE.,THE TURBULENCE IS CURRENTLY OFF
C
      TURB=.FALSE.
C
C     QLEVEL,TUQI,TUSI,AND TQ ARE ALL TURBULENCE QUANTITIES
C
      QLEVEL=.02
      TUQI=0.
      TUSI=0.
      TQ=0.
C
C     A PARTICLE CAN BE MOVED NO MORE THAN
```

```
C          SQRT(2.)*WMAXEF*SQRT(2.*SIGMA*DT) DUE TO THE TURBULENT
C          DIFFUSION EFFECT DURING ANY ONE CYCLE
C
      WMAXEF=2.
C
C     ANY PARTICLE WITH XJ.LE.RMINEF WILL NOT BE SUBJECT
C          TO TURBULENT DIFFUSION
C
      RMINEF=50.
C
C     TSTRTD IS THE TIME AT WHICH TO START TURBULENT DIFFUSION
C
      TSTRTD=1.
C
C     IST IS THE NUMBER OF PARTICLES TO FOLLOW AS A FUNCTION OF TIME.
C          IF IST.LT.0,NO PARTICLES WILL BE FOLLOWED.
C
      IST=-1
C
C     JDUMP IS THE FREQUENCY OF DUMP CYCLES
C
      JDUMP=999999
      RETURN
      END
```

```
      SUBROUTINE DMP(FWA,LWA,IFILE)
C
C     ROUTINE TO DUMP SCM FROM FWA TO LWA WITH THE OUTPUT DIRECTED
C          TO LOGICAL UNIT IFILE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      INTEGER FWA,GETIT
      DIMENSION IDUMP(4)
C
C     CHECK FOR ERRORS
C
      IF(FWA.LE.LWA) GO TO 10
C
C     YES, PRINT MESSAGE AND QUIT.
C
      WRITE(IFILE,100) FWA,LWA
      RETURN
   10 CONTINUE
C
C     ALL O.K.  PRINT DUMP HEADER.
C
      WRITE(IFILE,110) FWA,LWA
C
C     IFW IS THE ADDRESS OF THE FIRST WORD TO BE PRINTED ON THE
C          CURRENT LINE
```

```
C      ILW IS THE ADDRESS OF THE WORD CURRENTLY BEING PROCESSED
C
       IFW=FWA
       ILW=FWA
C
C      ICLW IS THE CONTENTS OF THE WORD WHOSE ADDRESS IS ILW
C
       ICLW=GETIT(ILW)
C
C      IDUMP CONTAINS THE WORDS TO BE PRINTED ON THE CURRENT LINE
C          (A MAXIMUM OF FOUR).
C      ISUB IS THE NO. OF THE LAST LOCATION OF IDUMP THAT WAS FILLED.
C
       ISUB=1
       IDUMP(1)=ICLW
    20 CONTINUE
C
C      EXAMINE THE NEXT WORD
C
       ILW=ILW+1
C
C      SEE IF WE ARE DONE
C
       IF(ILW.LE.LWA) GO TO 30
C
C      YES. FLUSH IDUMP IF NECESSARY AND QUIT.
C
       IF(ISUB.LE.0) RETURN
       WRITE(IFILE,120) IFW,(IDUMP(I),I=1,ISUB)
       RETURN
    30 CONTINUE
C
C      NO. SAVE THE CONTENTS OF THE NEXT-TO-THE-LAST WORD AND
C          GO GET THE CONTENTS OF THE CURRENT WORD.
C
       ICNLW=ICLW
       ICLW=GETIT(ILW)
C
C      SEE IF THE LAST WORD AND THE NEXT-TO-THE-LAST WORD ARE THE SAME
C
       IF(ICLW.NE.ICNLW) GO TO 80
C
C      YES. WE WILL GO INTO REPETITION MODE. FIRST FLUSH IDUMP
C          IF NECESSARY.
C
       ISUB=ISUB-1
       IF(ISUB.LE.0) GO TO 40
       WRITE(IFILE,120) IFW,(IDUMP(I),I=1,ISUB)
       ISUB=0
    40 CONTINUE
C
C      IFW IS NOW THE ADDRESS OF THE FIRST WORD OF THE REPETITION GROUP
C
       IFW=ILW-1
C
C      SCAN FORWARD UNTIL THE END OF THE REPETITION GROUP IS LOCATED
C
```

```
      IFWP=IFW+2
      DO 50 I=IFWP,LWA
      ICLW=GETIT(I)
      IF(ICLW.NE.ICNLW) GO TO 60
   50 CONTINUE
C
C     ALL THE REST OF THE REQUESTED DUMP REGION LIES WITHIN THE
C         REPETITION GROUP
C
      ILW=LWA
      GO TO 70
   60 CONTINUE
C
C     ILW IS THE ADDRESS OF THE LAST WORD IN THE REPETITION GROUP
C
      ILW=I-1
   70 CONTINUE
C
C     WRITE OUT THE REPETITION GROUP
C
      WRITE(IFILE,130) IFW,ILW,ICNLW
C
C     GO BACK AND CONTINUE PROCESSING UNLESS WE ARE DONE
C
      ILW=ILW+1
      IF(ILW.GT.LWA) RETURN
      IFW=ILW
      ISUB=1
      IDUMP(1)=ICLW
      GO TO 20
   80 CONTINUE
C
C     LAST AND NEXT-TO-THE-LAST WORD ARE DIFFERENT. SEE IF THE IDUMP
C         BUFFER IS FULL.
C
      IF(ISUB.LT.4) GO TO 90
C
C     YES. FLUSH IT.
C
      WRITE(IFILE,120) IFW,(IDUMP(I),I=1,4)
      IFW=ILW
      ISUB=0
   90 CONTINUE
C
C     ADD THE CURRENT WORD TO THE IDUMP BUFFER AND CONTINUE
C
      ISUB=ISUB+1
      IDUMP(ISUB)=ICLW
      GO TO 20
C
  100 FORMAT(1H ,50HERROR IN DMP ARGUMENTS. DMP BYPASSED. FWA AND LWA ,
     1 4HARE ,2I10)
  110 FORMAT(1H1,17HDUMP OF SCM FROM ,O6,4H TO ,O6///)
  120 FORMAT(1H ,O6,5X4O25)
  130 FORMAT(1H ,O6,6H THRU ,O6,13H ALL CONTAIN ,O20)
      END
```

```
      SUBROUTINE DMPPK(N,PACK)
C
C     ROUTINE TO DUMP EXCHANGE PACKAGE
C
C     WRITTEN BY LARRY RUDSINSKI AND JERRY MELENDEZ,LASL C-4
C     MODIFIED TO CORRECTLY PICK UP C(A0) - C(A7) BY J.L.NORTON,LASL TD3
C
      DIMENSION PACK(1),PARCEL(4)
      DIMENSION ISAVE(8)
      DIMENSION NPC(8),NAC(8),NBC(8),NXC(8)
      DIMENSION IAA(9),IA1(8),IA3(2)
      INTEGER PACK,PARCEL
      INTEGER PACK1
      INTEGER GETIT,SHIFT,AND
      DATA NPC/1HP,3HRAS,3HFLS,3HPSD,3HRAL,3HFLL,3HNEA,3HEEA/
      DATA NAC/2HA0,2HA1,2HA2,2HA3,2HA4,2HA5,2HA6,2HA7/
      DATA NBC/3HBPA,2HB1,2HB2,2HB3,2HB4,2HB5,2HB6,2HB7/
      DATA NXC/2HX0,2HX1,2HX2,2HX3,2HX4,2HX5,2HX6,2HX7/
      DATA IA1/5HC(A0),5HC(A1),5HC(A2),5HC(A3),5HC(A4),5HC(A5),5HC(A6),
     1 5HC(A7)/
      DATA IA3/7H OUT OF,6H RANGE/
      IF(N.EQ.0) N=6
C
C     PRINT CAUSE OF ABORT
C
      IARG=SHIFT(PACK(4),-36)
      CALL MODE(IARG,N)
C
C     GET THE FIELD LENGTH
C
      IAA(9)=SHIFT(PACK(3),-36)
C
C     IF FIELD LENGTH IS GARBAGED,SET TO MAXIMUM
C
      IF(IAA(9).GT.150077B) IAA(9)=150077B
C
C     PICK UP AND PRINT REGISTERS,ETC.
C
      DO 30 I=1,8
C
C     PICK UP B REGISTER I-1
C
      NB=AND(PACK(I),777777B)
C
C     PICK UP A REGISTER I-1
C
      NA=AND(SHIFT(PACK(I),-18),777777B)
C
C     GET C(AN)
C
      IF(NA.GE.IAA(9)) GO TO 10
      ISAVE(I)=GETIT(NA)
   10 CONTINUE
C
```

```
C      PICK UP DATA IN THE TOP OF EXCHANGE PACKAGE WORDS
C
       NP=SHIFT(PACK(I),-36)
C
C      PICK UP X REGISTER I=1 AND PREPARE TO PRINT IT IN 4 PIECES
C
       K=4
       DO 20 J=1,4
       PARCEL(K)=AND(SHIFT(PACK(I+8),-(J-1)*15),77777B)
       K=K-1
    20 CONTINUE
       IAA(I)=NA
       WRITE(N,70) NPC(I),NP,NAC(I),NA,NBC(I),NB,NXC(I),PARCEL(1),PARCEL(
      1 2),PARCEL(3),PARCEL(4),PACK(I+8),PACK(I+8)
    30 CONTINUE
C
C      PRINT C(AN)
C
       DO 60 I=1,8
C
C      CHECK FOR A REGISTER OUT OF RANGE
C
       IF(IAA(I).GE.IAA(9)) GO TO 50
       PACK1=ISAVE(I)
       K=4
       DO 40 J=1,4
       PARCEL(K)=AND(SHIFT(PACK1,-(J-1)*15),77777B)
       K=K-1
    40 CONTINUE
       WRITE(N,80) IA1(I),PARCEL(1),PARCEL(2),PARCEL(3),PARCEL(4),PACK1,
      1 PACK1
       GO TO 60
C
C      A REGISTER IS OUT OF RANGE
C
    50 WRITE(N,90) NAC(I),IA3(1),IA3(2)
    60 CONTINUE
       WRITE(N,100)
       RETURN
C
    70 FORMAT(1H ,A4,2X,O8,2X,A2,2X,O6,2X,A3,2X,O6,7X,A2,2X,O5,2X,O5,2X,
      1 O5,2X,O5,E25.13,8X,A10)
    80 FORMAT(45X,A5,3(1X,O5,1X),1X,O5,E25.13,8X,A10)
    90 FORMAT(47X,A2,1X,A7,A6)
   100 FORMAT(1H ,//)
       END
```

****

```
       SUBROUTINE DVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
C
C      ROUTINE TO CALCULATE MAXIMUM VELOCITY AND MAXIMUM AND MINIMUM
C          ZONE SIZES
```

```
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM       -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM       -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC       -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC       -----
*     ----- BEGIN COMDECK YAQEQ       -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
```

```
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ        -----
*     ----- END COMDECK YSTORE       -----
C     DR(Z)MIN(MAX) IS THE MINIMUM (MAXIMUM) OF THE MAGNITUDES OF
C         CELL SIDES IN THE R (Z) COORDINATE DIRECTION
C
      DRMIN=1.E+20
      DZMIN=1.E+20
      DRMAX=0.
      DZMAX=0.
C
C     VMAX IS THE MAXIMUM GRID VELOCITY MAGNITUDE ALONG EITHER
C         COORDINATE AXIS
C
      VMAX=0.
      IVM=0
      JVM=0
      CALL START
C
C     LOOP OVER ALL REAL ZONES
C
      DO 30 J=2,JP1
      DO 20 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      VMAXP=AMAX1(VMAX,ABS(U(IJ)),ABS(V(IJ)))
      IF(VMAXP.EQ.VMAX) GO TO 10
      IVM=I
      JVM=J
      VMAX=VMAXP
   10 CONTINUE
C
C     DETERMINE THE FOUR VERTICES OF CELL (I,J)
C
C     (X1,Y1) IS VERTEX (I+1,J)     (VERTEX 1)
C     (X2,Y2) IS VERTEX (I+1,J+1)   (VERTEX 2)
C     (X3,Y3) IS VERTEX (I,J+1)     (VERTEX 3)
C     (X4,Y4) IS VERTEX (I,J)       (VERTEX 4)
C
      X1=X(IPJ)
      X2=X(IPJP)
      X3=X(IJP)
      X4=X(IJ)
      Y1=Y(IPJ)
      Y2=Y(IPJP)
      Y3=Y(IJP)
      Y4=Y(IJ)
C
C     DETERMINE THE SQUARE OF THE LENGTH OF EACH SIDE
C
C     X(Y)NM IS THE SQUARE OF THE LENGTH OF THE SIDE BOUNDED BY
C         VERTICES N AND M
C
      X14=(X1-X4)**2+(Y1-Y4)**2
      X23=(X2-X3)**2+(Y2-Y3)**2
      Y21=(X2-X1)**2+(Y2-Y1)**2
      Y34=(X3-X4)**2+(Y3-Y4)**2
```

```
      DRMIN=AMIN1(DRMIN,X14,X23)
      DRMAX=AMAX1(DRMAX,X14,X23)
      DZMIN=AMIN1(DZMIN,Y21,Y34)
      DZMAX=AMAX1(DZMAX,Y21,Y34)
      IJ=IPJ
   20 IJP=IPJP
      CALL LOOP
   30 CONTINUE
      DRMIN=SQRT(DRMIN)
      DRMAX=SQRT(DRMAX)
      DZMIN=SQRT(DZMIN)
      DZMAX=SQRT(DZMAX)
      RETURN
      END
```

```
      FUNCTION ERF(Z)
C
C     ROUTINE TO CALCULATE THE STANDARD ERROR FUNCTION
C
C     STANDARD LIBRARY SUBPROGRAM - LOS ALAMOS SCIENTIFIC LABORATORY
C     DOCUMENTED BY J.L.NORTON,LASL TD-3,1972
C
      DIMENSION P(7,2),Q(6,2)
      DIMENSION A(14),B(12)
      EQUIVALENCE(A,P),(B,Q)
      X=ABS(Z)
C
C     IF ARGUMENT IS ZERO,ERF IS ZERO
C
      ERF=0.
      IF(X.EQ.0.) RETURN
      DATA (A(I),I=1,14)/1.1283791670955,.34197505591854,
     1 .86290601455206E-1,.12382023274723E-1,.11986242418302E-2,
     2 .76537302607825E-4,.25365482058342E-5,-.99999707603738,
     3 -1.4731794832805,-1.0573449601594,-.44078839213875,
     4 -.10684197950781,-.12636031836273E-1,-.11493933666616E-8/
      DATA (B(I),I=1,12)/-.36359916427762,.52205830591727E-1,
     1 -.30613035688519E-2,-.46856639020338E-4,.15601995561434E-4,
     2 -.62143556409287E-6,2.6015349994799,2.9929556755308,
     3 1.9684584582884,.79250795276064,.18937020051337,
     4 .22396882835053E-1/
C
C     IF ARGUMENT IS GREATER THAN 5.5, ERF IS UNITY
C
   10 ERF=SIGN(1.,Z)
C
C     USE RATIONAL APPROXIMATION TO COMPUTE ERF
C
      IF(X.GE.5.5) RETURN
      J=1
      IF(X.GT.1.5) J=2
```

```
      FJ=J-1
      D=X**(2-J)
      U=D*X
C
C     TWO SEPARATE RATIONAL APPROXIMATIONS ARE USED. BOTH HAVE BEEN
C         DERIVED USING A PROGRAM BASED ON MAEHLY=S SECOND DIRECT
C         METHOD DESCRIBED IN JOURNAL OF THE ACM,VOL.10,NO.3
C
      ERF=(D*EXP(-X*X)*(P(1,J)+U*(P(2,J)+U*(P(3,J)+U*(P(4,J)+U*(P(5,J)+U
     1 *(P(6,J)+U*P(7,J))))))))/(1.+U*(Q(1,J)+U*(Q(2,J)+U*(Q(3,J)+U*(Q(4,
     2 J)+U*(Q(5,J)+U*Q(6,J)))))))+FJ)*SIGN(1.,Z)
      RETURN
      END
```

```
      SUBROUTINE FILMCO
C
C     ROUTINE TO UPDATE GRID LIMITS AND PARTICLE QUANTITIES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM     -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM     -----
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
```

```
          2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
          3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
          4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
          5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
          6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
            COMMON/YSC2/ZZ
C           COMMON/YSC4/ITAB(ITABP)
            COMMON/YSC4/ITAB(1000)
            COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*           ***** END COMDECK YAQSC       *****
*           ***** BEGIN COMDECK YAQEQ       *****
            EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
          1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
          1 (8),E,ETIL,AREA,XR13K),
          2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
          3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
          4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
          5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
          6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
          7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
          8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
          9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
          1 AVYSV,X24K)
            REAL M,MP,MPAR,MTIL
*           ***** END COMDECK YAQEQ       *****
*           ***** END COMDECK YSTORE      *****
C           FIND THE GRID LIMITS. MAX(X)=XR,MAX(Y)=YT,MIN(X)=XL=0.,MIN(Y)=YB.
C
            XL=0.0
            YB=1.E+20
            XR=YT=-YB
            CALL START
            DO 20 J=2,JP2
            DO 10 I=1,IP1
            XR=AMAX1(XR,X(IJ))
            YB=AMIN1(YB,Y(IJ))
            YT=AMAX1(YT,Y(IJ))
         10 IJ=IJ+NQ
            CALL LOOP
         20 CONTINUE
C
C           VV IS USED IN SCALING VELOCITY VECTOR PLOTS.
C               VV = .9*MAX(X)/IBAR
C
            VV=0.9*XR*RIBAR
C
C           FIYB IS THE LOCATION OF MIN(Y) IN RASTER COUNTS
C
            FIYB=916.0
C
C           XD IS THE RATIO OF GRID WIDTH TO HEIGHT
C
            XD=XR/(YT-YB)
C
C           ONE WISHES TO MAKE THE PLOTS FILL THE FILM FRAME AS NEARLY AS
C               POSSIBLE. AT MOST THE PLOT CAN BE 1022 RASTER POINTS WIDE AND
C               900 RASTER POINTS TALL (LEAVING ROOM FOR LABELS AT THE BOTTOM
```

```
C           AND A 16 POINT MARGIN AT THE TOP.) AN IDEAL GRID, THEN,WOULD
C           HAVE XD=1022/900=1.13556. IF XD.GT.1.13556,THE GRID IS DEFINED
C           AS WIDER THAN HIGH. IF XD.LE.1.13556,THE GRID IS DEFINED AS
C           HIGHER THAN WIDE. IN THE FORMER CASE,THE X COORDINATE RASTER
C           BOUNDS ARE SET TO (0,1022) AND THE Y COORDINATE RASTER BOUNDS
C           TO (FIYT,916) WHERE FIYT IS DETERMINED SUCH THAT THE X AND Y
C           SCALES ARE THE SAME. IN THE LATTER CASE,THE Y COORDINATE
C           RASTER BOUNDS ARE SET TO (16,916) AND THE X COORDINATE RASTER
C           BOUNDS ARE DETERMINED SUCH THAT THE X AND Y SCALES ARE EQUAL
C           AND THE LEFT AND RIGHT MARGINS ARE THE SAME.
C
      IF(XD.LE.1.13556) GO TO 30
C
C     GRID WIDER THAN HIGH. THE X RASTER BOUNDS ARE (FIXL,FIXR). THE
C         Y RASTER BOUNDS ARE (FIYT,FIYB).
C
      FIXL=0.
      FIXR=1022.
      FIYT=916.-1022./XD
      GO TO 40
C
C     GRID HIGHER THAN WIDE
C
   30 CONTINUE
      FIXL=AMAX1(0.,511.-450.*XD)
      FIXR=511.+450.*XD
      FIYT=16.
   40 CONTINUE
C
C     XCONV AND YCONV ARE FACTORS TO CONVERT FROM X AND Y CARTESIAN
C         COORDINATES TO RASTER COORDINATES
C
      XCONV=(FIXR-FIXL)/(XR-XL)
      YCONV=(FIYT-FIYB)/(YT-YB)
C
C     PROVIDE FIXED POINT VALUES OF THE RASTER BOUNDS
C
      IXL=FIXL
      IXR=FIXR
      IYB=FIYB
      IYT=FIYT
C
C     IF THERE ARE NO PARTICLES,WE ARE DONE
C
      IF(NPT.EQ.0) RETURN
C
C     THERE ARE PARTICLES. CALCULATE THE CARTESIAN PARTICLE BOUNDS
C          FROM THE GRID BOUNDS.
C
      PXL=0.0
      PYB=YB
      PXR=XR
      PYT=YT
      FIPYB=916.0
      FIPXL=FIXL
      FIPXR=FIXR
      FIPYT=FIYT
```

```
C
C     CONVERSION FACTORS FROM PARTICLE COORDINATES TO RASTER COORDINATES
C
      PXCONV=(FIPXR-FIPXL)/(PXR-PXL)
      PYCONV=(FIPYT-FIPYB)/(PYT-PYB)
C
C     PROVIDE FIXED POINT RASTER BOUNDS
C
      IPXL=FIPXL
      IPXR=FIPXR
      IPYB=FIPYB
      IPYT=FIPYT
      RETURN
      END
```

```
      SUBROUTINE GETOMG
C
C     ROUTINE TO CALCULATE VORTICITY
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
```

```
     1 ILNG,NILNG,TP3,TUPOT,TDOSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC     -----
*     ----- BEGIN COMDECK YAQEQ     -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ     -----
*     ----- END COMDECK YSTORE     -----
   10 CALL START
      WMAX=EM10
      QMN=1.E30
      QMX=-QMN
      DO 30 J=2,JP1
      DO 20 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      IF(J.EQ.2) CQ(IJM)=0.
      IF(J.EQ.JP1) CQ(IJP)=0.
      IF(I.EQ.IBAR) CQ(IPJ)=0.
      X1=X(IPJ)
      Y1=Y(IPJ)
      U1=U(IPJ)
      V1=V(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      U2=U(IPJP)
      V2=V(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      U3=U(IJP)
      V3=V(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      U4=U(IJ)
      V4=V(IJ)
      R1=.125*RVOL(IJ)*(R(IPJ)+R(IPJP)+R(IJP)+R(IJ))
      CQ(IJ)=R1*((U1+U4)*(X1-X4)+(V1+V4)*(Y1-Y4)+(U2+U1)*(X2-X1)+(V2+V1)
     1 *(Y2-Y1)+(U3+U2)*(X3-X2)+(V3+V2)*(Y3-Y2)+(U4+U3)*(X4-X3)+(V4+V3)*
```

```
     2 (Y4-Y3))
      QMN=AMIN1(CQ(IJ),QMN)
      QMX=AMAX1(CQ(IJ),QMX)
      WSAV=WMAX
      WMAX=AMAX1(WMAX,ABS(CQ(IJ)))
      IF(WSAV.NE.WMAX) ISVW=I
      IF(WSAV.NE.WMAX) JSVW=J
      IJM=IJM+NQ
      IJ=IPJ
   20 IJP=IPJP
      CALL LOOP
   30 CONTINUE
      CALL DONE
      RETURN
      END
```

```
      SUBROUTINE MODE(IARG,NARG)
C
C     ROUTINE TO ANALYZE THE PROGRAM STATUS DESIGNATION
C
C     WRITTEN BY J.L.NORTON,LASL TD-3,1973
C
      DIMENSION ITROUB(12),ICAUSE(2,12)
      LOGICAL ITROUB
      INTEGER AND,COMP
      DATA ICAUSE/9HUNDERFLOW,1H ,8HOVERFLOW,1H ,10HINDEFINITE,1H ,
     1 4HSTEP,1H ,10HBREAKPOINT,1H ,10HPROGRAM RA,3HNGE,10HSCM DIRECT,
     2 6H RANGE,10HLCM DIRECT,6H RANGE,10HSCM BLOCK ,5HRANGE,
     3 10HLCM BLOCK ,5HRANGE,10HSCM PARITY,1H ,10HLCM PARITY,1H /
      ITEST=AND(IARG,COMP(777777B))
      IF(ITEST.NE.0) RETURN
      ITEST=1
      DO 10 IJ=1,12
      ITROUB(IJ)=.FALSE.
      IF(AND(IARG,ITEST).NE.0) ITROUB(IJ)=.TRUE.
      ITEST=2*ITEST
   10 CONTINUE
      DO 20 IJ=1,12
      IF(.NOT.ITROUB(IJ)) GO TO 20
      WRITE(NARG,30) ICAUSE(1,IJ),ICAUSE(2,IJ)
   20 CONTINUE
      RETURN
C
   30 FORMAT(1H 51HTHE FOLLOWING CONDITION EXISTED AT THE TIME OF THE ,
     1 7HABORT--,2A10)
      END
```

```
      SUBROUTINE MSHMKR
C
C     ROUTINE TO GENERATE THE INITIAL PROBLEM MESH AND SET THE
C         INITIAL QUANTITIES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK LCMATM      -----
C     COMMON/YSC3/UFIRE(IPFB),EFIRE(IPFB),RHOFIR(IPFB),XFIRE(IPFB)
      COMMON/YSC3/UFIRE(200),EFIRE(200),RHOFIR(200),XFIRE(200)
*     ----- END COMDECK LCMATM      -----
      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,THASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
```

```
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ===== END COMDECK YAQEQ      =====
*     ===== END COMDECK YSTORE     =====
      DIMENSION MESH(3,4)
      LOGICAL FBFILE
      DATA (MESH(II),II=1,12)/12*0/
      DATA IEFLAG/0/
      DATA PJLN/0./
C
C     SET UP THE NAMELIST INPUT TABLE
C
      ASSIGN 530 TO IERRT
      CALL TABDEF(MESH,4HFIRE,4,IERRT)
      CALL TABSET(MESH,4HNRAD,NRAD,IEFLAG,0,0,0,0)
      CALL TABSET(MESH,3HNTH,NTH,IEFLAG,0,0,0,0)
      CALL TABSET(MESH,6HFBFILE,FBFILE,IEFLAG,0,0,0,0)
C
C     INITIALIZE. NQIM IS THE NO. OF WORDS OF MEMORY NEEDED TO STORE
C         ONE ROW OF DATA (LESS 1).
C
      NQIM=NQI-1
      IF(.NOT.TURB) GO TO 10
      TNEG=0.
      TNEGSV=0.
   10 CONTINUE
C
C     DEFINE THE CELL VERTICES AS THOUGH THE ENTIRE MESH WERE UNIFORM
C
      XX=0.0
      YY=YB
      CALL START
      DO 50 J=2,JP2
      DO 40 I=1,IP1
      X(IJ)=XX
      Y(IJ)=YY
      R(IJ)=XX*CYL+OMCYL
      U(IJ)=0.
      V(IJ)=0.
      IF(J.NE.2) GO TO 20
      Y(IJM)=YY-DZ
      X(IJM)=XX
      R(IJM)=R(IJ)
      U(IJM)=0.
      V(IJM)=0.
   20 IF(J.NE.JP2) GO TO 30
      Y(IJP)=YY+DZ
      X(IJP)=XX
```

```
      R(IJP)=R(IJ)
      U(IJP)=0.
      V(IJP)=0.
   30 IJP=IJP+NQ
      IJM=IJM+NQ
      XX=XX+DR
   40 IJ=IJ+NQ
      XX=0.
      YY=YY+DZ
      CALL LOOP
   50 CONTINUE
      CALL DONE
C
C     SEE IF THE ENTIRE MESH ACTUALLY IS UNIFORM
C
      IF(FREZXR.EQ.1..AND.FREZYT.EQ.1..AND.FREZYB.EQ.1.) GO TO 130
C
C     NO. DO THE NON-UNIFORM GENERATION.
C
C     CONVERT JCEN TO AN ACTUAL VERTEX NO. JCEN IS THE J-LINE THAT
C         GOES THROUGH THE CENTER OF THE UNIFORM REGION.
C
      JCEN=JCEN+2
C
C     JTOP AND JBOT ARE THE J-LINES AT THE TOP AND BOTTOM OF THE
C         UNIFORM REGION
C
      JTOP=JCEN+JUNFO2
      JBOT=JCEN-JUNFO2
C
C     TJ IS THE DISTANCE FROM THE CENTER TO THE TOP (OR BOTTOM) OF
C         THE UNIFORM REGION
C
      TJ=FLOAT(JUNFO2)*DZ
C
C     LOOP TO SET THE NON-UNIFORM VERTICES
C
      CALL START
      DO 110 J=2,JP2
      DO 100 I=1,IP1
      IMJ=IJ-NQ
      IF(FREZXR.EQ.1.) GO TO 60
C
C     SEE IF WE ARE WITHIN THE UNIFORM X REGION
C
      IF(I.LE.IUNF+1) GO TO 60
C
C     NO. GENERATE THE NON-UNIFORM X USING GEOMETRIC PROGRESSION.
C
      X(IJ)=X(IMJ)+FREZXR*(X(IMJ)-X(IMJ-NQ))
      R(IJ)=X(IJ)*CYL+OMCYL
   60 CONTINUE
C
C     JDT IS THE NO. OF J LINES THAT THE CURRENT J IS ABOVE THE TOP
C         OF THE UNIFORM REGION
C
      JDT=J-JTOP
```

```
C
C     JDB IS THE NO. OF J LINES THAT THE CURRENT J IS BELOW THE BOTTOM
C         OF THE UNIFORM REGION
C
      JDB=JBOT-J
C
C     SEE IF THE CURRENT J IS BELOW THE UNIFORM REGION
C
      IF(JDB.GT.0) GO TO 70
C
C     NO. SEE IF IT IS ABOVE THE UNIFORM REGION.
C
      IF(JDT.GT.0) GO TO 80
C
C     NO. J IS IN UNIFORM REGION. SET IT AS SUCH.
C
      Y(IJ)=REZY0+FLOAT(J-JCEN)*DZ
      GO TO 90
   70 CONTINUE
C
C     J IS BELOW THE UNIFORM REGION. CALCULATE ITS POSITION USING
C         FORMULA FOR SUM OF GEOMETRIC PROGRESSION. THE FORMULA IS
C              SUM=A*(1-F**N)/(1-F)
C         WHERE A IS THE FIRST TERM,N IS THE NO. OF TERMS,AND F IS
C         THE RATIO OF THE MTH TERM TO THE (M-1)TH TERM. IN THIS CASE
C              A=DZ*FREZ, N=JDB, AND F=FREZ. THE POSITION OF JBOT IS
C         AT Y=REZY0-TJ. J IS SUM BELOW THIS OR
C              Y(J)=REZY0-TJ-DZ*FREZ*(1-FREZ**JDB)/(1-FREZ).
C
      IF(FREZYB.GT.1.) Y(IJ)=REZY0-TJ-DZ*FREZYB*(1.-FREZYB**JDB)*ROMFYB
      GO TO 90
   80 CONTINUE
C
C     J IS ABOVE THE UNIFORM REGION. USE GEOMETRIC PROGRESSION SUM TO
C         CALCULATE ITS VALUE ALSO.
C
      IF(FREZYT.GT.1.) Y(IJ)=REZY0+TJ+DZ*FREZYT*(1.-FREZYT**JDT)*ROMFYT
   90 CONTINUE
C
C     SEE IF J IS 2
C
      IF(J.NE.2) GO TO 100
C
C     YES. SET YB (BOTTOM Y).
C
      YB=Y(IJ)
C
C     MAKE THE FICTITIOUS CELLS AT THE BOTTOM AS THOUGH THERE WERE
C         ANOTHER REAL ROW OF CELLS
C
      X(IJM)=X(IJ)
      IJM=IJM+NQ
      IJP=IJP+NQ
  100 IJ=IJ+NQ
      CALL LOOP
  110 CONTINUE
      CALL DONE
```

```
C
C      CALCULATE Y FOR J=1 FICTITIOUS CELLS
C
       CALL START
       DO 120 I=1,IP1
       Y(IJM)=Y(IJ)-(Y(IJP)-Y(IJ))*FREZYB
       IJ=IJ+NQ
       IJM=IJM+NQ
       IJP=IJP+NQ
   120 CONTINUE
       CALL DONE
C
C      MESH IS GENERATED
C
   130 CONTINUE
C
C      *******************************************************************
C      GENERATE AN INITIAL FIREBALL AND ITS AMBIENT ATMOSPHERE
C      *******************************************************************
C
C      FIRST SET UP THE AMBIENT ATMOSPHERE
C
C      GET GAMMA-1 AT REZY0
C
       ETMP=REZSIE
       ROTMP=REZRON
       CONCJ=0.
       CALL AIR
C
C      XX IS THE ISOTHERMAL CONSTANT IN THE PRESSURE
C
       XX=GMONE*REZSIE
C
C      NOTE THAT A NEGATIVE GRAVITATIONAL FORCE IS ASSUMED HERE
C
       YY=.5*ABS(GZ)
C
C      BRING IN THE FIRST THREE ROWS
C
       CALL START
C
C      PROCESS J=1 AND 2 ROWS. YJC2 IS THE Y COORDINATE OF CENTERS OF
C          CELLS IN THE ROW J=2
C
       YJC2=.5*(Y(IJP)+Y(IJ))
C
C      ROSAV IS THE DENSITY OF THE CELLS IN ROW J=2 ASSUMING AN
C          ISOTHERMAL,IDEAL GAS ATMOSPHERE
C
       ROSAV=REZRON*EXP(-GZ*(REZY0-YJC2)/XX)
C
C      CALCULATE THE DENSITY OF CELLS IN ROW J=1 USING THE DIFFERENCE
C          FORM FOR HYDROSTATIC EQUILIBRIUM
C
       FNUM=(Y(IJP)-Y(IJ))*YY
       FDEN=FNUM*FREZYB
       ROJ1=ROSAV*(XX+FNUM)/(XX-FDEN)
```

```
C
C     SET AMBIENT QUANTITIES IN ROWS J=1 AND 2
C
      DO 140 I=1,IP1
      RO(IJ)=ROSAV
      RO(IJM)=ROJ1
      E(IJ)=REZSIE
      SIE(IJ)=REZSIE
      E(IJM)=REZSIE
      SIE(IJM)=REZSIE
      IJ=IJ+NQ
  140 IJM=IJM+NQ
C
C     BRING IN THE NEXT ROW
C
      CALL LOOP
C
C     LOOP OVER ALL THE OTHER REAL ROWS. BOOTSTRAP DENSITIES UPWARD
C         FROM ROW J=2.
C
      DO 160 J=3,JP1
      FDEN=(Y(IJP)-Y(IJ))*YY
      FNUM=(Y(IJ)-Y(IJM))*YY
      ROSAV=ROSAV*(XX-FNUM)/(XX+FDEN)
      DO 150 I=1,IP1
      RO(IJ)=ROSAV
      E(IJ)=REZSIE
      SIE(IJ)=REZSIE
  150 IJ=IJ+NQ
      CALL LOOP
  160 CONTINUE
C
C     SET THE TOP FICTITIOUS ROW
C
      FNUM=FNUM*FREZYT
      FDEN=FDEN*FREZYT
      ROJP2=ROSAV*(XX-FNUM)/(XX+FDEN)
      DO 170 I=1,IP1
      RO(IJ)=ROJP2
      E(IJ)=REZSIE
      SIE(IJ)=REZSIE
  170 IJ=IJ+NQ
      CALL DONE
C
C     THE AMBIENT ATMOSPHERE IS NOW SET. READ IN THE DATA TO GENERATE
C         THE FIREBALL. ISUB IS THE DATA POINT SUBSCRIPT. INTERPOLATION
C         TABLES ARE READ STARTING WITH ENTRY 2.
C
      ISUB=1
C
C     READ INFORMATION ABOUT THE FINENESS OF THE FIREBALL
C         INTERPOLATION AND FROM WHERE THE FIREBALL INPUT DATA
C         WILL COME
C
      IBFILE=5
      NRAD=5
      NTH=180
```

```
      FBFILE=.FALSE.
      CALL NAMLST(MESH,5,IEFLAG)
      IF(IEFLAG.NE.0) CALL UNCLE(4,6HMSHMKR,25,
     1 25HFIRE NAMELIST INPUT ERROR)
C
C     SEE IF INPUT WILL BE FROM SPECIAL FILE
C
      IF(.NOT.FBFILE) GO TO 200
C
C     YES. PRINT A MESSAGE TELLING THE USER SO.
C
      DO 180 IPX=6,IFD,6
  180 WRITE(IPX,540)
      IBFILE=3
C
C     PRINT THE DATA TABLE HEADER
C
      DO 190 IPX=IPD,IFD,6
  190 WRITE(IPX,570)
C
C     LOOP TO READ DATA POINTS
C
  200 CONTINUE
      ISUB=ISUB+1
C
C     SEE IF NEXT READ WILL OVERFLOW THE TEMPORARY STORAGE
C
      IF(ISUB.GT.IPFB) CALL UNCLE(4,6HMSHMKR,29,
     1 29HTOO MANY FIREBALL INPUT CARDS)
C
C     FIREBALL DATA IS IN SPHERICAL LAGRANGIAN FORM. INPUT IS
C         PRESSURE,RADIAL VELOCITY,SPECIFIC INTERNAL ENERGY,AND
C         DENSITY,IN CGS UNITS. THE VELOCITY IS DEFINED AT THE GIVEN
C         RADIUS. THE SPECIFIC INTERNAL ENERGY AND DENSITY FOR THE
C         CELL BETWEEN XFIRE(K-1) AND XFIRE(K) IS READ IN WITH
C         XFIRE(K). THE FIREBALL IS ASSUMED TO BE
C         CENTERED AT REZY0. THE PRESSURE IS ONLY INFORMATIVE. IT IS
C         NOT USED IN THE CALCULATION BUT RATHER IS RECALCULATED FROM
C         THE INPUT ENERGY AND DENSITY USING THE EQN-OF-STATE.
C
      READ(IBFILE,580) PJLN,UFIRE(ISUB),EFIRE(ISUB),RHOFIR(ISUB),XFIRE(
     1 ISUB)
C
C     CHECK FOR EOF ON INPUT. IF SO,CONSIDER END OF INPUT.
C
      IF(EOF,IBFILE) 230,210
  210 CONTINUE
C
C     CALCULATE THE PRESSURE FROM THE EQN-OF-STATE TO BE USED IN THE
C         PRINTOUT TO COMPARE WITH THE INPUT PRESSURE. A ZERO DENSITY
C         INDICATES THAT ALL THE DATA CARDS HAVE BEEN READ. (A BLANK
C         CARD TERMINATES THE INPUT.)
C
      ROTMP=RHOFIR(ISUB)
      IF(ROTMP.EQ.0.) GO TO 230
C
C     GO GET GAMMA-1 FROM THE EOS. IF SIE.GE.1.E10,METHANE IS
```

```
C           ASSUMED (CONCJ=1.). OTHERWISE,AIR IS USED (CONCJ=0.).
C
      ETMP=EFIRE(ISUB)
      CONCJ=0.
      IF(ETMP.GE.1.E10) CONCJ=1.
      CALL AIR
      PRHO=GMONE*ROTMP*ETMP
C
C     PRINT OUT THE INPUT DATA AND THE COMPARISON PRESSURES FOR
C         THIS DATA POINT
C
      DO 220 IPX=IPD,IFD,6
  220 WRITE(IPX,590) UFIRE(ISUB),EFIRE(ISUB),RHOFIR(ISUB),XFIRE(ISUB),
     1 PJLN,PRHO
C
C     SEE IF RADIUS IS MONOTONIC INCREASING UNLESS FIRST DATA POINT.
C         IN THE LATTER CASE,GO ON AND PROCESS THE NEXT POINT.
C
      IF(ISUB.EQ.2) GO TO 200
      IF(XFIRE(ISUB).LE.XFIRE(ISUB-1)) CALL UNCLE(4,6HMSHMKR,40,
     1 40HINPUT RADII ARE NOT MONOTONIC INCREASING)
C
C     GO BACK AND PROCESS THE NEXT POINT
C
      GO TO 200
  230 CONTINUE
      DO 240 IPX=6,IFD,6
  240 WRITE(IPX,550)
C
C     GET RID OF THE FIREBALL INPUT DATA FILE IF ONE EXISTS
C
      IF(FBFILE) CALL CLOSIT(3)
C
C     ALL DONE READING INPUT. SET THE DATA POINT COUNT TO THE TRUE NO.
C
      ISUB=ISUB-1
C
C     SET THE QUANTITIES AT THE CENTER OF THE SPHERE. THE ENERGY AND
C         DENSITY ARE SET TO THOSE AT THE FIRST DATA POINT READ. THE
C         VELOCITY IS ZERO,OF COURSE.
C
      UFIRE(1)=0.
      EFIRE(1)=EFIRE(2)
      RHOFIR(1)=RHOFIR(2)
      XFIRE(1)=0.
C
C     LOOP OVER ALL ZONES AND INTERPOLATE THE CELL QUANTITIES
C
      CALL START
      DO 300 J=2,JP1
      DO 290 I=1,IBAR
C
C     FIND THE VERTICES OF THE CELL OF INTEREST
C
      VXL=X(IJ)
      IPJ=IJ+NQ
      VXR=X(IPJ'
```

```
      VYB=Y(IJ)
      VYT=Y(IJP)
C
C     CONC(IJ)=0. INDICATES NOTHING HAS BEEN SET IN THE
C         ZONE YET
C
      CONC(IJ)=0.
C
C     LOOP OVER ALL 1-D ZONES
C
      DO 280 K=2,ISUB
C
C     SEE IF ZONE COULD NOT POSSIBLY FALL WITHIN THE YAQUI ZONE
C
      RVRT=VXR*VXR+(VYT-REZY0)**2
      RVRB=VXR*VXR+(VYB-REZY0)**2
      RFIRE=XFIRE(K-1)**2
      IF(RFIRE.GT.RVRT.AND.RFIRE.GT.RVRB) GO TO 280
      RVLT=VXL*VXL+(VYT-REZY0)**2
      RVLB=VXL*VXL+(VYB-REZY0)**2
      RFIRE=XFIRE(K)**2
      IF(RFIRE.LT.RVLT.AND.RFIRE.LT.RVLB) GO TO 280
C
C     LOOP OVER ANGULAR INCREMENTS
C
      THETA2=0.
      DNTH=180./FLOAT(NTH)
      DO 270 IT=1,NTH
      THETA1=THETA2
      THETA2=THETA2+DNTH
      CTHETA=.5*(THETA1+THETA2)
      SINT=SDG(CTHETA)
      COST=CDG(CTHETA)
C
C     LOOP OVER RADIAL INCREMENTS
C
      R2=XFIRE(K-1)
      DR=(XFIRE(K)-R2)/FLOAT(NRAD)
      DO 260 IR=1,NRAD
      R1=R2
      R2=R2+DR
      CR=.5*(R1+R2)
      XCEN=CR*SINT
C
C     SEE IF THE X-COORDINATE OF THE CENTER OF THE LAGRANGIAN
C         PIECE LIES IN THE YAQUI CELL OF INTEREST
C
      IF(XCEN.LT.VXL.OR.XCEN.GT.VXR) GO TO 260
C
C     YES, CHECK THE Y-COORDINATE
C
      YCEN=CR*COST+REZY0
      IF(YCEN.LT.VYB.OR.YCEN.GT.VYT) GO TO 260
C
C     CONSIDER THE PIECE IN THE YAQUI CELL, SEE IF ANY PIECES
C         HAVE PREVIOUSLY BEEN FOUND.
C
```

```
      IF(CONC(IJ).NE.0.) GO TO 250
C
C     NO, INITIALIZE.
C
      RO(IJ)=0.
      U(IJ)=0.
      V(IJ)=0.
      SIE(IJ)=0.
  250 CONTINUE
C
C     COMPUTE THE MASS OF THE PIECE, ACCUMULATE THE VOLUME
C         IN CONC.
C
      VOL=THIRD*(COS(THETA1)-COS(THETA2))*(R2**3-R1**3)
      CONC(IJ)=CONC(IJ)+VOL
      XMASS=VOL*RHOFIR(K)
C
C     COMPUTE THE VELOCITIES OF THE PIECE
C
      RATIO=(CR-XFIRE(K-1))/(XFIRE(K)-XFIRE(K-1))
      RDOT=UFIRE(K-1)+RATIO*(UFIRE(K)-UFIRE(K-1))
      XDOT=RDOT*SINT
      YDOT=RDOT*COST
C
C     DEPOSIT MASS,MOMENTUM,AND ENERGY IN THE YAQUI ZONE
C
      U(IJ)=U(IJ)+XDOT*XMASS
      V(IJ)=V(IJ)+YDOT*XMASS
      SIE(IJ)=SIE(IJ)+EFIRE(K)*XMASS
      RO(IJ)=RO(IJ)+XMASS
  260 CONTINUE
  270 CONTINUE
  280 CONTINUE
      IJ=IJ+NQ
      IJP=IJP+NQ
  290 CONTINUE
      CALL LOOP
  300 CONTINUE
      CALL DONE
C
C     LOOP OVER ALL CELLS AND CHECK FOR FIREBALL CELLS
C
      CALL START
      DO 390 J=2,JP1
      DO 380 I=1,IBAR
      IF(CONC(IJ).EQ.0.) GO TO 370
C
C     FIREBALL CELL FOUND, SEE IF ITS VOLUME IS VERY FAR OFF.
C
      IPJ=IJ+NQ
      VOL=.5*(Y(IJP)-Y(IJ))*(X(IPJ)**2-X(IJ)**2)
      RERROR=(CONC(IJ)-VOL)/VOL
      IF(ABS(RERROR).LT..01) GO TO 370
C
C     ERROR IS LARGE, INFORM THE USER AND CONSIDER CELL AS
C         FIREBALL BOUNDARY CELL.
C
```

```
      DO 310 IPX=6,IFD,6
  310 WRITE(IPX,560) I,J,RERROR
C
C     USE VALUES FROM NEIGHBORING CELL FOR DETERMINING
C         CONTRIBUTION OF PART OF CELL OUTSIDE OF THE FIREBALL
C
      IF(CONC(IPJ).NE.0.) GO TO 320
C
C     USE CELL TO RIGHT
C
      INDEX=IPJ
      GO TO 360
  320 CONTINUE
      IF(CONC(IJP).NE.0.) GO TO 330
C
C     USE CELL ABOVE
C
      INDEX=IJP
      GO TO 360
  330 CONTINUE
      IF(CONC(IJM).NE.0.) GO TO 340
C
C     USE CELL BELOW
C
      INDEX=IJM
      GO TO 360
  340 CONTINUE
      IPJP=IJP+NQ
      IF(CONC(IPJP).NE.0.) GO TO 350
C
C     USE CELL TO UPPER RIGHT
C
      INDEX=IPJP
      GO TO 360
  350 CONTINUE
      IPJM=IJM+NQ
      IF(CONC(IPJM).NE.0.) CALL UNCLE(1,6HMSHMKR,50,
     1 50HERROR IN PROCESSING CELL WITH LARGE RELATIVE ERROR)
C
C     USE CELL TO LOWER RIGHT
C
      INDEX=IPJM
  360 CONTINUE
C
C     ADD IN CONTRIBUTION FROM PART OF CELL OUTSIDE OF THE FIREBALL
C
      DV=VOL-CONC(IJ)
      DM=DV*RO(INDEX)
      RO(IJ)=RO(IJ)+DM
      SIE(IJ)=SIE(IJ)+DM*SIE(INDEX)
  370 CONTINUE
      IJ=IJ+NQ
      IJP=IJP+NQ
      IJM=IJM+NQ
  380 CONTINUE
      CALL LOOP
  390 CONTINUE
```

```
C
C     ALL FIREBALL CELLS ARE NOW COMPLETELY DEFINED.
C         GO BACK AND DEFINE THE VELOCITIES. STORE THEM
C         TEMPORARILY IN UG,VG.
C
      CALL START
      DO 440 J=2,JP1
      DO 430 I=1,IBAR
      UG(IJ)=0.
      VG(IJ)=0.
      IPJ=IJ+NQ
C
C     SEE IF THE CURRENT CELL IS A FIREBALL CELL
C
      IF(CONC(IJ).EQ.0.) GO TO 420
C
C     YES. ALL NEIGHBORING CELLS OF THE LOWER LEFTHAND VERTEX
C         MUST ALSO BE FIREBALL CELLS OR VELOCITY IS SET TO ZERO.
C
      IF(I.EQ.1) GO TO 400
      IMJ=IJ-NQ
      IF(CONC(IMJ).EQ.0.) GO TO 420
      IMJM=IJM-NQ
      IF(CONC(IMJM).EQ.0.) GO TO 420
  400 CONTINUE
      IF(CONC(IJM).EQ.0.) GO TO 420
C
C     ALL NEIGHBORS ARE FIREBALL CELLS. DIVIDE UP THE MOMENTUM.
C
      XMASS=RO(IJ)+RO(IJM)
      UG(IJ)=U(IJ)+U(IJM)
      VG(IJ)=V(IJ)+V(IJM)
      IF(I.EQ.1) GO TO 410
      IMJ=IJ-NQ
      IMJM=IJM-NQ
      XMASS=XMASS+RO(IMJ)+RO(IMJM)
      UG(IJ)=UG(IJ)+U(IMJ)+U(IMJM)
      VG(IJ)=VG(IJ)+V(IMJ)+V(IMJM)
  410 CONTINUE
      UG(IJ)=UG(IJ)/XMASS
      VG(IJ)=VG(IJ)/XMASS
  420 CONTINUE
      IJ=IJ+NQ
      IJP=IJP+NQ
      IJM=IJM+NQ
  430 CONTINUE
      CALL LOOP
  440 CONTINUE
      CALL DONE
C
C     NOW WE HAVE VERTEX VELOCITIES IN UG,VG. STORE THEM IN U,V
C         AND CONVERT MASSES TO DENSITIES AND INTERNAL ENERGIES
C         TO SPECIFIC INTERNAL ENERGIES.
C
      CALL START
      DO 470 J=2,JP1
      DO 460 I=1,IBAR
```

```
      IF(CONC(IJ).EQ.0.) GO TO 450
      U(IJ)=UG(IJ)
      V(IJ)=VG(IJ)
      IPJ=IJ+NQ
      VOL=.5*(Y(IJP)-Y(IJ))*(X(IPJ)**2-X(IJ)**2)
      SIE(IJ)=SIE(IJ)/RO(IJ)
      RO(IJ)=RO(IJ)/VOL
  450 CONTINUE
      IJ=IJ+NQ
      IJP=IJP+NQ
  460 CONTINUE
      CALL LOOP
  470 CONTINUE
      CALL DONE
C
C     SET CONCENTRATIONS. CONCJ=1. INDICATES METHANE,CONCJ=0. IS AIR.
C         EVERYTHING WITH SIE.GE.1.E10 IS CONSIDERED METHANE. ALL ELSE
C         IS AIR.
C
      CALL START
C
C     SET THE BOTTOM FICTITIOUS ROW
C
      DO 480 I=1,IP1
      CONC(IJM)=0.
      IJM=IJM+NQ
  480 CONTINUE
C
C     SET THE REAL ROWS AND THE RIGHT FICTITIOUS COLUMN
C
      DO 500 J=2,JP1
      DO 490 I=1,IP1
      CONC(IJ)=0.
      IF(SIE(IJ).GE.1.E10.AND.I.NE.IP1) CONC(IJ)=1.
      IJ=IJ+NQ
  490 CONTINUE
      CALL LOOP
  500 CONTINUE
C
C     SET THE TOP FICTITIOUS ROW
C
      DO 510 I=1,IP1
      CONC(IJ)=0.
      IJ=IJ+NQ
  510 CONTINUE
      CALL DONE
C
C     ZERO OUT U AND V IN THE BOTTOM FICTITIOUS ROW
C
      CALL START
      DO 520 I=1,IP1
      U(IJM)=0.
      V(IJM)=0.
      IJM=IJM+NQ
  520 CONTINUE
      CALL DONE
      RETURN
```

```
  530 CONTINUE
      CALL UNCLE(4,6HMSHMKR,34,34HMESH NAMELIST INITIALIZATION ERROR)
C
  540 FORMAT(/1H ,34HFIREBALL INPUT WILL BE FROM FILE 3)
  550 FORMAT(1H /1H ,28HALL FIREBALL INPUT DATA READ/1H )
  560 FORMAT(1H ,5HCELL ,2I5,32H HAS A RELATIVE VOLUME ERROR OF ,1PE12.5
     1 ,1H.,2X,46HIT WILL BE TREATED AS A FIREBALL BOUNDARY CELL)
  570 FORMAT(1H1,22HMESHMKR FIREBALL INPUT//1H ,13X1HU,19X1HE,18X3HRHO,
     1 17X1HX,19X1HP,17X4HPRHO/1H )
  580 FORMAT(5E15.0)
  590 FORMAT(1H ,1P6E19.7)
      END
```

```
      SUBROUTINE MTHANE
C
C     ROUTINE TO CALCULATE THE FACTOR GAMMA-1 FOR METHANE
C
C     UNITS ARE ALL CGS
C
C     ORIGINALLY OBTAINED FROM THE AIR FORCE WEAPONS LAB
C     MODIFIED BY J.L.NORTON,LASL T-3,1974
C
      COMMON/EQNST/RHO,E,GMONE
      DIMENSION A(8)
      DATA A/3.21782E-1,-1.56848E-4,-8.78899E-2,1.25271E-3,1.46612E-2,
     1 -1.72460E-6,-7.72322E-4,-8.63413E-6/
      U=E*1.E-10
      R=ALOG(RHO)
      GMONE=A(1)+R*(A(2)+U*(A(4)+U*(A(6)+A(8)*U)))+U*(A(3)+U*(A(5)+A(7)*
     1 U))
      RETURN
      END
```

```
      SUBROUTINE NAMLST(TABLE,IF,IEFLAG)
C
C     ROUTINE TO REPLACE COMPILER/SYSTEM DEPENDENT NAMELIST INPUT.
C         HERE,AN ATTEMPT HAS BEEN MADE TO ELIMINATE AS MANY
C         COMPILER- AND/OR SYSTEM-DEPENDENT FEATURES AS POSSIBLE. THOSE
C         STILL REMAINING ARE BOOLEAN ALGEBRA,60 BIT WORD ASSUMPTION,
C         INLINE SHIFT FUNCTION (CAN BE REPLACED BY ASSEMBLY-LANGUAGE
C         FUNCTION IF IT IS TYPED INTEGER),AND TWO ASSEMBLY-LANGUAGE
C         ROUTINES TO FETCH AND STORE WORDS FROM AND INTO ABSOLUTE
C         MEMORY LOCATIONS.
C
C     TABLE - NAMELIST DATA TABLE
```

```
C      IF - INPUT FILESET
C      IEFLAG - ERROR FLAG
C             - 0 - NO ERROR
C             - -1 - END-OF-FILE ON FILESET IF
C             - N (.NE.0 OR -1) - ERROR EXIT
C                  (ERROR MESSAGE WILL BE PRINTED BY MESSOT)
C                  (IF .LT.0,EOF ALSO OCCURRED)
C
C      LIMITATIONS - CURRENTLY,ONLY SMALL CORE VARIABLES MAY BE READ.
C          ARRAYS OF UP TO FOUR SUBSCRIPTS CAN BE INPUT ALTHOUGH MANY
C          COMPILERS CAN HANDLE A MAXIMUM OF ONLY THREE SUBSCRIPTS.
C
C      INPUT RULES - DATA MAY BE INPUT IN ANY OF THREE GENERAL FORMS --
C                          V=D1
C                          A=D1,D2,....,DN
C                          A(SUB)=D1,D2,....,DN
C          WHERE V IS A NON-SUBSCRIPTED VARIABLE,
C                A IS AN ARRAY,
C                SUB REPRESENTS FROM 1 TO 4 SUBSCRIPTS (INTEGER CONSTANT)
C            AND DN REPRESENTS A DATA ELEMENT.
C          THE DATA ELEMENTS MAY BE OF A NUMBER OF TYPES --
C               INTEGER - BASE 10 (EXAMPLE,-13569)
C                       - BASE 8 (FOLLOW CONSTANT WITH B) (7765B)
C               REAL    - FIXED (13.25)
C                       - FLOATING (3.265E29)
C               COMPLEX - (-13.59,3.14E-7)
C               DOUBLE  - 3.14159265358979323846D3
C               LOGICAL - .TRUE.,.FALSE.,.T.,.F.
C               HOLLERITH - LEFT-JUSTIFIED,BLANK-FILL (5HTHING)
C                         - LEFT-JUSTIFIED,ZERO-FILL (3LOUT)
C                         - RIGHT-JUSTIFIED,ZERO-FILL (4RWORD)
C          MULTIPLIERS ARE ALLOWED,BUT ONLY FOR ONE ELEMENT AT A TIME.
C               FOR EXAMPLE,5*23,6*15.3,21*-27.3E7,5*4HTEST ARE ALL
C               LEGAL. 6*(13.,21.7) WOULD WORK,BUT ONLY BECAUSE THE
C               QUANTITY INSIDE THE PARENTHESES IS INTERPRETED AS A
C               SINGLE COMPLEX CONSTANT. 22*(1.2,3.4,5.6) WOULD BE
C               FLAGGED AS AN ERROR. 5*13HABCDEFGHIJKLM IS ALSO ILLEGAL
C               AS THE CONSTANT IS MORE THAN ONE WORD LONG.
C          INTEGERS WITH MORE THAN 14 DIGITS AND REAL CONSTANTS OF
C               MORE THAN 14 DIGITS NOT INDICATED AS DOUBLE PRECISION
C               ARE ILLEGAL. FOR EXAMPLE,PI=3.1415926535897932 IS WRONG
C               BUT PI=3.1415926535897932D IS O.K.
C
C      *****************************************************************
C      NOTE CAREFULLY - NO CONVERSIONS OF DATA TYPE ARE MADE. THIS GIVES
C          ONE THE CAPABILITY OF STORING REAL CONSTANTS INTO INTEGER
C          VARIABLES,FOR EXAMPLE. HOWEVER,IF ONE FORGETS TO PUNCH A
C          DECIMAL POINT (X=5),THE CONSTANT WILL BE STORED AS AN
C          INTEGER.
C      *****************************************************************
C
C      OTHER CONVENTIONS AND CAPABILITIES ARE AS FOLLOWS --
C          ANY CARD WITH C$ IN CC1-2 WILL BE PRINTED OUT AND IGNORED.
C          ANY CARD WITH P$ IN CC1-2 WILL TURN ON A PRINT SWITCH AND
C               EACH CARD OF THE NAMELIST INPUT RECORD (UP TO THE NEXT $)
C               WILL BE PRINTED BEFORE PROCESSING.
C          AN ISOLATED - IS ILLEGAL. -E*-1.E0. -D*-1.D0. -.TRUE.*.FALSE.
```

```
      C           AND -.FALSE.=.TRUE. , -B=-0B. A MINUS SIGN IN FRONT OF AN
      C           OCTAL CONSTANT CAUSES THE CONSTANT TO BE COMPLEMENTED.
      C           A MINUS SIGN IN FRONT OF A HOLLERITH CONSTANT IS IGNORED.
      C      AN R HOLLERITH CONSTANT CANNOT BE MORE THAN 10 CHARACTERS
      C           LONG.
      C      THE TERMINATION $ MUST NOT OCCUR IN CC1 OR 2. IF IT OCCURS
      C           IN CC1,IT WILL BE IGNORED. AN ERROR WILL BE RETURNED IF
      C           IT IS IN CC2.
      C      A NAMELIST RECORD MAY BE OF ANY LENGTH. BLANKS ARE SIGNIFICANT
      C           AND MAY OCCUR ONLY AROUND NON-NUMERIC CHARACTERS OTHER
      C           THAN EXPONENTS. FOR EXAMPLE,LEGAL BLANKS ARE X = 5.,
      C           Y = - 10.3, L = .TRUE., C = ( 5. , 6. ), I=- 13.
      C           ILLEGAL BLANKS WOULD INCLUDE X=5. E 6, L= . T .,Y=. 5  .
      C           THE NAMELIST RECORD IS TREATED AS ONE LONG STRING OF
      C           CHARACTERS,ALL 80 CHARACTERS OF EACH CARD BEING SCANNED,
      C           EXCEPT THAT CC2 OF EACH CARD MAY NOT BE A $ UNLESS IT IS
      C           PART OF A HOLLERITH FIELD.
      C      BECAUSE OF THE UNLIMITED LENGTH OF A NAMELIST RECORD,A
      C           HOLLERITH CONSTANT MAY BE OF UNLIMITED LENGTH.
      C      BECAUSE TYPE IS NOT CHECKED,STORING A DOUBLE-PRECISION
      C           CONSTANT INTO A SINGLE-PRECISION VARIABLE WILL CLOBBER
      C           THE FOLLOWING LOCATION. (X=5.D WILL STORE A ZERO INTO
      C           THE LOCATION FOLLOWING X ASSUMING X IS NOT DOUBLE OR
      C           COMPLEX). A SIMILAR WARNING CAN BE GIVEN FOR X=(1.,2.).
      C
      C  ******************************************************************
      C  EXAMPLE OF USAGE -
      C
      C      SUPPOSE ONE WISHED TO REPLACE THE STATEMENTS
      C
      C           DIMENSION A(20),B(5,3),X(5,10,15)
      C           NAMELIST/CARDN/I,J,K,A,B,X
      C           READ(5,CARDN)
      C
      C      WITH SYSTEM-INDEPENDENT INPUT. FURTHERMORE,SUPPOSE ONE WISHED
      C           TO HAVE PS AND CS CARDS COME OUT ON BOTH PAPER AND FILM
      C           AND ERROR MESSAGES COME OUT ON FILE 59,THEN ONE WOULD
      C           NEED THE FOLLOWING - -
      C
      C           DIMENSION A(20),B(5,3),X(5,10,15),ITAB(2),TABLE(3,7)
      C       C
      C       C   SET UP FILE TABLE FOR MESSAGES
      C       C
      C           DATA ITAB/3LOUT,4LFILM/,JTAB/59/
      C           CALL NAMPRT(2,ITAB)
      C           CALL ERRPRT(1,JTAB)
      C       C
      C       C   DEFINE THE NAMELIST TABLE. IF THERE ARE TO BE N UNIQUE
      C       C        INPUT VARIABLE NAMES,THEN 3*(N+1) TABLE LOCATIONS
      C       C        ARE NEEDED. THUS,FOR SIX VARIABLES,3*7 LOCATIONS
      C       C        MUST BE SET ASIDE. THE THIRD ARGUMENT IS THE
      C       C        SECOND SUBSCRIPT OF TABLE. IF ANY ERRORS OCCUR IN
      C       C        SUBSEQUENT CALLS TO TABSET,TRANSFER WILL OCCUR TO
      C       C        THE STATEMENT NO. ASSIGNED TO IERRT. THIS ELIMINATES
      C       C        ERROR CHECKING AFTER EACH CALL TO TABSET. IEFLAG CAN
      C       C        THEN BE EXAMINED TO DETERMINE WHAT TYPE OF ERROR
      C       C        OCCURRED.
```

```
C          C
C                ASSIGN 999 TO IERRT
C                CALL TABDEF(TABLE,5HCARDN,7,IERRT)
C          C
C          C     ENTER EACH VARIABLE INTO THE TABLE. IF THERE IS SOME
C          C          ERROR,IEFLAG IS RETURNED NON-ZERO.
C          C
C                CALL TABSET(TABLE,1HI,I,IEFLAG,0,0,0,0)
C          C
C          C     (THE 0 DENOTES A NON-SUBSCRIPTED VARIABLE)
C          C
C                CALL TABSET(TABLE,1HJ,J,IEFLAG,0,0,0,0)
C                CALL TABSET(TABLE,1HK,K,IEFLAG,0,0,0,0)
C                CALL TABSET(TABLE,1HA,A,IEFLAG,1,0,0,0)
C          C
C          C     (THE 1 DENOTES A SUBSCRIPTED VARIABLE BUT THE SUBSCRIPT
C          C          NEED NOT BE GIVEN FOR A SINGLY-SUBSCRIPTED VARIABLE)
C          C
C                CALL TABSET(TABLE,1HB,B,IEFLAG,2,5,0,0)
C          C
C          C     (THE VARIABLE IS DOUBLY-SUBSCRIPTED BUT ONLY THE FIRST
C          C          SUBSCRIPT NEED BE GIVEN)
C          C
C                CALL TABSET(TABLE,1HX,X,IEFLAG,3,5,10,0)
C          C
C          C     (THE VARIABLE IS TRIPLY-SUBSCRIPTED BUT ONLY THE FIRST
C          C          TWO SUBSCRIPTS NEED BE GIVEN)
C          C
C          C     NOTE THAT TABSET IS CALLED WITH THE FULL NO. OF ARGUMENTS
C          C         WHETHER THEY ARE USED OR NOT. THIS IS NOT STRICTLY
C          C         NECESSARY ON MOST COMPILERS. CHECK YOUR LOCAL
C          C         CONVENTIONS.
C          C
C          C     READ IN A NAMELIST CARD. IF IEFLAG RETURNS 0,ALL WAS OK.
C          C          IF IEFLAG RETURNS -1,A NORMAL EOF WAS ENCOUNTERED.
C          C          FOR ALL OTHER VALUES OF IEFLAG,AN ERROR OCCURRED.
C          C          IF IEFLAG.LT.0,AN EOF ALSO OCCURRED.
C          C
C                CALL NAMLST(TABLE,5,IEFLAG)
C          C
C          C     (5 IS THE INPUT FILE NO.)
C          C
C                IF(IEFLAG.EQ.0) GO TO 10
C                IF(IEFLAG.EQ.(-1)) GO TO 20
C          C
C          C     A NAMELIST ERROR OCCURRED
C          C
C                STOP 7
C          C
C          C     A NAMELIST TABLE INITIALIZATION ERROR OCCURRED
C          C
C            999 CONTINUE
C                . . .
C          C
C          C     AN EOF OCCURRED
C          C
C             20 CONTINUE
```

```
C              . . .
C        C
C        C     NORMAL EXIT
C        C
C           10 CONTINUE
C              . . .
C     ******************************************************************
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
C     THIS VERSION RUNS ON A CDC 6600 OR 7600 USING THE RUN COMPILER
C
      DIMENSION TABLE(3,1),IDSTR(80)
      INTEGER TABLE,AND,OR,COMP,SHIFT
      LOGICAL LIST,CMPLEX,DIM,DELIM,FIX,MULT,CFRST,HOLLER,DOBLE
      LOGICAL SAVE,FIRST,DONE,EXPN,PERIOD,LCONT,LCONF,CDONE,CPFND
      COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
     1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
      COMMON/ERRORC/IERRT
      COMMON/CSHIFT/XLEFT,XRIGHT,XDUM
      DOUBLE PRECISION DB,DBP,DCONST,DTEMP,DZERO
      EQUIVALENCE(DB,CREAL)
      EQUIVALENCE(IVSUM,XSUM)
      DATA LCONT,LCONF/.TRUE.,.FALSE./
      DATA IAF,IAL/1RA,1RZ/
      DATA INF,INL/1R0,1R9/
      DATA DCONST/10.,0./,DZERO/0.,0./
      DATA IEMIN,IEMAX/-294,322/
      DATA NDSP,NDDP/14,28/
      DATA INAME/0/
C
C     LIST = IF TRUE,EACH NAMELIST CARD INPUT WILL BE LISTED
C
      LIST=.FALSE.
C
C     BEGINNING OF CODE FOR PROCESSING A NAMELIST RECORD
C
   10 CONTINUE
C
C     RESET THE ERROR FLAG
C
      IEFLAG=0
C
C     READ A CARD
C
      CALL READIT(IF,IEFLAG)
C
C     IF AN END-OF-FILE WAS ENCOUNTERED,QUIT
C
      IF(IEFLAG.LT.0) RETURN
C
C     ISUB IS THE POINTER TO THE COLUMN BEING PROCESSED OF THE CARD
C         LAST READ. BEGIN LOOKING AT CC2.
C
      ISUB=2
C
C     IF CC2 IS NOT A DOLLAR,IGNORE THE CARD AND GO BACK AND READ
```

```
C           ANOTHER ONE
C
        IF(ICHAR(2).NE.1H$) GO TO 10
C
C       CC2 WAS A DOLLAR. IF CC1 IS A =P=,TURN ON THE LIST FLAG.
C
        IF(ICHAR(1).EQ.1HP) LIST=.TRUE.
C
C       IF CC1 IS A C,PRINT THE CARD AND GO READ ANOTHER
C
        IF(ICHAR(1).NE.1HC) GO TO 20
        CALL MESSOT(1)
        GO TO 10
C
C       CARD IS TO BE PROCESSED.
C
     20 CONTINUE
C
C       IVNUM IS THE NO. OF CHARACTERS READ IN THE CURRENT STRING
C
        IVNUM=0
C
C       *****************************************************************
C       DECODE THE NAMELIST NAME
C       *****************************************************************
C
     30 CONTINUE
C
C       GO TO THE NEXT COLUMN
C
        ISUB=ISUB+1
C
C       ITEST IS THE CHARACTER IN CC ISUB,LEFT=JUSTIFIED,BLANK FILL
C
        ITEST=ICHAR(ISUB)
C
C       CHECK FOR A BLANK
C
        IF(ITEST.EQ.1H ) GO TO 70
C
C       NO. UP THE CHARACTER COUNT. IF MORE THAN SIX,FATAL ERROR.
C
        IVNUM=IVNUM+1
        IF(IVNUM.LE.6) GO TO 40
        CALL MESSOT(2)
        IEFLAG=2
        RETURN
     40 CONTINUE
C
C       ITEMP IS THE CHARACTER IN CC ISUB,RIGHT=JUSTIFIED,ZERO FILL
C
        ITEMP=AND(SHIFT(ITEST,-54),77B)
C
C       CHECK FOR THE FIRST CHARACTER OF THE STRING
C
        IF(IVNUM.GT.1) GO TO 50
C
```

```
C       YES, IF NON-ALPHABETIC,FATAL ERROR,
C
        IF(ITEMP.GE.IAF.AND.ITEMP.LE.IAL) GO TO 60
        CALL MESSOT(23)
        IEFLAG=23
        RETURN
     50 CONTINUE
C
C       IF ANY CHARACTER IS NOT A LETTER OR A NUMBER,FATAL ERROR.
C
        IF((ITEMP.GE.INF.AND.ITEMP.LE.INL).OR.(ITEMP.GE.IAF.AND.ITEMP.LE.
       1 IAL)) GO TO 60
        CALL MESSOT(24)
        IEFLAG=24
        RETURN
     60 CONTINUE
C
C       STORE THE CHARACTER AND GO LOOK AT THE NEXT
C
        IDSTR(IVNUM)=ITEST
        GO TO 30
C
C       A BLANK CHARACTER HAS BEEN FOUND
C
     70 CONTINUE
C
C       IF IT IS THE FIRST CHARACTER AFTER THE $,FATAL ERROR
C
        IF(IVNUM.GT.0) GO TO 80
        CALL MESSOT(11)
        IEFLAG=11
        RETURN
     80 CONTINUE
C
C       GO PUT THE STRING INTO ONE WORD (INAME),LEFT-JUSTIFIED,BLANK FILL
C
        CALL MASH(IDSTR,IVNUM,INAME,IEFLAG)
        IF(IEFLAG.NE.0) RETURN
C
C       SEE IF IT MATCHES THE NAME IN THE NAMELIST TABLE. IF NOT,GO BACK
C           AND READ THE NEXT CARD.
C
        IF(TABLE(1,1).NE.INAME) GO TO 10
C
C       NAME MATCHES. INE IS THE NUMBER OF VARIABLES REPRESENTED IN THE
C           TABLE +1. DELIM=.FALSE. INDICATES THAT A DELIMITER (, OR $)
C           IS NOT EXPECTED AS THE NEXT CHARACTER. FIRST=.TRUE. INDICATES
C           THAT A VARIABLE NAME MUST BE FOUND BEFORE A CONSTANT.
C
        INE=TABLE(3,1)+1
        DELIM=.FALSE.
        FIRST=.TRUE.
C
C       LIST THE CARD IF THE FLAG IS ON
C
        IF(LIST) CALL MESSOT(1)
        GO TO 100
```

```
C
C     *****************************************************************
C     MAIN LOOP FOR DECODING VARIABLE NAMES AND CONSTANTS
C     *****************************************************************
C
   90 CONTINUE
C
C     ONE VARIABLE NAME AND CONSTANT HAVE ALREADY BEEN PROCESSED,
C         EITHER A VARIABLE OR A CONSTANT CAN OCCUR NEXT,
C
      FIRST=.FALSE.
C
C     -----------------------------------------------------------------
C     SUBLOOP TO DETERMINE WHETHER NEXT STRING IS VARIABLE NAME OR
C         CONSTANT
C     -----------------------------------------------------------------
C
  100 CONTINUE
C
C     NXTCOL PUTS THE CONTENTS OF THE NEXT CARD COLUMN INTO ITEST,ALSO
C         UPDATING ISUB, IF NECESSARY,ANOTHER CARD IS READ AND ERROR
C         CHECKS ARE PERFORMED,
C
      CALL NXTCOL(IF,IEFLAG)
C
C     IF ANY ERRORS,QUIT RIGHT HERE
C
      IF(IEFLAG.NE.0) RETURN
C
C     NO ERRORS, SEE IF NEXT CHARACTER IS A BLANK, IF IT IS,GO ON TO
C         THE NEXT COLUMN,
C
      IF(ITEST.EQ.1H ) GO TO 100
C
C     NOT A BLANK, SEE IF IT IS A DOLLAR, IF SO,WE ARE ALL DONE,
C
      IF(ITEST.EQ.1H$) RETURN
C
C     NOT A DOLLAR, SEE IF A DELIMITER IS EXPECTED,
C
      IF(.NOT.DELIM) GO TO 110
C
C     YES, IF CHARACTER IS NOT A COMMA,FATAL ERROR,
C
      IF(ITEST.NE.1H,) GO TO 260
C
C     CHARACTER IS A COMMA, REMOVE DELIMITER FLAG AND GO EXAMINE
C         THE NEXT COLUMN,
C
      DELIM=.FALSE.
      GO TO 90
  110 CONTINUE
C
C     SEE IF CHARACTER IS ALPHABETIC
C
      ITEMP=AND(SHIFT(ITEST,-54),77B)
      IF(ITEMP.GE.IAF.AND.ITEMP.LE.IAL) GO TO 130
```

```
C
C     NO. IF A NAME IS EXPECTED,FATAL ERROR.
C
      IF(FIRST) GO TO 120
C
C     A NAME IS NOT NECESSARILY EXPECTED. LEGAL CHARACTERS ARE NUMBERS,
C         +,-,.,OR (. IF NONE OF THESE,FATAL ERROR. IF ONE OF THESE,
C         PROCEED TO SECTION WHICH DECODES CONSTANTS.
C
      IF(ITEMP.GE.INF.AND.ITEMP.LE.INL) GO TO 400
      IF(ITEST.EQ.1H+.OR.ITEST.EQ.1H-.OR.ITEST.EQ.1H..OR.ITEST.EQ.1H()
     1 GO TO 400
  120 CONTINUE
      CALL MESSOT(13)
      IEFLAG=13
      RETURN
C
C     ------------------------------------------------------------------
C     SUBLOOP TO DECODE VARIABLE NAMES
C     ------------------------------------------------------------------
C
  130 CONTINUE
C
C     STORE THE FIRST CHARACTER
C
      IVNUM=1
      IDSTR(1)=ITEST
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     SUBLOOP TO GET CHARACTERS OF VARIABLE NAME
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
  140 CONTINUE
C
C     GET THE NEXT CHARACTER AND CHECK FOR ERRORS
C
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     SEE IF CHARACTER IS ALPHANUMERIC. IF NOT,ASSUME THAT THE END OF
C         THE NAME HAS BEEN REACHED.
C
      ITEMP=AND(SHIFT(ITEST,-54),77B)
      IF((ITEMP.GE.IAF.AND.ITEMP.LE.IAL).OR.(ITEMP.GE.INF.AND.ITEMP.LE.
     1 INL)) GO TO 150
      GO TO 170
  150 CONTINUE
C
C     CHARACTER IS ALPHANUMERIC. SEE IF IT IS NO. 7. IF SO,FATAL ERROR.
C
      IF(IVNUM.GE.6) GO TO 160
C
C     NOT CHARACTER 7. STORE AND GO ON TO THE NEXT.
C
      IVNUM=IVNUM+1
      IDSTR(IVNUM)=ITEST
      GO TO 140
```

```
  160 CONTINUE
      CALL MESSOT(3)
      IEFLAG=3
      RETURN
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
  170 CONTINUE
C
C     NAME HAS BEEN DECODED, PUT INTO ONE WORD,
C
      CALL MASH(IDSTR,IVNUM,INAME,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     SCAN THE TABLE TO SEE IF IT IS A LEGAL VARIABLE, IF NOT,
C         FATAL ERROR,
C
      DO 180 I=2,INE
  180 IF(INAME.EQ.TABLE(1,I)) GO TO 190
      CALL MESSOT(4)
      IEFLAG=4
      RETURN
C
C     LEGAL VARIABLE NAME HAS BEEN FOUND, INITIALIZE FOR SUBSEQUENT
C         SCAN,
C
  190 CONTINUE
C
C     FIRST HERE DENOTES THAT A CONSTANT HAS NOT  YET BEEN FOUND FOR
C         THIS VARIABLE
C
      FIRST=.TRUE.
C
C     IENTRY IS THE SECOND SUBSCRIPT OF THE TABLE ENTRY FOR THIS
C         VARIABLE
C
      IENTRY=I
C
C     ISBSPT IS THE STORAGE OFFSET FOR THE VARIABLE, IF THE VARIABLE
C         IS LOCATED BEGINNING IN MEMORY LOCATION N,THEN THE NEXT
C         CONSTANT WILL BE STORED STARTING IN LOCATION N+ISBSPT-1,
C
      ISBSPT=1
C
C     NSUBV(I) IS THE VALUE OF THE ITH SUBSCRIPT
C
      DO 200 I=1,4
  200 NSUBV(I)=0
C
C     NSUB IS THE NUMBER OF SUBSCRIPTS
C
      NSUB=0
C
C     DIM IS .TRUE. IF THE VARIABLE IS DIMENSIONED
C
      DIM=.FALSE.
C
```

```
C       ----------------------------------------------------------------
C       DETERMINE ANY SUBSCRIPTING INFORMATION
C       ----------------------------------------------------------------
C
  210 CONTINUE
C
C     CHECK THE FIRST CHARACTER AFTER THE VARIABLE NAME, IF IT IS AN
C         EQUALS,NO SUBSCRIPTING INFORMATION, PROCEED TO CONSTANT SCAN.
C         OTHER LEGAL CHARACTERS ARE BLANK AND LEFT PAREN, IF NEITHER
C         OF THESE,FATAL ERROR.
C
      IF(ITEST.EQ.1H ) GO TO 220
      IF(ITEST.EQ.1H() GO TO 230
      IF(ITEST.EQ.1H=) GO TO 400
      CALL MESSOT(5)
      IEFLAG=5
      RETURN
C
C     CHARACTER WAS BLANK, FETCH NEXT COLUMN AND CONTINUE.
C
  220 CONTINUE
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
      GO TO 210
C
C     CHARACTER WAS LEFT PAREN, BEGIN SUBSCRIPT DECODING.
C
  230 CONTINUE
C
C     INITIALIZE. IVSUM IS THE CURRENT VALUE OF THE NUMBER BEING DECODED
C         AND IVSIGN IS ITS SIGN (+1 OR -1).
C
      DIM=.TRUE.
      NSUB=1
      IVNUM=0
      IVSUM=0
      IVSIGN=1
      DELIM=.FALSE.
C
C     NOSGN IS THE NO. OF SIGNS ENCOUNTERED IN THE FIELD
C
      NOSGN=0
C
C     +++ SUBLOOP TO DECODE A SUBSCRIPT
C
  240 CONTINUE
C
C     GET THE NEXT CHARACTER
C
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     SEE IF IT IS NUMERIC
C
      ITEMP=AND(SHIFT(ITEST,-54),77B)
      IF(ITEMP.GE.INF.AND.ITEMP.LE.INL) GO TO 250
C
```

```
C      NO, CHECK FOR BLANK,
C
       IF(ITEST.EQ.1H ) GO TO 310
C
C      NO, CHECK FOR COMMA,
C
       IF(ITEST.EQ.1H,) GO TO 340
C
C      NO, CHECK FOR RIGHT PAREN,
C
       IF(ITEST.EQ.1H)) GO TO 360
C
C      CHECK FOR SIGN OF SUBSCRIPT
C
       IF(ITEST.EQ.1H+.OR.ITEST.EQ.1H-) GO TO 290
C
C      NONE OF THESE, FATAL ERROR,
C
       CALL MESSOT(6)
       IEFLAG=6
       RETURN
C
C      CHARACTER WAS NUMERIC, SEE IF A DELIMITER WAS EXPECTED, IF IT WAS,
C          FATAL ERROR,
C
   250 CONTINUE
       IF(.NOT.DELIM) GO TO 270
   260 CONTINUE
       CALL MESSOT(7)
       IEFLAG=7
       RETURN
   270 CONTINUE
C
C      SEE IF THE SUBSCRIPT IS TOO LONG, IF MORE THAN 5 DIGITS,FATAL
C          ERROR,
C
       IF(IVNUM.LT.5) GO TO 280
       CALL MESSOT(43)
       IEFLAG=43
       RETURN
   280 CONTINUE
C
C      SUBSCRIPT IS OK SO FAR, UPDATE ITS VALUE AND CONTINUE,
C
       IVNUM=IVNUM+1
       IVSUM=IVSUM*10+ITEMP-INF
       GO TO 240
   290 CONTINUE
C
C      A SIGN HAS BEEN FOUND, MAKE SURE THERE IS ONLY ONE,
C
       IF(NOSGN.EQ.0) GO TO 300
       CALL MESSOT(44)
       IEFLAG=44
       RETURN
   300 CONTINUE
C
```

```
C      SIGN IS OK, UP THE COUNT,SET THE SIGN VARIABLE,AND CONTINUE SCAN.
C
       NOSGN=1
       IF(ITEST.EQ.1H-) IVSIGN=-1
       GO TO 240
   310 CONTINUE
C
C      CHARACTER IS A BLANK. IF NO NUMBERS HAVE BEEN FOUND YET,GO ON TO
C          THE NEXT CHARACTER.
C
       IF(IVNUM.EQ.0) GO TO 240
C
C      BLANK CONSIDERED FIELD TERMINATOR. TURN ON THE DELIMITER SWITCH.
C
       DELIM=.TRUE.
   320 CONTINUE
C
C      SUBSCRIPT DECODED. MAKE SURE THERE ARE NO MORE THAN 4.
C
       IF(NSUB.GT.4) GO TO 330
C
C      SUBSCRIPT IS OK. STORE THE SIGNED VALUED IN THE SUBSCRIPT ARRAY.
C
       NSUBV(NSUB)=ISIGN(IVSUM,IVSIGN)
C
C      REINITIALIZE AND GO SEARCH FOR NEXT SUBSCRIPT
C
       IVSUM=0
       IVNUM=0
       IVSIGN=1
       NOSGN=0
       NSUB=NSUB+1
       GO TO 240
C
C      MORE THAN FOUR SUBSCRIPTS. FATAL ERROR.
C
   330 CONTINUE
       CALL MESSOT(14)
       IEFLAG=14
       RETURN
   340 CONTINUE
C
C      COMMA ENCOUNTERED. SEE IF A DELIMITER IS EXPECTED.
C
       IF(DELIM) GO TO 350
C
C      NO. SEE IF A FIELD HAS BEGUN. IF NOT,FATAL ERROR. IF SO,CONSIDER
C          FIELD TERMINATED AND GO FINISH PROCESSING FOR THIS SUBSCRIPT.
C
       IF(IVNUM.NE.0) GO TO 320
       CALL MESSOT(8)
       IEFLAG=8
       RETURN
C
C      DELIMITER EXPECTED AND FOUND. TURN OFF FLAG AND CONTINUE SCAN.
C
   350 CONTINUE
```

```
      DELIM=.FALSE.
      GO TO 240
  360 CONTINUE
C
C     RIGHT PAREN FOUND, TERMINATE SUBSCRIPT PROCESSING, CHECK IF PAREN
C         IS FIELD TERMINATOR,
C
      IF(IVNUM.NE.0) GO TO 370
C
C     NO, CHECK FOR ISOLATED SIGN, IF SO,FATAL ERROR,
C
      IF(NOSGN.EQ.0) GO TO 380
      CALL MESSOT(30)
      IEFLAG=30
      RETURN
  370 CONTINUE
C
C     YES, STORE THE SUBSCRIPT AND PRETEND LIKE ANOTHER IS TO BE
C         DECODED.
C
      NSUBV(NSUB)=ISIGN(IVSUM,IVSIGN)
      NSUB=NSUB+1
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
  380 CONTINUE
C
C     LAST SUBSCRIPT COMPLETED, SET NSUB TO THE ACTUAL NO, OF SUBSCRIPTS
C
      NSUB=NSUB-1
C
C     --------------------------------------------------------------------
C
  390 CONTINUE
C
C     SCAN CHARACTERS UNTIL AN = IS FOUND, ONLY OTHER VALID CHARACTER
C         IS A BLANK,
C
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
      IF(ITEST.EQ.1H ) GO TO 390
      IF(ITEST.EQ.1H=) GO TO 400
      CALL MESSOT(9)
      IEFLAG=9
      RETURN
C
C     ********************************************************************
C     SECTION TO DECODE CONSTANTS
C     ********************************************************************
C
  400 CONTINUE
C
C     RESET ALL FLAGS
C
C     CMPLEX = .TRUE. IF A COMPLEX CONSTANT IS BEING DECODED
C     CDONE  = .TRUE. IF THE SECOND HALF OF A COMPLEX CONSTANT HAS
C                     BEEN DECODED BUT THE CLOSING PARENTHESIS HAS NOT
```

```
C                   BEEN FOUND
C       CPFND  = .TRUE. IF THE CLOSING PARENTHESIS OF A COMPLEX CONSTANT
C                   HAS BEEN FOUND
C       DB     =  STORAGE FOR DOUBLE PRECISION CONSTANT
C       MULT   =  .TRUE  IF A MULTIPLIER HAS BEEN FOUND
C       DELIM  =  .TRUE. IF A DELIMITER IS EXPECTED
C
        CMPLEX=.FALSE.
        CDONE=.FALSE.
        CPFND=.FALSE.
        DB=0.
        MULT=.FALSE.
        DELIM=.FALSE.
C
C       ONE RETURNS TO HERE IF THE FIRST NO. OF A COMPLEX CONSTANT HAS
C           BEEN PROCESSED AND STORED
C
    410 CONTINUE
C
C       HOLLER = .TRUE. IF A HOLLERITH CONSTANT IS BEING PROCESSED
C       IC     = CHARACTER COUNT OF CONSTANT STRING
C       DOBLE  = .TRUE. IF A DOBLE  PRECISION CONSTANT IS BEING PROCESSED
C       FIX    = .TRUE. IF A FIXED POINT CONSTANT IS BEING DECODED
C       IVSUM  = CURRENT VALUE OF FIXED POINT FIELD
C       IVNUM  = CURRENT NO. OF DIGITS IN THE FIXED POINT FIELD
C       EXPN   = .TRUE. IF AN EXPONENT IS BEING DECODED
C       XLEFT  = VALUE OF FLOATING POINT CONSTANT TO LEFT OF DECIMAL POINT
C       XRIGHT = VALUE OF FLOATING POINT CONSTANT TO RIGHT OF DECIMAL
C       NOSGN  = NO. OF + OR - SIGNS FOUND IN CONSTANT SO FAR
C       DONE   = .TRUE. IF A DOLLAR SIGN HAS BEEN FOUND
C       IVSIGN = VALUE OF THE LAST SIGN FOUND (+1 OR -1)
C       PERIOD = .TRUE. IF THE LAST CHARACTER FOUND WAS AN ISOLATED PERIOD
C
        HOLLER=.FALSE.
        IC=0
        DOBLE=.FALSE.
        FIX=.TRUE.
        IVSUM=0
        IVNUM=0
        EXPN=.FALSE.
        XLEFT=0.
        XRIGHT=0.
        NOSGN=0
        DONE=.FALSE.
        IVSIGN=1
        PERIOD=.FALSE.
C
C       IF THIS IS THE FIRST CONSTANT TO BE PROCESSED AFTER AN = SIGN
C           (FIRST=.TRUE.) OR A COMPLEX CONSTANT IS BEING DECODED,
C           ONE MUST FETCH THE NEXT CHARACTER. OTHERWISE,IT HAS PREVIOUSLY
C           BEEN READ.
C
        IF(.NOT.FIRST.AND..NOT.CMPLEX) GO TO 430
    420 CONTINUE
C
C       GET THE NEXT CHARACTER AND CHECK FOR ERRORS
C
```

```
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
  430 CONTINUE
C
C     -------------------------------------------------------------------
C     DECIDE WHAT TO DO BASED UPON WHAT CHARACTER WAS LAST READ
C     -------------------------------------------------------------------
C
C     CHECK FOR A BLANK
C
      IF(ITEST.EQ.1H ) GO TO 470
C
C     NO. CHECK FOR A NUMBER.
C
      ITEMP=AND(SHIFT(ITEST,-54),77B)
      IF(ITEMP.GE.INF.AND.ITEMP.LE.INL) GO TO 450
C
C     NO. CHECK FOR A COMMA.
C
      IF(ITEST.EQ.1H,) GO TO 1050
C
C     NO. CHECK FOR A PERIOD.
C
      IF(ITEST.EQ.1H.) GO TO 1140
C
C     NO. CHECK FOR A PLUS OR MINUS.
C
      IF(ITEST.EQ.1H+) GO TO 990
      IF(ITEST.EQ.1H-) GO TO 990
C
C     NO. CHECK FOR A DOLLAR SIGN.
C
      IF(ITEST.EQ.1H$) GO TO 1080
C
C     NO. CHECK FOR A T OR F.
C
      IF(ITEST.EQ.1HT) GO TO 1170
      IF(ITEST.EQ.1HF) GO TO 1170
C
C     NO. CHECK FOR AN E.
C
      IF(ITEST.EQ.1HE) GO TO 1100
C
C     NO. CHECK FOR AN ASTERISK.
C
      IF(ITEST.EQ.1H*) GO TO 680
C
C     NO. CHECK FOR AN H.
C
      IF(ITEST.EQ.1HH) GO TO 770
C
C     NO. CHECK FOR A B.
C
      IF(ITEST.EQ.1HB) GO TO 930
C
C     NO. CHECK FOR AN L OR AN R.
C
```

```
      IF(ITEST.EQ.1HL) GO TO 770
      IF(ITEST.EQ.1HR) GO TO 770
C
C     NO. CHECK FOR A D.
C
      IF(ITEST.EQ.1HD) GO TO 1130
C
C     NO. CHECK FOR A LEFT PARENTHESIS.
C
      IF(ITEST.EQ.1H() GO TO 720
C
C     NO. CHECK FOR A RIGHT PARENTHESIS.
C
      IF(ITEST.EQ.1H)) GO TO 750
C
C     LEGAL CHARACTER NOT FOUND. FATAL ERROR.
C
  440 CONTINUE
      CALL MESSOT(12)
      IEFLAG=12
      RETURN
C
C     **************************************************************
C     CHARACTER WAS A NUMBER
C     **************************************************************
C
  450 CONTINUE
C
C     SEE IF A DELIMITER WAS EXPECTED. IF SO,FATAL ERROR.
C
      IF(DELIM) GO TO 260
C
C     RESET ISOLATED PERIOD FLAG
C
      PERIOD=.FALSE.
C
C     UPDATE THE VALUE OF THE NUMERIC FIELD
C
      IVSUM=IVSUM*10+ITEMP*INF
C
C     UPDATE THE DIGIT COUNT
C
      IVNUM=IVNUM+1
C
C     UPDATE THE CHARACTER COUNT
C
      IC=IC+1
C
C     SEE IF THIS IS THE THIRTIETH CHARACTER
C
      IF(IC.LT.30) GO TO 460
C
C     YES. FATAL ERROR.
C
      GO TO 1160
C
C     NO. STORE THE CHARACTER AND GO ON TO THE NEXT.
```

```
C
  460 CONTINUE
      IDSTR(IC)=ITEST
      GO TO 420
C
C     -----------------------------------------------------------------
C     CHARACTER WAS A BLANK
C     -----------------------------------------------------------------
C
  470 CONTINUE
C
C     IF THE BEGINNING OF A CONSTANT HAS NOT YET BEEN FOUND,GO ON TO
C         THE NEXT CHARACTER
C
      IF(FIX.AND.IVNUM.EQ.0) GO TO 420
C
C     NOT THE BEGINNING OF A CONSTANT. BLANK IS CONSIDERED FIELD
C         TERMINATOR. TURN ON THE FLAG WHICH SAYS STORE THE CONSTANT.
C
      SAVE=.TRUE.
C
C     SEE IF CURRENT FIELD IS FIXED OR FLOATING
C
      IF(.NOT.FIX) GO TO 490
C
C     FIXED. DO NOT STORE NOW BUT TURN ON DELIMITER FLAG AND CONTINUE
C         CHARACTER SCAN.
C
      DELIM=.TRUE.
      GO TO 420
C
C     ENTRY POINT FOR STORING FLOATING POINT NO. IF FIELD IS TERMINATED
C         BY A , OR A $
C
  480 CONTINUE
      SAVE=.TRUE.
  490 CONTINUE
C
C     CHECK FOR ISOLATED PERIOD
C
      IF(.NOT.PERIOD) GO TO 500
C
C     YES. FATAL ERROR.
C
      CALL MESSOT(29)
      IEFLAG=29
      RETURN
  500 CONTINUE
C
C     NO. SEE IF WE ARE PROCESSING AN EXPONENT.
C
      IF(.NOT.EXPN) GO TO 530
C
C     YES. SEE IF IT IS ZERO.
C
      IF(IVSUM.EQ.0) GO TO 630
C
```

```
C     NO. GIVE IT THE PROPER SIGN.
C
      IVSUM=ISIGN(IVSUM,IVSIGN)
C
C     SEE IF IT IS WITHIN BOUNDS
C
      IF(IVSUM.GT.IEMIN.AND.IVSUM.LT.IEMAX) GO TO 510
C
C     NO. FATAL ERROR.
C
      CALL MESSOT(26)
      IEFLAG=26
      RETURN
  510 CONTINUE
C
C     YES. ADJUST THE NO. BASED ON THE EXPONENT AND PROCEED ON
C         TO THE GENERAL STORING CODE. USE INTERMEDIATE DOUBLE
C         PRECISION IN ALL CASES.
C
      IF(DOBLE) GO TO 520
      XLEFT=XLEFT*DCONST**IVSUM
      GO TO 630
  520 CONTINUE
      DB=DB*DCONST**IVSUM
      GO TO 630
C
C     CODE TO PROCESS FLOATING POINT NO. WITH NO EXPONENT
C
  530 CONTINUE
C
C     CHECK FOR FIELD LONGER THAN 15 CHARACTERS BUT NOT DECLARED
C         DOUBLE PRECISION. IF SO,FATAL ERROR.
C
      IF(IC.GT.NDSP+1.AND..NOT.DOBLE) GO TO 600
C
C     CHECK FOR DOUBLE PRECISION FIELD
C
      IF(.NOT.DOBLE) GO TO 590
C
C     YES. CHECK FOR FIELD LONGER THAN 29 CHARACTERS. IF SO,FATAL ERROR.
C
      IF(IC.GT.NDDP+1) GO TO 1160
C
C     ALL O.K. INITIALIZE FOR EVALUATION. ICP IS THE NO. OF THE FIELD
C         CHARACTER CURRENTLY BEING PROCESSED.
C
      DB=0.
      ICP=0
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     LOOP TO EVALUATE DOUBLE PRECISION CONSTANTS
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     INCREMENT THE CHARACTER COUNT
C
  540 ICP=ICP+1
C
```

```
C      CHECK FOR END OF THE FIELD
C
       IF(ICP.GT.IC) GO TO 580
C
C      NO. PICK UP THE NEXT CHARACTER.
C
       ITEST=IDSTR(ICP)
C
C      SEE IF IT IS A DECIMAL POINT
C
       IF(ITEST.EQ.1H.) GO TO 550
C
C      NO. ISOLATE THE CHARACTER LOW ORDER IN THE WORD.
C
       ITEMP=AND(SHIFT(ITEST,-54B),77B)
C
C      ADD IT INTO THE CONSTANT
C
       DBP=ITEMP-INF
       DB=DB*DCONST+DBP
C
C      CONTINUE LOOPING
C
       GO TO 540
   550 CONTINUE
C
C      A DECIMAL POINT HAS BEEN LOCATED. PREPARE TO EVALUATE THAT PORTION
C          OF THE CONSTANT TO THE RIGHT OF THE DECIMAL. ICPD IS THE NO.
C          OF DIGITS TO THE RIGHT OF THE DECIMAL.
C
       DBP=0.
       ICPD=0
C
C      LOOP TO EVALUATE PORTION TO THE RIGHT OF THE DECIMAL
C
   560 ICP=ICP+1
C
C      CHECK FOR THE END OF THE FIELD
C
       IF(ICP.GT.IC) GO TO 570
C
C      NO. PICK UP THE CHARACTER,ADD IT INTO THE SUM,AND CONTINUE.
C
       ITEST=IDSTR(ICP)
       ITEMP=AND(SHIFT(ITEST,-54),77B)
       DTEMP=ITEMP-INF
       DBP=DBP*DCONST+DTEMP
       ICPD=ICPD+1
       GO TO 560
   570 CONTINUE
C
C      END OF THE FIELD ENCOUNTERED. SEE IF THERE WAS ANYTHING NON-ZERO
C          TO THE RIGHT OF THE DECIMAL.
C
       ITEMPA=DBP-DZERO
       IF(ITEMPA.EQ.0) GO TO 580
C
```

```
C     YES, SHIFT THE DECIMAL TO THE LEFT AND ADD IT INTO THE PORTION
C         OF THE ENTIRE CONSTANT TO THE LEFT OF THE DECIMAL.
C
      DBP=DBP*DCONST**(-ICPD)
      DB=DB+DBP
C
C     NUMBER IS ENTIRELY DECODED. AFFIX THE PROPER SIGN.
C
  580 IF(IVSIGN.LT.0) DB=-DB
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     GO SEE IF THE NO. IS TO BE STORED
C
      GO TO 630
  590 CONTINUE
C
C     NO. IS SINGLE PRECISION. SEE IF IT IS ZERO.
C
      IF(IVSUM.EQ.0) GO TO 620
C
C     NO. SEE IF IT IS TOO LONG.
C
      IF(IVNUM.LE.NDSP) GO TO 610
C
C     YES. FATAL ERROR.
C
  600 CONTINUE
      CALL MESSOT(28)
      IEFLAG=28
      RETURN
  610 CONTINUE
C
C     NO. STORE IT.
C
      XRIGHT=IVSUM
C
C     SHIFT THE DECIMAL POINT OF THE RIGHTHAND SIDE TO THE FAR LEFT
C
      XRIGHT=XRIGHT*10.**(-IVNUM)
C
C     ADD THE LEFT AND RIGHT PARTS
C
      XLEFT=XLEFT+XRIGHT
  620 CONTINUE
C
C     AFFIX THE SIGN
C
      IF(IVSIGN.LT.0) XLEFT=-XLEFT
C
C     IF DOUBLE PRECISION,CONVERT TO THAT FORM
C
      IF(DOBLE) DB=XLEFT
  630 CONTINUE
C
C     DECODING FOR BOTH SINGLE AND DOUBLE PRECISION FINISHED.
C         REINITIALIZE.
```

```
      C
              IVNUM=0
              IVSUM=0
              IVSIGN=1
      C
      C       IF THE SAVE FLAG IS NOT SET,GO BACK AND EXAMINE THE NEXT
      C           CHARACTER
      C
              IF(.NOT.SAVE) GO TO 420
      C
      C       SEE IF A COMPLEX CONSTANT IS BEING PROCESSED
      C
              IF(.NOT.CMPLEX) GO TO 670
      C
      C       YES. SEE IF THIS IS THE FIRST OR SECOND PART.
      C
              IF(CFRST) GO TO 660
      C
      C       SECOND PART. SEE IF THE SECOND PART HAS ALREADY BEEN STORED.
      C
              IF(.NOT.CDONE) GO TO 650
      C
      C       YES. FATAL ERROR.
      C
          640 CONTINUE
              CALL MESSOT(42)
              IEFLAG=42
              RETURN
      C
      C       NO. SAVE THE SECOND PART AND SET THE FLAG.
      C
          650 CONTINUE
              CIMAG=XLEFT
              CDONE=.TRUE.
              GO TO 670
          660 CONTINUE
      C
      C       FIRST PART. CHECK FOR NO SECOND PART. IF SO,FATAL ERROR.
      C
              IF(CPFND) GO TO 640
      C
      C       SET FLAG TO INDICATE SECOND PART WILL BE PROCESSED NEXT
      C
              CFRST=.FALSE.
      C
      C       SAVE THE FIRST PART
      C
              CREAL=XLEFT
      C
      C       IF THE LAST CHARACTER FOUND WAS NOT A COMMA,TURN ON THE
      C           DELIMITER-EXPECTED FLAG
      C
              IF(ITEST.NE.1H,) DELIM=.TRUE.
      C
      C       GO BACK AND BEGIN PROCESSING THE SECOND PART
      C
              GO TO 410
```

```
  670 CONTINUE
C
C     READY TO STORE CONSTANT. IF IT IS NOT DOUBLE PRECISION OR COMPLEX,
C         SHIFT IT INTO STORAGE KNOWN TO THE ROUTINE STORE.
C
      IF(.NOT.DOBLE.AND..NOT.CMPLEX) XSUM=XLEFT
C
C     STORE THE CONSTANT
C
      CALL STORE(TABLE,IEFLAG)
C
C     IF ANY ERRORS,QUIT RIGHT HERE
C
      IF(IEFLAG.NE.0) RETURN
C
C     IF THE CLOSING $ HAS BEEN FOUND,WE ARE ALL DONE
C
      IF(DONE) RETURN
C
C     IF A COMPLEX CONSTANT IS BEING PROCESSED AND THE CLOSING
C         PARENTHESIS HAS NOT YET BEEN FOUND,CONTINUE ON TO NEXT
C         CHARACTER
C
      IF(CMPLEX.AND..NOT.CPFND) GO TO 420
C
C     IF THE LAST CHARACTER FOUND WAS NOT A COMMA,TURN ON THE
C         DELIMITER FLAG
C
      IF(ITEST.NE.1H,) DELIM=.TRUE.
C
C     GO BACK AND HUNT FOR THE BEGINNING OF ANOTHER VARIABLE NAME
C         OR CONSTANT
C
      GO TO 90
C
C     ******************************************************************
C     CHARACTER IS AN ASTERISK
C     ******************************************************************
C
  680 CONTINUE
C
C     MUST HAVE BEEN PRECEDED BY A FIXED POINT NO. IF NOT,FATAL ERROR.
C
      IF(FIX) GO TO 690
      CALL MESSOT(17)
      IEFLAG=17
      RETURN
  690 CONTINUE
C
C     SEE IF AN ASTERISK HAS PREVIOUSLY BEEN ENCOUNTERED. IF SO,
C         FATAL ERROR.
C
      IF(.NOT.MULT) GO TO 700
      CALL MESSOT(18)
      IEFLAG=18
      RETURN
  700 CONTINUE
```

```
C
C     IF PROCESSING OF A COMPLEX NO. HAS BEGUN,FATAL ERROR
C
      IF(.NOT.CMPLEX) GO TO 710
      CALL MESSOT(19)
      IEFLAG=19
      RETURN
  710 CONTINUE
C
C     ALL CONDITIONS O.K. TURN ON THE FLAG,SAVE THE MULTIPLIER,
C         REINITIALIZE,AND GO ON TO THE NEXT CHARACTER.
C
      MULT=.TRUE.
      MULTSV=IVSUM
      IVNUM=0
      IVSUM=0
      IVSIGN=1
      DELIM=.FALSE.
      GO TO 420
C
C     *********************************************************************
C     CHARACTER IS A LEFT PARENTHESIS
C     *********************************************************************
C
  720 CONTINUE
C
C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
C
      IF(DELIM) GO TO 260
C
C     SEE IF ANY OTHER FIELD CHARACTERS HAVE BEEN READ. IF SO,FATAL
C         ERROR.
C
      IF(IVNUM.EQ.0) GO TO 730
      CALL MESSOT(20)
      IEFLAG=20
      RETURN
  730 CONTINUE
C
C     SEE IF A SIGN HAS BEEN ENCOUNTERED. IF SO,FATAL ERROR.
C
      IF(NOSGN.EQ.0) GO TO 740
      CALL MESSOT(39)
      IEFLAG=39
      RETURN
  740 CONTINUE
C
C     EVERYTHING CHECKS. TURN ON THE COMPLEX FLAG,INDICATE THAT THE
C         FIRST HALF OF THE CONSTANT IS BEING PROCESSED,AND GO ON TO
C         THE NEXT CHARACTER.
C
      CMPLEX=.TRUE.
      CFRST=.TRUE.
      GO TO 420
C
C     *********************************************************************
C     CHARACTER IS A RIGHT PARENTHESIS
```

```
C     ******************************************************************
C
  750 CONTINUE
C
C     IF NOT PROCESSING A COMPLEX CONSTANT,FATAL ERROR
C
      IF(.NOT.CMPLEX) GO TO 440
C
C     IF A RIGHT PARENTHESIS HAS PREVIOUSLY BEEN FOUND,FATAL ERROR
C
      IF(CPFND) GO TO 440
C
C     SEE IF THE COMPLEX CONSTANT HAS ALREADY BEEN STORED
C
      IF(.NOT.CDONE) GO TO 760
C
C     YES. TURN OFF THE COMPLEX FLAG,TURN ON THE DELIMITER-EXPECTED
C         FLAG,AND GO LOOK FOR THE BEGINNING OF ANOTHER VARIABLE
C         OR CONSTANT.
C
      CMPLEX=.FALSE.
      DELIM=.TRUE.
      GO TO 90
  760 CONTINUE
C
C     IF STILL PROCESSING THE FIRST PART,OR EXPECTING A DELIMITER,
C         OR WORKING ON A FIX POINT PART,FATAL ERROR
C
      IF(CFRST.OR.DELIM.OR.FIX) GO TO 640
C
C     ALL CHECKS PASSED. INDICATE CLOSING PARENTHESIS FOUND AND GO
C         STORE THE RESULTS.
C
      CPFND=.TRUE.
      GO TO 480
C
C     ******************************************************************
C     CHARACTER WAS AN H,AN L,OR AN R
C     ******************************************************************
C
  770 CONTINUE
C
C     IF A DELIMITER IS EXPECTED,FATAL ERROR
C
      IF(DELIM) GO TO 260
C
C     IF PROCESSING A FLOATING POINT NO. OR A COMPLEX CONSTANT,
C         FATAL ERROR
C
      IF(FIX.AND..NOT.CMPLEX) GO TO 780
      CALL MESSOT(34)
      IEFLAG=34
      RETURN
  780 CONTINUE
C
C     IF NO CHARACTER COUNT HAS BEEN PREVIOUSLY FOUND,FATAL ERROR
C
```

```
      IF(IVNUM.NE.0) GO TO 790
      CALL MESSOT(35)
      IEFLAG=35
      RETURN
  790 CONTINUE
C
C     IF CHARACTER COUNT IS MORE THAN 10,A MULTIPLIER CANNOT BE USED.
C         IF SO,FATAL ERROR.
C
      IF(.NOT.MULT.OR.IVNUM.LE.10) GO TO 800
      CALL MESSOT(36)
      IEFLAG=36
      RETURN
  800 CONTINUE
C
C     EVERYTHING CHECKS. TURN ON THE HOLLERITH FLAG.
C
      HOLLER=.TRUE.
C
C     THE CHARACTER COUNT IS STORED IN IVNUM AND THE TYPE OF FIELD
C         IN ISAVE (H,L,OR R)
C
      IVNUM=IVSUM
      ISAVE=ITEST
C
C     SPECIAL PROCESSING IF MORE THAN A WORDS WORTH OF CHARACTERS
C
      IF(IVNUM.GT.10) GO TO 900
C
C     TEN OR LESS CHARACTERS TO PROCESS
C
  810 CONTINUE
C
C     PICK UP THE REMAINING CHARACTERS AND STORE IN IDSTR
C
      DO 820 I=1,IVNUM
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
      IDSTR(I)=ITEST
  820 CONTINUE
C
C     IF TEN CHARACTERS,NO DISTINCTION BETWEEN H,L,AND R
C
      IF(IVNUM.EQ.10) GO TO 860
C
C     CHECK FOR H
C
      IF(ISAVE.NE.1HH) GO TO 830
C
C     YES. PAD END OF WORD WITH BLANKS.
C
      IPAD=1H
      GO TO 840
  830 CONTINUE
C
C     NO. CHECK FOR R.
C
```

```
      IF(ISAVE.EQ.1HR) GO TO 870
C
C     NO. ASSUME L. PAD END OF WORD WITH ZEROS.
C
      IPAD=0
  840 CONTINUE
      IVP=IVNUM+1
      DO 850 I=IVP,10
  850 IDSTR(I)=IPAD
C
C     UP CHARACTER COUNT TO FULL WORD
C
      IVNUM=10
  860 CONTINUE
C
C     TAKE INDIVIDUAL CHARACTERS AND PUT INTO A SINGLE WORD (IVSUM)
C
      CALL MASH(IDSTR,IVNUM,IVSUM,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     TURN ON THE DELIMITER FLAG,STORE THE WORD,AND GO ON TO NEXT
C         VARIABLE NAME OR CONSTANT
C
      DELIM=.TRUE.
      CALL STORE(TABLE,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
      GO TO 90
C
C     R FIELD. RIGHT JUSTIFY CHARACTERS AND PAD UPPER PART WITH ZEROS.
C
  870 CONTINUE
      IDIFF=10-IVNUM
      II=11
      IIP=IVNUM+1
      DO 880 I=1,IVNUM
      II=II-1
      IIP=IIP-1
  880 IDSTR(II)=IDSTR(IIP)
      DO 890 I=1,IDIFF
  890 IDSTR(I)=0
      IVNUM=10
C
C     PUT THE CHARACTERS INTO ONE WORD AND STORE IT
C
      GO TO 860
C
C     FIELD IS LONGER THAN 10 CHARACTERS
C
  900 CONTINUE
C
C     CHECK FOR R FIELD. IF SO,FATAL ERROR.
C
      IF(ITEST.NE.1HR) GO TO 910
      CALL MESSOT(37)
      IEFLAG=37
      RETURN
  910 CONTINUE
```

```
C
C     PICK UP A WORDS WORTH OF CHARACTERS
C
      DO 920 I=1,10
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
  920 IDSTR(I)=ITEST
C
C     PUT THEM INTO ONE WORD AND STORE THEM
C
      CALL MASH(IDSTR,10,IVSUM,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
      CALL STORE(TABLE,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     REDUCE THE CHARACTER COUNT BY 10 AND SEE IF IT IS NOW LE.10.
C         IF SO,GO FINISH PROCESSING THE FIELD. IF NOT,GO BACK AND
C         PICK UP THE NEXT 10 CHARACTERS.
C
      IVNUM=IVNUM-10
      IF(IVNUM.LE.10) GO TO 810
      GO TO 910
C
C     ******************************************************************
C     CHARACTER IS A B
C     ******************************************************************
C
  930 CONTINUE
C
C     INSURE THAT PREVIOUS FIELD WAS FIXED POINT. IF NOT,FATAL ERROR.
C
      IF(FIX) GO TO 940
      CALL MESSOT(21)
      IEFLAG=21
      RETURN
  940 CONTINUE
C
C     SEE IF PREVIOUS FIELD WAS ZERO. IF SO,NO CONVERSION NECESSARY.
C
      IF(IVNUM.EQ.0) GO TO 970
C
C     SEE IF PREVIOUS FIELD WAS 20 CHAR OR LESS. IF NOT,FATAL ERROR.
C
      IF(IVNUM.LE.20) GO TO 950
      CALL MESSOT(40)
      IEFLAG=40
      RETURN
  950 CONTINUE
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     LOOP TO DECODE OCTAL NUMBERS
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     IVSUM WILL BE THE RESULTING NO.
C     ISHFT IS THE NO. OF BITS TO SHIFT THE CURRENT DIGIT TO THE LEFT
C         BEFORE OR-ING IT INTO IVSUM
C
```

```
      IVSUM=0
      ISHFT=3*IVNUM
      DO 960 I=1,IVNUM
C
C     ISOLATE THE CHARACTER LOW ORDER IN TEMPORARY STORAGE
C
      ITEMP=AND(SHIFT(IDSTR(I),-54),77B)
C
C     MAKE SURE IT IS NOT AN 8 OR A 9. IF SO,FATAL ERROR.
C
      IF(ITEMP.GT.INF+7) GO TO 980
C
C     CONVERT,SHIFT,AND OR
C
      ISHFT=ISHFT-3
      ITEMP=SHIFT(ITEMP-INF,ISHFT)
      IVSUM=OR(IVSUM,ITEMP)
  960 CONTINUE
  970 CONTINUE
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     TURN ON THE DELIMITER FLAG AND GO STORE THE CONVERTED CONSTANT
C
      DELIM=.TRUE.
      GO TO 1070
  980 CONTINUE
C
C     ILLEGAL CHARACTER IN OCTAL CONSTANT
C
      CALL MESSOT(41)
      IEFLAG=41
      RETURN
C
C     --------------------------------------------------------------
C     CHARACTER IS + OR -
C     --------------------------------------------------------------
C
  990 CONTINUE
C
C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
C
      IF(DELIM) GO TO 260
C
C     IF A SIGN HAS ALREADY BEEN FOUND,FATAL ERROR
C
      IF(NOSGN.NE.0) GO TO 1000
C
C     SEE IF FIXED POINT FLAG IS STILL SET
C
      IF(.NOT.FIX) GO TO 1020
C
C     YES, SIGN BETTER BE FIRST CHARACTER IN THE FIELD OR FATAL ERROR.
C
      IF(IVNUM.EQ.0) GO TO 1010
 1000 CONTINUE
      CALL MESSOT(22)
```

```
      IEFLAG=22
      RETURN
 1010 CONTINUE
C
C     SET IVSIGN BASED ON + OR -. PLUS IS THE DEFAULT.
C
      IF(ITEST.EQ.1H-) IVSIGN=-1
C
C     INDICATE A SIGN HAS BEEN FOUND AND GO LOOK AT THE NEXT
C         CHARACTER
C
      NOSGN=1
      GO TO 420
 1020 CONTINUE
C
C     SIGN FOUND BUT FLOATING POINT NO. BEING PROCESSED. EXPONENT FLAG
C         BETTER BE SET OR FATAL ERROR.
C
      IF(FXPN) GO TO 1040
 1030 CONTINUE
      CALL MESSOT(27)
      IEFLAG=27
      RETURN
 1040 CONTINUE
C
C     EXPONENT FLAG SET. SEE IF ANY CHARACTERS HAVE BEEN ENCOUNTERED
C         AFTER THE E OR D. IF SO,FATAL ERROR.
C
      IF(IVNUM.NE.0) GO TO 1030
C
C     ALL O.K. GO SET THE SIGN AND CONTINUE.
C
      GO TO 1010
C
C     ------------------------------------------------------------------
C     CHARACTER IS A COMMA
C     ------------------------------------------------------------------
C
 1050 CONTINUE
C
C     SEE IF THE DELIMITER-EXPECTED FLAG IS SET
C
      IF(DELIM) GO TO 1090
C
C     NO. SEE IF THE FIXED POINT FLAG IS SET. IF NOT,COMMA SIGNALS
C         TERMINATION OF FLOATING POINT FIELD. GO FINISH DECODING THE
C         NO. AND STORE IT.
C
      IF(.NOT.FIX) GO TO 480
 1060 CONTINUE
C
C     FIXED POINT FLAG IS SET. CHECK FOR MORE THAN 14 CHARACTERS IN THE
C         FIELD. IF SO,FATAL ERROR.
C
      IF(IVNUM.GT.NDSP) GO TO 600
C
C     CHECK FOR NO CHARACTERS FOUND. IF SO,FATAL ERROR.
```

```
C
      IF(IVNUM.NE.0) GO TO 1070
      CALL MESSOT(30)
      IEFLAG=30
      RETURN
 1070 CONTINUE
C
C     SEE IF COMPLEX FLAG IS SET. IF SO,INTEGER FIELD IN COMPLEX
C         CONSTANT - FATAL ERROR.
C
      IF(CMPLEX) GO TO 640
C
C     ALL IS O.K. AFFIX THE SIGN AND STORE THE CONSTANT.
C
      IF(IVSIGN.LT.0) IVSUM=-IVSUM
      CALL STORE(TABLE,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     IF DOLLAR SIGN WAS LAST CHARACTER ENCOUNTERED,ALL DONE. IF NOT,
C         GO ON AND LOOK FOR BEGINNING OF NEXT CONSTANT OR VARIABLE
C         NAME.
C
      IF(DONE) RETURN
      GO TO 90
C
C     ******************************************************************
C     CHARACTER IS $
C     ******************************************************************
C
 1080 CONTINUE
C
C     SET THE DONE FLAG. IF FLOATING POINT NO. IS BEING PROCESSED,
C         VIEW $ AS FIELD TERMINATOR AND GO FINISH PROCESSING THE NO.
C         IF FIXED POINT,GO PERFORM CHECKS AND STORE THE CONSTANT.
C
      DONE=.TRUE.
      IF(.NOT.FIX) GO TO 480
      GO TO 1060
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     JUMP TO HERE IF DELIMITER EXPECTED AND FOUND
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
 1090 CONTINUE
C
C     RESET THE DELIMITER-EXPECTED FLAG
C
      DELIM=.FALSE.
C
C     SEE IF COMPLEX CONSTANT IS BEING PROCESSED. IF NOT,ONE CAN ONLY
C         GET TO HERE WHILE PROCESSING A FIXED POINT NO. GO STORE IT.
C
      IF(.NOT.CMPLEX) GO TO 1060
C
C     COMPLEX CONSTANT BEING PROCESSED. IF CHARACTERS FOUND BEFORE
C         DELIMITER,FATAL ERROR.
C
```

```
      IF(IVNUM.NE.0) GO TO 640
C
C     EVERYTHING O.K. GO ON AND LOOK AT NEXT CHARACTER.
C
      GO TO 420
C
C     ----------------------------------------------------------------
C     CHARACTER IS AN E
C     ----------------------------------------------------------------
C
 1100 CONTINUE
C
C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
C
      IF(DELIM) GO TO 260
C
C     SEE IF MORE THAN 15 CHARACTERS HAVE BEEN FOUND IN THE PRECEDING
C         FIELD. IF SO,E SHOULD HAVE BEEN D. FATAL ERROR.
C
      IF(IC.GT.NDSP+1) GO TO 600
 1110 CONTINUE
C
C     ALL O.K. TURN ON THE EXPONENT FLAG AND RESET THE SIGN COUNT.
C
      EXPN=.TRUE.
      NOSGN=0
C
C     SEE IF THE FIELD CURRENTLY BEING PROCESSED IS FIXED POINT
C
      IF(.NOT.FIX) GO TO 1120
C
C     YES. SEE IF ANY CHARACTERS HAVE PREVIOUSLY BEEN FOUND.
C
      IF(IVNUM.NE.0) GO TO 1150
C
C     NO. DEFAULT TO 1. (AND USE A SIGN IF ONE WAS FOUND)
C
      IVNUM=1
      IVSUM=1
      IF(IVSIGN.LT.0) IVSUM=-IVSUM
C
C     RESET THE SIGN
C
      IVSIGN=1
C
C     GO FINISH PROCESSING PART TO LEFT OF THE EXPONENT
C
      GO TO 1150
 1120 CONTINUE
C
C     FLOATING POINT FIELD. GO FINISH PROCESSING PART TO THE LEFT OF
C         THE EXPONENT.
C
      SAVE=.FALSE.
      GO TO 530
C
C     ----------------------------------------------------------------
```

```
C      CHARACTER IS A D
C      ------------------------------------------------------------------
C
 1130 CONTINUE
C
C      IF A DELIMITER WAS EXPECTED,FATAL ERROR
C
       IF(DELIM) GO TO 260
C
C      SEE IF COMPLEX CONSTANT IS BEING PROCESSED. IF SO,FATAL ERROR.
C
       IF(CMPLEX) GO TO 640
C
C      EVERYTHING CHECKS. SET THE DOUBLE PRECISION FLAG AND GO FINISH
C          INITIALIZING FOR THE EXPONENT.
C
       DOBLE=.TRUE.
       GO TO 1110
C
C      ------------------------------------------------------------------
C      CHARACTER IS A PERIOD
C      ------------------------------------------------------------------
C
 1140 CONTINUE
C
C      SEE IF DELIMITER WAS EXPECTED. IF SO,FATAL ERROR.
C
       IF(DELIM) GO TO 260
C
C      IF FLOATING POINT FLAG IS ALREADY SET,FATAL ERROR
C
       IF(FIX) GO TO 1150
       CALL MESSOT(25)
       IEFLAG=25
       RETURN
 1150 CONTINUE
C
C      IF MORE THAN 28 CHARACTERS PREVIOUSLY FOUND,FATAL ERROR
C
       IF(IVNUM.GT.NDDP) GO TO 1160
C
C      IF THE PERIOD IS THE FIRST CHARACTER IN THE FIELD,SET THE
C          ISOLATED PERIOD FLAG
C
       IF(IVNUM.EQ.0) PERIOD=.TRUE.
C
C      NO PROBLEMS. STORE THE PART TO THE LEFT OF THE PERIOD AS A
C          FLOATING POINT NO.
C
       XLEFT=IVSUM
       IVSUM=0
C
C      RESET THE FIELD FLAGS
C
       IVNUM=0
C
C      TURN ON THE FLOATING POINT FLAG
```

```
C
      FIX=.FALSE.
C
C     STORE THE CHARACTER FOR LATER DOUBLE PRECISION PROCESSING IF
C         NECESSARY
C
      IC=IC+1
      IDSTR(IC)=1H.
C
C     IF THE DOUBLE PRECISION FLAG IS ON,STORE PART TO LEFT OF DECIMAL
C         IN DOUBLE PRECISION FORM
C
      IF(DOBLE) DB=XLEFT
C
C     GO ON AND LOOK AT THE NEXT CHARACTER
C
      GO TO 420
C
C     FATAL ERROR, TOO MANY DIGITS IN CONSTANT.
C
 1160 CONTINUE
      CALL MESSOT(38)
      IEFLAG=38
      RETURN
C
C     ****************************************************************
C     CHARACTER IS T OR F
C     ****************************************************************
C
 1170 CONTINUE
C
C     IF DELIMITER IS EXPECTED,FATAL ERROR
C
      IF(DELIM) GO TO 260
C
C     A PERIOD SHOULD HAVE PRECEDED EITHER T OR F. IF NOT,FATAL ERROR.
C
      IF(PERIOD) GO TO 1180
      CALL MESSOT(31)
      IEFLAG=31
      RETURN
 1180 CONTINUE
C
C     SAVE THE CHARACTER AND GO GET THE NEXT ONE
C
      ISAVE=ITEST
      CALL NXTCOL(IF,IEFLAG)
      IF(IEFLAG.NE.0) RETURN
C
C     SEE IF IT WAS A PERIOD
C
      IF(ITEST.NE.1H.) GO TO 1210
C
C     YES, SHORTENED FORM OF LOGICAL CONSTANT BEING USED
C
 1190 CONTINUE
C
```

```
C      SET THE DELIMITER-EXPECTED FLAG
C
       DELIM=.TRUE.
C
C      PICK UP EITHER A TRUE OR FALSE CONSTANT
C
       IVSUM=LCONT
       IF(ISAVE.EQ.1HF) IVSUM=LCONF
C
C      IF THE CONSTANT IS SIGNED,INTERPRET - TO MEAN COMPLEMENT
C
       IF(IVSIGN.GE.0) GO TO 1200
       IF(IVSUM.EQ.LCONF) IVTEMP=LCONT
       IF(IVSUM.EQ.LCONT) IVTEMP=LCONF
       IVSUM=IVTEMP
  1200 CONTINUE
C
C      STORE THE CONSTANT AND PROCEED ON TO NEXT VARIABLE OR CONSTANT
C
       CALL STORE(TABLE,IEFLAG)
       IF(IEFLAG.NE.0) RETURN
       GO TO 90
  1210 CONTINUE
C
C      LONG FORM OF LOGICAL CONSTANT BEING USED. CHECK FOR BEGINNING
C          OF TRUE OR FALSE CONSTANT.
C
       IF(ISAVE.EQ.1HF) GO TO 1250
C
C      TRUE. NEXT CHARACTER BETTER BE R OR FATAL ERROR.
C
       IF(ITEST.EQ.1HR) GO TO 1230
  1220 CONTINUE
       CALL MESSOT(10)
       IEFLAG=10
       RETURN
  1230 CONTINUE
C
C      TR FOUND. GET NEXT CHARACTER AND CHECK FOR U. IF NOT,FATAL ERROR.
C
       CALL NXTCOL(IF,IEFLAG)
       IF(IEFLAG.NE.0) RETURN
       IF(ITEST.NE.1HU) GO TO 1220
  1240 CONTINUE
C
C      ALL BUT E. SUCCESSFULLY FOUND. CHECK FOR THESE. IF NOT FOUND,
C          FATAL ERROR. IF FOUND,GO SET UP AND STORE CONSTANT.
C
       CALL NXTCOL(IF,IEFLAG)
       IF(IEFLAG.NE.0) RETURN
       IF(ITEST.NE.1HE) GO TO 1220
       CALL NXTCOL(IF,IEFLAG)
       IF(IEFLAG.NE.0) RETURN
       IF(ITEST.NE.1H.) GO TO 1220
       GO TO 1190
  1250 CONTINUE
C
```

```
C       FALSE BEING PROCESSED, CHECK FOR ALS,
C
        IF(ITEST.NE.1HA) GO TO 1220
        CALL NXTCOL(IF,IEFLAG)
        IF(IEFLAG.NE.0) RETURN
        IF(ITEST.NE.1HL) GO TO 1220
        CALL NXTCOL(IF,IEFLAG)
        IF(IEFLAG.NE.0) RETURN
        IF(ITEST.NE.1HS) GO TO 1220
C
C       ALL SUCCESSFULLY FOUND, GO CHECK REST OF CONSTANT,
C
        GO TO 1240
        END
```

```
        SUBROUTINE MESSOT(IEFLAG)
C
C       ROUTINE TO DO PRINTING FOR NAMLST
C
C       IEFLAG IS THE ERROR FLAG (SEE NAMLST)
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1974
C
        COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
       1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
        COMMON/IOTAB/NF,IFTAB
        COMMON/JOTAB/NFE,IFTABE
        INTEGER GETIT,SHIFT,OR,AND,COMP
        DIMENSION ILINE(8)
        DATA IPT/1R;/
C
C       LOOP OVER ALL OUTPUT FILES
C
        IF(IEFLAG.NE.1) GO TO 2
        NFP=NF
        IFT=IFTAB
        GO TO 3
      2 CONTINUE
        NFP=NFE
        IFT=IFTABE
      3 CONTINUE
        IF(NFP.EQ.0) NFP=1
        DO 500 I=1,NFP
C
C       IF NO. OF FILES IS 0,GO DEFAULT TO OUT
C
        IF(IFT.NE.0) GO TO 10
        IPX=6
        GO TO 20
     10 CONTINUE
C
```

```
C     PICK UP THE NEXT FILE NAME
C
      ITST=IFT+I-1
      IPX=GETIT(ITST)
   20 CONTINUE
C
C     PRINT A ROW OF ASTERISKS UNLESS JUST PRINTING THE CARD
C
      IF(IEFLAG.EQ.1) GO TO 30
      WRITE(IPX,970)
   30 CONTINUE
C
C     PRINT THE CARD ITSELF
C
      WRITE(IPX,510) ICHAR
C
C     ALL DONE IF JUST PRINTING THE CARD
C
      IF(IEFLAG.EQ.1) GO TO 500
C
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C     SET UP POINTER TO AREA OF CARD CAUSING ERROR
C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
C
C     BLANK OUT THE POINTER LINE
C
      DO 40 J=1,8
   40 ILINE(J)=1H
C
C     APPLY 80 AS UPPER BOUND ON CARD COLUMN NO.
C
      ISUBX=ISUB
      IF(ISUBX.GT.80) ISUBX=80
C
C     DETERMINE WORD COUNT (L) AND CHARACTER COUNT WITHIN WORD (K)
C         OF CARD WORD CONTAINING THE ERROR
C
      K=MOD(ISUBX,10)
      L=ISUBX/10
      IF(K.NE.0) L=L+1
      IF(K.EQ.0) K=10
C
C     SHIFT THE POINTER TO THE PROPER COLUMN POSITION
C
      KX=10-K
      IKX=6*KX
      IC=SHIFT(IPT,IKX)
C
C     MASK OUT THE PROPER POSITION IN THE LINE AND INSERT THE POINTER
C
      IM=SHIFT(77B,IKX)
      ILINE(L)=OR(IC,AND(ILINE(L),COMP(IM)))
C
C     PRINT THE POINTER LINE
C
      WRITE(IPX,520) ILINE
C
```

```
C       GO PRINT THE PROPER ERROR MESSAGE
C
        GO TO (500,50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,
       1 200,210,220,230,240,250,260,270,280,290,300,310,320,330,340,350,
       2 360,370,380,390,400,410,420,430,440,450,460,470,480),IEFLAG
   50 WRITE(IPX,530)
      GO TO 490
   60 WRITE(IPX,540)
      GO TO 490
   70 WRITE(IPX,550)
      GO TO 490
   80 WRITE(IPX,560)
      GO TO 490
   90 WRITE(IPX,570)
      GO TO 490
  100 WRITE(IPX,580)
      GO TO 490
  110 WRITE(IPX,590)
      GO TO 490
  120 WRITE(IPX,600)
      GO TO 490
  130 WRITE(IPX,610)
      GO TO 490
  140 WRITE(IPX,620)
      GO TO 490
  150 WRITE(IPX,630)
      GO TO 490
  160 WRITE(IPX,640)
      GO TO 490
  170 WRITE(IPX,650)
      GO TO 490
  180 WRITE(IPX,660)
      GO TO 490
  190 WRITE(IPX,670)
      GO TO 490
  200 WRITE(IPX,680)
      GO TO 490
  210 WRITE(IPX,690)
      GO TO 490
  220 WRITE(IPX,700)
      GO TO 490
  230 WRITE(IPX,710)
      GO TO 490
  240 WRITE(IPX,720)
      GO TO 490
  250 WRITE(IPX,730)
      GO TO 490
  260 WRITE(IPX,740)
      GO TO 490
  270 WRITE(IPX,750)
      GO TO 490
  280 WRITE(IPX,760)
      GO TO 490
  290 WRITE(IPX,770)
      GO TO 490
  300 WRITE(IPX,780)
      GO TO 490
```

```
  310 WRITE(IPX,790)
      GO TO 490
  320 WRITE(IPX,800)
      GO TO 490
  330 WRITE(IPX,810)
      GO TO 490
  340 WRITE(IPX,820)
      GO TO 490
  350 WRITE(IPX,830)
      GO TO 490
  360 WRITE(IPX,840)
      GO TO 490
  370 WRITE(IPX,850)
      GO TO 490
  380 WRITE(IPX,860)
      GO TO 490
  390 WRITE(IPX,870)
      GO TO 490
  400 WRITE(IPX,880)
      GO TO 490
  410 WRITE(IPX,890)
      GO TO 490
  420 WRITE(IPX,900)
      GO TO 490
  430 WRITE(IPX,910)
      GO TO 490
  440 WRITE(IPX,920)
      GO TO 490
  450 WRITE(IPX,930)
      GO TO 490
  460 WRITE(IPX,940)
      GO TO 490
  470 WRITE(IPX,950)
      GO TO 490
  480 WRITE(IPX,960)
  490 CONTINUE
C
C     PRINT ANOTHER ROW OF ASTERISKS
C
      WRITE(IPX,970)
  500 CONTINUE
      RETURN
C
  510 FORMAT(1H ,80A1)
  520 FORMAT(1H ,8A10)
  530 FORMAT(1H ,41HMORE THAN SIX CHARACTERS IN NAMELIST NAME)
  540 FORMAT(1H ,45HMORE THAN SIX CHARACTERS IN NAMELIST VARIABLE)
  550 FORMAT(1H ,39HNAMELIST VARIABLE NOT IN NAMELIST TABLE)
  560 FORMAT(1H ,50HILLEGAL CHARACTER FOLLOWING NAMELIST VARIABLE NAME)
  570 FORMAT(1H ,30HILLEGAL CHARACTER IN SUBSCRIPT)
  580 FORMAT(1H ,18HDELIMITER EXPECTED)
  590 FORMAT(1H ,20HUNEXPECTED DELIMITER)
  600 FORMAT(1H ,20H= EXPECTED,NOT FOUND)
  610 FORMAT(1H ,24HILLEGAL LOGICAL VARIABLE)
  620 FORMAT(1H ,43HNAMELIST NAME NOT FOUND IMMEDIATELY AFTER $)
  630 FORMAT(1H ,34HILLEGAL CHARACTER IN THIS POSITION)
  640 FORMAT(1H ,44HFIRST CHARACTER OF NAMELIST VARIABLE ILLEGAL)
```

```
  650 FORMAT(1H ,19HTOO MANY SUBSCRIPTS)
  660 FORMAT(1H ,49HATTEMPT TO USE UNDIMENSIONED VARIABLE AS AN ARRAY)
  670 FORMAT(1H ,37HTOO MANY SUBSCRIPTS FOR THIS VARIABLE)
  680 FORMAT(1H ,28HMULTIPLIER IS NOT AN INTEGER)
  690 FORMAT(1H ,24HMORE THAN ONE MULTIPLIER)
  700 FORMAT(1H ,34HMULTIPLIER WITHIN COMPLEX CONSTANT)
  710 FORMAT(1H ,25HIMPROPER LEFT PARENTHESIS)
  720 FORMAT(1H ,23HIMPROPER OCTAL CONSTANT)
  730 FORMAT(1H ,31HMORE THAN TWO SIGNS IN CONSTANT)
  740 FORMAT(1H ,40HFIRST CHARACTER OF NAMELIST NAME ILLEGAL)
  750 FORMAT(1H ,34HILLEGAL CHARACTER IN NAMELIST NAME)
  760 FORMAT(1H ,48HMORE THAN ONE DECIMAL POINT IN A FLOATING POINT ,
     1 8HCONSTANT)
  770 FORMAT(1H ,21HEXPONENT OUT OF RANGE)
  780 FORMAT(1H ,35HIMPROPER SIGN IN FLOATING POINT NO.)
  790 FORMAT(1H ,48HTOO MANY CHARACTERS IN SINGLE PRECISION CONSTANT)
  800 FORMAT(1H ,15HISOLATED PERIOD)
  810 FORMAT(1H ,19HZERO WIDTH CONSTANT)
  820 FORMAT(1H ,34HILLEGAL CHARACTER FOLLOWING PERIOD)
  830 FORMAT(1H ,14HUNEXPECTED EOF)
  840 FORMAT(1H ,27HUNEXPECTED CARD TERMINATION)
  850 FORMAT(1H ,27HILLEGAL H,L,OR R OCCURRENCE)
  860 FORMAT(1H ,23HHOLLERITH COUNT MISSING)
  870 FORMAT(1H ,50HMULTIPLIER NOT ALLOWED FOR HOLLERITH CONSTANTS OF ,
     1 23HMORE THAN 10 CHARACTERS)
  880 FORMAT(1H ,44HR FIELD CANNOT BE GREATER THAN 10 CHARACTERS)
  890 FORMAT(1H ,17HCONSTANT TOO LONG)
  900 FORMAT(1H ,36HSIGN CANNOT PRECEDE COMPLEX CONSTANT)
  910 FORMAT(1H ,33HTOO MANY DIGITS IN OCTAL CONSTANT)
  920 FORMAT(1H ,50HTHE DIGIT 8 OR 9 CANNOT OCCUR IN AN OCTAL CONSTANT)
  930 FORMAT(1H ,25HIMPROPER COMPLEX CONSTANT)
  940 FORMAT(1H ,19HSUBSCRIPT TOO LARGE)
  950 FORMAT(1H ,41HMORE THAN ONE SIGN IN AN INTEGER CONSTANT)
  960 FORMAT(1H ,13HERROR IN MASH)
  970 FORMAT(1H ,60(1H*))
      END
```

****

```
      SUBROUTINE STORE(TABLE,IEFLAG)
C
C     ROUTINE TO STORE THE CONSTANT LAST DECODED
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
     1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
      LOGICAL MULT,CMPLEX,DIM,DOBLE
      DIMENSION TABLE(3,1)
      INTEGER TABLE,SHIFT,AND,OR,COMP
      DATA CREAL/0./
C
C     GET THE VARIABLE SUBSCRIPT INFORMATION
```

```
C
      IARENT=TABLE(3,IENTRY)
C
C     GET THE ABSOLUTE ADDRESS OF THE VARIABLE
C
      IARADD=TABLE(2,IENTRY)
C
C     ISBSPT IS THE MEMORY OFFSET (+1) FOR THE VARIABLE BEING READ. IT
C         IS 1 UPON THE ENTRY TO STORE FOR A GIVEN VARIABLE NAME. SOME
C         CHECKING IS DONE ONLY UPON THE FIRST ENTRY. SEE IF THIS IS SO.
C
      IF(ISBSPT.LE.1) GO TO 10
C
C     NOT THE FIRST ENTRY. SEE IF THE VARIABLE IS SUBSCRIPTED, IF NOT,
C         FATAL ERROR FOR TRYING TO STORE MORE THAN ONE DATA ELEMENT
C         INTO A NON-SUBSCRIPTED VARIABLE.
C
      IF(IARENT.EQ.0) GO TO 60
C
C     EVERYTHING CHECKS. CONTINUE ON WITH THE STORE.
C
      GO TO 20
   10 CONTINUE
C
C     FIRST ENTRY. SEE IF A MULTIPLIER IS IN EFFECT OR VARIABLE HAS
C         SUBSCRIPTS.
C
      IF(MULT.OR.DIM) GO TO 50
C
C     NEITHER CONDITION HOLDS. CONTINUE ON WITH THE STORE.
C
   20 CONTINUE
C
C     SET THE MULTIPLIER COUNT. IF NO MULTIPLIER,USE 1.
C
      IF(.NOT.MULT) MULTSV=1
C
C     LOOP FOR MULTIPLE STORING
C
      DO 40 I=1,MULTSV
C
C     GET THE ACTUAL ABSOLUTE ADDRESS OF THE LOCATION INTO WHICH THE
C         STORE IS TO OCCUR
C
      IADD=IARADD+ISBSPT-1
C
C     SPECIAL HANDLING FOR DOUBLE OR COMPLEX
C
      IF(CMPLEX.OR.DOBLE) GO TO 30
C
C     ORDINARY ONE ELEMENT VARIABLE. CARRY OUT THE STORE.
C
      CALL STORIT(IVSUM,IADD)
C
C     INCREMENT THE POINTER AND GO TO THE END OF THE LOOP
C
      ISBSPT=ISBSPT+1
```

```
      GO TO 40
   30 CONTINUE
C
C     EITHER DOUBLE OR COMPLEX. STORE TWO WORDS.
C
      CALL STORIT(CREAL,IADD)
      CALL STORIT(CIMAG,IADD+1)
      ISBSPT=ISBSPT+2
   40 CONTINUE
C
C     ALL DONE
C
      RETURN
   50 CONTINUE
C
C     VARIABLE HAS SUBSCRIPTS OR CONSTANT HAS A MULTIPLIER. VARIABLE
C         MUST BE DIMENSIONED. IF NOT,FATAL ERROR.
C
      IF(IARENT.NE.0) GO TO 70
   60 CONTINUE
      CALL MESSOT(15)
      IEFLAG=15
      RETURN
   70 CONTINUE
C
C     IF VARIABLE IS NOT SUBSCRIPTED,ALL IS O.K. CARRY OUT THE STORE.
C
      IF(.NOT.DIM) GO TO 20
C
C     VARIABLE IS SUBSCRIPTED. GET THE TABLE DIMENSION INFORMATION.
C
      ISUB1=AND(IARENT,777777B)
      ISUB2=AND(SHIFT(IARENT,-18),777777B)
      ISUB3=AND(SHIFT(IARENT,-36),777777B)
C
C     CHECK FOR SUBSCRIPTING ERRORS. IS THERE A FOURTH SUBSCRIPT BUT
C         NO THIRD DIMENSION.
C
      IF(NSUB.EQ.4.AND.ISUB3.EQ.0) GO TO 80
C
C     IS THERE A THIRD SUBSCRIPT BUT NO SECOND DIMENSION
C
      IF(NSUB.EQ.3.AND.ISUB2.EQ.0) GO TO 80
C
C     IS THERE A SECOND SUBSCRIPT BUT EITHER NO FIRST DIMENSION OR A
C         FIRST DIMENSION INDICATING A SINGLY-SUBSCRIPTED ARRAY
C
      IF(NSUB.EQ.2.AND.(ISUB1.EQ.1.OR.ISUB1.EQ.0)) GO TO 80
C
C     IS THERE A SINGLE SUBSCRIPT BUT NO SUBSCRIPTING INFORMATION GIVEN
C
      IF(NSUB.EQ.1.AND.ISUB1.EQ.0) GO TO 80
C
C     IF N SUBSCRIPTS ARE GIVEN,DIMENSIONAL INFORMATION IS USED FOR
C         THE FIRST N-1
C
      IF(NSUBV(4).EQ.0) ISUB3=0
```

```
      IF(NSUBV(3).EQ.0) ISUB2=0
      IF(NSUBV(2).EQ.0) ISUB1=0
C
C     CALCULATE THE OFFSET FROM THE SUBSCRIPTING INFORMATION
C
      ISBSPT=NSUBV(1)+ISUB1*(NSUBV(2)-1+ISUB2*(NSUBV(3)-1+ISUB3*(NSUBV(4
     1 )-1)))
C
C     MODIFY THE OFFSET FOR COMPLEX VARIABLES
C
      IF(CMPLEX) ISBSPT=2*ISBSPT-1
C
C     GO BACK AND DO THE STORE
C
      GO TO 20
   80 CONTINUE
C
C     FATAL SUBSCRIPTING ERROR
C
      CALL MESSOT(16)
      IEFLAG=16
      RETURN
      END
```

```
      SUBROUTINE MASH(IDSTR,IVNUM,INAME,IEFLAG)
C
C     ROUTINE TO CONVERT AN ARRAY IDSTR CONTAINING ONE SIX-BIT
C         CHARACTER PER WORD,LEFT-JUSTIFIED,INTO ONE WORD,INAME,
C         CONTAINING THE SAME CHARACTERS LEFT TO RIGHT WITH BLANK FILL
C         AT THE RIGHT END IF NECESSARY. IVNUM CHARACTERS ARE PROCESSED
C         WHERE IVNUM MUST BE LE.10.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      DIMENSION IDSTR(1)
      INTEGER SHIFT,AND,OR,COMP
      DATA MASK/77000000000000000000B/
C
C     CHECK THE CHARACTER COUNT. IF GT.10,FATAL ERROR.
C
      IF(IVNUM.LE.10) GO TO 10
      CALL MESSOT(45)
      IEFLAG=45
      RETURN
   10 CONTINUE
C
C     INITIALIZE THE PACKED WORD
C
      INAME=1H
C
C     LOOP OVER ALL CHARACTERS
```

```
C
      DO 20 I=1,IVNUM
C
C     SHIFT MASK INTO PROPER POSITION
C
      ITEMPM=SHIFT(77B,60-6*I)
C
C     SHIFT CHARACTER INTO PROPER POSITION
C
      ITEMP=SHIFT(IDSTR(I),-6*(I-1))
C
C     DROP CHARACTER INTO INAME, FIRST MASK OUT THE BLANK, THEN ISOLATE
C         THE CHARACTER AND DROP IT INTO THE PREPARED SLOT.
C
   20 INAME=OR(AND(INAME,COMP(ITEMPM)),AND(ITEMP,ITEMPM))
C
C     ALL DONE
C
      RETURN
      END
```

```
      SUBROUTINE NAMPRT(NFX,FTABX)
C
C     ROUTINE TO SET THE FILES ON WHICH THE NAMLST ROUTINE WILL WRITE
C         WHENEVER ITS DOES OUTPUT OTHER THAN ERROR MESSAGES (PS OR CS).
C         THE CALLING SEQUENCE IS    CALL NAMPRT(N,TABLE)     WHERE
C                   N = NO. OF TABLE ENTRIES
C                   TABLE = ARRAY OF FILE NAMES,EITHER IN THE FORM OF
C                             LEFT-JUSTIFIED,ZERO-FILLED HOLLERITH
C                             CONSTANTS OR INTEGERS.
C
C         FOR EXAMPLE,IF ONE WANTED TO WRITE BOTH ON FILM AND PAPER,
C              THE TABLE WOULD APPEAR AS
C                        DATA TABLE/4LFILM,3LOUT/.
C              IF FILM WAS EQUIVALENCED TO FSET12 AND OUT TO FSET6,
C              AN ALTERNATE FORM WOULD BE
C                        DATA TABLE/12,6/
C              AND THE CALL WOULD BE
C                        CALL NAMPRT(2,TABLE).
C
C         IF NAMPRT IS NOT CALLED BEFORE NAMLST,THE LATTER WILL
C              DEFAULT ALL OUTPUT TO OUT.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      COMMON/IOTAB/NF,IFLOC
      INTEGER FTABX(1)
      DATA NF,IFLOC/0,0/
      IF(NFX.LE.0) RETURN
      NF=NFX
      IFLOC=LOCF(FTABX)
```

```
      RETURN
      END
```

```
************************************************************************
```

```
      SUBROUTINE ERRPRT(NFX,FTABX)
C
C     ROUTINE ANALOGOUS TO NAMPRT EXCEPT FOR PRINTING ERROR MESSAGES
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      COMMON/JOTAB/NF,IFLOC
      INTEGER FTABX(1)
      DATA NF,IFLOC/0,0/
      IF(NFX.LE.0) RETURN
      NF=NFX
      IFLOC=LOCF(FTABX)
      RETURN
      END
```

```
************************************************************************
```

```
      SUBROUTINE TABDEF(TABLE,NAME,ITSIZE,IERRT)
C
C     ROUTINE TO INITIALIZE A NAMELIST TABLE
C
C     TABLE - ARRAY WHICH WILL BE USED TO STORE NAMELIST DATA NAMES
C                 AND RELATED INFORMATION, IT MUST BE 3*(N+1) ELEMENTS
C                 LONG WHERE N IS THE NUMBER OF DISTINCT NAMELIST
C                 VARIABLES TO BE READ UNDER THIS NAME.
C     NAME  - NAMELIST NAME,LEFT-JUSTIFIED,BLANK-FILLED HOLLERITH, IF
C                 ONE WERE GOING TO READ CARDS OF THE FORM PSCODEIN .,
C                 NAME=6HCODEIN.
C     ITSIZE - SECOND SUBSCRIPT OF THE TABLE ARRAY
C                 (MAXIMUM NO. OF TABLE ENTRIES+1)
C     IERRT  - VARIABLE SET BY ASSIGNED GO TO AS RETURN POINT IF TABLE
C                 INITIALIZATION ENCOUNTERS AN ERROR CONDITION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      INTEGER TABLE
      DIMENSION TABLE(3,1)
      COMMON/ERRORC/IERRTP
      TABLE(1,1)=NAME
      TABLE(2,1)=ITSIZE
      TABLE(3,1)=0
      IERRTP=IERRT
      RETURN
      END
```

```
      SUBROUTINE TABSET(TABLE,NAME,LOCA,IEFLAG,NSUB,NSUB1,NSUB2,NSUB3)
C
C     ROUTINE TO ADD A VARIABLE TO A NAMELIST TABLE
C
C     TABLE = AN ARRAY WHICH HAS BEEN INITIALIZED BY A CALL TO TABDEF
C     NAME  = NAME OF VARIABLE TO BE ADDED,AS A HOLLERITH CONSTANT
C     LOCA  = THE VARIABLE ITSELF
C     IEFLAG= ERROR INDICATOR
C           = RETURNED ZERO IF NO ERROR WAS FOUND
C           = RETURNED WITH THE CONTENTS OF THE VARIABLE =NAME=
C               LEFT-JUSTIFIED (LEFT 6 CHARACTERS) AND THE ERROR NO.
C               RIGHT-JUSTIFIED IF AN ERROR OCCURRED
C     NSUB  = NUMBER OF SUBSCRIPTS (0,1,2,3,OR 4)
C     NSUB1 = FIRST ARRAY DIMENSION
C     NSUB2 = SECOND ARRAY DIMENSION
C     NSUB3 = THIRD ARRAY DIMENSION
C           = (IF THE NAMELIST VARIABLE HAS N SUBSCRIPTS (N.LE.4),
C                THEN NSUB1,NSUB2,....,NSUBN-1 ARE NEEDED)
C
C     THE ACTUAL FORM OF THE TABLE IS AS FOLLOWS --
C              WORD 1 = THE TABLE NAME (LEFT-JUSTIFIED,BLANK-FILLED
C                                                        HOLLERITH)
C              WORD 2 = UNUSED
C              WORD 3 = NO. OF ENTRIES IN THE TABLE
C         NEXT FOLLOW TRIPLETS OF WORDS FOR EACH NAMELIST VARIABLE =
C              WORD 1 = VARIABLE NAME (SAME AS TABLE NAME)
C              WORD 2 = VARIABLE LOCATION (RIGHT JUSTIFIED,RELATIVE
C                             TO BEGINNING OF CODE FIELD-LENGTH)
C              WORD 3 = SUBSCRIPT INFORMATION
C                     = IF ZERO,THE VARIABLE IS SINGLY SUBSCRIPTED
C                     = IF NON-ZERO,THE WORD IS INTERPRETED AS BEING
C                         MADE UP OF THREE 18 BIT FIELDS, DENOTING
C                         THE RIGHTMOST BIT AS 0, THE THREE FIELDS ARE
C                         BITS 0-17,18-35,AND 36-53,RESPECTIVELY.
C                         THESE THREE FIELDS CONTAIN NSUB1,NSUB2,AND
C                         NSUB3,RESPECTIVELY.
C         THE LENGTH OF THE TABLE SHOULD BE 3*(WORD(3)+1) LOCATIONS.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      INTEGER TABLE,SHIFT,AND,OR,COMP
      DIMENSION TABLE(3,1),FMT(10)
      COMMON/ERRORC/IERRT
      IEFLAG=0
C
C     GET NUMBER OF ENTRIES CURRENTLY IN THE TABLE
C
      NENTRY=TABLE(3,1)
C
C     COMPUTE SUBSCRIPT FOR NEW TABLE ENTRY
C
      ISUB=NENTRY+2
C
C     MAKE SURE THERE IS STILL TABLE SPACE LEFT
```

```
C
      IF(ISUB.LE.TABLE(2,1)) GO TO 10
C
C     NO. FATAL ERROR.
C
      IEFLAG=5
      GO TO 999
   10 CONTINUE
C
C     SET UP TABLE ENTRY
C
      TABLE(1,ISUB)=NAME
      TABLE(2,ISUB)=LOCF(LOCA)
      TABLE(3,ISUB)=0
C
C     COMPOSE SUBSCRIPT WORD IF VARIABLE IS AN ARRAY
C
      IF(NSUB.EQ.0) GO TO 30
C
C     SET THE TABLE BASED UPON HOW MANY SUBSCRIPTS THE ARRAY HAS
C
      IF(NSUB.GT.1) GO TO 20
C
C     ARRAY HAS ONLY ONE SUBSCRIPT. DENOTE THIS BY SETTING SUBSCRIPT
C         FIELD 1 TO 1 AND LEAVING ALL THE REST ZERO.
C
      TABLE(3,ISUB)=1
      GO TO 30
C
C     ARRAY HAS MORE THAN ONE SUBSCRIPT
C
   20 CONTINUE
C
C     SEE IF AT LEAST THE FIRST SUBSCRIPT IS SPECIFIED. IF NOT,ERROR.
C
      IF(NSUB1.LE.0) GO TO 40
C
C     FIRST SUBSCRIPT PRESENT. STORE IT AND SEE IF ARRAY HAS MORE THAN
C         TWO SUBSCRIPTS.
C
      TABLE(3,ISUB)=NSUB1
      IF(NSUB.EQ.2) GO TO 30
C
C     YES. THE SECOND SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR.
C
      IF(NSUB2.LE.0) GO TO 50
C
C     SECOND SUBSCRIPT PRESENT. STORE IT AND SEE IF ARRAY HAS MORE THAN
C         THREE SUBSCRIPTS.
C
      TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB2,18))
      IF(NSUB.EQ.3) GO TO 30
C
C     YES. THE THIRD SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR.
C
      IF(NSUB3.LE.0) GO TO 60
C
```

```
C     THIRD SUBSCRIPT PRESENT, STORE IT AND MAKE SURE THE TOTAL NO.
C         OF SUBSCRIPTS DOES NOT EXCEED FOUR.
C
      TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB3,36))
      IF(NSUB.NE.4) GO TO 70
   30 CONTINUE
C
C     ALL DONE, INCREMENT THE NO. OF ENTRIES AND RETURN.
C
      TABLE(3,1)=NENTRY+1
      RETURN
C
C     ERROR - MORE THAN ONE SUBSCRIPT INDICATED BUT NSUB1 NOT GIVEN OR
C              IN ERROR (ZERO OR NEGATIVE)
C
   40 IEFLAG=IEFLAG+1
C
C     ERROR - MORE THAN TWO SUBSCRIPTS INDICATED BUT NSUB2 NOT GIVEN OR
C              IN ERROR
C
   50 IEFLAG=IEFLAG+1
C
C     ERROR - MORE THAN THREE SUBSCRIPTS INDICATED BUT NSUB3 NOT GIVEN
C              OR IN ERROR
C
   60 IEFLAG=IEFLAG+1
C
C     ERROR - MORE THAN FOUR SUBSCRIPTS INDICATED
C
   70 IEFLAG=IEFLAG+1
  999 CONTINUE
      IEFLAG=OR(IEFLAG,AND(NAME,77777777777700000000B))
      GO TO IERRT
      END
```

```
      SUBROUTINE NXTCOL(IF,IEFLAG)
C
C     ROUTINE TO GET THE NEXT CHARACTER FROM THE CURRENT NAMELIST CARD
C         AND,IF NECESSARY,READ ANOTHER CARD FROM FILE IF
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
     1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
      LOGICAL LIST,HOLLER
C
C     INCREMENT THE COLUMN NO.
C
      ISUB=ISUB+1
C
C     PICK UP THE CHARACTER IN COLUMN ISUB
```

```
C
      ITEST=ICHAR(ISUB)
C
C     IF ISUB HAS NOT MOVED ACROSS A CARD BOUNDARY,WE ARE ALL DONE
C
      IF(ISUB.LE.80) GO TO 30
C
C     MUST READ NEXT CARD. IEFLAG WILL BE RETURNED NON-ZERO ONLY IF
C         AN EOF WAS ENCOUNTERED.
C
      CALL READIT(IF,IEFLAG)
      IF(IEFLAG.EQ.0) GO TO 10
C
C     AN EOF WAS READ, WE ARE ALL THROUGH.
C
      CALL MESSOT(32)
      IEFLAG=-32
      RETURN
   10 CONTINUE
C
C     READ WAS SUCCESSFUL. RESET COLUMN AND CHARACTER.
C
      ISUB=2
      ITEST=ICHAR(ISUB)
C
C     IF PROCESSING A HOLLERITH FIELD,NO MORE CHECKING
C
      IF(HOLLER) GO TO 20
C
C     CHECK FOR $ IN CC2. IF NOT,ALL FINISHED CHECKING.
C
      IF(ITEST.NE.1H$) GO TO 20
C
C     $ IN CC2. PROBABLY BEGINNING OF NEXT NAMELIST STATEMENT. MUST
C         HAVE HAD MISSING TERMINAL $ ON PREVIOUS CARD. GO BACK AND
C         READ THE CARD IN ERROR.
C
      BACKSPACE IF
      BACKSPACE IF
      CALL READIT(IF,IEFLAG)
C
C     SET THE COLUMN POINTER TO CC80,GO PRINT ERROR MESSAGE,AND QUIT
C
      ISUB=80
      CALL MESSOT(33)
      IEFLAG=33
      RETURN
   20 CONTINUE
C
C     EVERYTHING CHECKED OUT. SET COLUMN TO 1 AND PICK UP THE
C         CHARACTER IN CC1.
C
      ISUB=1
      ITEST=ICHAR(ISUB)
C
C     IF FLAG IS SET,PRINT THE LAST CARD READ
C
```

```
      IF(LIST) CALL MESSOT(1)
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE READIT(IF,IEFLAG)
C
C     ROUTINE TO READ THE NEXT CARD ON FILE IF
C
C     WRITTEN BY J.L.NORTON,LASL T=3,1974
C
      COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
     1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
C
C     READ THE NEXT CARD
C
      READ(IF,30) ICHAR
C
C     CHECK FOR END-OF-FILE
C
      IF(EOF,IF) 10,20
C
C     YES, SET THE FLAG.
C
   10 CONTINUE
      IEFLAG=-1
   20 CONTINUE
      RETURN
C
   30 FORMAT(80A1)
      END
```

```
      SUBROUTINE OPDMP(FWA,LWA,IFILE)
C
C     ROUTINE TO DUMP SMALL CORE MEMORY IN CDC COMPASS MNEUMONICS
C
C     FWA - FIRST WORD TO BE DUMPED
C     LWA - LAST WORD TO BE DUMPED
C     IFILE - FILE TO WHICH TO DIRECT OUTPUT
C
C     WRITTEN BY J.L.NORTON,LASL T=3,1975
C
      INTEGER FWA,FWAP,GETIT,SHIFT,AND,OR
      DIMENSION IDIGIT(20),ICODE(4),IREG(4),IOPRN(13,4),IDNAME(8)
      DIMENSION IOP(42),IHEQ(20)
```

```
      DATA IOP/2HPS,2HRL,2HWL,2HMJ,2HRX,2HWX,2HRI,2HTB,2HIB,2HRO,2HOB,
     1 2HRJ,2HJP,2HZR,2HNZ,2HPL,2HNG,2HIR,2HOR,2HDF,2HID,2HEQ,2HNE,2HGE,
     2 2HLT,2HBX,2HLX,2HAX,2HMX,2HNX,2HZX,2HUX,2HPX,2HFX,2HDX,2HRX,2HIX,
     3 2HNO,2HCX,2HSA,2HSB,2HSX/
      DATA IDNAME/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7/
C
C     IF FWA.GT.LWA,IGNORE THE CALL
C
      IF(FWA.GT.LWA) RETURN
      FWAP=FWA
C
C     LOOP OVER ALL WORDS TO BE DUMPED
C
   10 CONTINUE
C
C     GET THE CONTENTS OF THE CURRENT ADDRESS (FWAP) TO BE DUMPED
C
      IWORD=GETIT(FWAP)
C
C     BREAK THE WORD UP INTO INDIVIDUAL DIGITS
C
      DO 20 I=1,20
      IDIGIT(I)=AND(SHIFT(IWORD,3*I),7B)
   20 CONTINUE
C
C     COMPUTE HOLLERITH EQUIVALENT OF DIGITS
C
      DO 30 I=1,20
      ISUB=IDIGIT(I)+1
   30 IHEQ(I)=IDNAME(ISUB)
C
C     IDN IS THE CURRENT DIGIT NO. OF THE WORD BEING PROCESSED
C
      IDN=1
C
C     IPART IS THE NO. OF THE INSTRUCTION (1-4) BEING PROCESSED
C         FOR THE CURRENT WORD
C
      IPART=1
C
C     INITIALIZE THE OUTPUT FIELDS
C
      DO 50 I=1,4
      ICODE(I)=2H
      IREG(I)=1H
      DO 40 J=1,13
      IOPRN(J,I)=1H
   40 CONTINUE
   50 CONTINUE
C
C     LOOP OVER ALL DIGITS
C
   60 CONTINUE
C
C     IOPNO IS THE NO. OF THE CHARACTER OF THE OPERAND FIELD CURRENTLY
C         BEING FILLED
C
```

```
      IOPNO=1
      IGOT=IDIGIT(IDN)+1
      GO TO (70,360,440,500,520,620,620,620),IGOT
C
C     FIRST DIGIT OF OP IS 0
C
   70 CONTINUE
      IDN=IDN+1
      IGOT=IDIGIT(IDN)+1
      GO TO (80,190,330,340,350,350,350,350),IGOT
C
C     OP IS PS (00)
C
   80 CONTINUE
      ITYPE=1
C
C     CHECK FOR ERRORS
C
      IF(IDN.LE.12) GO TO 100
C
C     SEE IF THIS IS END OF WORD PADDED WITH ZEROS. IF SO,
C         PROCESS LIKE  NO.
C
      IF(IDN.EQ.17) GO TO 600
C
C     WORD IS IN ERROR, MUST NOT BE INSTRUCTION. PRINT MESSAGE AND GO ON
C         TO NEXT WORD.
C
   90 CONTINUE
      WRITE(IFILE,760) FWAP,IWORD
      GO TO 180
  100 CONTINUE
      ITYPE=1
C
C     PROCESS INSTRUCTION OF THE FORM OP  K
C
      IDN=IDN+2
C
C     PROCESS K
C
  110 CONTINUE
      ISUM=0
      DO 120 I=1,6
      IDN=IDN+1
      ISUM=OR(ISUM,SHIFT(IDIGIT(IDN),3*(6-I)))
  120 CONTINUE
      ISUM=SHIFT(SHIFT(ISUM,42),-42)
      IF(ISUM.GE.0) GO TO 130
      ISUM=-ISUM
      IOPRN(IOPNO,IPART)=1H-
      GO TO 140
  130 CONTINUE
      IOPRN(IOPNO,IPART)=1H+
  140 CONTINUE
      IOPNO=IOPNO+1
      DO 150 I=1,6
      ID=AND(SHIFT(ISUM,-3*(6-I)),7B)
```

```
          IOPRN(IOPNO,IPART)=IDNAME(ID+1)
          IOPNO=IOPNO+1
  150 CONTINUE
C
C         FINISH PARCEL
C
  160 CONTINUE
          ICODE(IPART)=IOP(ITYPE)
          IF(IDN.GE.20) GO TO 170
          IPART=IPART+1
          IDN=IDN+1
          IF(IDN.GT.16) GO TO 90
          GO TO 60
C
C         WORD IS FINISHED. PRINT THE LINE.
C
  170 CONTINUE
          WRITE(IFILE,770) FWAP,IWORD,(ICODE(I),IREG(I),(IOPRN(J,I),J=1,13),
         1 I=1,IPART)
C
C         SEE IF THIS WAS THE LAST WORD
C
  180 CONTINUE
          IF(FWAP.GE.LWA) RETURN
C
C         NO. CONTINUE.
C
          FWAP=FWAP+1
          GO TO 10
C
C         FIRST TWO DIGITS OF OP ARE 01
C
  190 CONTINUE
          IDN=IDN+1
          IGOT=IDIGIT(IDN)+1
          GO TO (200,210,210,230,240,240,250,310),IGOT
C
C         OP IS RJ (010)
C
  200 CONTINUE
          ITYPE=12
          IDN=IDN+1
          GO TO 110
C
C         OP IS 011 OR 012
C
  210 CONTINUE
          ITYPE=IGOT
C
C         PROCESS INSTRUCTION OF THE FORM   OP  BJ+K
C
  220 CONTINUE
          IF(IDN.GT.13) GO TO 90
          IDN=IDN+1
          IOPRN(1,IPART)=1HB
          IOPRN(2,IPART)=IHEQ(IDN)
          IOPNO=3
```

```
      GO TO 110
C
C     OP IS MJ
C
  230 CONTINUE
      ITYPE=4
      IF(IDIGIT(IDN+1).NE.0.OR.IDIGIT(IDN+2).NE.0) GO TO 220
      IF(IDN.GT.13) GO TO 90
      IDN=IDN+7
      GO TO 160
C
C     OP IS 014 OR 015
C
  240 CONTINUE
      ITYPE=IGOT
C
C     PROCESS INSTRUCTION OF THE FORM   OPJ   XK
C
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPRN(1,IPART)=1HX
      IOPRN(2,IPART)=IHEQ(IDN)
      GO TO 160
C
C     OP IS 016
C
  250 CONTINUE
      IF(IDIGIT(IDN+1).NE.0) GO TO 280
      ITYPE=7
C
C     PROCESS INSTRUCTION OF THE FORM  OP   BK
  260 CONTINUE
      IDN=IDN+2
  270 CONTINUE
      IOPRN(1,IPART)=1HB
      IOPRN(2,IPART)=IHEQ(IDN)
      GO TO 160
  280 CONTINUE
      IF(IDIGIT(IDN+2).NE.0) GO TO 290
      ITYPE=8
C
C     PROCESS INSTRUCTION OF THE FORM   OPJ
C
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      GO TO 160
  290 CONTINUE
      ITYPE=9
C
C     PROCESS INSTRUCTION OF THE FORM  OPJ   BK
C
  300 CONTINUE
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
```

```
      GO TO 270
C
C     OP IS 017
C
  310 CONTINUE
      IF(IDIGIT(IDN+1).NE.0) GO TO 320
      ITYPE=10
      GO TO 260
  320 CONTINUE
      ITYPE=11
      GO TO 300
C
C     OP IS JP (02)
C
  330 CONTINUE
      ITYPE=13
C
C     PROCESS INSTRUCTION OF THE FORM  OP   BJ+K
C
      IF(IDN.GT.12) GO TO 90
      IDN=IDN+1
      IOPRN(1,IPART)=1HB
      IOPRN(2,IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPNO=3
      GO TO 110
C
C     FIRST TWO DIGITS OF OP ARE 03
C
  340 CONTINUE
      IF(IDN.GT.12) GO TO 90
      IDN=IDN+1
      IGOT=IDIGIT(IDN)
      ITYPE=IGOT+14
C
C     PROCESS INSTRUCTION OF THE FORM  OP   XJ,K
C
      IDN=IDN+1
      IOPRN(1,IPART)=1HX
      IOPRN(2,IPART)=IHEQ(IDN)
      IOPRN(3,IPART)=1H,
      IOPNO=4
      GO TO 110
C
C     FIRST TWO DIGITS OF OP ARE 04,05,06,OR 07
C
  350 CONTINUE
      IF(IDN.GT.12) GO TO 90
      ITYPE=IGOT+17
C
C     PROCESS INSTRUCTION OF THE FORM  OP   BI,BJ,K
C
      IDN=IDN+1
      IOPRN(1,IPART)=1HB
      IOPRN(2,IPART)=IHEQ(IDN)
      IOPRN(3,IPART)=1H,
      IOPRN(4,IPART)=1HB
```

```
      IDN=IDN+1
      IOPRN(5,IPART)=IHEQ(IDN)
      IOPRN(6,IPART)=1H,
      IOPNO=7
      GO TO 110
C
C     FIRST DIGIT OF OP IS 1.  OP IS BXI.
C
  360 CONTINUE
      ITYPE=26
      IDN=IDN+1
      IGOT=IDIGIT(IDN)+1
      GO TO (370,370,370,370,430,430,430,430),IGOT
C
C     PROCESS INSTRUCTION OF THE FORM  OPI    XJ
C                                             XJ*XK
C                                             XJ+XK
C                                             XJ-XK
C                                             XJ/XK
C                                            (IGOT=1,2,3,4,OR 5)
C
  370 CONTINUE
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPRN(1,IPART)=1HX
      IOPRN(2,IPART)=IHEQ(IDN)
      IDN=IDN+1
      IF(IGOT.EQ.1) GO TO 160
      KREG=IHEQ(IDN)
      IOPNO=3
      IGOT=IGOT-1
      GO TO (380,400,410,420),IGOT
C
C     ADD ON *XK
C
  380 CONTINUE
      IOPRN(IOPNO,IPART)=1H*
  390 CONTINUE
      IOPRN(IOPNO+1,IPART)=1HX
      IOPRN(IOPNO+2,IPART)=KREG
      GO TO 160
C
C     ADD ON +XK
C
  400 CONTINUE
      IOPRN(IOPNO,IPART)=1H+
      GO TO 390
C
C     ADD ON -XK
C
  410 CONTINUE
      IOPRN(IOPNO,IPART)=1H-
      GO TO 390
C
C     ADD ON /XK
C
```

```
  420 CONTINUE
      IOPRN(IOPNO,IPART)=1H/
      GO TO 390
C
C     PROCESS INSTRUCTION OF THE FORM  OPI    =XK
C                                             =XK*XJ
C                                             =XK+XJ
C                                             =XK-XJ
C                                            (IGOT=5,6,7,OR 8)
C
  430 CONTINUE
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      KREG=IHEQ(IDN)
      IOPRN(1,IPART)=1H=
      IOPRN(2,IPART)=1HX
      IDN=IDN+1
      IOPRN(3,IPART)=IHEQ(IDN)
      IF(IGOT.EQ.5) GO TO 160
      IGOT=IGOT-5
      IOPNO=4
      GO TO (380,400,410),IGOT
C
C     FIRST DIGIT OF OP IS 2
C
  440 CONTINUE
      IDN=IDN+1
      IGOT=IDIGIT(IDN)+1
      GO TO (450,480,450,480,490,490,490,490),IGOT
C
C     OP IS LXI
C
  450 CONTINUE
      ITYPE=27
      IF(IGOT.EQ.1) GO TO 470
C
C     PROCESS INSTRUCTION OF THE FORM  OPI    BJ,XK
C
  460 CONTINUE
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPRN(1,IPART)=1HB
      IOPRN(2,IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPRN(3,IPART)=1H,
      IOPRN(4,IPART)=1HX
      IOPRN(5,IPART)=IHEQ(IDN)
      GO TO 160
C
C     PROCESS INSTRUCTION OF THE FORM  OPI    JK
C
  470 CONTINUE
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
```

```
      IOPRN(1,IPART)=IHEQ(IDN)
      IDN=IDN+1
      IOPRN(2,IPART)=IHEQ(IDN)
      IOPRN(3,IPART)=1HB
      GO TO 160
C
C     OP IS AXI
C
  480 CONTINUE
      ITYPE=28
      IF(IGOT.EQ.2) GO TO 470
      GO TO 460
C
C     OP IS NXI,ZXI,UXI,OR PXI
C
  490 CONTINUE
      ITYPE=IGOT+25
      GO TO 460
C
C     FIRST DIGIT OF OP IS 3
C
  500 CONTINUE
      IDN=IDN+1
      IGOTP=IDIGIT(IDN)
      ITYPE=IGOTP/2+34
      IF(MOD(IGOTP,2).NE.0) GO TO 510
      IGOT=3
      GO TO 370
  510 CONTINUE
      IGOT=4
      GO TO 370
C
C     FIRST DIGIT OF OP IS 4
C
  520 CONTINUE
      IDN=IDN+1
      IGOT=IDIGIT(IDN)+1
      GO TO (530,540,550,570,580,580,590,570),IGOT
C
C     OP IS FLOATING MULTIPLY
C
  530 CONTINUE
      ITYPE=34
      GO TO 560
  540 CONTINUE
      ITYPE=36
      GO TO 560
  550 CONTINUE
      ITYPE=35
  560 CONTINUE
      IGOT=2
      GO TO 370
C
C     OP IS MXI OR CXI
C
  570 CONTINUE
      ITYPE=29
```

```
      IF(IGOT.EQ.7) ITYPE=39
      GO TO 470
C
C     OP IS FLOATING DIVIDE
C
  580 CONTINUE
      ITYPE=34
      IF(IGOT.EQ.5) ITYPE=36
      IGOT=5
      GO TO 370
C
C     OP IS NO
C
  590 CONTINUE
      ITYPE=38
C
C     PROCESS 15 BIT INSTRUCTION OF THE FORM  OP   N
C
  600 CONTINUE
      IDN=IDN+1
      IF(IDIGIT(IDN).NE.0.OR.IDIGIT(IDN+1).NE.0.OR.IDIGIT(IDN+2).NE.0)
     1 GO TO 610
      IDN=IDN+2
      GO TO 160
  610 CONTINUE
      IOPRN(1,IPART)=IHEQ(IDN)
      IOPRN(2,IPART)=IHEQ(IDN+1)
      IOPRN(3,IPART)=IHEQ(IDN+2)
      IOPRN(4,IPART)=1HB
      IDN=IDN+2
      GO TO 160
C
C     OP BEGINS WITH 5,6,OR 7
C
  620 CONTINUE
      ITYPE=IGOT+34
      IDN=IDN+1
      IGOT=IDIGIT(IDN)+1
      IF(IGOT.LT.4.AND.IDN.GT.12) GO TO 90
      IDN=IDN+1
      IREG(IPART)=IHEQ(IDN)
      IDN=IDN+1
      JREG=IHEQ(IDN)
      IF(IGOT.LT.4) GO TO 630
      IDN=IDN+1
      KREG=IHEQ(IDN)
  630 CONTINUE
      GO TO (640,660,670,680,700,710,730,750),IGOT
  640 CONTINUE
      IOPRN(1,IPART)=1HA
  650 CONTINUE
      IOPRN(2,IPART)=JREG
      IOPNO=3
      GO TO 110
  660 CONTINUE
      IOPRN(1,IPART)=1HB
      GO TO 650
```

```
  670 CONTINUE
      IOPRN(1,IPART)=1HX
      GO TO 650
  680 CONTINUE
      IOPRN(1,IPART)=1HX
      IOPRN(3,IPART)=1H+
  690 CONTINUE
      IOPRN(2,IPART)=JREG
      IOPRN(4,IPART)=1HB
      IOPRN(5,IPART)=KREG
      GO TO 160
  700 CONTINUE
      IOPRN(3,IPART)=1H+
      GO TO 720
  710 CONTINUE
      IOPRN(3,IPART)=1H-
  720 CONTINUE
      IOPRN(1,IPART)=1HA
      GO TO 690
  730 CONTINUE
      IOPRN(3,IPART)=1H+
  740 CONTINUE
      IOPRN(1,IPART)=1HB
      GO TO 690
  750 CONTINUE
      IOPRN(3,IPART)=1H-
      GO TO 740
C
  760 FORMAT(1H ,O6,3XO20,3X8(1H=),
     1 54HWORD CONTAINS ILLEGAL INSTRUCTION(S), MUST BE DATA OR ,
     2 10HCLOBBERED,,8(1H=))
  770 FORMAT(1H ,O6,3XO20,4(3XA2,A1,3X13A1))
      END
```

```
      SUBROUTINE PABORT
C
C     ROUTINE TO HANDLE ABORT EXCHANGE PACKAGE DUMPS
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
      INTEGER SHIFT
      LOGICAL FILM
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
      COMMON/IEDMP/IDUMP(16)
C
C     PRINT CP TIME AT ABORT SO A TIME LIMIT ABORT IS EASILY
C         DISTINGUISHABLE
C
      CALL SECOND(TJLN)
      DO 10 IPX=6,IFD,6
   10 WRITE(IPX,30) TJLN
```

```
C
C     PRINT THE HARDWARE EXCHANGE PACKAGE
C
      DO 20 IPX=6,IFD,6
   20 WRITE(IPX,40)
      CALL DMPPK(6,IDUMP)
C
C     SEND IT TO FILM IF FILM IS .TRUE.
C
      IF(FILM) CALL DMPPK(12,IDUMP)
      ENTRY PDMPPK
C
C     GET THE P COUNTER FOR THE ABORT
C
      IFW=SHIFT(IDUMP,-36)
C
C     SUBTRACT 32 (40 OCTAL) FROM IT. IF RESULT IS NEGATIVE,SET IT
C         TO ZERO.
C
      IFW=IFW-32
      IF(IFW.LT.0) IFW=0
C
C     ADD 37 OCTAL (IFW-40B+77B) TO P COUNTER. IF RESULT IS OUT OF
C         RANGE,SET IT TO 77B.
C
      IFWB=IFW+77B
      IF(IFW.GT.150077B) IFW=0
      IF(IFWB.GT.150077B) IFWB=77B
C
C     PRINT OP CODE DUMP AROUND ABORT ADDRESS
C
      CALL OPDMP(IFW,IFWB,6)
C
C     SEND OP CODE DUMP TO FILM IF FILM IS .TRUE.
C
      IF(FILM) CALL OPDMP(IFW,IFWB,12)
      RETURN
C
   30 FORMAT(1H1,21HCP TIME AT ABORT WAS ,F9.3)
   40 FORMAT(//1H ,10(1H+),25HHARDWARD EXCHANGE PACKAGE,10(1H+))
      END
```

```
      SUBROUTINE PARPLT
C
C     ROUTINE TO PLOT PARTICLES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
```

```
      1 NLCP3,NLCP4,IFLMSZ
*      ----- END COMDECK PARAM       -----
*      ----- BEGIN COMDECK YSTORE     -----
*      ----- BEGIN COMDECK YAQDIM     -----
       DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
      1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
      2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
      3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
      4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ----- END COMDECK YAQDIM       -----
*      ----- BEGIN COMDECK YAQSC      -----
       LOGICAL RESTRT,FILM,PAPER,TURB
       REAL LAM,MU
C      COMMON/YSC1/AASC(NSCP1)
       COMMON/YSC1/AASC(9600)
       COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
      1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
       COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
       COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
       COMMON/YSC2/ZZ
C      COMMON/YSC4/ITAB(ITABP)
       COMMON/YSC4/ITAB(1000)
       COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ----- END COMDECK YAQSC        -----
*      ----- BEGIN COMDECK YAQEQ      -----
       EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
      1 (8),E,ETIL,AREA,XR13K),
      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
      1 AVYSV,X24K)
       REAL M,MP,MPAR,MTIL
*      ----- END COMDECK YAQEQ        -----
*      ----- END COMDECK YSTORE       -----
*      ----- BEGIN COMDECK PCALL      -----
       COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*      ----- END COMDECK PCALL        -----
```

```
      COMMON/XTENC/XTEN,YTEN
      DATA BCD/1H /
C
C     ADVANCE FILM TO NEXT FRAME
C
      CALL ADV(1)
C
C     INITIALIZE PLOT VALUES
C
      IPXL=65
      IPXR=660
      IPYB=900
      FIPXL=65.
      FIPXR=660.
      FIPYB=900.
      YUP=PTOP+2.*PRITE
      YLB=PBOTM-3.*PRITE
      PYCNVP=FLOAT(IPYT-IPYB)/(YUP-YLB)
      PXCONV=PXCONV*PYCNVP/PYCONV
      PYCONV=PYCNVP
C
C     SET UP CALL TO TICBOX
C
      XCONVP=PXCONV
      YCONVP=PYCONV
      IXLSV=IXL
      IXRSV=IXR
      IYBSV=IYB
      IYTSV=IYT
      IXL=IPXL
      IXR=IPXR
      IYB=IPYB
      IYT=IPYT
      FIYBSV=FIYB
      FIXLSV=FIXL
      FIYB=FIPYB
      FIXL=FIPXL
C
C     DRAW AND LABEL THE PLOT FRAME
C
      CALL TICBOX
C
C     LABEL THE PARTICLE PLOT
C
      CALL LINCNT(60)
      WRITE(IFD,100) PXR,PYB,PYT
      WRITE(IFD,110) JNM,NAME,T,NCYC
C
C     LOOP OVER ALL THE PARTICLES AND PLOT THEM
C
C     IECP IS THE LCM INDEX
C     NPPT IS THE PARTICLE COUNTER
C
      IECP=1
      NPPT=0
C
C     BRING A BUFFER-LOAD OF PARTICLE DATA FROM LCM INTO SCM
```

```
C
   10 CALL ECRD(AASC,NLCP1+IECP-1,LPB,IDUM)
C
C     KP IS THE SCM INDEX
C
      KP=1
C
C     SKIP THE PARTICLE IF ITS X-COORDINATE IS NEGATIVE
C
   20 IF(XPAR(KP).LT.0.) GO TO 30
C
C     CALCULATE THE RASTER COORDINATES OF THE PARTICLE
C
      IX1=FIPXL+(XPAR(KP)-PXL)*PXCONV
      IY1=FIPYB+(YPAR(KP)-YLB)*PYCONV
C
C     PLOT IT UNLESS IT IS OUTSIDE OF THE PLOTTING RECTANGLE
C
      IF(IX1.GT.IXR.OR.IX1.LT.IXL) GO TO 30
      IF((IY1.GT.IYB).OR.(IY1.LT.IYT)) GO TO 30
      CALL PLT(IX1,IY1,42)
      CALL PLT(IX1,IY1,42)
      CALL PLT(IX1,IY1,42)
C
C     INCREMENT THE PARTICLE COUNTER AND SEE IF ALL HAVE BEEN PLOTTED
C
   30 NPPT=NPPT+1
      IF(NPPT.EQ.NPT) GO TO 40
C
C     NO. INCREMENT THE SCM POINTER AND SEE IF THE BUFFER NEEDS TO
C         BE REFILLED.
C
      KP=KP+3
      IF(KP.LT.LPB) GO TO 20
C
C     YES. INCREMENT THE LCM INDEX AND GO REFILL THE SCM BUFFER.
C
      IECP=IECP+LPB
      GO TO 10
C
C     ALL PARTICLES HAVE BEEN PLOTTED. SEE IF ANY TIME-DEPENDENT
C         PLOTTING IS DESIRED.
C
   40 CONTINUE
      IF(IST.LE.0) GO TO 80
C
C     YES. SEE IF THERE HAVE BEEN AT LEAST TWO TIME PERIODS SAVED.
C
      IF(NILNG.LT.2) GO TO 80
C
C     YES. PREPARE TO DO THE TIME-DEPENDENT PARTICLE PLOT.
C
      CALL ADV(1)
      CALL TICBOX
      NIST=NPT/IST
      NTOT=NPT/NIST
      DO 70 I=1,NTOT
```

```
      DO 60 J=1,NILNG
C
C     SAVE THE LAST POSITION IN TIME
C
      IX3=IX2
      IY3=IY2
      IOK3=IOK2
      IBEGIN=NLCP1+NLCP2+2*(NP1*(J-1)+I-1)
      CALL ECRD(XTEN,IBEGIN,2,IDUM)
C     IX2=FIPXL+(PAX(I,J)-XL)*PXCONV
      IX2=FIPXL+(XTEN-XL)*PXCONV
C     IY2=FIPYB+(PAY(I,J)-YLB)*PYCONV
      IY2=FIPYB+(YTEN-YLB)*PYCONV
      IOK2=0
      IF(IX2.GT.IXR.OR.IX2.LT.IXL.OR.IY2.GT.IYB.OR.IY2.LT.IYT) IOK2=1
C
C     MARK THE INITIAL POSITION WITH A STAR
C
      IF(J.NE.1) GO TO 50
      IF(IOK2.EQ.0) CALL PLT(IX2,IY2,44)
      GO TO 60
   50 CONTINUE
C
C     MARK PARTICLES WITH A DOT
C
      IF(IOK2.EQ.0) CALL PLT(IX2,IY2,27)
C
C     CONNECT THE PARTICLE WITH ITS NEIGHBOR IN TIME
C
      IF(IOK2.EQ.0.AND.IOK3.EQ.0) CALL DRV(IX2,IY2,IX3,IY3)
   60 CONTINUE
   70 CONTINUE
C
C     LABEL THE PLOT
C
      CALL LINCNT(60)
      WRITE(IFD,90)
      WRITE(IFD,110) JNM,NAME,T,NCYC
   80 CONTINUE
C
C     RESTORE THE PLOT INDICES
C
      IXL=IXLSV
      IXR=IXRSV
      IYB=IYBSV
      IYT=IYTSV
      FIYB=FIYBSV
      IFXL=FIXLSV
      RETURN
C
   90 FORMAT(25H TIME-DEPENDENT PARTICLES)
  100 FORMAT(10H PARTICLES/11X5H PXR=1PE12.5,5H PYB=E12.5,5H PYT=E12.5)
  110 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
      END
```

```
      SUBROUTINE PCHCK
C
C     ROUTINE TO CHECK THE YAQUI LARGE AND SMALL CORE PARAMETERS
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,THAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXFF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
C     CHECK TO MAKE SURE THE SCM BUFFER WILL BE LARGE ENOUGH
C
      IF(3*NQI.GT.NSCP1) CALL UNCLE(4,5HPCHCK,33,
     1 33HNOT ENOUGH SMALL CORE FOR BUFFERS)
C
C     MAKE SURE LCM IS LARGE ENOUGH TO HOLD ALL THE ARRAYS
C
      IF(NLC.GT.NLCP1) CALL UNCLE(4,5HYASET,14,14HNOT ENOUGH LCM)
      RETURN
      END
```

```
      SUBROUTINE PRTGEN
```

```
C
C       ROUTINE TO GENERATE PARTICLES
C
C       ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C       MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*       ----- BEGIN COMDECK PARAM      -----
        COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
       1 NLCP3,NLCP4,IFLMSZ
*       ----- END COMDECK PARAM      -----
*       ----- BEGIN COMDECK YSTORE      -----
*       ----- BEGIN COMDECK YAQDIM      -----
        DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
       1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
       2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
       3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
       4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
       5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
       6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*       ----- END COMDECK YAQDIM      -----
*       ----- BEGIN COMDECK YAQSC      -----
        LOGICAL RESTRT,FILM,PAPER,TURB
        REAL LAM,MU
C       COMMON/YSC1/AASC(NSCP1)
        COMMON/YSC1/AASC(9600)
        COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
       1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
       2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
       3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
       4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
        COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
       1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
       2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
       3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
        COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
       1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,GLEVEL,TQ,IST,
       2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
       3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
       4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
       5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
       6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
        COMMON/YSC2/ZZ
C       COMMON/YSC4/ITAB(ITABP)
        COMMON/YSC4/ITAB(1000)
        COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*       ----- END COMDECK YAQSC      -----
*       ----- BEGIN COMDECK YAQEQ      -----
        EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
       1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
       1 (8),E,ETIL,AREA,XR13K),
       2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
       3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
       4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
       5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
       6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
       7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
       8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
```

```
      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
      1 AVYSV,X24K)
        REAL M,MP,MPAR,MTIL
*       ----- END COMDECK YAQEQ        -----
*       ----- END COMDECK YSTORE       -----
        DIMENSION PARTN(3,7)
        DATA (PARTN(II),II=1,21)/21*0/
        DATA IEFLAG/0/,NE/0/
        INTEGER COMP,AND,OR
C
C       SET UP THE NAMELIST INPUT TABLE
C
        ASSIGN 170 TO IERPT
        CALL TABDEF(PARTN,5HPARTN,7,IERRT)
        CALL TABSET(PARTN,5HDRPAR,DRPAR,IEFLAG,0,0,0,0)
        CALL TABSET(PARTN,5HDZPAR,DZPAR,IEFLAG,0,0,0,0)
        CALL TABSET(PARTN,2HXC,XC,IEFLAG,0,0,0,0)
        CALL TABSET(PARTN,2HYC,YC,IEFLAG,0,0,0,0)
        CALL TABSET(PARTN,2HXD,XD,IEFLAG,0,0,0,0)
        CALL TABSET(PARTN,2HYD,YD,IEFLAG,0,0,0,0)
C
C       INITIALIZE PARTICLE COUNT
C
        NPT=0
C
C       DEFINE CONSTANT
C
        NQI2=NQI*2
C
C       LPB IS THE LENGTH OF ONE SMALL CORE BUFFER ROUNDED DOWN TO THE
C           NEAREST MULTIPLE OF THREE. PARTICLE DATA WILL BE STORED IN
C           A SMALL CORE BUFFER IN THREE WORD BLOCKS.
C
        LPB=NQI/3*3
C
C       KP IS THE SCM BUFFER SUBSCRIPT
C
        KP=1
C
C       IECP IS THE CURRENT LCM READ ADDRESS
C
        IECP=1
C
C       IF THE MESH IS VARIABLE,RECALCULATE THE PARTICLE DR AND DZ
C
        IF(FREZXR.EQ.1..AND.FREZYT.EQ.1..AND.FREZYB.EQ.1.) GO TO 10
C
C       IT IS. COMPUTE XMAX,YMAX,AND YMIN BASED ON GEOMETRIC PROGRESSION
C           RELATIONS. THE FIRST IUNF CELLS ARE OF WIDTH DR AND CONTRIBUTE
C           IUNF*DR TO XMAX. THIS INCLUDES CELLS I=1 TO I=IUNF. CELL
C           IUNF+1 HAS WIDTH FREZ*DR,CELL IUNF+2 HAS WIDTH FREZ*(FREZ*DR),
C           ETC. THE LAST CELL I=IBAR WILL HAVE WIDTH FREZ**(IBAR-IUNF)*DR
C           AND THE SUM WILL BE THE PROGRESSION
C                 FREZ*DR + FREZ**2*DR + ... + FREZ**(IBAR-IUNF)*DR
C              = FREZ*(1 + FREZ + FREZ**2 + ... + FREZ**(IBAR-IUNF-1))*DR.
C              = FREZ*DR*(1-FREZ**(IBAR-IUNF))/(1-FREZ).
C           THUS, XMAX = IUNF*DR + FREZ*DR*(1-FREZ**(IBAR-IUNF))/(1-FREZ).
```

```
C          YMAX AND YMIN ARE COMPUTED IN A SIMILAR FASHION.
C
      IF(FREZXR.GT.1.) XMAX=(FLOAT(IUNF)+FREZXR*(1.-FREZXR**(IBAR-IUNF))
     1 *ROMFXR)*DR
      IF(FREZYT.GT.1.) YMAX=REZY0+(FLOAT(JUNFO2)+FREZYT*(1.-FREZYT**(
     1 JBAR-JCEN-JUNFO2))*ROMFYT)*DZ
      IF(FREZYB.GT.1.) YMIN=REZY0-(FLOAT(JUNFO2)+FREZYB*(1.-FREZYB**(
     1 JCEN-JUNFO2))*ROMFYB)*DZ
C
C     READ IN QUANTITIES FOR ONE PARTICLE REGION. FIRST SET DEFAULT
C          VALUES.
C
   10 CONTINUE
C
C     DRPAR IS THE R DIMENSION OF THE RECTANGLE IN THE MIDDLE OF WHICH
C          ONE PARTICLE WILL BE PLACED
C
      DRPAR=0.
C
C     DZPAR IS THE Z RECTANGULAR DIMENSION
C
      DZPAR=0.
C
C     XC,YC,XD,AND YD ARE THE DIMENSIONS OF THE ENTIRE PARTICLE REGION.
C          IF THE REGION IS RECTANGULAR,(XC,YC) IS THE LOWER LEFT-HAND
C          CORNER AND (XD,YD) IS THE UPPER RIGHT-HAND CORNER. IF THE
C          REGION IS CIRCULAR,YD=0.,XC IS UNUSED,YC IS THE Z VALUE OF
C          THE CENTER OF THE CIRCLE (ASSUMED TO BE ON THE AXIS OF
C          SYMMETRY),AND XD IS THE RADIUS OF THE CIRCLE.
C
      XC=0.
      YC=0.
      XD=0.
      YD=0.
C
C     DO THE ACTUAL READ
C
      CALL NAMLST(PARTN,5,IEFLAG)
C
C     CHECK FOR INPUT ERRORS
C
      IF(IEFLAG.NE.0) CALL UNCLE(4,6HPRTGEN,26,
     1 26HPARTN NAMELIST INPUT ERROR)
C
C     SEE IF THIS WAS THE LAST PARTICLE REGION. IF SO,WE ARE ALL DONE.
C
      IF(DRPAR.LE.0.) GO TO 140
C
C     NO. CHECK FOR INPUT ERRORS.
C
      IF(DZPAR.LE.0.) CALL UNCLE(4,6HPRTGEN,14,14HERROR IN DZPAR)
      IF(YD.EQ.0..AND.XD.LE.0.) CALL UNCLE(4,6HPRTGEN,11,11HERROR IN XD)
C
C     NO ERRORS. PRINT OUT THE INPUT VARIABLES.
C
      DO 20 IPX=6,IFO,6
   20 WRITE(IPX,180) DRPAR,DZPAR,XC,YC,XD,YD
```

```
C
C     SEE IF PARTICLE REGION IS RECTANGULAR OR CIRCULAR
C
      IF(YD.EQ.0.) GO TO 40
C
C     REGION IS RECTANGULAR, SET PARTICLE REGION BOUNDS, IF VARIABLE
C         ZONING IS BEING USED,PARTICLE REGION BOUNDS ARE SET TO
C         PROBLEM BOUNDS.
C
      IF(FREZXR.EQ.1..AND.FREZYT.EQ.1..AND.FREZYB.EQ.1.) GO TO 30
C
C     VARIABLE ZONING
C
      PTOP=YMAX
      PBOTM=YMIN
      PRITE=XMAX
      PLEFT=0.
      GO TO 50
C
C     UNIFORM ZONING
C
   30 CONTINUE
      PTOP=YD
      PBOTM=YC
      PRITE=XD
      PLEFT=XC
      GO TO 50
C
C     REGION IS CIRCULAR
C
   40 CONTINUE
      PTOP=YC+XD
      PBOTM=YC-XD
      PRITE=XD
      PLEFT=0.
      R2=XD**2
   50 CONTINUE
C
C     INITIALIZE THE OLD PTOP
C
      PTPOLD=PTOP
C
C     BEGIN LOOP OVER PARTICLES
C
C     INITIALIZE THE Y COORDINATE
C
      YTE=PBOTM+.5*DZPAR
   60 CONTINUE
C
C     INITIALIZE THE X-COORDINATE
C
      XTE=PLEFT+.5*DRPAR
   70 CONTINUE
C
C     IF PARTICLE REGION IS CIRCULAR,AN EXTRA CHECK MUST BE MADE TO
C         SEE IF THE PARTICLE IS IN THE REGION
C
```

```
      IF(YD.NE.0.) GO TO 80
      IF((YTE-YC)**2+XTE**2.GT.R2) GO TO 100
   80 CONTINUE
C
C     STORE THE PARTICLE COORDINATES AND THE PARTICLE MASS
C
      XPAR(KP)=XTE
      YPAR(KP)=YTE
      MPAR(KP)=0.
C
C     INCREMENT THE SCM BUFFER SUBSCRIPT AND THE PARTICLE COUNT
C
      KP=KP+3
      NPT=NPT+1
C
C     SEE IF THE SCM BUFFER NEEDS TO BE FLUSHED
C
      IF(KP.GT.LPB) GO TO 110
C
C     NO. INCREMENT THE PARTICLE X-COORDINATE.
C
   90 XTE=XTE+DRPAR
C
C     SEE IF WE HAVE GONE OUTSIDE OF THE PARTICLE REGION
C
      IF(XTE.LE.PRITE) GO TO 70
C
C     YES. INCREMENT THE PARTICLE Y-COORDINATE.
C
  100 YTE=YTE+DZPAR
C
C     SEE IF Y HAS GONE OUTSIDE THE PARTICLE REGION
C
      IF(YTE.LE.PTOP) GO TO 60
C
C     YES. WE ARE DONE WITH THIS PARTICLE REGION. GO READ IN ANOTHER
C         CARD.
C
      GO TO 10
C
C     FLUSH THE BUFFER TO LCM IF THERE IS ROOM
C
  110 CONTINUE
      IF(IECP+LPB.LE.NLCP2+NLCP1) GO TO 130
  120 CALL UNCLE(4,6HPRTGEN,25,25HPARTICLE ARRAY OVERFLOWED)
  130 CONTINUE
      CALL ECWR(AASC,NLCP1+IECP-1,LPB,IDUM)
C
C     INCREMENT THE LCM ADDRESS,RESET THE SCM SUBSCRIPT,AND PROCEED
C
      IECP=IECP+LPB
      KP=1
      GO TO 90
C
C     ALL DONE. FLUSH THE SCM BUFFER TO LCM IF THERE IS ROOM.
C
  140 CONTINUE
```

```
      NW=KP-1
      IF(NW.EQ.0) GO TO 150
      IF(IECP+NW.GT.NLCP2+NLCP1) GO TO 120
      CALL ECWR(AASC,NLCP1+IECP-1,NW,IDUM)
  150 CONTINUE
C
C     NPS IS THE NO. OF WORDS OF PARTICLE STORAGE
C
      NPS=3*NPT
C
C     PRINT OUT THE NO. OF PARTICLES GENERATED AND THEIR TOTAL MASS
C
      DO 160 IPX=6,IFD,6
  160 WRITE(IPX,190) NPT,MT
      RETURN
  170 CONTINUE
      CALL UNCLE(4,6HPRTGEN,35,35HPARTN NAMELIST INITIALIZATION ERROR)
C
  180 FORMAT(8H  DRPAR=1PE12.5,8H  DZPAR=E12.5,5H  XC=E12.5/5H  YC=E12.5
     1 ,5H  XD=E12.5,5H  YD=E12.5)
  190 FORMAT(4XI6,38H PARTICLES GENERATED, WITH TOTAL MASS=1PE12.5)
      END
```

```
      SUBROUTINE PRTMOV
C
C     ROUTINE TO MOVE PARTICLES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM     -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM     -----
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
```

```
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 TLNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC        -----
*     ----- BEGIN COMDECK YAQEQ        -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ        -----
*     ----- END COMDECK YSTORE       -----
*     ----- BEGIN COMDECK PCOM       -----
      COMMON/XTENC/XTEN,YTEN,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
      EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
      EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
*     ----- END COMDECK PCOM         -----
      DATA TOUM/0/
      PTPOLD=PTOP
      NIST=NPT/IST
C
C     DECIDE WHETHER TO ADD ANOTHER POINT TO THE TIME-DEPENDENT
C         PARTICLE DATA
C
      IFLAGP=0
      IF(NCYC.LE.1) GO TO 10
      IF(T+EM10.LT.TOUT) GO TO 20
   10 CONTINUE
C
C     YES
C
      NILNG=NILNG+1
      IFLAGP=1
   20 CONTINUE
```

```
      IECP=1
      NPPT=0
      JOLD=0
      PBOTM=1.E+10
      PRITE=-1.E10
      PTOP=-1.E10
   30 CALL ECRD(AASC,NLCP1+IECP-1,LPB,IDUM)
      KP=1
   40 NPPT=NPPT+1
      MPTST=0
      XTE=XPAR(KP)
      IF(XTE.LT.0.) GO TO 130
      YTE=YPAR(KP)
      ICEL=ITAB(NPPT)
      J=ICEL/IP1+1
      I=ICEL-(J-1)*IP1
      IF(J.EQ.JOLD) GO TO 50
      JOLD=J
      IEC=(J-1)*NQI
      CALL ECRD(AASC(ISC2),IEC,NQI2,IDUM)
   50 IJ=(I-1)*NQ+ISC2
      IPJ=IJ+NQ
      IJP=IJ+NQI
      IPJP=IPJ+NQI
      X1=X(IPJ)
      Y1=Y(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
   60 XTEN=XTE
      YTEN=YTE
      CALL PSUB1(IN)
      IF(IN.NE.0) GO TO 70
      XTE=0.25*(X1+X2+X3+X4)
      YTE=0.25*(Y1+Y2+Y3+Y4)
      GO TO 160
   70 U1=U(IPJ)
      V1=V(IPJ)
      U2=U(IPJP)
      V2=V(IPJP)
      U3=U(IJP)
      V3=V(IJP)
      U4=U(IJ)
      V4=V(IJ)
      X1Z=X1-XTE
      X41=X4-X1
      X21=X2-X1
      X3421=X3-X4-X21
      Y1Z=Y1-YTE
      Y41=Y4-Y1
      Y21=Y2-Y1
      Y3421=Y3-Y4-Y21
      XMU=Y3421*X21-Y21*X3421
      XNU=Y41*X21-Y21*X41+Y3421*X1Z-Y1Z*X3421
```

```
      XLA=Y41*X1Z-Y1Z*X41
      IF(XMU.NE.0.) GO TO 80
      THE=-XLA/XNU
      IF(X41.NE.0.) GO TO 90
      ETA=-Y1Z/Y41
      GO TO 100
   80 THE=(-XNU+SQRT(XNU*XNU-4.*XMU*XLA))/(2.*XMU)
   90 ETA=-(X1Z+X21*THE)/(X41+X3421*THE)
  100 OME=1.-ETA
      OMT=1.-THE
      UK=OME*OMT*U1+OME*THE*U2+OMT*ETA*U4+ETA*THE*U3
      VK=OME*OMT*V1+OME*THE*V2+OMT*ETA*V4+ETA*THE*V3
      IF(MPTST.EQ.1) GO TO 110
      UKSV=UK
      VKSV=VK
      XTESV=XTE
      YTESV=YTE
      XTE=XTE+DT*UK
      YTE=YTE+DT*VK
      MPTST=1
      GO TO 60
  110 UK=0.5*(UK+UKSV)
      VK=0.5*(VK+VKSV)
      XTE=XTESV
      YTE=YTESV
      XTEN=XTE+DT*UK
      YTEN=YTE+DT*VK
      IF(TURB.AND.T.GT.TSTRTD) CALL PTRBDF(XTEN,YTEN)
      XPAR(KP)=XTEN
      YPAR(KP)=YTEN
C
C     SEE IF TIME-DEPENDENT PARTICLE DATA IS TO BE COLLECTED THIS CYCLE
C
      IF(IFLAGP.EQ.0) GO TO 120
C
C     YES. SEE IF THE CURRENT PARTICLE IS TO BE SAVED.
C
      IF(MOD(NPPT,NIST).NE.0) GO TO 120
C
C     YES. SAVE IT.
C
      NN=NPPT/NIST
      IBEGIN=NLCP1+NLCP2+2*(NP1*(NILNG-1)+NN-1)
C     PAX(NN,NILNG)=XTEN
C     PAY(NN,NILNG)=YTEN
      CALL ECWR(XTEN,IBEGIN,2,IDUM)
  120 CONTINUE
      PBOTM=AMIN1(YTEN,PBOTM)
      PRITE=AMAX1(XTEN,PRITE)
      PTOP=AMAX1(YTEN,PTOP)
  130 IF(NPPT.EQ.NPT) GO TO 140
      KP=KP+3
      IF(KP.LT.LPB) GO TO 40
      CALL ECWR(AASC,NLCP1+IECP-1,LPB,IDUM)
      IECP=IECP+LPB
      GO TO 30
  140 CALL ECWR(AASC,NLCP1+IECP-1,LPB,IDUM)
```

```
      IF(NCYC.EQ.1) PTPOLD=PTOP
      RETURN
  150 CONTINUE
      IF(IFLAG.EQ.2) GO TO 130
  160 XX1=X1
      YY1=Y1
      XX2=X2
      YY2=Y2
      CALL PSUB2(IT,IFLAG)
      IF(IFLAG.NE.0) GO TO 150
      IF(IT.EQ.0) GO TO 170
      IF(I.EQ.IBAR) GO TO 250
      I=I+1
      GO TO 210
  170 XX1=X4
      YY1=Y4
      XX2=X3
      YY2=Y3
      CALL PSUB2(IT,IFLAG)
      IF(IFLAG.NE.0) GO TO 150
      IF(IT.EQ.0) GO TO 180
      IF(I.EQ.1) GO TO 250
      I=I-1
      GO TO 210
  180 XX1=X3
      YY1=Y3
      XX2=X2
      YY2=Y2
      CALL PSUB2(IT,IFLAG)
      IF(IFLAG.NE.0) GO TO 150
      IF(IT.EQ.0) GO TO 190
      IF(J.EQ.JP1) GO TO 250
      J=J+1
      GO TO 200
  190 XX1=X4
      YY1=Y4
      XX2=X1
      YY2=Y1
      CALL PSUB2(IT,IFLAG)
      IF(IFLAG.NE.0) GO TO 150
      IF(IT.EQ.0) GO TO 230
      IF(J.EQ.2) GO TO 250
      J=J-1
  200 IEC=(J-1)*NQI
      JOLD=J
      CALL ECRD(AASC(ISC2),IEC,NQI2,IDUM)
      GO TO 220
  210 IJ=(I-1)*NQ+ISC2
      IPJ=IJ+NQ
      IJP=IJ+NQI
      IPJP=IPJ+NQI
  220 X1=X(IPJ)
      Y1=Y(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
```

```
      X4=X(IJ)
      Y4=Y(IJ)
      CALL PSUB1(IN)
      IF(IN.EQ.0) GO TO 160
      XTE=XTEN
      YTE=YTEN
      ITAB(NPPT)=(J-1)*IP1+I
      GO TO 70
  230 CONTINUE
      DO 240 IPX=IPD,IFD,6
  240 WRITE(IPX,280) NPPT,ITAB(NPPT),XTE,YTE,XTEN,YTEN,X1,Y1
      ITAB(NPPT)=(J-1)*IP1+I
      GO TO 130
  250 XPAR(KP)=-1.E+3
      DO 260 IPX=IPD,IFD,6
  260 WRITE(IPX,270) NPPT,ITAB(NPPT)
      GO TO 130
C
  270 FORMAT(5X,8HPARTICLE,I10,25HTOSSED OUT,   CELL NUMBER,I10)
  280 FORMAT(5X,14HERROR PARTICLE,I5,6H  CELL,I5,12F7.4)
      END
```

```
      SUBROUTINE PSUB1(IN)
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK YSTORE       -----
*     ----- BEGIN COMDECK YAQDIM       -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM       -----
*     ----- BEGIN COMDECK YAQSC       -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
```

```
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIF),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ      -----
*     ----- END COMDECK YSTORE     -----
*     ----- BEGIN COMDECK PCOM      -----
      COMMON/XTENC/XTEN,YTEN,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
      EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
      EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
*     ----- END COMDECK PCOM      -----
      DIMENSION AMZ(7)
      DATA PI/3.14159265358979323846 26/
      DATA PIO2/1.57079632679489661923 13/
      DATA PIO2M1/0.57079632679489661923 13/
      DATA PI2/6.28318530717958647692 52/
   10 AMZMIN=1.E+10
      DO 60 K=1,4
      IF(X1(K).EQ.XTEN) GO TO 40
      XX=(Y1(K)-YTEN)/(X1(K)-XTEN)
      YY=ABS(XX)
      IF(YY.GT.1.) GO TO 20
      SLOPE=PIO2M1*XX
      GO TO 30
   20 SLOPE=SIGN(PIO2-1./YY,XX)
   30 IF(XTEN.GT.X1(K)) SLOPE=SLOPE+PI
      IF(XTEN.LT.X1(K).AND.SLOPE.LT.0.) SLOPE=SLOPE+PI2
      GO TO 50
   40 SLOPE=SIGN(PIO2,Y1(K)-YTEN)
   50 AMZ(K)=SLOPE
   60 AMZMIN=AMIN1(AMZMIN,AMZ(K))
      AMZ(5)=AMZ(1)-AMZMIN
      AMZ(1)=AMZ(5)
      AMZ(6)=AMZ(2)-AMZMIN
      AMZ(2)=AMZ(6)
```

```
      AMZ(7)=AMZ(3)-AMZMIN
      AMZ(3)=AMZ(7)
      AMZ(4)=AMZ(4)-AMZMIN
      IF(AMZ(1).EQ.0.) K=1
      IF(AMZ(2).EQ.0.) K=2
      IF(AMZ(3).EQ.0.) K=3
      IF(AMZ(4).EQ.0.) K=4
      INTE=0
      IN=0
      IF(AMZ(K+3).GT.AMZ(K+2).AND.AMZ(K+2).GT.AMZ(K+1).AND.AMZ(K+1).GT.
     1 AMZ(K)) INTE=1
      IF(INTE.EQ.0) RETURN
      IN=1
      DO 100 K=1,4
      AMZ1=AMZ(K)
      AMZ2=AMZ(K+1)
      AMZ3=AMZ(K+2)
      IF(AMZ1*AMZ2.NE.0.) GO TO 70
      AMXZ=AMAX1(AMZ1,AMZ2)
      IF(AMZ3.LT.AMXZ) GO TO 80
   70 PHI=ABS(AMZ1-AMZ2)
      GO TO 90
   80 PHI=PI2-AMXZ
   90 IF(PHI.GT.PI) IN=0
      IF(IN.EQ.0) RETURN
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE PSUB2(IT,IFLAG)
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
```

```
         2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
         3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
         4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
          COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
         1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
         2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
         3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
          COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
         1 ILNG,NILNG,TP3,TUPOT,TQQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
         2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
         3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
         4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
         5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
         6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
          COMMON/YSC2/ZZ
C         COMMON/YSC4/ITAB(ITABP)
          COMMON/YSC4/ITAB(1000)
          COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*         ----- END COMDECK YAQSC        -----
*         ----- BEGIN COMDECK YAQEQ       -----
          EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
         1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
         1 (8),F,ETIL,AREA,XR13K),
         2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
         3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
         4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
         5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
         6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
         7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
         8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
         9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
         1 AVYSV,X24K)
          REAL M,MP,MPAR,MTIL
*         ----- END COMDECK YAQEQ        -----
*         ----- END COMDECK YSTORE       -----
*         ----- BEGIN COMDECK PCOM        -----
          COMMON/XTENC/XTEN,YTEN,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
          EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
          EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
*         ----- END COMDECK PCOM         -----
          DATA ALF,ALF2/0.001,1.00000001/
       10 IT=0
          IFLAG=0
          IF((YTEN-YY2)*(XTEN-XX1).EQ.(YTEN-YY1)*(XTEN-XX2)) GO TO 30
          IF(ABS(XTEN-XTE).LT.EM10) GO TO 50
          PSL=(YTEN-YTE)/(XTEN-XTE)
          IF(ABS(XX1-XX2).LT.EM10) GO TO 60
          GSL=(YY2-YY1)/(XX2-XX1)
          IF(PSL.EQ.GSL) RETURN
          XINS=(YY1-YTE+PSL*XTE-GSL*XX1)/(PSL-GSL)
          YINS=YY1+GSL*(XINS-XX1)
       20 ITA=0
          IF(((XX1-XINS)*(XX2-XINS).LE.0.).AND.((YY1-YINS)*(YY2-YINS).LE.0.)
         1 ) ITA=1
          IF(((XTE-XINS)*(XTEN-XINS).LE.0.).AND.((YTE-YINS)*(YTEN-YINS).LE.
         1 0.)) ITA=ITA+1
          IF(ITA.EQ.2) GO TO 70
```

```
      RETURN
   30 XTEN=ALF2*XTEN+(1.-ALF2)*XTE+ALF
      YTEN=ALF2*YTEN+(1.-ALF2)*YTE
      XPAR(KP)=XTEN
      YPAR(KP)=YTEN
      IFLAG=1
      DO 40 IPX=IPD,IFD,6
   40 WRITE(IPX,100) IFLAG
      RETURN
   50 XINS=XTE
      YINS=YY1+(YY1-YY2)/(XX1-XX2)*(XTE-XX1)
      GO TO 20
   60 XINS=XX1
      YINS=YTE+PSL*(XX1-XTE)
      GO TO 20
   70 IF((ABS(XTEN-XINS).LT.1.E-07).AND.(ABS(YTEN-YINS).LT.1.E-07)) GO
     1 TO 80
C     WE NEED TO SCALE ALL ALPHAS
      XTE=XINS+ALF*(XTEN-XINS)
      YTE=YINS+ALF*(YTEN-YINS)
      IT=1
      RETURN
   80 XPAR(KP)=XTEN-1.E-6*(XINS-XTE)
      YPAR(KP)=YTEN-1.E-6*(YINS-YTE)
      IFLAG=2
      DO 90 IPX=IPD,IFD,6
   90 WRITE(IPX,100) IFLAG
C     IF PARTICLE LANDS ON OR NEAR BOUNDARY, PULL IT BACK TO ORIGINAL
C     CELL AND GO ON TO NEXT PARTICLE
      RETURN
C
  100 FORMAT(5X,7HIFLAG =,I5)
      END
```

```
      SUBROUTINE PTRBDF(XJ,YJ)
C
C     ROUTINE TO ADD TURBULENT DIFFUSION EFFECTS TO THE PARTICLE
C         MOTION
C
C     (XJ,YJ) IS THE POSITION OF THE PARTICLE AFTER
C         THE EFFECTS OF CONVECTION HAVE BEEN ADDED. THE POSITION
C         WILL BE RETURNED WITH THE TURBULENT DIFFUSION CORRECTION
C         ADDED.
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
```

```
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC     -----
*     ----- BEGIN COMDECK YAQEQ     -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ     -----
*     ----- END COMDECK YSTORE     -----
*     ----- BEGIN COMDECK TRBDIF     -----
      COMMON/CTDIF/ERFV(21),NERFV,DXEF
*     ----- END COMDECK TRBDIF     -----
C     IPASS=1 IS FOR X (R)
C     IPASS=2 IS FOR Y (Z)
C
      IPASS=1
C
C     SKIP THE R CORRECTION IF XJ.LE.RMINEF
C
```

```
      IF(XJ.LE.RMINEF) GO TO 50
C
C     GET A RANDOM NO. FROM A UNIFORM DISTRIBUTION
C         ON THE INTERVAL (0.,1.)
C
   10 CONTINUE
      XINPUT=0.
      ETA1=RNUMF(XINPUT)
C
C     MAP EACH HALF OF THE INTERVAL (0.,1.) INTO (0.,1.)
C
      ETA=ABS(1.-2.*ETA1)
C
C     FIND THE ERROR FUNCTION BOUNDS ON THE RANDOM NO.
C
      DO 20 L=2,NERFV
   20 IF(ETA.LE.ERFV(L)) GO TO 30
C
C     IF NO BOUNDS ARE FOUND,THE RANDOM NO. IS TOO CLOSE TO 1.
C         GO BACK AND TRY ANOTHER ONE.
C
      GO TO 10
   30 CONTINUE
C
C     BOUNDS HAVE BEEN FOUND. INTERPOLATE THE ERROR FUNCTION INVERSE.
C
      ERFINV=DXEF*(FLOAT(L-2)+(ETA-ERFV(L-1))/(ERFV(L)-ERFV(L-1)))
C
C     CALCULATE THE CORRECTION
C
      CORR=ERFINV*2.*SQRT(DT*SIG(IJ))
C
C     ADD ON THE SIGN OF THE CORRECTION
C
      IF(ETA1.GT..5) CORR=-CORR
C
C     GO CORRECT THE PROPER COORDINATE
C
      GO TO (40,60),IPASS
C
C     X (R) COORDINATE
C
   40 CONTINUE
      XJ=XJ+CORR
   50 CONTINUE
C
C     GO BACK AND DO THE Y (Z) COORDINATE
C
      IPASS=2
      GO TO 10
   60 CONTINUE
C
C     Y (Z) COORDINATE
C
      YJ=YJ+CORR
      RETURN
      END
```

```
      SUBROUTINE REZONE
C
C     ROUTINE TO CALCULATE THE GRID VELOCITIES,NEW GRID POSITIONS,
C         AND RESTORE THE AMBIENT ATMOSPHERE TO HYDROSTATIC
C         EQUILIBRIUM
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
```

```
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ===== END COMDECK YAQEQ      =====
*     ===== END COMDECK YSTORE     =====
      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
      FSTF=0.01
      ZRS=ZE=SUMRS=SUME=0.0
      DMN=1.0E+20
      CALL START
      DO 20 J=2,JP1
      DO 10 I=1,IBAR
      DEL=(Y(IJP)-Y(IJ))**2+(X(IJP)-X(IJ))**2
      DMN=AMIN1(DMN,DEL)
      RDEL=1.0/DEL
      DELE=E(IJ)-0.156
      SUME=SUME+DELE
      ZE=ZE+Y(IJ)*DELE
      SUMRS=SUMRS+RDEL
      ZRS=ZRS+Y(IJ)*RDEL
      IJ=IJ+NQ
      IJP=IJP+NQ
   10 CONTINUE
      CALL LOOP
   20 CONTINUE
      ZE=ZE/SUME
      ZRS=ZRS/SUMRS
      FC3=FCX=FCP2=0.0
      CALL START
      DO 40 J=2,JP1
      DO 30 I=1,IBAR
      IF(J.LE.4) FC3=AMAX1(FC3,ABS(VL(IJ)))
      IF(J.GE.JP1-11) FCP2=AMAX1(FCP2,ABS(VL(IJ)))
      IF(I.GE.IBAR-3) FCX=AMAX1(FCX,ABS(UL(IJ)))
      IJ=IJ+NQ
   30 CONTINUE
      CALL LOOP
   40 CONTINUE
      FCP2=AMAX1(0.,0.2*RDT*(ZORIG-(Y(IJ)-PTOP)/PRITE)*PRITE)
      FCX=AMAX1(0.,0.2*RDT*((ZORIG-X(IJ+NQIB)/PRITE)*PRITE))
      CALL START
      DO 150 J=2,JP2
      DO 140 I=1,IP1
      IPJ=IJ+NQ
      IMJ=IJ-NQ
      IF(I.EQ.1.OR.I.EQ.IP1.OR.J.EQ.2.OR.J.EQ.JP2) GO TO 50
      XAV=X(IPJ)+X(IMJ)+X(IJP)+X(IJM)
      UG(IJ)=0.5*(UL(IJ)+U(IJ))+FSTF*(XAV-4.*X(IJ))*RDT
      YAV=Y(IPJ)+Y(IMJ)+Y(IJP)+Y(IJM)
      VG(IJ)=0.5*(VL(IJ)+V(IJ))+FSTF*(YAV-4.*Y(IJ))*RDT
      IF(J.EQ.3) UG(IJM)=UG(IJ)
```

```
      IF(I.EQ.2) VG(IMJ)=VG(IJ)
      GO TO 130
   50 CONTINUE
      IF(I.NE.1) GO TO 60
      UG(IJ)=0.0
      GO TO 70
   60 IF(I.NE.IP1) GO TO 70
      UG(IJ)=FCX
      VG(IJ)=VG(IMJ)
   70 IF(J.NE.2) GO TO 80
      VG(IJ)=-FC3
      GO TO 90
   80 IF(J.NE.JP2) GO TO 90
      UG(IJ)=UG(IJM)
      VG(IJ)=FCP2
   90 IF(I.NE.1.OR.J.NE.2) GO TO 100
      VG(IJ)=-FC3
  100 IF(I.NE.IP1.OR.J.NE.2) GO TO 110
      UG(IJ)=FCX
      VG(IJ)=-FC3
  110 IF(I.NE.1.OR.J.NE.JP2) GO TO 120
      VG(IJ)=FCP2
  120 IF(I.NE.IP1.OR.J.NE.JP2) GO TO 130
      UG(IJ)=FCX
      VG(IJ)=FCP2
  130 CONTINUE
      IJ=IPJ
      IJP=IJP+NQ
      IJM=IJM+NQ
  140 CONTINUE
      CALL LOOP
  150 CONTINUE
      CALL DONE
      CALL START
      DO 180 J=2,JP2
      DO 170 I=1,IP1
      X(IJ)=X(IJ)+UG(IJ)*DT
      IF(J.NE.2) GO TO 160
      IF(Y(IJ)+VG(IJ)*DT.LE.0.0) VG(IJ)=-Y(IJ)*RDT
  160 CONTINUE
      Y(IJ)=Y(IJ)+VG(IJ)*DT
      R(IJ)=X(IJ)*CYL+OMCYL
  170 IJ=IJ+NQ
      CALL LOOP
  180 CONTINUE
      CALL DONE
      CALL FILMCO
      CALL START
      ETMP=REZSIE
      ROTMP=REZRON
      CONCJ=0.
      CALL AIR
      XX=GMONE*REZSIE
      YY=ABS(GZ)/XX
      DO 200 J=2,JP1
      DO 190 I=1,IBAR
      IPJ=IJ+NQ
```

```
      IPJP=IJP+NQ
      Y4=0.25*(Y(IJP)+Y(IPJP)+Y(IJ)+Y(IPJ))
      IF(J.EQ.2) ROL(IJM)=REZRON*EXP((REZY0-(Y(IJ)+Y(IPJ)-Y4))*YY)
      IF(I.EQ.IBAR) ROL(IPJ)=REZRON*EXP((REZY0-(Y(IPJ)+Y(IPJP)-Y4))*YY)
      IF(J.EQ.JP1) ROL(IJP)=REZRON*EXP((REZY0-(Y(IJP)+Y(IPJP)-Y4))*YY)
      IJM=IJM+NQ
      IJP=IJP+NQ
  190 IJ=IPJ
      CALL LOOP
  200 CONTINUE
      CALL DONE
      RETURN
      END
```

```
      SUBROUTINE RTAPE
C
C     ROUTINE TO READ IN ONE DUMP OF A YAQUI DUMP FILE (ALL BUT
C         THE HEADER RECORD)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM       -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM       -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
```

```
C     READ REST OF DUMP, FIRST CHECK THE PARAMETERS,
C
      CALL PCHCK
C
C     READ IN THE LCM ARRAY STORAGE
C
      CALL LCBUFF(0,NLC,7,0,1,IERROR)
      IF(IERROR.NE.0) GO TO 20
      GO TO 30
   10 CALL UNCLE(1,6HYARSRT,18,18HSCBUFF INPUT ERROR)
   20 CALL UNCLE(4,6HYARSRT,18,18HLCBUFF INPUT ERROR)
   30 CONTINUE
C
C     SEE IF THERE ARE ANY PARTICLES
C
      IF(NPT.LE.0) GO TO 40
C
C     YES, READ THEIR RECORDS,
C
      CALL LCBUFF(NLCP1,NPS,7,0,1,IERROR)
      IF(IERROR.NE.0) GO TO 20
      CALL SCBUFF(ITAB,ITABP,7,0,1,IERROR)
      IF(IERROR.NE.0) GO TO 10
C
C     SEE IF THERE IS ANY TIME-DEPENDENT PARTICLE DATA
C
      IF(NILNG.LE.0) GO TO 40
C
C     YES, READ IT IN,
C
      CALL LCBUFF(NLCP1+NLCP2,2*NP1*NILNG,7,0,1,IERROR)
      IF(IERROR.NE.0) GO TO 20
   40 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE R1ROW
C
C     ROUTINE TO READ ROW J FROM LCM TO SCM BUFFER ONE
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
```

```
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
      DATA NE/0/
      IEC=(J-1)*NQI
      CALL ECRD(AASC,IEC,NQI,IDUM)
      RETURN
      END
```

```
      SUBROUTINE SETIJ
C
C     ROUTINE TO SET THE SCM POINTER TO THE ITH ELEMENT IN ROW J,
C         ASSUMING ROW J IS IN SCM BUFFER ONE
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
```

```
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
      IJ=(I-1)*NQ+1
      RETURN
      END
```

```
      SUBROUTINE START
C
C     ROUTINE TO TRANSFER DATA BETWEEN LARGE CORE AND SMALL CORE IN
C         ORDER TO MINIMIZE SMALL CORE REQUIREMENTS
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
```

```
          6 ,TMASSV,WMAXEF,RMINEF,TSTRTO
           COMMON/YSC2/ZZ
C          COMMON/YSC4/ITAB(ITABP)
           COMMON/YSC4/ITAB(1000)
           COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*          ***** END COMDECK YAQSC      *****
           DATA NE/0/
C
C          READ IN THE FIRST THREE ROWS
C
C          IJPS IS THE SCM POINTER TO WHERE THE NEXT ROW IS TO BE READ
C              INTO SCM. ONE NEEDS TO THINK OF THE SMALL CORE AREA AS BEING
C              DIVIDED INTO THREE BUFFERS,SAY A,B,AND C. INITIALLY,ROW J=1
C              IS READ INTO BUFFER A,ROW J=2 INTO BUFFER B,AND ROW J=3 INTO
C              BUFFER C. WHEN A REQUEST IS MADE FOR ROW J=4,BUFFER A WHICH
C              CURRENTLY HOLDS ROW J=1 IS WRITTEN OUT TO LCM AND ROW J=4 IS
C              READ INTO BUFFER A. THIS PROCESS CONTINUES WITH ONE ROW BEING
C              WRITTEN TO LCM AND ONE ROW REPLACING IT IN SCM SO THAT THREE
C              ROWS ARE ALWAYS AVAILABLE,ONE IN EACH BUFFER.
C          IECR IS THE LCM ADDRESS FROM WHICH DATA WILL BE READ TO SCM NEXT
C          IECW IS THE LCM ADDRESS TO WHICH DATA WILL BE WRITTEN FROM SCM NXT
C
           IJPS=1
           IECR=0
           IECW=0
C
C          READ THE FIRST ROW INTO SCM INTO BUFFER ONE
C
C          AASC IS BEGINNING OF THE THREE SCM BUFFERS
C          NQI IS THE NUMBER OF WORDS COMPRISING ONE ROW OF DATA
C          NE IS AN ERROR FLAG WHICH IS UNUSED ON THE 7600
C
           CALL ECRD(AASC(IJPS),IECR,NQI,NE)
C
C          INCREMENT THE LCM READ ADDRESS
C
           IECR=IECR+NQI
C
C          SET THE BUFFER POINTER TO THE SECOND BUFFER, OBVIOUSLY,
C              ISC2=NQI+1
C
           IJPS=ISC2
C
C          READ IN THE SECOND ROW
C
           CALL ECRD(AASC(IJPS),IECR,NQI,NE)
C
C          INCREMENT THE LCM READ ADDRESS AGAIN
C
           IECR=IECR+NQI
C
C          UPDATE THE POINTERS TO READ THE NEXT ROW INTO BUFFER THREE.
C
C          IJP IS THE POINTER TO THE ELEMENT (1,J+1)
C          IJ IS THE POINTER TO THE ELEMENT(1,J)
C          IJM IS THE POINTER TO (1,J-1)
C          IJMS IS THE SCM ADDRESS TO BE WRITTEN OUT NEXT
```

```
C     IBUF IS THE NEXT BUFFER TO BE USED WHEN THE SUBROUTINE IS ENTERED
C
   10 CONTINUE
      IJP=ISC3
      IJPS=ISC3
      IJ=ISC2
      IJM=1
      IJMS=1
      IBUF=1
C
C     FILL THE THIRD BUFFER
C
   20 CALL ECRD(AASC(IJPS),IECR,NQI,NE)
C
C     UPDATE THE LCM POINTER FOR THE NEXT READ
C
      IECR=IECR+NQI
      RETURN
C
C     *******************************************************************
      ENTRY LOOP
C     *******************************************************************
C
C     READ IN THE NEXT ROW. FIRST,FLUSH THE OLDEST BUFFER.
C
      CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
C
C     UPDATE THE LCM POINTER FOR THE NEXT WRITE
C
      IECW=IECW+NQI
C
C     PREPARE THE POINTERS FOR THE NEXT READ DEPENDING ON WHICH BUFFER
C         IS TO BE FILLED
C
      GO TO (30,40,10),IBUF
C
C     PREPARE TO FILL BUFFER ONE
C
   30 CONTINUE
      IJP=1
      IJPS=1
      IJ=ISC3
      IJM=ISC2
      IJMS=ISC2
C
C     INDICATE THAT BUFFER TWO IS TO BE FILLED NEXT TIME
C
      IBUF=2
C
C     GO FILL BUFFER ONE
C
      GO TO 20
C
C     PREPARE TO FILL BUFFER TWO
C
   40 CONTINUE
      IJP=ISC2
```

```
      IJPS=ISC2
      IJ=1
      IJM=ISC3
      IJMS=ISC3
C
C     INDICATE THAT BUFFER THREE IS TO BE FILLED NEXT TIME
C
      IBUF=3
C
C     GO FILL BUFFER TWO
C
      GO TO 20
C
C     ******************************************************************
      ENTRY DONE
C     ******************************************************************
C
C     LOOP IS COMPLETED, WRITE OUT CONTENTS OF THE LAST TWO BUFFERS
C         THAT WERE LOADED, WRITE OUT BUFFER THAT WOULD HAVE BEEN
C         FILLED NEXT.
C
      CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
      IECW=IECW+NQI
C
C     DETERMINE WHICH IS THE OTHER BUFFER TO BE EMPTIED
C
      GO TO (50,60,70),IBUF
C
C     BUFFER ONE HAS BEEN WRITTEN OUT, WRITE OUT BUFFER TWO.
C
   50 IJMS=ISC2
      GO TO 80
C
C     BUFFER TWO HAS BEEN WRITTEN OUT, WRITE OUT BUFFER THREE.
C
   60 IJMS=ISC3
      GO TO 80
C
C     BUFFER THREE HAS BEEN WRITTEN OUT, WRITE OUT BUFFER ONE.
C
   70 IJMS=1
   80 CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
      RETURN
      END
```

```
      SUBROUTINE STARTD
C
C     THIS ROUTINE (AND ITS ENTRY POINT LOOPD) ALLOW ONE TO LOOP
C         BACKWARDS THROUGH THE MESH USING TWO SCM BUFFERS, ROWS J
C         AND J-1 ARE MADE AVAILABLE, THE CALL TO STARTD READS ROW JP2
C         INTO BUFFER TWO AND ROW JP1 INTO BUFFER ONE, AFTER STARTD,
```

```
      C         IJ POINTS TO CELL (IP1,JP2), SEE DOCUMENTATION IN SUBROUTINE
      C         START FOR SOME OF THE VARIABLE DEFINITIONS.
      C
      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
      C
      *      ----- BEGIN COMDECK PARAM      -----
             COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
            1 NLCP3,NLCP4,IFLMSZ
      *      ----- END COMDECK PARAM      -----
      *      ----- BEGIN COMDECK YAQSC      -----
             LOGICAL RESTRT,FILM,PAPER,TURB
             REAL LAM,MU
      C      COMMON/YSC1/AASC(NSCP1)
             COMMON/YSC1/AASC(9600)
             COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
            1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
            2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
            3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
            4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
             COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
            1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
            2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
            3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
             COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
            1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
            2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
            3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
            4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
            5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
            6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
             COMMON/YSC2/ZZ
      C      COMMON/YSC4/ITAB(ITABP)
             COMMON/YSC4/ITAB(1000)
             COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
      *      ----- END COMDECK YAQSC      -----
             DATA NE/0/
      C
      C      SET INITIAL SCM READ ADDRESS (BUFFER TWO)
      C
             IJMS=ISC2
      C
      C      SET NEXT LCM READ AND WRITE ADDRESSES.
      C
      C      ITV IS JP1*NQI OR THE LCM ADDRESS OF THE BEGINNING OF ROW JP2
      C
             IECR=ITV
             IECW=ITV
      C
      C      READ ROW JP2 INTO BUFFER TWO
      C
             CALL ECRD(AASC(IJMS),IECR,NQI,IDUM)
      C
      C      DECREMENT THE READ ADDRESS
      C
             IECR=IECR-NQI
      C
```

```
C      PREPARE TO READ IN ROW J-1 INTO BUFFER ONE
C
C      IJM POINTS TO (IP1,J-1) (THE END OF BUFFER ONE)
C
    10 CONTINUE
       IJM=ISC2-NQ
       IJMS=1
C
C      IBUF IN THIS ROUTINE IS THE LAST BUFFER THAT WAS FILLED
C
       IBUF=1
C
C      IJ=IJM+NQI (THE LAST CELL OF BUFFER TWO)
C
       IJ=IJM+NQI
       IJPS=ISC2
C
C      CHECK TO SEE IF THE BOTTOM OF THE MESH HAS BEEN REACHED
C
    20 IF(IECR.LT.0) GO TO 40
C
C      NO, GO AHEAD AND READ IN THE NEXT BUFFER
C
       CALL ECRD(AASC(IJMS),IECR,NQI,IDUM)
C
C      DECREMENT THE READ ADDRESS
C
       IECR=IECR-NQI
    30 RETURN
C
C      BOTTOM OF THE MESH HAS BEEN REACHED, SET A FLAG (IBUF=3),GO
C          WRITE OUT ROW J=2,AND QUIT
C
    40 IBUF=3
       GO TO 50
C
C      *****************************************************************
       ENTRY LOOPD
C      *****************************************************************
C
C      ENTRY TO CYCLE THROUGH ONE MORE ROW OF THE MESH IN A DESCENDING
C          FASHION AS DESCRIBED ABOVE
C
C      WRITE OUT ROW J AND READ ROW J-2 INTO ITS BUFFER
C
    50 CONTINUE
       CALL ECWR(AASC(IJPS),IECW,NQI,IDUM)
C
C      DECREMENT THE WRITE ADDRESS
C
       IECW=IECW-NQI
C
C      SET UP THE POINTERS FOR THE NEXT READ DEPENDING ON WHICH BUFFER
C          IS TO BE FILLED, IF IBUF=3,ONLY A RETURN IS EXECUTED SINCE
C          THE MESH IS FINISHED.
C
       GO TO (60,10,30),IBUF
```

```
C
C     PREPARE POINTERS TO FILL BUFFER TWO
C
   60 CONTINUE
      IBUF=2
      IJ=ISC2-NQ
      IJPS=1
      IJM=IJ+NQI
      IJMS=ISC2
C
C     GO DO THE ACTUAL READ
C
      GO TO 20
      END
```

```
      SUBROUTINE TICBOX
C
C     ROUTINE TO DRAW A BOX AROUND THE GRID AREA DISPLAYED
C         AND LABEL THE AXES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
```

```
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ***** END COMDECK YAQSC       *****
*     ***** BEGIN COMDECK PCALL     *****
      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*     ***** END COMDECK PCALL       *****
      COMMON/NCODEC/IMSIGN,XMAN,IESIGN,IEXP
      DATA BCD/1H /
C
C     DRAW THE PLOT FRAME
C
      CALL DRV(IXL,IYT,IXR,IYT)
      CALL DRV(IXR,IYT,IXR,IYB)
      CALL DRV(IXR,IYB,IXL,IYB)
      CALL DRV(IXL,IYB,IXL,IYT)
C
C     IX2,IX3,IY2,AND IY3 ARE THE RASTER COORDINATES OF THE ENDS OF
C         THE TIC MARKS TO BE DRAWN ON THE PLOTTING RECTANGLE
C
      IX2=IXL+8
      IX3=IXR-8
      IY2=IYB-8
      IY3=IYT+8
C
C     ESTABLISH TOP AND BOTTOM FOR PLOT INCREMENTS. ROUND TOP AND
C     BOTTOM GRID VALUES TO THE NEAREST POWER OF 10. (YUPI,YLBI).
C
      IXX=ALOG10(YUP)
      DTIC=10.**IXX
C
C     DETERMINE THE Y EXTREMA TO USE TO SET DTIC,THE TIC INCREMENT.
C         THE Y MINIMUM,YLBI,IS SET SUCH THAT IT IS LESS THAN YLB
C         AND EQUAL TO AN INTEGRAL MULTIPLE OF THE POWER OF TEN
C         NEAREST TO BUT LESS THAN YUP. THE Y MAXIMUM,YUPI,IS
C         DETERMINED SIMILARLY BUT SO THAT IT IS GREATER THAN YUP.
C
C     ASSUME THAT YLB IS .GE.0
C
      YLBI=0.
      IF(DTIC.GE.YLB) GO TO 20
      DO 10 I=1,9
      YLBI=YLBI+DTIC
      IF(YLBI.LT.YLB) GO TO 10
      YLBI=YLBI-DTIC
      GO TO 20
   10 CONTINUE
      CALL UNCLE(1,6HTICBOX,24,24HCOULD NOT DETERMINE YLBI)
   20 CONTINUE
      YUPI=DTIC
      DO 30 I=2,10
      YUPI=YUPI+DTIC
      IF(YUPI.GE.YUP) GO TO 40
   30 CONTINUE
      CALL UNCLE(1,6HTICBOX,24,24HCOULD NOT DETERMINE YUPI)
   40 CONTINUE
C
C     BOUNDS ESTABLISHED. DETERMINE THE LABELLING INCREMENT BASED
C         ON 10 INCREMENTS.
```

```
C
      DTIC=(YUPI-YLBI)*.1
   50 CONTINUE
C
C     COMPUTE HOW MANY TICS WILL BE DRAWN
C
      ITIC=0
      YTICT=YLBI
   60 CONTINUE
      IF(YTICT.GE.YLB) GO TO 70
      YTICT=YTICT+DTIC
      GO TO 60
   70 CONTINUE
      YLBI=YTICT-DTIC
   80 CONTINUE
      IF(YTICT.GE.YUP) GO TO 90
      ITIC=ITIC+1
      YTICT=YTICT+DTIC
      GO TO 80
   90 CONTINUE
C
C     MUST BE AT LEAST 6 TICS OR DTIC WILL BE REDUCED
C
      IF(ITIC.GE.6) GO TO 100
      DTIC=DTIC/2.
      GO TO 50
  100 CONTINUE
C
C     YTIC IS THE Y-COORDINATE OF THE LABEL
C
      YTIC=YLBI
  110 CONTINUE
      IF(YTIC.GE.YLB) GO TO 120
      YTIC=YTIC+DTIC
      GO TO 110
C
C     COMPUTE RASTER COORDINATE FOR YTIC
C
  120 CONTINUE
      IY1=FIYB+(YTIC-YLB)*YCONVP
C
C     IF THIS LABEL WOULD PUT US OUTSIDE OF THE GRID,CONSIDER THE
C         Y-LABELLING FINISHED
C
      IF(IY1.LT.IYT) GO TO 210
C
C     DRAW TICS ON THE LEFT AND RIGHT SIDES OF THE FRAME
C
      CALL DRV(IXL,IY1,IX2,IY1)
      CALL DRV(IX3,IY1,IXR,IY1)
C
C     PREPARE AND OUTPUT THE LABEL ITSELF
C
      IMSIGN=1H
      IF(YTIC.GT.0.) GO TO 140
      IF(YTIC.LT.0.) GO TO 130
      IMSIGN=2H0.
```

```
      CALL NCODE(8,6H(6XA2),IMSIGN,1,BCD)
      GO TO 200
  130 CONTINUE
      XLOG=ALOG10(ABS(YTIC))
      IMSIGN=1H-
      GO TO 150
  140 CONTINUE
      XLOG=ALOG10(YTIC)
  150 CONTINUE
      IEXP=XLOG
      XMAN=XLOG-FLOAT(IEXP)
      IF(XMAN.GE.0.) GO TO 160
      XMAN=XMAN+1.
      IEXP=IEXP-1
  160 CONTINUE
      XMAN=1.000001*XMAN
      IF(XMAN.LT.1.) GO TO 170
      XMAN=XMAN-1.
      IEXP=IEXP+1
  170 CONTINUE
      XMAN=10.**XMAN
      IESIGN=1H+
      IF(IEXP.GE.0) GO TO 180
      IEXP=-IEXP
      IESIGN=1H-
  180 CONTINUE
      IF(IEXP.GE.10) GO TO 190
      CALL NCODE(8,17H(1XA1,F4.2,A1,I1),IMSIGN,4,BCD)
      GO TO 200
  190 CONTINUE
      CALL NCODE(8,15H(A1,F4.2,A1,I2),IMSIGN,4,BCD)
  200 CONTINUE
      CALL TSP(1,IY1,8,BCD)
C
C     INCREMENT THE TIC VALUE AND PROCEED
C
      YTIC=YTIC+DTIC
      GO TO 120
C
C     SAME PROCEDURE AS FOR Y-AXIS
C
  210 CONTINUE
C
C     USE THE SAME LABELLING INCREMENT FOR THE X-AXIS
C
      XTIC=0.
  220 CONTINUE
      IX1=FIXL+(XTIC-XL)*XCONVP
      IF(IX1.GT.IXR) RETURN
      CALL DRV(IX1,IYB,IX1,IY2)
      CALL DRV(IX1,IY3,IX1,IYT)
      JRITE=IYB+9
      IRITE=IX1-29
      IMSIGN=1H
      IF(XTIC.GT.0.) GO TO 240
      IF(XTIC.LT.0.) GO TO 230
      IMSIGN=2H0.
```

```
      CALL NCODE(8,9H(3XA2,3X),IMSIGN,1,BCD)
      GO TO 300
  230 CONTINUE
      XLOG=ALOG10(ABS(XTIC))
      IMSIGN=1H-
      GO TO 250
  240 CONTINUE
      XLOG=ALOG10(XTIC)
  250 CONTINUE
      IEXP=XLOG
      XMAN=XLOG-FLOAT(IEXP)
      IF(XMAN.GE.0.) GO TO 260
      XMAN=XMAN+1.
      IEXP=IEXP-1
  260 CONTINUE
      XMAN=1.000001*XMAN
      IF(XMAN.LT.1.) GO TO 270
      XMAN=XMAN-1.
      IEXP=IEXP+1
  270 CONTINUE
      XMAN=10.**XMAN
      IESIGN=1H+
      IF(IEXP.GE.0) GO TO 280
      IEXP=-IEXP
      IESIGN=1H-
  280 CONTINUE
      IF(IEXP.GE.10) GO TO 290
      CALL NCODE(8,17H(1XA1,F4.2,A1,I1),IMSIGN,4,BCD)
      GO TO 300
  290 CONTINUE
      CALL NCODE(8,15H(A1,F4.2,A1,I2),IMSIGN,4,BCD)
  300 CONTINUE
      CALL TSP(IRITE,JRITE,8,BCD)
      XTIC=XTIC+DTIC
      GO TO 220
      END
```

```
      SUBROUTINE TRBCOR
C
C     ROUTINE TO CALCULATE TURBULENCE CORRECTIONS
C
C     ORIGINALLY WRITTEN BY HANS RUPPEL,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM        -----
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
```

```
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*       ----- END COMDECK YAQDIM      -----
*       ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNPO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*       ----- END COMDECK YAQSC       -----
*       ----- BEGIN COMDECK YAQEQ     -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DFLSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*       ----- END COMDECK YAQEQ       -----
*       ----- END COMDECK YSTORE      -----
      CALL START
      DO 20 J=2,JP1
      DO 10 I=1,IBAR
      IPJP=IJP+NQ
      IPJ=IJ+NQ
      IMJ=IJ-NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
```

```
      X2=X(IPJP)
      Y2=Y(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      Y21=Y2-Y1
      Y32=Y3-Y2
      Y43=Y4-Y3
      Y14=Y1-Y4
      X12=X1-X2
      X23=X2-X3
      X34=X3-X4
      X41=X4-X1
      AVXSV(IJ)=.25*(X1+X2+X3+X4)
      AVYSV(IJ)=.25*(Y1+Y2+Y3+Y4)
      R12=R(IPJ)+R(IPJP)
      R23=R(IJP)+R(IPJP)
      R34=R(IJP)+R(IJ)
      R41=R(IJ)+R(IPJ)
      C=CONC(IJ)
      SN=SIG(IJ)
      RC=RO(IJ)
      RCQ=RC*TUQ(IJ)
      RCS=RC*SIE(IJ)
      AC=AREA(IJ)
      IF(I.EQ.1) IMJ=IJ
      IF(I.EQ.IBAR) IPJ=IJ
      IF(J.EQ.2) IJM=IJ
      IF(J.EQ.JP1) IJP=IJ
      Z1=R12*(Y21**2+X12**2)*(SIG(IPJ)+SN)/(AREA(IPJ)+AC)
      Z2=R23*(Y32**2+X23**2)*(SIG(IJP)+SN)/(AREA(IJP)+AC)
      Z3=R34*(Y43**2+X34**2)*(SIG(IMJ)+SN)/(AREA(IMJ)+AC)
      Z4=R41*(Y14**2+X41**2)*(SIG(IJM)+SN)/(AREA(IJM)+AC)
      CTEMP(IJ)=0.5*((CONC(IPJ)-C)*Z1+(CONC(IJP)-C)*Z2+(CONC(IMJ)-C)*Z3+
     1 (CONC(IJM)-C)*Z4)
      DLSROI(IJ)=0.
      DLSROQ(IJ)=0.
      R1=RO(IPJ)
      R2=RO(IJP)
      R3=RO(IMJ)
      R4=RO(IJM)
      ROL(IJ)=0.5*((R1-RC)*Z1+(R2-RC)*Z2+(R3-RC)*Z3+(R4-RC)*Z4)
      DLSROQ(IJ)=.5*((R1*TUQ(IPJ)-RCQ)*Z1+(R2*TUQ(IJP)-RCQ)*Z2+(R3*TUQ(
     1 IMJ)-RCQ)*Z3+(R4*TUQ(IJM)-RCQ)*Z4)
      DLSROI(IJ)=0.5*((R1*SIE(IPJ)-RCS)*Z1+(R2*SIE(IJP)-RCS)*Z2+(R3*SIE(
     1 IMJ)-RCS)*Z3+(R4*SIE(IJM)-RCS)*Z4)
      IJ=IJ+NQ
      IJM=IJM+NQ
      IJP=IJP+NQ
   10 CONTINUE
      CALL LOOP
   20 CONTINUE
      CALL DONE
      CALL START
      DO 80 J=2,JP1
      DO 70 I=1,IBAR
```

```
      IPJ=IJ+NQ
      IMJ=IJ-NQ
      X1=AVXSV(IPJ)
      X2=AVXSV(IJP)
      X3=AVXSV(IMJ)
      X4=AVXSV(IJM)
      Y1=AVYSV(IPJ)
      Y2=AVYSV(IJP)
      Y3=AVYSV(IMJ)
      Y4=AVYSV(IJM)
      P1=P(IPJ)
      P2=P(IJP)
      P3=P(IMJ)
      P4=P(IJM)
      RO1=RO(IPJ)
      RO2=RO(IJP)
      RO3=RO(IMJ)
      RO4=RO(IJM)
      IF(I.NE.1) GO TO 30
      X3=-X1
      Y3=Y1
      P3=P1
      RO3=RO1
   30 IF(J.NE.2) GO TO 40
      X4=X2
      Y4=-Y2
      P4=P2
      RO4=RO2
   40 IF(I.NE.IBAR) GO TO 50
      X1=-X3
      Y1=Y3
      P1=P3
      RO1=RO3
   50 IF(J.NE.JP1) GO TO 60
      X2=X4
      Y2=-Y4
      P2=P4
      RO2=RO4
   60 CONTINUE
      RA=1./((X1-X3)*(Y2-Y4)-(X2-X4)*(Y1-Y3))
      Y31=(Y3-Y1)*RA
      Y42=(Y4-Y2)*RA
      X13=(X1-X3)*RA
      X24=(X2-X4)*RA
      DPDR=(P2-P4)*Y31+(P3-P1)*Y42
      DPDZ=(P2-P4)*X13+(P3-P1)*X24
      DRODR=(RO2-RO4)*Y31+(RO3-RO1)*Y42
      DRODZ=(RO2-RO4)*X13+(RO3-RO1)*X24
      GRROP(IJ)=DRODR*DPDR+DRODZ*DPDZ
      GRROR(IJ)=DRODR
      GRROZ(IJ)=DRODZ
      IJ=IPJ
      IJM=IJM+NQ
      IJP=IJP+NQ
   70 CONTINUE
      CALL LOOP
   80 CONTINUE
```

```
      CALL DONE
      CALL START
      CALL LOOP
      DO 120 J=3,JP1
      DO 110 I=1,IBAR
      IMJ=IJ-NQ
      IMJM=IJM-NQ
      IF(I.EQ.1) IMJ=IJ
      IF(I.EQ.1) IMJM=IJM
      SIJ=0.25*(SIG(IJ)+SIG(IMJ)+SIG(IMJM)+SIG(IJM))
      VVA=0.25*(1./RVOL(IJ)+1./RVOL(IMJ)+1./RVOL(IMJM)+1./RVOL(IJM))*RM(
     1 IJ)
      X1=AVXSV(IJM)
      X2=AVXSV(IJ)
      X3=AVXSV(IMJ)
      X4=AVXSV(IMJM)
      Y1=AVYSV(IJM)
      Y2=AVYSV(IJ)
      Y3=AVYSV(IMJ)
      Y4=AVYSV(IMJM)
      IF(I.NE.1) GO TO 90
      X3=-X2
      X4=-X1
      Y3=Y2
      Y4=Y1
   90 CONTINUE
      RXXA=1./((X1-X3)*(Y2-Y4)+(X2-X4)*(Y3-Y1))
      XXA=((RO(IJM)-RO(IMJ))*(Y2-Y4)+(RO(IJ)-RO(IMJM))*(Y3-Y1))*RXXA*SIJ
      YYA=((RO(IJ)-RO(IMJM))*(X1-X3)+(RO(IMJ)-RO(IJM))*(X2-X4))*RXXA*SIJ
      XXA=XXA*VVA
      YYA=YYA*VVA
      IF(NCYC.EQ.NCQ+1) GO TO 100
      UTIL(IJ)=UTIL(IJ)+XXA-GRSV(IJ)
      VTIL(IJ)=VTIL(IJ)+YYA-GZSV(IJ)
  100 CONTINUF
      GPSV(IJ)=XXA
      GZSV(IJ)=YYA
      IJ=IJ+NQ
      IJM=IJM+NQ
  110 CONTINUE
      CALL LOOP
  120 CONTINUE
      CALL DONE
      CALL START
      DO 140 J=2,JP1
      DO 130 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      SIJ=SIG(IJ)
      DALF=.25*DT*SIJ
      X13=X(IPJ)-X(IJP)
      X24=X(IPJP)-X(IJ)
      Y13=Y(IPJ)-Y(IJP)
      Y24=Y(IPJP)-Y(IJ)
      U13=U(IPJ)+U(IJP)
      U24=U(IPJP)+U(IJ)
      V13=V(IPJ)+V(IJP)
```

```
      V24=V(IPJP)+V(IJ)
      R13=(R(IPJ)+R(IJP))*DALF
      R24=(R(IPJP)+R(IJ))*DALF
      ZZ=GRROZ(IJ)
      ZR=GRROR(IJ)
      H13=0.5*(U13*ZZ+V13*ZR)
      H24=0.5*(U24*ZZ+V24*ZR)
      H1=(H24*Y24-V24*ZZ*X24)*R24
      H2=(-H13*Y13+V13*ZZ*X13)*R13
      H3=(-H24*X24+U24*ZR*Y24)*R24
      H4=(H13*X13-U13*ZR*Y13)*R13
      UTIL(IPJ)=UTIL(IPJ)-H3*RM(IPJ)
      UTIL(IPJP)=UTIL(IPJP)-H4*RM(IPJP)
      UTIL(IJP)=UTIL(IJP)+H3*RM(IJP)
      UTIL(IJ)=UTIL(IJ)+H4*RM(IJ)
      VTIL(IJ)=VTIL(IJ)+H2*RM(IJ)
      VTIL(IJP)=VTIL(IJP)+H1*RM(IJP)
      VTIL(IPJ)=VTIL(IPJ)-H1*RM(IPJ)
      VTIL(IPJP)=VTIL(IPJP)-H2*RM(IPJP)
C     *******TURBULENCE ENERGY EQUATION*******
      RIJ=RO(IJ)
      RECRHO=1./RIJ
      D=DELSM(IJ)
      XPR1=2.*SIJ/(TUS(IJ)**2*QLEVEL)
      TUQ(IJ)=RIJ*TUQ(IJ)+DT*(SIJ*(CAPGAM(IJ)*RIJ-GRROP(IJ)*RECRHO)+
     1 DLSROQ(IJ)*RVOL(IJ))
      TUQ(IJ)=(TUQ(IJ)-TWTHRD*D*D*SIJ*RIJ*DT)/((1.+(XPR1+TWTHRD*D)*DT)*
     1 RIJ)
      IF(TUQ(IJ).LT.0.) TNEG=TNEG+TUQ(IJ)*RIJ/RVOL(IJ)
      IF(TUQ(IJ).LT.0.) TUQ(IJ)=0.
C     ******INTERNAL ENERGY EQUATION******
      SIE(IJ)=SIE(IJ)+DT*(DLSROI(IJ)*RVOL(IJ)*RECRHO+XPR1*TUQ(IJ))
      IJ=IJ+NQ
      IJM=IJM+NQ
      IJP=IJP+NQ
  130 CONTINUE
      CALL LOOP
  140 CONTINUE
      CALL DONE
      RETURN
      END
```

```
      SUBROUTINE TRBERF
C
C     ROUTINE TO CALCULATE THE ERROR FUNCTION VALUES TO BE USED
C         FOR INTERPOLATION TO COMPUTE THE PARTICLE TURBULENT
C         DIFFUSION MOTION
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK TRBDIF     -----
```

```
      COMMON/CTDIF/ERFV(21),NERFV,DXEF
*     ----- END COMDECK TRBDIF     -----
*     ----- BEGIN COMDECK YAQSC     -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,I8VW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC     -----
C     FIRST,INITIALIZE THE VARIOUS PARAMETERS
C
C     NERFV IS THE NO. OF VALUES OF ERF IN THE INTERPOLATION TABLES
C
      NERFV=21
C
C     DXEF IS THE SPACING IN X BETWEEN THE ERF(X) TABLE ENTRIES
C
      DXEF=WMAXEF/FLOAT(NERFV-1)
C
C     COMPUTE THE TABLE VALUES
C
      XEF=-DXEF
      DO 10 I=1,NERFV
      XEF=XEF+DXEF
      ERFV(I)=ERF(XEF)
   10 CONTINUE
      IF(ERFV(NERFV).LE.0.) CALL UNCLE(4,6HTRBERF,26,
     1 26HERRORS IN CALCULATING ERFV)
      RETURN
      END
```

```
      SUBROUTINE UNCLE(ISFLAG,RNAME,NCHAR,CHAR)
```

```
C
C     ERROR PROCESSING ROUTINE
C
C     ISFLAG = TYPE OF CALL
C            = 1 = FATAL ERROR,DUMP ARRAYS (NOT IMPLEMENTED)
C            = 2 = NORMAL EXIT
C            = 3 = DUMP ARRAYS AND RETURN (NOT IMPLEMENTED)
C            = 4 = FATAL ERROR,NO DUMP
C     RNAME  = NAME OF CALLING ROUTINE
C     NCHAR  = NO. OF CHARACTERS IN ERROR MESSAGE
C     CHAR   = ERROR MESSAGE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TO,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
      DIMENSION CHAR(1)
      DATA TP/0./
      DO 10 IPX=6,IFD,6
   10 WRITE(IPX,140)
      GO TO (60,20,40,60),ISFLAG
   20 CONTINUE
      DO 30 IPX=6,IFD,6
   30 WRITE(IPX,150)
      GO TO 80
   40 CONTINUE
      DO 50 IPX=6,IFD,6
   50 WRITE(IPX,130)
      GO TO 80
```

```
   60 CONTINUE
      DO 70 IPX=6,IFD,6
   70 WRITE(IPX,110)
   80 CONTINUE
      NWDS=NCHAR/10
      IF(MOD(NCHAR,10).NE.0) NWDS=NWDS+1
      DO 90 IPX=6,IFD,6
      WRITE(IPX,120) RNAME,(CHAR(II),II=1,NWDS)
   90 WRITE(IPX,140)
      CALL SECOND(TP)
      WRITE(59,100) TP
      IF(ISFLAG.EQ.2) CALL EXIT
      IF(ISFLAG.EQ.3) RETURN
      CALL EXIT(2)
C
  100 FORMAT(1H ,25HCODE TERMINATION AT CP = ,F10.4)
  110 FORMAT(1H ,20HFATAL ERROR IN YAQUI)
  120 FORMAT(1H ,24HUNCLE CALLED BY ROUTINE ,A6,10X9A10)
  130 FORMAT(1H ,16HSTART UNCLE DUMP)
  140 FORMAT(1H ,60(1H=))
  150 FORMAT(1H ,25HNORMAL EXIT THROUGH UNCLE)
      END
```

```
      SUBROUTINE VELPLT(VMAX,IFLAG)
C
C     ROUTINE TO DO VELOCITY PLOTS
C
C     VMAX IS THE MAXIMUM OF THE VELOCITY CURRENTLY BEING PLOTTED
C         IN ANY ONE DIRECTION.
C     IFLAG IS 0 FOR FLUID VELOCITY PLOTS SCALED BY VMAX. IT IS 1 FOR
C         UNSCALED FLUID VELOCITY PLOTS AND 2 FOR SCALED RELATIVE
C         VELOCITY PLOTS.
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
```

```
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ      -----
*     ----- END COMDECK YSTORE     -----
*     ----- BEGIN COMDECK PCALL      -----
      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*     ----- END COMDECK PCALL      -----
      DIMENSION ITITLE(5,3)
      DIMENSION IJL2(5),IJL3(5)
      EQUIVALENCE(ITITLE(6),IJL2),(ITITLE(11),IJL3)
      DATA ITITLE/49HFLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY/
      DATA IJL2/31HUNSCALED FLUID VELOCITY VECTORS,1H /
      DATA IJL3/30HVELOCITY OF GRID REL. TO FLUID,
     1 20H SCALED TO MAX. VEL./
C
C     SET UP THE VELOCITY VECTOR SCALING FACTOR.
C     VV IS .9 TIMES AN AVERAGE DR. THIS IS USED TO SCALE THE LENGTH
C         OF THE VELOCITY VECTOR SO IT IS NOT LONGER THAN THE EXTENT
C         OF AN AVERAGE CELL.
C
```

```
      DROU=VV/VMAX
C
C     SPACE FORWARD TO THE NEXT FILM FRAME
C
      CALL ADV(1)
C
C     LOOP OVER ALL REAL ZONES
C
      CALL START
      DO 20 J=2,JP2
      DO 10 I=1,IP1
      IF(IFLAG.EQ.1) DROU=0.5*VV/SQRT(U(IJ)**2+V(IJ)**2+EM10)
C
C     (IX1,IY1) IS THE LOCATION OF THE VERTEX.
C     (IX2,IY2) IS THE LOCATION OF THE END OF THE VELOCITY VECTOR.
C     IF EITHER ARE OUTSIDE OF THE PLOTTING RECTANGLE,SKIP THIS VERTEX.
C
      IY1=FIYB+(Y(IJ)-YLB)*YCONVP
      IF(IY1.GT.IYB.OR.IY1.LT.IYT) GO TO 10
      IY2=FIYB+(Y(IJ)+V(IJ)*DROU-YLB)*YCONVP
      IF(IY2.GT.IYB.OR.IY2.LT.IYT) GO TO 10
      IX1=FIXL+(X(IJ)-XL)*XCONVP
      IF(IX1.GT.IXR) GO TO 10
      IX2=FIXL+(X(IJ)-XL+U(IJ)*DROU)*XCONVP
      IF(IX2.GT.IXR) GO TO 10
C
C     DRAW THE VECTOR
C
      CALL DRV(IX1,IY1,IX2,IY2)
C
C     PLOT A PLUS (+) AT THE VERTEX POSITION
C
      CALL PLT(IX1,IY1,16)
   10 IJ=IJ+NQ
      CALL LOOP
   20 CONTINUE
C
C     LABEL THE PLOT WITH VMAX
C
      CALL LINCNT(60)
      IFLGP=IFLAG+1
      WRITE(IFD,40)(ITITLE(I,IFLGP),I=1,5),VMAX
      WRITE(IFD,30) JNM,NAME,T,NCYC
      RETURN
C
   30 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
   40 FORMAT(1H ,5A10/18X5HVMAX=,1PE12.5)
      END
```

```
      SUBROUTINE W1ROW
C
```

```
C     ROUTINE TO WRITE ROW J FROM SCM BUFFER ONE TO LCM
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
      DATA NE/0/
      IEC=(J-1)*NQI
      CALL ECWR(AASC,IEC,NQI,IDUM)
      RETURN
      END
```

```
      SUBROUTINE YAQUI2
C
C     MAIN ROUTINE FOR RUNNING CODE PHYSICS
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
```

```
*      ===== END COMDECK PARAM      =====
*      ===== BEGIN COMDECK YSTORE      =====
*      ===== BEGIN COMDECK YAQDIM      =====
       DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
      1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
      2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
      3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
      4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ===== END COMDECK YAQDIM      =====
*      ===== BEGIN COMDECK YAQSC      =====
       LOGICAL RESTRT,FILM,PAPER,TURB
       REAL LAM,MU
C      COMMON/YSC1/AASC(NSCP1)
       COMMON/YSC1/AASC(9600)
       COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
      1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
       COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
       COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
       COMMON/YSC2/ZZ
C      COMMON/YSC4/ITAB(ITABP)
       COMMON/YSC4/ITAB(1000)
       COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ===== END COMDECK YAQSC      =====
*      ===== BEGIN COMDECK YAQEQ      =====
       EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
      1 (8),E,ETIL,AREA,XR13K),
      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
      1 AVYSV,X24K)
        REAL M,MP,MPAR,MTIL
*      ===== END COMDECK YAQEQ      =====
*      ===== END COMDECK YSTORE      =====
*      ===== BEGIN COMDECK ASTORE      =====
       COMMON/ASTC/AT(100),FT(100)
       DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
       EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
```

```
      1 ),(AT(9),YCO),(FT,CON)
*       ***** END COMDECK ASTORE     *****
*       ***** BEGIN COMDECK PCALL     *****
        COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*       ***** END COMDECK PCALL      *****
        COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
        LOGICAL EDIT,LAST,DUMP
        REAL LAMD
        DATA IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX/0,0,4*0./
        DATA ISWTCH/-1/
C
C       INITIALIZE ON STARTUP OR RESTART
C
        NUMIT=0
        TE=0.
        DTR=0.
        TMASS=0.
        DTVSAV=0.
        DTCSAV=0.
C
C       INITIALIZE IF THE TURBULENCE IS ON
C
        IF(.NOT.TURB) GO TO 10
        TNEG=0.
        TP3=0.
        SAV1=0.
        TUQENG=0.
        TDQSAV=1.
     10 CONTINUE
C
C       DO CERTAIN INITIALIZATION ONLY DURING RESTART
C
        IF(.NOT.RESTRT) GO TO 20
        CALL CINIT
        IF(TURB) CALL TRBERF
        GO TO 30
     20 CONTINUE
C
C       STARTUP. CONTINUE INITIALIZATION.
C
        DTV=0.
        IDTV=0
        JDTV=0
        DTC=0.
        IDTC=0
        JDTC=0
C
C       -----------------------------------------------------------------
C       BEGIN LOOP OVER CYCLES
C       -----------------------------------------------------------------
C
     30 CONTINUE
C
C       INITIALIZE TURBULENCE QUANTITIES
C
        IF(.NOT.TURB) GO TO 40
        TNEGSV=TNEGSV+TNEG
```

```
      TNEG=0.
      EP1=TK+TI+TP3+TUQENG
      TIP=TI-(TMASS-TMASSV)*REZSIE
   40 CONTINUE
C
C     TOLD IS THE TIME AT THE END OF THE LAST CYCLE
C
      TOLD=T2
C
C     IF JUST FINISHING INITIALIZATION,GO DO AN EDIT FIRST
C
      IF(NCYC.EQ.0) GO TO 150
   50 CONTINUE
C
C     READY TO BEGIN NEXT CYCLE. INCREMENT THE CYCLE NO.
C
      NCYC=NCYC+1
C
C     SET THE NEW TIMESTEP. SAVE THE OLD ONE IN DTLAST.
C
      DTLAST=DT
C
C     ON THE FIRST CYCLE,SET DEFAULT DTC AND DTV
C
      IF(NCYC.NE.1) GO TO 60
      DTC=DT
      DTV=DT
   60 CONTINUE
C
C     ON THE SECOND CYCLE,BOOST DT BY A FACTOR OF 10
C
      IF(NCYC.NE.2) GO TO 70
      DT=10.*DT
      DTSV=DT
   70 CONTINUE
C
C     EXCEPT ON THE FIRST AND SECOND CYCLES,SET DT BASED ON DTV AND DTC
C
      IF(NCYC.GE.3) DT=AMIN1(DTV,DTC)
C
C     DO NOT ALLOW THE TIMESTEP TO INCREASE BY MORE THAN 25 PER CENT
C         OVER THE LAST CYCLE
C
      DTFAC=1.25
      DT=AMIN1(DTFAC*DTSV,DT)
C
C     IF WE WILL BE DOING AN EDIT AFTER THE NEXT CYCLE,ADJUST DT TO
C         MAKE T EXACTLY EQUAL TO THE EDIT TIME. DTPOS IS DT BEFORE
C         ANY SUCH ADJUSTMENT. DTSV IS THE TIMESTEP SAVED FOR FUTURE
C         REFERENCE.
C
      DTPOS=DT
      DTSV=DT
      IF(T+DT.GT.TOUT) DT=TOUT-T
      IF(DT.LT.1.E-8) DT=1.E-8
C
C     UPDATE T
```

```
C
      T=T+DT
      IF(DT.NE.DTPOS) T=TOUT
C
C     *********************************************************************
C     TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS
C     *********************************************************************
C
      IF(T.GT..6) ANC=.05
C
C     ILNG IS ONE IF THIS IS AN EDIT CYCLE
C
      ILNG=0
      IF(ABS(TOUT-T).LT.EM10) ILNG=1
C
C     SET QUANTITIES DERIVED FROM THE TIMESTEP
C
      RDT=1./DT
      DTO2=.5*DT
      DTO8=.125*DT
C
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     PHASE 1 - EXPLICIT LAGRANGIAN PART
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      CALL YPH1
C
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     PHASE 2 - PRESSURE ITERATION
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      CALL YPH2
C
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     PHASE 3,PART I - CONTINUOUS REZONE
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      CALL YPH3
C
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C     PHASE 3,PART II - FLUXING
C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
      CALL YFLUX
C
C     CALCULATE AN INFORMATIVE RELATIVE GRID-FLUID TIMESTEP AND THE
C         MAXIMUM OF THE DIFFERENCE BETWEEN GRID AND FLUID VELOCITIES,
C         ALSO CALCULATE THE CENTER OF THE CELL WITH MAXIMUM
C         TEMPERATURE.
C
      DTR=1.E30
      VMAX=-1.E10
      CALL START
      DO 110 J=2,JP2
      DO 100 I=1,IP1
      U(IJ)=UG(IJ)-U(IJ)
      V(IJ)=VG(IJ)-V(IJ)
```

```
      VMAX=AMAX1(VMAX,ABS(U(IJ)),ABS(V(IJ)))
      IF(J.NE.JTM.OR.I.NE.ITM) GO TO 80
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      XTMAX=.25*(X(IJ)+X(IPJ)+X(IJP)+X(IPJP))
      YTMAX=.25*(Y(IJ)+Y(IPJ)+Y(IJP)+Y(IPJP))
   80 CONTINUE
      IF(I.EQ.IP1) GO TO 90
      IPJ=IJ+NQ
      IF(U(IJ).NE.0..AND.V(IJ).NE.0.) DTR=AMIN1(DTR,ABS((X(IPJ)-X(IJ))/U
     1 (IJ)),ABS((Y(IJP)-Y(IJ))/V(IJ)))
   90 CONTINUE
      IJ=IJ+NQ
      IJP=IJP+NQ
  100 CONTINUE
      CALL LOOP
  110 CONTINUE
      CALL DONE
      DTR=.1*DTR
C
C     SEE IF THIS IS AN EDIT CYCLE
C
      IF(ILNG.NE.1) GO TO 120
C
C     YES, PLOT VELOCITY OF GRID RELATIVE TO THE FLUID.
C
      FIYB=FIYBO
      IYB=IYBO
      YCONVP=YCNVLD
      XCONVP=XCNVLD
      FIXR=FIXRO
      FIXL=FIXLO
      IXR=IXRO
      IXL=IXLO
      YLB=YB
      CALL ADV(1)
      CALL VELPLT(VMAX,2)
  120 CONTINUE
C
C     STORE U,V,AND RM INTO THEIR FINAL LOCATIONS AND ZERO RMP
C
      CALL START
      TP3=0.
      DO 140 J=2,JP2
      DO 130 I=1,IP1
      U(IJ)=UP(IJ)
      V(IJ)=VP(IJ)
      RM(IJ)=RMP(IJ)
      TP3=TP3-GZ*Y(IJ)/RM(IJ)
      RMP(IJ)=0.
  130 IJ=IJ+NQ
      CALL LOOP
  140 CONTINUE
      CALL DONE
C
C     MOVE THE PARTICLES IF THERE ARE ANY
C
```

```
      IF(NPT.GT.0) CALL PRTMOV
C
C     FINISH THE CYCLE (PRINTS,SUMMARIES,ETC)
C
  150 CONTINUE
C
C     CALCULATE QUANTITIES TO BE USED IN AN EDIT AND FOR THE NEXT CYCLE
C
      CALL CINIT
C
C     T2 IS THE TIME AT THE END OF THE THIS CYCLE
C
      CALL SECOND(T2)
C
C     CALCULATE THE CP TIME USED PER ZONE (GRIND TIME)
C
      XX=(T2-TOLD)*RIBJB
C
C     CONVERT POTE TO ACTUAL GRAVITATIONAL POTENTIAL ENERGY
C
      EPOT=-POTE*GZ
C
C     COMPUTE FIREBALL DIAMETER AND AVERAGE HEIGHT
C
      PDIAM=2.*PRITE
      PAVHT=.5*(PTOP+PBOTM)
C
C     PRINT OUT CYCLE SUMMARY
C
      DO 160 IPX=6,IFD,6
  160 WRITE(IPX,340) NCYC,T,DT,T2,XX,NUMIT,CIRC,DTV,IDTV,JDTV,DTC,IDTC,
     1 JDTC,TMAX,ITM,JTM,XTMAX,YTMAX,TGMX,ITG,JTG,PRITE,PTOP,PBOTM,PDIAM
     2 ,PAVHT
      DO 170 IPX=6,IFD,6
  170 WRITE(IPX,300) TI,TK,EPOT,UMOM,VMOM
      DO 180 IPX=6,IFD,6
  180 WRITE(IPX,320) DTV,DTC
      CALL DVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
      DO 190 IPX=6,IFD,6
  190 WRITE(IPX,310) VMAX,IVM,JVM
C
C     SEE IF IT IS TIME TO TURN ON THE TURBULENCE BASED ON TIME
C
      IF(NCQ.EQ.0.AND.T.GE.TQ) GO TO 200
C
C     NO. SEE IF IT IS TIME BASED ON CYCLES.
C
      IF(NCYC.NE.0.AND.NCQ.EQ.NCYC) GO TO 200
C
C     NO. SKIP TURBULENCE SEEDING.
C
      GO TO 240
  200 CONTINUE
C
C     YES. FLIP ON THE SWITCH.
C
      TURB=.TRUE.
```

```
      NCQ=NCYC
C
C     GO GET THE VORTICITY
C
      CALL GETOMG
C
C     INITIALIZE TUQ AND TUSV
C
      TUSV=0.
      CUTOFF=-.1*QMN*TUQI
      CALL START
      DO 220 J=2,JP1
      DO 210 I=1,IBAR
      TEST=-TUQI*CQ(IJ)
      TUQ(IJ)=0.
      IF(TEST.GT.CUTOFF) TUQ(IJ)=TEST
      TUSV=TUSV+TUQ(IJ)*RO(IJ)/RVOL(IJ)
      IJ=IJ+NQ
  210 IJP=IJP+NQ
      CALL LOOP
  220 CONTINUE
      CALL DONE
C
C     INITIALIZE THE PARTICLE TURBULENT DIFFUSION
C         INTERPOLATION TABLES
C
      CALL TRBERF
C
C     INDICATE THAT THE TURBULENCE HAS BEEN SEEDED
C
      DO 230 IPX=6,IFD,6
  230 WRITE(IPX,330) NCQ,TUQI,TUSV
  240 CONTINUE
C
C     SEE IF TURBULENCE IS ON
C
      IF(.NOT.TURB) GO TO 270
C
C     YES, CALCULATE SIG AND TUS.
C
      CALL START
      DO 260 J=2,JP1
      DO 250 I=1,IBAR
      TUS(IJ)=TUSI
      SIG(IJ)=QLEVEL*TUS(IJ)*SQRT(2.*TUQ(IJ))
  250 IJ=IJ+NQ
      CALL LOOP
  260 CONTINUE
      CALL DONE
  270 CONTINUE
C
C     SEE IF IT IS TIME DO AN EDIT
C
      EDIT=.FALSE.
      IF(T+EM10.LT.TOUT) GO TO 280
C
C     YES, SET THE FLAG AND UPDATE OUTPUT TIME.
```

```
C
      EDIT=.TRUE.
      TOUT=TOUT+DTO(IDTO)
      IF(T+EM10.LT.DTOC(IDTO)) GO TO 280
      TOUT=DTOC(IDTO)+DTO(IDTO+1)
      IDTO=IDTO+1
  280 CONTINUE
C
C     IF TURBULENCE HAS BEEN SEEDED,DO AN EDIT
C
      IF(NCYC.EQ.NCQ) EDIT=.TRUE.
C
C     IF THIS IS STARTUP OR THE FIRST CYCLE,DO AN EDIT
C
      IF(NCYC.LE.1) EDIT=.TRUE.
C
C     IF THE ITERATION COUNT HAS BEEN EXCEEDED,DO AN EDIT FOR
C         DIAGNOSTIC PURPOSES
C
      IF(NUMIT.GE.500) EDIT=.TRUE.
C
C     SEE IF THIS IS THE LAST CYCLE
C
      LAST=.FALSE.
C
C     QUERY TTY
C
      CALL TTYTST(IFLAG)
      IF(IFLAG.NE.0) LAST=.TRUE.
      IF(T.GE.TWFIN) LAST=.TRUE.
      IF(T2.GE.TLIM) LAST=.TRUE.
      IF(NCYC.GE.NCLST) LAST=.TRUE.
      IF(LAST.AND.IFLAG.EQ.0) EDIT=.TRUE.
C
C     SEE IF IT IS TIME TO DUMP
C
      DUMP=.FALSE.
      IF(LAST) DUMP=.TRUE.
      IF(MOD(NCYC,JDUMP).EQ.0.AND.NCYC.NE.0) DUMP=.TRUE.
      IF(EDIT.AND.NCYC.GT.1) DUMP=.TRUE.
C
C     DO THE EDIT IF REQUIRED
C
      IF(.NOT.EDIT) GO TO 290
      IF(FILM) CALL YPLOT
      CALL YEDIT
  290 CONTINUE
C
C     DO THE DUMP IF REQUIRED
C
      IF(DUMP) CALL YDUMP
C
C     QUIT IF THIS IS THE LAST CYCLE
C
      IF(LAST) RETURN
C
C     IF THIS IS THE EDIT AFTER INITIALIZATION,IMMEDIATELY BEGIN
```

```
C           CYCLE 1
C
      IF(NCYC.EQ.0) GO TO 50
      GO TO 30
C
  300 FORMAT(1H ,20X24HTOTAL INTERNAL ENERGY = ,1PE14.7/1H ,20X
     1 23HTOTAL KINETIC ENERGY = ,E14.7/1H ,20X
     2 31HTOTAL GRAV. POTENTIAL ENERGY = ,E14.7/1H ,20X
     3 24HTOTAL RADIAL MOMENTUM = ,E14.7/1H ,20X
     4 23HTOTAL AXIAL MOMENTUM = ,E14.7)
  310 FORMAT(1H ,20X7HVMAX = ,1PE12.5,11H AT VERTEX ,2I5)
  320 FORMAT(1H ,20XO20,O25)
  330 FORMAT(/1H ,27HTURBULENCE SEEDED ON CYCLE ,I4,13H WITH TUQI = ,E12
     1 .5,1X1H./1H ,5X33HTOTAL TURBULENCE ENERGY SEEDED = ,E12.5)
  340 FORMAT(1H ,6(1H*),7H CYCLE ,I5,4H, T=,1PE12.5,5H, DT=,E12.5,
     1 5H, CP=,E12.5/1H ,20X7HGRINDS=,E12.5,8H, NUMIT=,I3,7H, CIRC=,E12
     2 .5/1H ,20X4HDTV=,E12.5,7H, IDTV=,I3,7H, JDTV=,I3/1H ,20X4HDTC=,
     3 E12.5,7H, IDTC=,I3,7H, JDTC=,I3/1H ,20X5HTMAX=,E12.5,6H, ITM=,I3,
     4 6H, JTM=,I3,8H, XTMAX=,E10.3,8H, YTMAX=,E10.3/1H ,20X5HTGMX=,E12
     5 .5,6H, ITG=,I3,6H, JTG=,I3/1H ,20X5HPRIT=,E10.3,7H, PTOP=,E10.3,
     6 7H, PBOT=,E10.3,8H, PDIAM=,E10.3,8H, PAVHT=,E10.3)
      END
```

```
      SUBROUTINE YARSRT
C
C     ROUTINE TO RESTART A YAQUI PROBLEM
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK PARAM       -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM       -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC       -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
```

```
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC       -----
*     ----- BEGIN COMDECK YAQEQ       -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RD),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ       -----
*     ----- END COMDECK YSTORE      -----
      INTEGER TAPE,AA
      DIMENSION RCYCLE(3,3)
      DATA (RCYCLE(II),II=1,9)/9*0/
      DATA IEFLAG/0/
      TLIMSV=TLIM
C
C     SET UP NAMELIST INPUT TABLE
C
      ASSIGN 120 TO IERRT
      CALL TABDEF(RCYCLE,6HRCYCLE,3,IERRT)
      CALL TABSET(RCYCLE,6HINTCYC,INTCYC,IEFLAG,0,0,0,0)
      CALL TABSET(RCYCLE,4HTAPE,TAPE,IEFLAG,0,0,0,0)
C
C     READ THE RESTART TAPE NO. AND THE RESTART CYCLE. INTCYC=-1
C         SIGNIFIES RESTART FROM THE LAST DUMP ON THE TAPE.
C
      INTCYC=-1
      TAPE=0
      CALL NAMLST(RCYCLE,5,IEFLAG)
C
C     CHECK FOR INPUT ERRORS
C
      IF(IEFLAG.NE.0) CALL UNCLE(4,6HYARSRT,27,
```

```
     1 27HRCYCLE NAMELIST INPUT ERROR)
C
C     OPEN THE INPUT DUMP FILE
C
      CALL OPENIT(7,1)
C
C     NO ERRORS, CALL ROUTINE TO GET TAPE IF TAPE NO, WAS READ,
C         (ONLY FUNCTIONAL ON CROS/7600)
C
      IF(TAPE.NE.0) CALL GETTPE(TAPE)
      REWIND 7
      JNSC=LOCF(ZZ)-LOCF(AA)+1
C
C     READ THE NEXT CYCLE ON THE TAPE
C
   10 CONTINUE
      CALL SCBUFF(AA,JNSC,7,0,1,IERROR)
C
C     CHECK FOR ERRORS
C
      IF(IERROR.GE.0) GO TO 30
   20 CALL UNCLE(1,6HYARSRT,18,18HSCBUFF INPUT ERROR)
   30 CONTINUE
      IF(IERROR.EQ.0) GO TO 40
C
C     SEE IF TERMINAL DUMP RECORD WAS FOUND
C
      IF(AA(1).EQ.666) GO TO 70
      GO TO 20
   40 CONTINUE
C
C     NO, PRINT CYCLE NO, OF DUMP LAST READ,
C
      DO 50 IPX=6,IFD,6
   50 WRITE(IPX,130) NCYC
C
C     SEE IF LAST CYCLE IS DESIRED
C
      IF(INTCYC.EQ.(-1)) GO TO 60
C
C     NO, SEE IF WE HAVE FOUND THE CORRECT CYCLE,
C
      IF(NCYC-INTCYC) 60,80,110
   60 CONTINUE
C
C     CORRECT CYCLE NOT FOUND YET, SKIP REST OF DUMP AND GO READ NEXT,
C
      CALL RTAPE
      GO TO 10
   70 CONTINUE
C
C     DUMP TAPE TERMINATION FOUND, FATAL UNLESS INTCYC IS -1,
C
      IF(INTCYC.NE.(-1)) CALL UNCLE(4,6HYARSRT,17,17HEOF ON INPUT TAPE)
C
C     O.K.  DUMP WAS THE LAST ONE READ,
C
```

```
      GO TO 90
   80 CONTINUE
      CALL RTAPE
   90 CONTINUE
C
C     PRINT THE CYCLE NO.,PROBLEM NAME,AND PROBLEM TIME
C
      DO 100 IPX=6,IFD,6
      WRITE(IPX,140) NCYC
  100 WRITE(IPX,150) NAME,T
C
C     RESTART COMPLETED. SEE IF THERE ARE CHANGES TO INPUT VARIABLES.
C
      CALL YINPUT
      TLIM=TLIMSV
C
C     CLOSE THE INPUT DUMP FILE
C
      CALL CLOSIT(7)
C
C     REFRESH THE CP TIME
C
      CALL SECOND(T2)
      RETURN
C
C     DUMP NOT ON TAPE. FATAL ERROR.
C
  110 CONTINUE
      CALL UNCLE(4,6HYARSRT,23,23HRESTART CYCLE NOT FOUND)
  120 CONTINUE
      CALL UNCLE(4,6HYARSRT,36,36HCYCLE NAMELIST INITIALIZATION ERROR)
C
  130 FORMAT(1H ,6HCYCLE ,I5,6H FOUND)
  140 FORMAT(1H ,21HRESTARTING FROM CYCLE,I5)
  150 FORMAT(1H ,8A10/1H ,2HT=,1PE12.5/1H )
      END
```

```
      SUBROUTINE YASET
C
C     ROUTINE TO GENERATE A NEW PROBLEM
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
```

```
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ----- END COMDECK YAQDIM      -----
*      ----- BEGIN COMDECK YAQSC       -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ----- END COMDECK YAQSC       -----
*      ----- BEGIN COMDECK YAQEQ       -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*      ----- END COMDECK YAQEQ       -----
*      ----- END COMDECK YSTORE      -----
*      ----- BEGIN COMDECK CTAB       -----
C     COMMON/CTABC/XTAB(ITABXP),YTAB(ITABYP)
      COMMON/CTABC/XTAB(101),YTAB(151)
*      ----- END COMDECK CTAB        -----
      DATA NE/0/
C
C     READ THE PROBLEM TITLE
C
```

```
      READ(5,320) NAME
C
C     INITIALIZE THE PROBLEM VARIABLES
C
      CALL DEFINE
C
C     READ THE INPUT VARIABLES
C
      CALL YINPUT
C
C     OUTPUT THE PROBLEM TITLE
C
      DO 10 IPX=6,IFD,6
   10 WRITE(IPX,320) NAME
C
C     IF NCQ.LT.0,THERE WILL BE NO TURBULENCE
C
      IF(NCQ.LT.0) GO TO 30
C
C     THERE WILL BE TURBULENCE. PRINT OUT THE INPUT QUANTITIES
C
      DO 20 IPX=6,IFD,6
   20 WRITE(IPX,290) QLEVEL,TUQI,TUSI,NCQ,TQ,TSTRTD,WMAXEF,RMINEF
      GO TO 50
C
C     THERE WILL BE NO TURBULENCE. INDICATE SUCH.
C
   30 CONTINUE
      DO 40 IPX=6,IFD,6
   40 WRITE(IPX,300)
   50 CONTINUE
C
C     PRINT GENERAL INPUT VARIABLES
C
      DO 60 KT=6,IFD,6
      WRITE(KT,330) IBAR,JBAR,IUNF,JUNF,JCEN,DR,DZ,CYL,GRDVEL,A0,A0M,B0,
     1 KXI
      WRITE(KT,340) MU,LAM,OM,EPS,GR,GZ
      WRITE(KT,350) FREZXR,FREZYT,FREZYB,ZORIG,YB,REZY0,REZRON,REZSIE
      WRITE(KT,360) GZP
      WRITE(KT,370) T,DT,NCLST,TWFIN,PAPER,FILM
      WRITE(KT,260) ANC,A0FAC
      WRITE(KT,310) IST
      WRITE(KT,380)(DTO(N),N=1,10)
      WRITE(KT,390)(DTOC(N),N=1,10)
   60 CONTINUE
C
C     CALCULATE AND STORE PROBLEM CONSTANTS
C
      IM1=IBAR-1
      IP1=IBAR+1
      IP2=IBAR+2
      JP1=JBAR+1
      JP2=JBAR+2
      JP4=JBAR+4
      RIBAR=1./FLOAT(IBAR)
      RIBJB=1./FLOAT(IBAR*JBAR)
```

```
      NQIB=NQ*IBAR
      OMCYL=1.-CYL
      EM10=1.E-10
      IF(FREZXR.NE.1.) ROMFXR=1./(1.-FREZXR)
      IF(FREZYT.NE.1.) ROMFYT=1./(1.-FREZYT)
      IF(FREZYB.NE.1.) ROMFYB=1./(1.-FREZYB)
C
C     NQI IS THE NO. OF WORDS OF DATA NEEDED TO STORE ONE ROW. IT IS
C         THE SIZE OF A SMALL CORE BUFFER.
C
      NQI=NQ*IP1
C
C     SEE ROUTINE -START- FOR DEFINITIONS
C
      ISC2=NQI+1
      ISC3=ISC2+NQI
C
C     SEE ROUTINE -STARTD- FOR DEFINITIONS
C
      ITV=JP1*NQI
C
C     NSC IS THE NO. OF WORDS OF SMALL CORE IN COMMON YSC2
C
      NSC=LOCF(ZZ)-LOCF(AA)+1
C
C     NLC IS THE NO. OF WORDS OF LARGE CORE USED IN COMMON YLC1. IT
C         SHOULD NOT EXCEED THE PARAMETER VALUE.
C
      NLC=JP4*NQI
C
C     GO CHECK THE LARGE AND SMALL CORE PARAMETERS
C
      CALL PCHCK
C
C     IDTO IS THE SUBSCRIPT IN THE DTOC ARRAY SUCH THAT
C                   DTOC(IDTO-1).LT.T.LE.DTOC(IDTO)
C
      IDTO=1
C
C     TOUT IS THE TIME AT WHICH OUTPUT SHOULD OCCUR NEXT
C
      TOUT=DTO(1)
      IF(TOUT.GT.T) GO TO 90
   70 CONTINUE
      TOUT=TOUT+DTO(IDTO)
      IF(TOUT.LE.DTOC(IDTO)) GO TO 80
      TOUT=DTOC(IDTO)
      IDTO=IDTO+1
   80 IF(TOUT.GT.T) GO TO 90
      GO TO 70
   90 CONTINUE
C
C     THE TIMESTEP FOR THE FIRST CYCLE IS REDUCED BY A FACTOR OF 10
C         FROM THE SPECIFIED INITIAL DT
C
      DT=.1*DT
C
```

```
C      DTPOS IS THE TIMESTEP THAT IS POSSIBLE BASED ON STABILITY
C          CRITERIA. THE ACTUAL TIMESTEP DT MAY BE LESS IF IT HAS BEEN
C          ADJUSTED ON AN OUTPUT CYCLE SO THAT T=TOUT EXACTLY.
C
       DTPOS=DT
C
C      DTSV IS THE TIMESTEP FROM THE LAST CYCLE
C
       DTSV=DT
C
C      NCYC IS THE CYCLE NO.
C
       NCYC=0
C
C      COLAMU IS 1/(2/3*(2*MU+LAM))
C
       COLAMU=1.5/(LAM+MU+MU+EM10)
C
C      NILNG IS THE NO. OF TIMES A PARTICLE HAS BEEN STORED FOR USE IN
C          MAKING TIME-DEPENDENT PARTICLE PLOTS
C
       NILNG=0
C
C      IUNF MUST BE AT LEAST 1
C
       IUNF=MAX0(IUNF,1)
C
C      JUNF MUST BE AT LEAST 2
C
       JUNF=MAX0(JUNF,2)
       JUNFO2=JUNF/2
C
C      IF JCEN IS ZERO, SET IT TO JBAR/2
C
       IF(JCEN.EQ.0) JCEN=JBAR/2
C
C      GO GENERATE THE PARTICLES
C
       CALL PRTGEN
C
C      GO GENERATE THE MESH
C
       CALL MSHMKR
C
C      GO SET THE PLOT QUANTITIES
C
       CALL FILMCO
C
C      ********************************************************************
C      LOCATE WHICH CELL EACH PARTICLE IS IN IF THERE ARE ANY PARTICLES
C      ********************************************************************
C
       IF(NPT.LE.0) GO TO 210
C
C      ASSUMING THAT THE MESH IS STILL RECTANGULAR,STORE THE X AND Y
C          VALUES IN XTAB AND YTAB. MAKE SURE THE ARRAYS ARE LARGE
C          ENOUGH.
```

```
C
      IF(JP2.GT.ITABYP) CALL UNCLE(4,5HYASET,21,21HYTAB ARRAY OVERFLOWED
     1 )
      IF(IP1.GT.ITABXP) CALL UNCLE(4,5HYASET,21,21HXTAB ARRAY OVERFLOWED
     1 )
      CALL START
      DO 120 J=2,JP2
      YTAB(J)=Y(IJ)
      IF(J.GT.2) GO TO 110
      DO 100 I=1,IP1
      XTAB(I)=X(IJ)
  100 IJ=IJ+NQ
  110 CALL LOOP
  120 CONTINUE
C
C     INITIALIZE THE LCM ADDRESS AND THE PARTICLE COUNT
C
      IECP=1
      NPPT=0
C
C     BRING IN A BUFFER-FULL OF PARTICLE DATA
C
C     LPB IS SET IN PARTGEN
C
  130 CALL ECRD(AASC,NLCP1+IECP-1,LPB,IDUM)
      KP=1
  140 CONTINUE
C
C     LOCATE THE I AND J OF THE CELL CONTAINING THE PARTICLE IN QUESTION
C
      DO 150 J=2,JP2
      IF(YTAB(J).GT.YPAR(KP)) GO TO 170
  150 CONTINUE
      DO 160 IPX=6,IFD,6
  160 WRITE(IPX,270) YTAB(JP2),YPAR(KP),KP
      CALL UNCLE(1,5HYASET,23,23HJ OF PARTICLE NOT FOUND)
  170 DO 180 I=1,IP1
      IF(XTAB(I).GT.XPAR(KP)) GO TO 200
  180 CONTINUE
      DO 190 IPX=6,IFD,6
  190 WRITE(IPX,280) XTAB(IP1),XPAR(KP),KP
      CALL UNCLE(1,5HYASET,23,23HI OF PARTICLE NOT FOUND)
  200 CONTINUE
C
C     INCREMENT THE PARTICLE COUNT
C
      NPPT=NPPT+1
C
C     I AND J OF CELL FOUND. CODE IT AND STORE IN ITAB. FIRST MAKE SURE
C         THAT STORAGE WILL NOT BE OVERFLOWED.
C
      IF(NPPT.GT.ITABP) CALL UNCLE(4,5HYASET,21,21HITAB ARRAY OVERFLOWED
     1 )
      ITAB(NPPT)=(J-2)*IP1+I-1
C
C     SEE IF ALL PARTICLES HAVE BEEN PROCESSED
C
```

```
      IF(NPPT.GE.NPT) GO TO 210
C
C     NO. SEE IF THE BUFFER NEEDS TO BE RELOADED.
C
      KP=KP+3
      IF(KP.LT.LPB) GO TO 140
C
C     YES. INCREMENT THE LCM POINT AND REFILL THE BUFFER
C
      IECP=IECP+LPB
      GO TO 130
  210 CONTINUE
C
C     ******************************************************************
C     CALCULATE THE CELL-CENTERED VOLUMES (RECIPROCALS),MASSES,AND
C         TOTAL ENERGIES FOR ALL CELLS
C     ******************************************************************
C
      CALL START
      DO 230 J=2,JP1
      DO 220 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      RVOL(IJ)=8./((R1+R2+R3+R4)*((X1-X3)*(Y2-Y4)-(Y1-Y3)*(X2-X4)))
      M(IJ)=RO(IJ)/RVOL(IJ)
      GRROR(IJ)=0.
      GRROR(IPJ)=0.
      GRROR(IPJP)=0.
      GRROR(IJP)=0.
      GRROZ(IJ)=0.
      GRROZ(IPJP)=0.
      GRROZ(IJP)=0.
      GRROZ(IPJ)=0.
      E(IJ)=SIE(IJ)+.125*(U(IPJ)**2+U(IPJP)**2+U(IJP)**2+U(IJ)**2+V(IPJ)
     1 **2+V(IPJP)**2+V(IJP)**2+V(IJ)**2)
      IJ=IPJ
  220 IJP=IPJP
      CALL LOOP
  230 CONTINUE
      CALL DONE
C
C     ******************************************************************
C     COMPUTE THE VERTEX MASSES.EXCEPT FOR BOUNDARY VERTICES,THE VERTEX
C         MASS IS JUST 1/4 OF THE MASSES OF ALL THE CELLS HAVING THE
C         VERTEX AS A CORNER.
```

```
C       ***********************************************************************
C
        CALL STARTD
        DO 250 JJ=2,JP2
        J=JP4-JJ
        DO 240 II=1,IP1
        I=IP2-II
        IMJ=IJ-NQ
        IMJM=IJM-NQ
        XX=0.0
        IF(I.NE.IP1.AND.J.NE.2) XX=M(IJM)
        IF(I.NE.IP1.AND.J.NE.JP2) XX=XX+M(IJ)
        IF(I.NE.1.AND.J.NE,JP2) XX=XX+M(IMJ)
        IF(I.NE.1.AND.J.NE.2) XX=XX+M(IMJM)
        RM(IJ)=4./XX
        IJ=IMJ
    240 IJM=IMJM
        CALL LOOPD
    250 CONTINUE
        RETURN
C
    260 FORMAT(1H ,3X4HANC=,1PE12.5/1H ,1X6HA0FAC=,E12.5)
    270 FORMAT(1H,12HYTAB,YPAR,KP,2E20.7,I10)
    280 FORMAT(1H,12HXTAB,XPAR,KP,2E20.7,I10)
    290 FORMAT(1H ,24HTHERE WILL BE TURBULENCE/1X7HQLEVEL=,1PE12.5/3X
      1 5HTUQI=,E12.5/3X5HTUSI=,E12.5/4X4HNCQ=,I5/5X3HTQ=,E12.5/1H ,7HTST
      1 E12.5/1H ,7HWMAXEF=,E12.5/1H ,7HRMINEF=,E12.5)
    300 FORMAT(15H  NO TURBULENCE)
    310 FORMAT(1H ,3X,4HIST=,I4)
    320 FORMAT(8A10)
    330 FORMAT(3X,5HIBAR=I4/3X,5HJBAR=I4/3X,5HIUNF=I4/3X,5HJUNF=I4/3X,
      1 5HJCEN=I4/5X,3HDR=1PE12.5/5X,3HDZ=E12.5/4X,4HCYL=E12.5/8H GRDVEL=
      2 E12.5/5X,3HA0=E12.5/4X,4HA0M=E12.5/5X,3HB0=E12.5/4X,4HKXI=I3)
    340 FORMAT(5X,3HMU=1PE12.5/4X,4HLAM=E12.5/5X,3HOM=E12.5/4X,4HEPS=E12.5
      1 /5X,3HGR=E12.5/5X,3HGZ=E12.5)
    350 FORMAT(1H ,7HFREZXR=,1PE12.5/1H ,7HFREZYT=,E12.5/1H ,7HFREZYB=,E12
      1 .5/1H ,1X6HZORIG=,E12.5/1H ,4X3HYB=,E12.5/1H ,1X6HREZY0=,E12.5/
      2 1H ,7HREZRON=,E12.5/1H ,7HREZSIE=,E12.5)
    360 FORMAT(4X,4HGZP=E12.5)
    370 FORMAT(1H ,5X2HT=,1PE12.5/1H ,4X3HDT=,E12.5/1H ,1X6HNCLST=,I6/1H ,
      1 1X6HTWFIN=,E12.5/1H ,1X6HPAPER=,I1/1H ,2X5HFILM=,I1)
    380 FORMAT(12H  DTO(1-10)=4(1PE14.5)/(12X,4E14.5))
    390 FORMAT(12H DTOC(1-10)=4(1PE14.5)/(12X,4E14.5))
        END
```

```
        SUBROUTINE YDUMP
C
C       ROUTINE TO DO A YAQUI DUMP
C
C       WRITTEN BY J.L.NORTON,LASL T-3,1975
C
```

```
*      ***** BEGIN COMDECK PARAM      *****
       COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*      ***** END COMDECK PARAM      *****
*      ***** BEGIN COMDECK YSTORE      *****
*      ***** BEGIN COMDECK YAQDIM      *****
       DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ***** END COMDECK YAQDIM      *****
*      ***** BEGIN COMDECK YAQSC      *****
       LOGICAL RESTRT,FILM,PAPER,TURB
       REAL LAM,MU
C      COMMON/YSC1/AASC(NSCP1)
       COMMON/YSC1/AASC(9600)
       COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
       COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
       COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIB,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
       COMMON/YSC2/ZZ
C      COMMON/YSC4/ITAB(ITABP)
       COMMON/YSC4/ITAB(1000)
       COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ***** END COMDECK YAQSC      *****
*      ***** BEGIN COMDECK YAQEQ      *****
       EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
       REAL M,MP,MPAR,MTIL
*      ***** END COMDECK YAQEQ      *****
*      ***** END COMDECK YSTORE      *****
       INTEGER AA
```

```
      DATA TP/0./
C
C     PRINT TIMING FIGURE
C
      CALL SECOND(TP)
      WRITE(59,70) TP,T,NCYC
C
C     PRINT THE TIME AND CYCLE BEING DUMPED
C
      DO 10 IPX=6,IFD,6
   10 WRITE(IPX,90) T,NCYC
      CALL OPENIT(8,1)
C
C     BACKSPACE OVER THE TRAILER RECORD WRITTEN BY THE LAST DUMP
C
      BACKSPACE 8
C
C     WRITE OUT THE SCM COMMON
C
      JNSC=LOCF(ZZ)-LOCF(AA)+1
      CALL SCBUFF(AA,JNSC,8,1,1,IERROR)
      IF(IERROR.EQ.0) GO TO 30
   20 CALL UNCLE(4,5HYDUMP,19,19HSCBUFF OUTPUT ERROR)
C
C     I/O SUCCESSFULLY COMPLETED
C
   30 CONTINUE
C
C     WRITE OUT THE ARRAYS FROM LCM
C
      CALL LCBUFF(0,NLC,8,1,1,IERROR)
C
C     CHECK FOR ERRORS IN LCBUFF
C
      IF(IERROR.EQ.0) GO TO 50
C
C     LCBUFF ERROR. KILL THE RUN.
C
   40 CALL UNCLE(4,5HYDUMP,19,19HLCBUFF OUTPUT ERROR)
C
C     NO ERRORS. CONTINUE
C
   50 CONTINUE
C
C     SEE IF THERE ARE ANY PARTICLES
C
      IF(NPT.LE.0) GO TO 60
C
C     YES. WRITE OUT THE PARTICLE ARRAYS FROM LCM.
C
      CALL LCBUFF(NLCP1,NPS,8,1,1,IERROR)
      IF(IERROR.NE.0) GO TO 40
C
C     WRITE OUT THE ITAB ARRAY FROM SCM
C
      CALL SCBUFF(ITAB,ITABP,8,1,1,IERROR)
      IF(IERROR.NE.0) GO TO 20
```

```
C
C     SEE IF THERE IS ANY TIME-DEPENDENT PARTICLE DATA
C
      IF(NILNG.LE.0) GO TO 60
C
C     YES, WRITE IT OUT FROM LCM.
C
      CALL LCBUFF(NLCP1+NLCP2,2*NP1*NILNG,8,1,1,IERROR)
      IF(IERROR.NE.0) GO TO 40
   60 CONTINUE
C
C     TERMINATE THE DUMP WITH A SPECIAL TRAILER RECORD
C
      IJUNK=666
      CALL SCBUFF(IJUNK,1,8,1,1,IERROR)
      IF(IERROR.NE.0) GO TO 20
      CALL SECOND(TP)
C
C     PRINT TIMING FIGURE
C
      WRITE(59,80) TP
      RETURN
C
   70 FORMAT(1H ,20HBEGIN YDUMP AT CP = ,F10.4,1H,,5X15HPROBLEM TIME = ,
     1 1PE12.5,1H,,5X8HCYCLE = ,I5)
   80 FORMAT(1H ,18HEND YDUMP AT CP = ,F10.4)
   90 FORMAT(1H ,15HTAPE DUMP AT T=,1PE12.5,8H, CYCLE=,I5)
      END
```

```
      SUBROUTINE YEDIT
C
C     ROUTINE TO PRINT FULL LISTING OF YAQUI MESH QUANTITIES
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
```

```
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPO8,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ      -----
*     ----- END COMDECK YSTORE     -----
      EQUIVALENCE(TEST,ITEST)
      DATA TP/0./
      DATA INDEF/17770000000000777777B/
      CALL SECOND(TP)
      WRITE(59,120) TP
C
C     ALLOW 60 LINES PER PAGE OF DATA. SET THE COUNT TO FORCE A
C         TOP-OF-PAGE HEADING FIRST THING
C
      LINESF=99
C     ********************************************************************
C     TEMPORARY PATCH FOR OUTPUTTING VORTICITY
C     ********************************************************************
      IF(.NOT.TURB) CALL GETOMG
      CALL START
      DO 100 J=1,JP2
```

```
      DO 90 I=1,IP1
C
C     SEE IF A HEADER NEEDS TO BE PRINTED
C
      IF(LINESF.LT.60) GO TO 30
C
C     YES, RESET THE LINE COUNT.
C
      LINESF=0
C
C     PRINT THE HEADER
C
      DO 20 IPX=IPD,IFD,6
      WRITE(IPX,140) JNM,NAME,T,NCYC
      IF(TURB) GO TO 10
      WRITE(IPX,150)
      GO TO 20
   10 CONTINUE
      WRITE(IPX,160)
   20 CONTINUE
   30 CONTINUE
      IPJM=IJM+NQ
      IPJ=IJ+NQ
      D=0.
      PRM=0.
      PRV=0.
      IF(TURB) GO TO 60
      TEST=RM(IJM)
      IF(ITEST.EQ.INDEF) GO TO 40
      IF(TEST.NE.0.) PRM=1./TEST
   40 CONTINUE
      TEST=RVOL(IJM)
      IF(ITEST.EQ.INDEF) GO TO 50
      IF(TEST.NE.0.) PRV=1./TEST
   50 CONTINUE
      D=CONC(IJM)
C     ***********************************************************************
C     TEMPORARY PATCH FOR OUTPUTTING VORTICITY
C     ***********************************************************************
      PRM=CQ(IJM)
      GO TO 70
   60 PRV=TUS(IJM)
      D=CONC(IJM)
      PRM=TUQ(IJM)
   70 CONTINUE
      DO 80 IPX=IPD,IFD,6
   80 WRITE(IPX,170) I,J,X(IJM),Y(IJM),U(IJM),V(IJM),SIE(IJM),RO(IJM),
     1 PRV,D,PRM,P(IJM)
      LINESF=LINESF+1
      IJ=IPJ
   90 IJM=IPJM
      CALL LOOP
  100 CONTINUE
      IF(.NOT.FILM) GO TO 110
      FIYB=FIYBO
      IYB=IYBO
      YCONV=YCNVLD
```

```
      XCONV=XCNVLD
      FIXR=FIXRO
      FIXL=FIXLO
      IXR=IXRO
      IXL=IXLO
      CALL ADV(1)
  110 CONTINUE
      CALL SECOND(TP)
      WRITE(59,130) TP
      RETURN
C
  120 FORMAT(1H ,20HBEGIN YEDIT AT CP = ,F10.4)
  130 FORMAT(1H ,18HEND YEDIT AT CP = ,F10.4)
  140 FORMAT(1H1,A10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
  150 FORMAT(1H ,2X1HI,3X1HJ,6X1HX,10X1HY,10X1HU,10X1HV,9X3HSIE,8X3HRHO,
     1 8X3HVOL,9X1HC,9X3HVTC,8X1HP)
  160 FORMAT(1H ,2X1HI,3X1HJ,6X1HX,10X1HY,10X1HU,10X1HV,9X3HSIE,8X3HRHO,
     1 8X4HSCAL,9X1HC,9X1HQ,10X1HP)
  170 FORMAT(1X,I3,1H,I3,10(1X,1PE10.3))
      END
```

```
      SUBROUTINE YEXIT(IABORT)
C
C     YAQUI ERROR RECOVERY ROUTINE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      CALL EXIT
      END
```

```
      SUBROUTINE YFLUX
C
C     ROUTINE TO FLUX MASS,MOMENTUM,AND ENERGY IF THE GRID VELOCITY IS
C         NOT EQUAL TO THE FLUID VELOCITY
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ===== BEGIN COMDECK YSTORE     =====
*     ===== BEGIN COMDECK YAQDIM     =====
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
```

```
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ***** END COMDECK YAQDIM      *****
*      ***** BEGIN COMDECK YAQSC      *****
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GROVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ***** END COMDECK YAQSC      *****
*      ***** BEGIN COMDECK YAQEQ      *****
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*      ***** END COMDECK YAQEQ      *****
*      ***** END COMDECK YSTORE      *****
*      ***** BEGIN COMDECK PARAM      *****
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*      ***** END COMDECK PARAM      *****
*      ***** BEGIN COMDECK ASTORE      *****
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*      ***** END COMDECK ASTORE      *****
      DTC=1.E30
      CALL START
```

```
      DO 50 J=2,JP1
      DO 40 I=1,IBAR
      IMJ=IJ-NQ
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      UL1=UL(IPJ)
      VL1=VL(IPJ)
      UL2=UL(IPJP)
      VL2=VL(IPJP)
      UL3=UL(IJP)
      VL3=VL(IJP)
      UL4=UL(IJ)
      VL4=VL(IJ)
      UD1=UG(IPJ)-0.5*(UL1+U(IPJ))
      VD1=VG(IPJ)-0.5*(VL1+V(IPJ))
      UD2=UG(IPJP)-0.5*(UL2+U(IPJP))
      VD2=VG(IPJP)-0.5*(VL2+V(IPJP))
      UD3=UG(IJP)-0.5*(UL3+U(IJP))
      VD3=VG(IJP)-0.5*(VL3+V(IJP))
      UD4=UG(IJ)-0.5*(UL4+U(IJ))
      VD4=VG(IJ)-0.5*(VL4+V(IJ))
      X12=X1-X2
      X23=X2-X3
      X34=X3-X4
      X41=X4-X1
      Y21=Y2-Y1
      Y32=Y3-Y2
      Y43=Y4-Y3
      Y14=Y1-Y4
      Y31=Y3-Y1
      R12=R1+R2
      R23=R2+R3
      R34=R3+R4
      R41=R4+R1
      U12=UL1+UL2
      U23=UL2+UL3
      U34=UL3+UL4
      U41=UL4+UL1
      V12=VL1+VL2
      V23=VL2+VL3
      V34=VL3+VL4
      V41=VL4+VL1
      D=.25*RVOL(IJ)*(R12*(U12*Y21+V12*X12)+R23*(U23*Y32+V23*X23)+R34*(
     1 U34*Y43+V34*X34)+R41*(U41*Y14+V41*X41))
      VOLR=VOLT=VOLC=1./RVOL(IJ)
```

```
      IF(I.NE.IBAR) VOLR=1./RVOL(IPJ)
      IF(J.NE.JP1) VOLT=1./RVOL(IJP)
      IF(I.EQ.1) GO TO 60
      FL=-FR
      AL=-AR
   10 IF(J.EQ.2) GO TO 70
      FB=-FT(I)
      AB=-AT(I)
   20 FR=DTO8*R12*((UD1+UD2)*Y21+(VD1+VD2)*X12)
      AR=A0M*SIGN(1.,FR)+B0*4.*FR/(VOLR+VOLC)
      FT(I)=DTO8*R23*((UD2+UD3)*Y32+(VD2+VD3)*X23)
      AT(I)=A0M*SIGN(1.,FT(I))+B0*4.*FT(I)/(VOLT+VOLC)
      XX=AMAX1(ABS(FB),ABS(FR),ABS(FT(I)),ABS(FL))
      DTC=AMIN1(DTC,DTPOS*A0FAC/(XX*RVOL(IJ)+DTPOS*ABS(D)+EM10))
      IF(DTCSAV.NE.DTC) IDTC=I
      IF(DTCSAV.NE.DTC) JDTC=J
      DTCSAV=DTC
      MP(IJ)=MTIL(IJ)*VOLC+FR*((1.-AR)*ROL(IJ)+(1.+AR)*ROL(IPJ))+FT(I)*(
     1 (1.-AT(I))*ROL(IJ)+(1.+AT(I))*ROL(IJP))+FL*((1.-AL)*ROL(IJ)+(1.+
     2 AL)*ROL(IMJ))+FB*((1.-AB)*ROL(IJ)+(1.+AB)*ROL(IJM))
      ROE=RO(IJ)*ETIL(IJ)
      SIE(IJ)=1./MP(IJ)*(ROE*VOLC+FR*((1.-AR)*ROE+(1.+AR)*RO(IPJ)*ETIL(
     1 IPJ))+FT(I)*((1.-AT(I))*ROE+(1.+AT(I))*RO(IJP)*ETIL(IJP))+FL*((1.
     2 -AL)*ROE+(1.+AL)*RO(IMJ)*ETIL(IMJ))+FB*((1.-AB)*ROE+(1.+AB)*RO(
     3 IJM)*ETIL(IJM)))
      IF(.NOT.TURB) GO TO 30
      ROQ=RO(IJ)*TUQ(IJ)
      TUQVEC(IJ)=1./MP(IJ)*(ROQ*VOLC+FR*((1.-AR)*ROQ+(1.+AR)*RO(IPJ)*TUQ
     1 (IPJ))+FT(I)*((1.-AT(I))*ROQ+(1.+AT(I))*RO(IJP)*TUQ(IJP))+FL*((1.
     2 -AL)*ROQ+(1.+AL)*RO(IMJ)*TUQ(IMJ))+FB*((1.-AB)*ROQ+(1.+AB)*RO(IJM
     3 )*TUQ(IJM)))
   30 CONTINUE
      ROQ=CONC(IJ)
      CTEMP(IJ)=VOLC*ROQ+FR*((1.-AR)*ROQ+(1.+AR)*CONC(IPJ))+FT(I)*((1.-
     1 AT(I))*ROQ+(1.+AT(I))*CONC(IJP))+FL*((1.-AL)*ROQ+(1.+AL)*CONC(IMJ
     2 ))+FB*((1.-AB)*ROQ+(1.+AB)*CONC(IJM))
      CTEMP(IJ)=CTEMP(IJ)*RVOL(IJ)
      RVOL(IJ)=8./((R1+R2+R3+R4)*((X1-X3)*(Y2-Y4)-(Y1-Y3)*(X2-X4)))
      IJ=IPJ
      IJP=IPJP
   40 IJM=IJM+NQ
      CALL LOOP
   50 CONTINUE
      CALL DONE
      GO TO 80
   60 FL=DTO8*R34*((UD3+UD4)*Y43+(VD3+VD4)*X34)
      AL=A0M*SIGN(1.,FL)+B0*2.*FL*RVOL(IJ)
      GO TO 10
   70 FB=DTO8*R41*((UD4+UD1)*Y14+(VD4+VD1)*X41)
      AB=A0M*SIGN(1.,FB)+B0*2.*FB*RVOL(IJ)
      GO TO 20
   80 CALL START
      DO 110 J=2,JP1
      DO 100 I=1,IBAR
      RO(IJ)=MP(IJ)*RVOL(IJ)
      CONC(IJ)=CTEMP(IJ)
      IF(J.EQ.2) RO(IJM)=ROL(IJM)
```

```
      IF(.NOT.TURB) GO TO 90
      TUQ(IJ)=TUQVEC(IJ)
      IF(TUQ(IJ).LT.0.) TNEG=TNEG+TUQ(IJ)*RO(IJ)/RVOL(IJ)
      IF(TUQ(IJ).LT.0.) TUQ(IJ)=0.
   90 CONTINUE
      IF(J.EQ.JP1) RO(IJP)=ROL(IJP)
      IF(I.EQ.IBAR) RO(IJ+NQ)=ROL(IJ+NQ)
      IJM=IJM+NQ
      IJP=IJP+NQ
  100 IJ=IJ+NQ
      CALL LOOP
  110 CONTINUE
      CALL DONE
      CALL STARTD
      DO 130 JJ=2,JP2
      J=JP4-JJ
      DO 120 II=1,IP1
      I=IP2-II
      IMJ=IJ-NQ
      IMJM=IJM-NQ
      XX=0.
      IF(I.NE.IP1.AND.J.NE.2) XX=MP(IJM)
      IF(I.NE.IP1.AND.J.NE.JP2) XX=XX+MP(IJ)
      IF(I.NE.1.AND.J.NE.JP2) XX=XX+MP(IMJ)
      IF(I.NE.1.AND.J.NE.2) XX=XX+MP(IMJM)
      RMP(IJ)=4./XX
      IJ=IMJ
  120 IJM=IMJM
      CALL LOOPD
  130 CONTINUE
      CALL START
      DO 150 J=2,JP2
      DO 140 I=1,IP1
      XX=RMP(IJ)/RM(IJ)
      UP(IJ)=XX*UL(IJ)
      VP(IJ)=XX*VL(IJ)
  140 IJ=IJ+NQ
      CALL LOOP
  150 CONTINUE
      CALL DONE
      CALL START
      DO 260 J=2,JP1
      DO 250 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      UL1=UL(IPJ)
      UG1=UG(IPJ)
      VL1=VL(IPJ)
      VG1=VG(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      UL2=UL(IPJP)
      UG2=UG(IPJP)
```

```
      VL2=VL(IPJP)
      VG2=VG(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      UL3=UL(IJP)
      UG3=UG(IJP)
      VL3=VL(IJP)
      VG3=VG(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      UL4=UL(IJ)
      UG4=UG(IJ)
      VL4=VL(IJ)
      VG4=VG(IJ)
      XX=DTO8*ROL(IJ)
      UL13=0.5*(UL1+UL3+U(IPJ)+U(IJP))
      VL13=0.5*(VL1+VL3+V(IPJ)+V(IJP))
      UL24=0.5*(UL2+UL4+U(IPJP)+U(IJ))
      VL24=0.5*(VL2+VL4+V(IPJP)+V(IJ))
      F13=XX*(R1+R3)*((UG1+UG3-UL13)*(Y3-Y1)+(VG1+VG3-VL13)*(X1-X3))
      F24=XX*(R2+R4)*((UG2+UG4-UL24)*(Y2-Y4)+(VG2+VG4-VL24)*(X4-X2))
      FM1=F24*RMP(IPJ)
      FM2=F13*RMP(IPJP)
      FM3=F24*RMP(IJP)
      FM4=F13*RMP(IJ)
      XC=.25*(X1+X2+X3+X4)
      YC=.25*(Y1+Y2+Y3+Y4)
      UC=.25*(UL13+UL24)
      VC=.25*(VL13+VL24)
      UGC=.25*(UG1+UG2+UG3+UG4)
      VGC=.25*(VG1+VG2+VG3+VG4)
      UGCUC=UGC-UC
      VGCVC=VGC-VC
      A=UGCUC*(Y3-Y1)+VGCVC*(X1-X3)
      B=UGCUC*(Y4-Y2)+VGCVC*(X2-X4)
      IF(A) 160,180,170
  160 IF(B) 230,220,220
  170 IF(B) 210,200,200
  180 IF(B) 210,190,200
  190 W=H=0.5
      GO TO 240
  200 H=1.
      W=((Y3-YC)*UGCUC+(XC-X3)*VGCVC)/((Y3-Y2)*UGCUC+(X2-X3)*VGCVC)
      GO TO 240
  210 W=1.
      H=((YC-Y1)*UGCUC+(X1-XC)*VGCVC)/((Y2-Y1)*UGCUC+(X1-X2)*VGCVC)
      GO TO 240
  220 W=0.
      H=((YC-Y4)*UGCUC+(X4-XC)*VGCVC)/((Y3-Y4)*UGCUC+(X4-X3)*VGCVC)
      GO TO 240
  230 H=0.
      W=((YC-Y4)*UGCUC+(X4-XC)*VGCVC)/((Y1-Y4)*UGCUC+(X4-X1)*VGCVC)
  240 OMW=1.-W
      OMH=1.-H
      UB=W*H*UL2+OMW*H*UL3+W*OMH*UL1+OMW*OMH*UL4
```

```
      VB=W*H*VL2+OMW*H*VL3+W*OMH*VL1+OMW*OMH*VL4
      ALTE=A0+B0*(ABS(F13)+ABS(F24))*RVOL(IJ)/ROL(IJ)
      XX=(1.-ALTE)*UC+ALTE*UB
      UP(IPJ)=UP(IPJ)-FM1*XX
      UP(IJP)=UP(IJP)+FM3*XX
      UP(IPJP)=UP(IPJP)-FM2*XX
      UP(IJ)=UP(IJ)+FM4*XX
      XX=(1.-ALTE)*VC+ALTE*VB
      VP(IPJ)=VP(IPJ)-FM1*XX
      VP(IJP)=VP(IJP)+FM3*XX
      VP(IPJP)=VP(IPJP)-FM2*XX
      VP(IJ)=VP(IJ)+FM4*XX
      IJ=IPJ
  250 IJP=IPJP
      CALL LOOP
  260 CONTINUE
      CALL DONE
C
C     GO SET THE BOUNDARY CONDITIONS
C
      CALL BC(3)
      RETURN
      END
```

```
      SUBROUTINE YINIT
C
C     ROUTINE TO INITIALIZE THE CODE
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1974
C
*     ----- BEGIN COMDECK PARAM      -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
```

```
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB\IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYØ,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1ØØØ)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ     -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PMØ,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(1Ø),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(2Ø),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(3Ø),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ      -----
*     ----- END COMDECK YSTORE     -----
      COMMON/FTABC/FTAB(2)
      DIMENSION START(3,5)
      DATA (START(II),II=1,15)/15*Ø/
      DATA II,IDUM/Ø,Ø/
      DATA IEFLAG/Ø/
      DATA RESTRT,PAPER,FILM,WRAPUP/.TRUE.,.FALSE.,.TRUE.,2Ø./
C
C     GET THE CP TIME AT JOB START
C
      CALL SECOND(T2)
C
C     MAKE SURE ENOUGH LCM IS AVAILABLE
C
      INEED=NLCP1+NLCP2+NLCP3+IFLMSZ+NLCP4+1
      CALL GETLCM(ILSIZE)
      IF(INEED.GT.ILSIZE) CALL UNCLE(4,5HYINIT,27,
     1 27HNOT ENOUGH LCM IS AVAILABLE)
C
C     NQ IS THE NO. OF DISTINCT LCM ARRAYS ALLOCATED. ARRAYS CAN BE
C         ADDED OR DELETED AS LONG AS NQ IS CHANGED CONSISTENTLY.
C
      NQ=34
```

```
C
C     READ THE JOB PARAMETERS. FIRST SET UP FOR NAMELIST INPUT.
C
      ASSIGN 30 TO IERRT
      CALL TABDEF(START,5HSTART,5,IERRT)
      CALL TABSET(START,6HRESTRT,RESTRT,IEFLAG,0,0,0,0)
      CALL TABSET(START,4HFILM,FILM,IEFLAG,0,0,0,0)
      CALL TABSET(START,5HPAPER,PAPER,IEFLAG,0,0,0,0)
      CALL TABSET(START,6HWRAPUP,WRAPUP,IEFLAG,0,0,0,0)
C
C     DO THE ACTUAL READ
C
C         RESTRT = IF .TRUE.,THE CODE WILL BE RESTARTED FROM A
C                        DUMP TAPE (DEFAULT)
C                = IF .FALSE.,THE PROBLEM WILL BE GENERATED FROM INPUT
C                        DATA TO FOLLOW
C         PAPER  = IF .TRUE.,OUTPUT WILL OCCUR ON PAPER (DEFAULT=FALSE)
C         FILM   = IF .TRUE.,OUTPUT WILL OCCUR ON FILM (DEFAULT=TRUE)
C         (IF PAPER AND FILM ARE BOTH FALSE,FILM IS SET TO .TRUE.)
C         WRAPUP = NO. OF CP SECONDS TO ALLOW FOR PROBLEM TERMINATION
C                        AFTER THE LAST CYCLE (DEFAULT=20.)
C
      CALL NAMLST(START,5,IEFLAG)
C
C     CHECK FOR ERRORS
C
      IF(IEFLAG.NE.0) CALL UNCLE(4,5HYINIT,23,23HERROR IN START NAMELIST
     1 )
C
C     NO ERRORS. SET THE OUTPUT DO LOOP INDICES.
C
      IPD=6
      IFD=12
      IF(.NOT.PAPER.AND..NOT.FILM) FILM=.TRUE.
      IF(.NOT.PAPER) IPD=12
      IF(.NOT.FILM) IFD=6
C
C     SET UP THE CONSTANT 1/3
C
      THIRD=1./3.
C
C     SET UP THE CONSTANT 2/3
C
      TWTHRD=2.*THIRD
C
C     GET THE JOB ID
C
      CALL GETJOB(JNM)
C
C     GET THE JOB TIME LIMIT
C
      CALL GETJTL(TLIM)
C
C     GIVE A LITTLE TIME FOR WRAPUP
C
      TLIM=TLIM-WRAPUP
C
```

```
C      IF FILM IS TRUE,SET UP OUTPUT FROM NAMELIST ROUTINES TO GO TO
C         FILM AND PAPER BOTH
C
       IF(FILM) CALL NAMPRT(2,FTAB)
C
C      SET LCM TO INDEFINITES
C
       I=0
       ILAST=NLCP1+NLCP2+NLCP3-1
    10 CONTINUE
       CALL ECWR(17770000000000777777B,I,1,IDUM)
       IF(I.EQ.ILAST) GO TO 20
       I=I+1
       GO TO 10
    20 CONTINUE
       RETURN
    30 CONTINUE
       CALL UNCLE(4,5HYINIT,35,35HSTART NAMELIST INITIALIZATION ERROR)
       END
```

```
       SUBROUTINE YINPUT
C
C      ROUTINE TO READ YAQUI INPUT VARIABLES
C
C      WRITTEN BY J.L.NORTON,LASL T-3,1975
C
*      ----- BEGIN COMDECK YSTORE      -----
*      ----- BEGIN COMDECK YAQDIM      -----
       DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
      1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
      2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
      3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
      4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ----- END COMDECK YAQDIM      -----
*      ----- BEGIN COMDECK YAQSC      -----
       LOGICAL RESTRT,FILM,PAPER,TURB
       REAL LAM,MU
C      COMMON/YSC1/AASC(NSCP1)
       COMMON/YSC1/AASC(9600)
       COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
      1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
       COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
       COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
```

```
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,HMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRO,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ***** END COMDECK YAQSC      *****
*     ***** BEGIN COMDECK YAQEQ    *****
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ***** END COMDECK YAQEQ      *****
*     ***** END COMDECK YSTORE     *****
      DIMENSION CARDN(3,49)
      DATA (CARDN(II),II=1,147)/147*0/
      DATA IEFLAG/0/
C
C     SET UP THE NAMELIST INPUT TABLE
C
      ASSIGN 40 TO IERRT
      CALL TABDEF(CARDN,5HCARDN,49,IERRT)
      CALL TABSET(CARDN,2HMU,MU,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HLAM,LAM,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HOM,OM,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HEPS,EPS,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HGR,GR,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HGZ,GZ,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HFREZXR,FREZXR,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HYB,YB,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HREZY0,REZY0,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HREZRON,REZRON,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HREZSIE,REZSIE,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HGZP,GZP,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,1HT,T,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HDT,DT,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HNCLST,NCLST,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HTWFIN,TWFIN,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HNCQ,NCQ,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HQLEVEL,QLEVEL,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HTUQI,TUQI,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HTUSI,TUSI,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HTQ,TQ,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HIST,IST,IEFLAG,0,0,0,0)
```

```
      CALL TABSET(CARDN,4HIBAR,IBAR,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HJBAR,JBAR,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HIUNF,IUNF,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HJUNF,JUNF,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HJCEN,JCEN,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HDR,DR,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HDZ,DZ,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HCYL,CYL,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HGRDVEL,GRDVEL,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HA0,A0,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HA0M,A0M,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,2HB0,B0,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HKXI,KXI,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HANC,ANC,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HA0FAC,A0FAC,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,3HDTO,DTO,IEFLAG,1,0,0,0)
      CALL TABSET(CARDN,4HDTOC,DTOC,IEFLAG,1,0,0,0)
      CALL TABSET(CARDN,6HFREZYT,FREZYT,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HFREZYB,FREZYB,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HZORIG,ZORIG,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,5HJDUMP,JDUMP,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HNAME,NAME,IEFLAG,1,0,0,0)
      CALL TABSET(CARDN,6HWMAXEF,WMAXEF,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HRMINEF,RMINEF,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,6HTSTRTD,TSTRTD,IEFLAG,0,0,0,0)
      CALL TABSET(CARDN,4HIEOF,IEOF,IEFLAG,0,0,0,0)
C
C     READ THE INPUT VARIABLES
C
   10 CONTINUE
      IEOF=0
      CALL NAMLST(CARDN,5,IEFLAG)
C
C     CHECK FOR INPUT ERRORS
C
      IF(IEFLAG.NE.0) CALL UNCLE(4,6HYINPUT,26,
     1 26HCARDN NAMELIST INPUT ERROR)
C
C     NO ERRORS. CHECK FOR EOF.
C
      IF(IEOF.NE.0) GO TO 20
C
C     NO. CONTINUE READING.
C
      GO TO 10
   20 CONTINUE
C
C     CHECK THE VALUES OF THE INPUT PARAMETERS
C
      IF(MU.LT.0.) CALL UNCLE(4,6HYINPUT,8,8HMU.LT.0.)
      IF(LAM.LT.0.) CALL UNCLE(4,6HYINPUT,9,9HLAM.LT.0.)
      IF(OM.LT.1..OR.OM.GT.2.) CALL UNCLE(4,6HYINPUT,10,10HILLEGAL OM)
      IF(EPS.LE.0.) CALL UNCLE(4,6HYINPUT,9,9HEPS.LE.0.)
      IF(FREZXR.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZXR.LT.1.)
      IF(FREZYT.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZYT.LT.1.)
      IF(FREZYB.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZYB.LT.1.)
      IF(REZRON.LE.0.) CALL UNCLE(4,6HYINPUT,12,12HREZRON.LE.0.)
```

```
      IF(REZSIE.LE.0.) CALL UNCLE(4,6HYINPUT,12,12HREZSIE.LE.0.)
      IF(DT.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HDT.LE.0.)
      DO 30 II=2,10
      IF(DTOC(II).EQ.0..AND.DTO(II).EQ.0.) GO TO 30
      IF(DTOC(II).LE.DTOC(II-1)) CALL UNCLE(4,6HYINPUT,22,
     1 22HDTOC MUST BE MONOTONIC)
      IF(DTO(II).LE.0.) CALL UNCLE(4,6HYINPUT,20,20HILLEGAL VALUE OF DTO
     1 )
   30 CONTINUE
      IF(IBAR.LE.0) CALL UNCLE(4,6HYINPUT,9,9HIBAR.LE.0)
      IF(JBAR.LE.0) CALL UNCLE(4,6HYINPUT,9,9HJBAR.LE.0)
      IF(DR.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HDR.LE.0.)
      IF(DZ.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HDZ.LE.0.)
      IF(CYL.NE.1..AND.CYL.NE.0.) CALL UNCLE(4,6HYINPUT,20,
     1 20HILLEGAL VALUE OF CYL)
      IF(GRDVEL.NE.0..AND.GRDVEL.NE.1..AND.GRDVEL.NE.2.) CALL UNCLE(4,
     1 6HYINPUT,23,23HILLEGAL VALUE OF GRDVEL)
      IF(A0.LT.0..OR.A0.GT.1.) CALL UNCLE(4,6HYINPUT,10,10HA0 ILLEGAL)
      IF(B0.LT.0..OR.B0.GT.2.) CALL UNCLE(4,6HYINPUT,10,10HB0 ILLEGAL)
      IF(A0M.LT.0..OR.A0M.GT.1.) CALL UNCLE(4,6HYINPUT,11,11HA0M ILLEGAL
     1 )
      IF(KXI.LT.(-1).OR.KXI.GT.1) CALL UNCLE(4,6HYINPUT,11,
     1 11HKXI ILLEGAL)
      IF(ANC.LE.0.) CALL UNCLE(4,6HYINPUT,9,9HANC.LE.0.)
      IF(A0FAC.LT.0..OR.A0FAC.GT.1.) CALL UNCLE(4,6HYINPUT,13,
     1 13HA0FAC ILLEGAL)
      RETURN
   40 CONTINUE
      CALL UNCLE(4,6HYINPUT,35,35HCARDN NAMELIST INITIALIZATION ERROR)
      END
```

```
      SUBROUTINE YPH1
C
C     ROUTINE TO DO THE EXPLICIT LAGRANGIAN HYDRO (PHASE 1)
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROO(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),QZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
```

```
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ***** END COMDECK YAQSC       *****
*     ***** BEGIN COMDECK YAQEQ     *****
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ***** END COMDECK YAQEQ       *****
*     ***** END COMDECK YSTORE      *****
*     ***** BEGIN COMDECK PARAM     *****
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ***** END COMDECK PARAM       *****
*     ***** BEGIN COMDECK ASTORE    *****
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*     ***** END COMDECK ASTORE      *****
      REAL LAMD,MUO2
C
C     APPLY THE NODE COUPLER TO ALL VERTICES
C
      CALL START
      Y1=ANC*RDT
      DO 100 J=2,JP2
      DO 90 I=1,IP1
      IMJ=IJ-NQ
```

```
      IPJ=IJ+NQ
C
C     XX IS ZERO IF I IS 1 OR IP1, 1 OTHERWISE
C     YY IS ZERO IF J IS 2 OR JP2, 1 OTHERWISE
C
      XX=1.
      YY=1.
C
C     U1,V1 ARE VELOCITIES AT VERTEX (I-1,J) UNLESS I=1 IN WHICH CASE
C         VERTEX (I,J) IS USED
C
      IF(I.EQ.1) GO TO 10
      U1=U(IMJ)
      V1=V(IMJ)
      GO TO 20
   10 XX=0.0
      U1=U(IJ)
      V1=V(IJ)
C
C     U2,V2 ARE VELOCITIES AT VERTEX (I+1,J) UNLESS I=IP1 IN WHICH CASE
C         VERTEX (I,J) IS USED
C
   20 IF(I.EQ.IP1) GO TO 30
      U2=U(IPJ)
      V2=V(IPJ)
      GO TO 40
   30 U2=U(IJ)
      V2=V(IJ)
      XX=0.0
C
C     U3,V3 ARE VELOCITIES AT VERTEX (I,J-1) UNLESS J=2 IN WHICH CASE
C     VERTEX (I,J) IS USED
C
   40 IF(J.EQ.2) GO TO 50
      U3=U(IJM)
      V3=V(IJM)
      GO TO 60
   50 U3=U(IJ)
      V3=V(IJ)
      YY=0.0
C
C     U4,V4 ARE VELOCITIES AT VERTEX (I,J+1) UNLESS J=JP2 IN WHICH CASE
C         VERTEX (I,J) IS USED
C
   60 IF(J.EQ.JP2) GO TO 70
      U4=U(IJP)
      V4=V(IJP)
      GO TO 80
   70 YY=0.0
      U4=U(IJ)
      V4=V(IJ)
C
C     U5,V5 ARE VELOCITIES AT VERTEX (I,J)
C
   80 U5=U(IJ)
      V5=V(IJ)
C
```

```
C     THE NODE COUPLER IS APPLIED TO THE U (V) VELOCITY IF FLAGU (V)
C         IS 1.
C
      FLAGU=0.
      FLAGV=0.
C
C     SET FLAGU AND FLAGV. THE TECHNIQUE IS TO EXAMINE THE THREE
C         VERTICES (I-1,J),(I,J),AND (I+1), OR (I,J-1),(I,J),AND
C         (I,J+1).  IF THE VELOCITY OF VERTEX (I,J) IS THE MINIMUM OR
C         MAXIMUM OF EITHER TRIPLET,THE NODE COUPLER WILL BE APPLIED.
C         OTHERWISE, IT WILL NOT BE APPLIED.
C
      IF(U5.EQ.AMAX1(U1,U2,U5).OR.U5.EQ.AMIN1(U1,U2,U5)) FLAGU=1.0
      IF(V5.EQ.AMAX1(V1,V2,V5).OR.V5.EQ.AMIN1(V1,V2,V5)) FLAGV=1.0
      IF(U5.EQ.AMAX1(U3,U4,U5).OR.U5.EQ.AMIN1(U3,U4,U5)) FLAGU=1.0
      IF(V5.EQ.AMAX1(V3,V4,V5).OR.V5.EQ.AMIN1(V3,V4,V5)) FLAGV=1.0
C
C     UAV,VAV ARE THE DIFFERENCES OF THE AVERAGE VELOCITIES OF THE
C         NEIGHBORING VERTICES AND THE VELOCITIES OF VERTEX (I,J)
C         ITSELF
C
      UAV=0.25*(U1+U2+U3+U4)-U5
      VAV=0.25*(V1+V2+V3+V4)-V5
C
C     CALCULATE THE BODY FORCE ACCELERATIONS. THE SECOND TERM IS
C         THE NODE COUPLER IF ONE IS TO BE APPLIED.
C
      ANCX=Y1*FLAGU*UAV
      ANCY=Y1*FLAGV*VAV
      AX=GR+ANCX
      AY=GZ+ANCY
C
C     UPDATE THE VERTEX VELOCITIES DUE TO THE BODY FORCES AND THE
C         NODE COUPLER.  NOTE THAT THE WALLS ARE RIGID.
C
      UTIL(IJ)=(U(IJ)+DT*AX)*XX
      VTIL(IJ)=(V(IJ)+DT*AY)*YY
C
C     SAVE THE PART OF THE VELOCITY CHANGE DUE TO THE NODE COUPLER
C
      ANCU(IJ)=ANCX*XX*DT
      ANCV(IJ)=ANCY*YY*DT
      IJ=IPJ
      IJP=IJP+NQ
   90 IJM=IJM+NQ
      CALL LOOP
  100 CONTINUE
      CALL DONE
C
C     CALCULATE THE CHANGE TO THE VERTEX VELOCITIES DUE TO THE PRESSURE
C         TERM IN THE MOMENTUM EQN.  LOOP OVER ALL CELLS,NOT VERTICES.
C
      CALL START
      DO 160 J=2,JP1
      DO 150 I=1,IBAR
      ROL(IJ)=0.
      IPJ=IJ+NQ
```

```
      IPJP=IJP+NQ
C
C     VERTEX (I+1,J) IS LABELLED 1, (I+1,J+1) IS 2, (I,J+1) IS 3,
C         AND (I,J) IS 4
C
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      U1=U(IPJ)
      V1=V(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      U2=U(IPJP)
      V2=V(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      U3=U(IJP)
      V3=V(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      U4=U(IJ)
      V4=V(IJ)
C
C     X(Y)NM = X(Y)N-X(Y)M
C
      X24=X2-X4
      Y24=Y2-Y4
      X31=X3-X1
      Y31=Y3-Y1
C
C     UOR IS THE AVERAGE U VELOCITY DIVIDED BY THE AVERAGE X-POSITION
C         OF THE VERTICES,FOR CYLINDRICAL GEOMETRY. FOR SLAB GEOMETRY
C         UOR IS ZERO.
C
      UOR=(U1+U2+U3+U4)/(R1+R2+R3+R4)*CYL
C
C     HRMN IS .5*(RM+RN)
C
      HR13=.5*(R1+R3)
      HR24=.5*(R2+R4)
C
C     DTO2MN = DTO2/(MASS OF VERTEX N)
C
      DTO2M1=DTO2*RM(IPJ)
      DTO2M2=DTO2*RM(IPJP)
      DTO2M3=DTO2*RM(IJP)
      DTO2M4=DTO2*RM(IJ)
C
C     XY IS THE CROSS PRODUCT OF THE DIAGONALS OF THE CELL
C
      XY=X24*Y31-X31*Y24
C
C     CAREA IS THE CELL AREA WHICH IS JUST HALF OF THE CROSS PRODUCT
C         OF THE DIAGONALS
```

```
      C
            CAREA=.5*XY
      C
      C     SAVE THE AREA FOR USE IN THE TURBULENCE CALCULATION
      C         IN TRBCOR
      C
            AREA(IJ)=CAREA
      C
      C     RXY IS THE RECIPROCAL OF XY (HALF THE RECIPROCAL OF THE CELL AREA)
      C
            RXY=1./XY
      C
      C     U(V)IJ=U(V)I-U(V)J
      C
            U24=U2-U4
            U31=U3-U1
            V24=V2-V4
            V31=V3-V1
      C
      C     DU(V)DR(Z) IS THE DIFFERENCE FORM FOR THE DERIVATIVE OF THE
      C         RADIAL (AXIAL) VELOCITY WITH RESPECT TO R (Z) AT THE
      C         CELL CENTER
      C
            DUDR=RXY*(U24*Y31-U31*Y24)
            DUDY=RXY*(X24*U31-X31*U24)
            DVDR=RXY*(V24*Y31-V31*Y24)
            DVDY=RXY*(X24*V31-X31*V24)
      C
      C     COMPUTE DIV(VELOCITY) - VOLUME DILATATION
      C
            D=(DUDR+DVDY)*(1.+UOR*DT)+UOR
      C
      C     SAVE THE VELOCITY DIVERGENCE
      C
            DELSM(IJ)=D
      C
      C     DEFINE THE CELL AREA AS BEING AN AVERAGE DX TIMES AN AVERAGE DY -
      C         CAREA = AVX * AVY. THEN DEFINE AVX(Y) AS THE SQUARE ROOT OF
      C         THE AVERAGE OF THE SQUARES OF THE CELL DIAGONAL X(Y)
      C         COMPONENTS. THEN,ONE WAY OF DETERMINING AVX(Y) GIVEN AVY(X)
      C         IS   AVX(Y)= CAREA/(((Y(X)D1**2+Y(X)D2**2)/2.)**.5 WHERE
      C         Y(X)D1,2 ARE THE CELL DIAGONAL Y(X) COMPONENTS. THUS,
      C         XXA(YYA) IS THE SQUARE OF THE AVERAGE CELL DX(Y)
      C         DETERMINED BY THE PRECEDING METHOD.
      C
            XXA=2./(Y24**2+Y31**2)*CAREA**2
            YYA=2./(X24**2+X31**2)*CAREA**2
      C
      C     DKLSM IS 2.*DT*(1./AVX**2+1./AVY**2) FOR USE IN THE PRESSURE
      C         ITERATION
      C
            DKLSM(IJ)=2.*DT*(XXA+YYA)/(XXA*YYA)
      C
      C     XX AND YY ARE THE ACTUAL AVX AND AVY
      C
            XX=SQRT(XXA)
            YY=SQRT(YYA)
```

```
C
C     CALCULATE THE EFFECTIVE VISCOSITY COEFFICIENTS.
C
C     LAMBDA(MU) EFFECTIVE = LAM(MU) INPUT TIMES AK WHERE THE LATTER
C         IS DEPENDENT ON THE VALUE OF KXI. IF KXI IS 0 OR 1,
C         AK=1. OR RHO(IJ),RESPECTIVELY. IF KXI IS -1, AK IS
C         DETERMINED FROM NUMERICAL STABILITY REQUIREMENTS.
C
      KP=KXI+2
      GO TO (130,110,120),KP
  110 CONTINUE
      AK=1.
      GO TO 140
  120 CONTINUE
      AK=RO(IJ)
      GO TO 140
  130 CONTINUE
      VELIJ=U4*U4+V4*V4
      VELMX=0.7*AMAX1(ABS(U4*XX),ABS(V4*YY))
      AK=RO(IJ)*COLAMU*(DTO2*VELIJ+VELMX)+EM10
  140 CONTINUE
C
C     DETERMINE LAMBDA*DIVERGENCE(VELOCITY)-THIS IS LAMD.
C         NOTE THAT LAMD IS ALWAYS .LE.0.
C
      ROSIG=0.
      IF(TURB) ROSIG=RO(IJ)*SIG(IJ)
      LAMD=AMIN1(D,0.)*AK*LAM-TWTHRD*ROSIG*D
C
C     MUO2 IS HALF MU EFFECTIVE
C
      MUO2=0.5*(AK*MU+ROSIG)
      IF(TURB) CAPGAM(IJ)=2.*(DUDR**2+DVDY**2+.5*(DUDY+DVDR)**2+UOR**2)
C
C     PIXX,PIYY,PIXY,AND PITHTA ARE COMPONENTS OF THE VISCOSITY
C         STRESS TENSOR
C
      PIXX=4.*MUO2*DUDR+LAMD
      PIYY=4.*MUO2*DVDY+LAMD
      PIXY=2.*MUO2*(DUDY+DVDR)
      PITHTA=4.*MUO2*UOR+LAMD*CYL
C
C     PITH IS 1/4 OF THE CELL AREA * PITHTA
C
      PITH=.25*XY*PITHTA
C
C     CALCULATE THE VELOCITY CORRECTIONS FOR THE FOUR CELL VERTICES
C         USING THE EXPLICIT LAGRANGIAN FORM OF THE MOMENTUM EQN.
C         NOTE THAT THE TILDE VELOCITIES WERE INITIALIZED ABOVE IN
C         OOBEUCONBLERE BODY FORCES AND THE EFFECTS OF THE
      TUQYY=P(IJ)
      IF(TURB) TUQYY=TUQYY+TWTHRD*RO(IJ)*TUQ(IJ)
      YY=Y24*TUQYY
      XX=HR24*(PIXY*X24-PIXX*Y24)
      UTIL(IPJ)=UTIL(IPJ)+DTO2M1*(XX+R1*YY-PITH)
```

```
      UTIL(IJP)=UTIL(IJP)-DTO2M3*(XX+R3*YY+PITH)
      XX=HR13*(PIXY*X31-PIXX*Y31)
      YY=Y31*TUQYY
      UTIL(IPJP)=UTIL(IPJP)+DTO2M2*(XX+R2*YY-PITH)
      UTIL(IJ)=UTIL(IJ)-DTO2M4*(XX+R4*YY+PITH)
      PYYMP=PIYY-TUQYY
      XX=HR24*(PYYMP*X24-PIXY*Y24)
      VTIL(IPJ)=VTIL(IPJ)+DTO2M1*XX
      VTIL(IJP)=VTIL(IJP)-DTO2M3*XX
      XX=HR13*(PYYMP*X31-PIXY*Y31)
      VTIL(IPJP)=VTIL(IPJP)+DTO2M2*XX
      VTIL(IJ)=VTIL(IJ)-DTO2M4*XX
      IJ=IPJ
  150 IJP=IPJP
      CALL LOOP
  160 CONTINUE
      CALL DONE
C
C     GO SET THE BOUNDARY CONDITIONS
C
      CALL BC(1)
      RETURN
      END
```

```
      SUBROUTINE YPH2
C
C     ROUTINE TO DO YAQUI PRESSURE ITERATION
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*     ----- BEGIN COMDECK YSTORE      -----
*     ----- BEGIN COMDECK YAQDIM      -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM      -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
```

```
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ     -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
*     ----- END COMDECK YAQEQ      -----
*     ----- END COMDECK YSTORE     -----
*     ----- BEGIN COMDECK PARAM     -----
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK ASTORE     -----
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*     ----- END COMDECK ASTORE     -----
      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
C
C     IF THE TURBULENCE IS ON,GO COMPUTE CORRECTIONS DUE TO IT
C
      IF(TURB) CALL TRBCOR
      CALL START
      DO 20 J=2,JP1
      DO 10 I=1,IBAR
      IMJ=IJ-NQ
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      ROTMP=RO(IJ)
      ETMP=SIE(IJ)
      CONCJ=CONC(IJ)
      CALL AIR
```

```
      GM1=GMONE
      GGM1=GM1*(GM1+1.0)
C     ******** SET DENSITY FOR START OF PRESSURE ITERATION *********
      MTIL(IJ)=RO(IJ)+ROL(IJ)*RVOL(IJ)*DT
      ROL(IJ)=MTIL(IJ)
      RCSQ(IJ)=1./(EM10+GGM1*AMAX1(SIE(IJ),0.))
      IF(TURB.AND.T.GT.TSTRTD) CONC(IJ)=CONC(IJ)+CTEMP(IJ)*RVOL(IJ)*DT
      UG(IJ)=1./(R(IJ)+R(IJP)+R(IPJP)+R(IPJ))*CYL
      X13=X(IPJ)-X(IJP)
      X13K(IJ)=X13
      X24=X(IPJP)-X(IJ)
      X24K(IJ)=X24
      Y13=Y(IPJ)-Y(IJP)
      Y13K(IJ)=Y13
      Y24=Y(IPJP)-Y(IJ)
      Y24K(IJ)=Y24
      VG(IJ)=1./(X13*Y24-X24*Y13)
      XR13K(IJ)=0.5*(R(IPJ)+R(IJP))*X13
      XR24K(IJ)=0.5*(R(IPJP)+R(IJ))*X24
      IJ=IPJ
      IJP=IPJP
   10 IJM=IJM+NQ
      CALL LOOP
   20 CONTINUE
      CALL DONE
      NUMIT=0
      MUSTIT=1
      PLMAX=EM10
   30 CALL START
      DO 60 J=2,JP1
      DO 50 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      U1=UL(IPJ)
      V1=VL(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      U2=UL(IPJP)
      V2=VL(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      U3=UL(IJP)
      V3=VL(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      U4=UL(IJ)
      V4=VL(IJ)
      UOR=(U1+U2+U3+U4)*UG(IJ)
      RAR=VG(IJ)
      DUDX=((U1-U3)*(Y2-Y4)-(U2-U4)*(Y1-Y3))*RAR
      DVDY=((V2-V4)*(X1-X3)-(V1-V3)*(X2-X4))*RAR
```

```
      D=(DUDX+DVDY)*(1.+UOR*DT)+UOR
      S=RDT*(ROL(IJ)-MTIL(IJ))+ROL(IJ)*D
      RA=RCSQ(IJ)*(RDT+D)+DKLSM(IJ)
      DP=-OM*S/RA
      ROL(IJ)=ROL(IJ)+RCSQ(IJ)*DP
      PLMAX=AMAX1(PLMAX,ABS(PL(IJ)))
      IF(ABS(DP).LE.EPS*PLMAX) GO TO 40
      MUSTIT=1
      PL(IJ)=PL(IJ)+DP
      Y24=Y2-Y4
      Y31=Y3-Y1
      XR13=.5*(R1+R3)*(X1-X3)
      XR24=.5*(R2+R4)*(X2-X4)
      XX=DTO2*DP
      DTO2M1=XX*RM(IPJ)
      DTO2M2=XX*RM(IPJP)
      DTO2M3=XX*RM(IJP)
      DTO2M4=XX*RM(IJ)
      UL(IPJ)=U1+DTO2M1*R1*Y24
      UL(IPJP)=U2+DTO2M2*R2*Y31
      UL(IJP)=U3-DTO2M3*R3*Y24
      UL(IJ)=U4-DTO2M4*R4*Y31
      VL(IPJ)=V1-DTO2M1*XR24
      VL(IJ)=V4-DTO2M4*XR13
      VL(IPJP)=V2+DTO2M2*XR13
      VL(IJP)=V3+DTO2M3*XR24
   40 IJ=IPJ
   50 IJP=IPJP
      CALL LOOP
   60 CONTINUE
      CALL DONE
C
C     GO SET THE BOUNDARY CONDITIONS
C
      CALL BC(2)
      NUMIT=NUMIT+1
      IF(MUSTIT.EQ.0) GO TO 80
      MUSTIT=0
      IF(NUMIT.LT.500) GO TO 30
      DO 70 IPX=IPD,IFD,6
   70 WRITE(IPX,90)
   80 CONTINUE
      RETURN
C
   90 FORMAT(42H ITERATION LIMIT EXCEEDED - RUN MAY ABORT.)
      END
```

```
      SUBROUTINE YPH3
C
C     ROUTINE TO FINISH PHASE 1 AND CALCULATE GRID VELOCITIES
C
```

```
C       ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
C       MODIFIED BY J.L.NORTON,LASL T-3,1975
C
*       ----- BEGIN COMDECK YSTORE       -----
*       ----- BEGIN COMDECK YAQDIM       -----
        DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
       1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
       2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
       3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
       4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
       5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
       6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*       ----- END COMDECK YAQDIM         -----
*       ----- BEGIN COMDECK YAQSC        -----
        LOGICAL RESTRT,FILM,PAPER,TURB
        REAL LAM,MU
C       COMMON/YSC1/AASC(NSCP1)
        COMMON/YSC1/AASC(9600)
        COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
       1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
       2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
       3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
       4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
        COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
       1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
       2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
       3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
        COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
       1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
       2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
       3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
       4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
       5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
       6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
        COMMON/YSC2/ZZ
C       COMMON/YSC4/ITAB(ITABP)
        COMMON/YSC4/ITAB(1000)
        COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*       ----- END COMDECK YAQSC          -----
*       ----- BEGIN COMDECK YAQEQ        -----
        EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
       1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
       1 (8),E,ETIL,AREA,XR13K),
       2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
       3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
       4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
       5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
       6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
       7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
       8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
       9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
       1 AVYSV,X24K)
        REAL M,MP,MPAR,MTIL
*       ----- END COMDECK YAQEQ          -----
*       ----- END COMDECK YSTORE         -----
*       ----- BEGIN COMDECK PARAM        -----
        COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
```

```
      1 NLCP3,NLCP4,IFLMSZ
*     ===== END COMDECK PARAM      =====
*     ===== BEGIN COMDECK ASTORE    =====
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*     ===== END COMDECK ASTORE     =====
      REAL LAMD,MUO2
      DTV=1.E30
      CALL START
      DO 40 J=2,JP1
      DO 30 I=1,IBAR
      IMJ=IJ-NQ
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      X1=X(IPJ)
      Y1=Y(IPJ)
      R1=R(IPJ)
      U1L=UL(IPJ)
      U1=U(IPJ)
      V1L=VL(IPJ)
      V1=V(IPJ)
      X2=X(IPJP)
      Y2=Y(IPJP)
      R2=R(IPJP)
      U2L=UL(IPJP)
      U2=U(IPJP)
      V2L=VL(IPJP)
      V2=V(IPJP)
      X3=X(IJP)
      Y3=Y(IJP)
      R3=R(IJP)
      U3L=UL(IJP)
      U3=U(IJP)
      V3L=VL(IJP)
      V3=V(IJP)
      X4=X(IJ)
      Y4=Y(IJ)
      R4=R(IJ)
      U4L=UL(IJ)
      U4=U(IJ)
      V4L=VL(IJ)
      V4=V(IJ)
      X12=X1-X2
      X23=X2-X3
      X34=X3-X4
      X41=X4-X1
      X24=X24K(IJ)
      X31=-X13K(IJ)
      Y24=Y24K(IJ)
      Y31=-Y13K(IJ)
      Y21=Y2-Y1
      Y32=Y3-Y2
      Y43=Y4-Y3
      Y14=Y1-Y4
      HR13=0.5*(R1+R3)
```

```
      HR24=0.5*(R2+R4)
      U12=U1+U2
      U23=U2+U3
      U34=U3+U4
      U41=U4+U1
      U24=U2+U4
      U13=U1+U3
      V12=V1+V2
      V23=V2+V3
      V34=V3+V4
      V41=V4+V1
      V24=V2+V4
      V13=V1+V3
      XY=X24*Y31-X31*Y24
      UOR=(U12+U34)*UG(IJ)
      U24M=U2-U4
      U13M=U1-U3
      V24M=V2-V4
      V13M=V1-V3
      CAREA=.5*XY
      RXY=VG(IJ)
      DUDX=RXY*(U24M*Y31+U13M*Y24)
      DUDY=RXY*(-X24*U13M-X31*U24M)
      DVDX=RXY*(V24M*Y31+V13M*Y24)
      DVDY=RXY*(-X24*V13M-X31*V24M)
      D=(DUDX+DVDY)*(1.+UOR*DT)+UOR
      XX=SQRT(2./(Y24**2+Y31**2))*CAREA
      YY=SQRT(2./(X24**2+X31**2))*CAREA
      IF(KXI.LT.0) GO TO 10
      AK=RO(IJ)**KXI
      GO TO 20
   10 UD4=U4**2+V4**2
      VD4=AMAX1(ABS(U4*XX),ABS(V4*YY))
      AK=RO(IJ)*COLAMU*(DTO2*UD4+VD4*0.7)
   20 ALAM=AK*LAM
      AMU=AK*MU
      LAMD=AMIN1(D,0.)*ALAM
      MUO2=.5*AMU
      FMUO2=4.*MUO2
      PIXX=FMUO2*DUDX+LAMD
      PIYY=FMUO2*DVDY+LAMD
      PIXY=2.*MUO2*(DUDY+DVDX)
      PITH=0.25*XY*(FMUO2*UOR+LAMD*CYL)
      XX1=HR24*(PIXY*X24-PIXX*Y24)
      XX2=HR13*(PIXY*X31-PIXX*Y31)
      XX3=-HR24*PIXY*Y24
      XX4=-HR13*PIXY*Y31
      XX=XX*XX
      YY=YY*YY
      TEMPJ=ALAM+2.*AMU
      IF(TURB) TEMPJ=TEMPJ-RO(IJ)*SIG(IJ)*(TWTHRD*D-2.)
      DQ=RO(IJ)*(1.-ANC)*XX*YY/(2.*TEMPJ*(XX+YY)+EM10)
      DQ=ABS(DQ)
      DTV=AMIN1(.5*DQ,DTV)
      IF(DTVSAV.NE.DTV) IDTV=I
      IF(DTVSAV.NE.DTV) JDTV=J
      DTVSAV=DTV
```

```
      XR13=XR13K(IJ)
      XR24=XR24K(IJ)
      QX=P(IJ)
      RRO=1./RO(IJ)
      RMC=RRO*RVOL(IJ)
      QY=QX-PIYY
      DELE=0.25*DT*RVOL(IJ)*((U1L+U1)*R1*Y24*QX+(U2L+U2)*R2*Y31*QX-(U3L+
     1 U3)*R3*Y24*QX-(U4L+U4)*R4*Y31*QX-(V1L+V1)*XR24*QY+(V2L+V2)*XR13*
     2 QY+(V3L+V3)*XR24*QY-(V4L+V4)*XR13*QY)
      DELE=DELE+0.25*DT*RVOL(IJ)*((U1L+U1)*(XX1-PITH)+(U2L+U2)*(XX2-PITH
     1 )-(U3L+U3)*(XX1+PITH)-(U4L+U4)*(XX2+PITH)+(V1L+V1)*XX3+(V2L+V2)*
     2 XX4-(V3L+V3)*XX3-(V4L+V4)*XX4)
      ETIL(IJ)=SIE(IJ)-DELE*RRO
      IJ=IPJ
   30 IJP=IPJP
      CALL LOOP
   40 CONTINUE
      CALL DONE
      CALL START
      IF(.NOT.TURB) GO TO 50
      TNEGT=TNEG
      TK2=0.
      TI2=0.
      TP2=0.
      TQ2=0.
      TUPOT=0.
      TUPOTA=0.
   50 CONTINUE
      DO 130 J=2,JP2
      DO 120 I=1,IP1
      IF(.NOT.TURB) GO TO 60
      FF=0.5/RM(IJ)
      TK2=TK2+FF*(UL(IJ)**2+VL(IJ)**2)
      TP2=TP2-FF*GZ*DT*(VL(IJ)+V(IJ))
   60 CONTINUE
      IMJ=IJ-NQ
      IPJ=IJ+NQ
      IMJM=IJM-NQ
      VTEMP=-.125/RM(IJ)*(ANCU(IJ)*(UTIL(IJ)+U(IJ))+ANCV(IJ)*(VTIL(IJ)+V
     1 (IJ)))
      XX=1.
      YY=1.
      IF((I.EQ.1).OR.(I.EQ.IP1)) XX=0.
      IF((J.EQ.2).OR.(J.EQ.JP2)) YY=0.
      IF(XX*YY.EQ.0.) GO TO 70
      E(IMJM)=E(IMJM)+VTEMP*RVOL(IMJM)/RO(IMJM)
      E(IJM)=E(IJM)+VTEMP*RVOL(IJM)/RO(IJM)
      E(IMJ)=E(IMJ)+VTEMP*RVOL(IMJ)/RO(IMJ)
      E(IJ)=E(IJ)+VTEMP*RVOL(IJ)/RO(IJ)
      GO TO 110
   70 IF(I.NE.1) GO TO 80
      IF((J.EQ.2).OR.(J.EQ.JP2)) GO TO 110
      E(IJM)=E(IJM)+2.*VTEMP*RVOL(IJM)/RO(IJM)
      E(IJ)=E(IJ)+2.*VTEMP*RVOL(IJ)/RO(IJ)
      GO TO 110
   80 IF(I.NE.IP1) GO TO 90
      IF((J.EQ.2).OR.(J.EQ.JP2)) GO TO 110
```

```
      E(IMJ)=E(IMJ)+2.*VTEMP*RVOL(IMJ)/RO(IMJ)
      E(IMJM)=E(IMJM)+2.*VTEMP*RVOL(IMJM)/RO(IMJM)
      GO TO 110
   90 IF(J.EQ.2) GO TO 100
      E(IMJM)=E(IMJM)+2.*VTEMP*RVOL(IMJM)/RO(IMJM)
      E(IJM)=E(IJM)+2.*VTEMP*RVOL(IJM)/RO(IJM)
      GO TO 110
  100 E(IJ)=E(IJ)+2.*VTEMP*RVOL(IJ)/RO(IJ)
      E(IMJ)=E(IMJ)+2.*VTEMP*RVOL(IMJ)/RO(IMJ)
  110 CONTINUE
      IJ=IPJ
      IJP=IJP+NQ
  120 IJM=IJM+NQ
      CALL LOOP
  130 CONTINUE
      CALL DONE
      CALL START
      IFAKE=IJ
      DO 190 J=2,JP1
      DO 180 I=1,IBAR
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      IF(.NOT.TURB) GO TO 140
      FF=RO(IJ)/RVOL(IJ)
      TI2=TI2+E(IJ)*FF
      TQ2=TQ2+FF*TUQ(IJ)
      IMJ=IJ-NQ
      SS=SIG(IJ)/RVOL(IJ)
      TUPOT=TUPOT-.25*SS*(GRROR(IJ)*(U(IJ)+U(IPJ)+U(IPJP)+U(IJP))+GRROZ(
     1 IJ)*(V(IJ)+V(IPJ)+V(IPJP)+V(IJP)))
      TUPOTA=TUPOTA-SS*GZ*DT*GRROZ(IJ)
  140 CONTINUE
      IF(J.NE.2) GO TO 150
C     *********  SET BOTTOM FICTICIOUS ROW   ***************
      ROL(IJM)=RO(IJM)
      ETIL(IJM)=ETIL(IJ)
      IF(TURB) TUQ(IJM)=TUQ(IJ)
      CONC(IJM)=CONC(IJ)
      IF(I.NE.IBAR) GO TO 150
C     *********  SET LOWER RIGHT FICTICIOUS CORNER  ***********
      ROL(IJM+NQ)=RO(IJM+NQ)
      ETIL(IJM+NQ)=ETIL(IFAKE)
      IF(TURB) TUQ(IJM+NQ)=TUQ(IFAKE)
      CONC(IJM+NQ)=CONC(IFAKE)
      GO TO 160
  150 IF(J.NE.JP1) GO TO 160
C     *********  SET TOP FICTICIOUS ROW *********
      ROL(IJP)=RO(IJP)
      IF(TURB) TUQ(IJP)=TUQ(IJ)
      CONC(IJP)=CONC(IJ)
  160 IF(I.NE.IBAR) GO TO 170
C     *********  SET RIGHT HAND FICTICIOUS COLUMN  *********
      ROL(IPJ)=RO(IPJ)
      ETIL(IPJ)=ETIL(IJ)
      IF(TURB) TUQ(IPJ)=TUQ(IJ)
      CONC(IPJ)=CONC(IJ)
  170 IJ=IPJ
```

```
      IJP=IPJP
  180 IJM=IJM+NQ
      CALL LOOP
  190 CONTINUE
      CALL DONE
      IF(.NOT.TURB) GO TO 200
      TDQ=TQ2-TUQENG
      TE=TK+TI+TUQENG
      DELTAE=EP1+TUPOT+TUPOTA
      DELEWR=DELTAE
      FIDEL=DELEWR-SAV1
      IF(TDQSAV.NE.0.) ERATIO=FIDEL/TDQSAV
      TDQSAV=TDQ
      SAV1=DELTAE
  200 CONTINUE
      IF(GRDVEL.GT.1.99) GO TO 250
      CALL START
      DO 220 J=2,JP2
      DO 210 I=1,IP1
      UG(IJ)=0.5*GRDVEL*(UL(IJ)+U(IJ))
      VG(IJ)=0.5*GRDVEL*(VL(IJ)+V(IJ))
  210 IJ=IJ+NQ
      CALL LOOP
  220 CONTINUE
      CALL DONE
      CALL START
      DO 240 J=2,JP2
      DO 230 I=1,IP1
      X(IJ)=X(IJ)+UG(IJ)*DT
      Y(IJ)=Y(IJ)+VG(IJ)*DT
      R(IJ)=X(IJ)*CYL+OMCYL
  230 IJ=IJ+NQ
      CALL LOOP
  240 CONTINUE
      CALL DONE
      GO TO 260
  250 CALL REZONE
  260 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE YPLOT
C
C     ROUTINE TO DO PLOTTING FOR YAQUI
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*     ***** BEGIN COMDECK PARAM      *****
      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
     1 NLCP3,NLCP4,IFLMSZ
```

```
*     ----- END COMDECK PARAM      -----
*     ----- BEGIN COMDECK ASTORE     -----
      COMMON/ASTC/AT(100),FT(100)
      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
     1 ),(AT(9),YCO),(FT,CON)
*     ----- END COMDECK ASTORE     -----
*     ----- BEGIN COMDECK YSTORE     -----
*     ----- BEGIN COMDECK YAQDIM     -----
      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*     ----- END COMDECK YAQDIM     -----
*     ----- BEGIN COMDECK YAQSC      -----
      LOGICAL RESTRT,FILM,PAPER,TURB
      REAL LAM,MU
C     COMMON/YSC1/AASC(NSCP1)
      COMMON/YSC1/AASC(9600)
      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*     ----- END COMDECK YAQSC      -----
*     ----- BEGIN COMDECK YAQEQ      -----
      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
     4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
     9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
     1 AVYSV,X24K)
      REAL M,MP,MPAR,MTIL
```

```
*       ***** END COMDECK YAGEQ       *****
*       ***** END COMDECK YSTORE      *****
*       ***** BEGIN COMDECK PCALL       *****
        COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*       ***** END COMDECK PCALL       *****
        LOGICAL EULER
        DIMENSION BCD(2)
        DIMENSION ITITLE(6,8),INWT(8)
        DIMENSION IJL1(6),IJL2(6),IJL3(6),IJL4(6),IJL5(6),IJL6(6)
        DIMENSION IJL7(6),IJL8(6)
        EQUIVALENCE(ITITLE,IJL1),(ITITLE(7),IJL2),(ITITLE(13),IJL3),(
       1 ITITLE(19),IJL4),(ITITLE(25),IJL5),(ITITLE(31),IJL6),(ITITLE(37),
       2 IJL7),(ITITLE(43),IJL8)
        DATA IJL1/7HDENSITY,5*1H /
        DATA IJL2/24HSPECIFIC INTERNAL ENERGY,3*1H /
        DATA IJL3/9HVORTICITY,5*1H /
        DATA IJL4/18HVELOCITY MAGNITUDE,4*1H /
        DATA IJL5/26HSPECIFIC TURBULENCE ENERGY,3*1H /
        DATA IJL6/30HRATIO OF SPEC. TURB. ENERGY TO,
       1 21H SPEC. KINETIC ENERGY/
        DATA IJL7/28HTURBULENCE VISCOSITY (SIGMA),3*1H /
        DATA IJL8/22HCONCENTRATION X RADIUS,3*1H /
        DATA INWT/1,3,1,2,3,6,3,3/
        DATA BCD/1H /
        DATA VMAX,DRMIN,DZMIN,DRMAX,DZMAX/5*0./,IVM,JVM/0,0/
        DATA TP/0./
        CALL SECOND(TP)
        WRITE(59,210) TP
C
C       IF THERE ARE ANY PARTICLES, GO PLOT THEM
C
        IF(NPT.GT.0) CALL PARPLT
C
C       EULER IS TRUE IF THE CALCULATION IS EULERIAN
C
        EULER=.FALSE.
        IF(GRDVFL.EQ.0.) EULER=.TRUE.
C
C       INITIALIZE COORDINATE-TO-RASTER CONVERSION FACTORS IN COMMON
C
        YCONVP=YCONV
        XCONVP=XCONV
C
C       INITIALIZE MINIMUM PLOT Y VALUE IN COMMON
C
        YLB=YB
C
C       GO GET THE MAXIMUM VELOCITY ABSOLUTE VALUE ALONG THE COORDINATE
C           DIRECTIONS AND THE MINIMUM AND MAXIMUM ZONE SIZES
C
        CALL DVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
        VTEST=.05*VMAX
C
C       IF DOING AN EULERIAN CALCULATION,ONLY DO ZONE PLOTS UPON
C           STARTUP. OTHERWISE,DO THEM EVERY OUTPUT CYCLE.
C
        IF(EULER.AND.NCYC.NE.0) GO TO 20
```

```
      CALL ADV(1)
      CALL ZONPLT
C
C     PUT LABELS ON THE ZONE PLOT AND OUT TO THE CYCLE SUMMARY PRINT
C
      CALL LINCNT(59)
      WRITE(IFD,220)
      DO 10 IPX=6,IFD,6
   10 WRITE(IPX,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
      WRITE(IFD,260) JNM,NAME,T,NCYC
   20 CONTINUE
C
C     SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO.
C         OTHERWISE,PLOT VELOCITIES SCALED TO THE MAXIMUM VELOCITY.
C
      IF(VMAX.GT.EM10) CALL VELPLT(VMAX,0)
C
C     THE FULL PLOTS HAVE BEEN MADE. ADJUST PLOTS NOW SO THAT THE
C         REGION DISPLAYED INCLUDES THAT PART OF THE PROBLEM TWO
C         FIREBALL RADII ABOVE THE TOP OF THE FIREBALL TO THREE RADII
C         BELOW THE BOTTOM OF THE FIREBALL
C
      YUP=PTOP+2.*PRITE
      YLB=PBOTM-3.*PRITE
      IF(YUP.GT.YT) YUP=YT
      IF(YLB.LT.YB) YLB=YB
C
C     READJUST THE CARTESIAN TO RASTER RATIOS TO ENCOMPASS THE NEW
C         REGION BUT SAVE THE OLD COORDINATES
C
      FIYBO=FIYB
      IYBO=IYB
      FIYB=900.
      IYB=900
      YCONVP=FLOAT(IYT-IYB)/(YUP-YLB)
      XCONVP=XCONV*YCONVP/YCONV
      FIXLO=FIXL
      IXLO=IXL
      FIXRO=FIXR
      IXRO=IXR
      FIXL=65.
      IXL=65
      FIXR=660.
      IXR=660
      YCNVLD=YCONV
      YCONV=YCONVP
      XCNVLD=XCONV
      XCONV=XCONVP
C
C     DO THE PLOTS AGAIN FOR THE SMALLER REGION
C
      IF(EULER.AND.NCYC.NE.0) GO TO 30
      CALL ADV(1)
      CALL ZONPLT
C
C     PUT LABELS ON THE ZONE PLOT
C
```

```
      CALL LINCNT(59)
      WRITE(IFD,230)
      WRITE(IFD,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
      WRITE(IFD,260) JNM,NAME,T,NCYC
C
C     DRAW THE PLOTTING RECTANGLE AND LABEL IT
C
      CALL TICBOX
   30 CONTINUE
C
C     SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO
C
      IF(VMAX.LT.EM10) GO TO 40
C
C     PLOT VELOCITIES SCALED TO MAXIMUM VELOCITY
C
      CALL VELPLT(VMAX,0)
      CALL TICBOX
C
C     PLOT UNSCALED VELOCITIES
C
      CALL VELPLT(VMAX,1)
      CALL TICBOX
   40 CONTINUE
C
C     SECTION TO DO CONTOUR PLOTS
C
C     L=1, DENSITY (ISOPYCNICS)
C     L=2, SPECIFIC INTERNAL ENERGY (ISOTHERMS)
C     L=3, VORTICITY
C     L=4, VELOCITY MAGNITUDE
C     L=5, SPECIFIC TURBULENCE ENERGY
C     L=6, RATIO OF SPEC. TURB. ENERGY TO SPEC. KINETIC ENERGY
C     L=7, TURBULENCE VISCOSITY (SIGMA)
C     L=8, CONCENTRATION X RADIUS
C
C     IF TURBULENCE IS NOT ON,DO NOT PLOT THE TURBULENCE QUANTITIES
C
      LP=8
      IF(.NOT.TURB) LP=4
C
C     PREPARE THE PLOTTING ARRAYS
C
      DO 200 L=1,LP
C
C     CHECK FOR VORTICITY PLOT
C
      IF(L.NE.3) GO TO 50
C
C     YES. SPECIAL SUBROUTINE PREPARES VORTICITY.
C
      CALL GETOMG
      GO TO 190
   50 CONTINUE
      CALL START
      DO 180 J=2,JP1
      DO 170 I=1,IBAR
```

```
      GO TO (60,70,200,80,80,80,140,80),L
   60 CONTINUE
C
C     DENSITY
C
      CQ(IJ)=RQ(IJ)
      GO TO 160
   70 CONTINUE
C
C     SPECIFIC INTERNAL ENERGY
C
      CQ(IJ)=SIE(IJ)
      GO TO 160
   80 CONTINUE
      IPJ=IJ+NQ
      IPJP=IJP+NQ
      IF(L.EQ.5.OR.L.EQ.8) GO TO 90
      XXA=(0.25*(U(IJ)+U(IPJ)+U(IPJP)+U(IJP)))**2+(0.25*(V(IJ)+V(IPJ)+V(
     1 IPJP)+V(IJP)))**2
      GO TO 100
   90 CONTINUE
      RAV=0.25*(R(IJ)+R(IPJ)+R(IPJP)+R(IJP))
  100 CONTINUE
      LPX=L-3
      GO TO (110,120,130,140,150),LPX
  110 CONTINUE
C
C     VELOCITY MAGNITUDE
C
      CQ(IJ)=SQRT(XXA)
      GO TO 160
  120 CONTINUE
C
C     SPECIFIC TURBULENCE ENERGY
C
      CQ(IJ)=TUQ(IJ)
      GO TO 160
  130 CONTINUE
C
C     RATIO OF SPEC. TURB. ENERGY TO SPEC. KINETIC ENERGY
C
      CQ(IJ)=TUQ(IJ)/(XXA+EM10)*2.
      TEST=AMAX1(ABS(U(IJ)),ABS(V(IJ)))
      IF(TEST.LT.VTEST) CQ(IJ)=0.
      GO TO 160
  140 CONTINUE
C
C     TURBULENCE VISCOSITY (SIGMA)
C
      CQ(IJ)=SIG(IJ)
      GO TO 160
  150 CONTINUE
C
C     CONCENTRATION X RADIUS
C
      CQ(IJ)=CONC(IJ)*RAV
  160 CONTINUE
```

```
      IF(ABS(CQ(IJ)).LT.1.E-50) CQ(IJ)=0.
      IJP=IJP+NQ
  170 IJ=IJ+NQ
      CALL LOOP
  180 CONTINUE
      CALL DONE
  190 CONTINUE
C
C     DECIDE WHETHER TO REQUEST LOGARITHMIC OR LINEAR CONTOUR INCREMENTS
C
C     PLOTS 1 AND 2 ARE  LOGARITHMIC - THE OTHERS LINEAR
C
      ILOG=1
      IF(L.GT.2) ILOG=0
C
C     GO DO THE PLOT
C
C     ITITLE IS THE PLOT TITLE
C     INWT IS THE NO. OF COMPUTER WORDS IN THE TITLE
C
      CALL CONTOR(ILOG,ITITLE(1,L),INWT(L))
  200 CONTINUE
C
C     RESTORE THE OLD RASTER VALUES
C
      FIYB=FIYBO
      IYB=IYBO
      YCONV=YCNVLD
      XCONV=XCNVLD
      FIXR=FIXRO
      FIXL=FIXLO
      IXR=IXRO
      IXL=IXLO
      CALL SECOND(TP)
      WRITE(59,240) TP
      RETURN
C
  210 FORMAT(1H ,20HBEGIN YPLOT AT CP = ,F10.4)
  220 FORMAT(10H ALL ZONES)
  230 FORMAT(29H ZONES IN THE FIREBALL REGION)
  240 FORMAT(1H ,18HEND YPLOT AT CP = ,F10.4)
  250 FORMAT(7H DRMIN=1PE12.5,7H DRMAX=E12.5,7H DZMIN=E12.5/7H DZMAX=E12
     1 .5,4H XR=E12.5,4H YB=E12.5,4H YT=E12.5)
  260 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
      END
```

```
      SUBROUTINE ZONPLT
C
C     ROUTINE TO DO FULL AND PARTIAL ZONE PLOTS
C
C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
```

```
C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
C
*      ----- BEGIN COMDECK PARAM       -----
       COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
      1 NLCP3,NLCP4,IFLMSZ
*      ----- END COMDECK PARAM       -----
*      ----- BEGIN COMDECK YSTORE      -----
*      ----- BEGIN COMDECK YAQDIM      -----
       DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
      1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
      2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
      3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
      4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
*      ----- END COMDECK YAQDIM      -----
*      ----- BEGIN COMDECK YAQSC      -----
       LOGICAL RESTRT,FILM,PAPER,TURB
       REAL LAM,MU
C      COMMON/YSC1/AASC(NSCP1)
       COMMON/YSC1/AASC(9600)
       COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
      1 DTO(10),DTOC(10),DTD2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
       COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
       COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
       COMMON/YSC2/ZZ
C      COMMON/YSC4/ITAB(ITABP)
       COMMON/YSC4/ITAB(1000)
       COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
*      ----- END COMDECK YAQSC      -----
*      ----- BEGIN COMDECK YAQEQ      -----
       EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
      1 (8),E,ETIL,AREA,XR13K),
      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
      1 AVYSV,X24K)
       REAL M,MP,MPAR,MTIL
*      ----- END COMDECK YAQEQ      -----
```

```
*      ***** END COMDECK YSTORE     *****
*      ***** BEGIN COMDECK PCALL      *****
       COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
*      ***** END COMDECK PCALL      *****
C      LOOP OVER ALL REAL ZONES
C
       JB=0
       CALL START
       DO 60 J=2,JP1
       DO 50 I=1,IBAR
       IPJ=IJ+NQ
       IPJP=IJP+NQ
C
C      DETERMINE THE FOUR VERTICES OF CELL (I,J)
C
C      (X1,Y1) IS VERTEX (I+1,J)     (VERTEX 1)
C      (X2,Y2) IS VERTEX (I+1,J+1)   (VERTEX 2)
C      (X3,Y3) IS VERTEX (I,J+1)     (VERTEX 3)
C      (X4,Y4) IS VERTEX (I,J)       (VERTEX 4)
C
       X1=X(IPJ)
       X2=X(IPJP)
       X3=X(IJP)
       X4=X(IJ)
       Y1=Y(IPJ)
       Y2=Y(IPJP)
       Y3=Y(IJP)
       Y4=Y(IJ)
C
C      (IXN,IYN) ARE THE PLOT RASTER COORDINATES OF VERTEX N
C
       IX1=FIXL+(X1-XL)*XCONVP
       IX2=FIXL+(X2-XL)*XCONVP
       IX3=FIXL+(X3-XL)*XCONVP
       IX4=FIXL+(X4-XL)*XCONVP
       IY1=FIYB+(Y1-YLB)*YCONVP
       IY2=FIYB+(Y2-YLB)*YCONVP
       IY3=FIYB+(Y3-YLB)*YCONVP
       IY4=FIYB+(Y4-YLB)*YCONVP
C
C      IF VERTEX 1 OR 2 IS ABOVE THE TOP OR BELOW THE BOTTOM OF THE
C          PLOT REGION,DO NOT PLOT THIS ZONE
C
       IF(IY1.GT.IYB.OR.IY1.LT.IYT) GO TO 40
       IF(IY2.GT.IYB.OR.IY2.LT.IYT) GO TO 40
C
C      IF VERTEX 1 OR 2 IS TO THE RIGHT OF THE PLOT REGION,SKIP
C          THIS ZONE
C
       IF(IX1.GT.IXR) GO TO 40
       IF(IX2.GT.IXR) GO TO 40
C
C      IF THIS ZONE IS ON THE SYMMETRY AXIS,DRAW THE LEFT SIDE
C
       IF(I.EQ.1) CALL DRV(IX3,IY3,IX4,IY4)
C
C      IF THIS ZONE IS IN THE BOTTOMMOST ROW TO BE PLOTTED,DRAW THE
```

```
C          BOTTOM SIDE
C
      IF(JB.NE.0) GO TO 10
      JB=J
      GO TO 20
   10 CONTINUE
      IF(J.NE.JB) GO TO 30
   20 CONTINUE
      CALL DRV(IX4,IY4,IX1,IY1)
   30 CONTINUE
C
C     DRAW THE RIGHT SIDE
C
      CALL DRV(IX1,IY1,IX2,IY2)
C
C     DRAW THE TOP SIDE
C
      CALL DRV(IX2,IY2,IX3,IY3)
   40 IJ=IPJ
   50 IJP=IPJP
      CALL LOOP
   60 CONTINUE
      RETURN
      END
```

REFERENCES

1. F. H. Harlow and A. A. Amsden, "A Numerical Fluid Dynamics Calculation Method for All Flow Speeds," J. Comp. Phys. 8, 197 (1971).
2. C. W. Hirt, A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," J. Comp. 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. W. E. Pracht, "Calculating Three-Dimensional Fluid Flows at All Speeds with an Eulerian-Lagrangian Computing Mesh," J. Comp. Phys. 17, 132 (1975).
5. H. M. Ruppel, R. A. Gentry, and B. J. Daly, "Simulation of Turbulence in Fireballs," Los Alamos Scientific Laboratory report LA-5449-MS (November 1973).
6. H. M. Ruppel and J. L. Norton, "Theoretical Simulations of the Gas Explosive Simulation Technique (GEST) Experiments," Los Alamos Scientific Laboratory report LA-6154-MS (December 1975).

☆ U.S. GOVERNMENT PRINTING OFFICE: 1976—777-018/32