LA-6261-M
Manual



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

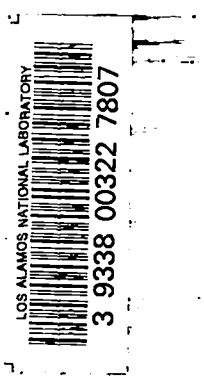
UNITED STATES
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
CONTRACT W-7405-ENG. 36

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





# 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 JCEN$^{th}$ 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

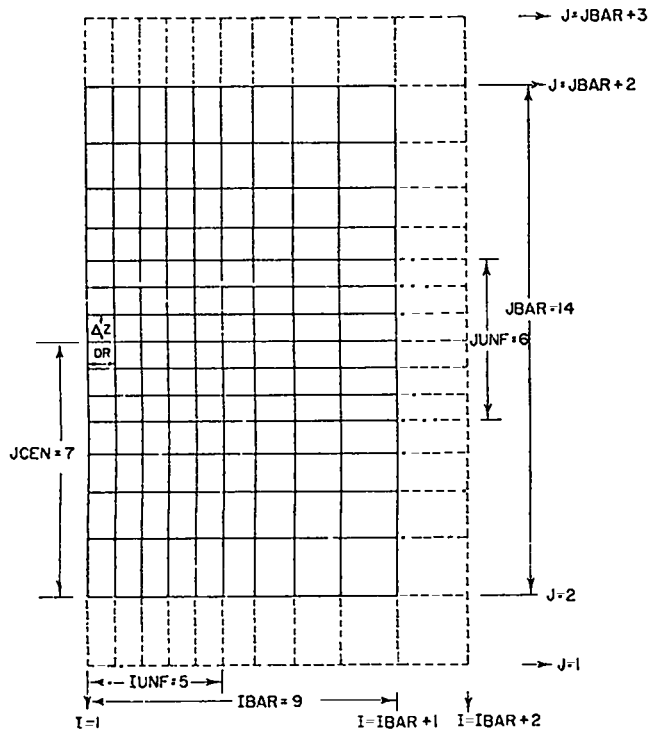



Fig. 1. Sample initial zoning setup.

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

The top of the JCEN$^{th}$ 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 $\rho_i$, 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

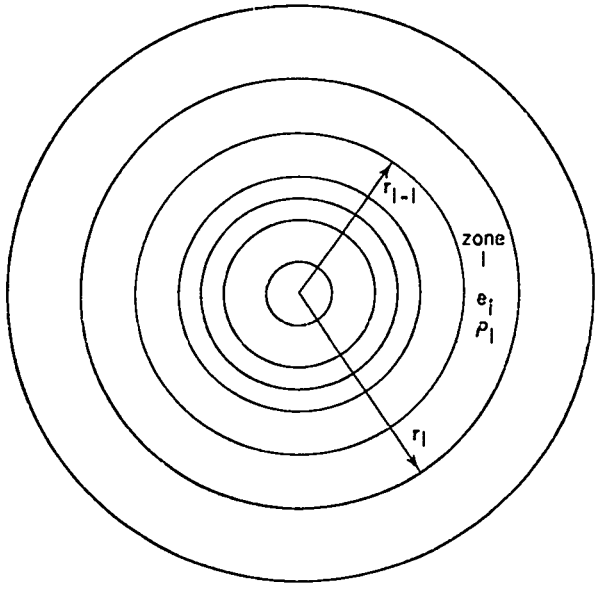



Fig. 2. Sample 1D fireball initial data configuration.

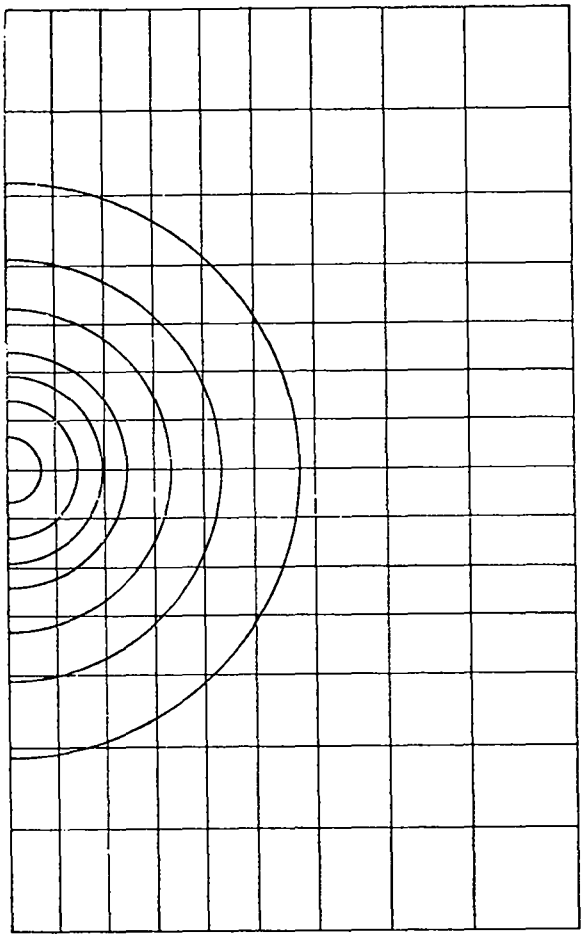


Fig. 3. Superposition of 1D fireball data on the 2D YAQUI grid.

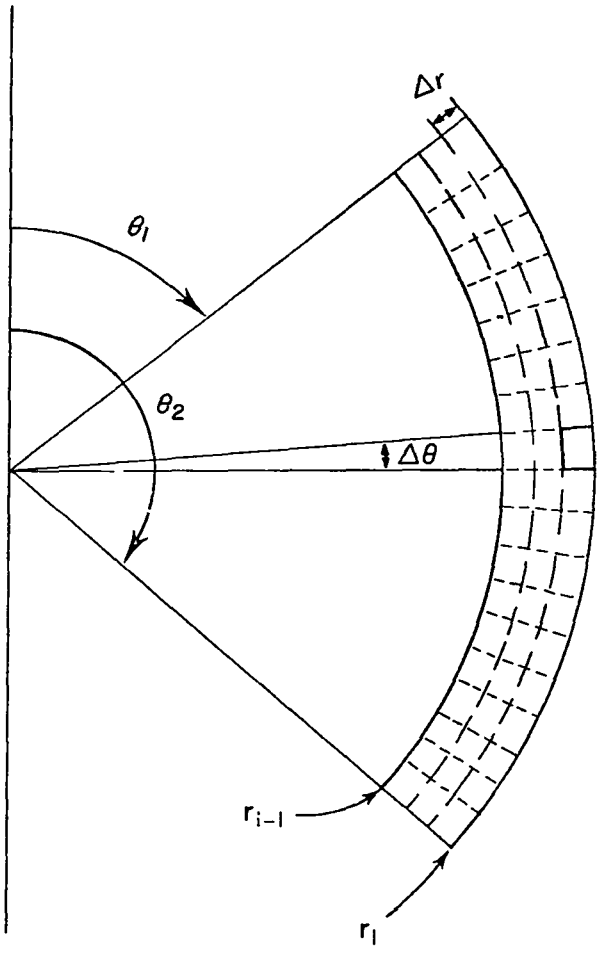



Fig. 4. Division of 1D fireball data into subzones for interpolation purposes.

$\theta_1$ is 0° and $\theta_2$ is 180°. $\Delta\theta$ is determined from 180°/NTH, where NTH is an input quantity; $\Delta r$ is determined from $\frac{r_i - r_{i-1}}{NRAD}$, 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 center of the subzone falls. Assign all 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.

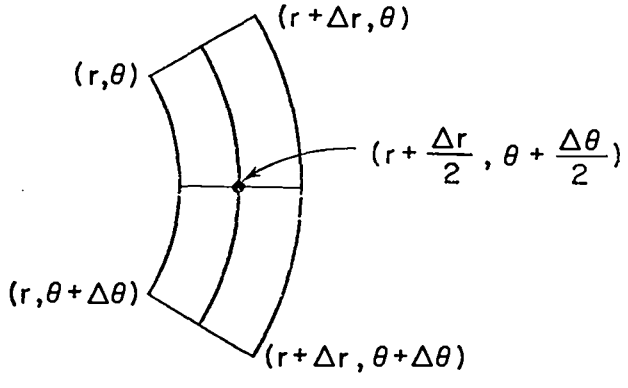



Fig. 5. A typical 1D Interpolation subzone.

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 ($\Delta r$ 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.

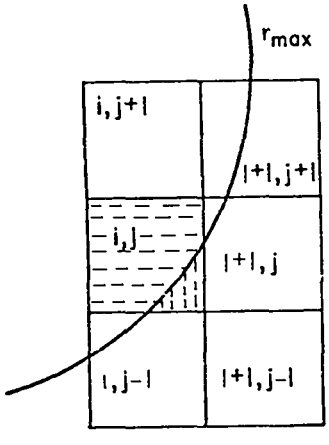



Fig. 6. Technique for handling YAQUI zones only partially within the fireball.

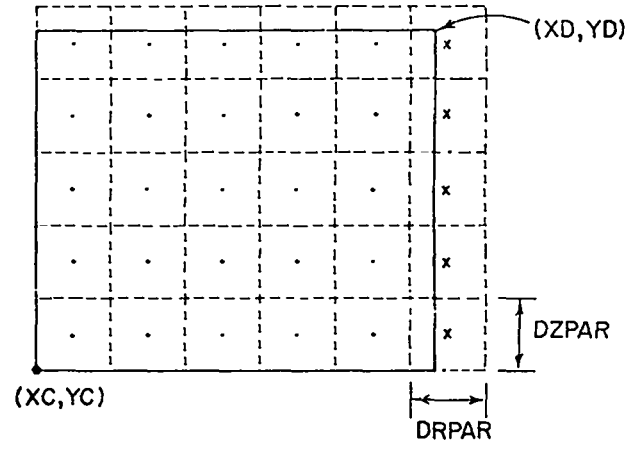



Fig. 7. Particle generation scheme.

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 $\Delta x$ and $\Delta y$ 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 $\pi$ 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 $\pi$, and the particle lies outside the cell.

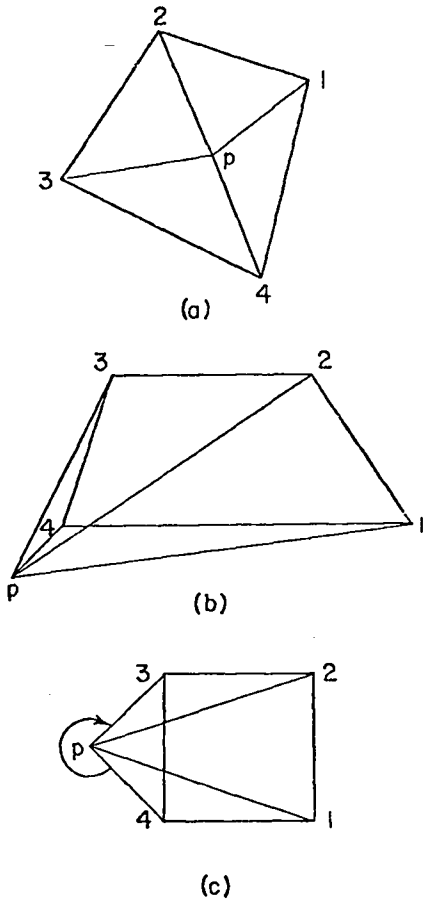



Fig. 8. Determination of particle positions.

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 η = 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:

$$\Pi_{ij} = (\mu+\rho\sigma)e_{ij} + \sigma H_{ij} + \delta_{ij}\left[\left(\lambda-\frac{2}{3}\rho\sigma\right)\frac{\partial u_k}{\partial x_k} - \left(p+\frac{2}{3}\rho q\right)\right]\ , \quad (8)$$

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

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

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

The parameters of the model are $\beta$ and the scale, s. Values are obtained phenomenologically from fits to experiment. For $\beta$ 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

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

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 $\mu \to \mu + \rho\sigma$, $\lambda \to \lambda - \frac{2}{3}\rho\sigma$ and $p \to p + \frac{2}{3}\rho q$. Similarly $H_{ij}$ is calculated from $\rho$ 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 $\Pi_{ij}$. 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 $\Gamma$ takes the form

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

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

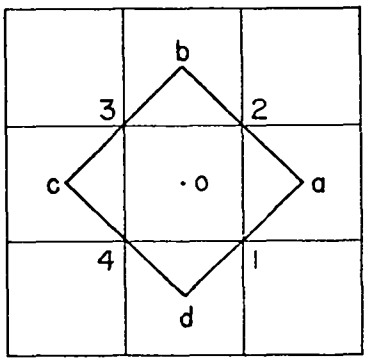



Fig. 9. Quantities involved in YAQUI derivative evaluation.

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 $\delta x$, $\delta y$, and constant $\sigma$, 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 $\rho q$ 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.

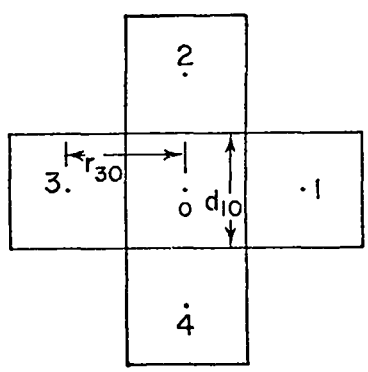



Fig. 10. Definition of quantities used in differencing of turbulence part of the concentration equation.

As far as convection is concerned, the quantity $\rho q$ 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 $\rho$ is the density and $\lambda$ 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 $\lambda$. 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 $\delta t$ and $\delta x$, 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, $\delta x$ 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 $\sigma$ 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 $\delta t$.

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

$$p(x) = 1 \ , \quad 0 \leq x \leq 1$$
$$= 0 \ , \quad \text{elsewhere.} \tag{25}$$

What we need is the distribution given by Eq. (23) with $\sigma$ 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,

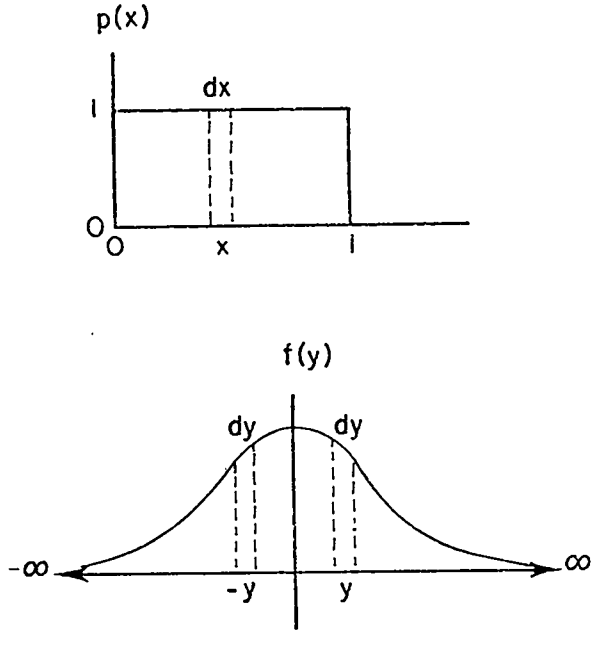



Fig. 11. (a) Uniform distribution, (b) Gaussian distribution.

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

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

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

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

By definition, the standard error function is

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

therefore,

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

and from Eq. (24),

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

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 $\lambda$ 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 $|\text{erf}(y)|$ approaches 1 asymptotically (see Fig. 12). Care must be taken in using $\text{erf}^{-1}(x)$ for $x \sim 1$, for in this region $\text{erf}^{-1}(x)$ is very large which could lead to an abnormally large particle displacement. To

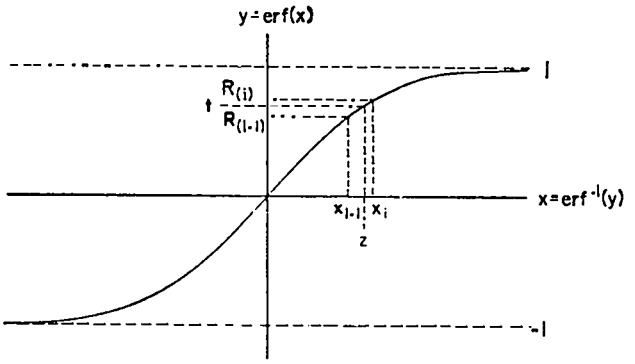



Fig. 12. Standard error function.

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

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

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

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

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

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

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

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

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

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

or

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

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

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

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 $\Delta r$'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 $\gamma_{meth}$ to refer to the effective $\gamma$ 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 $\gamma$ 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) | $0. \leq WRAPUP < \infty$ | 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) | $0. \leq AO \leq 1.$ | 0.1 |
| AOFAC | Real (none) | $0. < AOFAC \leq .5$ | 0.2 |
| AOM | Real (none) | $0. \leq AOM \leq 1.$ | 1.0 |
| ANC | Real (none) | $0. \leq ANC \leq .2$ | 0.05 |
| BO | Real (none) | 0., 2. | 0.0 |
| CYL | Real (none) | 0., 1. | 1.0 |
| DR | Real (cm) | $0. < DR < \infty$ | Must be specified |
| DT | Real (s) | $0. < DT < \infty$ | $10^{-3}$ |
| DTØ | Real (s) | $0. < DTØ < \infty$ | 1.0 |
| DTØC | Real (s) | $0. < DTØC < \infty$ | $10^{30}$ |
| DZ | Real (cm) | $0. < DZ < \infty$ | Must be specified |
| EPS | Real (none) | $0. < EPS < 1.$ | $10^{-5}$ |
| FREZXR | Real (none) | $1. \leq FREZXR < \infty$ | 1.0 |
| FREZYB | Real (none) | $1. \leq FREZYB < \infty$ | 1.0 |
| FREZYT | Real (none) | $1. \leq FREZYT < \infty$ | 1.0 |
| GR | Real ($cm/s^2$) | $-\infty < GR < \infty$ | 0.0 |
| GRDVEL | Real (none) | 0., 1., 2. | 2.0 |
| GZ | Real ($cm/s^2$) | $-\infty < GZ < \infty$ | -980.0 |
| GZP | Real ($cm/s^2$) | $-\infty < GZP < \infty$ | 0.0 |
| IBAR | Integer (none) | $1 \leq IBAR <$ Storage | Must be specified |
| IEOF | Integer (none) | 0, 1 | 0 |
| IST | Integer (none) | -1, $1 \leq IST <$ storage | -1 |
| IUNF | Integer (none) | $0 \leq IUNF \leq IBAR$ | Must be specified |
| JBAR | Integer (none) | $1 \leq JBAR <$ storage | Must be specified |
| JCEN | Integer (none) | $1 \leq JCEN \leq JBAR$ | Must be specified |
| JDUMP | Integer (none) | $1 \leq JDUMP < \infty$ | 999999 |
| JUNF | Integer (none) | $0 \leq JUNF \leq JBAR$ | Must be specified |
| KXI | Integer (none) | -1, 0, 1 | -1 |
| LAM | Real (see Ref. 3) | $0. \leq LAM < \infty$ | 0.6 |
| MU | Real | $0. \leq MU < \infty$ | 0.0 |
| NCLST | Integer (none) | $1 \leq NCLST < \infty$ | 999999 |
| NCQ | Integer (none) | $-\infty < NCQ < \infty$ | -1 |
| ØM | Real (none) | $0. < ØM < 2.$ | 1.0 |
| QLEVEL | Real (none) | .02 | 0.02 |
| REXRØN | Real ($gm/cm^3$) | $0. < REZRØN < \infty$ | 0.001 |
| REZSIE | Real ($cm^2/s^2$) | $0. < REZSIE < \infty$ | $2 \times 10^{10}$ |
| REZYO | Real (cm) | $YB < REZYO < \infty$ | 0.0 |
| RMINEF | Real (cm) | $0. \leq RMINEF < \infty$ | 50.0 |
| T | Real (s) | $0. \leq T < \infty$ | 0.0 |
| TQ | Real (s) | $0. \leq TQ < \infty$ | 0.0 |
| TSTRTD | Real (s) | $0. \leq TSTRTD < \infty$ | 1.0 |
| TUQI | Real ($cm^2 \cdot s$) | $0. < TUQI < \infty$ | 0.0 |
| TUSI | Real (cm) | $0. < TUSI \leq$ fireball radius | 0.0 |
| TWFIN | Real (s) | $0. < TWFIN < \infty$ | 1.E30 |
| WMAXEF | Real (none) | $0. < WMAXEF < \infty$ | 2.0 |
| YB | Real (cm) | $0. \leq YB < \infty$ | 0.0 |
| ZØRIG | Real (none) | $0. \leq ZØRIG < \infty$ | 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 $DT\emptyset C(I-1) < T \leq DT\emptyset C(I)$. $DT\emptyset C(0) \equiv 0$.

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 $\Delta r$'s in the right region of nonuniform zoning (see Sec. I.A).

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

FREZYT - Geometric ratio of zone $\Delta z$'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 $\leq$ 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).

$\emptyset M$ - Relaxation parameter in the pressure iteration.

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

REZR$\emptyset$N - 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$\leq$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 ~t = 1.25 s.

### A. Initial Conditions

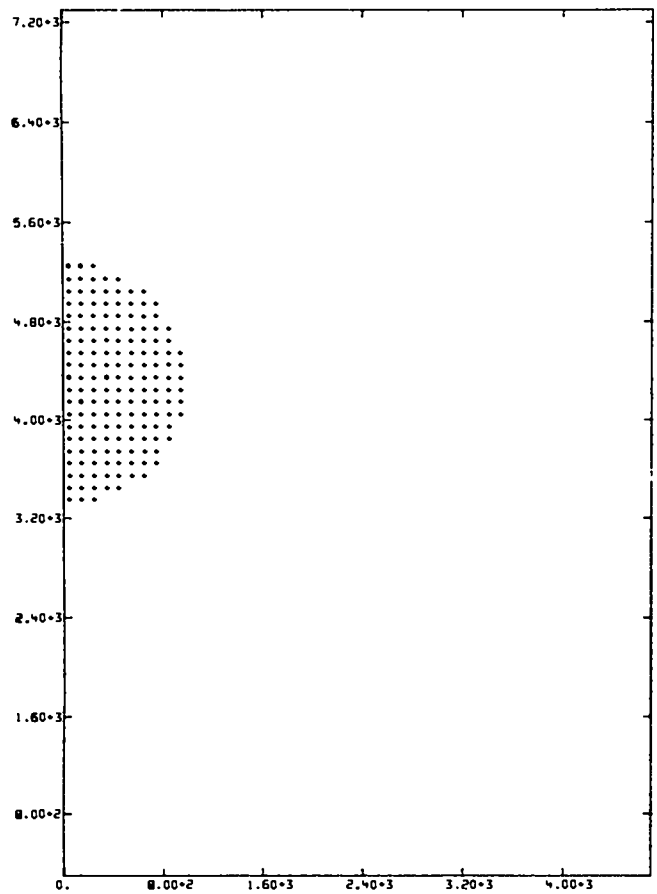
```
BALLOON - 30 X 45 - CONC. CMGS., PARTICLE TURB. DIFF., FIX IN TPBCOR
THERE WILL BE TURBULENCE
QLEVEL= 2.00000E-02
  TUQ1= 5.00000E+00
  TUS1= 1.00000E+03
   NCO=    0
    TO= 5.00000E-01
TSTRTD= 1.00000E+00
HMAXEF= 2.00000E+00
RMINEF= 5.00000E-01
  IBAR=  30
  JBAR=  45
  IUNF=   5
  JUNF=  10
  JCEN=  15
    DR= 2.00000E+02
    DZ= 2.00000E+02
   CYL= 1.00000E+00
GRDVEL= 2.00000E+00
    AO= 1.00000E+01
   AOM= 1.00000E+00
    BO= 0.
   KX1= .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= 0.30000E-03
    DT= 1.00000E-03
 NCLST=    1
 TMFIN= 3.00000E+01
 PAPER=0
  FILM=1
   ANC= 2.00000E+01
 AOFAC= 2.00000E+01
   IST=  10
 DTOI1-101=  1.00000E-01  1.00000E+00  5.00000E+00  0.
             0.           0.           0.           0.
             0.           0.
 DTOCI1-101=  1.00000E+00  1.00000E+01  3.00000E+01  0.
             0.           0.           0.           0.
             0.           0.
P$PARTN DRPAR=100.,DZPAR=100.,YC=4300.,XO=1000.,YO=0.,XC=0. $
 DRPAR= 1.00000E+02  DZPAR= 1.00000E+02  XC= 0.
 YC= 4.30000E+03  XO= 1.00000E+03  YO= 0.
P$PARTN DRPAR=0.$
     150 PARTICLES GENERATED. WITH TOTAL MASS= 0.
P$FIRE FBFILE=.TRUE. $

FIREBALL ,NPUT WILL BE FROM FILE 3
```

MESHMKR FIREBALL INPUT

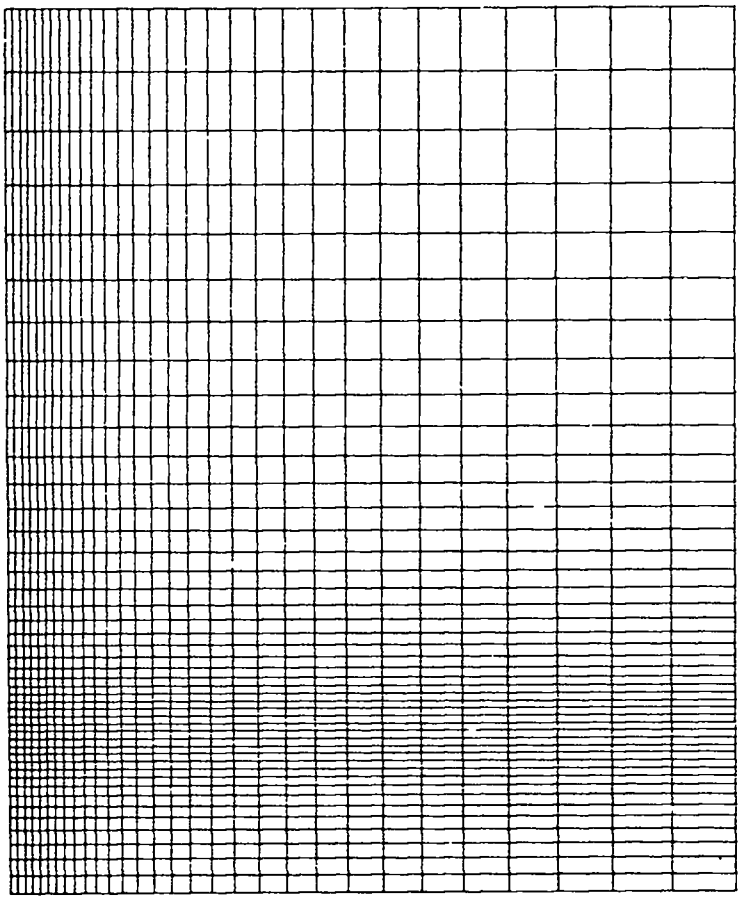
| U | E | RHO | X | P | PRHO |
|---|---|---|---|---|---|
| 3.7424474E+03 | 7.1150927E+10 | 1.9859535E-04 | 8.5614114E+00 | 1.5978265E+06 | 1.5978265E-06 |
| 7.4272784E-03 | 6.8662873E+10 | 2.0756046E-04 | 1.6904335E+01 | 1.6002742E+06 | 1.6002742E-06 |
| 1.1017913E+04 | 6.5330351E+10 | 2.2051980E-04 | 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 | 1.6222338E+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 | 1.5280966E-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 | 1.4500901E-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 |
| 6.2286853E+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 | 6.3154866E+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 | 1.2505825E-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 | 1.3713999E-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 | 6.4422838E+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 | 6.4580768E+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 |
| 4.5906287E+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 | 2.3940572E-04 | 4.5725828E+02 | 1.7236436E+06 | 1.7236436E-06 |
| 4.3975257E+04 | 6.4856745E+10 | 2.3976666E-04 | 4.6480005E+02 | 1.7330244E+06 | 1.7330244E-06 |
| 4.4891995E+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 | 2.4786693E-04 | 5.1711709E+02 | 1.8052738E+06 | 1.8052738E-06 |
| 4.1861138E+04 | 6.5158236E+10 | 2.4909695E-04 | 5.2455387E+02 | 1.8163692E+06 | 1.8163692E-06 |
| 4.1873625E+04 | 6.5221982E+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 | 1.8258586E+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 | 5.6899154E+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 | 2.6129194E-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 | 6.5622866E+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 | 6.5753264E+10 | 2.6842126E-04 | 6.5665120E+02 | 1.9863244E+06 | 1.9863245E-06 |
| 3.7523874E+04 | 6.5771740E+10 | 2.7008194E-04 | 6.6385539E+02 | 2.0000209E+06 | 2.0000209E-06 |
| 3.7602140E+04 | 6.5755132E+10 | 2.7014826E-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 | 6.5872889E+10 | 2.7595965E-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 | 7.4979994E+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 | 5.5309300E+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 | 2.1276444E+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 | 2.1440802E+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 | 7.9320512E+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 |
| 3.4878806E+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+06 |
| 3.4666535E+04 | 3.6728596E+09 | 1.5182728E-03 | 8.1415708E+02 | 2.2072736E+06 | 2.2073738E+06 |
| 3.4676816E+04 | 3.6431271E+09 | 1.5351190E-03 | 8.1605383E+02 | 2.2146754E+06 | 2.2146754E+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 | 2.2565508E+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 | 2.2884185E+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 | 1.8372907E-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 | 2.4439054E+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 |
| 3.3907938E+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 | 3.1276544E+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 |
| 3.4442366E+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 | 2.6515859E+06 | 2.6515859E+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 | 2.7493946E+06 | 2.7493946E+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 | 2.8492251E+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 |
| 3.7392726E+04 | 3.1372589E+09 | 2.4946243E-03 | 9.4925066E+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 |
| 6.5841253E+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 | 1.9680586E+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 | 8.1663626E+05 | 8.1883625E+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 | 8.1494952E+05 |

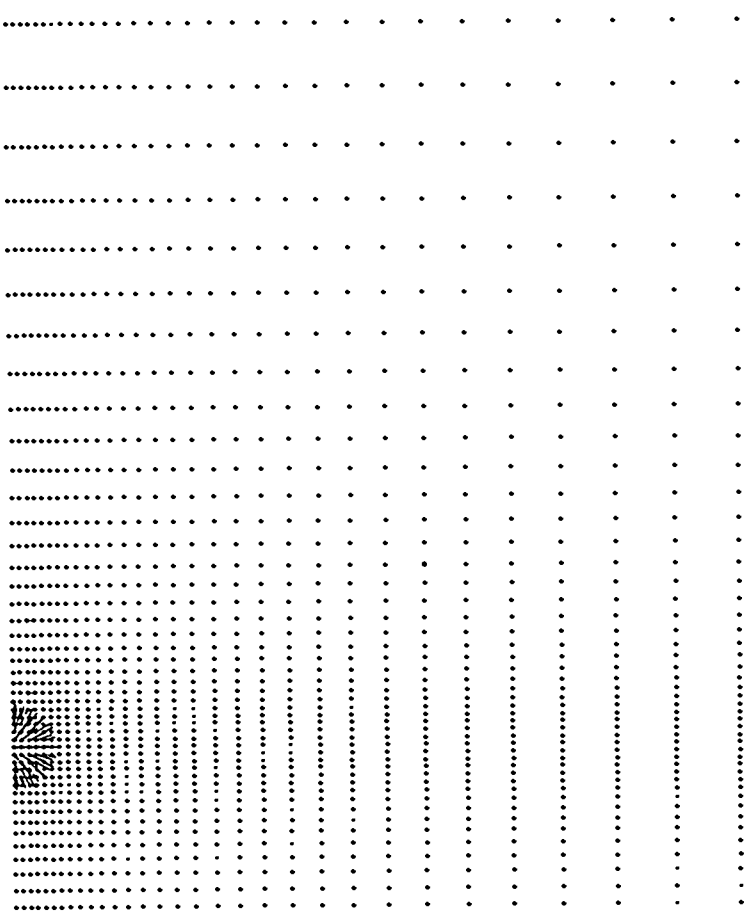
```
ALL FIREBALL INPUT DATA READ

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 BOUNDARY CELL
CELL     5   19 HAS A RELATIVE VOLUME ERROR 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
                0000000000000000000     00000000000000000000
                VMAX =  4.86465E+04 AT VERTEX     2   17
```



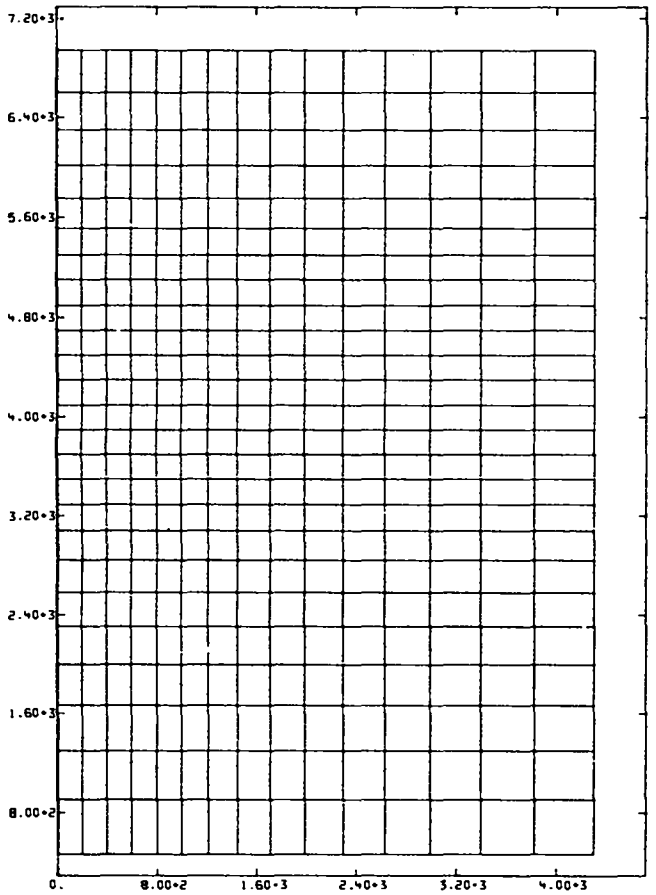

```
PARTICLES
        PXR= 1.92?9E+04 PYR=-1.38538E-03 PYT= 2.3579?E-04
   YAQUI-LT5S BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRSCOR            T= 8.30000E-03 CYCLE=    0
```



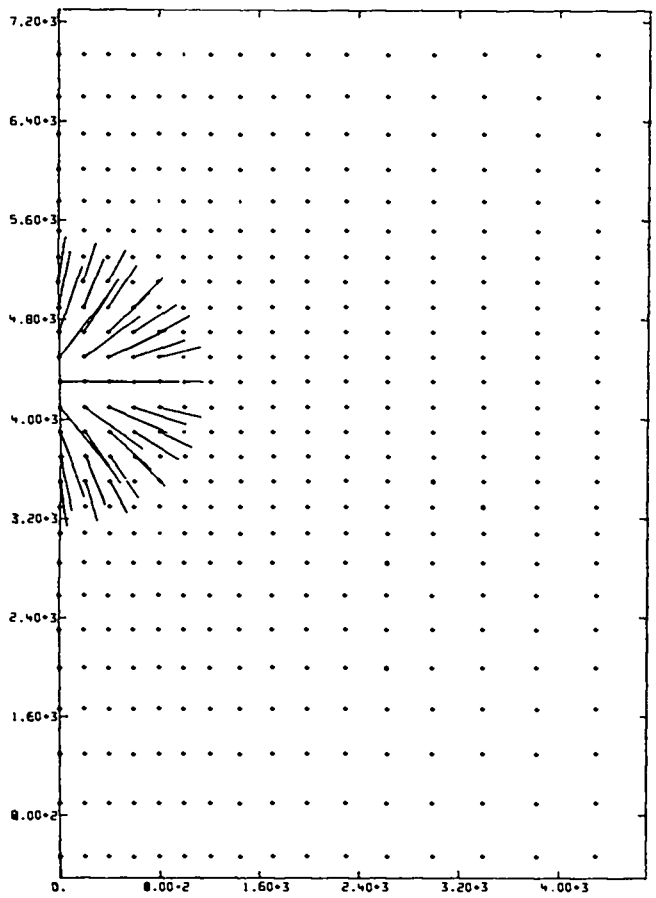
ALL ZONES
DRMIN= 2.00000E+02 DRMAX= 1.69944E+03 DZMIN= 2.00000E+02
DZMAX= 1.69944E+03 XR= 1.92794E+04 YB=-1.38538E+03 YT= 2.35794E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E+03 CYCLE= 0



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= 0.30000E-03 CYCLE= 0

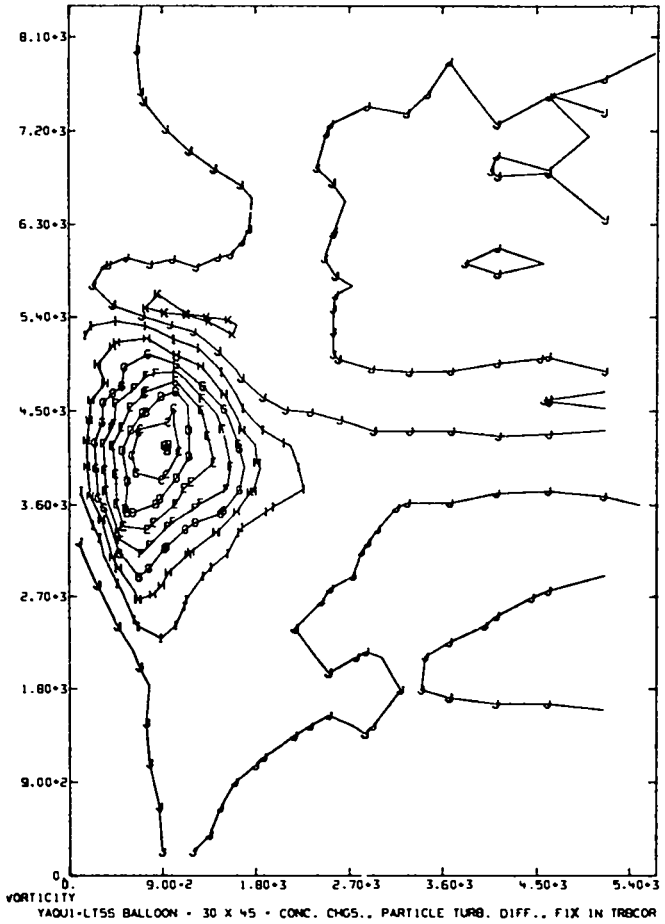



ZONES IN THE FIREBALL REGION
DRMIN= 2.00000E+02 DRMAX= 1.69944E+03 DZMIN= 2.00000E+02
DZMAX= 1.69944E+03 XR= 1.92794E+04 YB=-1.38538E+03 YT= 2.35794E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E-03 CYCLE= 3



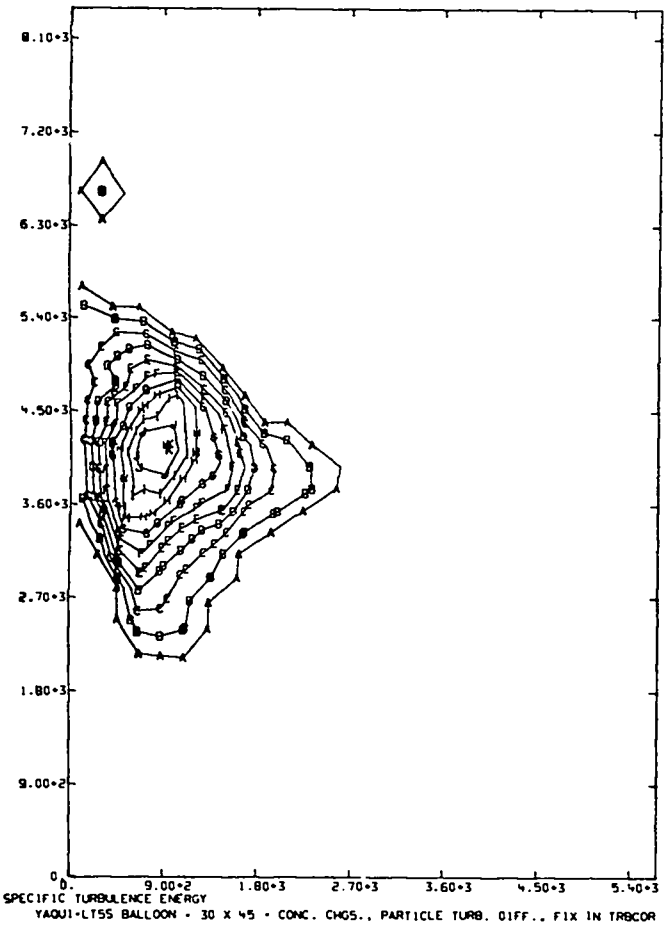

FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 4.86465E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 8.30000E+03 CYCLE= 0

B. Turbulence Seeding Conditions




| | |
|---|---|
| A | +2.250E+00 |
| B | +2.000E+00 |
| C | +1.750E+00 |
| D | +1.500E+00 |
| E | +1.250E+00 |
| F | +1.000E+00 |
| G | +7.500E-01 |
| H | +5.000E-01 |
| I | +2.500E-01 |
| J | 0. |
| K | 2.500E-01 |
| L | 5.000E-01 |
| QMN | +2.026E+00 |
| QMX | 3.103E-01 |

T= 5.13512E-01 CYCLE= 81



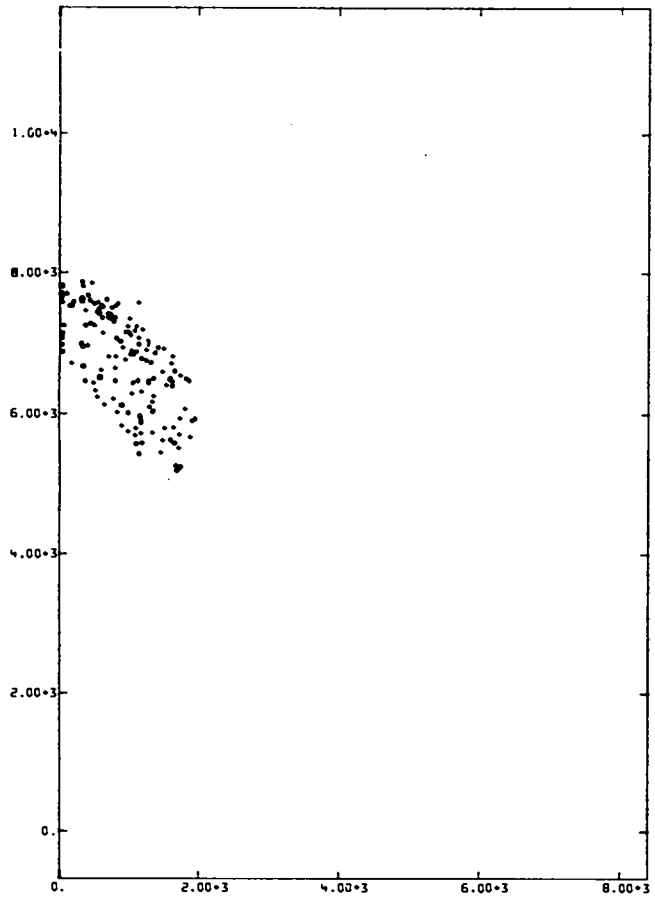

SPECIFIC TURBULENCE ENERGY

YAQUI-LT5S BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

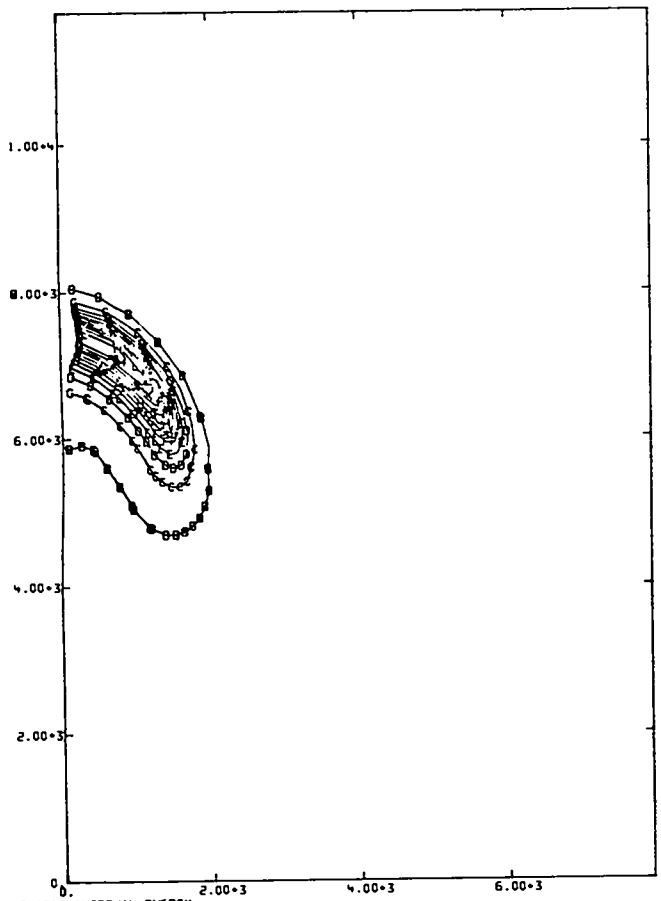
| | |
|---|---|
| A | 0. |
| B | 1.000E+00 |
| C | 2.000E+00 |
| D | 3.000E+00 |
| E | 4.000E+00 |
| F | 5.000E+00 |
| G | 6.000E+00 |
| H | 7.000E+00 |
| I | 8.000E+00 |
| J | 9.000E+00 |
| K | 1.000E+01 |
| L | 1.100E+01 |
| QMN | 0. |
| QMX | 1.013E-01 |

T= 5.13512E-01 CYCLE= 81

C. Turbulence Equilibrium Conditions



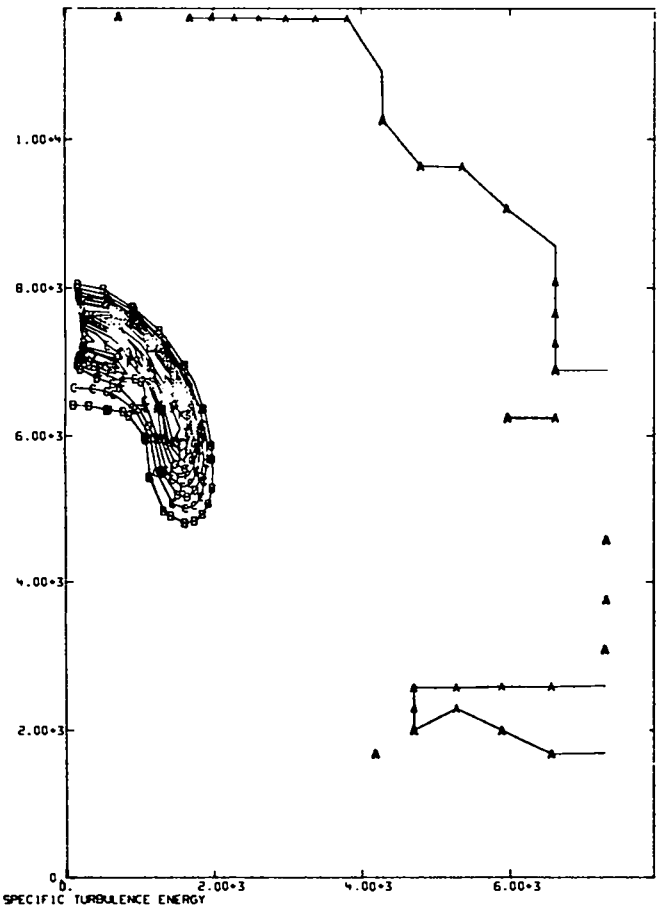

PARTICLES
PXR= 1.92794E+04 PYB= 0. PYT= 2.35794E+04
YAQUI-LT55 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 3.00000E+00 CYCLE= 64




| | |
|---|---|
| A | 1.074E+09 |
| B | 2.147E+09 |
| C | 3.221E+09 |
| D | 4.295E+09 |
| E | 5.369E+09 |
| F | 6.442E+09 |
| G | 7.516E+09 |
| H | 8.590E+09 |
| I | 9.664E+09 |
| J | 1.074E+10 |
| K | 1.181E+10 |
| L | 1.288E+10 |
| M | 1.396E+10 |
| N | 1.503E+10 |
| O | 1.611E+10 |
| QMN | 1.958E+05 |
| QMX | 1.548E+10 |

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

T= 3.00000E+00 CYCLE= 464



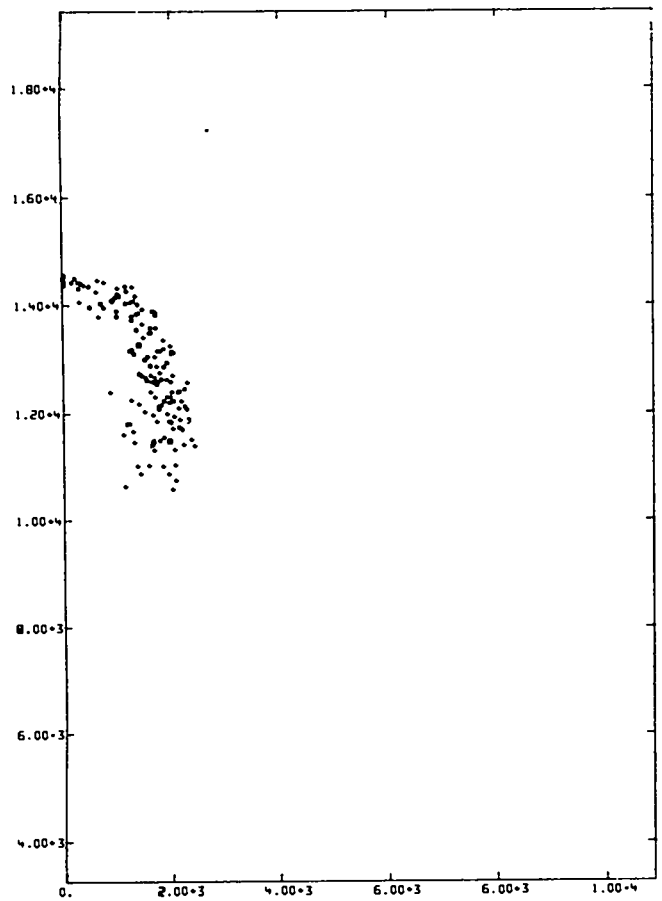

| | |
|---|---|
| A | 0. |
| B | 1.638E+04 |
| C | 3.277E+04 |
| D | 4.915E+04 |
| E | 6.554E+04 |
| F | 8.192E+04 |
| G | 9.830E+04 |
| H | 1.147E+05 |
| I | 1.311E+05 |
| J | 1.475E+05 |
| K | 1.638E+05 |
| L | 1.802E+05 |
| M | 1.966E+05 |
| N | 2.130E+05 |
| QMN | 0. |
| QMX | 2.034E+05 |

SPECIFIC TURBULENCE ENERGY
YAQUI-LT55 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

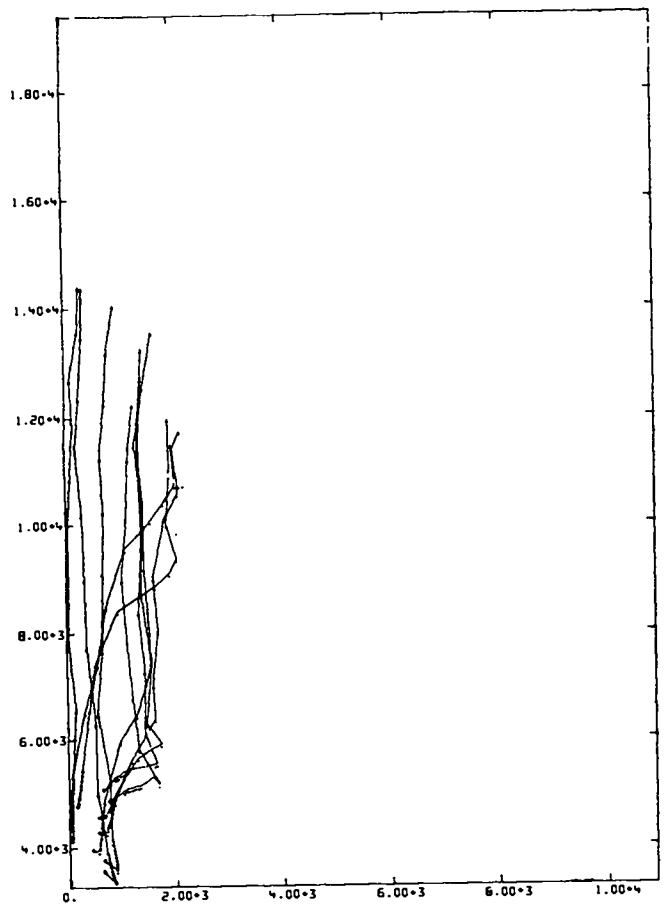
T= 3.00000E+00 CYCLE= 464

## 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= 1.3410E-03. CP= 2.97676E+00
                    GRIDS= 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 =  6.89867+8E+16
                    TOTAL RADIAL MOMENTUM =  4.6797458E+09
                    TOTAL AXIAL MOMENTUM = -2.6862275E+10
                    17104271067616720223     17147033011630461263
                    VMAX =  1.12413E+03 AT VERTEX     5   16
```

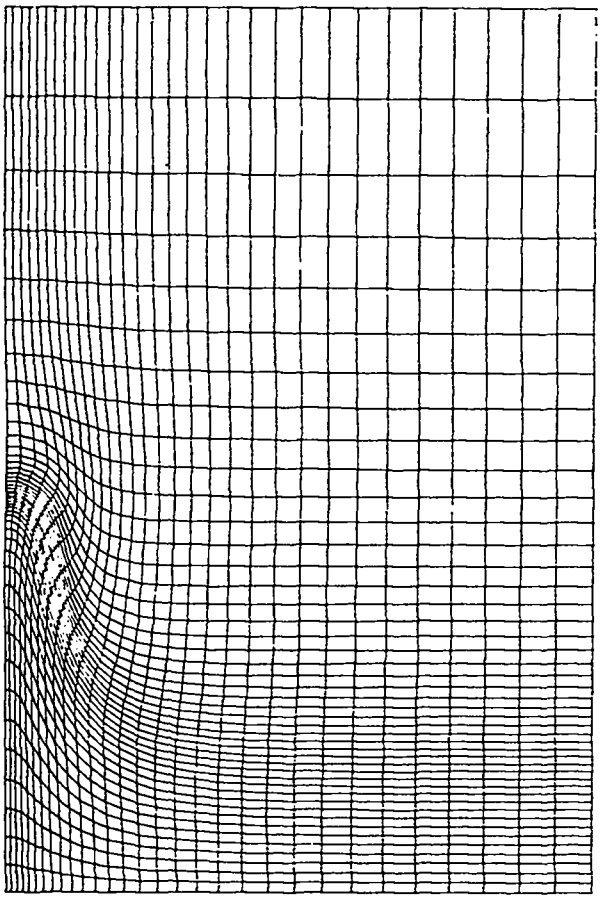



```
PARTICLES
          PXR= 1.92794E+04 PYB= 0.          PYT= 2.92102E+04
    YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TPBCOR          T= 1.00000E+01 CYCLE= 2369
```

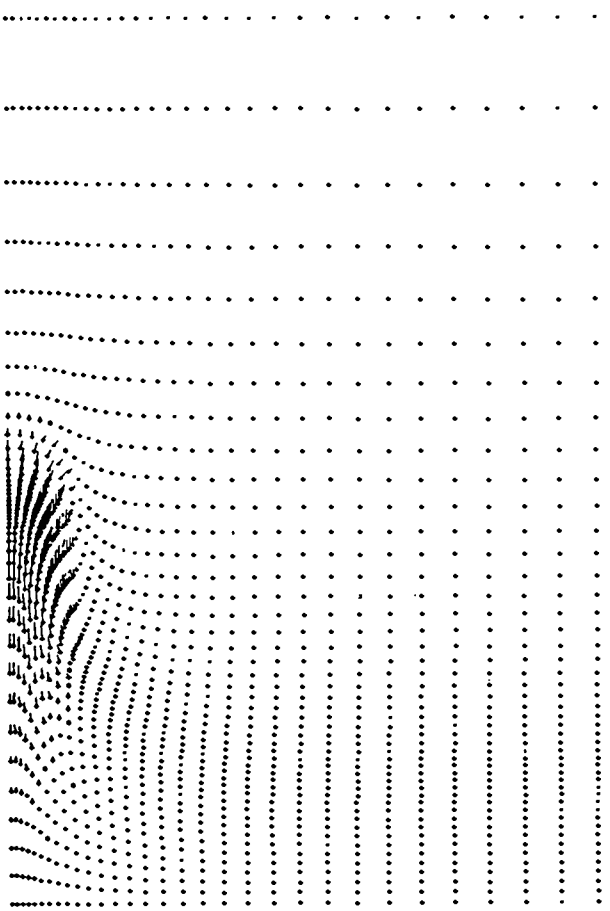



TIME-DEPENDENT PARTICLES
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN T98COR

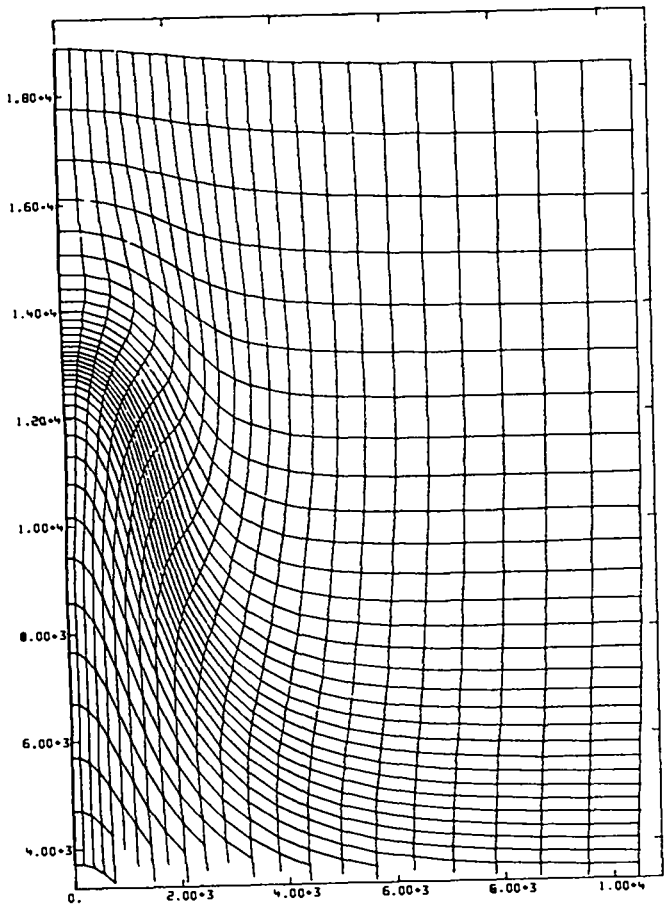
T= 1.00030E+01 CYCLE= 2099



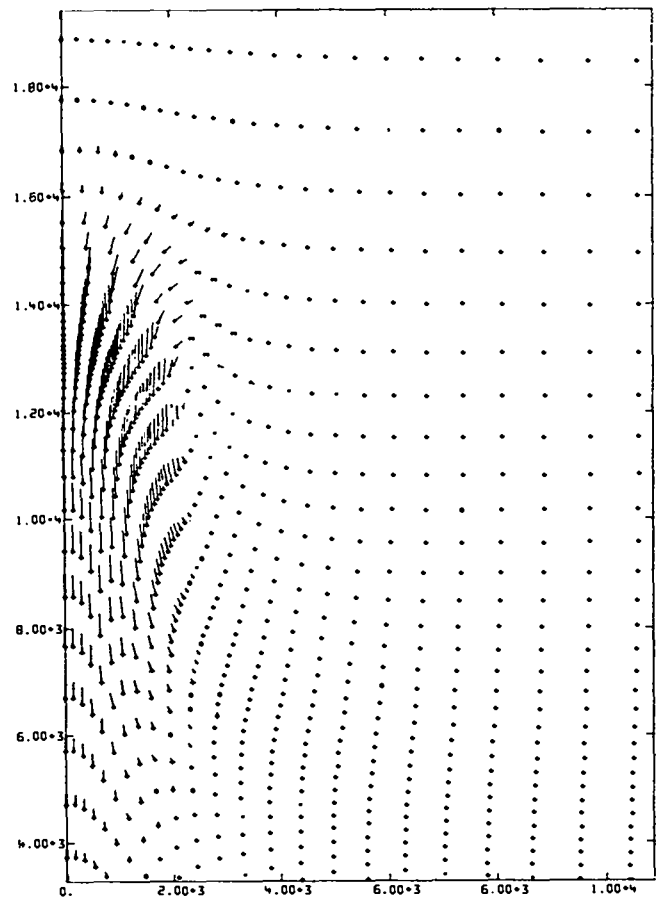
ALL ZONES
DRMIN= 1.59383E+02 DRMAX= 1.18653E+03 DZMIN= 8.03520E+01
DZMAX= 2.99070E+03 XR= 1.92794E+04 YB= 0. YT= 2.92102E+04
YAQUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2299



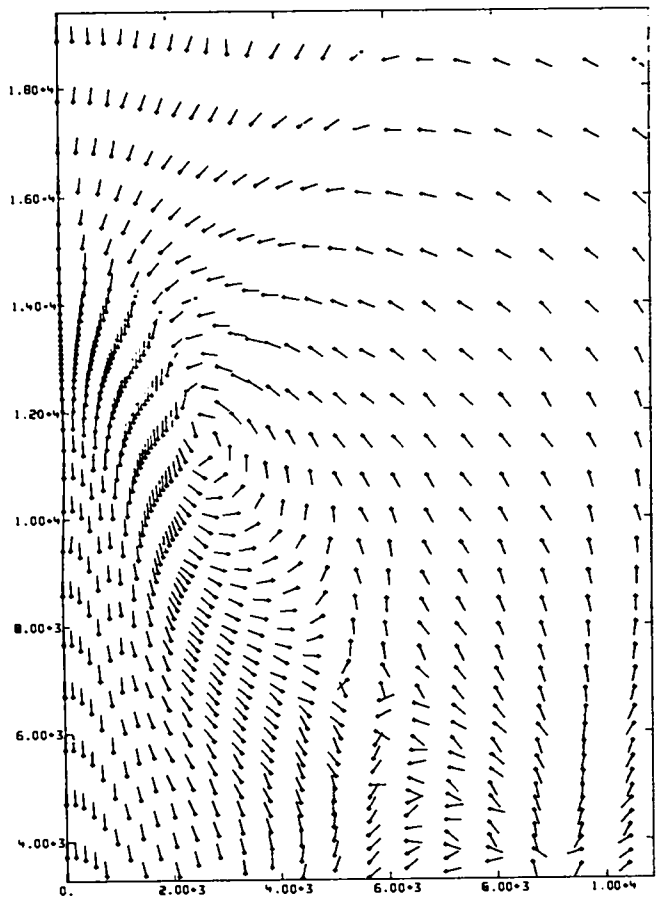
FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 1.1241E+03
YAQUI-LESS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 1.300205+01 CYCLE= 2299



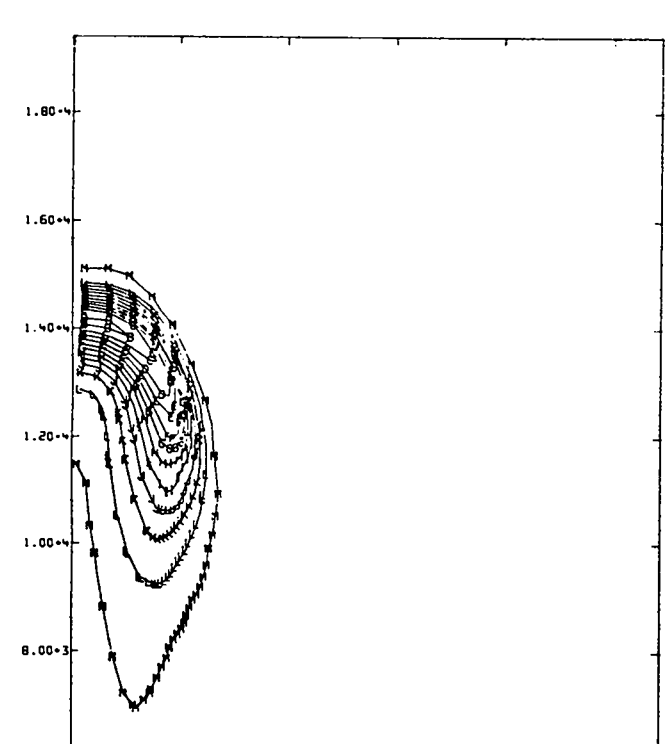

ZONES IN THE FIREBALL REGION
DRMIN= 1.59383E+02 DRMAX= 1.18653E+03 DZMIN= 8.03520E+01
DZMAX= 2.990 0E+03 XR= 1.9279 E+04 YB= 0. YT= 2.92102E+04
YAQUI-LT55 BALLOON - 30 X 46 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN TRBCOR T= 1.00000E+01 CYCLE= 2299




FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 1.1241E+03
YAQUI-LT5S BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 1.0003E+01 CYCLE= 2289



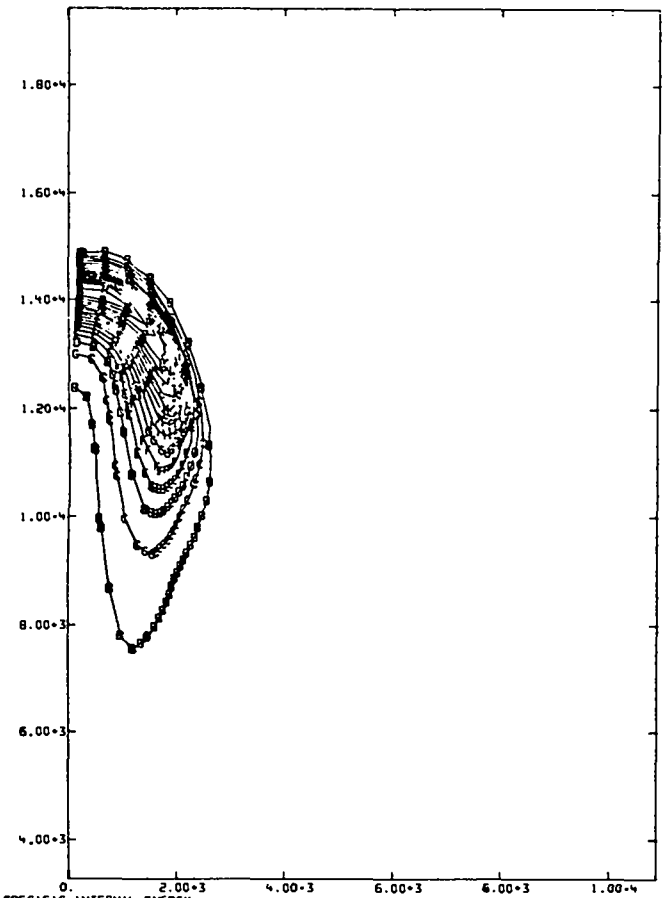

UNSCALED FLUID VELOCITY VECTORS
VMAX= 1.12413E-03
YAQUI-LT55 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR T= 1.00000E-01 CYCLE= 2289




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

| | |
|---|---|
| A | 6.13E-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 |

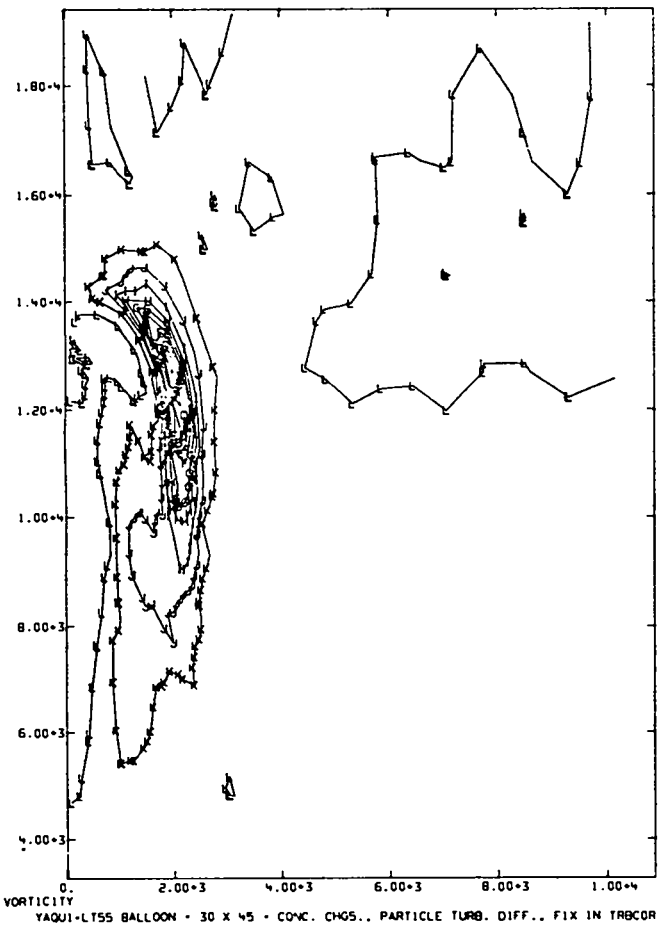
T= 1.00000E+01 CYCLE= 2283




| | |
|---|---|
| A | 1.946E+09 |
| B | 2.013E+09 |
| C | 2.080E+09 |
| D | 2.147E+09 |
| E | 2.215E+09 |
| F | 2.282E+09 |
| G | 2.349E+09 |
| H | 2.416E+09 |
| I | 2.483E+09 |
| J | 2.550E+09 |
| K | 2.617E+09 |
| L | 2.684E+09 |
| M | 2.751E+09 |
| N | 2.819E+09 |
| O | 2.886E+09 |
| P | 2.953E+09 |
| Q | 3.020E+09 |
| R | 3.087E+09 |
| S | 3.154E+09 |
| QMN | 1.951E+09 |
| QMX | 3.149E+09 |

SPECIFIC INTERNAL ENERGY
YAQU1-LT55 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

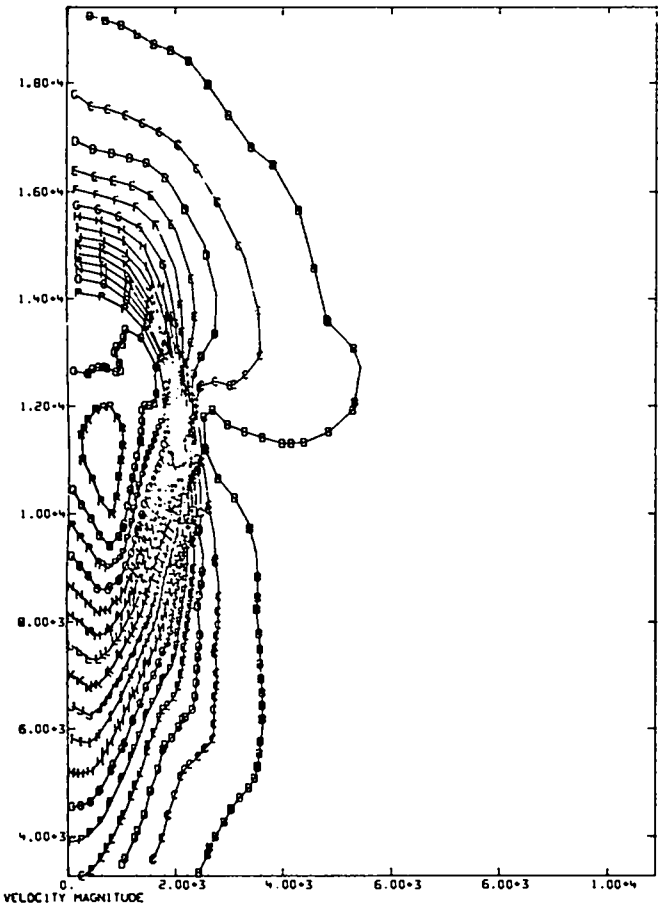
T= 1.000000E+01 CYCLE= 2289




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
QMN +2.576E-00
QMX 2.413E-01

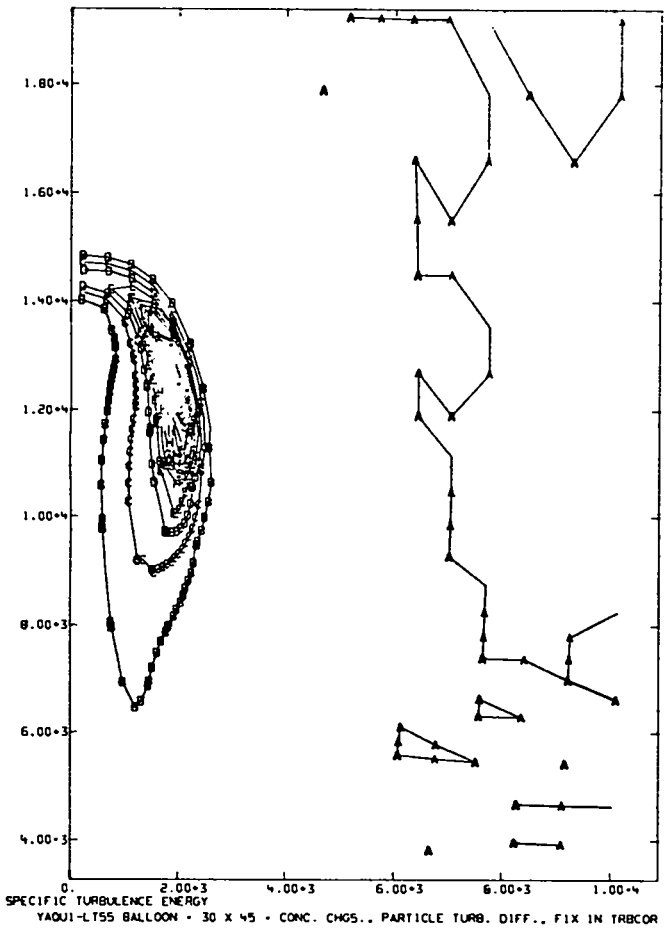
YAQUI-LT55 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

T= 1.00000E-01 CYCLE= 2299



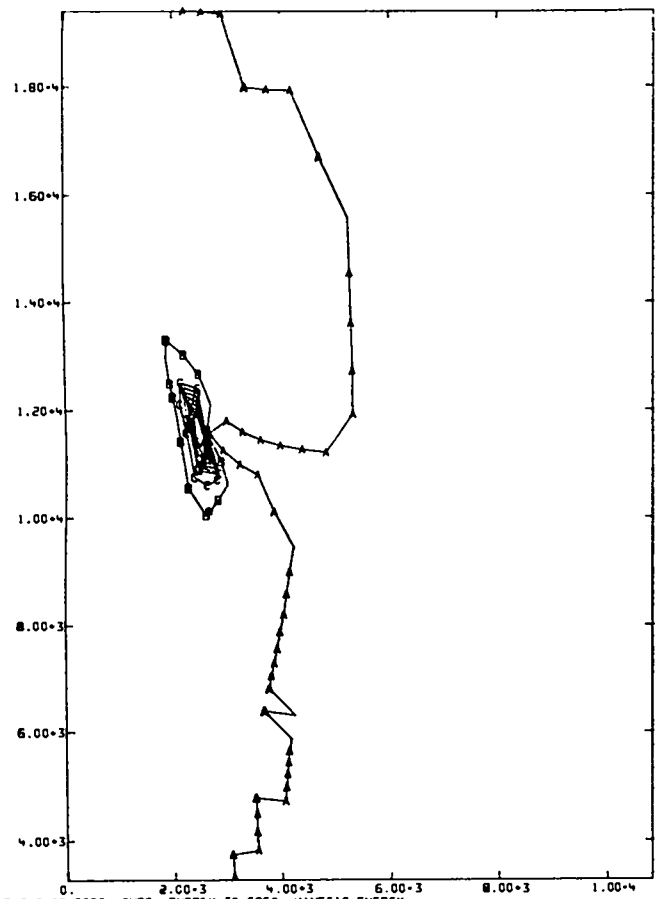

| | |
|---|---|
| A | 0. |
| B | 6.400E+01 |
| C | 1.280E+02 |
| D | 1.920E+02 |
| E | 2.560E+02 |
| F | 3.200E+02 |
| G | 3.840E+02 |
| H | 4.480E+02 |
| I | 5.120E+02 |
| J | 5.760E+02 |
| K | 6.400E+02 |
| L | 7.040E+02 |
| M | 7.680E+02 |
| N | 8.320E+02 |
| O | 8.960E+02 |
| P | 9.600E+02 |
| Q | 1.024E+03 |
| R | 1.088E+03 |
| S | 1.152E+03 |
| QMN | 1.076E+00 |
| QMX | 1.122E+03 |

T= 1.000000E+01 CYCLE= 2289




| | |
|---|---|
| A | 0. |
| B | 2.048E+03 |
| C | 4.096E+03 |
| D | 6.144E+03 |
| E | 8.192E+03 |
| F | 1.024E+04 |
| G | 1.229E+04 |
| H | 1.434E+04 |
| I | 1.638E+04 |
| J | 1.843E+04 |
| K | 2.048E+04 |
| L | 2.253E+04 |
| M | 2.458E+04 |
| N | 2.662E+04 |
| O | 2.867E+04 |
| P | 3.072E+04 |
| QMN | 0. |
| QMX | 3.009E+04 |

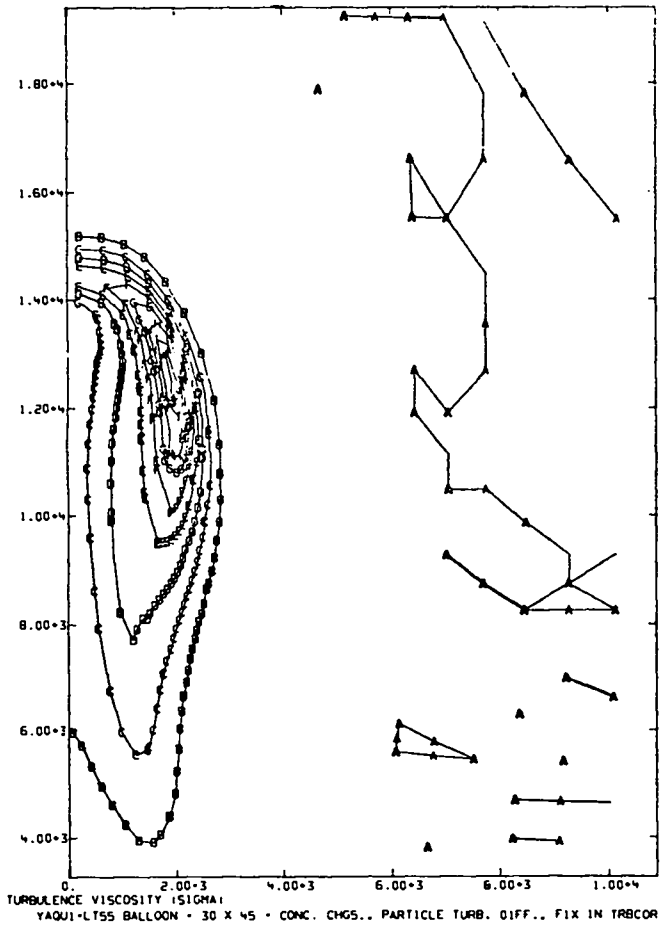
T= 1.00000E+01 CYCLE= 2289




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

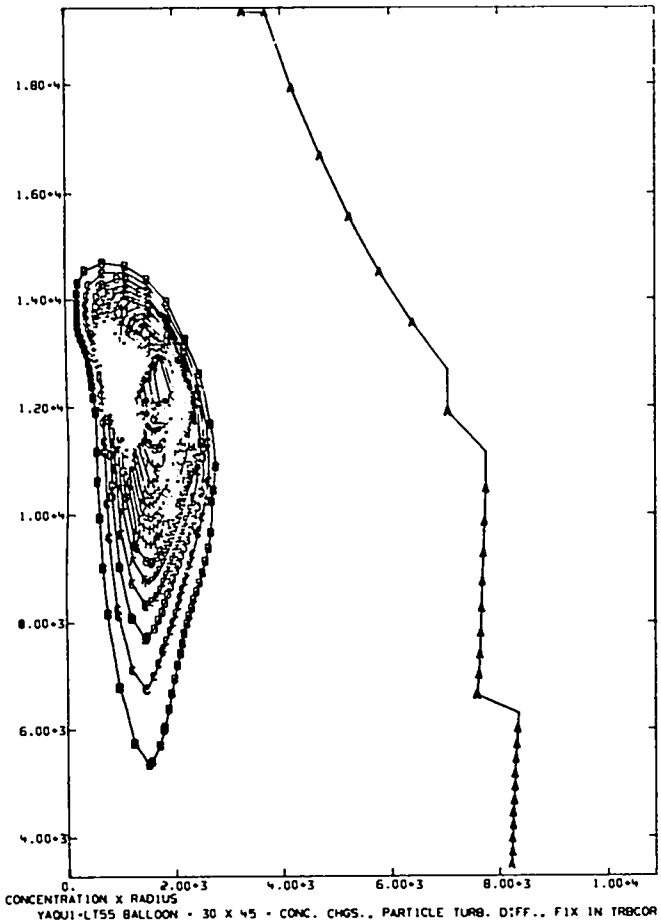
| | |
|---|---|
| A | 0. |
| B | 1.250E-01 |
| C | 2.500E-01 |
| D | 3.750E-01 |
| E | 5.000E-01 |
| F | 6.250E-01 |
| G | 7.500E-01 |
| H | 8.750E-01 |
| I | 1.000E+00 |
| J | 1.125E+00 |
| K | 1.250E+00 |
| L | 1.375E+00 |
| M | 1.500E+00 |
| QMN | 0. |
| QMX | 1.463E+00 |

T= 1.00000E+01 CYCLE= 2289




| | |
|---|---|
| A | 0. |
| B | 5.120E+02 |
| C | 1.024E+03 |
| D | 1.536E+03 |
| E | 2.048E+03 |
| F | 2.560E+03 |
| G | 3.072E+03 |
| H | 3.584E+03 |
| I | 4.096E+03 |
| J | 4.608E+03 |
| K | 5.120E+03 |
| QMN | 0. |
| QMX | 4.936E+03 |

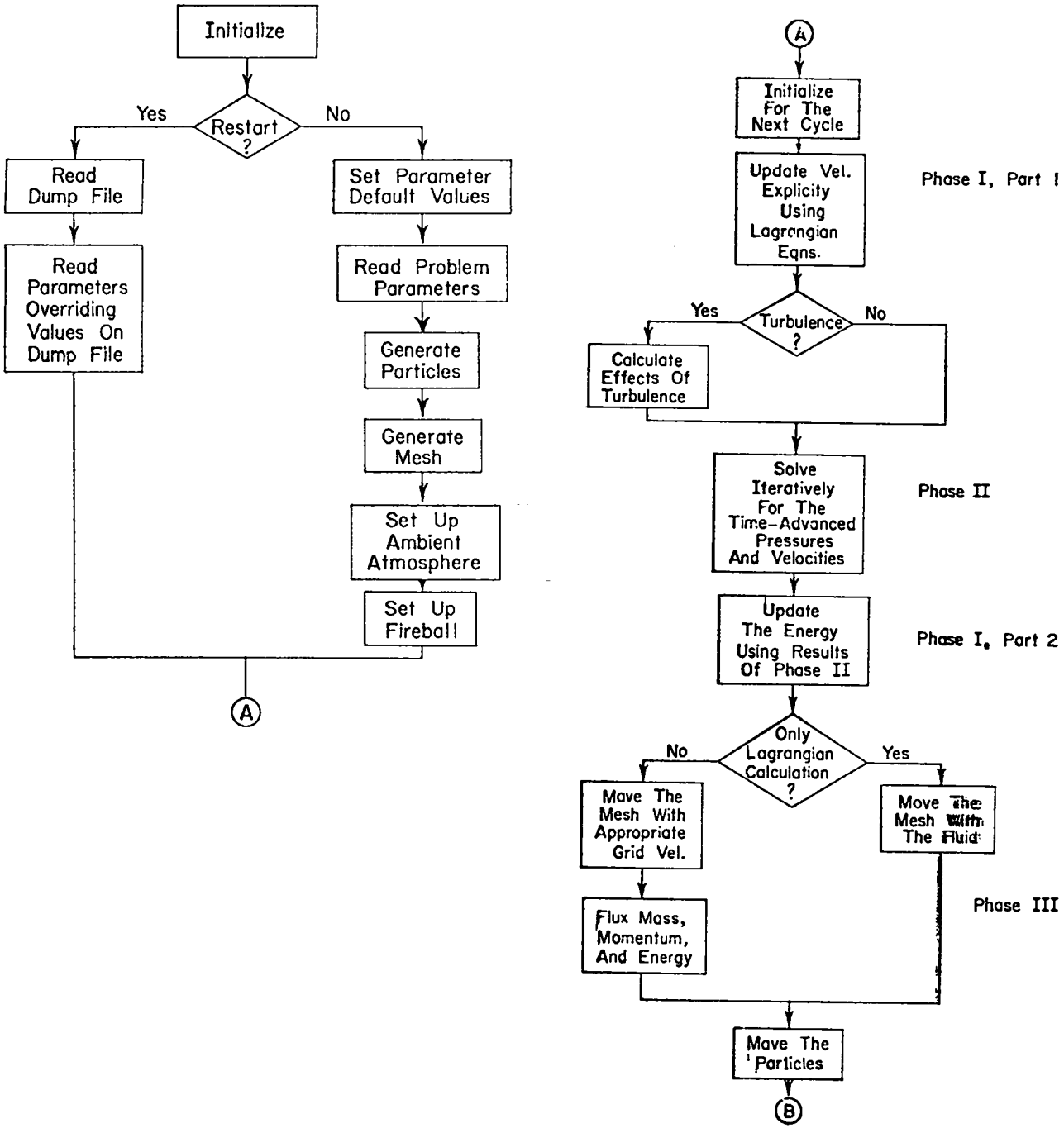
T= 1.00000E+01 CYCLE= 2289

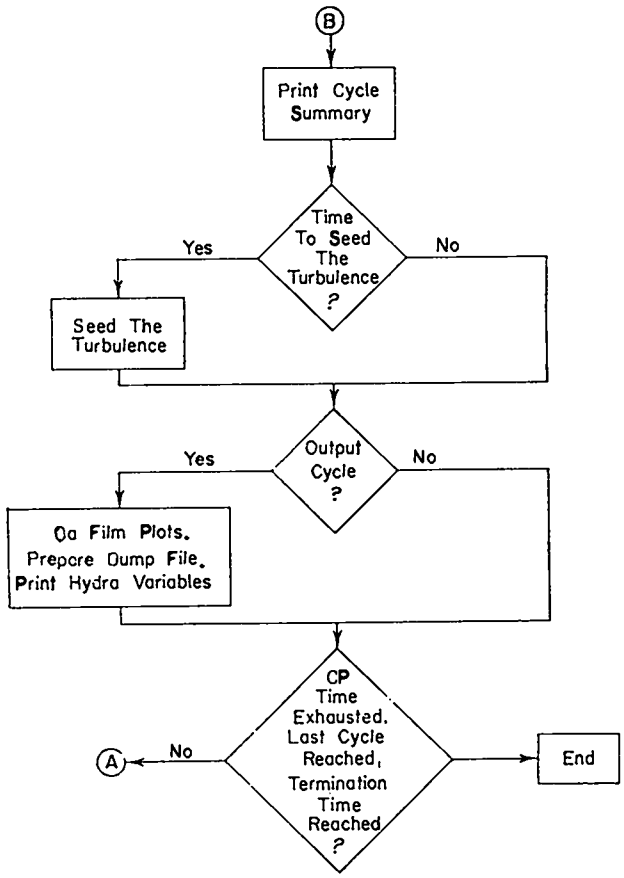



| | |
|---|---|
| A | 0. |
| B | 8.000E-00 |
| C | 1.600E+01 |
| D | 2.400E+01 |
| E | 3.200E+01 |
| F | 4.000E+01 |
| G | 4.800E+01 |
| H | 5.600E+01 |
| I | 6.400E+01 |
| J | 7.200E+01 |
| K | 8.000E+01 |
| L | 8.800E+01 |
| M | 9.600E+01 |
| N | 1.040E+02 |
| O | 1.120E+02 |
| P | 1.200E+02 |
| Q | 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
Qa Film Plots. Prepare Dump File. Print Hydra Variables
CP Time Exhausted, Last Cycle Reached, Termination Time Reached ?
A
No
End

```
      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 ECRO(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)
```

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

---

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

---

```
 1            INTEGER GETIT
 2            FUNCTION GETIT(I)
 3      C
 4      C     LTSS ROUTINE TO ALLOW FETCHING CONTENTS OF A WORD GIVEN ITS
 5      C         ABSOLUTE ADDRESS
 6      C
 7      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
 8      C
 9            ABSOLUTE IADDR(1)
10            DIMENSION IADDR(1)
11            GETIT=IADDR(I)
12            RETURN
13            END
```

---

```
 1            SUBROUTINE GETJOB(JOBID)
 2      C
```

```
      C     LTSS ROUTINE FOR RETURNING THE JOB ID
      C
      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
      C
            JOBID=1ØHYAQUI-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 ØØTIM(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/GØBCOM/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 COC 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,NWROS,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(NWROS.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/.0174532925199433/
      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(IA00)=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 IOENT80 -
      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 IOENT80(12,FLMBUF,IFLMSZ,1000000,105,IOUM)
            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 1-3,1975
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=7777777777778
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=77777777778
      IFILEP=IFILEP-10*(IT-1)
   10 CONTINUE
C
```

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

---

```
 1              SUBROUTINE TRAP(IARG)
 2      C
 3      C       LTSS ROUTINE TO HANDLE ERROR INTERCEPTION
 4      C
 5      C       WRITTEN BY J.L.NORTON,LASL T-3,1975
 6      C
 7              COMMON/SYSC5/RESTRT,FILM,PAPER,IPD,IFO
 8              COMMON/ORERR/IERR(121)
 9              COMMON/GOBCOM/IG(1)
10              ABSOLUTE TWO(2)
11              INTEGER TWO,SHIFT
12              LOGICAL FILM
13              COMMON/IEOMP/IDUMP(16)
14      C
15      C       SET UP ABSOLUTE LOCATION 2 WITH A JUMP TO STATEMENT IRETA
16      C           AND SET THE EXCHANGE PACKAGE DUMP AREA TO IDUMP
17      C
18              TWO=2.SHL.54.UN.(.LOC.IRETA.SHL.30).UN.(.LOC.IDUMP)
19      C
20      C       SET UP ORDERLIB ERROR INTERCEPTION
21      C
22              DO 10 I=1,99
23           10 CALL CONTROL(I,IORDL,IERO)
24              RETURN
25      C
26      C       ORDERLIB ERROR HAS OCCURRED, LET THE USER KNOW.
27      C
28      IORDL CONTINUE
29              DO 20 IPX=6,IFD,6
30           20 WRITE(IPX,50)(IERR(I),I=117,119)
31              IDUMP(1)=SHIFT(IERR(118),36)
32              CALL POMPPK
33              IFLS=IG(15)
34              IF(FILM) GO TO 30
35              GO TO 40
36      C
37      C       PROGRAM HAS ABORTED
38      C
39      IRETA CONTINUE
40      C       DUMP THE EXCHANGE PACKAGE AND SMALL CORE ON EITHER SIDE
41      C           OF THE LOCATION OF THE ERROR
42      C
43              CALL PABORT
44      C
45      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 OMP(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,777777000000000000008),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 IMOISP(5)
       DATA IMOISP/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,0,0,0,0,0,0,10000,0,0)
C
C      ZERO OUT THE LAST TWO CHARACTERS OF THE TAPE LABEL
C
       IMOISP(5)=LABEL.AND.(.NOT.77778)
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,IMOISP,0,0,10000,0)
C
C      INCREMENT THE STAGE COUNT
C
       ICNT=ICNT+1
```

```
41      C
42      C     SEE IF THERE WERE ANY PARITY ERRORS
43      C
44            CALL PARITY(IF,ICHECK)
45            IF(ICHECK.NE.0) GO TO 30
46      C
47      C     NO. PRINT MESSAGE AND RETURN.
48      C
49            DO 20 IPX=IPD,IFD,6
50         20 WRITE(IPX,50) LABEL,IF,IF
51            RETURN
52         30 CONTINUE
53      C
54      C     YES. SEE IF THIS WAS THE FOURTH STAGE.
55      C
56            IF(ICNT.LT.4) GO TO 40
57      C
58      C     YES. SET THE ERROR FLAG AND RETURN.
59      C
60            IFLAG=1
61            RETURN
62         40 CONTINUE
63      C
64      C     NO. RELEASE THE FILE AND TRY AGAIN.
65      C
66            CALL AFSREL(IF,0,0,0)
67            GO TO 10
68      C
69         50 FORMAT(1H0,5HTAPE ,A8,35H HAS BEEN SUCCESSFULLY STAGED INTO ,
70           1 8HFILESET ,I2,2H (,A8,1H))
71            END
```

```
 1            SUBROUTINE SYINIT
 2      C
 3      C     CROS/7600 ROUTINE TO PERFORM YAQUI SYSTEM INITIALIZATION
 4      C
 5      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
 6      C
 7      C     CALL SETQ(4LKOPX,0)
 8            CALL OPEN(3LOUT,0,0,0,0,0,0,10000,0,0)
 9            CALL OPEN(4LFILM,0,0,0,0,0,0,10000,0,0)
10            CALL OPEN(7,0,0,0,0,0,0,10000,0,0)
11            CALL OPEN(3,0,0,0,0,0,0,10000,0,0)
12            CALL MEMREQ(400000,1)
13            RETURN
14            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,IPO,IFO
      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
```

```
11            END
```

---

```
 1            SUBROUTINE GETLCM(IFLLCM)
 2      C
 3      C     KRONOS/6600 ROUTINE TO RETURN THE AMOUNT OF LCM AVAILABLE TO
 4      C         THE USER
 5      C
 6      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
 7      C
 8            IFLLCM=500000
 9            RETURN
10            END
```

---

```
 1            SUBROUTINE GETTPE(IDUM)
 2      C
 3      C     KRONOS/6600 DUMMY ROUTINE
 4      C
 5      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
 6      C
 7            RETURN
 8            END
```

---

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

---

```
 1            SUBROUTINE SYINIT
 2      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 OMP,R,0,0
            SYSTEM OMP,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 INTEPSECTION
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,NWROS,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,ITARP,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 OP
      OR=I.OR.J
      RETURN
      END
```

---

```
      FUNCTION ORR(I,J)
C
C     KRONOS/6600 AND CROS/7600 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/6600 AND CROS/7600 ROUTINE TO RETURN RANDOM NOS. WITH
C         UNIFORM DISTRIBUTION ON THE INTERVAL (0.,1.)
C
C     WRITTEN BY J.L.NORTON,LASL T-3,1975
C
      RNUMF=RANF(X)
      RETURN
      END
```

---

```
      SUBROUTINE SCMIFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
C
C     KRONOS/6600 AND CROS/7600 ROUTINE TO READ OR WRITE SCM FROM
C         OR TO A DISK FILE
C
C     SEE LCDIFF 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    42/4HSHIFT,18/2
SHIFT DATA   0
      SA1    B2
      SA2    B1
      ZR     X1,ZERO
      PL     X1,LEFT
      NX3    -X1
      SB3    X3
      AX6    B3,X2
      JP     SHIFT
LEFT  SB3    X1
      LX6    B3,X2
      JP     SHIFT
ZERO  BX6    X2
```

```
29                JP      SHIFT
30                END
```

---

```
 1                IDENT   STORIT
 2
 3      *
 4      *       CROS/7600 AND KRONOS/6600 SUBROUTINE STORIT(IWORD,IADD) WHICH
 5      *           PUTS IWORD INTO ABSOLUTE SCM LOCATION IADD
 6      *
 7                ENTRY   STORIT
 8                VFD     42/0HSTORIT,18/2
 9       STORIT   DATA    0
10                SA1     B1              PICK UP IWORD IN X1
11                SA2     B2              PICK UP IADD IN X2
12                BX6     X1              STORE IWORD INTO IADD
13                SA6     X2
14                JP      STORIT
15                END
```

---

```
 1              SUBROUTINE TTYTST(IFLAG)
 2      C
 3      C       KRONOS/6600 AND CROS/7600 ROUTINE TO DUMMY UP CAPABILITY OF
 4      C           LTSS TO MODIFY THE DROPFILE
 5      C
 6      C       WRITTEN BY J.L.NORTON,LASL T-3,1975
 7      C
 8              IFLAG=0
 9              RETURN
10              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,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,TPS,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,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,IOTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASSV,WMAXEF,RMINEF,TSTRTQ
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPO,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.)
```

```
118             FN=EXP(-E/25.5)*W8
119         140 E3=(E-160.)/6.
120             BETA=BETA*(1.-W8)+.045*W8
121             IF(E3+5.)150,160,160
122         150 FE=0.
123             GO TO 170
124         160 FE=1./(EXP(-E3)+1.)
125         170 RHOFAC=RHOT**BETA
126             GMONE=(.161+.255*FO+.280*FON+.137*FN+.050*FE)*RHOFAC
127       C
128       C     IF IFLAG IS ZERO,ONLY AIR IS DESIRED AND WE ARE ALL DONE
129       C
130             IF(IFLAG.EQ.0) RETURN
131       C
132       C     CALCULATE THE GAMMA-1. LINEAR COMBINATION
133       C
134             GMONE=CONCJ*GMM+(1.-CONCJ)*GMONE
135             RETURN
136             END
```

```
1             SUBROUTINE BC(IFLAG)
2       C
3       C     ROUTINE TO SET RIGID WALL BOUNDARY CONDITIONS IN YAQUI
4       C
5       C     IFLAG = INDEX INDICATING WHICH VELOCITY ARRAYS ARE TO BE SET
6       C           = 1 = UTIL,VTIL
7       C           = 2 = UL,VL
8       C           = 3 = UP,VP
9       C
10      C     WRITTEN BY J.L.NORTON,LASL T-3,1975
11      C
12      *     ***** BEGIN COMDECK YSTORE     *****
13      *     ***** BEGIN COMDECK YAQDIM     *****
14            DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
15           1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
16           2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
17           3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
18           4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
19           5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
20           6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
21      *     ***** END COMDECK YAQDIM      *****
22      *     ***** BEGIN COMDECK YAQSC      *****
23            LOGICAL RESTRT,FILM,PAPER,TURB
24            REAL LAM,MU
25      C     COMMON/YSC1/AASC(NSCP1)
26            COMMON/YSC1/AASC(9600)
27            COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
28           1 DTO(10),DTOC(10),DTO2,DTO8,DTPO8,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29           2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
30           3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
31           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,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,OTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     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,IOTC,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,IFO
*     ***** 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),OELSM,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),OLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),OLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TU0VEC,
     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),OLSROI(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),OKLSM(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,ROT,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,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,JOUMP,TWTHRD,TE,OTR,TMASS,DTVSAV,DTCSAV,IOTV
           5 ,JOTV,IOTC,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.
            TUQENG=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+V(IJP)/RM(IJP)
      IF(TURB) TUQENG=TUQENG+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
```

```
178      C
179           IF(J.EQ.2) P(IJM)=GM1*RO(IJM)*SIE(IJM)
180      C
181      C     SET PRESSURE OF TOP FICTITIOUS ROW USING GAMMA OF CELLS BELOW
182      C
183           IF(J.EQ.JP1) P(IJP)=GM1*RO(IJP)*SIE(IJP)
184      C
185      C     SEE IF WE ARE PROCESSING THE RIGHTMOST REAL COLUMN
186      C
187           IF(I.NE.IBAR) GO TO 40
188      C
189      C     YES.
190      C
191      C     SET PRESSURE OF RIGHT FICTITIOUS COLUMN USING GAMMA OF CELLS
192      C         TO THE LEFT
193      C
194           P(IPJ)=GM1*RO(IPJ)*SIE(IPJ)
195      C
196      C     SEE IF WE ARE PROCESSING THE BOTTOM REAL ROW
197      C
198           IF(J.NE.2) GO TO 30
199      C
200      C     YES.
201      C
202      C     SET PRESSURE OF BOTTOM CELL IN RIGHT FICTITIOUS COLUMN USING
203      C         GAMMA OF CELL (IBAR,2)
204      C
205           IPJM=IJM+NQ
206           P(IPJM)=GM1*RO(IPJM)*SIE(IPJM)
207           GO TO 40
208        30 CONTINUE
209      C
210      C     NO, SEE IF WE ARE PROCESSING THE TOP REAL ROW.
211      C
212           IF(J.NE.JP1) GO TO 40
213      C
214      C     YES.
215      C
216      C     SET PRESSURE OF UPPER CELL IN RIGHT FICTITIOUS COLUMN USING
217      C         GAMMA OF CELL (IBAR,JP1)
218      C
219           P(IPJP)=GM1*RO(IPJP)*SIE(IPJP)
220        40 CONTINUE
221           IJ=IPJ
222           IJP=IJP+NQ
223           IJM=IJM+NQ
224        50 CONTINUE
225      C
226      C     ADD THE I=IP1 VERTEX INTO THE POTENTIAL ENERGY
227      C
228           POTE=POTE+Y(IJ)/RM(IJ)
229      C
230      C     ADD THE (IP1,JP2) VERTEX INTO THE POTENTIAL ENERGY
231      C
232           IF(J.EQ.JP1) POTE=POTE+Y(IJP)/RM(IJP)
233           CALL LOOP
234        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,IFLMBZ
*     ***** 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,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,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,EPI,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLO,DTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
```

```
     5 ,JOTV,IOTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
     6 ,TMASV,WMAXEF,RMINEF,TSTRTD
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFO
*     ***** 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,OKLSM,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 BCO(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(IFO,250) ILABEL(KK),CON(KK)
        100 CONTINUE
            WRITE(IFO,260) QMN,QMX
      C
      C     LABEL THE TYPE OF PLOT
      C
            CALL LINCNT(58)
            WRITE(IFO,270)(ITITLE(I),I=1,NWT)
            WRITE(IFO,290) JNM,NAME,T,NCYC
      C
      C     PUT CONTOUR INFORMATION OUT TO CYCLE SUMMARY
      C
            IPOX=6
            WRITE(IPOX,280)(ITITLE(I),I=1,NWT)
            WRITE(IPOX,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
```

```
276           IJ1=IJM
277           IJ2=IJ
278     C
279     C     GO SET UP PLOT COORDINATES FOR THIS CROSSING
280     C
281           GO TO 170
282     C
283     C     SEE IF THE BOTTOM IS CROSSED BY THE CONTOUR
284     C
285       140 IF(K1+K2.NE.1) GO TO 150
286     C
287     C     YES
288     C
289           IC1=1
290           IC2=2
291           IJ1=IJM
292           IJ2=IPJM
293           GO TO 170
294     C
295     C     SEE IF RIGHT IS CROSSED
296     C
297       150 IF(K2+K4.NE.1) GO TO 160
298           IC1=2
299           IC2=4
300           IJ1=IPJM
301           IJ2=IPJ
302           GO TO 170
303     C
304     C     SEE IF TOP IS CROSSED
305     C
306       160 IF(K3+K4.NE.1) GO TO 220
307           IC1=3
308           IC2=4
309           IJ1=IJ
310           IJ2=IPJ
311     C
312     C     INCREMENT THE NO. OF SIDES CROSSED
313     C
314       170 LL=LL+1
315     C
316     C     CONTOUR LIES BETWEEN CQ(IJ1) AND CQ(IJ2). CALCULATE HOW FAR
317     C         ALONG THE LINE CONNECTING VERTICES IC1 AND IC2 THE
318     C         INTERSECTION SHOULD ACTUALLY LIE.
319     C
320           XX=(CON(KK)-CQ(IJ1))/(CQ(IJ2)-CQ(IJ1))
321     C
322     C     IX1(LL) AND IY1(LL) ARE THE RASTER COORDINATES OF THE POINT OF
323     C         INTERSECTION OF THE CONTOUR AND THE CONTOUR AREA SIDE
324     C         FOR INTERSECTION LL
325     C
326           IX1(LL)=FIXL+(XCO(IC1)+XX*(XCO(IC2)-XCO(IC1))-XL)*XCONV
327           IY1(LL)=FIYB+(YCO(IC1)+XX*(YCO(IC2)-YCO(IC1))-YLB)*YCONV
328     C
329     C     IF FOR SOME REASON THE INTERSECTION POINT LIES OUTSIDE OF THE
330     C         PLOTTING RECTANGLE,IGNORE THE INTERSECTION
331     C
332           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 ORV(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 QQ=E12.5)
```

```
390           END
```

---

```
 1            SUBROUTINE DEFINE
 2      C
 3      C     ROUTINE TO INITIALIZE PROBLEM INPUT VARIABLES
 4      C
 5      C     WRITTEN BY J.L.NORTON,LASL T-3,1974
 6      C
 7      *     ***** BEGIN COMDECK PARAM      *****
 8            COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
 9           1 NLCP3,NLCP4,IFLMSZ
10      *     ***** END COMDECK PARAM      *****
11      *     ***** BEGIN COMDECK YAQSC      *****
12            LOGICAL RESTRT,FILM,PAPER,TURB
13            REAL LAM,MU
14      C     COMMON/YSC1/AASC(NSCP1)
15            COMMON/YSC1/AASC(9600)
16            COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
17           1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,OZ,EM10,EPS,FIPXL,FIPXR,
18           2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
19           3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
20           4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
21            COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
22           1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
23           2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
24           3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
25            COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
26           1 ILNG,NILNG,TP3,TUPOT,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
27           2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,OTLAST,FIYBO,IYBO,YCNVLD,
28           3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,OMN,OMX,WMAX,JNM,T2,TLIM,
29           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
30           5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
31           6 ,TMASSV,WMAXEF,RMINEF,TSTRTO
32            COMMON/YSC2/ZZ
33      C     COMMON/YSC4/ITAB(ITABP)
34            COMMON/YSC4/ITAB(1000)
35            COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
36      *     ***** END COMDECK YAQSC      *****
37      C     MU AND LAM ARE THE VISCOSITY COEFFICIENTS
38      C
39            MU=0.
40            LAM=.6
41      C
42      C     OM IS THE PHASE TWO ITERATION RELAXATION PARAMETER OMEGA
43      C
44            OM=1.
45      C
46      C     EPS IS THE PRESSURE ITERATION CONVERGENCE CRITERION
47      C
48            EPS=1.E-5
49      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=0
C
C     OR IS THE UNIFORM RADIAL MESH SPACING
C
      DR=0.
C
C     OZ IS THE UNIFORM AXIAL MESH SPACING
C
      DZ=0.
C
C     CYL IS THE GEOMETRY INDICATOR, CYL=1. FOR CYLINORICAL GEOMETRY
C                                    =0. FOR PLANE (SLAB) GEOMETRY
C
      CYL=1.
C
C     GROVEL IS THE REZONE INDICATOR, GROVEL=0. FOR EULERIAN
C                                     =1. FOR LAGRANGIAN
C                                     =2. FOR ALE
C
      GROVEL=2.
C
C     ANC IS THE NODE COUPLER CONSTANT
C
      ANC=.05
C
C     A0FAC IS A FACTOR USED IN CALCULATING THE COURANT TIMESTEP
C
      A0FAC=.2
C
C     A0,A0M,AND B0 ARE REZONE COEFFICIENTS WHICH DETERMINE THE TYPE OF
C         DIFFERENCING USED IN CALCULATING THE FLUXING TERMS, A0 APPLIES
C         TO THE MOMENTUM EQN,A0M TO THE MASS AND ENERGY EQNS,AND
C         B0 TO ALL THREE, SOME EXAMPLES OF THEIR SIGNIFICANCE ARE
C              A0=0.,B0=0.,   CENTERED DIFFERENCING
C              A0=1.,B0=0.,   FULL DONOR CELL
C              A0=0.,B0=2.,   INTERPOLATED DONOR CELL,
C         NOTE THAT THE EQNS ARE UNSTABLE IN THE FIRST CASE,
C
      A0=.1
      A0M=1.
      B0=0.
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=0 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,3HRA$,3HFL$,3HP$D,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
```

---

```
      1     SUBROUTINE OVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
      2 C
      3 C     ROUTINE TO CALCULATE MAXIMUM VELOCITY AND MAXIMUM AND MINIMUM
      4 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 YAGDIM     -----
            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),OLSROI(1),OLSROQ(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 YAGDIM     -----
      *     ----- BEGIN COMDECK YAGSC      -----
            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,JUNF02,KXI,LAM,LPB,MU,NAME(8),
           1 NCYC,NLC,NPS,NPT,NQ,NQI,NQI8,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,YCNVLO,
           3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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 YAGSC      -----
      *     ----- BEGIN COMDECK YAGEQ      -----
            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),OLSROI,Y13K),(AASC(24),GZSV
           7 ),(AASC(25),OLSROQ,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
```

```
119            ORMIN=AMIN1(ORMIN,X14,X23)
120            DRMAX=AMAX1(DRMAX,X14,X23)
121            OZMIN=AMIN1(OZMIN,Y21,Y34)
122            OZMAX=AMAX1(DZMAX,Y21,Y34)
123            IJ=IPJ
124         20 IJP=IPJP
125            CALL LOOP
126         30 CONTINUE
127            DRMIN=SQRT(DRMIN)
128            DRMAX=SQRT(DRMAX)
129            OZMIN=SQRT(OZMIN)
130            DZMAX=SQRT(OZMAX)
131            RETURN
132            END
```

---

```
  1            FUNCTION ERF(Z)
  2      C
  3      C     ROUTINE TO CALCULATE THE STANDARD ERROR FUNCTION
  4      C
  5      C     STANDARD LIBRARY SUBPROGRAM - LOS ALAMOS SCIENTIFIC LABORATORY
  6      C     DOCUMENTED BY J.L.NORTON,LASL TD-3,1972
  7      C
  8            DIMENSION P(7,2),Q(6,2)
  9            DIMENSION A(14),B(12)
 10            EQUIVALENCE(A,P),(B,Q)
 11            X=ABS(Z)
 12      C
 13      C     IF ARGUMENT IS ZERO,ERF IS ZERO
 14      C
 15            ERF=0.
 16            IF(X.EQ.0.) RETURN
 17            DATA (A(I),I=1,14)/1.1283791670955,.34197505591854,
 18           1 .86290601455206E-1,.12382023274723E-1,.11986242418302E-2,
 19           2 .76537302607825E-4,.25365482058342E-5,-.99999707603738,
 20           3 -1.4731794832805,-1.0573449601594,-.44078839213875,
 21           4 -.10684197950781,-.12636031836273E-1,-.1149393366616E-8/
 22            DATA (B(I),I=1,12)/-.36359916427762,.52205830591727E-1,
 23           1 -.30613035688519E-2,-.46856639020338E-4,.15601995561434E-4,
 24           2 -.62143556409287E-6,2.6015349994799,2.9929556755308,
 25           3 1.9684584582884,.79250795276064,.18937020051337,
 26           4 .22396882835053E-1/
 27      C
 28      C     IF ARGUMENT IS GREATER THAN 5.5, ERF IS UNITY
 29      C
 30         10 ERF=SIGN(1.,Z)
 31      C
 32      C     USE RATIONAL APPROXIMATION TO COMPUTE ERF
 33      C
 34            IF(X.GE.5.5) RETURN
 35            J=1
 36            IF(X.GT.1.5) J=2
```

```
      FJ=J-1
      O=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,GROVEL,GZ,GZP,I,IBAR,
     3 IOTO,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,ROT,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,OTSV,OTLAST,FIYBO,IYBO,YCNVLO,
      3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,OTR,TMASS,OTVSAV,OTCSAV,IDTV
      5 ,JOTV,IDTC,JOTC,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      *****
*      ***** 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,OKLSM,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),GRRDR),(AASC(22),GRRDZ),(AASC(23),DLSRDI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),DLSRDQ,VG),(AASC(26),GRSV),(AASC(27),GRRDP,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,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,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,ROT,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,IY80,YCNVLD,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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,TSTRTO
      COMMON/YSC2/ZZ
C     COMMON/YSC4/ITAB(ITABP)
      COMMON/YSC4/ITAB(1000)
      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFO
*     ***** 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),QELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,QKLSM,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),QLSROQ,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)*
```

```
 94           2 (Y4-Y3))
 95             QMN=AMIN1(CQ(IJ),QMN)
 96             QMX=AMAX1(CQ(IJ),QMX)
 97             WSAV=WMAX
 98             WMAX=AMAX1(WMAX,ABS(CQ(IJ)))
 99             IF(WSAV.NE.WMAX) ISVW=I
100             IF(WSAV.NE.WMAX) JSVW=J
101             IJM=IJM+NQ
102             IJ=IPJ
103          20 IJP=IPJP
104             CALL LOOP
105          30 CONTINUE
106             CALL DONE
107             RETURN
108             END
```

```
 1              SUBROUTINE MODE(IARG,NARG)
 2       C
 3       C      ROUTINE TO ANALYZE THE PROGRAM STATUS DESIGNATION
 4       C
 5       C      WRITTEN BY J.L.NORTON,LASL TD=3,1973
 6       C
 7              DIMENSION ITROUB(12),ICAUSE(2,12)
 8              LOGICAL ITROUB
 9              INTEGER AND,COMP
10              DATA ICAUSE/9HUNDERFLOW,1H ,8HOVERFLOW,1H ,10HINDEFINITE,1H ,
11             1 4HSTEP,1H ,10HBREAKPOINT,1H ,10HPROGRAM RA,3HNGE,10HSCM DIRECT,
12             2 6H RANGE,10HLCM DIRECT,6H RANGE,10HSCM BLOCK ,5HRANGE,
13             3 10HLCM BLOCK ,5HRANGE,10HSCM PARITY,1H ,10HLCM PARITY,1H /
14              ITEST=AND(IARG,COMP(777777B))
15              IF(ITEST.NE.0) RETURN
16              ITEST=1
17              DO 10 IJ=1,12
18              ITROUB(IJ)=.FALSE.
19              IF(AND(IARG,ITEST).NE.0) ITROUB(IJ)=.TRUE.
20              ITEST=2*ITEST
21           10 CONTINUE
22              DO 20 IJ=1,12
23              IF(.NOT.ITROUB(IJ)) GO TO 20
24              WRITE(NARG,30) ICAUSE(1,IJ),ICAUSE(2,IJ)
25           20 CONTINUE
26              RETURN
27       C
28           30 FORMAT(1H 51HTHE FOLLOWING CONDITION EXISTED AT THE TIME OF THE ,
29             1 7HABORT=,2A10)
30              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),DKL$M(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 IOTO,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,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,JOUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IOTV
     5 ,JOTV,IOTC,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,IPO,IFO
*     ----- 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,OKLSM,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
```

```
114            R(IJP)=R(IJ)
115            U(IJP)=0.
116            V(IJP)=0.
117         30 IJP=IJP+NQ
118            IJM=IJM+NQ
119            XX=XX+DR
120         40 IJ=IJ+NQ
121            XX=0.
122            YY=YY+DZ
123            CALL LOOP
124         50 CONTINUE
125            CALL DONE
126      C
127      C     SEE IF THE ENTIRE MESH ACTUALLY IS UNIFORM
128      C
129            IF(FREZXR.EQ.1..AND.FREZYT.EQ.1..AND.FREZYB.EQ.1.) GO TO 130
130      C
131      C     NO. DO THE NON-UNIFORM GENERATION.
132      C
133      C     CONVERT JCEN TO AN ACTUAL VERTEX NO. JCEN IS THE J-LINE THAT
134      C         GOES THROUGH THE CENTER OF THE UNIFORM REGION.
135      C
136            JCEN=JCEN+2
137      C
138      C     JTOP AND JBOT ARE THE J-LINES AT THE TOP AND BOTTOM OF THE
139      C         UNIFORM REGION
140      C
141            JTOP=JCEN+JUNF02
142            JBOT=JCEN-JUNF02
143      C
144      C     TJ IS THE DISTANCE FROM THE CENTER TO THE TOP (OR BOTTOM) OF
145      C         THE UNIFORM REGION
146      C
147            TJ=FLOAT(JUNF02)*DZ
148      C
149      C     LOOP TO SET THE NON-UNIFORM VERTICES
150      C
151            CALL START
152            DO 110 J=2,JP2
153            DO 100 I=1,IP1
154            IMJ=IJ-NQ
155            IF(FREZXR.EQ.1.) GO TO 60
156      C
157      C     SEE IF WE ARE WITHIN THE UNIFORM X REGION
158      C
159            IF(I.LE.IUNF+1) GO TO 60
160      C
161      C     NO. GENERATE THE NON-UNIFORM X USING GEOMETRIC PROGRESSION.
162      C
163            X(IJ)=X(IMJ)+FREZXR*(X(IMJ)-X(IMJ-NQ))
164            R(IJ)=X(IJ)*CYL+OMCYL
165         60 CONTINUE
166      C
167      C     JDT IS THE NO. OF J LINES THAT THE CURRENT J IS ABOVE THE TOP
168      C         OF THE UNIFORM REGION
169      C
170            JDT=J-JTOP
```

```
      C
      C     JOB IS THE NO. OF J LINES THAT THE CURRENT J IS BELOW THE BOTTOM
      C         OF THE UNIFORM REGION
      C
            JOB=JBOT-J
      C
      C     SEE IF THE CURRENT J IS BELOW THE UNIFORM REGION
      C
            IF(JOB.GT.0) GO TO 70
      C
      C     NO. SEE IF IT IS ABOVE THE UNIFORM REGION.
      C
            IF(JOT.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=JOB, 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**JOB)/(1-FREZ).
      C
            IF(FREZYB.GT.1.) Y(IJ)=REZY0-TJ-DZ*FREZYB*(1.-FREZYB**JOB)*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**JOT)*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
```

```
228      C
229      C     CALCULATE Y FOR J=1 FICTITIOUS CELLS
230      C
231           CALL START
232           DO 120 I=1,IP1
233           Y(IJM)=Y(IJ)-(Y(IJP)-Y(IJ))*FREZYB
234           IJ=IJ+NQ
235           IJM=IJM+NQ
236           IJP=IJP+NQ
237       120 CONTINUE
238           CALL DONE
239      C
240      C     MESH IS GENERATED
241      C
242       130 CONTINUE
243      C
244      C     ***************************************************************************
245      C     GENERATE AN INITIAL FIREBALL AND ITS AMBIENT ATMOSPHERE
246      C     ***************************************************************************
247      C
248      C     FIRST SET UP THE AMBIENT ATMOSPHERE
249      C
250      C     GET GAMMA-1 AT REZY0
251      C
252           ETMP=REZSIE
253           ROTMP=REZRON
254           CONCJ=0.
255           CALL AIR
256      C
257      C     XX IS THE ISOTHERMAL CONSTANT IN THE PRESSURE
258      C
259           XX=GMONE*REZSIE
260      C
261      C     NOTE THAT A NEGATIVE GRAVITATIONAL FORCE IS ASSUMED HERE
262      C
263           YY=.5*ABS(GZ)
264      C
265      C     BRING IN THE FIRST THREE ROWS
266      C
267           CALL START
268      C
269      C     PROCESS J=1 AND 2 ROWS. YJC2 IS THE Y COORDINATE OF CENTERS OF
270      C         CELLS IN THE ROW J=2
271      C
272           YJC2=.5*(Y(IJP)+Y(IJ))
273      C
274      C     ROSAV IS THE DENSITY OF THE CELLS IN ROW J=2 ASSUMING AN
275      C         ISOTHERMAL,IDEAL GAS ATMOSPHERE
276      C
277           ROSAV=REZRON*EXP(-GZ*(REZY0-YJC2)/XX)
278      C
279      C     CALCULATE THE DENSITY OF CELLS IN ROW J=1 USING THE DIFFERENCE
280      C         FORM FOR HYDROSTATIC EQUILIBRIUM
281      C
282           FNUM=(Y(IJP)-Y(IJ))*YY
283           FDEN=FNUM*FREZYB
284           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=IPO,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
```

```
456             VYB=Y(IJ)
457             VYT=Y(IJP)
458       C
459       C     CONC(IJ)=0. INDICATES NOTHING HAS BEEN SET IN THE
460       C         ZONE YET
461       C
462             CONC(IJ)=0.
463       C
464       C     LOOP OVER ALL 1-D ZONES
465       C
466             DO 280 K=2,ISUB
467       C
468       C     SEE IF ZONE COULD NOT POSSIBLY FALL WITHIN THE YAQUI ZONE
469       C
470             RVRT=VXR*VXR+(VYT-REZY0)**2
471             RVRB=VXR*VXR+(VYB-REZY0)**2
472             RFIRE=XFIRE(K-1)**2
473             IF(RFIRE.GT.RVRT.AND.RFIRE.GT.RVRB) GO TO 280
474             RVLT=VXL*VXL+(VYT-REZY0)**2
475             RVLB=VXL*VXL+(VYB-REZY0)**2
476             RFIRE=XFIRE(K)**2
477             IF(RFIRE.LT.RVLT.AND.RFIRE.LT.RVLB) GO TO 280
478       C
479       C     LOOP OVER ANGULAR INCREMENTS
480       C
481             THETA2=0.
482             DNTH=180./FLOAT(NTH)
483             DO 270 IT=1,NTH
484             THETA1=THETA2
485             THETA2=THETA2+DNTH
486             CTHETA=.5*(THETA1+THETA2)
487             SINT=SDG(CTHETA)
488             COST=CDG(CTHETA)
489       C
490       C     LOOP OVER RADIAL INCREMENTS
491       C
492             R2=XFIRE(K-1)
493             DR=(XFIRE(K)-R2)/FLOAT(NRAD)
494             DO 260 IR=1,NRAD
495             R1=R2
496             R2=R2+DR
497             CR=.5*(R1+R2)
498             XCEN=CR*SINT
499       C
500       C     SEE IF THE X-COORDINATE OF THE CENTER OF THE LAGRANGIAN
501       C         PIECE LIES IN THE YAQUI CELL OF INTEREST
502       C
503             IF(XCEN.LT.VXL.OR.XCEN.GT.VXR) GO TO 260
504       C
505       C     YES, CHECK THE Y-COORDINATE
506       C
507             YCEN=CR*COST+REZY0
508             IF(YCEN.LT.VYB.OR.YCEN.GT.VYT) GO TO 260
509       C
510       C     CONSIDER THE PIECE IN THE YAQUI CELL, SEE IF ANY PIECES
511       C         HAVE PREVIOUSLY BEEN FOUND.
512       C
```

```
513             IF(CONC(IJ).NE.0.) GO TO 250
514       C
515       C     NO. INITIALIZE.
516       C
517             RO(IJ)=0.
518             U(IJ)=0.
519             V(IJ)=0.
520             SIE(IJ)=0.
521         250 CONTINUE
522       C
523       C     COMPUTE THE MASS OF THE PIECE. ACCUMULATE THE VOLUME
524       C         IN CONC.
525       C
526             VOL=THIRD*(COG(THETA1)-COG(THETA2))*(R2**3-R1**3)
527             CONC(IJ)=CONC(IJ)+VOL
528             XMASS=VOL*RHOFIR(K)
529       C
530       C     COMPUTE THE VELOCITIES OF THE PIECE
531       C
532             RATIO=(CR-XFIRE(K-1))/(XFIRE(K)-XFIRE(K-1))
533             RDOT=UFIRE(K-1)+RATIO*(UFIRE(K)-UFIRE(K-1))
534             XDOT=RDOT*SINT
535             YDOT=RDOT*COST
536       C
537       C     DEPOSIT MASS,MOMENTUM,AND ENERGY IN THE YAQUI ZONE
538       C
539             U(IJ)=U(IJ)+XDOT*XMASS
540             V(IJ)=V(IJ)+YDOT*XMASS
541             SIE(IJ)=SIE(IJ)+EFIRE(K)*XMASS
542             RO(IJ)=RO(IJ)+XMASS
543         260 CONTINUE
544         270 CONTINUE
545         280 CONTINUE
546             IJ=IJ+NQ
547             IJP=IJP+NQ
548         290 CONTINUE
549             CALL LOOP
550         300 CONTINUE
551             CALL DONE
552       C
553       C     LOOP OVER ALL CELLS AND CHECK FOR FIREBALL CELLS
554       C
555             CALL START
556             DO 390 J=2,JP1
557             DO 380 I=1,IBAR
558             IF(CONC(IJ).EQ.0.) GO TO 370
559       C
560       C     FIREBALL CELL FOUND. SEE IF ITS VOLUME IS VERY FAR OFF.
561       C
562             IPJ=IJ+NQ
563             VOL=.5*(Y(IJP)-Y(IJ))*(X(IPJ)**2-X(IJ)**2)
564             RERROR=(CONC(IJ)-VOL)/VOL
565             IF(ABS(RERROR).LT..01) GO TO 370
566       C
567       C     ERROR IS LARGE. INFORM THE USER AND CONSIDER CELL AS
568       C         FIREBALL BOUNDARY CELL.
569       C
```

```
570            DO 310 IPX=6,IFD,6
571          310 WRITE(IPX,560) I,J,RERROR
572        C
573        C       USE VALUES FROM NEIGHBORING CELL FOR DETERMINING
574        C           CONTRIBUTION OF PART OF CELL OUTSIDE OF THE FIREBALL
575        C
576            IF(CONC(IPJ).NE.0.) GO TO 320
577        C
578        C       USE CELL TO RIGHT
579        C
580            INDEX=IPJ
581            GO TO 360
582          320 CONTINUE
583            IF(CONC(IJP).NE.0.) GO TO 330
584        C
585        C       USE CELL ABOVE
586        C
587            INDEX=IJP
588            GO TO 360
589          330 CONTINUE
590            IF(CONC(IJM).NE.0.) GO TO 340
591        C
592        C       USE CELL BELOW
593        C
594            INDEX=IJM
595            GO TO 360
596          340 CONTINUE
597            IPJP=IJP+NQ
598            IF(CONC(IPJP).NE.0.) GO TO 350
599        C
600        C       USE CELL TO UPPER RIGHT
601        C
602            INDEX=IPJP
603            GO TO 360
604          350 CONTINUE
605            IPJM=IJM+NQ
606            IF(CONC(IPJM).NE.0.) CALL UNCLE(1,6HMSHMKR,50,
607           1 50HERROR IN PROCESSING CELL WITH LARGE RELATIVE ERROR)
608        C
609        C       USE CELL TO LOWER RIGHT
610        C
611            INDEX=IPJM
612          360 CONTINUE
613        C
614        C       ADD IN CONTRIBUTION FROM PART OF CELL OUTSIDE OF THE FIREBALL
615        C
616            DV=VOL-CONC(IJ)
617            DM=DV*RO(INDEX)
618            RO(IJ)=RO(IJ)+DM
619            SIE(IJ)=SIE(IJ)+DM*SIE(INDEX)
620          370 CONTINUE
621            IJ=IJ+NQ
622            IJP=IJP+NQ
623            IJM=IJM+NQ
624          380 CONTINUE
625            CALL LOOP
626          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
```

```
684             IF(CONC(IJ).EQ.0.) GO TO 450
685             U(IJ)=UG(IJ)
686             V(IJ)=VG(IJ)
687             IPJ=IJ+NQ
688             VOL=.5*(Y(IJP)-Y(IJ))*(X(IPJ)**2-X(IJ)**2)
689             SIE(IJ)=SIE(IJ)/RO(IJ)
690             RO(IJ)=RO(IJ)/VOL
691           450 CONTINUE
692             IJ=IJ+NQ
693             IJP=IJP+NQ
694           460 CONTINUE
695             CALL LOOP
696           470 CONTINUE
697             CALL DONE
698         C
699         C       SET CONCENTRATIONS. CONCJ=1. INDICATES METHANE,CONCJ=0. IS AIR.
700         C           EVERYTHING WITH SIE.GE.1.E10 IS CONSIDERED METHANE. ALL ELSE
701         C           IS AIR.
702         C
703             CALL START
704         C
705         C       SET THE BOTTOM FICTITIOUS ROW
706         C
707             DO 480 I=1,IP1
708             CONC(IJM)=0.
709             IJM=IJM+NQ
710           480 CONTINUE
711         C
712         C       SET THE REAL ROWS AND THE RIGHT FICTITIOUS COLUMN
713         C
714             DO 500 J=2,JP1
715             DO 490 I=1,IP1
716             CONC(IJ)=0.
717             IF(SIE(IJ).GE.1.E10.AND.I.NE.IP1) CONC(IJ)=1.
718             IJ=IJ+NQ
719           490 CONTINUE
720             CALL LOOP
721           500 CONTINUE
722         C
723         C       SET THE TOP FICTITIOUS ROW
724         C
725             DO 510 I=1,IP1
726             CONC(IJ)=0.
727             IJ=IJ+NQ
728           510 CONTINUE
729             CALL DONE
730         C
731         C       ZERO OUT U AND V IN THE BOTTOM FICTITIOUS ROW
732         C
733             CALL START
734             DO 520 I=1,IP1
735             U(IJM)=0.
736             V(IJM)=0.
737             IJM=IJM+NQ
738           520 CONTINUE
739             CALL DONE
740             RETURN
```

```
741         530 CONTINUE
742             CALL UNCLE(4,6HMSHMKR,34,34HMESH NAMELIST INITIALIZATION ERROR)
743       C
744         540 FORMAT(/1H ,34HFIREBALL INPUT WILL BE FROM FILE 3)
745         550 FORMAT(1H /1H ,28HALL FIREBALL INPUT DATA READ/1H )
746         560 FORMAT(1H ,5HCELL ,2I5,32H HAS A RELATIVE VOLUME ERROR OF ,1PE12.5
747            1 ,1H.,2X,46HIT WILL BE TREATED AS A FIREBALL BOUNDARY CELL)
748         570 FORMAT(1H1,22HMESHMKR FIREBALL INPUT//1H ,13X1HU,19X1HE,18X3HRHO,
749            1 17X1HX,19X1HP,17X4HPRHO/1H )
750         580 FORMAT(5E15.0)
751         590 FORMAT(1H ,1P6E19.7)
752             END
```

```
1             SUBROUTINE MTHANE
2       C
3       C     ROUTINE TO CALCULATE THE FACTOR GAMMA-1 FOR METHANE
4       C
5       C     UNITS ARE ALL CGS
6       C
7       C     ORIGINALLY OBTAINED FROM THE AIR FORCE WEAPONS LAB
8       C     MODIFIED BY J.L.NORTON,LASL T-3,1974
9       C
10            COMMON/EQNST/RHO,E,GMONE
11            DIMENSION A(8)
12            DATA A/3.21782E-1,-1.56848E-4,-8.78899E-2,1.25271E-3,1.46612E-2,
13           1 -1.72460E-6,-7.72322E-4,-8.63413E-6/
14            U=E*1.E-10
15            R=ALOG(RHO)
16            GMONE=A(1)+R*(A(2)+U*(A(4)+U*(A(6)+A(8)*U)))+U*(A(3)+U*(A(5)+A(7)*
17           1 U))
18            RETURN
19            END
```

```
1             SUBROUTINE NAMLST(TABLE,IF,IEFLAG)
2       C
3       C     ROUTINE TO REPLACE COMPILER/SYSTEM DEPENDENT NAMELIST INPUT.
4       C         HERE,AN ATTEMPT HAS BEEN MADE TO ELIMINATE AS MANY
5       C         COMPILER- AND/OR SYSTEM-DEPENDENT FEATURES AS POSSIBLE. THOSE
6       C         STILL REMAINING ARE BOOLEAN ALGEBRA,60 BIT WORD ASSUMPTION,
7       C         INLINE SHIFT FUNCTION (CAN BE REPLACED BY ASSEMBLY-LANGUAGE
8       C         FUNCTION IF IT IS TYPED INTEGER),AND TWO ASSEMBLY-LANGUAGE
9       C         ROUTINES TO FETCH AND STORE WORDS FROM AND INTO ABSOLUTE
10      C         MEMORY LOCATIONS.
11      C
12      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 MESSDT)
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) (77655B)
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=3.14159265358979320 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 CS IN CC1-2 WILL BE PRINTED OUT AND IGNORED.
C          ANY CARD WITH PS 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. -0=-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/CARON/I,J,K,A,B,X
C               READ(5,CARON)
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
            ASSIGN 999 TO IERRT
            CALL TABDEF(TABLE,5HCARDN,7,IERRT)
      C
      C     ENTER EACH VARIABLE INTO THE TABLE. IF THERE IS SOME
      C          ERROR,IEFLAG IS RETURNED NON-ZERO.
      C
            CALL TABSET(TABLE,1HI,I,IEFLAG,0,0,0,0)
      C
      C     (THE 0 DENOTES A NON-SUBSCRIPTED VARIABLE)
      C
            CALL TABSET(TABLE,1HJ,J,IEFLAG,0,0,0,0)
            CALL TABSET(TABLE,1HK,K,IEFLAG,0,0,0,0)
            CALL TABSET(TABLE,1HA,A,IEFLAG,1,0,0,0)
      C
      C     (THE 1 DENOTES A SUBSCRIPTED VARIABLE BUT THE SUBSCRIPT
      C          NEED NOT BE GIVEN FOR A SINGLY-SUBSCRIPTED VARIABLE)
      C
            CALL TABSET(TABLE,1HB,B,IEFLAG,2,5,0,0)
      C
      C     (THE VARIABLE IS DOUBLY-SUBSCRIPTED BUT ONLY THE FIRST
      C          SUBSCRIPT NEED BE GIVEN)
      C
            CALL TABSET(TABLE,1HX,X,IEFLAG,3,5,10,0)
      C
      C     (THE VARIABLE IS TRIPLY-SUBSCRIPTED BUT ONLY THE FIRST
      C          TWO SUBSCRIPTS NEED BE GIVEN)
      C
      C     NOTE THAT TABSET IS CALLED WITH THE FULL NO. OF ARGUMENTS
      C         WHETHER THEY ARE USED OR NOT. THIS IS NOT STRICTLY
      C         NECESSARY ON MOST COMPILERS. CHECK YOUR LOCAL
      C         CONVENTIONS.
      C
      C     READ IN A NAMELIST CARD. IF IEFLAG RETURNS 0,ALL WAS OK.
      C          IF IEFLAG RETURNS -1,A NORMAL EOF WAS ENCOUNTERED.
      C          FOR ALL OTHER VALUES OF IEFLAG,AN ERROR OCCURRED.
      C          IF IEFLAG.LT.0,AN EOF ALSO OCCURRED.
      C
            CALL NAMLST(TABLE,5,IEFLAG)
      C
      C     (5 IS THE INPUT FILE NO.)
      C
            IF(IEFLAG.EQ.0) GO TO 10
            IF(IEFLAG.EQ.(-1)) GO TO 20
      C
      C     A NAMELIST ERROR OCCURRED
      C
            STOP 7
      C
      C     A NAMELIST TABLE INITIALIZATION ERROR OCCURRED
      C
        999 CONTINUE
            . . .
      C
      C     AN EOF OCCURRED
      C
         20 CONTINUE
```

```
184      C                 . . .
185      C          C
186      C          C      NORMAL EXIT
187      C          C
188      C             10 CONTINUE
189      C                 . . .
190      C     ************************************************************************
191      C
192      C     WRITTEN BY J.L.NORTON,LASL T-3,1974
193      C
194      C     THIS VERSION RUNS ON A CDC 6600 OR 7600 USING THE RUN COMPILER
195      C
196            DIMENSION TABLE(3,1),IDSTR(80)
197            INTEGER TABLE,AND,OR,COMP,SHIFT
198            LOGICAL LIST,CMPLEX,DIM,DELIM,FIX,MULT,CFRST,HOLLER,DOBLE
199            LOGICAL SAVE,FIRST,DONE,EXPN,PERIOD,LCONT,LCONF,CDONE,CPFND
200            COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
201           1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
202            COMMON/ERRORC/IERRT
203            COMMON/CSHIFT/XLEFT,XRIGHT,XOUM
204            DOUBLE PRECISION DB,DBP,DCONST,DTEMP,DZERO
205            EQUIVALENCE(DB,CREAL)
206            EQUIVALENCE(IVSUM,XSUM)
207            DATA LCONT,LCONF/.TRUE.,.FALSE./
208            DATA IAF,IAL/1RA,1RZ/
209            DATA INF,INL/1R0,1R9/
210            DATA DCONST/10.,0./,DZERO/0.,0./
211            DATA IEMIN,IEMAX/-294,322/
212            DATA NDSP,NDDP/14,28/
213            DATA INAME/0/
214      C
215      C     LIST = IF TRUE,EACH NAMELIST CARD INPUT WILL BE LISTED
216      C
217            LIST=.FALSE.
218      C
219      C     BEGINNING OF CODE FOR PROCESSING A NAMELIST RECORD
220      C
221         10 CONTINUE
222      C
223      C     RESET THE ERROR FLAG
224      C
225            IEFLAG=0
226      C
227      C     READ A CARD
228      C
229            CALL READIT(IF,IEFLAG)
230      C
231      C     IF AN END-OF-FILE WAS ENCOUNTERED,QUIT
232      C
233            IF(IEFLAG.LT.0) RETURN
234      C
235      C     ISUB IS THE POINTER TO THE COLUMN BEING PROCESSED OF THE CARD
236      C        LAST READ, BEGIN LOOKING AT CC2.
237      C
238            ISUB=2
239      C
240      C     IF CC2 IS NOT A DOLLAR,IGNORE THE CARD AND GO BACK AND READ
```

```
241      C          ANOTHER ONE
242      C
243            IF(ICHAR(2),NE,1H$) GO TO 10
244      C
245      C     CC2 WAS A DOLLAR. IF CC1 IS A "P",TURN ON THE LIST FLAG.
246      C
247            IF(ICHAR(1),EQ,1HP) LIST=,TRUE.
248      C
249      C     IF CC1 IS A C,PRINT THE CARD AND GO READ ANOTHER
250      C
251            IF(ICHAR(1),NE,1HC) GO TO 20
252            CALL MESSOT(1)
253            GO TO 10
254      C
255      C     CARD IS TO BE PROCESSED.
256      C
257         20 CONTINUE
258      C
259      C     IVNUM IS THE NO. OF CHARACTERS READ IN THE CURRENT STRING
260      C
261            IVNUM=0
262      C
263      C     ************************************************************************
264      C     DECODE THE NAMELIST NAME
265      C     ************************************************************************
266      C
267         30 CONTINUE
268      C
269      C     GO TO THE NEXT COLUMN
270      C
271            ISUB=ISUB+1
272      C
273      C     ITEST IS THE CHARACTER IN CC ISUB,LEFT-JUSTIFIED,BLANK FILL
274      C
275            ITEST=ICHAR(ISUB)
276      C
277      C     CHECK FOR A BLANK
278      C
279            IF(ITEST,EQ,1H ) GO TO 70
280      C
281      C     NO, UP THE CHARACTER COUNT. IF MORE THAN SIX,FATAL ERROR.
282      C
283            IVNUM=IVNUM+1
284            IF(IVNUM,LE,6) GO TO 40
285            CALL MESSOT(2)
286            IEFLAG=2
287            RETURN
288         40 CONTINUE
289      C
290      C     ITEMP IS THE CHARACTER IN CC ISUB,RIGHT-JUSTIFIED,ZERO FILL
291      C
292            ITEMP=AND(SHIFT(ITEST,-54),77B)
293      C
294      C     CHECK FOR THE FIRST CHARACTER OF THE STRING
295      C
296            IF(IVNUM,GT,1) GO TO 50
297      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
      IOSTR(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(IOSTR,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
```

```
355      C
356      C     ***************************************************************
357      C     MAIN LOOP FOR DECODING VARIABLE NAMES AND CONSTANTS
358      C     ***************************************************************
359      C
360         90 CONTINUE
361      C
362      C     ONE VARIABLE NAME AND CONSTANT HAVE ALREADY BEEN PROCESSED,
363      C         EITHER A VARIABLE OR A CONSTANT CAN OCCUR NEXT,
364      C
365           FIRST=.FALSE.
366      C
367      C     ---------------------------------------------------------------
368      C     SUBLOOP TO DETERMINE WHETHER NEXT STRING IS VARIABLE NAME OR
369      C         CONSTANT
370      C     ---------------------------------------------------------------
371      C
372        100 CONTINUE
373      C
374      C     NXTCOL PUTS THE CONTENTS OF THE NEXT CARD COLUMN INTO ITEST,ALSO
375      C         UPDATING ISUB, IF NECESSARY,ANOTHER CARD IS READ AND ERROR
376      C         CHECKS ARE PERFORMED,
377      C
378           CALL NXTCOL(IF,IEFLAG)
379      C
380      C     IF ANY ERRORS,QUIT RIGHT HERE
381      C
382           IF(IEFLAG.NE.0) RETURN
383      C
384      C     NO ERRORS, SEE IF NEXT CHARACTER IS A BLANK, IF IT IS,GO ON TO
385      C         THE NEXT COLUMN,
386      C
387           IF(ITEST.EQ.1H ) GO TO 100
388      C
389      C     NOT A BLANK, SEE IF IT IS A DOLLAR, IF SO,WE ARE ALL DONE,
390      C
391           IF(ITEST.EQ.1H$) RETURN
392      C
393      C     NOT A DOLLAR, SEE IF A DELIMITER IS EXPECTED,
394      C
395           IF(.NOT.DELIM) GO TO 110
396      C
397      C     YES, IF CHARACTER IS NOT A COMMA,FATAL ERROR,
398      C
399           IF(ITEST.NE.1H,) GO TO 260
400      C
401      C     CHARACTER IS A COMMA, REMOVE DELIMITER FLAG AND GO EXAMINE
402      C         THE NEXT COLUMN,
403      C
404           DELIM=.FALSE.
405           GO TO 90
406        110 CONTINUE
407      C
408      C     SEE IF CHARACTER IS ALPHABETIC
409      C
410           ITEMP=AND(SHIFT(ITEST,-54),77B)
411           IF(ITEMP.GE.IAF.AND.ITEMP.LE.IAL) GO TO 130
```

```
412      C
413      C     NO. IF A NAME IS EXPECTED,FATAL ERROR.
414      C
415           IF(FIRST) GO TO 120
416      C
417      C     A NAME IS NOT NECESSARILY EXPECTED. LEGAL CHARACTERS ARE NUMBERS,
418      C         +,-,.,OR (. IF NONE OF THESE,FATAL ERROR. IF ONE OF THESE,
419      C         PROCEED TO SECTION WHICH DECODES CONSTANTS.
420      C
421           IF(ITEMP.GE.INF.AND.ITEMP.LE.INL) GO TO 400
422           IF(ITEST.EQ.1H+.OR.ITEST.EQ.1H-.OR.ITEST.EQ.1H..OR.ITEST.EQ.1H(
423          1 GO TO 400
424       120 CONTINUE
425           CALL MESSOT(13)
426           IEFLAG=13
427           RETURN
428      C
429      C     ------------------------------------------------------------
430      C     SUBLOOP TO DECODE VARIABLE NAMES
431      C     ------------------------------------------------------------
432      C
433       130 CONTINUE
434      C
435      C     STORE THE FIRST CHARACTER
436      C
437           IVNUM=1
438           IDSTR(1)=ITEST
439      C
440      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
441      C     SUBLOOP TO GET CHARACTERS OF VARIABLE NAME
442      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
443      C
444       140 CONTINUE
445      C
446      C     GET THE NEXT CHARACTER AND CHECK FOR ERRORS
447      C
448           CALL NXTCOL(IF,IEFLAG)
449           IF(IEFLAG.NE.0) RETURN
450      C
451      C     SEE IF CHARACTER IS ALPHANUMERIC. IF NOT,ASSUME THAT THE END OF
452      C         THE NAME HAS BEEN REACHED.
453      C
454           ITEMP=AND(SHIFT(ITEST,-54),77B)
455           IF((ITEMP.GE.IAF.AND.ITEMP.LE.IAL).OR.(ITEMP.GE.INF.AND.ITEMP.LE.
456          1 INL)) GO TO 150
457           GO TO 170
458       150 CONTINUE
459      C
460      C     CHARACTER IS ALPHANUMERIC. SEE IF IT IS NO. 7. IF SO,FATAL ERROR.
461      C
462           IF(IVNUM.GE.6) GO TO 160
463      C
464      C     NOT CHARACTER 7. STORE AND GO ON TO THE NEXT.
465      C
466           IVNUM=IVNUM+1
467           IDSTR(IVNUM)=ITEST
468           GO TO 140
```

```
469       160 CONTINUE
470           CALL MESSOT(3)
471           IEFLAG=3
472           RETURN
473       C
474       C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
475       C
476       170 CONTINUE
477       C
478       C     NAME HAS BEEN DECODED, PUT INTO ONE WORD.
479       C
480           CALL MASH(IDSTR,IVNUM,INAME,IEFLAG)
481           IF(IEFLAG.NE.0) RETURN
482       C
483       C     SCAN THE TABLE TO SEE IF IT IS A LEGAL VARIABLE, IF NOT,
484       C         FATAL ERROR.
485       C
486           DO 180 I=2,INE
487       180 IF(INAME.EQ.TABLE(1,I)) GO TO 190
488           CALL MESSOT(4)
489           IEFLAG=4
490           RETURN
491       C
492       C     LEGAL VARIABLE NAME HAS BEEN FOUND, INITIALIZE FOR SUBSEQUENT
493       C         SCAN.
494       C
495       190 CONTINUE
496       C
497       C     FIRST HERE DENOTES THAT A CONSTANT HAS NOT  YET BEEN FOUND FOR
498       C         THIS VARIABLE
499       C
500           FIRST=.TRUE.
501       C
502       C     IENTRY IS THE SECOND SUBSCRIPT OF THE TABLE ENTRY FOR THIS
503       C         VARIABLE
504       C
505           IENTRY=I
506       C
507       C     ISBSPT IS THE STORAGE OFFSET FOR THE VARIABLE, IF THE VARIABLE
508       C         IS LOCATED BEGINNING IN MEMORY LOCATION N,THEN THE NEXT
509       C         CONSTANT WILL BE STORED STARTING IN LOCATION N+ISBSPT-1.
510       C
511           ISBSPT=1
512       C
513       C     NSUBV(I) IS THE VALUE OF THE ITH SUBSCRIPT
514       C
515           DO 200 I=1,4
516       200 NSUBV(I)=0
517       C
518       C     NSUB IS THE NUMBER OF SUBSCRIPTS
519       C
520           NSUB=0
521       C
522       C     DIM IS .TRUE. IF THE VARIABLE IS DIMENSIONED
523       C
524           DIM=.FALSE.
525       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
```

```
697           DELIM=.FALSE.
698           GO TO 240
699         360 CONTINUE
700       C
701       C     RIGHT PAREN FOUND, TERMINATE SUBSCRIPT PROCESSING, CHECK IF PAREN
702       C         IS FIELD TERMINATOR.
703       C
704           IF(IVNUM.NE.0) GO TO 370
705       C
706       C     NO, CHECK FOR ISOLATED SIGN, IF SO,FATAL ERROR.
707       C
708           IF(NOSGN.EQ.0) GO TO 380
709           CALL MESSOT(30)
710           IEFLAG=30
711           RETURN
712         370 CONTINUE
713       C
714       C     YES, STORE THE SUBSCRIPT AND PRETEND LIKE ANOTHER IS TO BE
715       C         DECODED.
716       C
717           NSUBV(NSUB)=ISIGN(IVSUM,IVSIGN)
718           NSUB=NSUB+1
719       C
720       C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
721       C
722         380 CONTINUE
723       C
724       C     LAST SUBSCRIPT COMPLETED, SET NSUB TO THE ACTUAL NO. OF SUBSCRIPTS
725       C
726           NSUB=NSUB-1
727       C
728       C     ----------------------------------------------------------------
729       C
730         390 CONTINUE
731       C
732       C     SCAN CHARACTERS UNTIL AN = IS FOUND, ONLY OTHER VALID CHARACTER
733       C         IS A BLANK.
734       C
735           CALL NXTCOL(IF,IEFLAG)
736           IF(IEFLAG.NE.0) RETURN
737           IF(ITEST.EQ.1H ) GO TO 390
738           IF(ITEST.EQ.1H=) GO TO 400
739           CALL MESSOT(9)
740           IEFLAG=9
741           RETURN
742       C
743       C     ****************************************************************
744       C     SECTION TO DECODE CONSTANTS
745       C     ****************************************************************
746       C
747         400 CONTINUE
748       C
749       C     RESET ALL FLAGS
750       C
751       C     CMPLEX = .TRUE. IF A COMPLEX CONSTANT IS BEING DECODED
752       C     CDONE  = .TRUE. IF THE SECOND HALF OF A COMPLEX CONSTANT HAS
753       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
```

```
811             CALL NXTCOL(IF,IEFLAG)
812             IF(IEFLAG.NE.0) RETURN
813         430 CONTINUE
814       C
815       C     ------------------------------------------------------------
816       C     DECIDE WHAT TO DO BASED UPON WHAT CHARACTER WAS LAST READ
817       C     ------------------------------------------------------------
818       C
819       C     CHECK FOR A BLANK
820       C
821             IF(ITEST.EQ.1H ) GO TO 470
822       C
823       C     NO. CHECK FOR A NUMBER.
824       C
825             ITEMP=AND(SHIFT(ITEST,-54),778)
826             IF(ITEMP.GE.INF.AND.ITEMP.LE.INL) GO TO 450
827       C
828       C     NO. CHECK FOR A COMMA.
829       C
830             IF(ITEST.EQ.1H,) GO TO 1050
831       C
832       C     NO. CHECK FOR A PERIOD.
833       C
834             IF(ITEST.EQ.1H.) GO TO 1140
835       C
836       C     NO. CHECK FOR A PLUS OR MINUS.
837       C
838             IF(ITEST.EQ.1H+) GO TO 990
839             IF(ITEST.EQ.1H-) GO TO 990
840       C
841       C     NO. CHECK FOR A DOLLAR SIGN.
842       C
843             IF(ITEST.EQ.1HS) GO TO 1080
844       C
845       C     NO. CHECK FOR A T OR F.
846       C
847             IF(ITEST.EQ.1HT) GO TO 1170
848             IF(ITEST.EQ.1HF) GO TO 1170
849       C
850       C     NO. CHECK FOR AN E.
851       C
852             IF(ITEST.EQ.1HE) GO TO 1100
853       C
854       C     NO. CHECK FOR AN ASTERISK.
855       C
856             IF(ITEST.EQ.1H*) GO TO 680
857       C
858       C     NO. CHECK FOR AN H.
859       C
860             IF(ITEST.EQ.1HH) GO TO 770
861       C
862       C     NO. CHECK FOR A B.
863       C
864             IF(ITEST.EQ.1HB) GO TO 930
865       C
866       C     NO. CHECK FOR AN L OR AN R.
867       C
```

```
868             IF(ITEST.EQ.1HL) GO TO 770
869             IF(ITEST.EQ.1HR) GO TO 770
870       C
871       C     NO. CHECK FOR A D.
872       C
873             IF(ITEST.EQ.1HD) GO TO 1130
874       C
875       C     NO. CHECK FOR A LEFT PARENTHESIS.
876       C
877             IF(ITEST.EQ.1H() GO TO 720
878       C
879       C     NO. CHECK FOR A RIGHT PARENTHESIS.
880       C
881             IF(ITEST.EQ.1H)) GO TO 750
882       C
883       C     LEGAL CHARACTER NOT FOUND. FATAL ERROR.
884       C
885         440 CONTINUE
886             CALL MESSGT(12)
887             IEFLAG=12
888             RETURN
889       C
890       C     ***************************************************************
891       C     CHARACTER WAS A NUMBER
892       C     ***************************************************************
893       C
894         450 CONTINUE
895       C
896       C     SEE IF A DELIMITER WAS EXPECTED. IF SO,FATAL ERROR.
897       C
898             IF(DELIM) GO TO 260
899       C
900       C     RESET ISOLATED PERIOD FLAG
901       C
902             PERIOD=.FALSE.
903       C
904       C     UPDATE THE VALUE OF THE NUMERIC FIELD
905       C
906             IVSUM=IVSUM*10+ITEMP-INF
907       C
908       C     UPDATE THE DIGIT COUNT
909       C
910             IVNUM=IVNUM+1
911       C
912       C     UPDATE THE CHARACTER COUNT
913       C
914             IC=IC+1
915       C
916       C     SEE IF THIS IS THE THIRTIETH CHARACTER
917       C
918             IF(IC.LT.30) GO TO 460
919       C
920       C     YES. FATAL ERROR.
921       C
922             GO TO 1160
923       C
924       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.NOSP+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.NODP+1) GO TO 1160
C
C      ALL D.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
```

```
1039      C     CHECK FOR END OF THE FIELD
1040      C
1041            IF(ICP.GT.IC) GO TO 580
1042      C
1043      C     NO. PICK UP THE NEXT CHARACTER.
1044      C
1045            ITEST=IDSTR(ICP)
1046      C
1047      C     SEE IF IT IS A DECIMAL POINT
1048      C
1049            IF(ITEST.EQ.1H.) GO TO 550
1050      C
1051      C     NO. ISOLATE THE CHARACTER LOW ORDER IN THE WORD.
1052      C
1053            ITEMP=AND(SHIFT(ITEST,-548),77B)
1054      C
1055      C     ADD IT INTO THE CONSTANT
1056      C
1057            DBP=ITEMP-INF
1058            DB=DB*DCONST+DBP
1059      C
1060      C     CONTINUE LOOPING
1061      C
1062            GO TO 540
1063        550 CONTINUE
1064      C
1065      C     A DECIMAL POINT HAS BEEN LOCATED. PREPARE TO EVALUATE THAT PORTION
1066      C         OF THE CONSTANT TO THE RIGHT OF THE DECIMAL. ICPD IS THE NO.
1067      C         OF DIGITS TO THE RIGHT OF THE DECIMAL.
1068      C
1069            DBP=0.
1070            ICPD=0
1071      C
1072      C     LOOP TO EVALUATE PORTION TO THE RIGHT OF THE DECIMAL
1073      C
1074        560 ICP=ICP+1
1075      C
1076      C     CHECK FOR THE END OF THE FIELD
1077      C
1078            IF(ICP.GT.IC) GO TO 570
1079      C
1080      C     NO. PICK UP THE CHARACTER,ADD IT INTO THE SUM,AND CONTINUE.
1081      C
1082            ITEST=IDSTR(ICP)
1083            ITEMP=AND(SHIFT(ITEST,-54),77B)
1084            DTEMP=ITEMP-INF
1085            DBP=DBP*DCONST+DTEMP
1086            ICPD=ICPD+1
1087            GO TO 560
1088        570 CONTINUE
1089      C
1090      C     END OF THE FIELD ENCOUNTERED. SEE IF THERE WAS ANYTHING NON-ZERO
1091      C         TO THE RIGHT OF THE DECIMAL.
1092      C
1093            ITEMPA=DBP-DZERO
1094            IF(ITEMPA.EQ.0) GO TO 580
1095      C
```

```
1096      C     YES, SHIFT THE DECIMAL TO THE LEFT AND ADD IT INTO THE PORTION
1097      C         OF THE ENTIRE CONSTANT TO THE LEFT OF THE DECIMAL,
1098      C
1099            DBP=DBP*DCONST**(-ICPD)
1100            DB=DB+DBP
1101      C
1102      C     NUMBER IS ENTIRELY DECODED, AFFIX THE PROPER SIGN,
1103      C
1104        580 IF(IVSIGN.LT.0) DB=-DB
1105      C
1106      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1107      C
1108      C     GO SEE IF THE NO, IS TO BE STORED
1109      C
1110            GO TO 630
1111        590 CONTINUE
1112      C
1113      C     NO, IS SINGLE PRECISION, SEE IF IT IS ZERO,
1114      C
1115            IF(IVSUM.EQ.0) GO TO 620
1116      C
1117      C     NO, SEE IF IT IS TOO LONG,
1118      C
1119            IF(IVNUM.LE.NDSP) GO TO 610
1120      C
1121      C     YES, FATAL ERROR,
1122      C
1123        600 CONTINUE
1124            CALL MESSOT(28)
1125            IERFLAG=28
1126            RETURN
1127        610 CONTINUE
1128      C
1129      C     NO, STORE IT,
1130      C
1131            XRIGHT=IVSUM
1132      C
1133      C     SHIFT THE DECIMAL POINT OF THE RIGHTHAND SIDE TO THE FAR LEFT
1134      C
1135            XRIGHT=XRIGHT*10.**(-IVNUM)
1136      C
1137      C     ADD THE LEFT AND RIGHT PARTS
1138      C
1139            XLEFT=XLEFT+XRIGHT
1140        620 CONTINUE
1141      C
1142      C     AFFIX THE SIGN
1143      C
1144            IF(IVSIGN.LT.0) XLEFT=-XLEFT
1145      C
1146      C     IF DOUBLE PRECISION,CONVERT TO THAT FORM
1147      C
1148            IF(DOBLE) DB=XLEFT
1149        630 CONTINUE
1150      C
1151      C     DECODING FOR BOTH SINGLE AND DOUBLE PRECISION FINISHED,
1152      C         REINITIALIZE,
```

```
1153      C
1154            IVNUM=0
1155            IV8UM=0
1156            IV8IGN=1
1157      C
1158      C     IF THE SAVE FLAG IS NOT SET,GO BACK AND EXAMINE THE NEXT
1159      C         CHARACTER
1160      C
1161            IF(.NOT.SAVE) GO TO 420
1162      C
1163      C     SEE IF A COMPLEX CONSTANT IS BEING PROCESSED
1164      C
1165            IF(.NOT.CMPLEX) GO TO 670
1166      C
1167      C     YES. SEE IF THIS IS THE FIRST OR SECOND PART.
1168      C
1169            IF(CFRST) GO TO 660
1170      C
1171      C     SECOND PART. SEE IF THE SECOND PART HAS ALREADY BEEN STORED.
1172      C
1173            IF(.NOT.CDONE) GO TO 650
1174      C
1175      C     YES. FATAL ERROR.
1176      C
1177        640 CONTINUE
1178            CALL MESSOT(42)
1179            IEFLAG=42
1180            RETURN
1181      C
1182      C     NO. SAVE THE SECOND PART AND SET THE FLAG.
1183      C
1184        650 CONTINUE
1185            CIMAG=XLEFT
1186            CDONE=.TRUE.
1187            GO TO 670
1188        660 CONTINUE
1189      C
1190      C     FIRST PART. CHECK FOR NO SECOND PART. IF SO,FATAL ERROR.
1191      C
1192            IF(CPFNO) GO TO 640
1193      C
1194      C     SET FLAG TO INDICATE SECOND PART WILL BE PROCESSED NEXT
1195      C
1196            CFRST=.FALSE.
1197      C
1198      C     SAVE THE FIRST PART
1199      C
1200            CREAL=XLEFT
1201      C
1202      C     IF THE LAST CHARACTER FOUND WAS NOT A COMMA,TURN ON THE
1203      C         DELIMITER=EXPECTED FLAG
1204      C
1205            IF(ITEST.NE.1H,) DELIM=.TRUE.
1206      C
1207      C     GO BACK AND BEGIN PROCESSING THE SECOND PART
1208      C
1209            GO TO 410
```

```
1210       670 CONTINUE
1211     C
1212     C     READY TO STORE CONSTANT, IF IT IS NOT DOUBLE PRECISION OR COMPLEX,
1213     C         SHIFT IT INTO STORAGE KNOWN TO THE ROUTINE STORE,
1214     C
1215           IF(.NOT.DOBLE.AND..NOT.CMPLEX) XSUM=XLEFT
1216     C
1217     C     STORE THE CONSTANT
1218     C
1219           CALL STORE(TABLE,IEFLAG)
1220     C
1221     C     IF ANY ERRORS,QUIT RIGHT HERE
1222     C
1223           IF(IEFLAG.NE.0) RETURN
1224     C
1225     C     IF THE CLOSING $ HAS BEEN FOUND,WE ARE ALL DONE
1226     C
1227           IF(DONE) RETURN
1228     C
1229     C     IF A COMPLEX CONSTANT IS BEING PROCESSED AND THE CLOSING
1230     C         PARENTHESIS HAS NOT YET BEEN FOUND,CONTINUE ON TO NEXT
1231     C         CHARACTER
1232     C
1233           IF(CMPLEX.AND..NOT.CPFND) GO TO 420
1234     C
1235     C     IF THE LAST CHARACTER FOUND WAS NOT A COMMA,TURN ON THE
1236     C         DELIMITER FLAG
1237     C
1238           IF(ITEST.NE.1H,) DELIM=.TRUE.
1239     C
1240     C     GO BACK AND HUNT FOR THE BEGINNING OF ANOTHER VARIABLE NAME
1241     C         OR CONSTANT
1242     C
1243           GO TO 90
1244     C
1245     C     ***************************************************************
1246     C     CHARACTER IS AN ASTERISK
1247     C     ***************************************************************
1248     C
1249       680 CONTINUE
1250     C
1251     C     MUST HAVE BEEN PRECEDED BY A FIXED POINT NO, IF NOT,FATAL ERROR,
1252     C
1253           IF(FIX) GO TO 690
1254           CALL MESSOT(17)
1255           IEFLAG=17
1256           RETURN
1257       690 CONTINUE
1258     C
1259     C     SEE IF AN ASTERISK HAS PREVIOUSLY BEEN ENCOUNTERED, IF SO,
1260     C         FATAL ERROR,
1261     C
1262           IF(.NOT.MULT) GO TO 700
1263           CALL MESSOT(18)
1264           IEFLAG=18
1265           RETURN
1266       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 MESSGT(34)
      IEFLAG=34
      RETURN
  780 CONTINUE
C
C     IF NO CHARACTER COUNT HAS BEEN PREVIOUSLY FOUND,FATAL ERROR
C
```

```
1381            IF(IVNUM.NE.0) GO TO 790
1382            CALL MESSOT(35)
1383            IEFLAG=35
1384            RETURN
1385          790 CONTINUE
1386      C
1387      C     IF CHARACTER COUNT IS MORE THAN 10,A MULTIPLIER CANNOT BE USED.
1388      C         IF SO,FATAL ERROR.
1389      C
1390            IF(.NOT.MULT.OR.IVNUM.LE.10) GO TO 800
1391            CALL MESSOT(36)
1392            IEFLAG=36
1393            RETURN
1394          800 CONTINUE
1395      C
1396      C     EVERYTHING CHECKS. TURN ON THE HOLLERITH FLAG.
1397      C
1398            HOLLER=.TRUE.
1399      C
1400      C     THE CHARACTER COUNT IS STORED IN IVNUM AND THE TYPE OF FIELD
1401      C         IN ISAVE (H,L,OR R)
1402      C
1403            IVNUM=IVSUM
1404            ISAVE=ITEST
1405      C
1406      C     SPECIAL PROCESSING IF MORE THAN A WORDS WORTH OF CHARACTERS
1407      C
1408            IF(IVNUM.GT.10) GO TO 900
1409      C
1410      C     TEN OR LESS CHARACTERS TO PROCESS
1411      C
1412          810 CONTINUE
1413      C
1414      C     PICK UP THE REMAINING CHARACTERS AND STORE IN IOSTR
1415      C
1416            DO 820 I=1,IVNUM
1417            CALL NXTCOL(IF,IEFLAG)
1418            IF(IEFLAG.NE.0) RETURN
1419            IOSTR(I)=ITEST
1420          820 CONTINUE
1421      C
1422      C     IF TEN CHARACTERS,NO DISTINCTION BETWEEN H,L,AND R
1423      C
1424            IF(IVNUM.EQ.10) GO TO 860
1425      C
1426      C     CHECK FOR H
1427      C
1428            IF(ISAVE.NE.1HH) GO TO 830
1429      C
1430      C     YES. PAD END OF WORD WITH BLANKS.
1431      C
1432            IPAD=1H
1433            GO TO 840
1434          830 CONTINUE
1435      C
1436      C     NO. CHECK FOR R.
1437      C
```

```
1438            IF(ISAVE.EQ.1HR) GO TO 870
1439      C
1440      C     NO. ASSUME L. PAD END OF WORD WITH ZEROS.
1441      C
1442            IPAD=0
1443          840 CONTINUE
1444            IVP=IVNUM+1
1445            DO 850 I=IVP,10
1446          850 IDSTR(I)=IPAD
1447      C
1448      C     UP CHARACTER COUNT TO FULL WORD
1449      C
1450            IVNUM=10
1451          860 CONTINUE
1452      C
1453      C     TAKE INDIVIDUAL CHARACTERS AND PUT INTO A SINGLE WORD (IVSUM)
1454      C
1455            CALL MASH(IDSTR,IVNUM,IVSUM,IEFLAG)
1456            IF(IEFLAG.NE.0) RETURN
1457      C
1458      C     TURN ON THE DELIMITER FLAG,STORE THE WORD,AND GO ON TO NEXT
1459      C         VARIABLE NAME OR CONSTANT
1460      C
1461            DELIM=.TRUE.
1462            CALL STORE(TABLE,IEFLAG)
1463            IF(IEFLAG.NE.0) RETURN
1464            GO TO 90
1465      C
1466      C     R FIELD. RIGHT JUSTIFY CHARACTERS AND PAD UPPER PART WITH ZEROS.
1467      C
1468          870 CONTINUE
1469            IDIFF=10-IVNUM
1470            II=11
1471            IIP=IVNUM+1
1472            DO 880 I=1,IVNUM
1473            II=II-1
1474            IIP=IIP-1
1475          880 IDSTR(II)=IDSTR(IIP)
1476            DO 890 I=1,IDIFF
1477          890 IDSTR(I)=0
1478            IVNUM=10
1479      C
1480      C     PUT THE CHARACTERS INTO ONE WORD AND STORE IT
1481      C
1482            GO TO 860
1483      C
1484      C     FIELD IS LONGER THAN 10 CHARACTERS
1485      C
1486          900 CONTINUE
1487      C
1488      C     CHECK FOR R FIELD. IF SO,FATAL ERROR.
1489      C
1490            IF(ITEST.NE.1HR) GO TO 910
1491            CALL MESSOT(37)
1492            IEFLAG=37
1493            RETURN
1494          910 CONTINUE
```

```
1495      C
1496      C     PICK UP A WORDS WORTH OF CHARACTERS
1497      C
1498            DO 920 I=1,10
1499            CALL NXTCOL(IF,IEFLAG)
1500            IF(IEFLAG.NE.0) RETURN
1501          920 IDSTR(I)=ITEST
1502      C
1503      C     PUT THEM INTO ONE WORD AND STORE THEM
1504      C
1505            CALL MASH(IDSTR,10,IVSUM,IEFLAG)
1506            IF(IEFLAG.NE.0) RETURN
1507            CALL STORE(TABLE,IEFLAG)
1508            IF(IEFLAG.NE.0) RETURN
1509      C
1510      C     REDUCE THE CHARACTER COUNT BY 10 AND SEE IF IT IS NOW LE.10,
1511      C         IF SO,GO FINISH PROCESSING THE FIELD. IF NOT,GO BACK AND
1512      C         PICK UP THE NEXT 10 CHARACTERS.
1513      C
1514            IVNUM=IVNUM-10
1515            IF(IVNUM.LE.10) GO TO 810
1516            GO TO 910
1517      C
1518      C     ***************************************************************
1519      C     CHARACTER IS A B
1520      C     ***************************************************************
1521      C
1522          930 CONTINUE
1523      C
1524      C     INSURE THAT PREVIOUS FIELD WAS FIXED POINT. IF NOT,FATAL ERROR.
1525      C
1526            IF(FIX) GO TO 940
1527            CALL MESSOT(21)
1528            IEFLAG=21
1529            RETURN
1530          940 CONTINUE
1531      C
1532      C     SEE IF PREVIOUS FIELD WAS ZERO. IF SO,NO CONVERSION NECESSARY.
1533      C
1534            IF(IVNUM.EQ.0) GO TO 970
1535      C
1536      C     SEE IF PREVIOUS FIELD WAS 20 CHAR OR LESS. IF NOT,FATAL ERROR.
1537      C
1538            IF(IVNUM.LE.20) GO TO 950
1539            CALL MESSOT(40)
1540            IEFLAG=40
1541            RETURN
1542          950 CONTINUE
1543      C
1544      C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1545      C     LOOP TO DECODE OCTAL NUMBERS
1546      C     +++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1547      C
1548      C     IVSUM WILL BE THE RESULTING NO.
1549      C     ISHFT IS THE NO. OF BITS TO SHIFT THE CURRENT DIGIT TO THE LEFT
1550      C         BEFORE OR-ING IT INTO IVSUM
1551      C
```

```
1552            IVSUM=0
1553            ISHFT=3*IVNUM
1554            DO 960 I=1,IVNUM
1555      C
1556      C     ISOLATE THE CHARACTER LOW ORDER IN TEMPORARY STORAGE
1557      C
1558            ITEMP=AND(SHIFT(IOSTR(I),-54),77B)
1559      C
1560      C     MAKE SURE IT IS NOT AN 8 OR A 9. IF SO,FATAL ERROR.
1561      C
1562            IF(ITEMP.GT.INF+7) GO TO 980
1563      C
1564      C     CONVERT,SHIFT,AND OR
1565      C
1566            ISHFT=ISHFT-3
1567            ITEMP=SHIFT(ITEMP-INF,ISHFT)
1568            IVSUM=OR(IVSUM,ITEMP)
1569        960 CONTINUE
1570        970 CONTINUE
1571      C
1572      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1573      C
1574      C     TURN ON THE DELIMITER FLAG AND GO STORE THE CONVERTED CONSTANT
1575      C
1576            DELIM=.TRUE.
1577            GO TO 1070
1578        980 CONTINUE
1579      C
1580      C     ILLEGAL CHARACTER IN OCTAL CONSTANT
1581      C
1582            CALL MESSOT(41)
1583            IEFLAG=41
1584            RETURN
1585      C
1586      C     ----------------------------------------------------------------
1587      C     CHARACTER IS + OR -
1588      C     ----------------------------------------------------------------
1589      C
1590        990 CONTINUE
1591      C
1592      C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
1593      C
1594            IF(DELIM) GO TO 260
1595      C
1596      C     IF A SIGN HAS ALREADY BEEN FOUND,FATAL ERROR
1597      C
1598            IF(NOSGN.NE.0) GO TO 1000
1599      C
1600      C     SEE IF FIXED POINT FLAG IS STILL SET
1601      C
1602            IF(.NOT.FIX) GO TO 1020
1603      C
1604      C     YES, SIGN BETTER BE FIRST CHARACTER IN THE FIELD OR FATAL ERROR.
1605      C
1606            IF(IVNUM.EQ.0) GO TO 1010
1607       1000 CONTINUE
1608            CALL MESSOT(22)
```

```
1609            IEFLAG=22
1610            RETURN
1611         1010 CONTINUE
1612       C
1613       C     SET IVSIGN BASED ON + OR -, PLUS IS THE DEFAULT.
1614       C
1615            IF(ITEST.EQ.1H-) IVSIGN=-1
1616       C
1617       C     INDICATE A SIGN HAS BEEN FOUND AND GO LOOK AT THE NEXT
1618       C         CHARACTER
1619       C
1620            NOSGN=1
1621            GO TO 420
1622         1020 CONTINUE
1623       C
1624       C     SIGN FOUND BUT FLOATING POINT NO. BEING PROCESSED, EXPONENT FLAG
1625       C         BETTER BE SET OR FATAL ERROR.
1626       C
1627            IF(FXPN) GO TO 1040
1628         1030 CONTINUE
1629            CALL MESSOT(27)
1630            IEFLAG=27
1631            RETURN
1632         1040 CONTINUE
1633       C
1634       C     EXPONENT FLAG SET, SEE IF ANY CHARACTERS HAVE BEEN ENCOUNTERED
1635       C         AFTER THE E OR D, IF SO,FATAL ERROR.
1636       C
1637            IF(IVNUM.NE.0) GO TO 1030
1638       C
1639       C     ALL O.K. GO SET THE SIGN AND CONTINUE.
1640       C
1641            GO TO 1010
1642       C
1643       C     ------------------------------------------------------------
1644       C     CHARACTER IS A COMMA
1645       C     ------------------------------------------------------------
1646       C
1647         1050 CONTINUE
1648       C
1649       C     SEE IF THE DELIMITER-EXPECTED FLAG IS SET
1650       C
1651            IF(DELIM) GO TO 1090
1652       C
1653       C     NO. SEE IF THE FIXED POINT FLAG IS SET. IF NOT,COMMA SIGNALS
1654       C         TERMINATION OF FLOATING POINT FIELD. GO FINISH DECODING THE
1655       C         NO. AND STORE IT.
1656       C
1657            IF(.NOT.FIX) GO TO 480
1658         1060 CONTINUE
1659       C
1660       C     FIXED POINT FLAG IS SET. CHECK FOR MORE THAN 14 CHARACTERS IN THE
1661       C         FIELD. IF SO,FATAL ERROR.
1662       C
1663            IF(IVNUM.GT.NOSP) GO TO 600
1664       C
1665       C     CHECK FOR NO CHARACTERS FOUND. IF SO,FATAL ERROR.
```

```
1666      C
1667            IF(IVNUM.NE.0) GO TO 1070
1668            CALL MESSOT(30)
1669            IEFLAG=30
1670            RETURN
1671       1070 CONTINUE
1672      C
1673      C     SEE IF COMPLEX FLAG IS SET. IF SO,INTEGER FIELD IN COMPLEX
1674      C         CONSTANT = FATAL ERROR.
1675      C
1676            IF(CMPLEX) GO TO 640
1677      C
1678      C     ALL IS O.K. AFFIX THE SIGN AND STORE THE CONSTANT.
1679      C
1680            IF(IVSIGN.LT.0) IVSUM=-IVSUM
1681            CALL STORE(TABLE,IEFLAG)
1682            IF(IEFLAG.NE.0) RETURN
1683      C
1684      C     IF DOLLAR SIGN WAS LAST CHARACTER ENCOUNTERED,ALL DONE. IF NOT,
1685      C         GO ON AND LOOK FOR BEGINNING OF NEXT CONSTANT OR VARIABLE
1686      C         NAME.
1687      C
1688            IF(DONE) RETURN
1689            GO TO 90
1690      C
1691      C     ----------------------------------------------------------------
1692      C     CHARACTER IS $
1693      C     ----------------------------------------------------------------
1694      C
1695       1080 CONTINUE
1696      C
1697      C     SET THE DONE FLAG. IF FLOATING POINT NO. IS BEING PROCESSED,
1698      C         VIEW $ AS FIELD TERMINATOR AND GO FINISH PROCESSING THE NO.
1699      C         IF FIXED POINT,GO PERFORM CHECKS AND STORE THE CONSTANT.
1700      C
1701            DONE=.TRUE.
1702            IF(.NOT.FIX) GO TO 480
1703            GO TO 1060
1704      C
1705      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1706      C     JUMP TO HERE IF DELIMITER EXPECTED AND FOUND
1707      C     ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
1708      C
1709       1090 CONTINUE
1710      C
1711      C     RESET THE DELIMITER-EXPECTED FLAG
1712      C
1713            DELIM=.FALSE.
1714      C
1715      C     SEE IF COMPLEX CONSTANT IS BEING PROCESSED. IF NOT,ONE CAN ONLY
1716      C         GET TO HERE WHILE PROCESSING A FIXED POINT NO. GO STORE IT.
1717      C
1718            IF(.NOT.CMPLEX) GO TO 1060
1719      C
1720      C     COMPLEX CONSTANT BEING PROCESSED. IF CHARACTERS FOUND BEFORE
1721      C         DELIMITER,FATAL ERROR.
1722      C
```

```
1723            IF(IVNUM.NE.0) GO TO 640
1724      C
1725      C     EVERYTHING O.K. GO ON AND LOOK AT NEXT CHARACTER.
1726      C
1727            GO TO 420
1728      C
1729      C     ***************************************************************
1730      C     CHARACTER IS AN E
1731      C     ***************************************************************
1732      C
1733       1100 CONTINUE
1734      C
1735      C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
1736      C
1737            IF(DELIM) GO TO 260
1738      C
1739      C     SEE IF MORE THAN 15 CHARACTERS HAVE BEEN FOUND IN THE PRECEDING
1740      C         FIELD. IF SO,E SHOULD HAVE BEEN D. FATAL ERROR.
1741      C
1742            IF(IC.GT.NDSP+1) GO TO 600
1743       1110 CONTINUE
1744      C
1745      C     ALL O.K. TURN ON THE EXPONENT FLAG AND RESET THE SIGN COUNT.
1746      C
1747            EXPN=.TRUE.
1748            NOSGN=0
1749      C
1750      C     SEE IF THE FIELD CURRENTLY BEING PROCESSED IS FIXED POINT
1751      C
1752            IF(.NOT.FIX) GO TO 1120
1753      C
1754      C     YES. SEE IF ANY CHARACTERS HAVE PREVIOUSLY BEEN FOUND.
1755      C
1756            IF(IVNUM.NE.0) GO TO 1150
1757      C
1758      C     NO. DEFAULT TO 1. (AND USE A SIGN IF ONE WAS FOUND)
1759      C
1760            IVNUM=1
1761            IVSUM=1
1762            IF(IVSIGN.LT.0) IVSUM=-IVSUM
1763      C
1764      C     RESET THE SIGN
1765      C
1766            IVSIGN=1
1767      C
1768      C     GO FINISH PROCESSING PART TO LEFT OF THE EXPONENT
1769      C
1770            GO TO 1150
1771       1120 CONTINUE
1772      C
1773      C     FLOATING POINT FIELD. GO FINISH PROCESSING PART TO THE LEFT OF
1774      C         THE EXPONENT.
1775      C
1776            SAVE=.FALSE.
1777            GO TO 530
1778      C
1779      C     ***************************************************************
```

```
1780      C     CHARACTER IS A D
1781      C     ****************************************************************
1782      C
1783       1130 CONTINUE
1784      C
1785      C     IF A DELIMITER WAS EXPECTED,FATAL ERROR
1786      C
1787           IF(DELIM) GO TO 260
1788      C
1789      C     SEE IF COMPLEX CONSTANT IS BEING PROCESSED, IF SO,FATAL ERROR,
1790      C
1791           IF(CMPLEX) GO TO 640
1792      C
1793      C     EVERYTHING CHECKS, SET THE DOUBLE PRECISION FLAG AND GO FINISH
1794      C         INITIALIZING FOR THE EXPONENT,
1795      C
1796           DOBLE=.TRUE.
1797           GO TO 1110
1798      C
1799      C     ****************************************************************
1800      C     CHARACTER IS A PERIOD
1801      C     ****************************************************************
1802      C
1803       1140 CONTINUE
1804      C
1805      C     SEE IF DELIMITER WAS EXPECTED, IF SO,FATAL ERROR,
1806      C
1807           IF(DELIM) GO TO 260
1808      C
1809      C     IF FLOATING POINT FLAG IS ALREADY SET,FATAL ERROR
1810      C
1811           IF(FIX) GO TO 1150
1812           CALL MESSOT(25)
1813           IEFLAG=25
1814           RETURN
1815       1150 CONTINUE
1816      C
1817      C     IF MORE THAN 28 CHARACTERS PREVIOUSLY FOUND,FATAL ERROR
1818      C
1819           IF(IVNUM.GT.NODP) GO TO 1160
1820      C
1821      C     IF THE PERIOD IS THE FIRST CHARACTER IN THE FIELD,SET THE
1822      C         ISOLATED PERIOD FLAG
1823      C
1824           IF(IVNUM.EQ.0) PERIOD=.TRUE.
1825      C
1826      C     NO PROBLEMS, STORE THE PART TO THE LEFT OF THE PERIOD AS A
1827      C         FLOATING POINT NO,
1828      C
1829           XLEFT=IVSUM
1830           IVSUM=0
1831      C
1832      C     RESET THE FIELD FLAGS
1833      C
1834           IVNUM=0
1835      C
1836      C     TURN ON THE FLOATING POINT FLAG
```

```
1837      C
1838            FIX=.FALSE.
1839      C
1840      C     STORE THE CHARACTER FOR LATER DOUBLE PRECISION PROCESSING IF
1841      C         NECESSARY
1842      C
1843            IC=IC+1
1844            IOSTR(IC)=1H.
1845      C
1846      C     IF THE DOUBLE PRECISION FLAG IS ON,STORE PART TO LEFT OF DECIMAL
1847      C         IN DOUBLE PRECISION FORM
1848      C
1849            IF(DOBLE) DB=XLEFT
1850      C
1851      C     GO ON AND LOOK AT THE NEXT CHARACTER
1852      C
1853            GO TO 420
1854      C
1855      C     FATAL ERROR, TOO MANY DIGITS IN CONSTANT.
1856      C
1857       1160 CONTINUE
1858            CALL MESSOT(38)
1859            IEFLAG=38
1860            RETURN
1861      C
1862      C     ***************************************************************
1863      C     CHARACTER IS T OR F
1864      C     ***************************************************************
1865      C
1866       1170 CONTINUE
1867      C
1868      C     IF DELIMITER IS EXPECTED,FATAL ERROR
1869      C
1870            IF(DELIM) GO TO 260
1871      C
1872      C     A PERIOD SHOULD HAVE PRECEDED EITHER T OR F, IF NOT,FATAL ERROR.
1873      C
1874            IF(PERIOD) GO TO 1180
1875            CALL MESSOT(31)
1876            IEFLAG=31
1877            RETURN
1878       1180 CONTINUE
1879      C
1880      C     SAVE THE CHARACTER AND GO GET THE NEXT ONE
1881      C
1882            ISAVE=ITEST
1883            CALL NXTCOL(IF,IEFLAG)
1884            IF(IEFLAG.NE.0) RETURN
1885      C
1886      C     SEE IF IT WAS A PERIOD
1887      C
1888            IF(ITEST.NE.1H.) GO TO 1210
1889      C
1890      C     YES, SHORTENED FORM OF LOGICAL CONSTANT BEING USED
1891      C
1892       1190 CONTINUE
1893      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
```

```
1951      C     FALSE BEING PROCESSED, CHECK FOR ALS.
1952      C
1953            IF(ITEST.NE.1HA) GO TO 1220
1954            CALL NXTCOL(IF,IEFLAG)
1955            IF(IEFLAG.NE.0) RETURN
1956            IF(ITEST.NE.1HL) GO TO 1220
1957            CALL NXTCOL(IF,IEFLAG)
1958            IF(IEFLAG.NE.0) RETURN
1959            IF(ITEST.NE.1HS) GO TO 1220
1960      C
1961      C     ALL SUCCESSFULLY FOUND, GO CHECK REST OF CONSTANT.
1962      C
1963            GO TO 1240
1964            END
```

---

```
   1            SUBROUTINE MESSOT(IEFLAG)
   2      C
   3      C     ROUTINE TO DO PRINTING FOR NAMLST
   4      C
   5      C     IEFLAG IS THE ERROR FLAG (SEE NAMLST)
   6      C
   7      C     WRITTEN BY J.L.NORTON,LASL T-3,1974
   8      C
   9            COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
  10           1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
  11            COMMON/IOTAB/NF,IFTAB
  12            COMMON/JOTAB/NFE,IFTABE
  13            INTEGER GETIT,SHIFT,OR,AND,COMP
  14            DIMENSION ILINE(8)
  15            DATA IPT/1R;/
  16      C
  17      C     LOOP OVER ALL OUTPUT FILES
  18      C
  19            IF(IEFLAG.NE.1) GO TO 2
  20            NFP=NF
  21            IFT=IFTAB
  22            GO TO 3
  23          2 CONTINUE
  24            NFP=NFE
  25            IFT=IFTABE
  26          3 CONTINUE
  27            IF(NFP.EQ.0) NFP=1
  28            DO 500 I=1,NFP
  29      C
  30      C     IF NO. OF FILES IS 0,GO DEFAULT TO OUT
  31      C
  32            IF(IFT.NE.0) GO TO 10
  33            IPX=6
  34            GO TO 20
  35         10 CONTINUE
  36      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
```

```
151         310 WRITE(IPX,790)
152             GO TO 490
153         320 WRITE(IPX,800)
154             GO TO 490
155         330 WRITE(IPX,810)
156             GO TO 490
157         340 WRITE(IPX,820)
158             GO TO 490
159         350 WRITE(IPX,830)
160             GO TO 490
161         360 WRITE(IPX,840)
162             GO TO 490
163         370 WRITE(IPX,850)
164             GO TO 490
165         380 WRITE(IPX,860)
166             GO TO 490
167         390 WRITE(IPX,870)
168             GO TO 490
169         400 WRITE(IPX,880)
170             GO TO 490
171         410 WRITE(IPX,890)
172             GO TO 490
173         420 WRITE(IPX,900)
174             GO TO 490
175         430 WRITE(IPX,910)
176             GO TO 490
177         440 WRITE(IPX,920)
178             GO TO 490
179         450 WRITE(IPX,930)
180             GO TO 490
181         460 WRITE(IPX,940)
182             GO TO 490
183         470 WRITE(IPX,950)
184             GO TO 490
185         480 WRITE(IPX,960)
186         490 CONTINUE
187       C
188       C     PRINT ANOTHER ROW OF ASTERISKS
189       C
190             WRITE(IPX,970)
191         500 CONTINUE
192             RETURN
193       C
194         510 FORMAT(1H ,80A1)
195         520 FORMAT(1H ,8A10)
196         530 FORMAT(1H ,41HMORE THAN SIX CHARACTERS IN NAMELIST NAME)
197         540 FORMAT(1H ,45HMORE THAN SIX CHARACTERS IN NAMELIST VARIABLE)
198         550 FORMAT(1H ,39HNAMELIST VARIABLE NOT IN NAMELIST TABLE)
199         560 FORMAT(1H ,50HILLEGAL CHARACTER FOLLOWING NAMELIST VARIABLE NAME)
200         570 FORMAT(1H ,30HILLEGAL CHARACTER IN SUBSCRIPT)
201         580 FORMAT(1H ,18HDELIMITER EXPECTED)
202         590 FORMAT(1H ,20HUNEXPECTED DELIMITER)
203         600 FORMAT(1H ,20H= EXPECTED,NOT FOUND)
204         610 FORMAT(1H ,24HILLEGAL LOGICAL VARIABLE)
205         620 FORMAT(1H ,43HNAMELIST NAME NOT FOUND IMMEDIATELY AFTER $)
206         630 FORMAT(1H ,34HILLEGAL CHARACTER IN THIS POSITION)
207         640 FORMAT(1H ,44HFIRST CHARACTER OF NAMELIST VARIABLE ILLEGAL)
```

```
208        650 FORMAT(1H ,19HTOO MANY SUBSCRIPTS)
209        660 FORMAT(1H ,49HATTEMPT TO USE UNDIMENSIONED VARIABLE AS AN ARRAY)
210        670 FORMAT(1H ,37HTOO MANY SUBSCRIPTS FOR THIS VARIABLE)
211        680 FORMAT(1H ,28HMULTIPLIER IS NOT AN INTEGER)
212        690 FORMAT(1H ,24HMORE THAN ONE MULTIPLIER)
213        700 FORMAT(1H ,34HMULTIPLIER WITHIN COMPLEX CONSTANT)
214        710 FORMAT(1H ,25HIMPROPER LEFT PARENTHESIS)
215        720 FORMAT(1H ,23HIMPROPER OCTAL CONSTANT)
216        730 FORMAT(1H ,31HMORE THAN TWO SIGNS IN CONSTANT)
217        740 FORMAT(1H ,40HFIRST CHARACTER OF NAMELIST NAME ILLEGAL)
218        750 FORMAT(1H ,34HILLEGAL CHARACTER IN NAMELIST NAME)
219        760 FORMAT(1H ,48HMORE THAN ONE DECIMAL POINT IN A FLOATING POINT ,
220           1 8HCONSTANT)
221        770 FORMAT(1H ,21HEXPONENT OUT OF RANGE)
222        780 FORMAT(1H ,35HIMPROPER SIGN IN FLOATING POINT NO.)
223        790 FORMAT(1H ,48HTOO MANY CHARACTERS IN SINGLE PRECISION CONSTANT)
224        800 FORMAT(1H ,15HISOLATED PERIOD)
225        810 FORMAT(1H ,19HZERO WIDTH CONSTANT)
226        820 FORMAT(1H ,34HILLEGAL CHARACTER FOLLOWING PERIOD)
227        830 FORMAT(1H ,14HUNEXPECTED EOF)
228        840 FORMAT(1H ,27HUNEXPECTED CARD TERMINATION)
229        850 FORMAT(1H ,27HILLEGAL H,L,OR R OCCURRENCE)
230        860 FORMAT(1H ,23HHOLLERITH COUNT MISSING)
231        870 FORMAT(1H ,50HMULTIPLIER NOT ALLOWED FOR HOLLERITH CONSTANTS OF ,
232           1 23HMORE THAN 10 CHARACTERS)
233        880 FORMAT(1H ,44HR FIELD CANNOT BE GREATER THAN 10 CHARACTERS)
234        890 FORMAT(1H ,17HCONSTANT TOO LONG)
235        900 FORMAT(1H ,36HSIGN CANNOT PRECEDE COMPLEX CONSTANT)
236        910 FORMAT(1H ,33HTOO MANY DIGITS IN OCTAL CONSTANT)
237        920 FORMAT(1H ,50HTHE DIGIT 8 OR 9 CANNOT OCCUR IN AN OCTAL CONSTANT)
238        930 FORMAT(1H ,25HIMPROPER COMPLEX CONSTANT)
239        940 FORMAT(1H ,19HSUBSCRIPT TOO LARGE)
240        950 FORMAT(1H ,41HMORE THAN ONE SIGN IN AN INTEGER CONSTANT)
241        960 FORMAT(1H ,13HERROR IN MASH)
242        970 FORMAT(1H ,60(1H*))
243            END
```

**********

```
  1            SUBROUTINE STORE(TABLE,IEFLAG)
  2      C
  3      C     ROUTINE TO STORE THE CONSTANT LAST DECODED
  4      C
  5      C     WRITTEN BY J.L.NORTON,LASL T=3,1974
  6      C
  7            COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
  8           1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
  9            LOGICAL MULT,CMPLEX,DIM,DOBLE
 10            DIMENSION TABLE(3,1)
 11            INTEGER TABLE,SHIFT,AND,OR,COMP
 12            DATA CREAL/0./
 13      C
 14      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
```

```
55      C
56            IF(ISUB.LE.TABLE(2,1)) GO TO 10
57      C
58      C     NO. FATAL ERROR.
59      C
60            IEFLAG=5
61            GO TO 999
62         10 CONTINUE
63      C
64      C     SET UP TABLE ENTRY
65      C
66            TABLE(1,ISUB)=NAME
67            TABLE(2,ISUB)=LOCF(LOCA)
68            TABLE(3,ISUB)=0
69      C
70      C     COMPOSE SUBSCRIPT WORD IF VARIABLE IS AN ARRAY
71      C
72            IF(NSUB.EQ.0) GO TO 30
73      C
74      C     SET THE TABLE BASED UPON HOW MANY SUBSCRIPTS THE ARRAY HAS
75      C
76            IF(NSUB.GT.1) GO TO 20
77      C
78      C     ARRAY HAS ONLY ONE SUBSCRIPT. DENOTE THIS BY SETTING SUBSCRIPT
79      C         FIELD 1 TO 1 AND LEAVING ALL THE REST ZERO.
80      C
81            TABLE(3,ISUB)=1
82            GO TO 30
83      C
84      C     ARRAY HAS MORE THAN ONE SUBSCRIPT
85      C
86         20 CONTINUE
87      C
88      C     SEE IF AT LEAST THE FIRST SUBSCRIPT IS SPECIFIED. IF NOT,ERROR.
89      C
90            IF(NSUB1.LE.0) GO TO 40
91      C
92      C     FIRST SUBSCRIPT PRESENT. STORE IT AND SEE IF ARRAY HAS MORE THAN
93      C         TWO SUBSCRIPTS.
94      C
95            TABLE(3,ISUB)=NSUB1
96            IF(NSUB.EQ.2) GO TO 30
97      C
98      C     YES. THE SECOND SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR.
99      C
100           IF(NSUB2.LE.0) GO TO 50
101     C
102     C     SECOND SUBSCRIPT PRESENT. STORE IT AND SEE IF ARRAY HAS MORE THAN
103     C         THREE SUBSCRIPTS.
104     C
105           TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB2,18))
106           IF(NSUB.EQ.3) GO TO 30
107     C
108     C     YES. THE THIRD SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR.
109     C
110           IF(NSUB3.LE.0) GO TO 60
111     C
```

```
112      C     THIRD SUBSCRIPT PRESENT, STORE IT AND MAKE SURE THE TOTAL NO.
113      C         OF SUBSCRIPTS DOES NOT EXCEED FOUR.
114      C
115           TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB3,36))
116           IF(NSUB.NE.4) GO TO 70
117        30 CONTINUE
118      C
119      C     ALL DONE. INCREMENT THE NO. OF ENTRIES AND RETURN.
120      C
121           TABLE(3,1)=NENTRY+1
122           RETURN
123      C
124      C     ERROR - MORE THAN ONE SUBSCRIPT INDICATED BUT NSUB1 NOT GIVEN OR
125      C               IN ERROR (ZERO OR NEGATIVE)
126      C
127        40 IEFLAG=IEFLAG+1
128      C
129      C     ERROR - MORE THAN TWO SUBSCRIPTS INDICATED BUT NSUB2 NOT GIVEN OR
130      C               IN ERROR
131      C
132        50 IEFLAG=IEFLAG+1
133      C
134      C     ERROR - MORE THAN THREE SUBSCRIPTS INDICATED BUT NSUB3 NOT GIVEN
135      C               OR IN ERROR
136      C
137        60 IEFLAG=IEFLAG+1
138      C
139      C     ERROR - MORE THAN FOUR SUBSCRIPTS INDICATED
140      C
141        70 IEFLAG=IEFLAG+1
142       999 CONTINUE
143           IEFLAG=OR(IEFLAG,AND(NAME,777777777770000000B))
144           GO TO IERRT
145           END
```

---

```
 1           SUBROUTINE NXTCOL(IF,IEFLAG)
 2      C
 3      C     ROUTINE TO GET THE NEXT CHARACTER FROM THE CURRENT NAMELIST CARD
 4      C         AND,IF NECESSARY,READ ANOTHER CARD FROM FILE IF
 5      C
 6      C     WRITTEN BY J.L.NORTON,LASL T-3,1974
 7      C
 8           COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
 9          1 CREAL,CIMAG,CMPLEX,DIM,MULTSV,IV$UM,MULT,LIST,HOLLER,DOBLE
10           LOGICAL LIST,HOLLER
11      C
12      C     INCREMENT THE COLUMN NO.
13      C
14           ISUB=ISUB+1
15      C
16      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,DR
      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,2HOF,2HID,2HEQ,2HNE,2HGE,
     2 2HLT,2HBX,2HLX,2HAX,2HMX,2HNX,2HZX,2HUX,2HPX,2HFX,2HOX,2HRX,2HIX,
     3 2HNO,2HCX,2H8A,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)
```

```
128           IOPRN(IOPNO,IPART)=IONAME(ID+1)
129           IOPNO=IOPNO+1
130         150 CONTINUE
131       C
132       C     FINISH PARCEL
133       C
134         160 CONTINUE
135           ICODE(IPART)=IOP(ITYPE)
136           IF(ION.GE.20) GO TO 170
137           IPART=IPART+1
138           ION=ION+1
139           IF(ION.GT.16) GO TO 90
140           GO TO 60
141       C
142       C     WORD IS FINISHED. PRINT THE LINE.
143       C
144         170 CONTINUE
145           WRITE(IFILE,770) FWAP,IWORD,(ICODE(I),IREG(I),(IOPRN(J,I),J=1,13),
146          1 I=1,IPART)
147       C
148       C     SEE IF THIS WAS THE LAST WORD
149       C
150         180 CONTINUE
151           IF(FWAP.GE.LWA) RETURN
152       C
153       C     NO. CONTINUE.
154       C
155           FWAP=FWAP+1
156           GO TO 10
157       C
158       C     FIRST TWO DIGITS OF OP ARE 01
159       C
160         190 CONTINUE
161           ION=ION+1
162           IGOT=IDIGIT(ION)+1
163           GO TO (200,210,210,230,240,240,250,310),IGOT
164       C
165       C     OP IS RJ (010)
166       C
167         200 CONTINUE
168           ITYPE=12
169           ION=ION+1
170           GO TO 110
171       C
172       C     OP IS 011 OR 012
173       C
174         210 CONTINUE
175           ITYPE=IGOT
176       C
177       C     PROCESS INSTRUCTION OF THE FORM   OP  BJ+K
178       C
179         220 CONTINUE
180           IF(ION.GT.13) GO TO 90
181           ION=ION+1
182           IOPRN(1,IPART)=1HB
183           IOPRN(2,IPART)=IHEQ(ION)
184           IOPNO=3
```

```
185           GO TO 110
186     C
187     C     OP IS MJ
188     C
189       230 CONTINUE
190           ITYPE=4
191           IF(IDIGIT(ION+1).NE.0.OR.IDIGIT(ION+2).NE.0) GO TO 220
192           IF(IDN.GT.13) GO TO 90
193           IDN=IDN+7
194           GO TO 160
195     C
196     C     OP IS 014 OR 015
197     C
198       240 CONTINUE
199           ITYPE=IGOT
200     C
201     C     PROCESS INSTRUCTION OF THE FORM   OPJ   XK
202     C
203           ION=IDN+1
204           IREG(IPART)=IHEQ(IDN)
205           IDN=IDN+1
206           IOPRN(1,IPART)=1HX
207           IOPRN(2,IPART)=IHEQ(IDN)
208           GO TO 160
209     C
210     C     OP IS 016
211     C
212       250 CONTINUE
213           IF(IDIGIT(ION+1).NE.0) GO TO 280
214           ITYPE=7
215     C
216     C     PROCESS INSTRUCTION OF THE FORM  OP   BK
217       260 CONTINUE
218           IDN=IDN+2
219       270 CONTINUE
220           IOPRN(1,IPART)=1HB
221           IOPRN(2,IPART)=IHEQ(ION)
222           GO TO 160
223       280 CONTINUE
224           IF(IDIGIT(ION+2).NE.0) GO TO 290
225           ITYPE=8
226     C
227     C     PROCESS INSTRUCTION OF THE FORM   OPJ
228     C
229           ION=IDN+1
230           IREG(IPART)=IHEQ(ION)
231           IDN=IDN+1
232           GO TO 160
233       290 CONTINUE
234           ITYPE=9
235     C
236     C     PROCESS INSTRUCTION OF THE FORM  OPJ   BK
237     C
238       300 CONTINUE
239           ION=IDN+1
240           IREG(IPART)=IHEQ(ION)
241           IDN=IDN+1
```

```
242           GO TO 270
243     C
244     C     OP IS 017
245     C
246       310 CONTINUE
247           IF(IDIGIT(ION+1).NE.0) GO TO 320
248           ITYPE=10
249           GO TO 260
250       320 CONTINUE
251           ITYPE=11
252           GO TO 300
253     C
254     C     OP IS JP (02)
255     C
256       330 CONTINUE
257           ITYPE=13
258     C
259     C     PROCESS INSTRUCTION OF THE FORM  OP   BJ+K
260     C
261           IF(ION.GT.12) GO TO 90
262           ION=ION+1
263           IOPRN(1,IPART)=1HB
264           IOPRN(2,IPART)=IHEQ(ION)
265           ION=ION+1
266           IOPNO=3
267           GO TO 110
268     C
269     C     FIRST TWO DIGITS OF OP ARE 03
270     C
271       340 CONTINUE
272           IF(ION.GT.12) GO TO 90
273           ION=ION+1
274           IGOT=IDIGIT(ION)
275           ITYPE=IGOT+14
276     C
277     C     PROCESS INSTRUCTION OF THE FORM  OP   XJ,K
278     C
279           ION=ION+1
280           IOPRN(1,IPART)=1HX
281           IOPRN(2,IPART)=IHEQ(ION)
282           IOPRN(3,IPART)=1H,
283           IOPNO=4
284           GO TO 110
285     C
286     C     FIRST TWO DIGITS OF OP ARE 04,05,06,OR 07
287     C
288       350 CONTINUE
289           IF(ION.GT.12) GO TO 90
290           ITYPE=IGOT+17
291     C
292     C     PROCESS INSTRUCTION OF THE FORM  OP   BI,BJ,K
293     C
294           ION=ION+1
295           IOPRN(1,IPART)=1HB
296           IOPRN(2,IPART)=IHEQ(ION)
297           IOPRN(3,IPART)=1H,
298           IOPRN(4,IPART)=1HB
```

```
299            ION=ION+1
300            IOPRN(5,IPART)=IHEQ(ION)
301            IOPRN(6,IPART)=1H,
302            IOPNO=7
303            GO TO 110
304      C
305      C     FIRST DIGIT OF OP IS 1.  OP IS BXI,
306      C
307        360 CONTINUE
308            ITYPE=26
309            ION=ION+1
310            IGOT=IOIGIT(ION)+1
311            GO TO (370,370,370,370,430,430,430,430),IGOT
312      C
313      C     PROCESS INSTRUCTION OF THE FORM  OPI    XJ
314      C                                             XJ*XK
315      C                                             XJ+XK
316      C                                             XJ-XK
317      C                                             XJ/XK
318      C                                            (IGOT=1,2,3,4,OR 5)
319      C
320        370 CONTINUE
321            ION=ION+1
322            IREG(IPART)=IHEQ(ION)
323            ION=ION+1
324            IOPRN(1,IPART)=1HX
325            IOPRN(2,IPART)=IHEQ(ION)
326            ION=ION+1
327            IF(IGOT.EQ.1) GO TO 160
328            KREG=IHEQ(ION)
329            IOPNO=3
330            IGOT=IGOT-1
331            GO TO (380,400,410,420),IGOT
332      C
333      C     ADD ON *XK
334      C
335        380 CONTINUE
336            IOPRN(IOPNO,IPART)=1H*
337        390 CONTINUE
338            IOPRN(IOPNO+1,IPART)=1HX
339            IOPRN(IOPNO+2,IPART)=KREG
340            GO TO 160
341      C
342      C     ADD ON +XK
343      C
344        400 CONTINUE
345            IOPRN(IOPNO,IPART)=1H+
346            GO TO 390
347      C
348      C     ADD ON -XK
349      C
350        410 CONTINUE
351            IOPRN(IOPNO,IPART)=1H-
352            GO TO 390
353      C
354      C     ADD ON /XK
355      C
```

```
356         420 CONTINUE
357             IOPRN(IOPNO,IPART)=1H/
358             GO TO 390
359       C
360       C     PROCESS INSTRUCTION OF THE FORM  OPI    =XK
361       C                                             =XK*XJ
362       C                                             =XK+XJ
363       C                                             =XK-XJ
364       C                                            (IGDT=5,6,7,OR 8)
365       C
366         430 CONTINUE
367             ION=ION+1
368             IREG(IPART)=IHEQ(ION)
369             ION=ION+1
370             KREG=IHEQ(ION)
371             IOPRN(1,IPART)=1H=
372             IOPRN(2,IPART)=1HX
373             ION=ION+1
374             IOPRN(3,IPART)=IHEQ(ION)
375             IF(IGDT.EQ.5) GO TO 160
376             IGOT=IGOT-5
377             IOPNO=4
378             GO TO (380,400,410),IGOT
379       C
380       C     FIRST DIGIT OF OP IS 2
381       C
382         440 CONTINUE
383             ION=ION+1
384             IGOT=IOIGIT(ION)+1
385             GO TO (450,480,450,480,490,490,490,490),IGOT
386       C
387       C     OP IS LXI
388       C
389         450 CONTINUE
390             ITYPE=27
391             IF(IGOT.EQ.1) GO TO 470
392       C
393       C     PROCESS INSTRUCTION OF THE FORM  OPI    BJ,XK
394       C
395         460 CONTINUE
396             ION=ION+1
397             IREG(IPART)=IHEQ(ION)
398             ION=ION+1
399             IOPRN(1,IPART)=1HB
400             IOPRN(2,IPART)=IHEQ(ION)
401             ION=ION+1
402             IOPRN(3,IPART)=1H,
403             IOPRN(4,IPART)=1HX
404             IOPRN(5,IPART)=IHEQ(ION)
405             GO TO 160
406       C
407       C     PROCESS INSTRUCTION OF THE FORM  OPI    JK
408       C
409         470 CONTINUE
410             ION=ION+1
411             IREG(IPART)=IHEQ(ION)
412             ION=ION+1
```

```
413             IOPRN(1,IPART)=IHEQ(ION)
414             ION=ION+1
415             IOPRN(2,IPART)=IHEQ(ION)
416             IOPRN(3,IPART)=1HB
417             GO TO 160
418       C
419       C     OP IS AXI
420       C
421         480 CONTINUE
422             ITYPE=28
423             IF(IGOT.EQ.2) GO TO 470
424             GO TO 460
425       C
426       C     OP IS NXI,ZXI,UXI,OR PXI
427       C
428         490 CONTINUE
429             ITYPE=IGOT+25
430             GO TO 460
431       C
432       C     FIRST DIGIT OF OP IS 3
433       C
434         500 CONTINUE
435             ION=ION+1
436             IGOTP=IDIGIT(ION)
437             ITYPE=IGOTP/2+34
438             IF(MOD(IGOTP,2).NE.0) GO TO 510
439             IGOT=3
440             GO TO 370
441         510 CONTINUE
442             IGOT=4
443             GO TO 370
444       C
445       C     FIRST DIGIT OF OP IS 4
446       C
447         520 CONTINUE
448             ION=ION+1
449             IGOT=IDIGIT(ION)+1
450             GO TO (530,540,550,570,580,580,590,570),IGOT
451       C
452       C     OP IS FLOATING MULTIPLY
453       C
454         530 CONTINUE
455             ITYPE=34
456             GO TO 560
457         540 CONTINUE
458             ITYPE=36
459             GO TO 560
460         550 CONTINUE
461             ITYPE=35
462         560 CONTINUE
463             IGOT=2
464             GO TO 370
465       C
466       C     OP IS MXI OR CXI
467       C
468         570 CONTINUE
469             ITYPE=29
```

```
470           IF(IGOT.EQ.7) ITYPE=39
471           GO TO 470
472     C
473     C     OP IS FLOATING DIVIDE
474     C
475       580 CONTINUE
476           ITYPE=34
477           IF(IGOT.EQ.5) ITYPE=36
478           IGOT=5
479           GO TO 370
480     C
481     C     OP IS NO
482     C
483       590 CONTINUE
484           ITYPE=38
485     C
486     C     PROCESS 15 BIT INSTRUCTION OF THE FORM  OP   N
487     C
488       600 CONTINUE
489           IDN=IDN+1
490           IF(IDIGIT(IDN).NE.0.OR.IDIGIT(ION+1).NE.0.OR.IDIGIT(ION+2).NE.0)
491          1 GO TO 610
492           IDN=IDN+2
493           GO TO 160
494       610 CONTINUE
495           IOPRN(1,IPART)=IHEQ(ION)
496           IOPRN(2,IPART)=IHEQ(IDN+1)
497           IOPRN(3,IPART)=IHEQ(IDN+2)
498           IOPRN(4,IPART)=1HB
499           IDN=ION+2
500           GO TO 160
501     C
502     C     OP BEGINS WITH 5,6,OR 7
503     C
504       620 CONTINUE
505           ITYPE=IGOT+34
506           ION=ION+1
507           IGOT=IDIGIT(ION)+1
508           IF(IGOT.LT.4.AND.IDN.GT.12) GO TO 90
509           ION=ION+1
510           REG(IPART)=IHEQ(IDN)
511           IDN=IDN+1
512           JREG=IHEQ(IDN)
513           IF(IGOT.LT.4) GO TO 630
514           IDN=IDN+1
515           KREG=IHEQ(IDN)
516       630 CONTINUE
517           GO TO (640,660,670,680,700,710,730,750),IGOT
518       640 CONTINUE
519           IOPRN(1,IPART)=1HA
520       650 CONTINUE
521           IOPRN(2,IPART)=JREG
522           IOPNO=3
523           GO TO 110
524       660 CONTINUE
525           IOPRN(1,IPART)=1HB
526           GO TO 650
```

```
527         670 CONTINUE
528             IOPRN(1,IPART)=1HX
529             GO TO 650
530         680 CONTINUE
531             IOPRN(1,IPART)=1HX
532             IOPRN(3,IPART)=1H+
533         690 CONTINUE
534             IOPRN(2,IPART)=JREG
535             IOPRN(4,IPART)=1HB
536             IOPRN(5,IPART)=KREG
537             GO TO 160
538         700 CONTINUE
539             IOPRN(3,IPART)=1H+
540             GO TO 720
541         710 CONTINUE
542             IOPRN(3,IPART)=1H-
543         720 CONTINUE
544             IOPRN(1,IPART)=1HA
545             GO TO 690
546         730 CONTINUE
547             IOPRN(3,IPART)=1H+
548         740 CONTINUE
549             IOPRN(1,IPART)=1HB
550             GO TO 690
551         750 CONTINUE
552             IOPRN(3,IPART)=1H-
553             GO TO 740
554       C
555         760 FORMAT(1H ,06,3X020,3X8(1H-),
556            1 54HWORD CONTAINS ILLEGAL INSTRUCTION(S), MUST BE DATA OR ,
557            2 10HCLOBBERED.,8(1H-))
558         770 FORMAT(1H ,06,3X020,4(3XA2,A1,3X13A1))
559             END
```

---

```
  1             SUBROUTINE PABORT
  2       C
  3       C     ROUTINE TO HANDLE ABORT EXCHANGE PACKAGE DUMPS
  4       C
  5       C     WRITTEN BY J.L.NORTON,LASL T=3,1974
  6       C
  7             INTEGER SHIFT
  8             LOGICAL FILM
  9             COMMON/YSCS/RESTRT,FILM,PAPER,IPD,IFD
 10             COMMON/IEOMP/IDUMP(16)
 11       C
 12       C     PRINT CP TIME AT ABORT SO A TIME LIMIT ABORT IS EASILY
 13       C         DISTINGUISHABLE
 14       C
 15             CALL SECOND(TJLN)
 16             DO 10 IPX=6,IFD,6
 17          10 WRITE(IPX,30) TJLN
```

```
18      C
19      C     PRINT THE HARDWARE EXCHANGE PACKAGE
20      C
21            DO 20 IPX=6,IFD,6
22         20 WRITE(IPX,40)
23            CALL OMPPK(6,IDUMP)
24      C
25      C     SEND IT TO FILM IF FILM IS .TRUE.
26      C
27            IF(FILM) CALL OMPPK(12,IDUMP)
28            ENTRY PDMPPK
29      C
30      C     GET THE P COUNTER FOR THE ABORT
31      C
32            IFW=SHIFT(IDUMP,-36)
33      C
34      C     SUBTRACT 32 (40 OCTAL) FROM IT. IF RESULT IS NEGATIVE,SET IT
35      C         TO ZERO.
36      C
37            IFW=IFW-32
38            IF(IFW.LT.0) IFW=0
39      C
40      C     ADD 37 OCTAL (IFW=40B+77B) TO P COUNTER. IF RESULT IS OUT OF
41      C         RANGE,SET IT TO 77B.
42      C
43            IFWB=IFW+77B
44            IF(IFW.GT.150077B) IFW=0
45            IF(IFWB.GT.150077B) IFWB=77B
46      C
47      C     PRINT OP CODE DUMP AROUND ABORT ADDRESS
48      C
49            CALL OPDMP(IFW,IFWB,6)
50      C
51      C     SEND OP CODE DUMP TO FILM IF FILM IS .TRUE.
52      C
53            IF(FILM) CALL OPDMP(IFW,IFWB,12)
54            RETURN
55      C
56         30 FORMAT(1H1,21HCP TIME AT ABORT WAS ,F9.3)
57         40 FORMAT(//1H ,10(1H+),25HHAROWARO EXCHANGE PACKAGE,10(1H+))
58            END
```

---

```
 1            SUBROUTINE PARPLT
 2      C
 3      C     RDUTINE TO PLOT PARTICLES
 4      C
 5      C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
 6      C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
 7      C
 8      *     ***** BEGIN COMDECK PARAM      *****
 9            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),OLSROI(1),OLSROQ(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),OKLSM(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 IOTO,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,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,OTSV,OTLAST,FIYBO,IYBO,YCNVLD,
     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,OTR,TMASS,OTVSAV,OTCSAV,IDTV
     5 ,JOTV,IOTC,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,IPO,IPD
*      ***** 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,OKLSM,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),OLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),OLSROQ,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(IFO,100) PXR,PYB,PYT
      WRITE(IFO,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 ECRO(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
      FIXL=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,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,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,ROT,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,FIYBD,IYBD,YCNVLD,
     3 XCNVLD,FIXRD,FIXLD,IXRD,IXLD,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     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 YADIM      -----
            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 YADIM      -----
      *     ----- BEGIN COMDECK YASC      -----
            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,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,ROT,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,YCNVLO,
           3 XCNVLO,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 YASC      -----
      *     ----- BEGIN COMDECK YAEQ      -----
            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/,NF/0/
       INTEGER COMP,AND,OR
C
C      SET UP THE NAMFLIST INPUT TABLE
C
       ASSIGN 170 TO IERPT
       CALL TABDEF(PARTN,5HPARTN,7,IERRT)
       CALL TABSET(PARTN,5HORPAR,ORPAR,IEFLAG,0,0,0,0,0)
       CALL TABSET(PARTN,5HOZPAR,OZPAR,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,IPD,6
   20 WRITE(IPX,180) DRPAR,DZPAR,XC,YC,XD,YD
```

```
C
C     SEE IF PARTICLE REGION IS RECTANGULAR OR CIRCULAR
C
      IF(YO.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=YO
      PBOTM=YC
      PRITE=XO
      PLEFT=XC
      GO TO 50
C
C     REGION IS CIRCULAR
C
   40 CONTINUE
      PTOP=YC+XO
      PBOTM=YC-XO
      PRITE=XO
      PLEFT=0.
      R2=XO**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
```

```
230            IF(YD.NE.0.) GO TO 80
231            IF((YTE-YC)**2+XTE**2.GT.R2) GO TO 100
232         80 CONTINUE
233      C
234      C     STORE THE PARTICLE COORDINATES AND THE PARTICLE MASS
235      C
236            XPAR(KP)=XTE
237            YPAR(KP)=YTE
238            MPAR(KP)=0.
239      C
240      C     INCREMENT THE SCM BUFFER SUBSCRIPT AND THE PARTICLE COUNT
241      C
242            KP=KP+3
243            NPT=NPT+1
244      C
245      C     SEE IF THE SCM BUFFER NEEDS TO BE FLUSHED
246      C
247            IF(KP.GT.LPB) GO TO 110
248      C
249      C     NO. INCREMENT THE PARTICLE X-COORDINATE.
250      C
251         90 XTE=XTE+DRPAR
252      C
253      C     SEE IF WE HAVE GONE OUTSIDE OF THE PARTICLE REGION
254      C
255            IF(XTE.LE.PRITE) GO TO 70
256      C
257      C     YES. INCREMENT THE PARTICLE Y-COORDINATE.
258      C
259        100 YTE=YTE+DZPAR
260      C
261      C     SEE IF Y HAS GONE OUTSIDE THE PARTICLE REGION
262      C
263            IF(YTE.LE.PTOP) GO TO 60
264      C
265      C     YES. WE ARE DONE WITH THIS PARTICLE REGION. GO READ IN ANOTHER
266      C         CARD.
267      C
268            GO TO 10
269      C
270      C     FLUSH THE BUFFER TO LCM IF THERE IS ROOM
271      C
272        110 CONTINUE
273            IF(IECP+LPB.LE.NLCP2+NLCP1) GO TO 130
274        120 CALL UNCLE(4,6HPRTGEN,25,25HPARTICLE ARRAY OVERFLOWED)
275        130 CONTINUE
276            CALL ECWR(AASC,NLCP1+IECP-1,LPB,IDUM)
277      C
278      C     INCREMENT THE LCM ADDRESS,RESET THE SCM SUBSCRIPT,AND PROCEED
279      C
280            IECP=IECP+LPB
281            KP=1
282            GO TO 90
283      C
284      C     ALL DONE. FLUSH THE SCM BUFFER TO LCM IF THERE IS ROOM.
285      C
286        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,IOUM)
  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=F12.5,5H  XC=E12.5/5H  YC=E12.5
     1 ,5H  XO=E12.5,5H  YO=E12.5)
  190 FORMAT(4X16,38H 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),OLSROI(1),OLSROQ(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),OKLSM(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,ROT,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,OTSV,OTLAST,FIYBO,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,OTR,TMASS,OTVSAV,OTCSAV,IDTV
      5 ,JDTV,IDTC,JOTC,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,IPO,IFO
*      ----- 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),GRRDZ),(AASC(23),DLSRQI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),DLSRQQ,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 IDUM/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
      PROTM=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
```

```
143           XLA=Y41*X1Z-Y1Z*X41
144           IF(XMU.NE.0.) GO TO 80
145           THE=-XLA/XNU
146           IF(X41.NE.0.) GO TO 90
147           ETA=-Y1Z/Y41
148           GO TO 100
149        80 THE=(-XNU+SQRT(XNU*XNU-4.*XMU*XLA))/(2.*XMU)
150        90 ETA=-(X1Z+X21*THE)/(X41+X3421*THE)
151       100 OME=1.-ETA
152           OMT=1.-THE
153           UK=OME*OMT*U1+OME*THE*U2+OMT*ETA*U4+ETA*THE*U3
154           VK=OME*OMT*V1+OME*THE*V2+OMT*ETA*V4+ETA*THE*V3
155           IF(MPTST.EQ.1) GO TO 110
156           UKSV=UK
157           VKSV=VK
158           XTESV=XTE
159           YTESV=YTE
160           XTE=XTE+DT*UK
161           YTE=YTE+DT*VK
162           MPTST=1
163           GO TO 60
164       110 UK=0.5*(UK+UKSV)
165           VK=0.5*(VK+VKSV)
166           XTE=XTESV
167           YTE=YTESV
168           XTEN=XTE+DT*UK
169           YTEN=YTE+DT*VK
170           IF(TURB.AND.T.GT.TSTRTD) CALL PTRBOF(XTEN,YTEN)
171           XPAR(KP)=XTEN
172           YPAR(KP)=YTEN
173     C
174     C     SEE IF TIME-DEPENDENT PARTICLE DATA IS TO BE COLLECTED THIS CYCLE
175     C
176           IF(IFLAGP.EQ.0) GO TO 120
177     C
178     C     YES, SEE IF THE CURRENT PARTICLE IS TO BE SAVED.
179     C
180           IF(MOD(NPPT,NIST).NE.0) GO TO 120
181     C
182     C     YES, SAVE IT.
183     C
184           NN=NPPT/NIST
185           IBEGIN=NLCP1+NLCP2+2*(NP1*(NILNG-1)+NN-1)
186     C     PAX(NN,NILNG)=XTEN
187     C     PAY(NN,NILNG)=YTEN
188           CALL ECWR(XTEN,IBEGIN,2,IOUM)
189       120 CONTINUE
190           PBOTM=AMIN1(YTEN,PBOTM)
191           PRITE=AMAX1(XTEN,PRITE)
192           PTOP=AMAX1(YTEN,PTOP)
193       130 IF(NPPT.EQ.NPT) GO TO 140
194           KP=KP+3
195           IF(KP.LT.LPB) GO TO 40
196           CALL ECWR(AASC,NLCP1+IECP-1,LPB,IOUM)
197           IECP=IECP+LPB
198           GO TO 30
199       140 CALL ECWR(AASC,NLCP1+IECP-1,LPB,IOUM)
```

```
200           IF(NCYC.EQ.1) PTPOLD=PTOP
201           RETURN
202       150 CONTINUE
203           IF(IFLAG.EQ.2) GO TO 130
204       160 XX1=X1
205           YY1=Y1
206           XX2=X2
207           YY2=Y2
208           CALL PSUB2(IT,IFLAG)
209           IF(IFLAG.NE.0) GO TO 150
210           IF(IT.EQ.0) GO TO 170
211           IF(I.EQ.IBAR) GO TO 250
212           I=I+1
213           GO TO 210
214       170 XX1=X4
215           YY1=Y4
216           XX2=X3
217           YY2=Y3
218           CALL PSUB2(IT,IFLAG)
219           IF(IFLAG.NE.0) GO TO 150
220           IF(IT.EQ.0) GO TO 180
221           IF(I.EQ.1) GO TO 250
222           I=I-1
223           GO TO 210
224       180 XX1=X3
225           YY1=Y3
226           XX2=X2
227           YY2=Y2
228           CALL PSUB2(IT,IFLAG)
229           IF(IFLAG.NE.0) GO TO 150
230           IF(IT.EQ.0) GO TO 190
231           IF(J.EQ.JP1) GO TO 250
232           J=J+1
233           GO TO 200
234       190 XX1=X4
235           YY1=Y4
236           XX2=X1
237           YY2=Y1
238           CALL PSUB2(IT,IFLAG)
239           IF(IFLAG.NE.0) GO TO 150
240           IF(IT.EQ.0) GO TO 230
241           IF(J.EQ.2) GO TO 250
242           J=J-1
243       200 IEC=(J-1)*NQI
244           JOLO=J
245           CALL ECRO(AASC(ISC2),IEC,NQI2,IOUM)
246           GO TO 220
247       210 IJ=(I-1)*NQ+ISC2
248           IPJ=IJ+NQ
249           IJP=IJ+NQI
250           IPJP=IPJ+NQI
251       220 X1=X(IPJ)
252           Y1=Y(IPJ)
253           X2=X(IPJP)
254           Y2=Y(IPJP)
255           X3=X(IJP)
256           Y3=Y(IJP)
```

```
257           X4=X(IJ)
258           Y4=Y(IJ)
259           CALL PSUB1(IN)
260           IF(IN.EQ.0) GO TO 160
261           XTE=XTEN
262           YTE=YTEN
263           ITAB(NPPT)=(J-1)*IP1+I
264           GO TO 70
265         230 CONTINUE
266           DO 240 IPX=IPD,IFD,6
267         240 WRITE(IPX,280) NPPT,ITAB(NPPT),XTE,YTE,XTEN,YTEN,X1,Y1
268           ITAB(NPPT)=(J-1)*IP1+I
269           GO TO 130
270         250 XPAR(KP)=-1.E+3
271           DO 260 IPX=IPD,IFD,6
272         260 WRITE(IPX,270) NPPT,ITAB(NPPT)
273           GO TO 130
274       C
275         270 FORMAT(5X,8HPARTICLE,I10,25HTOSSED OUT.   CELL NUMBER,I10)
276         280 FORMAT(5X,14HERROR PARTICLE,I5,6H  CELL,I5,12F7.4)
277           END
```

---

```
  1           SUBROUTINE PSUB1(IN)
  2       C
  3       C     ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
  4       C     MODIFIED BY J.L.NORTON,LASL T-3,1975
  5       C
  6       *     ----- BEGIN COMDECK YSTORE     -----
  7       *     ----- BEGIN COMDECK YAQDIM     -----
  8             DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
  9            1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
 10            2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
 11            3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
 12            4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
 13            5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
 14            6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
 15       *     ----- END COMDECK YAQDIM     -----
 16       *     ----- BEGIN COMDECK YAQSC     -----
 17             LOGICAL RESTRT,FILM,PAPER,TURB
 18             REAL LAM,MU
 19       C     COMMON/YSC1/AASC(NSCP1)
 20             COMMON/YSC1/AASC(9600)
 21             COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
 22            1 DTO(10),OTOC(10),DTO2,OTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
 23            2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
 24            3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
 25            4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
 26             COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
 27            1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
 28            2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
 29            3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
```

```
      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBDTM,
     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,JOUMP,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,TSTRTO
      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,OKLSM,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/3.1415926535897932384626/
      DATA PIO2/1.5707963267948966192313/
      DATA PIO2M1/0.5707963267948966192313/
      DATA PI2/6.2831853071795864769252/
   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-5,1975
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),ROSQ(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 IDTD,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,ROT,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,RTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,DTR,TMASS,DTVSAV,DTCSAV,IOTV
      5 ,JDTV,IOTC,JOTC,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,IPO,IFO
*      ***** 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,OKLSM,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=IPO,IFO,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=IPO,IFO,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 PTRBOF(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),GRRDP(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),OKLSM(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 IDTD,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,ROT,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,PTPOLO,DTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRD,TE,OTR,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      -----
*     ----- 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,OKLSM,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=DXERF*(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 DTD(10),DTDC(10),DTD2,DTD8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GROVEL,GZ,GZP,I,IBAR,
     3 IDTD,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,ROT,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,YCNVLO,
     3 XCNVLO,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),BIG),(AASC(20),TUS),(AASC(
     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),OLSRDI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),DLSRQQ,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/RDTMP,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)
      ROEL=1.0/DEL
      DELE=E(IJ)-0.156
      SUME=SUME+DELE
      ZE=ZE+Y(IJ)*DELE
      SUMRS=SUMRS+ROEL
      ZRS=ZRS+Y(IJ)*ROEL
      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)
```

```
112           IF(I.EQ.2) VG(IMJ)=VG(IJ)
113           GO TO 130
114        50 CONTINUE
115           IF(I.NE.1) GO TO 60
116           UG(IJ)=0.0
117           GO TO 70
118        60 IF(I.NE.IP1) GO TO 70
119           UG(IJ)=FCX
120           VG(IJ)=VG(IMJ)
121        70 IF(J.NE.2) GO TO 80
122           VG(IJ)=-FC3
123           GO TO 90
124        80 IF(J.NE.JP2) GO TO 90
125           UG(IJ)=UG(IJM)
126           VG(IJ)=FCP2
127        90 IF(I.NE.1.OR.J.NE.2) GO TO 100
128           VG(IJ)=-FC3
129       100 IF(I.NE.IP1.OR.J.NE.2) GO TO 110
130           UG(IJ)=FCX
131           VG(IJ)=-FC3
132       110 IF(I.NE.1.OR.J.NE.JP2) GO TO 120
133           VG(IJ)=FCP2
134       120 IF(I.NE.IP1.OR.J.NE.JP2) GO TO 130
135           UG(IJ)=FCX
136           VG(IJ)=FCP2
137       130 CONTINUE
138           IJ=IPJ
139           IJP=IJP+NQ
140           IJM=IJM+NQ
141       140 CONTINUE
142           CALL LOOP
143       150 CONTINUE
144           CALL DONE
145           CALL START
146           DO 180 J=2,JP2
147           DO 170 I=1,IP1
148           X(IJ)=X(IJ)+UG(IJ)*DT
149           IF(J.NE.2) GO TO 160
150           IF(Y(IJ)+VG(IJ)*DT.LE.0.0) VG(IJ)=-Y(IJ)*RDT
151       160 CONTINUE
152           Y(IJ)=Y(IJ)+VG(IJ)*DT
153           R(IJ)=X(IJ)*CYL+DMCYL
154       170 IJ=IJ+NQ
155           CALL LOOP
156       180 CONTINUE
157           CALL DONE
158           CALL FILMCO
159           CALL START
160           ETMP=REZSIE
161           ROTMP=REZRON
162           CONCJ=0.
163           CALL AIR
164           XX=GMONE*REZSIE
165           YY=ABS(GZ)/XX
166           DO 200 J=2,JP1
167           DO 190 I=1,IBAR
168           IPJ=IJ+NQ
```

```
169           IPJP=IJP+NQ
170           Y4=0.25*(Y(IJP)+Y(IPJP)+Y(IJ)+Y(IPJ))
171           IF(J.EQ.2) ROL(IJM)=REZRON*EXP((REZY0-(Y(IJ)+Y(IPJ)-Y4))*YY)
172           IF(I.EQ.IBAR) ROL(IPJ)=REZRON*EXP((REZY0-(Y(IPJ)+Y(IPJP)-Y4))*YY)
173           IF(J.EQ.JP1) ROL(IJP)=REZRON*EXP((REZY0-(Y(IJP)+Y(IPJP)-Y4))*YY)
174           IJM=IJM+NQ
175           IJP=IJP+NQ
176       190 IJ=IPJ
177           CALL LOOP
178       200 CONTINUE
179           CALL DONE
180           RETURN
181           END
```

```
  1           SUBROUTINE RTAPE
  2     C
  3     C     ROUTINE TO READ IN ONE DUMP OF A YAQUI DUMP FILE (ALL BUT
  4     C         THE HEADER RECORD)
  5     C
  6     C     WRITTEN BY J.L.NORTON,LASL T-3,1975
  7     C
  8     *     ----- BEGIN COMDECK PARAM       -----
  9           COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
 10          1 NLCP3,NLCP4,IFLMSZ
 11     *     ----- END COMDECK PARAM       -----
 12     *     ----- BEGIN COMDECK YAQSC      -----
 13           LOGICAL RESTRT,FILM,PAPER,TURB
 14           REAL LAM,MU
 15     C     COMMON/YSC1/AASC(NSCP1)
 16           COMMON/YSC1/AASC(9600)
 17           COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
 18          1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
 19          2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
 20          3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
 21          4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
 22           COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
 23          1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
 24          2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
 25          3 FREZYT,FREZYB,ROMFR,T,THIRO,NCLST,TOUT,TWFIN
 26           COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
 27          1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
 28          2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIY80,IY80,YCNVLD,
 29          3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
 30          4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRO,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
 31          5 ,JDTV,IDTC,JDTC,CIRC,TIS,PDTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
 32          6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
 33           COMMON/YSC2/ZZ
 34     C     COMMON/YSC4/ITAB(ITABP)
 35           COMMON/YSC4/ITAB(1000)
 36           COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
 37     *     ----- 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,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBQ,IYBO,YCNVLO,
     3 XCNVLO,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,IPFR,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,OTVSAV,OTCSAV,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,GROVEL,GZ,GZP,I,IBAR,
     3 IDTD,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,LP8,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,OTVSAV,OTCSAV,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      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,IOUM)
      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
```

```
147           IJPS=ISC2
148           IJ=1
149           IJM=ISC3
150           IJMS=ISC3
151     C
152     C     INDICATE THAT BUFFER THREE IS TO BE FILLED NEXT TIME
153     C
154           IBUF=3
155     C
156     C     GO FILL BUFFER TWO
157     C
158           GO TO 20
159     C
160     C     ***************************************************************
161           ENTRY DONE
162     C     ***************************************************************
163     C
164     C     LOOP IS COMPLETED, WRITE OUT CONTENTS OF THE LAST TWO BUFFERS
165     C         THAT WERE LOADED, WRITE OUT BUFFER THAT WOULD HAVE BEEN
166     C         FILLED NEXT.
167     C
168           CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
169           IECW=IECW+NQI
170     C
171     C     DETERMINE WHICH IS THE OTHER BUFFER TO BE EMPTIED
172     C
173           GO TO (50,60,70),IBUF
174     C
175     C     BUFFER ONE HAS BEEN WRITTEN OUT, WRITE OUT BUFFER TWO.
176     C
177        50 IJMS=ISC2
178           GO TO 80
179     C
180     C     BUFFER TWO HAS BEEN WRITTEN OUT, WRITE OUT BUFFER THREE.
181     C
182        60 IJMS=ISC3
183           GO TO 80
184     C
185     C     BUFFER THREE HAS BEEN WRITTEN OUT, WRITE OUT BUFFER ONE.
186     C
187        70 IJMS=1
188        80 CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
189           RETURN
190           END
```

---

```
1             SUBROUTINE STARTD
2       C
3       C     THIS ROUTINE (AND ITS ENTRY POINT LOOPD) ALLOW ONE TO LOOP
4       C         BACKWARDS THROUGH THE MESH USING TWO SCM BUFFERS, ROWS J
5       C         AND J-1 ARE MADE AVAILABLE. THE CALL TO STARTD READS ROW JP2
6       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),DT02,DT08,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,ROT,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,FIYBD,IYBO,YCNVLO,
           3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,OTR,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 YAGSC      -----
      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 JUNF,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,ROT,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,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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/YSCS/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.
```

```
151             CALL NCODE(8,6H(6XA2),IMSIGN,1,BCO)
152             GO TO 200
153         130 CONTINUE
154             XLOG=ALOG10(ABS(YTIC))
155             IMSIGN=1H-
156             GO TO 150
157         140 CONTINUE
158             XLOG=ALOG10(YTIC)
159         150 CONTINUE
160             IEXP=XLOG
161             XMAN=XLOG-FLOAT(IEXP)
162             IF(XMAN.GE.0.) GO TO 160
163             XMAN=XMAN+1.
164             IEXP=IEXP-1
165         160 CONTINUE
166             XMAN=1.000001*XMAN
167             IF(XMAN.LT.1.) GO TO 170
168             XMAN=XMAN-1.
169             IEXP=IEXP+1
170         170 CONTINUE
171             XMAN=10.**XMAN
172             IESIGN=1H+
173             IF(IEXP.GE.0) GO TO 180
174             IEXP=-IEXP
175             IESIGN=1H-
176         180 CONTINUE
177             IF(IEXP.GE.10) GO TO 190
178             CALL NCODE(8,17H(1XA1,F4.2,A1,I1),IMSIGN,4,BCO)
179             GO TO 200
180         190 CONTINUE
181             CALL NCODE(8,15H(A1,F4.2,A1,I2),IMSIGN,4,BCO)
182         200 CONTINUE
183             CALL TSP(1,IY1,8,BCO)
184     C
185     C       INCREMENT THE TIC VALUE AND PROCEED
186     C
187             YTIC=YTIC+DTIC
188             GO TO 120
189     C
190     C       SAME PROCEDURE AS FOR Y-AXIS
191     C
192         210 CONTINUE
193     C
194     C       USE THE SAME LABELLING INCREMENT FOR THE X-AXIS
195     C
196             XTIC=0.
197         220 CONTINUE
198             IX1=FIXL+(XTIC-XL)*XCONVP
199             IF(IX1.GT.IXR) RETURN
200             CALL DRV(IX1,IYB,IX1,IY2)
201             CALL DRV(IX1,IY3,IX1,IYT)
202             JRITE=IYB+9
203             IRITE=IX1-29
204             IMSIGN=1H
205             IF(XTIC.GT.0.) GO TO 240
206             IF(XTIC.LT.0.) GO TO 230
207             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),OLSRQI(1),OLSRQQ(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),OKLSM(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,ROT,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,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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),OELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,OKLSM,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),OLSRQI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),OLSRQQ,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)
      OLSROI(IJ)=0.
      OLSROQ(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)
      OLSROQ(IJ)=.5*((R1*TUQ(IPJ)-RCQ)*Z1+(R2*TUQ(IJP)-RCQ)*Z2+(R3*TUQ(
     1 IMJ)-RCQ)*Z3+(R4*TUQ(IJM)-RCQ)*Z4)
      OLSROI(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
      GRRDP(IJ)=DRODR*DPDR+DRODZ*DPDZ
      GRRDR(IJ)=DRODR
      GRRDZ(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=AVX8V(IJM)
      X2=AVXSV(IJ)
      X3=AVXSV(IMJ)
      X4=AVX8V(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 CONTINUE
      GRSV(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
      O=OELSM(IJ)
      XPR1=2.*SIJ/(TUS(IJ)**2*QLEVEL)
      TUQ(IJ)=RIJ*TUQ(IJ)+DT*(SIJ*(CAPGAM(IJ)*RIJ-GRROP(IJ)*RECRHO)+
     1 OLSROQ(IJ)*RVOL(IJ))
      TUQ(IJ)=(TUQ(IJ)-TWTHRO*O*O*SIJ*RIJ*DT)/((1.+(XPR1+TWTHRO*O)*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*(OLSROI(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,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,ROT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
     3 FREZYT,FREZYB,ROMFR,T,THIRO,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,PTPOLO,DTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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      -----
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,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,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,ROT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
           3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
            COMMON/YSC2/TUQI,TUSI,NQQ,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,OTSV,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,OTVSAV,DTCSAV,IOTV
           5 ,JDTV,IOTC,JOTC,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,IFO
      *     ----- END COMDECK YAQSC      -----
            DIMENSION CHAR(1)
            DATA TP/0./
            DO 10 IPX=6,IFO,6
         10 WRITE(IPX,140)
            GO TO (60,20,40,60),ISFLAG
         20 CONTINUE
            DO 30 IPX=6,IFO,6
         30 WRITE(IPX,150)
            GO TO 80
         40 CONTINUE
            DO 50 IPX=6,IFO,6
         50 WRITE(IPX,130)
            GO TO 80
```

```
59          60 CONTINUE
60             DO 70 IPX=6,IFO,6
61          70 WRITE(IPX,110)
62          80 CONTINUE
63             NWOS=NCHAR/10
64             IF(MOD(NCHAR,10).NE.0) NWOS=NWOS+1
65             DO 90 IPX=6,IFO,6
66             WRITE(IPX,120) RNAME,(CHAR(II),II=1,NWOS)
67          90 WRITE(IPX,140)
68             CALL SECOND(TP)
69             WRITE(59,100) TP
70             IF(ISFLAG.EQ.2) CALL EXIT
71             IF(ISFLAG.EQ.3) RETURN
72             CALL EXIT(2)
73       C
74         100 FORMAT(1H ,25HCODE TERMINATION AT CP = ,F10.4)
75         110 FORMAT(1H ,20HFATAL ERROR IN YAQUI)
76         120 FORMAT(1H ,24HUNCLE CALLED BY ROUTINE ,A6,10X9A10)
77         130 FORMAT(1H ,16HSTART UNCLE DUMP)
78         140 FORMAT(1H ,60(1H=))
79         150 FORMAT(1H ,25HNORMAL EXIT THROUGH UNCLE)
80             END
```

---

```
 1             SUBROUTINE VELPLT(VMAX,IFLAG)
 2       C
 3       C     ROUTINE TO DO VELOCITY PLOTS
 4       C
 5       C     VMAX IS THE MAXIMUM OF THE VELOCITY CURRENTLY BEING PLOTTED
 6       C         IN ANY ONE DIRECTION.
 7       C     IFLAG IS 0 FOR FLUID VELOCITY PLOTS SCALED BY VMAX. IT IS 1 FOR
 8       C         UNSCALED FLUID VELOCITY PLOTS AND 2 FOR SCALED RELATIVE
 9       C         VELOCITY PLOTS.
10       C
11       C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
12       C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
13       C
14       *     ----- BEGIN COMDECK PARAM       -----
15             COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
16            1 NLCP3,NLCP4,IFLMSZ
17       *     ----- END COMDECK PARAM       -----
18       *     ----- BEGIN COMDECK YSTORE      -----
19       *     ----- BEGIN COMDECK YAQDIM      -----
20             DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
21            1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
22            2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
23            3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),OLSROI(1),OLSROQ(1),CAPGAM(1),TUQ
24            4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
25            5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
26            6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),OKLSM(1),AREA(1)
27       *     ----- END COMDECK YAQDIM      -----
28       *     ----- 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 IOTO,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,ROT,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,PTPOLO,DTSV,DTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,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       -----
*     ----- 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,OKLSM,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),OLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),OLSROQ,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
```

```
 86            OROU=VV/VMAX
 87      C
 88      C     SPACE FORWARD TO THE NEXT FILM FRAME
 89      C
 90            CALL ADV(1)
 91      C
 92      C     LOOP OVER ALL REAL ZONES
 93      C
 94            CALL START
 95            DO 20 J=2,JP2
 96            DO 10 I=1,IP1
 97            IF(IFLAG.EQ.1) OROU=0.5*VV/SQRT(U(IJ)**2+V(IJ)**2+EM10)
 98      C
 99      C     (IX1,IY1) IS THE LOCATION OF THE VERTEX.
100      C     (IX2,IY2) IS THE LOCATION OF THE END OF THE VELOCITY VECTOR.
101      C     IF EITHER ARE OUTSIDE OF THE PLOTTING RECTANGLE,SKIP THIS VERTEX.
102      C
103           IY1=FIYB+(Y(IJ)-YLB)*YCONVP
104           IF(IY1.GT.IYB.OR.IY1.LT.IYT) GO TO 10
105           IY2=FIYB+(Y(IJ)+V(IJ)*OROU-YLB)*YCONVP
106           IF(IY2.GT.IYB.OR.IY2.LT.IYT) GO TO 10
107           IX1=FIXL+(X(IJ)-XL)*XCONVP
108           IF(IX1.GT.IXR) GO TO 10
109           IX2=FIXL+(X(IJ)-XL+U(IJ)*OROU)*XCONVP
110           IF(IX2.GT.IXR) GO TO 10
111     C
112     C     DRAW THE VECTOR
113     C
114           CALL DRV(IX1,IY1,IX2,IY2)
115     C
116     C     PLOT A PLUS (+) AT THE VERTEX POSITION
117     C
118           CALL PLT(IX1,IY1,16)
119        10 IJ=IJ+NQ
120           CALL LOOP
121        20 CONTINUE
122     C
123     C     LABEL THE PLOT WITH VMAX
124     C
125           CALL LINCNT(60)
126           IFLGP=IFLAG+1
127           WRITE(IFO,40)(ITITLE(I,IFLGP),I=1,5),VMAX
128           WRITE(IFO,30) JNM,NAME,T,NCYC
129           RETURN
130     C
131        30 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
132        40 FORMAT(1H ,5A10/18X5HVMAX=,1PE12.5)
133           END
```

```
 1            SUBROUTINE W1ROW
 2      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,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,ROT,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 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRD,TE,OTR,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),OLSROI(1),OLSROQ(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,TOQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,PIYBD,IYBO,YCNVLD,
      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JOUMP,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),OLSROI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),OLSROQ,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.
      TQQSAV=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
```

```
125           TNEG=0.
126           EP1=TK+TI+TP3+TUQENG
127           TIP=TI-(TMASS-TMASSV)*REZSIE
128        40 CONTINUE
129     C
130     C     TOLD IS THE TIME AT THE END OF THE LAST CYCLE
131     C
132           TOLD=T2
133     C
134     C     IF JUST FINISHING INITIALIZATION,GO DO AN EDIT FIRST
135     C
136           IF(NCYC.EQ.0) GO TO 150
137        50 CONTINUE
138     C
139     C     READY TO BEGIN NEXT CYCLE, INCREMENT THE CYCLE NO.
140     C
141           NCYC=NCYC+1
142     C
143     C     SET THE NEW TIMESTEP, SAVE THE OLD ONE IN DTLAST.
144     C
145           DTLAST=DT
146     C
147     C     ON THE FIRST CYCLE,SET DEFAULT DTC AND DTV
148     C
149           IF(NCYC.NE.1) GO TO 60
150           DTC=DT
151           DTV=DT
152        60 CONTINUE
153     C
154     C     ON THE SECOND CYCLE,BOOST DT BY A FACTOR OF 10
155     C
156           IF(NCYC.NE.2) GO TO 70
157           DT=10.*DT
158           DTSV=DT
159        70 CONTINUE
160     C
161     C     EXCEPT ON THE FIRST AND SECOND CYCLES,SET DT BASED ON DTV AND DTC
162     C
163           IF(NCYC.GE.3) DT=AMIN1(DTV,DTC)
164     C
165     C     DO NOT ALLOW THE TIMESTEP TO INCREASE BY MORE THAN 25 PER CENT
166     C         OVER THE LAST CYCLE
167     C
168           DTFAC=1.25
169           DT=AMIN1(DTFAC*DTSV,DT)
170     C
171     C     IF WE WILL BE DOING AN EDIT AFTER THE NEXT CYCLE,ADJUST DT TO
172     C         MAKE T EXACTLY EQUAL TO THE EDIT TIME. DTPOS IS DT BEFORE
173     C         ANY SUCH ADJUSTMENT. DTSV IS THE TIMESTEP SAVED FOR FUTURE
174     C         REFERENCE.
175     C
176           DTPOS=DT
177           DTSV=DT
178           IF(T+DT.GT.TOUT) DT=TOUT-T
179           IF(DT.LT.1.E-8) DT=1.E-8
180     C
181     C     UPDATE T
```

```
182      C
183            T=T+DT
184            IF(DT.NE.DTPOS) T=TOUT
185      C
186      C     ***************************************************************************
187      C     TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS
188      C     ***************************************************************************
189      C
190            IF(T.GT..6) ANC=.05
191      C
192      C     ILNG IS ONE IF THIS IS AN EDIT CYCLE
193      C
194            ILNG=0
195            IF(ABS(TOUT-T).LT.EM10) ILNG=1
196      C
197      C     SET QUANTITIES DERIVED FROM THE TIMESTEP
198      C
199            ROT=1./DT
200            DT02=.5*DT
201            DT08=.125*DT
202      C
203      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
204      C     PHASE 1 - EXPLICIT LAGRANGIAN PART
205      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
206      C
207            CALL YPH1
208      C
209      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
210      C     PHASE 2 - PRESSURE ITERATION
211      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
212      C
213            CALL YPH2
214      C
215      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
216      C     PHASE 3,PART I - CONTINUOUS REZONE
217      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
218      C
219            CALL YPH3
220      C
221      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
222      C     PHASE 3,PART II - FLUXING
223      C     $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
224      C
225            CALL YFLUX
226      C
227      C     CALCULATE AN INFORMATIVE RELATIVE GRID-FLUID TIMESTEP AND THE
228      C         MAXIMUM OF THE DIFFERENCE BETWEEN GRID AND FLUID VELOCITIES,
229      C         ALSO CALCULATE THE CENTER OF THE CELL WITH MAXIMUM
230      C         TEMPERATURE.
231      C
232            DTR=1.E30
233            VMAX=-1.E10
234            CALL START
235            DO 110 J=2,JP2
236            DO 100 I=1,IP1
237            U(IJ)=UG(IJ)-U(IJ)
238            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=YCNVLO
      XCONVP=XCNVLO
      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
```

```
296           IF(NPT.GT.0) CALL PRTMOV
297     C
298     C     FINISH THE CYCLE (PRINTS,SUMMARIES,ETC)
299     C
300       150 CONTINUE
301     C
302     C     CALCULATE QUANTITIES TO BE USED IN AN EDIT AND FOR THE NEXT CYCLE
303     C
304           CALL CINIT
305     C
306     C     T2 IS THE TIME AT THE END OF THE THIS CYCLE
307     C
308           CALL SECOND(T2)
309     C
310     C     CALCULATE THE CP TIME USED PER ZONE (GRIND TIME)
311     C
312           XX=(T2-TOLD)*RIBJB
313     C
314     C     CONVERT POTE TO ACTUAL GRAVITATIONAL POTENTIAL ENERGY
315     C
316           EPOT=-POTE*GZ
317     C
318     C     COMPUTE FIREBALL DIAMETER AND AVERAGE HEIGHT
319     C
320           PDIAM=2.*PRITE
321           PAVHT=.5*(PTOP+PBOTM)
322     C
323     C     PRINT OUT CYCLE SUMMARY
324     C
325           DO 160 IPX=6,IFO,6
326       160 WRITE(IPX,340) NCYC,T,DT,T2,XX,NUMIT,CIRC,DTV,IDTV,JDTV,DTC,IDTC,
327          1 JDTC,TMAX,ITM,JTM,XTMAX,YTMAX,TGMX,ITG,JTG,PRITE,PTOP,PBOTM,PDIAM
328          2 ,PAVHT
329           DO 170 IPX=6,IFO,6
330       170 WRITE(IPX,300) TI,TK,EPOT,UMOM,VMOM
331           DO 180 IPX=6,IFO,6
332       180 WRITE(IPX,320) DTV,DTC
333           CALL OVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
334           DO 190 IPX=6,IFO,6
335       190 WRITE(IPX,310) VMAX,IVM,JVM
336     C
337     C     SEE IF IT IS TIME TO TURN ON THE TURBULENCE BASED ON TIME
338     C
339           IF(NCQ.EQ.0.AND.T.GE.TQ) GO TO 200
340     C
341     C     NO. SEE IF IT IS TIME BASED ON CYCLES.
342     C
343           IF(NCYC.NE.0.AND.NCQ.EQ.NCYC) GO TO 200
344     C
345     C     NO. SKIP TURBULENCE SEEDING.
346     C
347           GO TO 240
348       200 CONTINUE
349     C
350     C     YES. FLIP ON THE SWITCH.
351     C
352           TURB=.TRUE.
```

```
353           NCQ=NCYC
354     C
355     C     GO GET THE VORTICITY
356     C
357           CALL GETOMG
358     C
359     C     INITIALIZE TUQ AND TUSV
360     C
361           TUSV=0.
362           CUTOFF=-.1*QMN*TUQI
363           CALL START
364           DO 220 J=2,JP1
365           DO 210 I=1,IBAR
366           TEST=-TUQI*CQ(IJ)
367           TUQ(IJ)=0.
368           IF(TEST.GT.CUTOFF) TUQ(IJ)=TEST
369           TUSV=TUSV+TUQ(IJ)*RO(IJ)/RVOL(IJ)
370           IJ=IJ+NQ
371       210 IJP=IJP+NQ
372           CALL LOOP
373       220 CONTINUE
374           CALL DONE
375     C
376     C     INITIALIZE THE PARTICLE TURBULENT DIFFUSION
377     C         INTERPOLATION TABLES
378     C
379           CALL TRBERF
380     C
381     C     INDICATE THAT THE TURBULENCE HAS BEEN SEEDED
382     C
383           DO 230 IPX=6,IFO,6
384       230 WRITE(IPX,330) NCQ,TUQI,TUSV
385       240 CONTINUE
386     C
387     C     SEE IF TURBULENCE IS ON
388     C
389           IF(.NOT.TURB) GO TO 270
390     C
391     C     YES, CALCULATE SIG AND TUS.
392     C
393           CALL START
394           DO 260 J=2,JP1
395           DO 250 I=1,IBAR
396           TUS(IJ)=TUSI
397           SIG(IJ)=QLEVEL*TUS(IJ)*SQRT(2.*TUQ(IJ))
398       250 IJ=IJ+NQ
399           CALL LOOP
400       260 CONTINUE
401           CALL DONE
402       270 CONTINUE
403     C
404     C     SEE IF IT IS TIME DO AN EDIT
405     C
406           EDIT=.FALSE.
407           IF(T+EM10.LT.TOUT) GO TO 280
408     C
409     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 ,20X020.025)
  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, POIAM=,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,GROVEL,GZ,GZP,I,IBAR,
```

```
29           3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
30           4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
31            COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNF02,KXI,LAM,LPB,MU,NAME(8),
32           1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
33           2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
34           3 FREZYT,FREZYB,ROMFR,T,THIRO,NCLST,TOUT,TWFIN
35            COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
36           1 ILNG,NILNG,TP3,TUPOT,TOQSAV,TK,TI,TUQENG,EP1,SAV1,OLEVEL,TQ,IST,
37           2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,OTSV,OTLAST,FIYBO,IYBO,YCNVLD,
38           3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
39           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,OTR,TMASS,OTVSAV,OTCSAV,IOTV
40           5 ,JOTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
41           6 ,TMASSV,WMAXEF,RMINEF,TSTRTO
42            COMMON/YSC2/ZZ
43      C     COMMON/YSC4/ITAB(ITABP)
44            COMMON/YSC4/ITAB(1000)
45            COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
46      *     ----- END COMDECK YAQSC       -----
47      *     ----- BEGIN COMDECK YAQEQ      -----
48            EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
49           1 AASC(4),U),(AASC(5),V),(AASC(6),RQ),(AASC(7),OELSM,RCSQ,MP),(AASC
50           1 (8),E,ETIL,AREA,XR13K),
51           2 (AASC(15),SIE),(AASC(16),PM0,OKLSM,RMP),(AASC(9
52           3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
53           4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
54           5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
55           6 21),GRROR),(AASC(22),GRROZ),(AASC(23),OLSROI,Y13K),(AASC(24),GZSV
56           7 ),(AASC(25),OLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
57           8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
58           9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
59           1 AVYSV,X24K)
60            REAL M,MP,MPAR,MTIL
61      *     ----- END COMDECK YAQEQ       -----
62      *     ----- END COMDECK YSTORE      -----
63            INTEGER TAPE,AA
64            DIMENSION RCYCLE(3,3)
65            DATA (RCYCLE(II),II=1,9)/9*0/
66            DATA IEFLAG/0/
67            TLIMSV=TLIM
68      C
69      C     SET UP NAMELIST INPUT TABLE
70      C
71            ASSIGN 120 TO IERRT
72            CALL TABDEF(RCYCLE,6HRCYCLE,3,IERRT)
73            CALL TABSET(RCYCLE,6HINTCYC,INTCYC,IEFLAG,0,0,0,0)
74            CALL TABSET(RCYCLE,4HTAPE,TAPE,IEFLAG,0,0,0,0)
75      C
76      C     READ THE RESTART TAPE NO. AND THE RESTART CYCLE. INTCYC=-1
77      C         SIGNIFIES RESTART FROM THE LAST DUMP ON THE TAPE.
78      C
79            INTCYC=-1
80            TAPE=0
81            CALL NAMLST(RCYCLE,5,IEFLAG)
82      C
83      C     CHECK FOR INPUT ERRORS
84      C
85            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,36,36HRCYCLE 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,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,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,ROT,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,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),OELSM,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,IFO,6
   10 WRITE(IPX,320) NAME
C
C     IF NCO.LT.0,THERE WILL BE NO TURBULENCE
C
      IF(NCO.LT.0) GO TO 30
C
C     THERE WILL BE TURBULENCE. PRINT OUT THE INPUT QUANTITIES
C
      DO 20 IPX=6,IFO,6
   20 WRITE(IPX,290) QLEVEL,TUQI,TUSI,NCO,TQ,TSTRTO,WMAXEF,RMINEF
      GO TO 50
C
C     THERE WILL BE NO TURBULENCE. INDICATE SUCH.
C
   30 CONTINUE
      DO 40 IPX=6,IFO,6
   40 WRITE(IPX,300)
   50 CONTINUE
C
C     PRINT GENERAL INPUT VARIABLES
C
      DO 60 KT=6,IFO,6
      WRITE(KT,330) IBAR,JBAR,IUNF,JUNF,JCEN,DR,DZ,CYL,GROVEL,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
      DMCYL=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 -STARTO- 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     IOTO IS THE SUBSCRIPT IN THE DTOC ARRAY SUCH THAT
C                    DTOC(IOTO-1).LT.T.LE.DTOC(IOTO)
C
      IOTO=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(IOTO)
      IF(TOUT.LE.DTOC(IOTO)) GO TO 80
      TOUT=DTOC(IOTO)
      IOTO=IOTO+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
```

```
186      C      DTPOS IS THE TIMESTEP THAT IS POSSIBLE BASED ON STABILITY
187      C          CRITERIA. THE ACTUAL TIMESTEP DT MAY BE LESS IF IT HAS BEEN
188      C          ADJUSTED ON AN OUTPUT CYCLE SO THAT T=TOUT EXACTLY.
189      C
190              DTPOS=DT
191      C
192      C      DTSV IS THE TIMESTEP FROM THE LAST CYCLE
193      C
194              DTSV=DT
195      C
196      C      NCYC IS THE CYCLE NO.
197      C
198              NCYC=0
199      C
200      C      COLAMU IS 1/(2/3*(2*MU+LAM))
201      C
202              COLAMU=1.5/(LAM+MU+MU+EM10)
203      C
204      C      NILNG IS THE NO. OF TIMES A PARTICLE HAS BEEN STORED FOR USE IN
205      C          MAKING TIME-DEPENDENT PARTICLE PLOTS
206      C
207              NILNG=0
208      C
209      C      IUNF MUST BE AT LEAST 1
210      C
211              IUNF=MAX0(IUNF,1)
212      C
213      C      JUNF MUST BE AT LEAST 2
214      C
215              JUNF=MAX0(JUNF,2)
216              JUNF02=JUNF/2
217      C
218      C      IF JCEN IS ZERO, SET IT TO JBAR/2
219      C
220              IF(JCEN.EQ.0) JCEN=JBAR/2
221      C
222      C      GO GENERATE THE PARTICLES
223      C
224              CALL PRTGEN
225      C
226      C      GO GENERATE THE MESH
227      C
228              CALL MSHMKR
229      C
230      C      GO SET THE PLOT QUANTITIES
231      C
232              CALL FILMCO
233      C
234      C      ************************************************************************
235      C      LOCATE WHICH CELL EACH PARTICLE IS IN IF THERE ARE ANY PARTICLES
236      C      ************************************************************************
237      C
238              IF(NPT.LE.0) GO TO 210
239      C
240      C      ASSUMING THAT THE MESH IS STILL RECTANGULAR,STORE THE X AND Y
241      C          VALUES IN XTAB AND YTAB. MAKE SURE THE ARRAYS ARE LARGE
242      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,IOUM)
           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,IFO,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,IFO,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 STARTO
            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 LOOPO
        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,3HOR=1PE12.5/5X,3HOZ=E12.5/4X,4HCYL=E12.5/8H GROVEL=
           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 ,4X3HOT=,E12.5/1H ,1X6HNCLST=,I6/1H ,
           1 1X6HTWFIN=,E12.5/1H ,1X6HPAPER=,I1/1H ,2X5HFILM=,I1)
        380 FORMAT(12H  OTO(1=10)=4(1PE14.5)/(12X,4E14.5))
        390 FORMAT(12H OTOC(1=10)=4(1PE14.5)/(12X,4E14.5))
            END
```

---

```
      1           SUBROUTINE YOUMP
      2     C
      3     C     ROUTINE TO DO A YAQUI DUMP
      4     C
      5     C     WRITTEN BY J.L.NORTON,LASL T=3,1975
      6     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,GROVEL,GZ,GZP,I,IBAR,
           3 IOTO,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,THIRO,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,OTSV,OTLAST,FIYBO,IYBO,YCNVLD,
           3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,DTR,TMASS,OTVSAV,DTCSAV,IOTV
           5 ,JOTV,IOTC,JOTC,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,IPO,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,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,ROT,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,YCNVLO,
     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,IPO,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),DLSRDI,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/1777000000000007777777B/
      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=IPO,IFO,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=IPO,IFO,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=YCNVLO
```

```
      XCONV=XCNVLO
      FIXR=FIXRO
      FIXL=FIXLO
      IXR=IXRO
      IXL=IXLO
      CALL AOV(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),OLSROI(1),OLSROQ(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,ROT,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,IY80,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,IPO,IFO
*      ***** 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),OELSM,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),OLSROI,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
      0=.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)
```

```
187           IF(.NOT.TURB) GO TO 90
188           TUQ(IJ)=TUQVEC(IJ)
189           IF(TUQ(IJ).LT.0.) TNEG=TNEG+TUQ(IJ)*RO(IJ)/RVOL(IJ)
190           IF(TUQ(IJ).LT.0.) TUQ(IJ)=0.
191        90 CONTINUE
192           IF(J.EQ.JP1) RO(IJP)=ROL(IJP)
193           IF(I.EQ.IBAR) RO(IJ+NQ)=ROL(IJ+NQ)
194           IJM=IJM+NQ
195           IJP=IJP+NQ
196       100 IJ=IJ+NQ
197           CALL LOOP
198       110 CONTINUE
199           CALL DONE
200           CALL STARTD
201           DO 130 JJ=2,JP2
202           J=JP4-JJ
203           DO 120 II=1,IP1
204           I=IP2-II
205           IMJ=IJ-NQ
206           IMJM=IJM-NQ
207           XX=0.
208           IF(I.NE.IP1.AND.J.NE.2) XX=MP(IJM)
209           IF(I.NE.IP1.AND.J.NE.JP2) XX=XX+MP(IJ)
210           IF(I.NE.1.AND.J.NE.JP2) XX=XX+MP(IMJ)
211           IF(I.NE.1.AND.J.NE.2) XX=XX+MP(IMJM)
212           RMP(IJ)=4./XX
213           IJ=IMJ
214       120 IJM=IMJM
215           CALL LOOPD
216       130 CONTINUE
217           CALL START
218           DO 150 J=2.JP2
219           DO 140 I=1,IP1
220           XX=RMP(IJ)/RM(IJ)
221           UP(IJ)=XX*UL(IJ)
222           VP(IJ)=XX*VL(IJ)
223       140 IJ=IJ+NQ
224           CALL LOOP
225       150 CONTINUE
226           CALL DONE
227           CALL START
228           DO 260 J=2,JP1
229           DO 250 I=1,IBAR
230           IPJ=IJ+NQ
231           IPJP=IJP+NQ
232           X1=X(IPJ)
233           Y1=Y(IPJ)
234           R1=R(IPJ)
235           UL1=UL(IPJ)
236           UG1=UG(IPJ)
237           VL1=VL(IPJ)
238           VG1=VG(IPJ)
239           X2=X(IPJP)
240           Y2=Y(IPJP)
241           R2=R(IPJP)
242           UL2=UL(IPJP)
243           UG2=UG(IPJP)
```

```
244             VL2=VL(IPJP)
245             VG2=VG(IPJP)
246             X3=X(IJP)
247             Y3=Y(IJP)
248             R3=R(IJP)
249             UL3=UL(IJP)
250             UG3=UG(IJP)
251             VL3=VL(IJP)
252             VG3=VG(IJP)
253             X4=X(IJ)
254             Y4=Y(IJ)
255             R4=R(IJ)
256             UL4=UL(IJ)
257             UG4=UG(IJ)
258             VL4=VL(IJ)
259             VG4=VG(IJ)
260             XX=DT08*ROL(IJ)
261             UL13=0.5*(UL1+UL3+U(IPJ)+U(IJP))
262             VL13=0.5*(VL1+VL3+V(IPJ)+V(IJP))
263             UL24=0.5*(UL2+UL4+U(IPJP)+U(IJ))
264             VL24=0.5*(VL2+VL4+V(IPJP)+V(IJ))
265             F13=XX*(R1+R3)*((UG1+UG3-UL13)*(Y3-Y1)+(VG1+VG3-VL13)*(X1-X3))
266             F24=XX*(R2+R4)*((UG2+UG4-UL24)*(Y2-Y4)+(VG2+VG4-VL24)*(X4-X2))
267             FM1=F24*RMP(IPJ)
268             FM2=F13*RMP(IPJP)
269             FM3=F24*RMP(IJP)
270             FM4=F13*RMP(IJ)
271             XC=.25*(X1+X2+X3+X4)
272             YC=.25*(Y1+Y2+Y3+Y4)
273             UC=.25*(UL13+UL24)
274             VC=.25*(VL13+VL24)
275             UGC=.25*(UG1+UG2+UG3+UG4)
276             VGC=.25*(VG1+VG2+VG3+VG4)
277             UGCUC=UGC-UC
278             VGCVC=VGC-VC
279             A=UGCUC*(Y3-Y1)+VGCVC*(X1-X3)
280             B=UGCUC*(Y4-Y2)+VGCVC*(X2-X4)
281             IF(A) 160,180,170
282         160 IF(B) 230,220,220
283         170 IF(B) 210,200,200
284         180 IF(B) 210,190,200
285         190 W=H=0.5
286             GO TO 240
287         200 H=1.
288             W=((Y3-YC)*UGCUC+(XC-X3)*VGCVC)/((Y3-Y2)*UGCUC+(X2-X3)*VGCVC)
289             GO TO 240
290         210 W=1.
291             H=((YC-Y1)*UGCUC+(X1-XC)*VGCVC)/((Y2-Y1)*UGCUC+(X1-X2)*VGCVC)
292             GO TO 240
293         220 W=0.
294             H=((YC-Y4)*UGCUC+(X4-XC)*VGCVC)/((Y3-Y4)*UGCUC+(X4-X3)*VGCVC)
295             GO TO 240
296         230 H=0.
297             W=((YC-Y4)*UGCUC+(X4-XC)*VGCVC)/((Y1-Y4)*UGCUC+(X4-X1)*VGCVC)
298         240 OMW=1.-W
299             OMH=1.-H
300             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 OONE
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 YAQOIM     -----
      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),OLSROI(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,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,ROT,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,YCNVLO,
      3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,OTR,TMASS,OTVSAV,DTCSAV,IOTV
      5 ,JOTV,IOTC,JOTC,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,OKLSM,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),OLSROQ,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/FTABC/FTAB(2)
       DIMENSION START(3,5)
       DATA (START(II),II=1,15)/15*0/
       DATA II,IOUM/0,0/
       DATA IEFLAG/0/
       DATA RESTRT,PAPER,FILM,WRAPUP/.TRUE.,.FALSE.,.TRUE.,20./
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 RDUTINES 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),DLSRQI(1),DLSRQQ(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,ROT,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,HMAX,JNM,T2,TLIM,
      4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IOTV
      5 ,JOTV,IDTC,JOTC,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,IPO,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,OKLSM,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),OLSROI,Y13K),(AASC(24),GZSV
      7 ),(AASC(25),OLSROQ,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,2HOT,OT,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,6HGROVEL,GROVEL,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,5HJOUMP,JOUMP,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,6HTSTRTO,TSTRTO,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 26HCARON 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.)
```

```
146           IF(REZSIE.LE.0.) CALL UNCLE(4,6HYINPUT,12,12HREZSIE.LE.0.)
147           IF(OT.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HOT.LE.0.)
148           DO 30 II=2,10
149           IF(DTOC(II).EQ.0..AND.DTO(II).EQ.0.) GO TO 30
150           IF(DTOC(II).LE.DTOC(II-1)) CALL UNCLE(4,6HYINPUT,22,
151          1 22HDTOC MUST BE MONOTONIC)
152           IF(DTO(II).LE.0.) CALL UNCLE(4,6HYINPUT,20,20HILLEGAL VALUE OF DTO
153          1 )
154         30 CONTINUE
155           IF(IBAR.LE.0) CALL UNCLE(4,6HYINPUT,9,9HIBAR.LE.0)
156           IF(JBAR.LE.0) CALL UNCLE(4,6HYINPUT,9,9HJBAR.LE.0)
157           IF(DR.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HDR.LE.0.)
158           IF(DZ.LE.0.) CALL UNCLE(4,6HYINPUT,8,8HDZ.LE.0.)
159           IF(CYL.NE.1..AND.CYL.NE.0.) CALL UNCLE(4,6HYINPUT,20,
160          1 20HILLEGAL VALUE OF CYL)
161           IF(GRDVEL.NE.0..AND.GROVEL.NE.1..AND.GROVEL.NE.2.) CALL UNCLE(4,
162          1 6HYINPUT,23,23HILLEGAL VALUE OF GRDVEL)
163           IF(A0.LT.0..OR.A0.GT.1.) CALL UNCLE(4,6HYINPUT,10,10HA0 ILLEGAL)
164           IF(B0.LT.0..OR.B0.GT.2.) CALL UNCLE(4,6HYINPUT,10,10HB0 ILLEGAL)
165           IF(A0M.LT.0..OR.A0M.GT.1.) CALL UNCLE(4,6HYINPUT,11,11HA0M ILLEGAL
166          1 )
167           IF(KXI.LT.(-1).OR.KXI.GT.1) CALL UNCLE(4,6HYINPUT,11,
168          1 11HKXI ILLEGAL)
169           IF(ANC.LE.0.) CALL UNCLE(4,6HYINPUT,9,9HANC.LE.0.)
170           IF(A0FAC.LT.0..OR.A0FAC.GT.1.) CALL UNCLE(4,6HYINPUT,13,
171          1 13HA0FAC ILLEGAL)
172           RETURN
173         40 CONTINUE
174           CALL UNCLE(4,6HYINPUT,35,35HCARON NAMELIST INITIALIZATION ERROR)
175           END
```

---

```
  1           SUBROUTINE YPH1
  2     C
  3     C     ROUTINE TO DO THE EXPLICIT LAGRANGIAN HYDRO (PHASE 1)
  4     C
  5     C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
  6     C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
  7     *     ----- BEGIN COMDECK YSTORE      -----
  8     *     ----- BEGIN COMDECK YAQDIM      -----
  9           DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
 10          1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
 11          2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
 12          3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSRDI(1),DLSRDQ(1),CAPGAM(1),TUQ
 13          4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
 14          5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),QZSV(1),X13K(1),X24K(1),
 15          6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
 16     *     ----- END COMDECK YAQDIM      -----
 17     *     ----- BEGIN COMDECK YAQSC      -----
 18           LOGICAL RESTRT,FILM,PAPER,TURB
 19           REAL LAM,MU
 20     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),DT02,DT08,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,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,ROT,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,OTSV,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,OTR,TMASS,OTVSAV,DTCSAV,IOTV
     5 ,JDTV,IDTC,JOTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TQMX,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),QELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PM0,QKLSM,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),QLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),QLSROQ,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,MU02
C
C     APPLY THE NODE COUPLER TO ALL VERTICES
C
      CALL START
      Y1=ANC*ROT
      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
```

```
135      C     THE NODE COUPLER IS APPLIED TO THE U (V) VELOCITY IF FLAGU (V)
136      C         IS 1.
137      C
138           FLAGU=0.
139           FLAGV=0.
140      C
141      C     SET FLAGU AND FLAGV. THE TECHNIQUE IS TO EXAMINE THE THREE
142      C         VERTICES (I-1,J),(I,J),AND (I+1), OR (I,J-1),(I,J),AND
143      C         (I,J+1).  IF THE VELOCITY OF VERTEX (I,J) IS THE MINIMUM OR
144      C         MAXIMUM OF EITHER TRIPLET,THE NODE COUPLER WILL BE APPLIED.
145      C         OTHERWISE, IT WILL NOT BE APPLIED.
146      C
147           IF(U5.EQ.AMAX1(U1,U2,U5).OR.U5.EQ.AMIN1(U1,U2,U5)) FLAGU=1.0
148           IF(V5.EQ.AMAX1(V1,V2,V5).OR.V5.EQ.AMIN1(V1,V2,V5)) FLAGV=1.0
149           IF(U5.EQ.AMAX1(U3,U4,U5).OR.U5.EQ.AMIN1(U3,U4,U5)) FLAGU=1.0
150           IF(V5.EQ.AMAX1(V3,V4,V5).OR.V5.EQ.AMIN1(V3,V4,V5)) FLAGV=1.0
151      C
152      C     UAV,VAV ARE THE DIFFERENCES OF THE AVERAGE VELOCITIES OF THE
153      C         NEIGHBORING VERTICES AND THE VELOCITIES OF VERTEX (I,J)
154      C         ITSELF
155      C
156           UAV=0.25*(U1+U2+U3+U4)-U5
157           VAV=0.25*(V1+V2+V3+V4)-V5
158      C
159      C     CALCULATE THE BODY FORCE ACCELERATIONS. THE SECOND TERM IS
160      C         THE NODE COUPLER IF ONE IS TO BE APPLIED.
161      C
162           ANCX=Y1*FLAGU*UAV
163           ANCY=Y1*FLAGV*VAV
164           AX=GR+ANCX
165           AY=GZ+ANCY
166      C
167      C     UPDATE THE VERTEX VELOCITIES DUE TO THE BODY FORCES AND THE
168      C         NODE COUPLER.  NOTE THAT THE WALLS ARE RIGID.
169      C
170           UTIL(IJ)=(U(IJ)+DT*AX)*XX
171           VTIL(IJ)=(V(IJ)+DT*AY)*YY
172      C
173      C     SAVE THE PART OF THE VELOCITY CHANGE DUE TO THE NODE COUPLER
174      C
175           ANCU(IJ)=ANCX*XX*DT
176           ANCV(IJ)=ANCY*YY*DT
177           IJ=IPJ
178           IJP=IJP+NQ
179        90 IJM=IJM+NQ
180           CALL LOOP
181       100 CONTINUE
182           CALL DONE
183      C
184      C     CALCULATE THE CHANGE TO THE VERTEX VELOCITIES DUE TO THE PRESSURE
185      C         TERM IN THE MOMENTUM EQN.  LOOP OVER ALL CELLS,NOT VERTICES.
186      C
187           CALL START
188           DO 160 J=2,JP1
189           DO 150 I=1,IBAR
190           ROL(IJ)=0.
191           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
```

```
249      C
250            CAREA=.5*XY
251      C
252      C     SAVE THE AREA FOR USE IN THE TURBULENCE CALCULATION
253      C         IN TRBCOR
254      C
255            AREA(IJ)=CAREA
256      C
257      C     RXY IS THE RECIPROCAL OF XY (HALF THE RECIPROCAL OF THE CELL AREA)
258      C
259            RXY=1./XY
260      C
261      C     U(V)IJ=U(V)I-U(V)J
262      C
263            U24=U2-U4
264            U31=U3-U1
265            V24=V2-V4
266            V31=V3-V1
267      C
268      C     DU(V)DR(Z) IS THE DIFFERENCE FORM FOR THE DERIVATIVE OF THE
269      C         RADIAL (AXIAL) VELOCITY WITH RESPECT TO R (Z) AT THE
270      C         CELL CENTER
271      C
272            DUDR=RXY*(U24*Y31-U31*Y24)
273            DUDY=RXY*(X24*U31-X31*U24)
274            DVDR=RXY*(V24*Y31-V31*Y24)
275            DVDY=RXY*(X24*V31-X31*V24)
276      C
277      C     COMPUTE DIV(VELOCITY) - VOLUME DILATATION
278      C
279            D=(DUDR+DVDY)*(1.+UDR*DT)+UDR
280      C
281      C     SAVE THE VELOCITY DIVERGENCE
282      C
283            DELSM(IJ)=D
284      C
285      C     DEFINE THE CELL AREA AS BEING AN AVERAGE DX TIMES AN AVERAGE DY -
286      C         CAREA = AVX * AVY. THEN DEFINE AVX(Y) AS THE SQUARE ROOT OF
287      C         THE AVERAGE OF THE SQUARES OF THE CELL DIAGONAL X(Y)
288      C         COMPONENTS. THEN,ONE WAY OF DETERMINING AVX(Y) GIVEN AVY(X)
289      C         IS   AVX(Y)= CAREA/((Y(X)D1**2+Y(X)D2**2)/2.)**.5 WHERE
290      C         Y(X)D1,2 ARE THE CELL DIAGONAL Y(X) COMPONENTS. THUS,
291      C         XXA(YYA) IS THE SQUARE OF THE AVERAGE CELL DX(Y)
292      C         DETERMINED BY THE PRECEDING METHOD.
293      C
294            XXA=2./(Y24**2+Y31**2)*CAREA**2
295            YYA=2./(X24**2+X31**2)*CAREA**2
296      C
297      C     DKLSM IS 2.*DT*(1./AVX**2+1./AVY**2) FOR USE IN THE PRESSURE
298      C         ITERATION
299      C
300            DKLSM(IJ)=2.*DT*(XXA+YYA)/(XXA*YYA)
301      C
302      C     XX AND YY ARE THE ACTUAL AVX AND AVY
303      C
304            XX=SQRT(XXA)
305            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(0.0,)*AK*LAM=TWTHRD*ROSIG*D
      C
      C     MUO2 IS HALF MU EFFECTIVE
      C
            MUO2=0.5*(AK*MU+ROSIG)
            IF(TURB) CAPGAM(IJ)=2.*(OUOR**2+OVOY**2+.5*(OUOY+OVOR)**2+UOR**2)
      C
      C     PIXX,PIYY,PIXY,AND PITHTA ARE COMPONENTS OF THE VISCOSITY
      C         STRESS TENSOR
      C
            PIXX=4.*MUO2*OUOR+LAMD
            PIYY=4.*MUO2*OVOY+LAMD
            PIXY=2.*MUO2*(OUOY+OVOR)
            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         SUBROUTINE 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)
```

```
363           UTIL(IJP)=UTIL(IJP)-DTO2M3*(XX+R3*YY+PITH)
364           XX=HR13*(PIXY*X31-PIXX*Y31)
365           YY=Y31*TUQYY
366           UTIL(IPJP)=UTIL(IPJP)+DTO2M2*(XX+R2*YY-PITH)
367           UTIL(IJ)=UTIL(IJ)-DTO2M4*(XX+R4*YY+PITH)
368           PYYMP=PIYY-TUQYY
369           XX=HR24*(PYYMP*X24-PIXY*Y24)
370           VTIL(IPJ)=VTIL(IPJ)+DTO2M1*XX
371           VTIL(IJP)=VTIL(IJP)-DTO2M3*XX
372           XX=HR13*(PYYMP*X31-PIXY*Y31)
373           VTIL(IPJP)=VTIL(IPJP)+DTO2M2*XX
374           VTIL(IJ)=VTIL(IJ)-DTO2M4*XX
375           IJ=IPJ
376       150 IJP=IPJP
377           CALL LOOP
378       160 CONTINUE
379           CALL DONE
380     C
381     C     GO SET THE BOUNDARY CONDITIONS
382     C
383           CALL BC(1)
384           RETURN
385           END
```

---

```
  1           SUBROUTINE YPH2
  2     C
  3     C     ROUTINE TO DO YAQUI PRESSURE ITERATION
  4     C
  5     C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
  6     C     MODIFIED BY J.L.NORTON,LASL T-3,1975
  7     C
  8     *     ----- BEGIN COMDECK YSTORE      -----
  9     *     ----- BEGIN COMDECK YAQDIM      -----
 10           DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
 11          1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
 12          2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
 13          3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),OLSROQ(1),CAPGAM(1),TUQ
 14          4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
 15          5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
 16          6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
 17     *     ----- END COMDECK YAQDIM      -----
 18     *     ----- BEGIN COMDECK YAQSC      -----
 19           LOGICAL RESTRT,FILM,PAPER,TURB
 20           REAL LAM,MU
 21     C     COMMON/YSC1/AASC(NSCP1)
 22           COMMON/YSC1/AASC(9600)
 23           COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
 24          1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
 25          2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GROVEL,GZ,GZP,I,IBAR,
 26          3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
 27          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,ROT,REZRON,REZSIE,REZYQ,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,OTSV,OTLAST,FIYBO,IYBO,YCNVLO,
     3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
     4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,OTR,TMASS,OTVSAV,OTCSAV,IOTV
     5 ,JDTV,IOTC,JOTC,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,IFO
*     ***** 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),OELSM,RCSQ,MP),(AASC
     1 (8),E,ETIL,AREA,XR13K),
     2 (AASC(15),SIE),(AASC(16),PMO,OKLSM,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),OLSROI,Y13K),(AASC(24),GZSV
     7 ),(AASC(25),OLSROQ,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.TSTRTO) 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.+UDR*DT)+UDR
      S=ROT*(ROL(IJ)-MTIL(IJ))+ROL(IJ)*D
      RA=RCSQ(IJ)*(ROT+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=DT02*DP
      DT02M1=XX*RM(IPJ)
      DT02M2=XX*RM(IPJP)
      DT02M3=XX*RM(IJP)
      DT02M4=XX*RM(IJ)
      UL(IPJ)=U1+DT02M1*R1*Y24
      UL(IPJP)=U2+DT02M2*R2*Y31
      UL(IJP)=U3-DT02M3*R3*Y24
      UL(IJ)=U4-DT02M4*R4*Y31
      VL(IPJ)=V1-DT02M1*XR24
      VL(IJ)=V4-DT02M4*XR13
      VL(IPJP)=V2+DT02M2*XR13
      VL(IJP)=V3+DT02M3*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=IPO,IFO,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.AMSOEN 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),OLSROI(1),OLSROQ(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 IOTQ,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,YCNVLO,
           3 XCNVLO,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IOTV
           5 ,JOTV,IOTC,JOTC,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,IPO
      *     ----- 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),OLSROI,Y13K),(AASC(24),GZSV
           7 ),(AASC(25),OLSROQ,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
      OTV=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)
```

```
119           HR24=0.5*(R2+R4)
120           U12=U1+U2
121           U23=U2+U3
122           U34=U3+U4
123           U41=U4+U1
124           U24=U2+U4
125           U13=U1+U3
126           V12=V1+V2
127           V23=V2+V3
128           V34=V3+V4
129           V41=V4+V1
130           V24=V2+V4
131           V13=V1+V3
132           XY=X24*Y31-X31*Y24
133           UOR=(U12+U34)*UG(IJ)
134           U24M=U2-U4
135           U13M=U1-U3
136           V24M=V2-V4
137           V13M=V1-V3
138           CAREA=.5*XY
139           RXY=VG(IJ)
140           DUDX=RXY*(U24M*Y31+U13M*Y24)
141           DUDY=RXY*(-X24*U13M-X31*U24M)
142           DVDX=RXY*(V24M*Y31+V13M*Y24)
143           DVDY=RXY*(-X24*V13M-X31*V24M)
144           D=(DUDX+DVDY)*(1.+UOR*DT)+UOR
145           XX=SQRT(2./(Y24**2+Y31**2))*CAREA
146           YY=SQRT(2./(X24**2+X31**2))*CAREA
147           IF(KXI.LT.0) GO TO 10
148           AK=RO(IJ)**KXI
149           GO TO 20
150        10 UD4=U4**2+V4**2
151           VD4=AMAX1(ABS(U4*XX),ABS(V4*YY))
152           AK=RO(IJ)*COLAMU*(DT02*UD4+VD4*0.7)
153        20 ALAM=AK*LAM
154           AMU=AK*MU
155           LAMD=AMIN1(D,0.)*ALAM
156           MUD2=.5*AMU
157           FMUD2=4.*MUD2
158           PIXX=FMUD2*DUDX+LAMD
159           PIYY=FMUD2*DVDY+LAMD
160           PIXY=2.*MUD2*(DUDY+DVDX)
161           PITH=0.25*XY*(FMUD2*UOR+LAMD*CYL)
162           XX1=HR24*(PIXY*X24-PIXX*Y24)
163           XX2=HR13*(PIXY*X31-PIXX*Y31)
164           XX3=-HR24*PIXY*Y24
165           XX4=-HR13*PIXY*Y31
166           XX=XX*XX
167           YY=YY*YY
168           TEMPJ=ALAM+2.*AMU
169           IF(TURB) TEMPJ=TEMPJ-RO(IJ)*SIG(IJ)*(TWTHRD*D-2.)
170           DQ=RO(IJ)*(1.-ANC)*XX*YY/(2.*TEMPJ*(XX+YY)+EM10)
171           DQ=ABS(DQ)
172           DTV=AMIN1(.5*DQ,DTV)
173           IF(DTVSAV.NE.DTV) ITDV=I
174           IF(DTVSAV.NE.DTV) JTDV=J
175           DTVSAV=DTV
```

```
176            XR13=XR13K(IJ)
177            XR24=XR24K(IJ)
178            QX=P(IJ)
179            RRO=1./RO(IJ)
180            RMC=RRO*RVOL(IJ)
181            QY=QX-PIYY
182            DELE=0.25*DT*RVOL(IJ)*((U1L+U1)*R1*Y24*QX+(U2L+U2)*R2*Y31*QX-(U3L+
183           1 U3)*R3*Y24*QX-(U4L+U4)*R4*Y31*QX-(V1L+V1)*XR24*QY+(V2L+V2)*XR13*
184           2 QY+(V3L+V3)*XR24*QY-(V4L+V4)*XR13*QY)
185            DELE=DELE+0.25*DT*RVOL(IJ)*((U1L+U1)*(XX1-PITH)+(U2L+U2)*(XX2-PITH
186           1 )-(U3L+U3)*(XX1+PITH)-(U4L+U4)*(XX2+PITH)+(V1L+V1)*XX3+(V2L+V2)*
187           2 XX4-(V3L+V3)*XX3-(V4L+V4)*XX4)
188            ETIL(IJ)=SIE(IJ)-DELE*RRO
189            IJ=IPJ
190         30 IJP=IPJP
191            CALL LOOP
192         40 CONTINUE
193            CALL DONE
194            CALL START
195            IF(.NOT.TURB) GO TO 50
196            TNEGT=TNEG
197            TK2=0.
198            TI2=0.
199            TP2=0.
200            TQ2=0.
201            TUPOT=0.
202            TUPOTA=0.
203         50 CONTINUE
204            DO 130 J=2,JP2
205            DO 120 I=1,IP1
206            IF(.NOT.TURB) GO TO 60
207            FF=0.5/RM(IJ)
208            TK2=TK2+FF*(UL(IJ)**2+VL(IJ)**2)
209            TP2=TP2-FF*GZ*DT*(VL(IJ)+V(IJ))
210         60 CONTINUE
211            IMJ=IJ-NQ
212            IPJ=IJ+NQ
213            IMJM=IJM-NQ
214            VTEMP=-.125/RM(IJ)*(ANCU(IJ)*(UTIL(IJ)+U(IJ))+ANCV(IJ)*(VTIL(IJ)+V
215           1 (IJ)))
216            XX=1.
217            YY=1.
218            IF((I.EQ.1).OR.(I.EQ.IP1)) XX=0.
219            IF((J.EQ.2).OR.(J.EQ.JP2)) YY=0.
220            IF(XX*YY.EQ.0.) GO TO 70
221            E(IMJM)=E(IMJM)+VTEMP*RVOL(IMJM)/RO(IMJM)
222            E(IJM)=E(IJM)+VTEMP*RVOL(IJM)/RO(IJM)
223            E(IMJ)=E(IMJ)+VTEMP*RVOL(IMJ)/RO(IMJ)
224            E(IJ)=E(IJ)+VTEMP*RVOL(IJ)/RO(IJ)
225            GO TO 110
226         70 IF(I.NE.1) GO TO 80
227            IF((J.EQ.2).OR.(J.EQ.JP2)) GO TO 110
228            E(IJM)=E(IJM)+2.*VTEMP*RVOL(IJM)/RO(IJM)
229            E(IJ)=E(IJ)+2.*VTEMP*RVOL(IJ)/RO(IJ)
230            GO TO 110
231         80 IF(I.NE.IP1) GO TO 90
232            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
```

```
290           IJP=IPJP
291       180 IJM=IJM+NQ
292           CALL LOOP
293       190 CONTINUE
294           CALL DONE
295           IF(.NOT.TURB) GO TO 200
296           TOQ=TQ2+TUQENG
297           TE=TK+TI+TUQENG
298           DELTAE=EP1+TUPOT+TUPOTA
299           DELEWR=DELTAE
300           FIDEL=DELEWR-SAV1
301           IF(TOQSAV.NE.0.) ERATIO=FIDEL/TOQSAV
302           TOQSAV=TDQ
303           SAV1=DELTAE
304       200 CONTINUE
305           IF(GROVEL.GT.1.99) GO TO 250
306           CALL START
307           DO 220 J=2,JP2
308           DO 210 I=1,IP1
309           UG(IJ)=0.5*GROVEL*(UL(IJ)+U(IJ))
310           VG(IJ)=0.5*GROVEL*(VL(IJ)+V(IJ))
311       210 IJ=IJ+NQ
312           CALL LOOP
313       220 CONTINUE
314           CALL DONE
315           CALL START
316           DO 240 J=2,JP2
317           DO 230 I=1,IP1
318           X(IJ)=X(IJ)+UG(IJ)*DT
319           Y(IJ)=Y(IJ)+VG(IJ)*DT
320           R(IJ)=X(IJ)*CYL+OMCYL
321       230 IJ=IJ+NQ
322           CALL LOOP
323       240 CONTINUE
324           CALL DONE
325           GO TO 260
326       250 CALL REZONE
327       260 CONTINUE
328           RE URN
329           END
```

---

```
  1           SUBROUTINE YPLOT
  2     C
  3     C     ROUTINE TO DO PLOTTING FOR YAQUI
  4     C
  5     C     ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
  6     C     MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1975
  7     C
  8     *     ***** BEGIN COMDECK PARAM      *****
  9           COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPF8,NP1,NP2,NLCP1,NLCP2,
 10          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,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,ROT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
           3 FREZYT,FREZYB,ROMFR,T,THIRO,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,TWTHRO,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
           5 ,JOTV,IDTC,JOTC,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       *****
            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(GROVFL.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
```

```
125           CALL AOV(1)
126           CALL ZONPLT
127     C
128     C     PUT LABELS ON THE ZONE PLOT AND OUT TO THE CYCLE SUMMARY PRINT
129     C
130           CALL LINCNT(59)
131           WRITE(IFO,220)
132           DO 10 IPX=6,IFO,6
133        10 WRITE(IPX,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
134           WRITE(IFO,260) JNM,NAME,T,NCYC
135        20 CONTINUE
136     C
137     C     SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO.
138     C         OTHERWISE,PLOT VELOCITIES SCALED TO THE MAXIMUM VELOCITY.
139     C
140           IF(VMAX.GT.EM10) CALL VELPLT(VMAX,0)
141     C
142     C     THE FULL PLOTS HAVE BEEN MADE. ADJUST PLOTS NOW SO THAT THE
143     C         REGION DISPLAYED INCLUDES THAT PART OF THE PROBLEM TWO
144     C         FIREBALL RADII ABOVE THE TOP OF THE FIREBALL TO THREE RADII
145     C         BELOW THE BOTTOM OF THE FIREBALL
146     C
147           YUP=PTOP+2.*PRITE
148           YLB=PBOTM-3.*PRITE
149           IF(YUP.GT.YT) YUP=YT
150           IF(YLB.LT.YB) YLB=YB
151     C
152     C     READJUST THE CARTESIAN TO RASTER RATIOS TO ENCOMPASS THE NEW
153     C         REGION BUT SAVE THE OLD COORDINATES
154     C
155           FIYBO=FIYB
156           IYBO=IYB
157           FIYB=900.
158           IYB=900
159           YCONVP=FLOAT(IYT-IYB)/(YUP-YLB)
160           XCONVP=XCONV*YCONVP/YCONV
161           FIXLO=FIXL
162           IXLO=IXL
163           FIXRO=FIXR
164           IXRO=IXR
165           FIXL=65.
166           IXL=65
167           FIXR=660.
168           IXR=660
169           YCNVLO=YCONV
170           YCONV=YCONVP
171           XCNVLO=XCONV
172           XCONV=XCONVP
173     C
174     C     DO THE PLOTS AGAIN FOR THE SMALLER REGION
175     C
176           IF(EULER.AND.NCYC.NE.0) GO TO 30
177           CALL ADV(1)
178           CALL ZONPLT
179     C
180     C     PUT LABELS ON THE ZONE PLOT
181     C
```

```
182           CALL LINCNT(59)
183           WRITE(IFO,230)
184           WRITE(IFO,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
185           WRITE(IFO,260) JNM,NAME,T,NCYC
186     C
187     C     DRAW THE PLOTTING RECTANGLE AND LABEL IT
188     C
189           CALL TICBOX
190        30 CONTINUE
191     C
192     C     SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO
193     C
194           IF(VMAX.LT.EM10) GO TO 40
195     C
196     C     PLOT VELOCITIES SCALED TO MAXIMUM VELOCITY
197     C
198           CALL VELPLT(VMAX,0)
199           CALL TICBOX
200     C
201     C     PLOT UNSCALED VELOCITIES
202     C
203           CALL VELPLT(VMAX,1)
204           CALL TICBOX
205        40 CONTINUE
206     C
207     C     SECTION TO DO CONTOUR PLOTS
208     C
209     C     L=1, DENSITY (ISOPYCNICS)
210     C     L=2, SPECIFIC INTERNAL ENERGY (ISOTHERMS)
211     C     L=3, VORTICITY
212     C     L=4, VELOCITY MAGNITUDE
213     C     L=5, SPECIFIC TURBULENCE ENERGY
214     C     L=6, RATIO OF SPEC. TURB. ENERGY TO SPEC. KINETIC ENERGY
215     C     L=7, TURBULENCE VISCOSITY (SIGMA)
216     C     L=8, CONCENTRATION X RADIUS
217     C
218     C     IF TURBULENCE IS NOT ON,DO NOT PLOT THE TURBULENCE QUANTITIES
219     C
220           LP=8
221           IF(.NOT.TURB) LP=4
222     C
223     C     PREPARE THE PLOTTING ARRAYS
224     C
225           DO 200 L=1,LP
226     C
227     C     CHECK FOR VORTICITY PLOT
228     C
229           IF(L.NE.3) GO TO 50
230     C
231     C     YES, SPECIAL SUBROUTINE PREPARES VORTICITY.
232     C
233           CALL GETOMG
234           GO TO 190
235        50 CONTINUE
236           CALL START
237           DO 180 J=2,JP1
238           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
```

```
296            IF(ABS(CQ(IJ)).LT.1.E-50) CQ(IJ)=0.
297            IJP=IJP+NQ
298          170 IJ=IJ+NQ
299            CALL LOOP
300          180 CONTINUE
301            CALL DONE
302          190 CONTINUE
303      C
304      C     DECIDE WHETHER TO REQUEST LOGARITHMIC OR LINEAR CONTOUR INCREMENTS
305      C
306      C     PLOTS 1 AND 2 ARE  LOGARITHMIC - THE OTHERS LINEAR
307      C
308            ILOG=1
309            IF(L.GT.2) ILOG=0
310      C
311      C     GO DO THE PLOT
312      C
313      C     ITITLE IS THE PLOT TITLE
314      C     INWT IS THE NO. OF COMPUTER WORDS IN THE TITLE
315      C
316            CALL CONTOR(ILOG,ITITLE(1,L),INWT(L))
317          200 CONTINUE
318      C
319      C     RESTORE THE OLD RASTER VALUES
320      C
321            FIYB=FIYBO
322            IYB=IYBO
323            YCONV=YCNVLO
324            XCONV=XCNVLO
325            FIXR=FIXRO
326            FIXL=FIXLO
327            IXR=IXRO
328            IXL=IXLO
329            CALL SECOND(TP)
330            WRITE(59,240) TP
331            RETURN
332      C
333          210 FORMAT(1H ,20HBEGIN YPLOT AT CP = ,F10.4)
334          220 FORMAT(10H ALL ZONES)
335          230 FORMAT(29H ZONES IN THE FIREBALL REGION)
336          240 FORMAT(1H ,18HEND YPLOT AT CP = ,F10.4)
337          250 FORMAT(7H ORMIN=1PE12.5,7H ORMAX=E12.5,7H DZMIN=E12.5/7H DZMAX=E12
338             1 .5,4H XR=E12.5,4H YB=E12.5,4H YT=E12.5)
339          260 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,I5)
340            END
```

---

```
 1            SUBROUTINE ZONPLT
 2      C
 3      C     ROUTINE TO DO FULL AND PARTIAL ZONE PLOTS
 4      C
 5      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,GROVEL,GZ,GZP,I,IBAR,
           3 IOTO,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,TUR8,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,OTSV,OTLAST,FIYBO,IYBO,YCNVLO,
           3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
           4 ROMFXR,ROMFYT,ROMFYB,JOUMP,TWTHRO,TE,DTR,TMASS,OTVSAV,OTCSAV,IDTV
           5 ,JOTV,IOTC,JOTC,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      -----
      *     ----- 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
```

```
120      C          BOTTOM SIDE
121      C
122           IF(JB.NE.0) GO TO 10
123           JB=J
124           GO TO 20
125        10 CONTINUE
126           IF(J.NE.JB) GO TO 30
127        20 CONTINUE
128           CALL DRV(IX4,IY4,IX1,IY1)
129        30 CONTINUE
130      C
131      C     DRAW THE RIGHT SIDE
132      C
133           CALL DRV(IX1,IY1,IX2,IY2)
134      C
135      C     DRAW THE TOP SIDE
136      C
137           CALL DRV(IX2,IY2,IX3,IY3)
138        40 IJ=IPJ
139        50 IJP=IPJP
140           CALL LOOP
141        60 CONTINUE
142           RETURN
143           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 Simulatio Technique (GEST) Experiments," Los Alamos Scientific Laboratory report LA-6154-MS (December 1975).


☆ U.S. GOVERNMENT PRINTING OFFICE: 1978—777-018/32