

LA-6261-M

Manual

C.3

CIC-14 REPORT COLLECTION
REPRODUCTION
COPY

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

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represented that its use would not infringe privately owned rights.

CONTENTS

I.	DESCRIPTION OF CODE	1
A.	Initial Zoning	1
B.	Fireball Initialization	2
C.	Particle Setup	4
D.	Particle Movement	5
E.	Turbulence	6
F.	Rezone Techniques	12
G.	Mixed Equation of State	12
H.	System Improvements	13
II.	DESCRIPTION OF INPUT	13
III.	RESULTS OF A SAMPLE CALCULATION	17
A.	Initial Conditions	17
B.	Turbulence Seeding Conditions	25
C.	Turbulence Equilibrium Conditions	27
D.	Moderately Late-Time Conditions (Eight Torus-Formation Times)	30
IV.	FLOW DIAGRAM	45
V.	CODE LISTING	47
A.	LTSS-Dependent Code	47
B.	CROS-Dependent Code	61
C.	KRONOS-Dependent Code	66
D.	KRONOS/CROS-Dependent Code	69
E.	System-Independent Code	76
REFERENCES		297



YAQUI USER'S MANUAL FOR
FIREBALL CALCULATIONS

by

J. L. Norton and H. M. Ruppel

ABSTRACT

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

I. DESCRIPTION OF CODE

The YAQUI code is a combination of two fluid-dynamical techniques, ALE and ICE. ALE is an acronym for Arbitrary-Lagrangian-Eulerian; using this method the finite difference mesh points can be moved with the fluid (Lagrangian), held stationary (Eulerian), or moved by some prescribed rezone algorithm (Arbitrary). ICE stands for Implicit Continuous-fluid Eulerian; because the hydrodynamic equations are solved implicitly, the technique can be used to solve for flow at all speeds. The analyses of these techniques were presented by Harlow and Amsden¹ and by Hirt et al.,² and an initial version of the code was described by Amsden and Hirt.³

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

A. Initial Zoning

An initial YAQUI zoning setup is shown in Fig. 1. The initial grid is determined by the parameters DR, DZ, IBAR, JBAR, and YB. First, a uniform grid of IBAR zones in the r-direction and JBAR zones in the z-direction with the bottom at $z = YB$ is generated. The zones are all DR cm wide by DZ cm high. Then the parameters FREZXR, FREZYB, and FREZYT are examined.

If any one of them is larger than 1, it is assumed that a region of nonuniform zoning will surround the uniform region. The parameters that describe the nonuniform region are IUNF, JUNF, JCEN, and REZYO. IUNF is the number of zones starting at the left that are to remain uniform. To the right of this region, the zone widths are related by

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

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

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

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

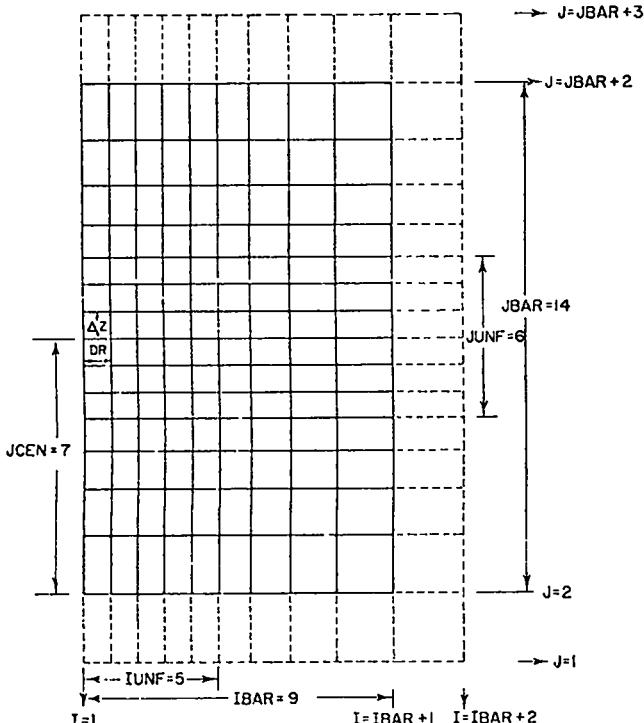


Fig. 1. Sample initial zoning setup.

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

The top of the J_{CEN}^{th} zone is redefined to be at the point $z = REZY0$. Because FREZYB will move the bottom of the mesh, it is necessary to recalculate YB based on JUNF, JCEN, REZY0, 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 fictitious 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 fictitious cells; and $J = JBAR + 3$ is at the top of the top fictitious 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 (v_i), and the specific internal energy and density of the zone (e_i and ρ_i , respectively) in the format 4E15.0. Cards are read until a blank card or an end-of-file is encountered. The inner radius of the first zone is assumed to be zero ($r_0 = 0$) as is the velocity at the center ($v_0 = 0$). 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

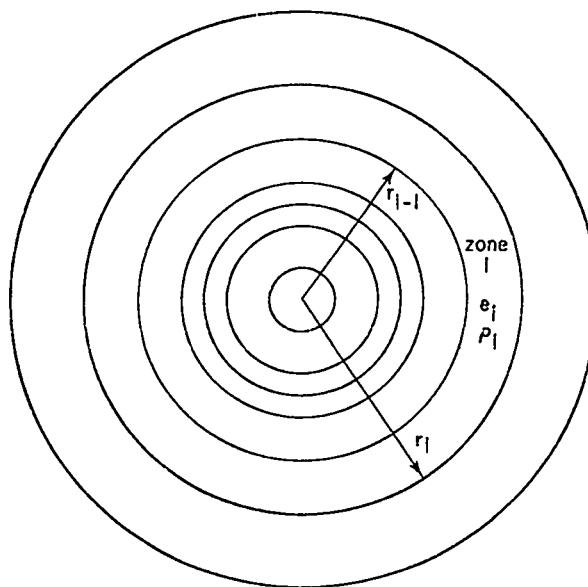


Fig. 2. Sample 1D fireball initial data configuration.

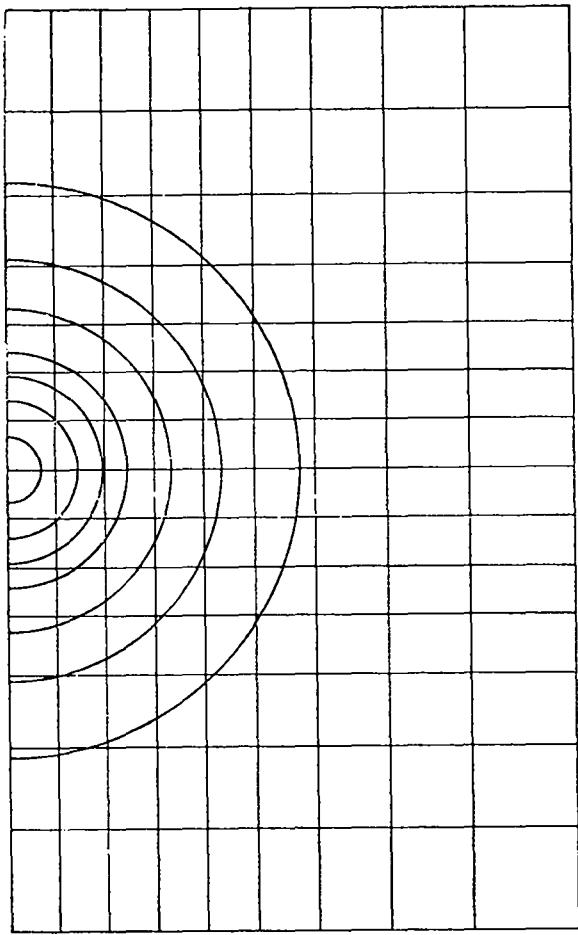


Fig. 3. Superposition of 1D fireball data on the 2D YAQUI grid.

θ_1 is 0° and θ_2 is 180° . $\Delta\theta$ is determined from $180^\circ/NTH$, where NTH is an input quantity; Δr 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.

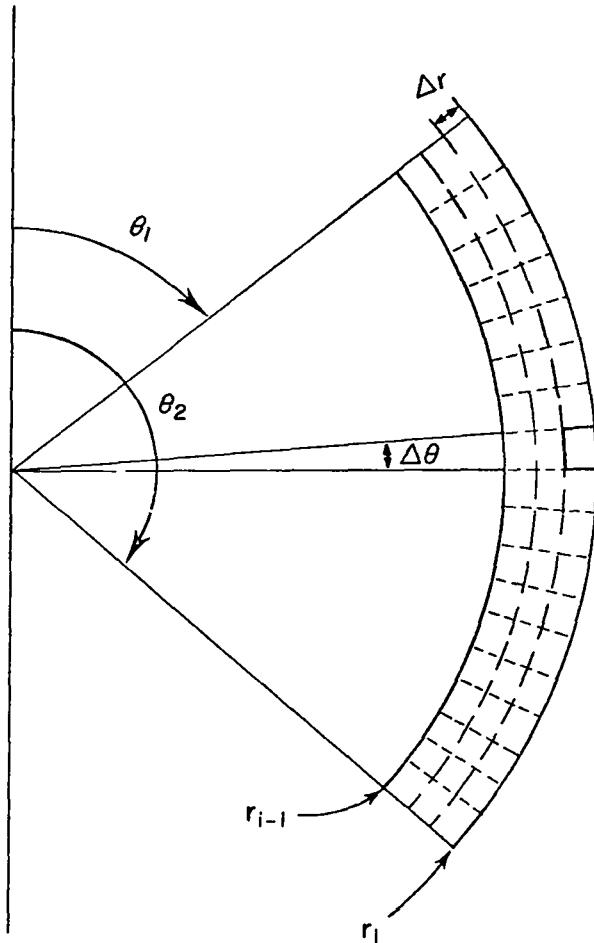


Fig. 4. Division of 1D fireball data into subzones for interpolation purposes.

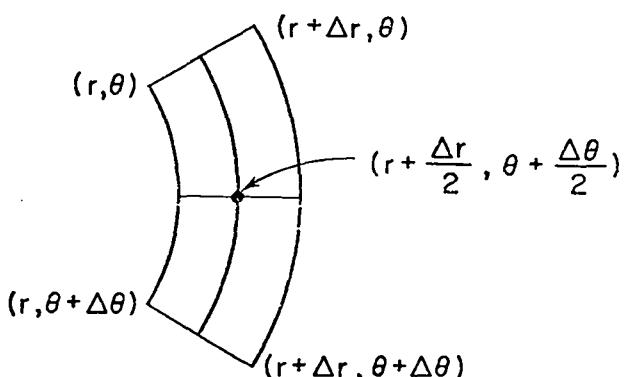


Fig. 5. A typical 1D Interpolation subzone.

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

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

Once the internal energy, mass, and momentum components are established for all YAQUI zones that fall completely or partially within the fireball data, the specific internal energy is determined by dividing the zone internal energy by the zone

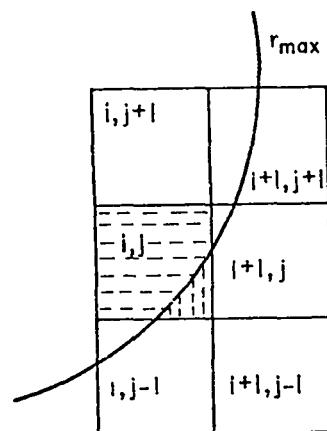


Fig. 6. Technique for handling YAQUI zones only partially within the fireball.

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 falls within the rectangular 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.

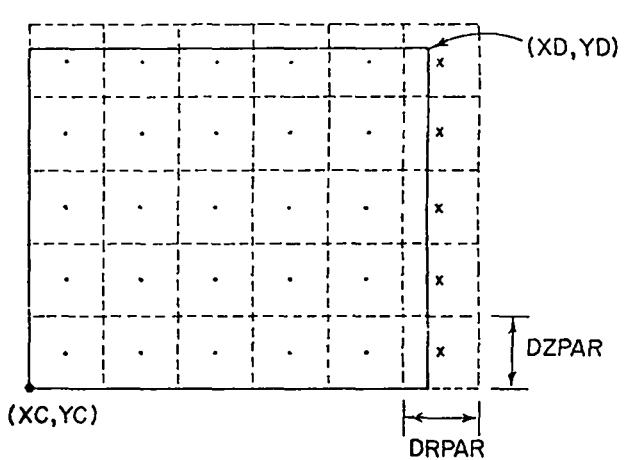


Fig. 7. Particle generation scheme.

D. Particle Movement

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

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

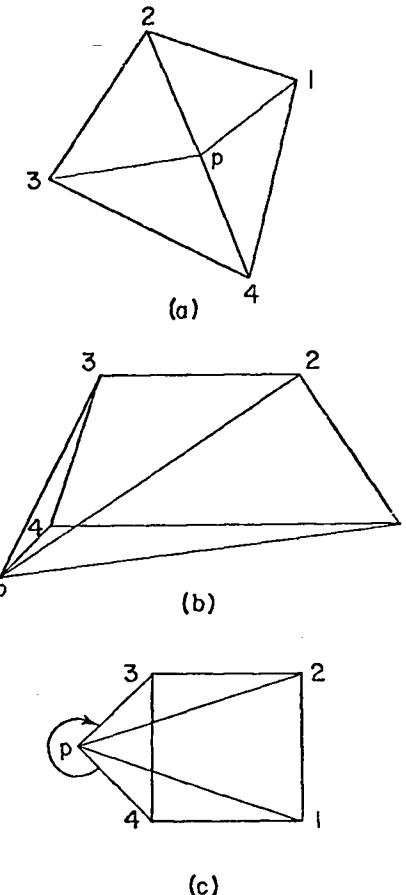


Fig. 8. Determination of particle positions.

The numbering of vertices is arbitrary 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 $\vec{x}_1, \dots, \vec{x}_4$ 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 $\vec{x}_1, \dots, \vec{x}_4$, 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 Pracht⁴ eliminates the separate testing and automatically defines the direction of search. If the particle lies in the cell in question, the values of η and θ obtained from Eq. (1) will both lie between 0 and 1. If either does not, the cell indicated by the values of η and θ is examined. For example, if $\eta = 1.2$, increase i by one and recalculate η and θ . This approach is neater and more efficient. In future versions of the particle transport, we plan to incorporate such a scheme into YAQUI.

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

E. Turbulence

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

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

the momentum equation

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

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

$$\begin{aligned} \frac{\partial \rho q}{\partial t} &= - \frac{\sigma}{\rho} \nabla \rho \cdot \nabla p + \rho \sigma \Gamma - \left(\frac{2}{3} \nabla \cdot \vec{u} + \frac{2\sigma}{\beta s^2} \right) \rho q \\ &- \frac{2}{3} \sigma \rho (\nabla \cdot \vec{u})^2 + \nabla \cdot \sigma \nabla \rho q. \end{aligned} \quad (6)$$

The turbulence viscosity is given by

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

For conciseness we have used the following definitions:

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

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

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

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

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

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

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

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

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

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 $\mu + \mu + \rho\sigma$, $\lambda + \lambda - \frac{2}{3}\rho\sigma$ and $p + p + \frac{2}{3}\rho q$. Similarly H_{ij} is calculated from ρ and \vec{u} at the old time level; this allows the stress tensor to be handled explicitly in Phase I simplifying the addition.

The term $\frac{\partial}{\partial t} \overline{\rho' u'_i}$ in the momentum equation is modeled, using the flux approximation, as $-\frac{\partial}{\partial t} \sigma \frac{\partial \rho}{\partial x_i}$; this

requires saving the two components of $\sigma \frac{\partial \rho}{\partial x_i}$ for each cell from the previous time step. Clearly such an approach centers the time derivative about $t - \frac{\delta t}{2}$ and not at the same time level as H_{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

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

In cylindrical geometry Γ takes the form

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

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

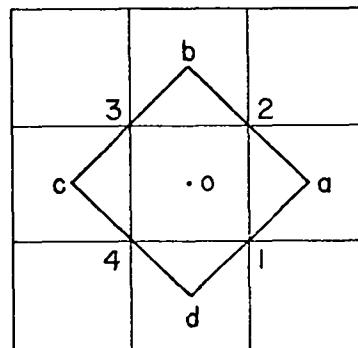


Fig. 9. Quantities involved in YAQUI derivative evaluation.

vertices 1-4 and then average to obtain $(\nabla p \cdot \nabla p)_o$. 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,

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

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

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

and

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

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,

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

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

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

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

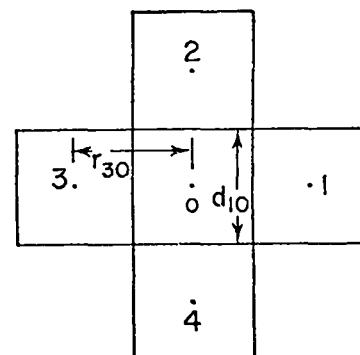


Fig. 10. Definition of quantities used in differencing of turbulence part of the concentration equation.

As far as convection is concerned, the quantity ρq is convected and, because the difference equations for the convection conserve the convected quantity identically, the turbulence energy is conserved in Phase III. The total energy, however, is not. In the initial code version³ 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 $p\delta V$ 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, $E = \rho I + \rho u^2 + pq$, 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

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

(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 initiated 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}_0 is described by the diffusion equation

$$\frac{\partial \rho}{\partial t} = \lambda \nabla^2 \rho + \delta(\vec{r} - \vec{r}_0), \quad (21)$$

where ρ is the density and λ is a constant diffusion coefficient. Define the quantities $\delta\vec{r} = \vec{r} - \vec{r}_0$, $\delta x = x - x_0$, $\delta y = y - y_0$, and $\delta z = z - z_0$. 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

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

with similar expressions for Y and Z .

In calculating the additional particle motion due to turbulence, consider the particle at time $t_0 = 0$ to be a massless point at \vec{r}_0 and use Eq. (22) as a probability distribution function for determining the position of the particle at time $t = t_0 + \delta t$. The turbulence viscosity then serves as the diffusion coefficient λ . In cylindrical coordinates one should solve Eq. (21) in a cylindrical basis

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

However, R cannot be determined in closed form and for small δt and δx , with $\delta x > \frac{\lambda\delta t}{x_0}$, R reduces to the form of X in Eq. (22). Thus, Eq. (22) can be used with confidence in cylindrical geometry as long as it is applied in its region of validity. Furthermore, δx must be small compared with a cell dimension so that the turbulence viscosity being used as the diffusion coefficient remains constant in the region and during the time considered.

The general Gaussian (normal) distribution is

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

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

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

$$y = \delta x, \quad (24a)$$

where we have replaced t by δt .

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

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

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

$$p(x) dx = 2 f(y) dy, \quad (26)$$

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

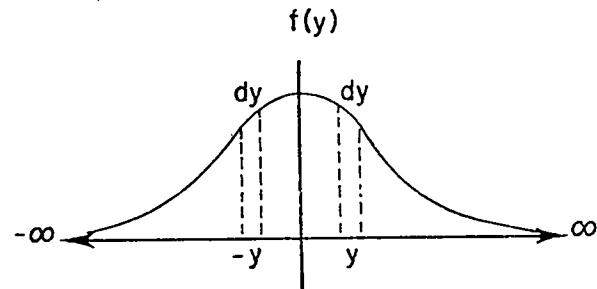
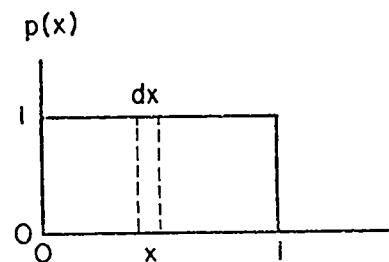


Fig. 11. (a) Uniform distribution, (b) Gaussian distribution.

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

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

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

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

By definition, the standard error function is

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

therefore,

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

and from Eq. (24),

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

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

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

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

The error function is monotonic increasing on the interval $[0,1]$; $\text{erf}(0) = 0$; 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

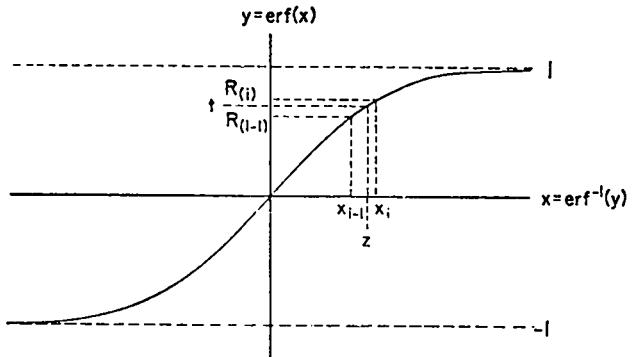


Fig. 12. Standard error function.

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

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

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

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

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

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

(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

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

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

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

or

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

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 Δr 's of the zones on the axis.

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

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

F. REZONE TECHNIQUES

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

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

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 $\vec{u}_g = \vec{u}_{\text{fluid}}$, the difference velocity vanishes, and there is no fluxing. In general the prescription 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

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

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), $c_{\text{meth}} + c_{\text{air}} = 1$ so that one only needs to keep track of c_{meth} . Here we use c_{meth} to refer to the concentration of combustion products and γ_{meth} to refer to the effective γ for the combustion products.

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

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

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

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

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

The pressure is then obtained from

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

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 ccl of the next card.
- An input record is terminated by a \$ anywhere on the card except for ccl 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 ccl 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
RESTART	LOGICAL	.TRUE., .FALSE.	.TRUE.
FILM	LOGICAL	.TRUE., .FALSE.	.TRUE.
PAPER	LOGICAL	.TRUE., .FALSE.	.FALSE.
WRAPUP	REAL (s)	0.≤WRAPUP≤∞	20.0
(a) RESTART - .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 RESTART = .TRUE., FILM = .TRUE.,
PAPER = .FALSE., WRAPUP = 30.\$

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

* * * INITIAL PROBLEM SETUP * * *

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

Record 3: Namelist name - CARDN

Namelist variables			
NAME	TYPE (units)	Possible values	Default
AO	Real (none)	0.≤AO≤1.	0.1
AOFAC	Real (none)	0.≤AOFAC≤.5	0.2
AOM	Real (none)	0.≤AOM≤1.	1.0
ANC	Real (none)	0.≤ANC≤.2	0.05
BO	Real (none)	0., 2.	0.0
CYL	Real (none)	0., 1.	1.0
DR	Real (cm)	0.≤DR≤∞	Must be specified
DT	Real (s)	0.≤DT≤∞	10^{-3}
DTØ	Real (s)	0.≤DTØ≤∞	1.0
DTØC	Real (s)	0.≤DTØC≤∞	10^{30}
DZ	Real (cm)	0.≤DZ≤∞	Must be specified
EPS	Real (none)	0.≤EPS≤1.	10^{-5}
FREZXR	Real (none)	1.≤FREZXR≤∞	1.0
FREZYB	Real (none)	1.≤FREZYB≤∞	1.0
FREZYT	Real (none)	1.≤FREZYT≤∞	1.0
GR	Real (cm/s ²)	-∞≤GR≤∞	0.0
GRDVEL	Real (none)	0., 1., 2.	2.0
GZ	Real (cm/s ²)	-∞≤GZ≤∞	-980.0

NAME	TYPE (units)	Possible values	Default
GZP	Real (cm/s ²)	-∞≤GZP≤∞	0.0
IBAR	Integer (none)	1≤IBAR<Storage	Must be specified
IEOF	Integer (none)	0, 1	0
IST	Integer (none)	-1, 1≤IST≤storage	-1
IUNF	Integer (none)	0≤IUNF≤IBAR	Must be specified
JBAR	Integer (none)	1≤JBAR<storage	Must be specified
JCEN	Integer (none)	1≤JCEN≤JBAR	Must be specified
JDUMP	Integer (none)	1≤JDUMP≤∞	999999
JUNF	Integer (none)	0≤JUNF≤JBAR	Must be specified
KXI	Integer (none)	-1, 0, 1	-1
LAM	Real (see Ref. 0.≤LAM≤3)	0.≤LAM≤3	0.6
MU	Real	0.≤MU≤∞	0.0
NCLST	Integer (none)	1≤NCLST≤∞	999999
NCQ	Integer (none)	-∞≤NCQ≤∞	-1
ØM	Real (none)	0.≤ØM≤2.	1.0
QLEVEL	Real (none)	.02	0.02
REXRØN	Real (gm/cm ³)	0.≤REXRØN≤∞	0.001
REZSIE	Real (cm ² /s ²)	0.≤REZSIE≤∞	2×10^{10}
REZYO	Real (cm)	YB≤REZYO≤∞	0.0
RMINEF	Real (cm)	0.≤RMINEF≤∞	50.0
T	Real (s)	0.≤T≤∞	0.0
TQ	Real (s)	0.≤TQ≤∞	0.0
TSTRTD	Real (s)	0.≤TSTRTD≤∞	1.0
TUQI	Real (cm ² ·s)	0.≤TUQI≤∞	0.0
TUSI	Real (cm)	0.≤TUSI<fireball radius	0.0
TWFIN	Real (s)	0.≤TWFIN≤∞	1.E30
WMAXEF	Real (none)	0.≤WMAXEF≤∞	2.0
YB	Real (cm)	0.≤YB≤∞	0.0
ZØRIG	Real (none)	0.≤ZØRIG≤∞	6.0
AO	- Amount of donor cell in momentum fluxing (0. is centered differencing; 1. is full donor cell).		
AOFAC	- Stability condition $u\Delta t/\Delta x \leq AOFAC$.		
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 = U., 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).	JDUMP - Frequency of dumps based on cycles. Dumps will occur when the cycle number is an even multiple of JDUMP.
DT	- Initial time step.	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).
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 $DT\bar{O}C(I-1) < T \leq DT\bar{O}C(I)$. $DT\bar{O}C(0) \equiv 0$.	KXI - Viscosity flag (see Ref. 3).
DZ	- Same as DR except zone height (see Sec. I.A).	LAM - Viscosity parameter (see Ref. 3).
EPS	- Convergence criterion in the pressure iteration (see Ref. 2).	MU - Viscosity parameter (see Ref. 3.)
FREZZR	- Geometric ratio of zone Δr 's in the right region of nonuniform zoning (see Sec. I.A).	NCLST - Cycle number after which to terminate the run.
FREZYB	- Geometric ratio of zone Δz 's in the bottom region of nonuniform zoning (see Sec. I.A).	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).
FREZYT	- Geometric ratio of zone Δz 's in the top region of nonuniform zoning (see Sec. I.A).	ØM - Relaxation parameter in the pressure iteration.
GR	- Body force acceleration in the radial direction.	QLEVEL - Phenomenological turbulence viscosity parameter relating to specific turbulence energy (see Ref. 5).
GRDVEL	- Type of rezone. GRDVEL = 0. is Eulerian, = 1. is Lagrangian, and =2. causes the rezone subroutine to be called.	REZRØN - Initial density of the ambient atmosphere at $y = REZY0$; the density of the atmosphere above and below REZY0 is determined by the condition that the entire atmosphere initially be in hydrostatic equilibrium.
GZ	- Body force acceleration in the axial direction (usually gravity).	REZSIE - Specific internal energy of the entire ambient atmosphere.
GZP	- Particle acceleration in the axial direction (not applied unless particles have mass.)	REZY0 - Center of y -coordinate of the region of uniform zoning (see Sec. I.A).
IBAR	- Number of real zones in the radial direction (see Sec. I.A).	RMINEF - Particles with $r \leq RMINEF$ are not subject to turbulent diffusion.
IEOF	- Input of the record 3 section is terminated by inputting a record with IEOF = 1.	T - Time at which the problem begins.
IST	- Number of particles whose positions are to be plotted as a function of time. If IST < 0, no particles are followed.	TQ - Time at which to seed the turbulence if NCQ = 0 (see NCQ).
IUNF	- Number of zones in the radial direction in the region of uniform zoning (see Sec. I.A).	TSTRTD - Time at which to start turbulent particle diffusion if the turbulence is on.
JBAR	- Number of real zones in the axial direction (see Sec. I.A).	TUQI - Proportionality constant for seeding turbulence energy. Should be chosen such that turbulence energy is a few percent of kinetic energy in any cell.
JCEN	- Number of real zones from bottom of the problem to center of region of uniform zoning (see Sec. I.A).	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 REZY0 = 4300. $
$CARDN REZRDN = 1.E-3 $
$CARDN REZSIE = 1.95E9 $
$CARDN T = .0083 $
$CARDN TWFIN = 30. $
$CARDN DT0 = .1, 1., 5. $
$CARDN DT0C = 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      IE0F = 1 $
```

III. RESULTS OF A SAMPLE CALCULATION

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

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

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

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

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

A. Initial Conditions

```
BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF., FIX IN TPBCOR
THERE WILL BE TURBULENCE
QLEVEL= 2.0000E-02
TU01= 5.0000E+00
TU11= 1.0000E+03
NCO= 0
TO= 5.0000E-01
TSTRIDE= 1.0000E+00
RMAXEF= 2.0000E+00
RMINEF= 5.0000E+01
IBAR= 30
JBAR= 45
IUNF= 5
JUNF= 10
JCEN= 15
DR= 2.0000E+02
DZ= 2.0000E+02
CYL= 1.0000E+00
GROVEL= 2.0000E+00
AO= 1.0000E-01
ADM= 1.0000E+00
BD= 0.
XXI= -1
MU= 0.
LAM= 6.0000E-01
OM= 1.0000E+00
EPS= 1.0000E-04
CR= 0.
CZ= 9.8000E+02
FREZXR= 1.08936E+00
FREZYT= 1.08936E+00
FREZYB= 1.08936E+00
ZORIG= 6.0000E+00
YB= 0.
REZYG= 4.3000E+03
REZRON= 1.0000E-03
REZSIE= 1.95930E+09
GZP= 9.8000E-02
T= 0.3000E-03
DT= 1.0000E-03
NCLST= 1
TMFIN= 3.0000E+01
PAPER=0
FILM=1
AMC= 2.0000E-01
ADFAC= 2.0000E-01
IST= 10
DT01(-10)= 1.0000E-01 1.0000E+00 5.0000E+00 0.
0. 0. 0. 0.
0. 0. 0. 0.
DT0C(-1)-10)= 1.0000E+00 1.0000E+01 3.0000E+01 0.
0. 0. 0. 0.
0. 0. 0. 0.
PSPARTN DRPAR=100..DZPAR=100..YC=4300..XD=1000..YD=0..XC=0. $
DRPAR= 1.0000E-02 DZPAR= 1.0000E+02 XC= 0.
YC= 4.3000E+03 XD= 1.0000E+03 YD= 0.
PSPARTN DRPAR=0. $
150 PARTICLES GENERATED, WITH TOTAL MASS= 0.
PSIRE FOFILE=.TRUE. $

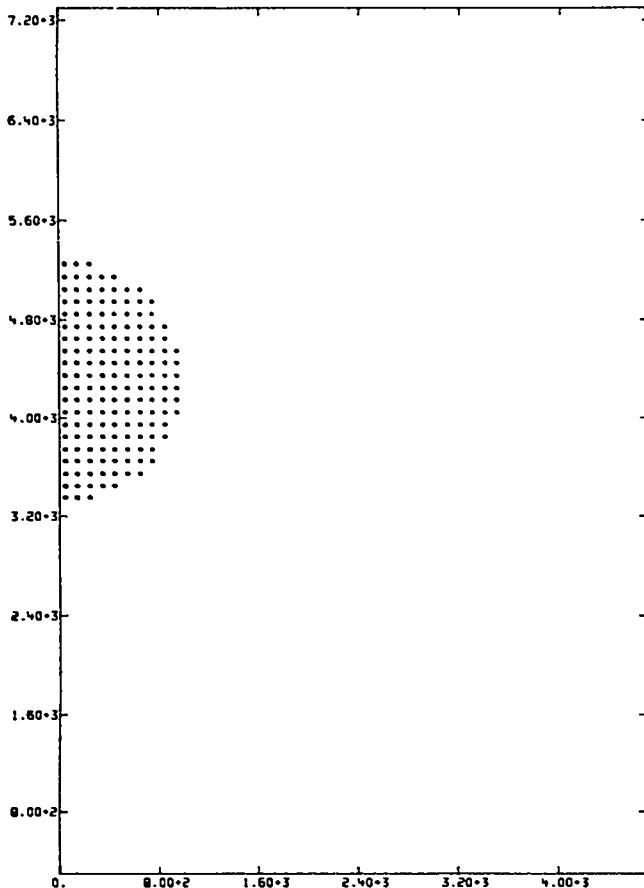
FIREBALL INPUT WILL BE FROM FILE 3
```

MESHMOR FIREBALL INPUT

U	E	RHO	X	P	PRHO
3.7424474E+03	7.1150927E+10	1.9059535E-04	8.5614114E+00	1.5978265E+06	1.5978265E+06
7.427278E+03	6.8652673E+10	2.075646E+04	1.6904335E+01	1.6002742E+05	1.6002742E+05
1.1017913E+04	6.5330351E+10	2.2051980E+04	2.4959608E+01	1.6034271E+05	1.6034271E+05
1.5978185E+04	6.4165566E+10	2.2570703E+04	3.3018635E+01	1.6076643E+05	1.6076643E+05
1.8224794E+04	6.4177399E+10	2.2653017E+04	4.1125711E+01	1.61161628E+05	1.61161628E+05
2.1052264E+04	6.4378798E+10	2.2683715E+04	4.9262853E+01	1.6222339E+05	1.6222339E+05
2.5396177E+04	6.4492809E+10	2.2752965E+04	5.7399511E+01	1.6312426E+05	1.6312426E+05
2.8897839E+04	6.4526913E+10	2.2857219E+04	6.5519210E+01	1.6401362E+05	1.6401363E+05
3.2108888E+04	6.4551871E+10	2.2948266E+04	7.3825620E+01	1.6477522E+05	1.6477521E+05
3.514531E+04	6.4572301E+10	2.3008191E+04	8.1727593E+01	1.6529134E+05	1.6529134E+05
3.7904370E+04	6.4562536E+10	2.3026145E+04	8.9836791E+01	1.6539651E+05	1.653952E+05
4.0391666E+04	6.4599287E+10	2.2985509E+04	9.7968690E+01	1.6505087E+05	1.6505087E+05
4.2650180E+04	6.4516297E+10	2.2899916E+04	1.0613698E+02	1.6422730E+05	1.6422731E+05
4.4720316E+04	6.4615766E+10	2.2750392E+04	1.1413512E+02	1.6300991E+05	1.6300991E+05
4.6645869E+04	6.4399621E+10	2.2566698E+04	1.2262183E+02	1.6143934E+05	1.6143934E+05
4.8464773E+04	6.4338951E+10	2.2346126E+04	1.3095664E+02	1.5960185E+05	1.5960184E+05
5.0196018E+04	6.4273095E+10	2.2093046E+04	1.3938314E+02	1.5751189E+05	1.5751189E+05
5.18172022E+04	6.4181020E+10	2.1824969E+04	1.4784309E+02	1.5523834E+05	1.5523834E+05
5.3382969E+04	6.4085894E+10	2.1536178E+04	1.5640135E+02	1.5260966E+05	1.5260956E+05
5.5049004E+04	6.3961910E+10	2.1291419E+04	1.6503797E+02	1.5026514E+05	1.5026514E+05
5.6562111E+04	6.3873930E+10	2.0934356E+04	1.7375626E+02	1.4766793E+05	1.4766792E+05
5.8020308E+04	6.3738324E+10	2.0613250E+04	1.8256091E+02	1.4500901E+05	1.4500901E+05
5.95456217E+04	6.3651833E+10	2.0281039E+04	1.9145599E+02	1.4231457E+05	1.4231457E+05
6.0876553E+04	6.3599576E+10	1.9949170E+04	2.0044101E+02	1.3957321E+05	1.3957321E+05
6.2296085E+04	6.3420689E+10	1.9604256E+04	2.0952238E+02	1.3673251E+05	1.3673251E+05
6.3716637E+04	6.3299368E+10	1.9242942E+04	2.1870822E+02	1.3378075E+05	1.3378075E+05
6.5171513E+04	6.3154868E+10	1.8880368E+04	2.2799977E+02	1.3078363E+05	1.3078363E+05
6.6618302E+04	6.3017089E+10	1.8520866E+04	2.3733724E+02	1.2783904E+05	1.2783904E+05
6.7955213E+04	6.2877213E+10	1.8182404E+04	2.4689197E+02	1.2505825E+05	1.2505825E+05
6.9030814E+04	6.2765359E+10	1.7879916E+04	2.5646779E+02	1.2261441E+05	1.2261441E+05
6.9496811E+04	6.2702621E+10	1.7657756E+04	2.6608521E+02	1.2087810E+05	1.2087810E+05
6.7215425E+04	6.2739456E+10	1.7369701E+04	2.7561025E+02	1.2122939E+05	1.2122939E+05
6.4305656E+04	6.2974556E+10	1.8224201E+04	2.8489098E+02	1.2552554E+05	1.2552554E+05
6.0032417E+04	6.3620777E+10	1.9059656E+04	2.9358387E+02	1.3239276E+05	1.3239276E+05
5.7894019E+04	6.3459537E+10	1.9650939E+04	3.0209328E+02	1.3713999E+05	1.3713999E+05
5.5119095E+04	6.3680516E+10	2.0362195E+04	3.1031970E+02	1.4298534E+05	1.4298534E+05
5.3192959E+04	6.3895024E+10	2.0939162E+04	3.1834928E+02	1.4783551E+05	1.4783551E+05
5.2393687E+04	6.7961905E+10	2.1131638E+04	3.2632708E+02	1.4945335E+05	1.4945335E+05
5.3249521E+04	6.3951020E+10	2.1110238E+04	3.3436146E+02	1.4926047E+05	1.4926047E+05
5.1030180E+04	6.4054493E+10	2.1455131E+04	3.4266483E+02	1.5211647E+05	1.5211647E+05
5.0466555E+04	6.4156887E+10	2.1804769E+04	3.5008782E+02	1.5504324E+05	1.5504324E+05
5.0581713E+04	6.4178665E+10	2.1814626E+04	3.5793869E+02	1.5515396E+05	1.5515396E+05
4.9911189E+04	6.2111336E+10	2.1860281E+04	3.6580240E+02	1.5558895E+05	1.5558895E+05
4.8331464E+04	6.3572293E+10	2.2347513E+04	3.7352467E+02	1.5966836E+05	1.5966836E+05
4.8353275E+04	6.4442936E+10	2.2622028E+04	3.8118959E+02	1.6192717E+05	1.6192717E+05
4.8646106E+04	6.4359591E+10	2.2435784E+04	3.8699160E+02	1.6032665E+05	1.6032665E+05
4.7011089E+04	6.4466569E+10	2.2775749E+04	3.9661081E+02	1.6321559E+05	1.6321559E+05
4.7051374E+04	6.4576961E+10	2.3070594E+04	4.0421261E+02	1.6577911E+05	1.6577911E+05
4.5617931E+04	6.4588768E+10	2.3044152E+04	4.1195349E+02	1.6561355E+05	1.6561355E+05
4.6103565E+04	6.4533765E+10	2.3156348E+04	4.1947813E+02	1.6556348E+05	1.6556348E+05
4.5816455E+04	6.4727400E+10	2.3414791E+04	4.2706375E+02	1.6684806E+05	1.6684806E+05
4.5996297E+04	6.4717721E+10	2.3451733E+04	4.3654303E+02	1.6913946E+05	1.6913946E+05
4.5311040E+04	6.4753135E+10	2.3678506E+04	4.4220132E+02	1.7092774E+05	1.7092774E+05
4.4030974E+04	6.4803890E+10	2.3893740E+04	4.4978030E+02	1.7272210E+05	1.7272210E+05
4.4513807E+04	6.4818062E+10	2.3940572E+04	4.5752829E+02	1.7236536E+05	1.7236536E+05
4.3975257E+04	6.4865755E+10	2.3947666E+04	4.6800050E+02	1.7330246E+05	1.7330246E+05
4.4648919E+04	6.4935132E+10	2.4171781E+04	4.7229509E+02	1.7531010E+05	1.7531010E+05
4.4115429E+04	6.4970164E+10	2.4215048E+04	4.7980360E+02	1.7570601E+05	1.7570601E+05
4.3098205E+04	6.5529143E+10	2.4444924E+04	4.8727391E+02	1.7742509E+05	1.7742509E+05
4.3193320E+04	6.5021735E+10	2.4496571E+04	4.9747452E+02	1.7802213E+05	1.7802213E+05
4.2798233E+04	6.5018733E+10	2.4507802E+04	5.0223856E+02	1.7809781E+05	1.7809781E+05
4.2527114E+04	6.5105351E+10	2.4782966E+04	5.0966485E+02	1.8048870E+05	1.8048870E+05
4.2927887E+04	6.5112061E+10	2.4784693E+04	5.1711709E+02	1.8052738E+05	1.8052738E+05
4.1861613E+04	6.5158236E+10	2.4909695E+04	5.2455387E+02	1.8163692E+05	1.8163692E+05
4.1873625E+04	6.5229782E+10	2.5115161E+04	5.3195161E+02	1.8345182E+05	1.8345182E+05
4.197613C+04	6.5208243E+10	2.5011141E+04	5.3940142E+02	1.8259585E+05	1.8259585E+05
4.1428305E+04	6.5270835E+10	2.5297217E+04	5.4678771E+02	1.8699209E+05	1.8699209E+05
4.1553763E+04	6.5254348E+10	2.5320570E+04	5.5418794E+02	1.8508328E+05	1.8508328E+05
4.0920655E+04	6.5279191E+10	2.5375363E+04	5.6195214E+02	1.8550692E+05	1.8560602E+05
4.0956392E+04	6.5353104E+10	2.5633405E+04	5.6899156E+02	1.8786608E+05	1.8786608E+05
4.0610766E+04	6.5335144E+10	2.5521026E+04	5.7634304E+02	1.8693426E+05	1.8693426E+05
4.0362853E+04	6.5006498E+10	2.5747003E+04	5.8369826E+02	1.8890239E+05	1.8890239E+05
4.0562110E+04	6.5113393E+10	2.5821800E+04	5.9105143E+02	1.8952354E+05	1.8952354E+05
3.9861656E+04	6.55635692E+10	2.5967806E+04	5.98381179E+02	1.9079145E+05	1.9079145E+05
3.9997421E+04	6.5501983E+10	2.6126194E+04	6.0569701E+02	1.9216991E+05	1.9216991E+05
3.9680666E+04	6.5530305E+10	2.6103773E+04	6.1301629E+02	1.9201654E+05	1.9201654E+05
3.9457733E+04	6.5552986E+10	2.6308612E+04	6.2030664E+02	1.9377829E+05	1.9377829E+05
3.9014056E+04	6.5516868E+10	2.6230653E+04	6.2763591E+02	1.9304576E+05	1.9304576E+05
3.8695930E+04	6.5578593E+10	2.6489805E+04	6.3491089E+02	1.9527217E+05	1.9527217E+05
3.8579595E+04	6.5628256E+10	2.6553749E+04	6.4218579E+02	1.9592838E+05	1.9592838E+05
3.8210083E+04	6.5719199E+10	2.6768740E+04	6.4941965E+02	1.9793262E+05	1.9793262E+05
3.8109627E+04	6.5672026E+10	2.6862126E+04	6.5665120E+02	1.9863244E+05	1.9863244E+05
3.7523874E+04	6.5771707E+10	2.7008194E+04	6.6385559E+02	2.0000209E+05	2.0000209E+05
3.7602140E+04	6.5755132E+10	2.7074826E+04	6.7105946E+02	2.0045789E+05	2.0045789E+05
3.6981140E+04	6.5783460E+10	2.7230035E+04	6.7823332E+02	2.0176998E+05	2.0176998E+05
3.7180971E+04	6.5787036E+10	2.7244616E+04	6.8593212E+02	2.0189702E+05	2.0189702E+05
3.6557707E+04	6.5836225E+10	2.7429381E+04	6.9259292E+02	2.0352341E+05	2.0352341E+05
3.6906060E+04	6.5872888E+10	2.7459556E+04	6.9976241E+02	2.0389166E+05	2.0389166E+05
3.6368277E+04	6.5919721E+10	2.7611735E+04	7.0690859E+02	2.0525859E+05	2.0525859E+05
3.65882275E+04	6.5933070E+10	2.7626351E+04	7.1406598E+02	2.0542831E+05	2.0542831E+05
3.6123990E+04	6.5959950E+10	2.7761017E+04	7.2120532E+02	2.0557759E+05	2.0557759E+05
3.6115551E+04	6.5967132E+10	2.7761170E+04	7.2836195E+02	2.0559707E+05	2.0559707E+05
3.5802443E+04	6.5995075E+10	2.7909742E+04	7.3549285E+02	2.0787474E+05	2.0787474E+05
3.6013123E+04	6.6006732E+10	2.7923670E+04	7.4263603E+02	2.0802752E+05	2.0802752E+05
3.5609793E+04	6.6045790E+10	2.8069695E+04	7.4974994E+02	2.0953533E+05	2.0953533E+05
3.5653239E+04	6.6046143E+10	2.8063880E+04	7.5686681E+02	2.0927784E+05	2.0927784E+05
3.5453206E+04	6.6093776E+10	2.8168492E+04	7.6491139E+02	2.1028282E+05	2.1028282E+05
3.55565310E+04	6.6087005E+10	2.8229404E+04	7.7113453E+02	2.1073302E+05	2.1073302E+05
3.54481835E+04	6.6109558E+10	2.8312804E+04	7.7825059E+02	2.1147091E+05	2.1147091E+05
3.5310565E+04	6.6143				

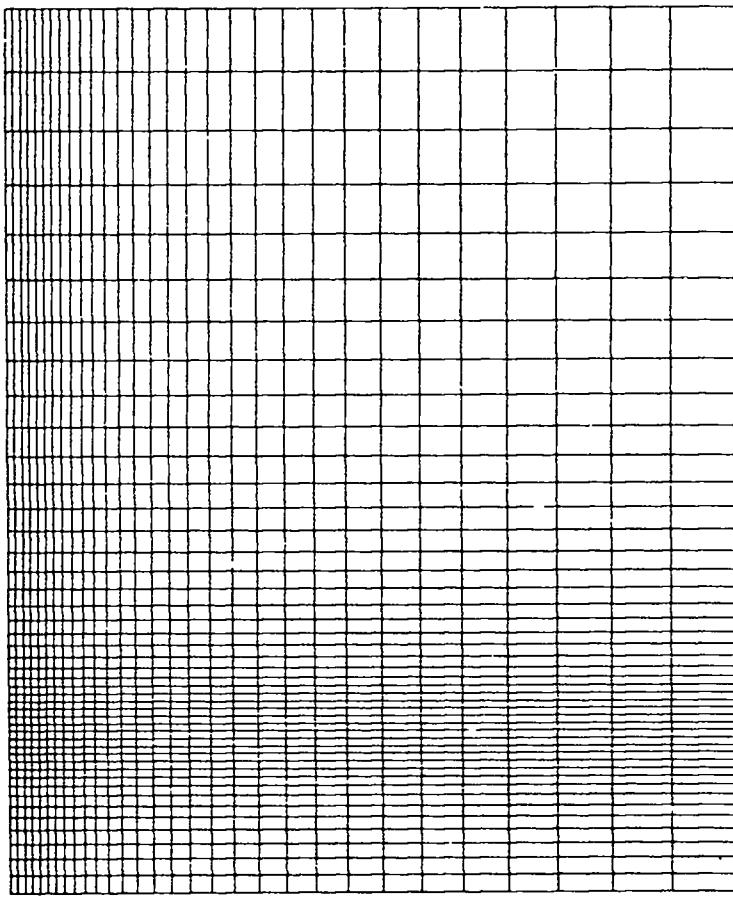
3.5115795E+04	4.0395760E+09	1.3544449E-03	7.9736131E+02	2.1552814E+06	2.1552813E+06
3.4998955E+04	3.9627654E+09	1.3797260E-03	7.9920512E+02	2.1662647E+06	2.1662647E+06
3.4980834E+04	3.9315579E+09	1.3976571E-03	8.0105426E+02	2.1676758E+06	2.1676757E+06
3.4927112E+04	3.8867795E+09	1.4152509E-03	8.0290913E+02	2.1710943E+06	2.1710943E+06
3.4866744E+04	3.8463971E+09	1.4336627E-03	8.0476857E+02	2.1778156E+06	2.1778156E+06
3.4872906E+04	3.8071720E+09	1.4498187E-03	8.0653544E+02	2.1810343E+06	2.1810342E+06
3.4862563E+04	3.7716673E+09	1.4672379E-03	8.0850010E+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	3.7058059E+09	1.5027809E-03	8.1226638E+02	2.2034832E+06	2.2034833E+06
3.4666535E+04	3.6728596E+09	1.5182729E-03	8.1415709E+02	2.2073735E+06	2.2073736E+06
3.4676816E+04	3.6431271E+09	1.5351190E-03	8.1605383E+02	2.2146775E+06	2.2146775E+06
3.4695515E+04	3.6143614E+09	1.5508051E-03	8.1795796E+02	2.2209886E+06	2.2209887E+06
3.4627353E+04	3.5875752E+09	1.5664986E-03	8.1986934E+02	2.2271302E+06	2.2271301E+06
3.4562070E+04	3.5621810E+09	1.5839974E-03	8.2178567E+02	2.2380425E+06	2.2380425E+06
3.4547129E+04	3.5377923E+09	1.5961116E-03	8.2371334E+02	2.2392271E+06	2.2392271E+06
3.4552210E+04	3.5119800E+09	1.6104773E-03	8.2569497E+02	2.2454942E+06	2.2454942E+06
3.4441207E+04	3.4931143E+09	1.6242718E-03	8.2759452E+02	2.2514300E+06	2.2514300E+06
3.4407332E+04	3.4716822E+09	1.6376664E-03	8.2954901E+02	2.2565508E+06	2.2565508E+06
3.4273689E+04	3.4512874E+09	1.6514490E-03	8.3151221E+02	2.2628117E+06	2.2628117E+06
3.4397673E+04	3.4319637E+09	1.6660010E-03	8.3348305E+02	2.2706127E+06	2.2706127E+06
3.4298110E+04	3.4112192E+09	1.6797741E-03	8.3546239E+02	2.2768664E+06	2.2768663E+06
3.4227273E+04	3.3957254E+09	1.6962041E-03	8.3741659E+02	2.2808185E+06	2.2808185E+06
3.4203121E+04	3.3758874E+09	1.7087326E-03	8.3941114E+02	2.2931063E+06	2.2931063E+06
3.4156343E+04	3.3581616E+09	1.7198941E-03	8.4146464E+02	2.2958637E+06	2.2958637E+06
3.4057786E+04	3.3430190E+09	1.7320466E-03	8.4358996E+02	2.3049913E+06	2.3049912E+06
3.4071515E+04	3.3271710E+09	1.7431560E-03	8.4548100E+02	2.3047879E+06	2.3047879E+06
3.3930328E+04	3.3103217E+09	1.7562606E-03	8.4751913E+02	2.3124800E+06	2.3124800E+06
3.3950430E+04	3.2954770E+09	1.7678914E-03	8.4956326E+02	2.3177797E+06	2.3177797E+06
3.3866926E+04	3.2801693E+09	1.7781635E-03	8.5161876E+02	2.3209315E+06	2.3209315E+06
3.3789672E+04	3.2680505E+09	1.7916209E-03	8.5368172E+02	2.3302033E+06	2.3302033E+06
3.3775590E+04	3.2527515E+09	1.7997060E-03	8.5575821E+02	2.3302694E+06	2.3302694E+06
3.3698677E+04	3.2404706E+09	1.8126121E-03	8.5780294E+02	2.3385056E+06	2.3385056E+06
3.3683397E+04	3.2296884E+09	1.8283332E-03	8.5993124E+02	2.3512732E+06	2.3512731E+06
3.3659585E+04	3.2163368E+09	1.8332907E-03	8.6203861E+02	2.3683203E+06	2.3683203E+06
3.3621172E+04	3.2124636E+09	1.8490261E-03	8.6414611E+02	2.3857470E+06	2.3857470E+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
3.3682092E+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	2.4168093E+06	2.4168092E+06
3.3717361E+04	3.1578656E+09	1.9471212E-03	8.7913071E+02	2.4439052E+06	2.4439052E+06
3.3686282E+04	3.1488634E+09	1.9504839E-03	8.8129657E+02	2.4482309E+06	2.4482309E+06
3.3666510E+04	3.1459366E+09	1.9666511E-03	8.8346509E+02	2.4659499E+06	2.4659499E+06
3.3907936E+04	3.1414258E+09	1.9923665E-03	8.8563870E+02	2.4826188E+06	2.4826187E+06
3.4078515E+04	3.1333618E+09	1.9918356E-03	8.8781813E+02	2.4883423E+06	2.4883423E+06
3.4058595E+04	3.1355896E+09	2.0170949E-03	8.8992132E+02	2.5216114E+06	2.5216113E+06
3.4248190E+04	3.1272549E+09	2.0241597E-03	8.9217833E+02	2.5840010E+06	2.5840010E+06
3.4228645E+04	3.1288303E+09	2.0477886E-03	8.9458854E+02	2.5945981E+06	2.5945981E+06
3.4365829E+04	3.1217103E+09	2.0566002E-03	8.9654976E+02	2.5600095E+06	2.5600095E+06
3.4442966E+04	3.1202787E+09	2.0769691E-03	8.9870276E+02	2.5814284E+06	2.5814284E+06
3.4593747E+04	3.1202309E+09	2.0949301E-03	9.0092891E+02	2.6066277E+06	2.6066277E+06
3.4700485E+04	3.1154214E+09	2.1060512E-03	9.0312491E+02	2.6166020E+06	2.6166020E+06
3.4808810E+04	3.1192590E+09	2.1317003E-03	9.0531319E+02	2.6515840E+06	2.6515840E+06
3.4918817E+04	3.1137531E+09	2.1451568E-03	9.0750996E+02	2.6593540E+06	2.6593540E+06
3.4957669E+04	3.1151266E+09	2.1624766E-03	9.0970393E+02	2.6863877E+06	2.6863876E+06
3.5176885E+04	3.1124720E+09	2.1761640E-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
3.5192986E+04	3.1119803E+09	2.2117615E-03	9.1629695E+02	2.7493736E+06	2.7493736E+06
3.5168387E+04	3.1129494E+09	2.2308590E-03	9.1696313E+02	2.7695470E+06	2.7695470E+06
3.5573106E+04	3.1141848E+09	2.2507700E-03	9.2069225E+02	2.7951290E+06	2.7951290E+06
3.5649490E+04	3.1095271E+09	2.2594636E-03	9.2209726E+02	2.8044427E+06	2.8044427E+06
3.5794816E+04	3.1132518E+09	2.2842903E-03	9.2509689E+02	2.8368151E+06	2.8368151E+06
3.5595717E+04	3.1114875E+09	2.2960763E-03	9.2730109E+02	2.8516419E+06	2.8516419E+06
3.5974976E+04	3.1154394E+09	2.3194394E-03	9.2950090E+02	2.8664831E+06	2.8664831E+06
3.6259565E+04	3.1126982E+09	2.3310621E-03	9.3170586E+02	2.8952379E+06	2.8952379E+06
3.6295193E+04	3.1196232E+09	2.3582757E-03	9.3390203E+02	2.9362633E+06	2.9337464E+06
3.6532198E+04	3.1178701E+09	2.3716635E-03	9.3610249E+02	2.9551055E+06	2.9495633E+06
3.6619990E+04	3.1221194E+09	2.3930645E-03	9.3829951E+02	2.9818666E+06	2.9793074E+06
3.6891669E+04	3.1263956E+09	2.4161098E-03	9.4091913E+02	3.0145619E+06	3.0119439E+06
3.6923684E+04	3.1279237E+09	2.4336871E-03	9.4268489E+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
3.7205326E+04	3.1316120E+09	2.4694429E-03	9.4706768E+02	3.0866072E+06	3.0833358E+06
3.7392724E+04	3.1372598E+09	2.4946243E-03	9.4925406E+02	3.1293727E+06	3.1201751E+06
3.7070561E+04	3.1326599E+09	2.4990925E-03	9.5145219E+02	3.1240914E+06	3.1213258E+06
3.5159110E+04	3.0880550E+09	2.4293866E-03	9.5373301E+02	2.9905566E+06	2.9979649E+06
3.0214252E+04	2.9794008E+09	2.1758195E-03	9.5629219E+02	2.5570794E+06	2.5570794E+06
2.2358707E+04	2.6747876E+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	1.3914276E+06	1.3914276E+06
5.561253E+03	2.1364384E+09	1.2188572E-03	9.6788737E+02	1.0514459E+06	1.0514459E+06
1.3584421E+03	2.0100951E+09	1.0848125E-03	9.7307547E+02	8.8530822E+05	8.8530822E+05
2.0436807E+02	1.9688566E+09	1.3391039E-03	9.7951043E+02	8.2854024E+05	8.2854024E+05
2.5655060E+01	1.9606290E+09	1.0296106E-03	9.8400125E+02	8.16e3626E+05	8.1693625E+05
2.9506314E+00	1.9595630E+09	1.0271566E-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	1.0035000E+03	8.1494950E+05	8.1494950E+05

ALL FIREBALL INPUT DATA READ



PARTICLES PXR= 1.9279E+04 PYB=-1.38538E-03 PYT= 2.3579E+04
YACUH-LISS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN IRECOR

T = 8.30000E-03 CYCLE = 9

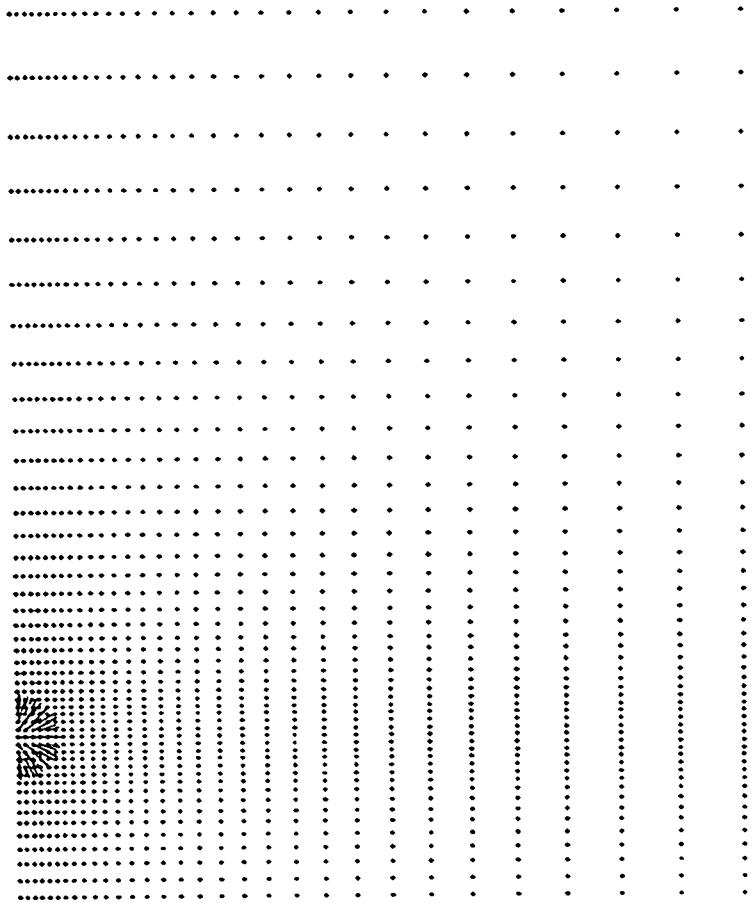


ALL ZONES

DZMIN= 2.00000E+02 DZMAX= 1.69944E+03 DZMIN= 2.00000E-02
DZMAX= 1.69944E-03 XR= 1.9279E+04 YR=-1.38538E-03 YT= 2.3579E+04

YACUI-LTSS BALLOON - 30 X 45 - CONC. CHGS.. PARTICLE TURB. DIFF.. FIX IN IRBCOR

T= 8.30000E-03 CYCLE= 0

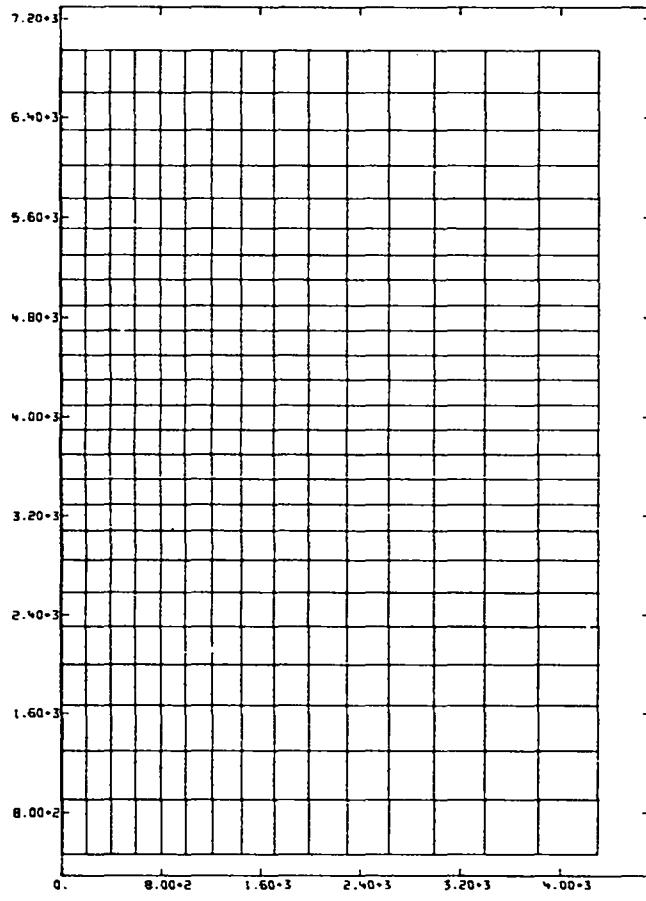


FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY

VMAX= 4.86465E+04

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

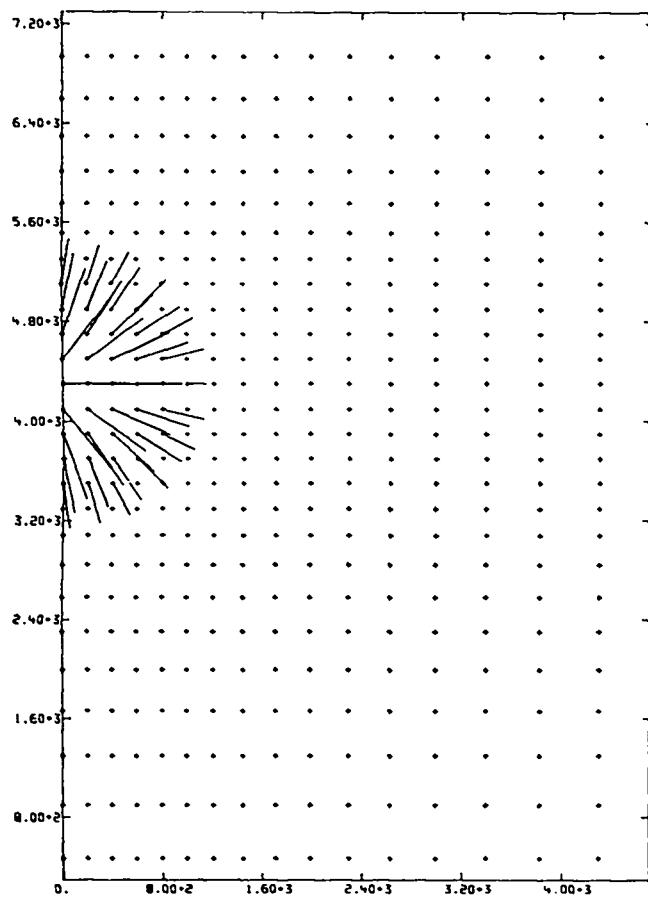
T= 8.30000E-03 CYCLE= 0



ZONES IN THE FIREBALL REGION

DZMIN= 2.0000E-02 DZMAX= 1.6994E-03 DZMIN= 2.0000E-02
 DZMAX= 1.6994E-03 XA= 1.9279E+04 YA=-1.30538E-03 TA= 2.3579E+04
 VACU-LYSS BALLOON - 30 x 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRICOR

T= 8.3000E-03 CYCLE= 0



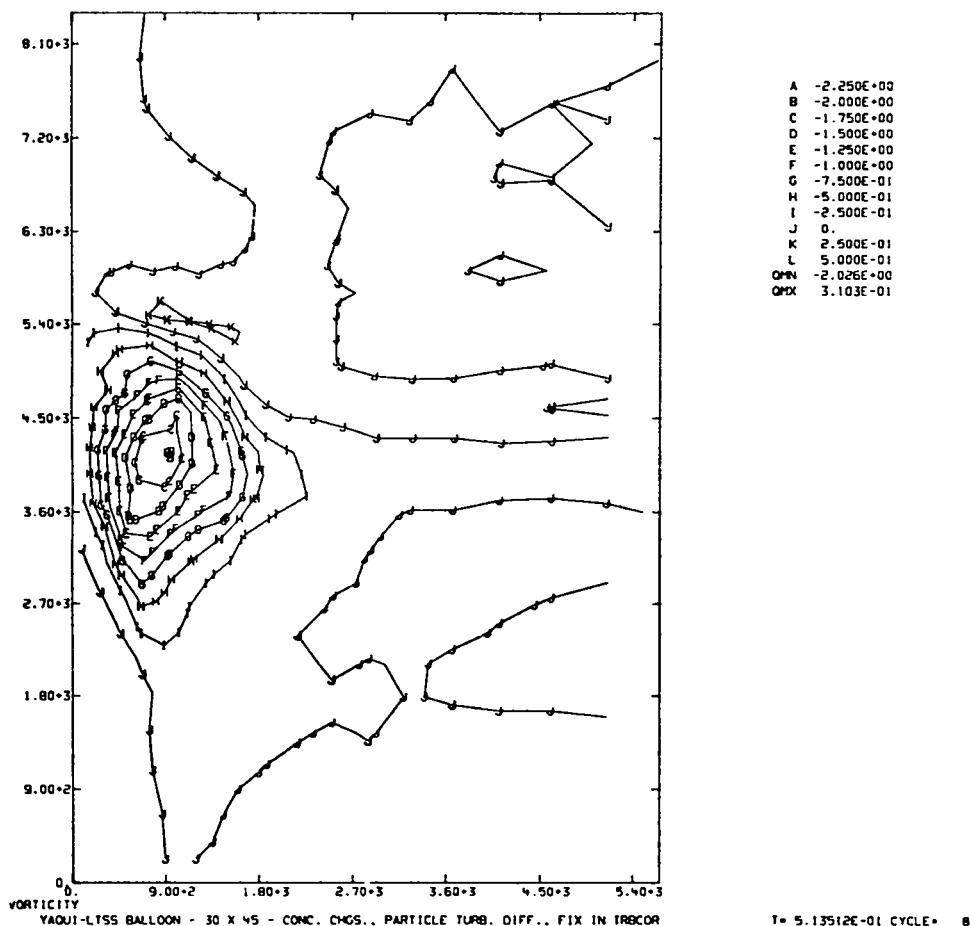
FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY

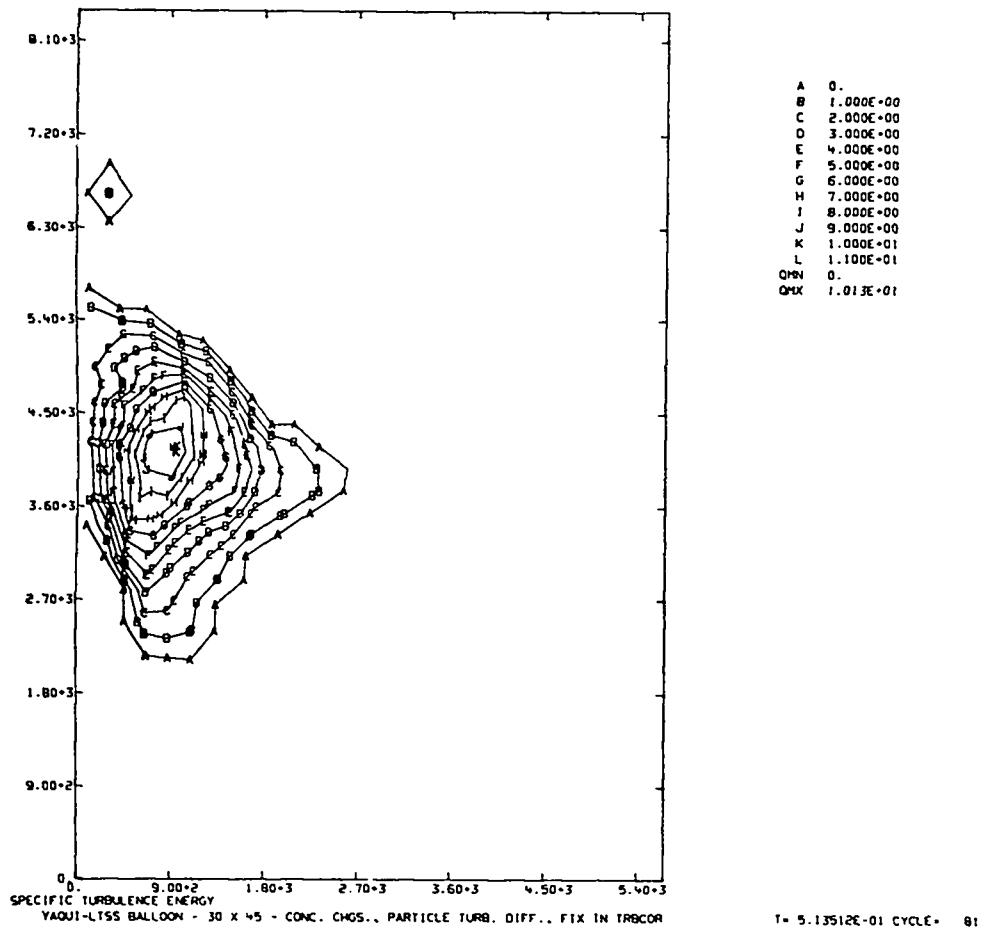
VMAX= 4.86465E+04

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

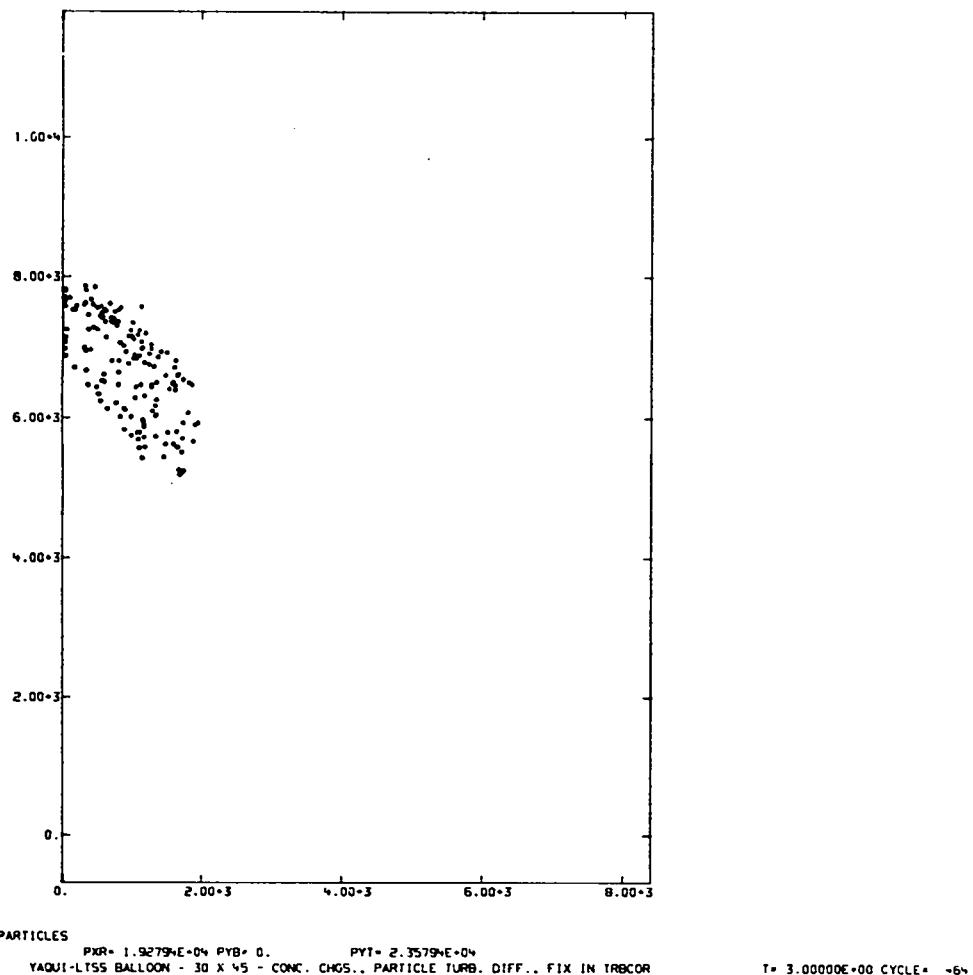
T= 8.3000E-03 CYCLE= 0

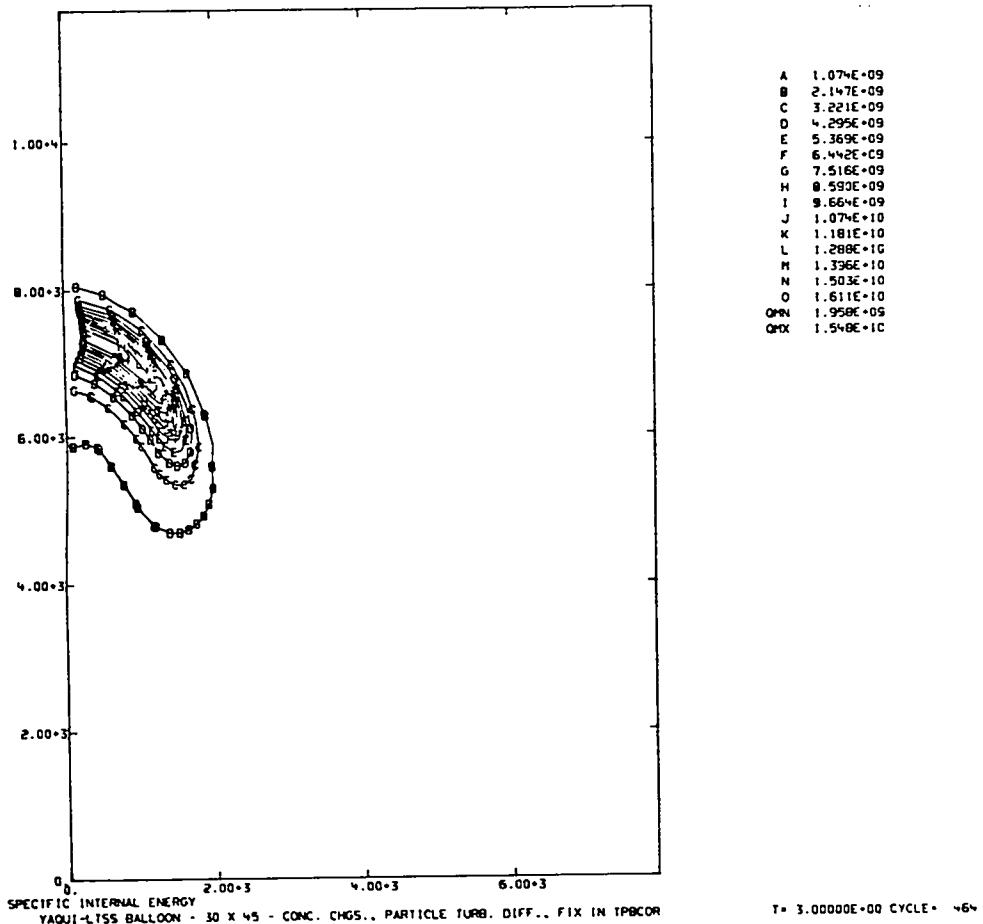
B. Turbulence Seeding Conditions

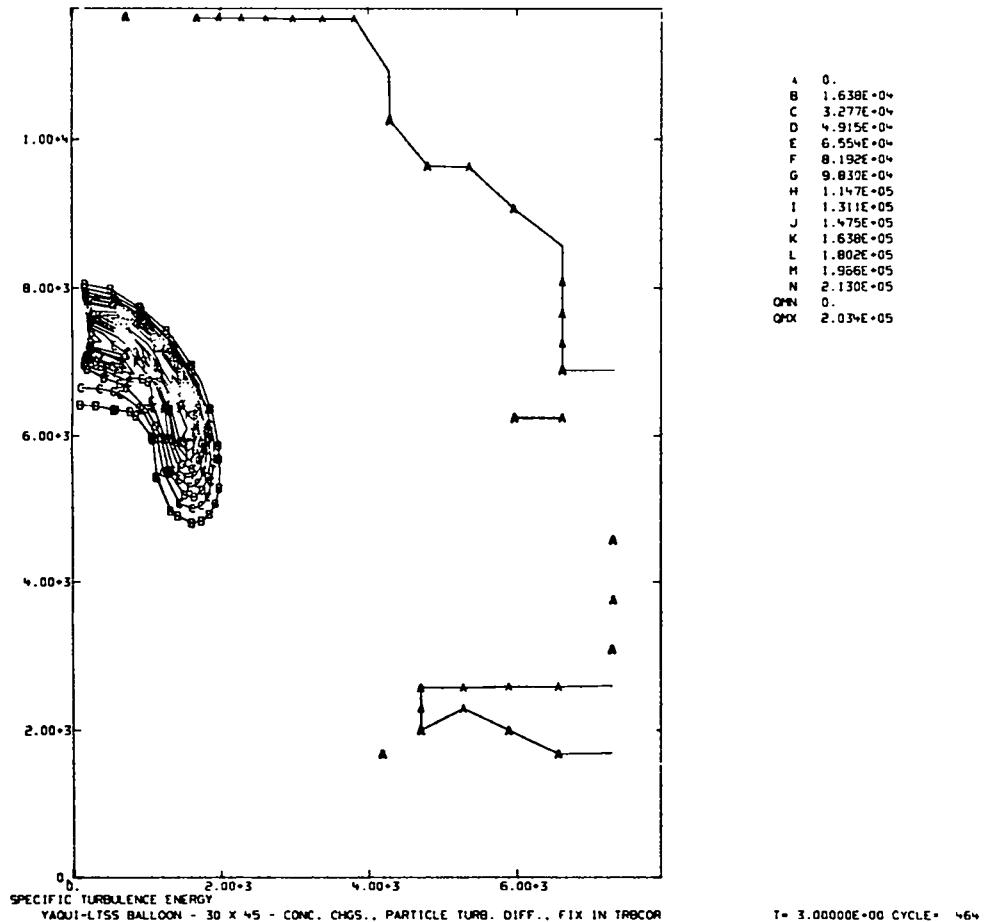




C. Turbulence Equilibrium Conditions







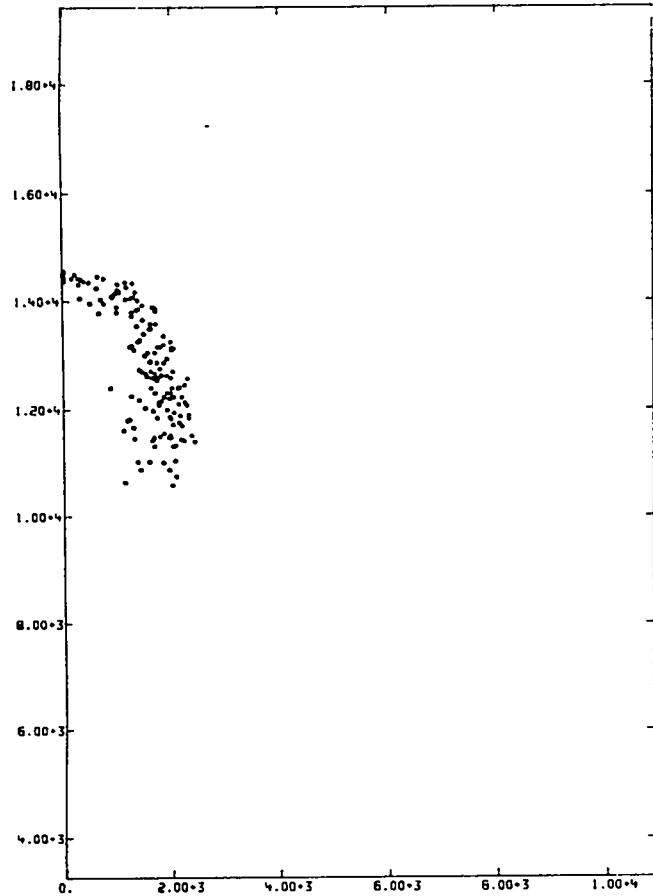
D. Moderately Late-Time Conditions (Eight Torus-Formation Times)

```

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

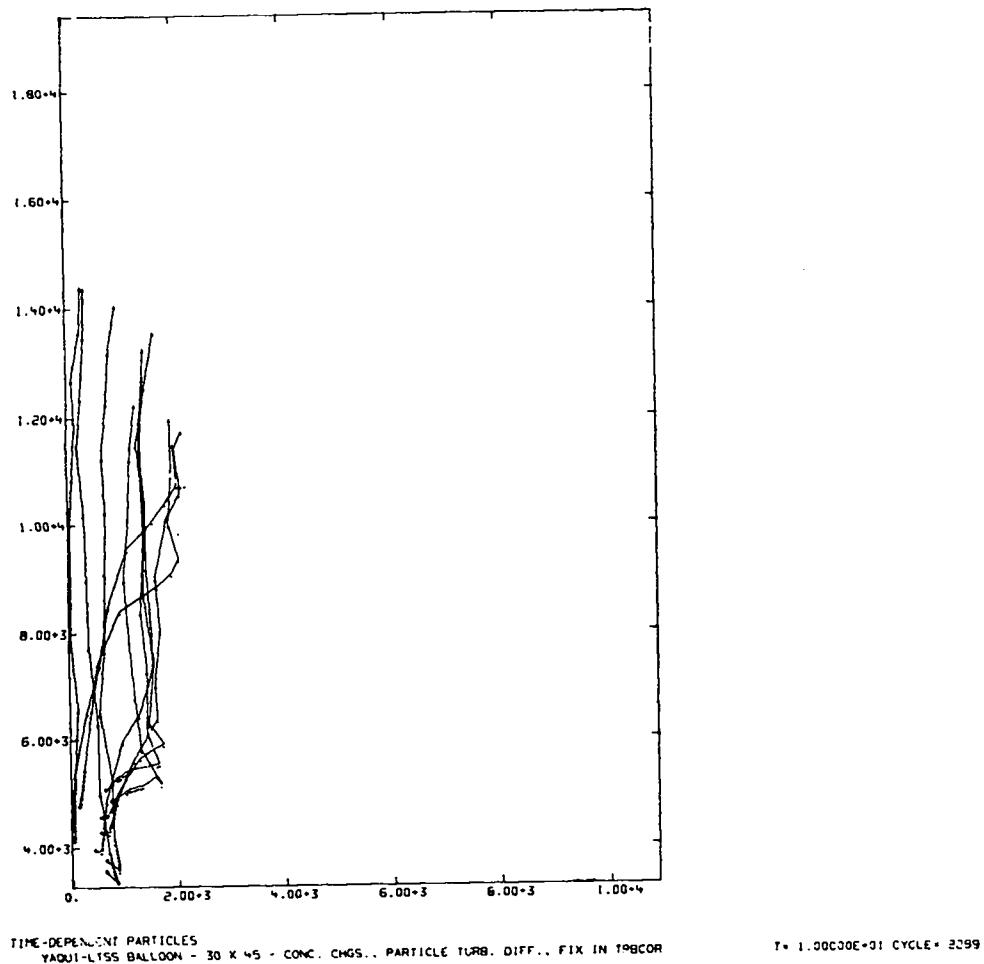
***** CYCLE 2289, T= 1.00000E+01, DT= 1.34E00E-03, CP= 2.97676E+00
GRINDS= 1.29761E-04, NMUTT= 0, CIRC= 1.09160E+07
DTIV= 4.25932E-03, IDTV= 7, JDTIV= 22
DTC= 1.10200E-01, IDTC= 1, JDTIC= 23
TMAX= 3.14920E+09, IIM= 2, JIM= 33, XTMAX= 0, YTMAX= 0,
TGMAX= 2.65171E+06, ITG= 1, JTG= 34
PRIT= 2.435E+03, PTOP= 1.455E+04, PBOT= 1.058E+04, POIAM= 4.871E+03, PAVHI= 1.257E+04
TOTAL INTERNAL ENERGY = 1.0505706E+19
TOTAL KINETIC ENERGY = 7.1391765E+12
TOTAL GRAV. POTENTIAL ENERGY = 6.8996748E+16
TOTAL RADIAL MOMENTUM = 4.6797558E+09
TOTAL AXIAL MOMENTUM = -2.6862275E+10
I7104271067616720223   17147033011630+61263
VMAX = 1.12913E+03 AT VERTEX 5 16

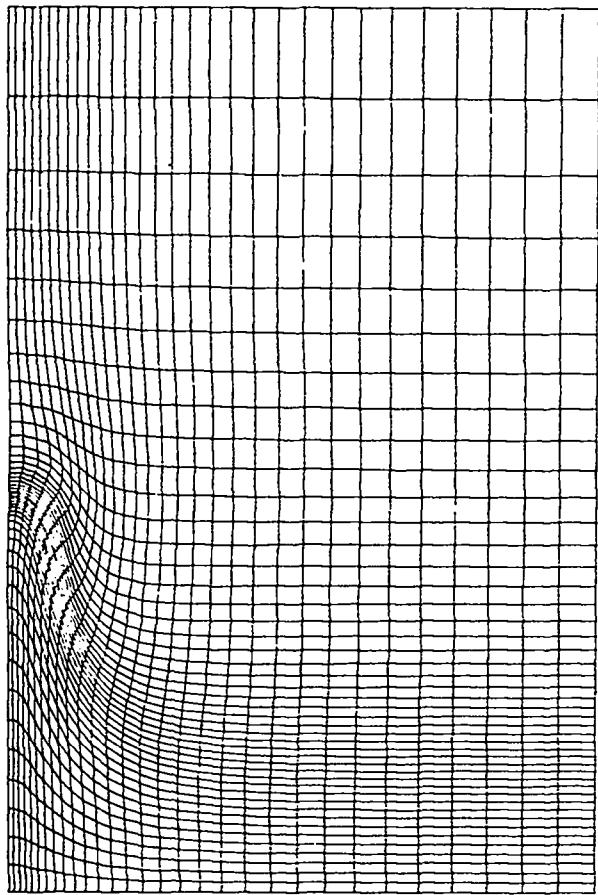
```



PARTICLES PXR= 1.92794E+04 PYB= 0. PYT= 2.92102E+04
 YAOUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TPBCOR

T= 1.00000E+01 CYCLE= 2289

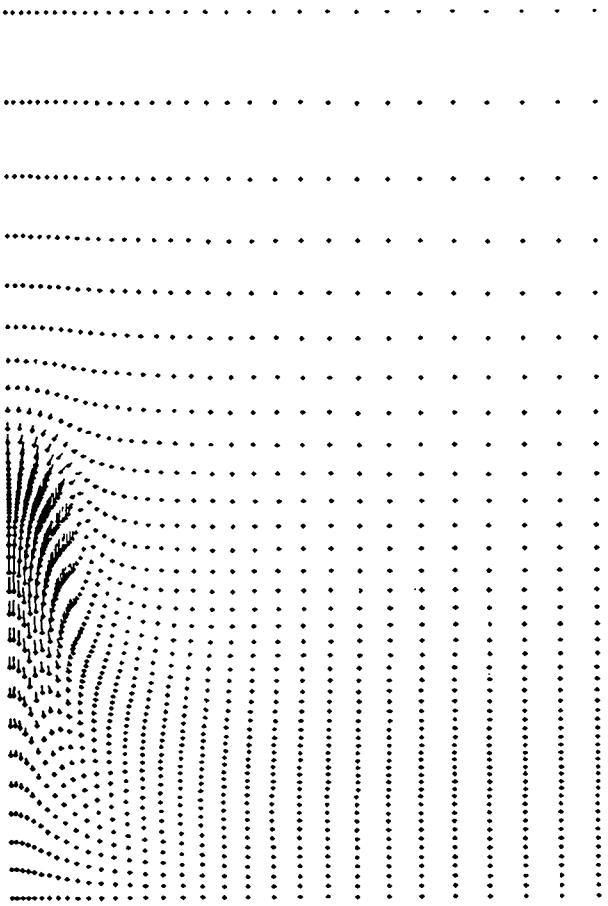




ALL ZONES

DRMIN= 1.59383E+02 DRMAX= 1.18653E+03 DZMIN= 8.03520E-01
DZMAX= 2.99070E+03 XR= 1.92794E+04 YR= 0.
YAOUJ-L155 BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN FRCOR

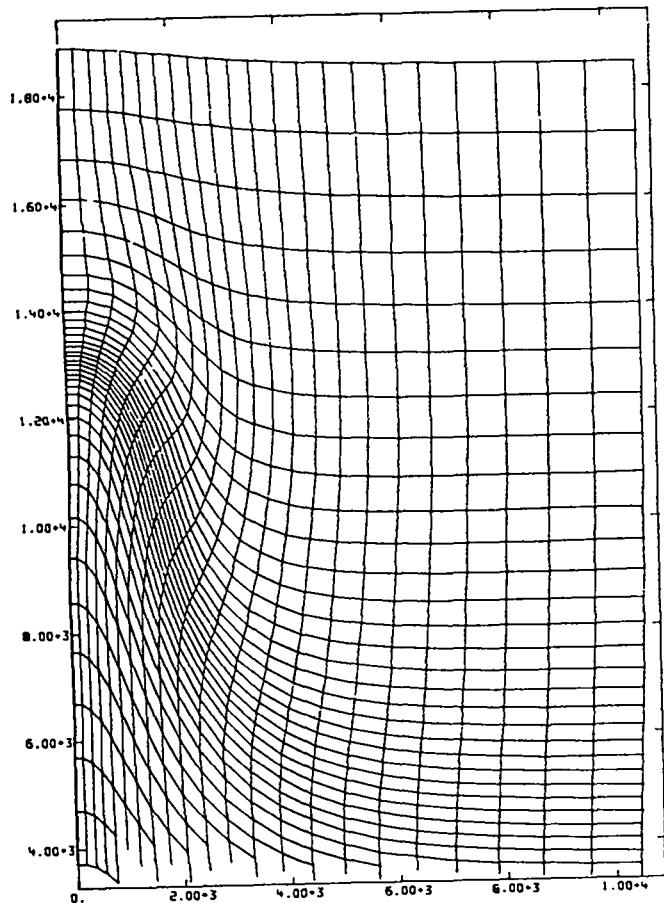
T= 1.00000E+01 CYCLE= 2299



FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY
VMAX= 1.12413E+03

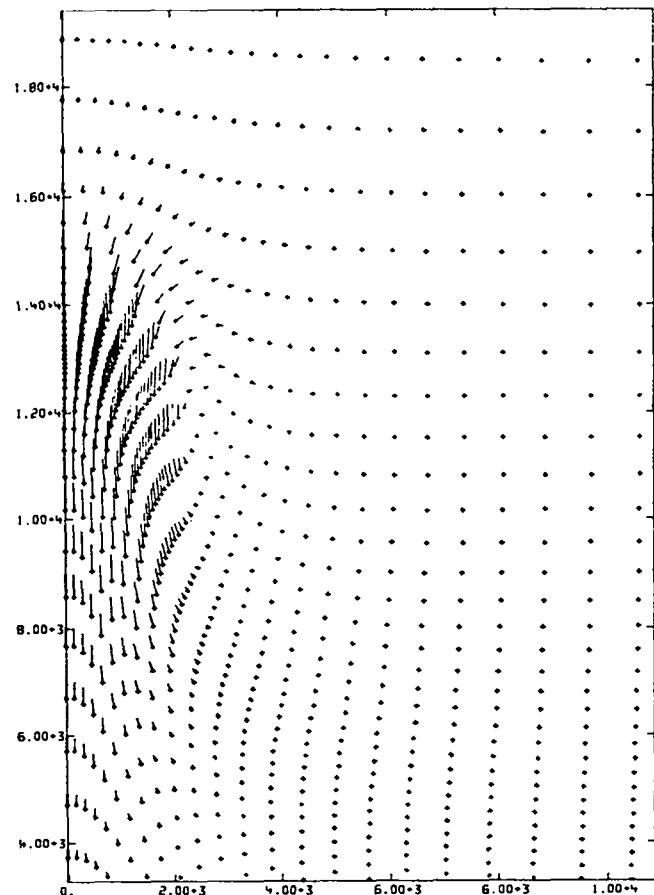
YAGU1-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

T= 1.0000E+01 CYCLE= 2299



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

$t = 1.00000E+01$ CYCLE= 2299

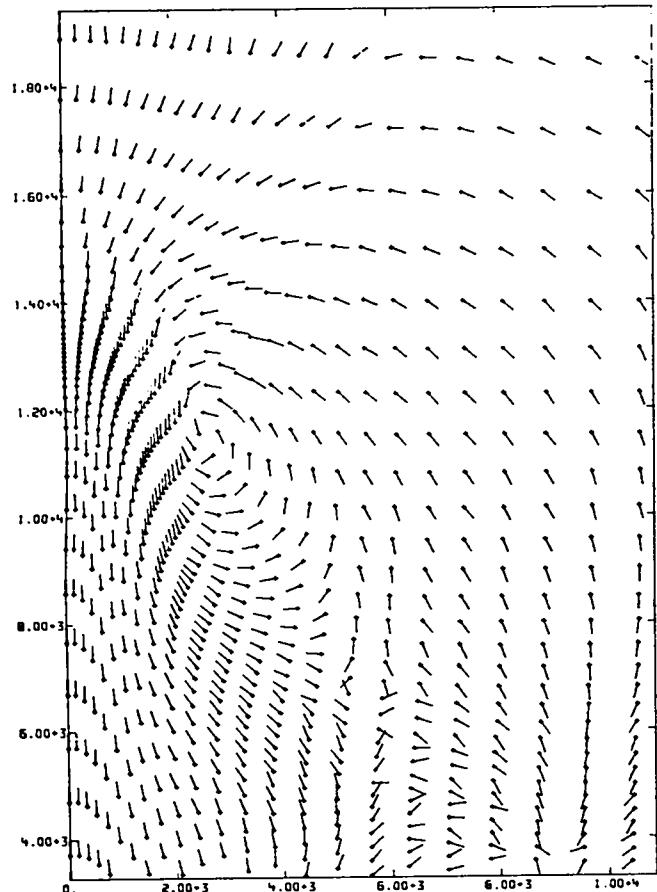


FLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY

VMAX= 1.12413E-03

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

T= 1.00000E-01 CYCLE= 2289

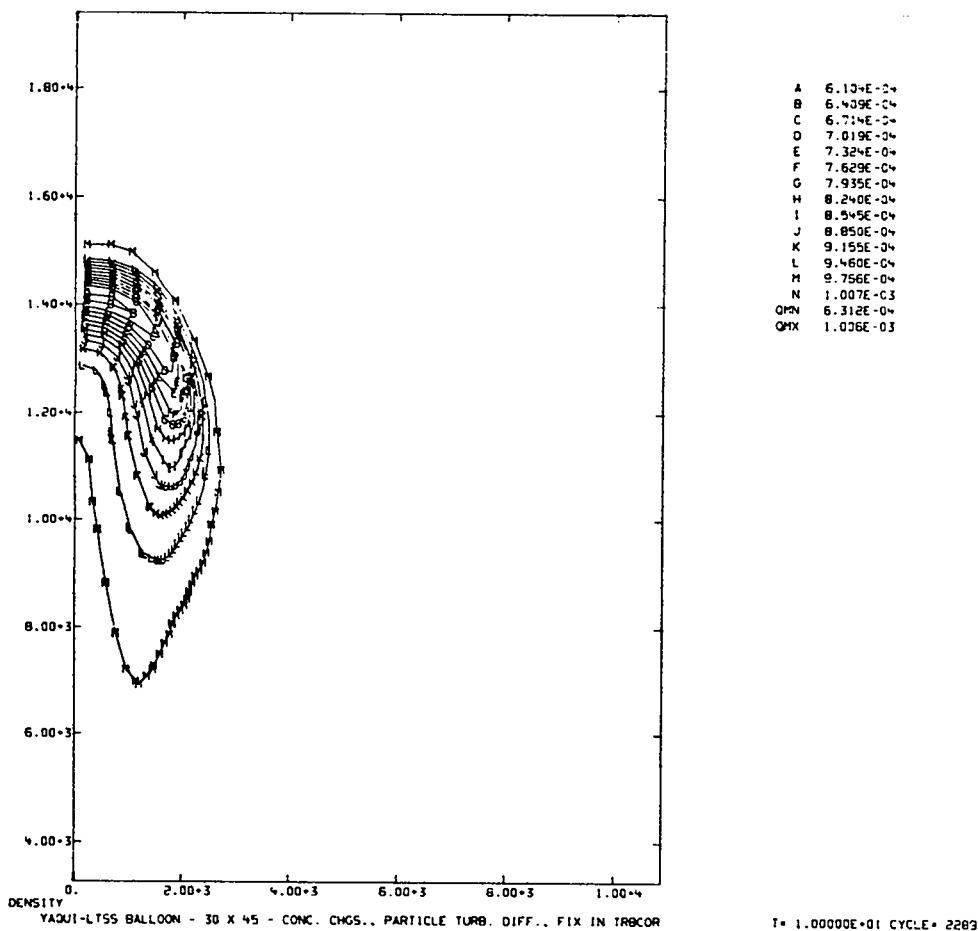


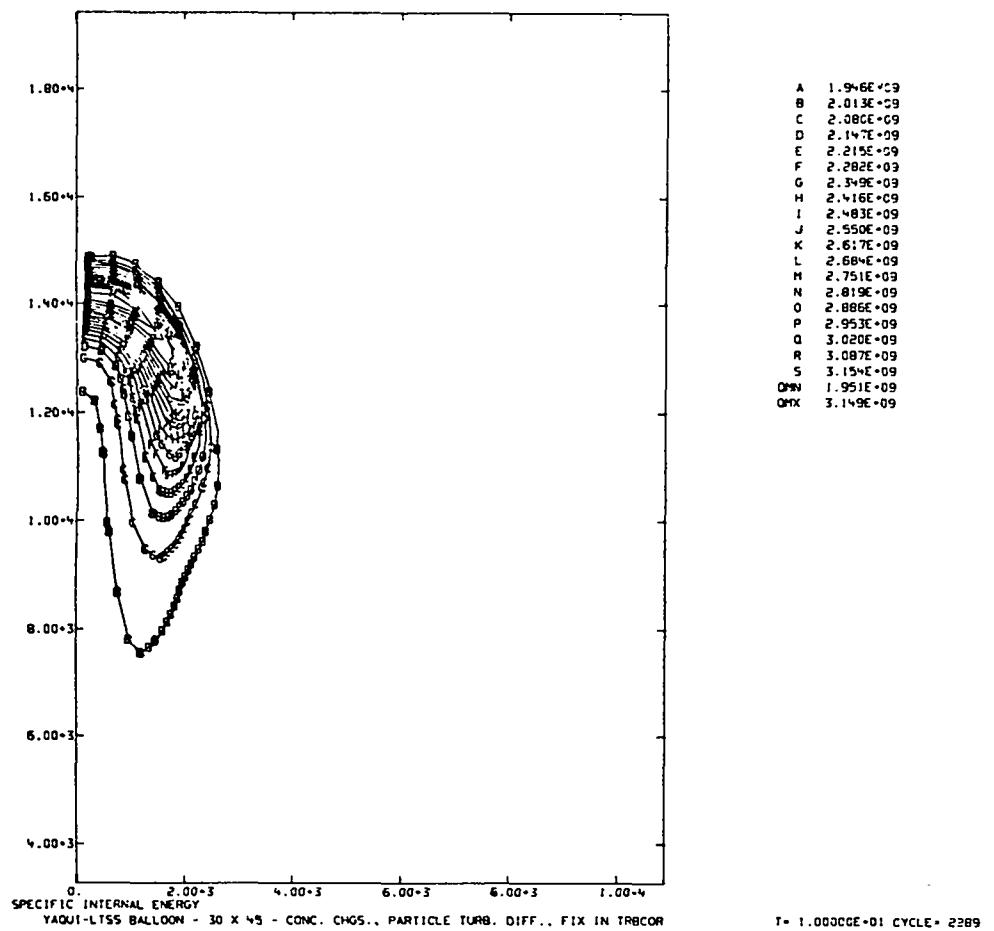
UNSCALED FLUID VELOCITY VECTORS

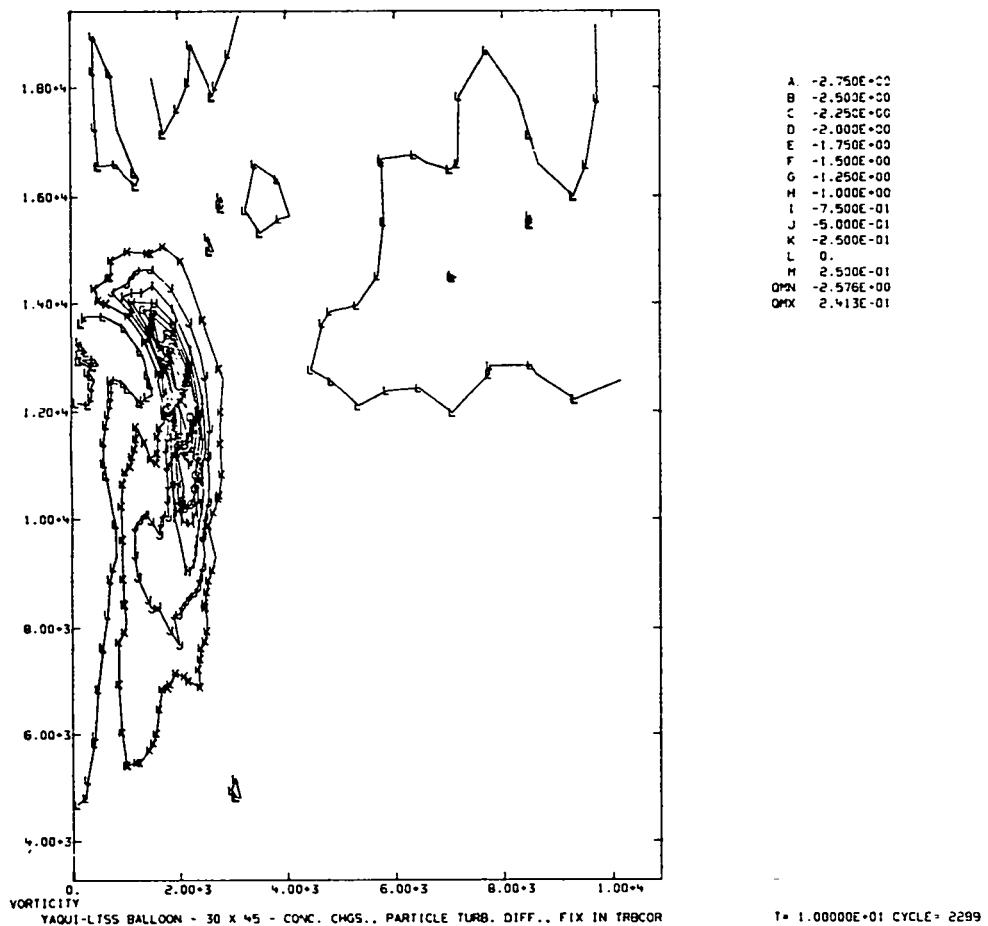
VMAX= 1.12e+03

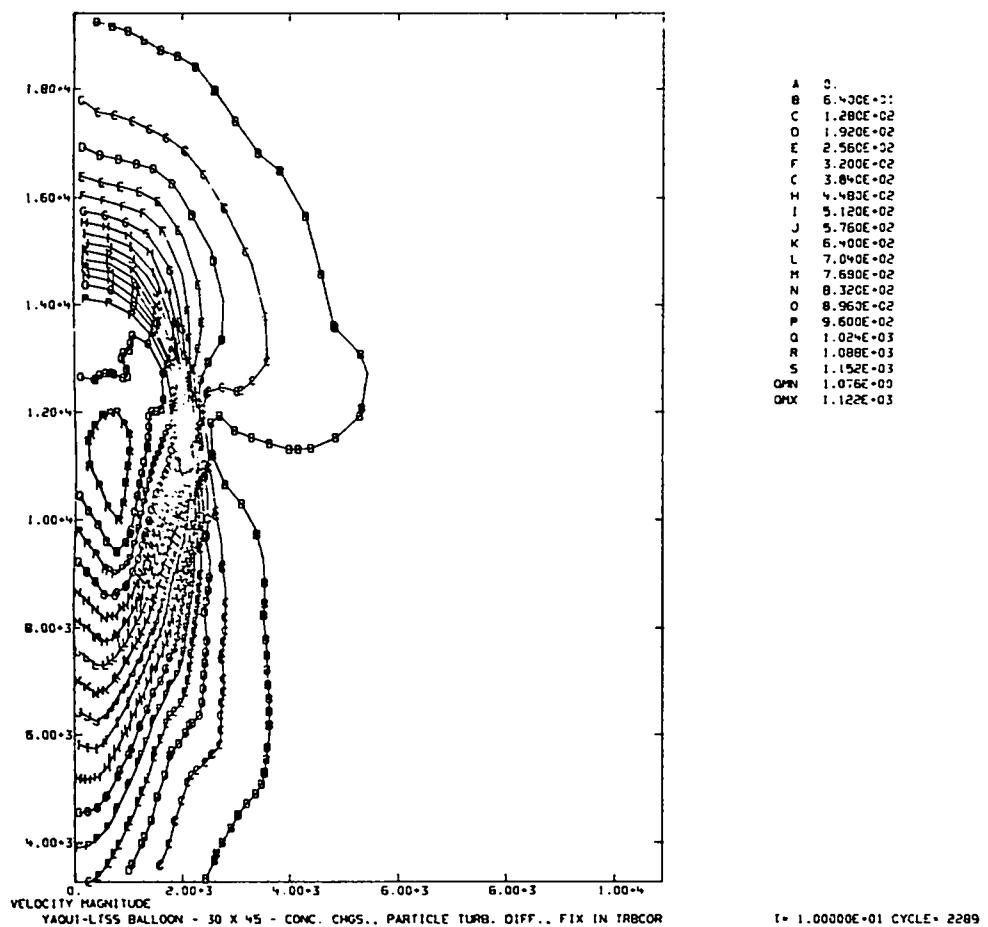
YACUI-LTSS BALLOON - 30 X 45 - CONC. CHGS., PARTICLE TURB. DIFF., FIX IN TRBCOR

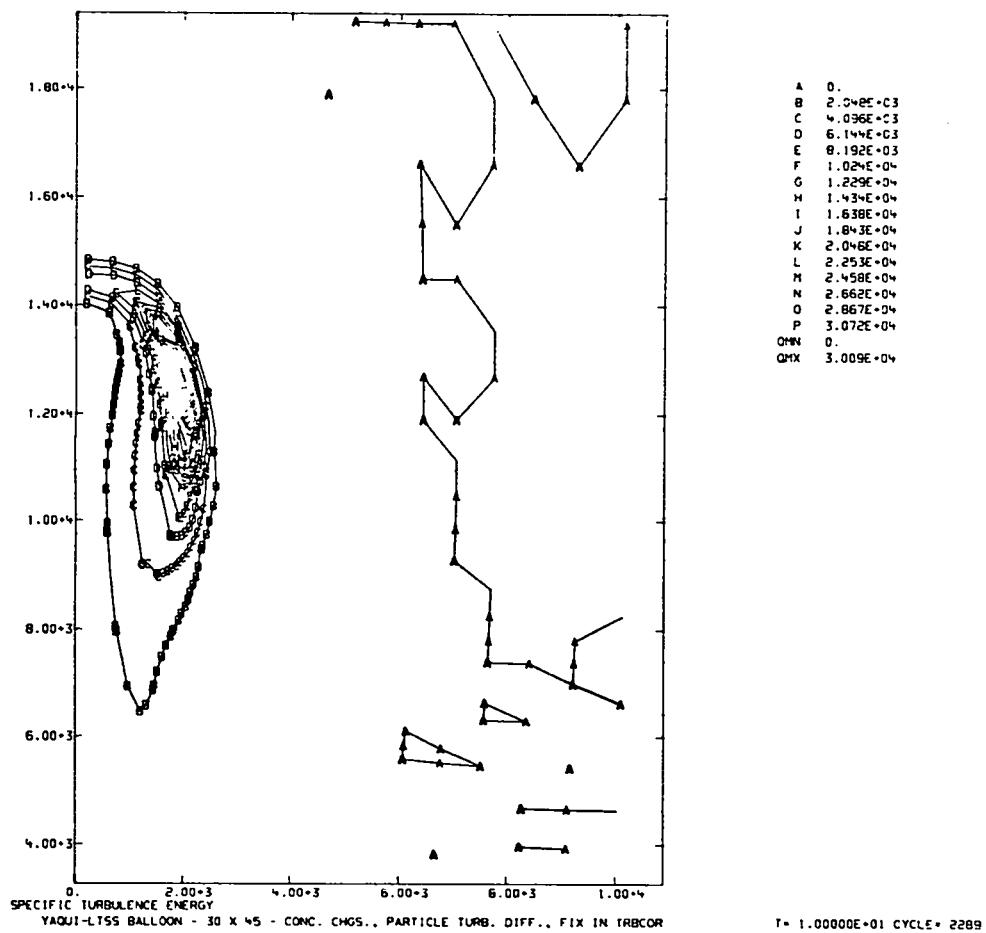
T= 1.00000E+01 CYCLE= 2283

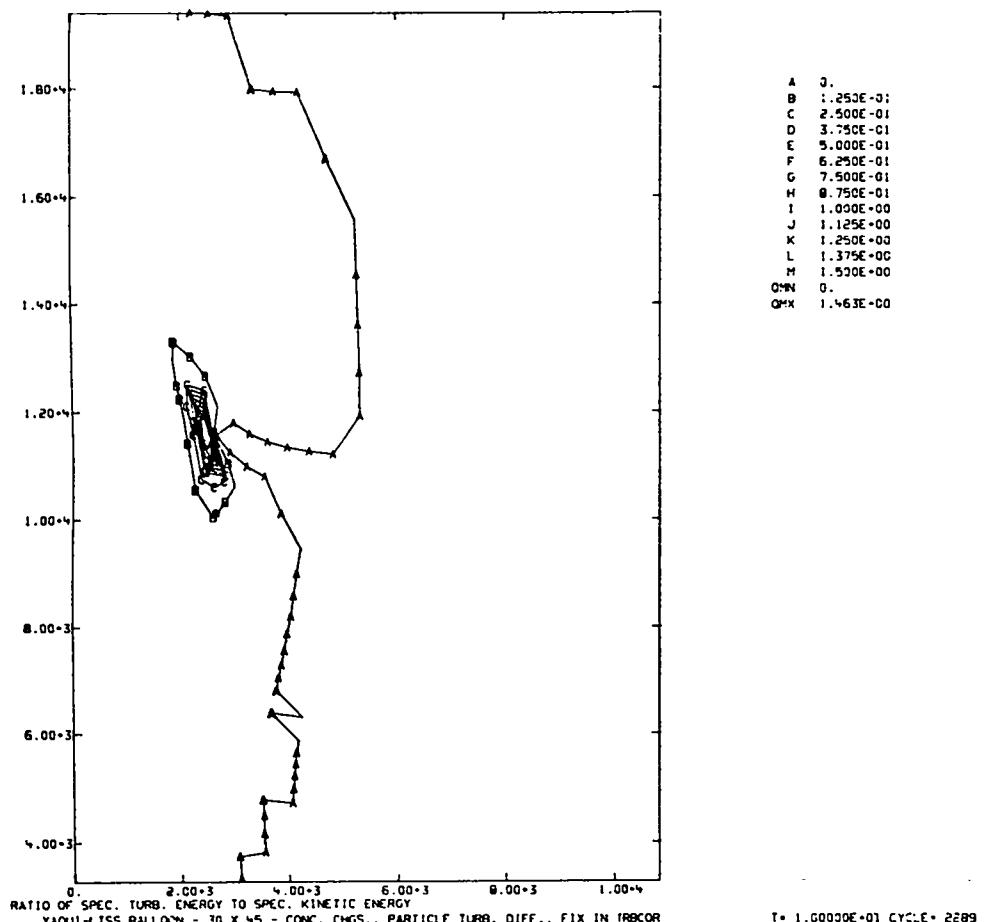


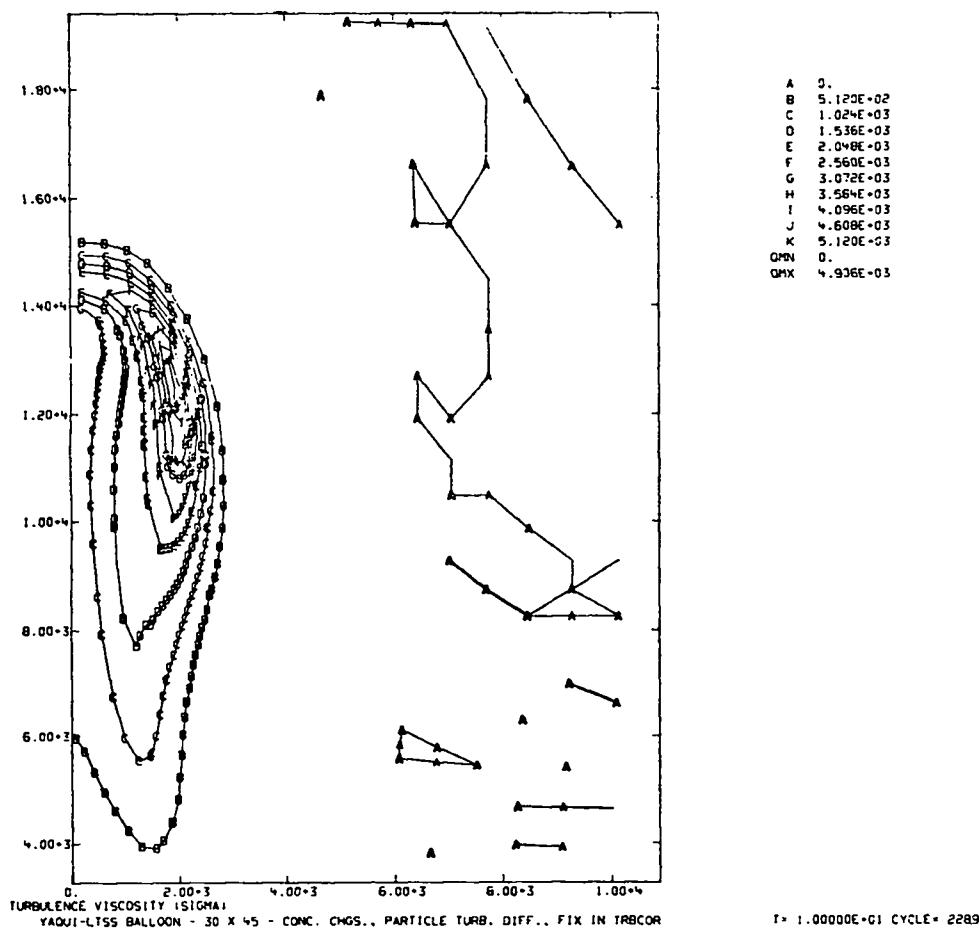


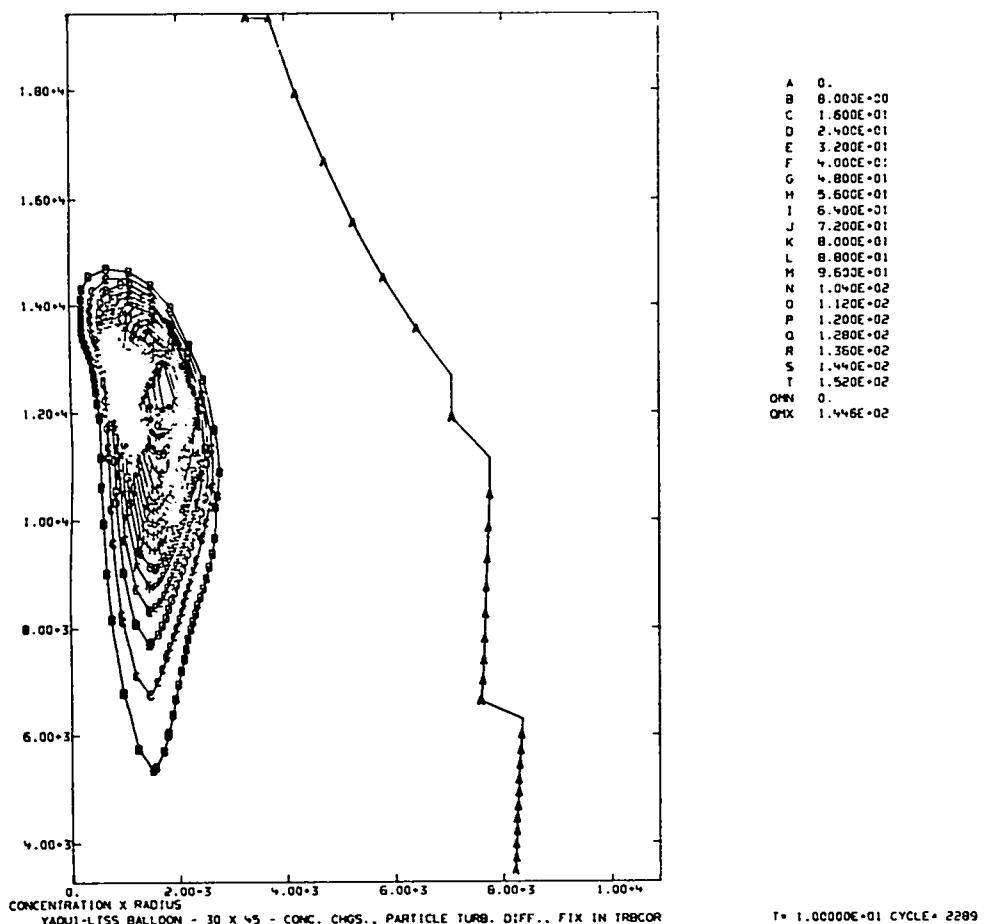




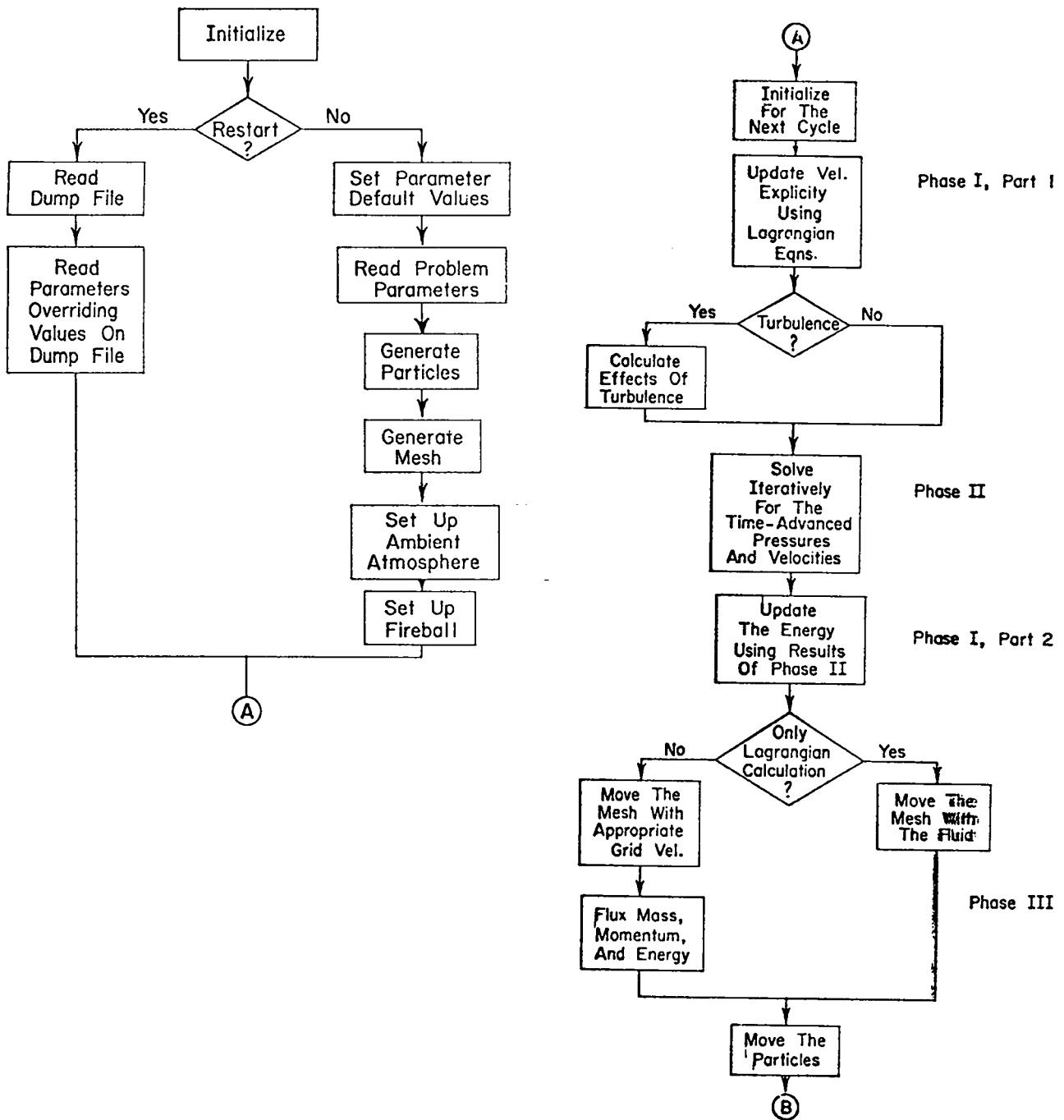


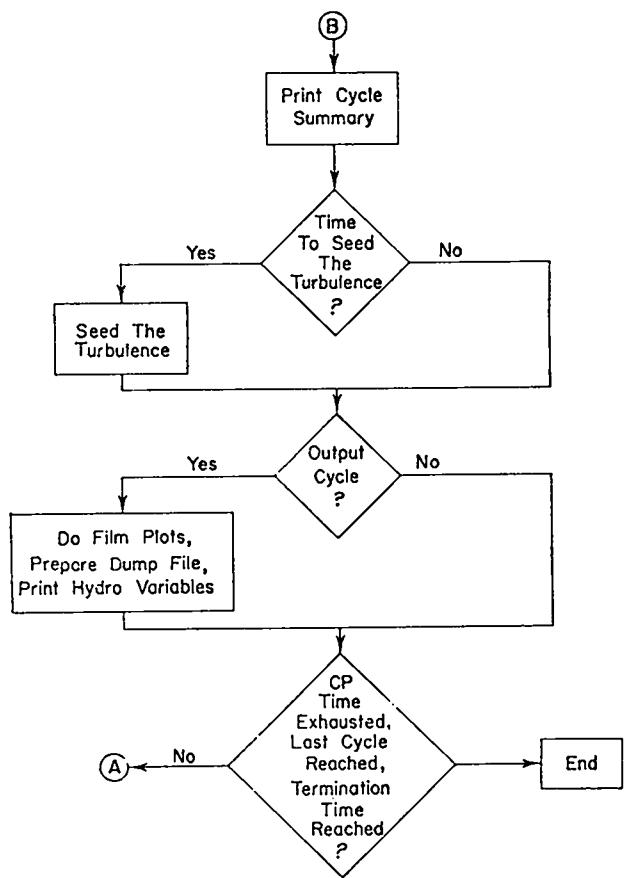






IV. FLOW DIAGRAM





A. LTSS-Dependent Code

```

1      PROGRAM MAIN(TAPE1,TAPE5=TAPE1,DTAPE1,TAPE6=DTAPE1,TAPE63=100B,
2      TAPE3)
3      C
4      C      LTSS MAIN ROUTINE FOR YAQUI
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      C      CALL YAQUI
9      C      END
-----
```

```

1      INTEGER AND
2      FUNCTION AND(I,J)
3      C
4      C      LTSS ROUTINE TO TAKE THE BOOLEAN INTERSECTION OF TWO VARIABLES
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      C      AND=I.INT.J
9      C      RETURN
10     C      END
-----
```

```

1      INTEGER ANDR
2      FUNCTION ANDR(I,J)
3      C
4      C      LTSS INTERSECTION ROUTINE RETURNED IN A FLOATING-POINT VARIABLE
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      C      ANDR=I.INT.J
9      C      RETURN
10     C      END
-----
```

```

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

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

```
-----  
1      INTEGER COMP
2      FUNCTION COMP(I)
3      C
4      C      LTSS ROUTINE TO TAKE THE COMPLEMENT OF A VARIABLE
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      COMP=.COMP.I
9      RETURN
10     END
```

```
-----  
1      INTEGER COMPR
2      FUNCTION COMPR(I)
3      C
4      C      LTSS COMPLEMENT ROUTINE RETURNED IN A FLOATING-POINT VARIABLE
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      COMPR=.COMP.I
9      RETURN
10     END
```

```
-----  
1      SUBROUTINE ECRD(SCARR,IADDLC,NW,IERROR)
2      C
3      C      LTSS/7600 ROUTINE TO COPY DATA FROM LARGE CORE TO SMALL CORE
4      C
5      C      SCARR = SMALL CORE ARRAY INTO WHICH DATA IS TO BE COPIED
6      C      IADDLC = LARGE CORE ADDRESS FROM 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(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(JOBTP)
```

```
3      C      LTSS ROUTINE FOR RETURNING THE JOB ID
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
6      C
7      C      JOBID=1AHYAQUI=LTSS
8      C      RETURN
9      C      END
```

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

```
1      C      SUBROUTINE GETLCM(IFLLCM)
2      C
3      C      LTSS/7600 ROUTINE TO RETURN THE AMOUNT OF LCM AVAILABLE TO THE
4      C      USER
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      C      COMMON/GUBCOM/IDUM(1)
9      C      IFLLCM=IDUM(16)-IDUM(15)
10     C      RETURN
11     C      END
```

```
1      C      SUBROUTINE GETTPE(IDUM)
2      C
3      C      LTSS ROUTINE TO DUMMY UP TAPE FETCHING AVAILABLE ON CROS/7600
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
6      C
7      C      RETURN
8      C      END
```

```

1      SUBROUTINE LCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)
2
3      C   LTSS/7600 ROUTINE TO READ OR WRITE LCM FROM OR TO A DISK FILE
4
5      C   FWA    = FIRST LCM ADDRESS
6      C   NWRDS = NO. OF WORDS TO TRANSFER
7      C   IFILE   = LOGICAL UNIT NO. OF DISK FILE
8      C   IFLAG   = READ OR WRITE FLAG
9      C           = 0 = READ DISK
10     C           = 1 = WRITE DISK
11     C   IRET    = RETURN FLAG
12     C           = 0 = RETURN IMMEDIATELY AFTER ISSUING THE I/O REQUEST
13     C           = 1 = WAIT UNTIL I/O IS COMPLETED BEFORE RETURNING
14     C   IERROR  = ERROR FLAG
15     C           = 0 = NO ERROR
16     C           = 1 = FRROR
17     C           = -1 = END-OF-FILE ON INPUT
18
19     C   WRITTEN BY J.L.NORTON,LASL T-3,1975
20
21     C   LCM FWLCMC
22     C   COMMON/FWLCMC/AA1(1)
23     C   INTEGER FWA
24
25     C   CLEAR ERROR FLAG
26
27     C   IERROR=0
28
29     C   SEE WHETHER REQUEST IS READ INTO OR WRITE FROM LCM
30
31     C   IF(IFLAG,NE,0) GO TO 10
32
33     C   REQUEST IS WRITE LCM (READ DISK)
34
35     C   BUFFER INC(IFILE,1)(AA1(FWA+2),AA1(FWA+NWRDS+1))
36     C   GO TO 20
37
38     C   REQUEST IS READ LCM (WRITE DISK)
39
40     C   10 CONTINUE
41     C   BUFFER OUT(IFILE,1)(AA1(FWA+2),AA1(FWA+NWRDS+1))
42
43     C   SEE IF USER WISHES TO WAIT UNTIL I/O IS COMPLETE
44
45     C   20 CONTINUE
46     C   IF(IRET,EQ,0) RETURN
47
48     C   YES, WAIT FOR I/O TO COMPLETE
49
50     C   30 IF(UNIT,IFILE) 30,40,50,60
51
52     C   I/O SUCCESSFULLY COMPLETED. ALL DONE.
53
54     C   40 CONTINUE

```

```
55      RETURN
56      C
57      C      END-OF-FILE, SET ERROR FLAG AND RETURN.
58      C
59      50 CONTINUE
60      IERROR=1
61      RETURN
62      C
63      C      I/O ERROR, SET ERROR FLAG AND RETURN.
64      C
65      60 CONTINUE
66      IERROR=1
67      RETURN
68      END
```

```
-----  
1      FUNCTION LOCF(I)
2      C
3      C      LTSS ROUTINE TO HANDLE LOCATION FUNCTION
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      C      LOCF=,LOC,I
8      C      RETURN
9      C      END
```

```
-----  
1      SUBROUTINE NCODE(NC,IFORM,INTAB,NIN,OUTTAB)
2      C
3      C      LTSS ROUTINE TO SIMULATE CDC ENCODE STATEMENT
4      C
5      C      ENCODE(NC,IFORM,OUTTAB) (INTAB(I),I=1,NIN)
6      C
7      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
8      C
9      C      DIMENSION INTAB(NIN),OUTTAB(1),ITEMP(10),IFORM(1)
10     DO 10 I=1,10
11     10 ITEMPI=IFORM(I)
12     REWIND 63
13     NW=NC/10+1
14     IF(MOD(NC,10).EQ.0) NW=NW-1
15     WRITE(63,ITEMP)(INTAB(I),I=1,NIN)
16     READ(63,20)(OUTTAB(I),I=1,NW)
17     RETURN
18     C
19     20 FORMAT(15A10)
20     END
```

```
1      SUBROUTINE OPENIT(IFILE,MODE)
2      C
3      C      LTSS/7600 ROUTINE TO OPEN A FILE
4      C
5      C      IFILE = LOGICAL UNIT NO. OF THE FILE
6      C      MODE  = TYPE OF FILE
7      C          = 0 = BCD
8      C          = 1 = BINARY
9      C
10     C      WRITTEN BY J.L.NORTON,LASL T=3,1975
11     C
12     IF(MODE.EQ.0) RETURN
13     LENGTH=10000000
14     IOC=0
15     CALL TPGEN(IFILE,ITP)
16 10  CONTINUE
17     CALL CREATE(ITP,LENGTH,IOC)
18     IF(IOC.GE.0) GO TO 20
19     LENGTH=FLOAT(LENGTH)*.9
20     GO TO 10
21 20  CONTINUE
22     CALL ASSIGN(IFILE,0,ITP)
23     RETURN
24     END
```

```
1      INTEGER OR
2      FUNCTION OR(I,J)
3      C
4      C      LTSS ROUTINE TO TAKE THE BOOLEAN UNION OF TWO VARIABLES
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      OR=I.UN.J
9      RETURN
10     END
```

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

```
7   C  
8     ORR=I,UN,J  
9     RETURN  
10    END
```

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

```
-----  
1      SUBROUTINE SCBUFF(FWA,NWRDS,IFILE,IFLAG,IRET,IERROR)  
2      C  
3      C      LTSS/7600 ROUTINE TO READ OR WRITE SCM FROM OR TO A DISK FILE  
4      C  
5      C      SEE LCBUFF FOR ARGUMENT DOCUMENTATION  
6      C  
7      C      FWA = BEGINNING OF SCM BLOCK TO BE WRITTEN  
8      C  
9      C      WRITTEN BY J.L.NORTON,LASL T-3,1975  
10     C  
11     DIMENSION FWA(1)  
12     C      LCM FWLCMC  
13     C      COMMON/FWLCMC/AA1(1)  
14     C      COMMON/LCSCRC/ILSIZE,IFWASC  
15     C  
16     C      CLEAR ERROR FLAG  
17     C  
18     C      IERROR=0  
19     C  
20     C      SEE IF THE LCM SCRATCH AREA IS LARGE ENOUGH TO HOLD THE  
21     C      SCM BLOCK  
22     C  
23     C      IF(NWRDS.GT.ILSIZE) CALL UNCLE(4,6HSCBUFF,25,  
24     C      1 25HNOT ENOUGH LCM FOR BUFFER)  
25     C  
26     C      YES. SEE IF DISK IS TO BE READ OR WRITTEN.  
27     C  
28     C      IF(IFLAG.NE.0) GO TO 60  
29     C
```

```

30   C      READ DISK INTO THE LCM SCRATCH AREA
31   C
32   C      BUFFER IN(IFILE,1)(AA1(IFWASC+2),AA1(IFWASC+NWRDS+1))
33   C
34   C      WAIT FOR I/O TO COMPLETE
35   C
36   10 IF(UNIT,IFILE) 10,20,30,40
37   20 CONTINUE
38   C
39   C      COPY LCM INTO SCM
40   C
41   C      CALL BLOCKCOPY(AA1(IFWASC+2),FWA,NWRDS)
42   C      RETURN
43   C
44   C      EOF ENCOUNTERED, SET THE ERROR FLAG,
45   C
46   30 CONTINUE
47   IERROR=-1
48   RETURN
49   C
50   C      DISK ERROR OR INPUT RECORD WAS SHORTER THAN EXPECTED
51   C
52   40 CONTINUE
53   C
54   C      GET THE LENGTH OF THE RECORD AND IF NOT ZERO,COPY IT TO SCM
55   C
56   C      L=LENGTH(IFILE)
57   C      IF(L.NE.0) CALL BLOCKCOPY(AA1(IFWASC+2),FWA,L)
58   C
59   C      SET THE ERROR FLAG
60   C
61   50 CONTINUE
62   IERROR=1
63   RETURN
64   C
65   C      DISK IS TO BE WRITTEN
66   C
67   60 CONTINUE
68   C
69   C      FIRST COPY SCM BLOCK TO LCM SCRATCH AREA
70   C
71   C      CALL BLOCKCOPY(FWA,AA1(IFWASC+2),NWRDS)
72   C
73   C      WRITE LCM TO DISK
74   C
75   C      BUFFER OUT(IFILE,1)(AA1(IFWASC+2),AA1(IFWASC+NWRDS+1))
76   C
77   C      WAIT FOR I/O TO COMPLETE IF SO REQUESTED
78   C
79   C      IF(IRET.EQ.0) RETURN
80   70 IF(UNIT,IFILE) 70,80,30,50
81   80 CONTINUE
82   RETURN
83   END

```

```
1      SUBROUTINE SDG(X)
2      C
3      C      LTSS/7600 ROUTINE TO FIND SINES AND COSINES
4      C      OF ARGUMENTS IN DEGREES
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      DATA DEGRAD/.017453292519943/
9      Y=X*DEGRAD
10     SDG=SIN(Y)
11     RETURN
12     ENTRY CDG(X)
13     Y=X*DFGRAD
14     SDG=COS(Y)
15     RETURN
16     END
```

```
1      INTEGER SHIFT
2      FUNCTION SHIFT(IX,N)
3      C
4      C      LTSS ROUTINE TO PERFORM BIT SHIFTING
5      C
6      C      IX IS THE QUANTITY TO BE SHIFTED
7      C      N IS THE NO. OF BITS TO SHIFT. N POSITIVE MEANS LEFT END-AROUND
8      C      SHIFT AND N NEGATIVE MEANS RIGHT END-OFF SHIFT.
9      C
10     C      WRITTEN BY J.L.NORTON,LASI. T-3,1975
11     C
12     IF(N.GE.0) GO TO 10
13     NP=N
14     SHIFT=IX,SHR,NP
15     RETURN
16     10 CONTINUE
17     SHIFT=IX,SHL,N
18     RETURN
19     END
```

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

```
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      ABSOLUTE IADDR(1)
9      DIMENSION IADDR(1)
10     IADDR(IADD)=IVAR
11     RETURN
12     END
```

```
-----  
1      C      SUBROUTINE SYINIT
2      C
3      C      LTSS/7600 ROUTINE TO DO SYSTEM-DEPENDENT INITIALIZATION
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      LCM FWLCMC,YLC1,YLC2,YLC3,(FILMLB),LCMSCR
8      COMMON/FWLCMC/FWLCKM
9      COMMON/YLC1/AA1(72300)/YLC2/AA2(5000)
10     COMMON/YLC3/PAXY(10,200)/FILMLB/FLMBUF(4000)
11     COMMON/LCMSCR/SCRTH(1000)
12     *      ----- BEGIN COMDECK PARAM -----  
13     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,  
14     NLCP3,NLCP4,IFLMSZ
15     *      ----- END COMDECK PARAM -----  
16     COMMON/LCSCRC/ILSIZE,IFWASC
17     C
18     C      CHANGE THE DROPFILE NAME TO +YAQUIB
19     C
20     CALL CHANGE(7H+YAQUIB)
21     C
22     C      INITIALIZE THE FILM ROUTINES.  THE ARGUMENTS ARE AS FOLLOWS --
23     C      FOR IDENT80 -
24     C          (1) LOGICAL UNIT NO. OF THE FILM FILE
25     C          (2) ARRAY IN LCM TO BE USED AS A FILM BUFFER
26     C          (3) SIZE OF THE FILM BUFFER
27     C          (4) SIZE OF DISK FILE FOR FILM
28     C          (5) TYPE OF MICROFILM OUTPUT
29     C              - 35 FOR 35 MM
30     C              - 105 FOR MICROFICHE
31     C          (6) NOT USED
32     C      - FOR HEAD80 -
33     C          (1) LOGICAL UNIT NO. OF THE FILM FILE
34     C          (2) NO. OF CHARACTERS IN THE HEADER (60 MAXIMUM)
35     C          (3) THE HEADER ARRAY
36     C          (4) FILM IDENTIFICATION ARRAY
37     C      FOR KEEPFILM -
38     C          (1) LOGICAL UNIT NO. OF THE FILM FILE NOT TO BE GIVEN
39     C              TO THE SYSTEM
40
41     CALL FILM80
42     CALL IDENT80(12,FLMBUF,IFLMSZ,1000000,105,1DUM)
43     CALL HEAD80(12,10,10HT3JLN,9HBOX T3JLN)
```

```

44      CALL KEEPFLM(12)
45      C
46      C      INITIALIZE THE SIZE OF THE LCM SCRATCH AREA
47      C
48      C      JLSTZE=NLCP4
49      C
50      C      SET THE ADDRESS OF THE FIRST WORD OF THE SCRATCH AREA
51      C
52      C      IFWASC=.LOC,SCRATCH(1)=.LOC,FWLGM
53      C      RETURN
54      C      END
-----
```

```

1      SUBROUTINE TPGEN(IFILE,ITP)
2      C
3      C      LTSS/7600 ROUTINE TO TAKE AN INTEGER LOGICAL UNIT NO. IFILE
4      C      AND CONVERT IT INTO A DISK FILE NAME OF THE FORM TAPEN
5      C      OR TAPENN WHICH IS RETURNED LEFT-JUSTIFIED IN ITP
6      C
7      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
8      C
9      C      INTEGER OR,SHIFT,AND
10     C      DIMENSION INUM(10)
11     C      DATA INUM/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
12     C
13     C      INITIALIZE
14     C
15     C      IFILEP=IFILE
16     C      IC=-24
17     C      ITP=4HTAPE
18     C      MASK=77777777777778
19     C
20     C      SFE IF LOGICAL UNIT NO. IS ONE OR TWO DIGITS
21     C
22     C      IF(IFILEP,LE,9) GO TO 10
23     C
24     C      TWO DIGITS. ISOLATE THE FIRST DIGIT.
25     C
26     C      IT=IFILEP/10+1
27     C      IF(MOD(IFILEP,10),EQ,0) IT=IT-1
28     C
29     C      PUT THE FIRST DIGIT INTO PLACE
30     C
31     C      ITP=OR(AND SHIFT(INUM(IT),IC),MASK),ITP)
32     C
33     C      MODIFY PARAMETERS FOR SECOND DIGIT
34     C
35     C      IC=IC-6
36     C      MASK=77777777778
37     C      IFILEP=IFILEP-10*(IT-1)
38     C      10 CONTINUE
39     C
```

```
40      C      PUT THE LAST DIGIT INTO PLACE
41      C
42      C      ITP=OR(AND(SHIFT(INUM(IFILEP+1),IC),MASK),ITP)
43      C      RETURN
44      C      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/YSC5/RESTART,FILM,PAPER,IPD,IFD
8      COMMON/OBERR/IERR(121)
9      COMMON/GOBCOM/IG(1)
10     ABSOLUTE TWO(2)
11     INTEGER TWO,SHIFT
12     LOGICAL FILM
13     COMMON/IEDMP/IDUMP(16)
14
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 PDMPK
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
```

```
46      C
47      C      IF(.,NOT,FILM) GO TO 40
48      C
49      C      PICK THE SMALL CORE FIELD LENGTH OUT OF THE EXCHANGE PACKAGE
50      C
51      C      IFLS=(IDUMP(3),INT,(.COMP,7777777778)),SHR,36
52      30 CONTINUE
53      C      IFLS=IFLS-1
54      C      CALL DMP(0,IFLS,12)
55      40 CONTINUE
56      C
57      C      TERMINATE BUT SAVE THE DROPFILE
58      C
59      C      CALL EXIT(2)
60      C      RETURN
61      C
62      50 FORMAT(/1H ,14HORDERLIB ERROR,I3,21H OCCURRED AT LOCATION,07,
63      129H WHICH IS THE CALL TO ROUTINE,A10)
64      END
```

```
1      SUBROUTINE TTYTST(IFLAG)
2      C
3      C      LTSS ROUTINE TO SEE IF DROPFILE VARIABLE HAS BEEN CHANGED TO
4      C      SIGNAL JOB TERMINATION
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      COMMON/OFFC/ISWTCH
9      DATA ISWTCH/-1/
10     IFLAG=0
11     IF(ISWTCH,EQ,0) IFLAG=1
12     RETURN
13     END
```

B. CROS-Dependent Code

```
1      PROGRAM MAIN(INP,OUT,FILM,FSET7,FSET8,FSET5=INP,FSET6=OUT,FSET12=
1 FILM,FSET3,FSET59=OUT)
2
3      C
4      C      CROS/7600 MAIN ROUTINE FOR YAQUI
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      C      CALL YAQUI
9      C      END
```

```
1      SUBROUTINE CLOSIT(I)
2      C
3      C      CROS/7600 ROUTINE FOR DESTROYING DISK FILES
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
6      C
7      C      CALL AFSREL(I,0,0,0)
8      C      RETURN
9      C      END
```

```
1      SUBROUTINE GETJOB(JOBID)
2      C
3      C      CROS/7600 ROUTINE FOR RETURNING THE JOB ID
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
6      C
7      C      CALL GETQ(4LKJBN,JOBID)
8      C      RETURN
9      C      END
```

```
1      SUBROUTINE GETJTL(TL)
2      C
3      C      CROS/7600 ROUTINE TO RETURN THE JOB TIME LIMIT IN SECONDS
4      C
5      C      CALL GETQ(4LKJLM,II)
6      C      TL=27.5E-9*FLOAT(II)
7      C      RETURN
8      C      END
```

```
1      SUBROUTINE GETLCM(ISIZE)
2      C
3      C      CROS/7600 ROUTINE TO RETURN THE AMOUNT OF LARGE CORE MEMORY
4      C      AVAILABLE TO THE JOB FOR DATA STORAGE
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
7      C
8      C      CALL GETQ(3LKMA,ISIZE)
9      C      RETURN
10     C      END
```

```
1      SUBROUTINE GETTPE(TAPE)
2      C
3      C      CROS/7600 ROUTINE FOR CHECKING TAPE LABEL AND STAGING TAPE
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1975
6      C
7      C      INTEGER TAPE,AND
8      C
9      C      TAPE WAS READ, SEE IF IT WAS LEGAL.
10     C
11     C      IF(AND(TAPE,777777000000000000000000B),NE,3LXX0) CALL UNCLE(4,6HGETTPE
12     C      1,29,29HTAPE NO, INPUT ERROR)
13     C
14     C      GO TO STAGE IN THE TAPE
15     C
16     C      CALL STAGE(7,TAPE,IEFLAG)
17     C
18     C      SEE IF STAGE WAS SUCCESSFUL
19     C
20     C      IF(IEFLAG,NE,9) CALL UNCLE(4,6HGETTPE,23,
21     C      1 23HUNSUCCESSFUL TAPE STAGE)
22     C
23     C      YES, ALL DONE.
24     C
25     C      RETURN
26     C      END
```

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

```

5   C      IFILE = LOGICAL UNIT NO. OF THE FILE
6   C      MODE  = TYPE OF FILE
7   C          - 0 = BCD
8   C          - 1 = BINARY
9   C
10  C      WRITTEN BY J.L.NORTON,LASL T=3,1975
11  C
12  C      CALL OPEN(IFILE,0,0,0,0,0,10000,0,0)
13  C      RETURN
14  C      END

```

```

1      SUBROUTINE STAGE(IF,LABEL,IFLAG)
2      C      ROUTINE TO STAGE IN A TAPE WITH LABEL=XX0NNNN INTO FILESET IF
3      C
4      C      WRITTEN BY J.L.NORTON,LASL TD=3,1973
5      C
6      C      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
7      C      DIMENSION IMDISP(5)
8      C      DATA IMDISP/4,0,0,0,0/
9
10     C      IFLAG IS RETURNED ZERO IF THE TAPE WAS SUCCESSFULLY STAGED. IT IS
11     C          RETURNED ONE IF FOUR STAGE ATTEMPTS FAILED.
12
13     C      IFLAG=0
14
15     C      OPEN THE FILE TO A LARGE TRACK SECTOR LIMIT
16
17     C      CALL OPEN(IF,0,0,0,0,0,10000,0,0)
18
19     C      ZERO OUT THE LAST TWO CHARACTERS OF THE TAPE LABEL
20
21     C      IMDISP(5)=LABEL,AND,(,NOT,77778)
22
23     C      INITIALIZE THE NO. OF STAGE ATTEMPTS
24
25     C      ICNT=0
26
27     C      GET JOB CLASSIFICATION
28
29     C      CALL GETQ(4LKCLA,JCLASS)
30     C      KCLASS=1LG
31     C      IF(JCLASS,NE,5) KCLASS=1LU
32
33     C      INITIATE THE STAGE
34
35     C      10 CALL CREATE(IF,KCLASS,2LST,0,0,0,IMDISP,0,0,10000,0)
36
37     C      INCREMENT THE STAGE COUNT
38
39     C      ICNT=ICNT+1

```

```

41      C      SEE IF THERE WERE ANY PARITY ERRORS
42      C
43      C      CALL PARITY(IF,ICHECK)
44      C      IF(ICHECK,NE,0) GO TO 30
45
46      C      NO. PRINT MESSAGE AND RETURN.
47      C
48      C      DO 20 IPX=IPD,IFD,6
49      20 WRITE(IPX,50) LABEL,IF,IF
50      RETURN
51
52      30 CONTINUE
53
54      C      YES, SEE IF THIS WAS THE FOURTH STAGE.
55      C
56      C      IF(ICNT,LT,4) GO TO 40
57
58      C      YES, SET THE ERROR FLAG AND RETURN.
59      C
60      C      IFLAG=1
61      RETURN
62
63      40 CONTINUE
64
65      C      NO, RELEASE THE FILE AND TRY AGAIN.
66
67      C      CALL AFSREL(IF,0,0,0)
68
69      C      GO TO 10
70
71      50 FORMAT(1H0,5HTAPE ,A8,35H HAS BEEN SUCCESSFULLY STAGED INTO ,
72      1 8HFILESET ,I2,2H (,A8,1H))
73      END

```

```

1      SUBROUTINE SYINIT
2
3      C      CROS/7600 ROUTINE TO PERFORM YAQUI SYSTEM INITIALIZATION
4
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6
7      C      CALL SETQ(4LKDPX,0)
8      CALL OPEN(3LDOUT,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

```

```

1      SUBROUTINE TRAP(RCOVER)
2
3      CROS/7600 ROUTINE TO INITIALIZE FOR RECOVERY FROM FATAL EXECUTION
4      ERRORS
5
6      RCOVER IS A SUBROUTINE TO CALL AFTER INTERCEPTING THE ERROR
7
8      WRITTEN BY J.L.NORTON,LASL T=3,1975
9
10     DIMENSION IDUMP(16)
11     COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
12     INTEGER SHIFT,AND,COMP
13     LOGICAL FILM
14     ASSIGN 1W TO IGOXIT
15     CALL XIT(IGOXIT)
16
17     IABORT IS THE NO. OF ABORT TRAPS THAT HAVE OCCURRED
18
19     IABORT=0
20     RETURN
21 10  CONTINUE
22     CALL GETHPK(IDUMP)
23     CALL PABORT(IDUMP)
24     IF(.NOT.FILM) GO TO 20
25     IFLS=SHIFT(AND(IDUMP(4),COMP(7777777777B)),+36)
26     IFLS=IFLS-1
27     CALL DMP(0,IFLS,12)
28 20  CONTINUE
29
30     CODE HAS ABORTED. INCREMENT THE ABORT COUNT.
31
32     IABORT=IABORT+1
33
34     CALL THE RECOVERY ROUTINE.
35
36     CALL RCOVER(IABORT)
37     RETURN
38     END

```

C. KRONOS-Dependent Code

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

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

```
1      SUBROUTINE GETJOB(JOBID)
2      C
3      C      KRONOS/6600 ROUTINE TO RETURN THE JOB ID
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      CALL GETJN(JOBID)
8      RETURN
9      END
```

```
1      SUBROUTINE GETJTL(TL)
2      C
3      C      KRONOS/6600 ROUTINE TO RETURN THE JOB TIME LIMIT IN SECONDS
4      C      AS A FLOATING POINT NO.
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      CALL GETTL(ITL)
9      TL=ITL
10     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      C      IFLLCM=500000  
9      C      RETURN  
10     C      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      C      RETURN  
8      C      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      C      RETURN  
8      C      END
```

```
1      SUBROUTINE SYINIT  
2      C
```

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

```
1          IDENT  TRAP
2
3
4
5      *
6      *      KRONOS/6600 ROUTINE TO INTERCEPT HARDWARE OR SOFTWARE ABORTS
7      *
8      *      REGISTERS AND SMALL CORE ARE DUMPED TO OUTPUT,A MESSAGE IS PUT
9      *      IN THE SYSTEM DAYFILE,AND THE EXTERNAL SUPPLIED IN THE CALL
10     *      TO TRAP (CALL TRAP(RCOVER)) IS CALLED WHENEVER AN ERROR
11     *      IS DETECTED
12     *
13     *      WRITTEN BY J.L.NORTON,LASL T-3,1975
14     *
15          ENTRY  TRAP
16      TRAP    DATA    0
17          SX6    B1
18          SA6    SAVA
19          EREXIT RCOVER
20          JP     TRAP
21      RCOVER   SYSTEM DMP,R,0,0
22          SYSTEM DMP,R,135000B,0
23          MESSAGE MESS,3,R
24          SA1    SAVA
25          SR1    X1+1
26          JP     B1
27      MESS     DATA    17L--- JOB ABORT ---
28          DATA    0
29      SAVA     DATA    0
30          END
```

D. KRONOS/CROS-Dependent Code

```
1      FUNCTION AND(I,J)
2      C
3      C      KRONOS/6600 AND CROS/7600 ROUTINE TO DO BOOLEAN INTERSECTION
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      C      INTEGER AND
8      C      AND=I,AND,J
9      C      RETURN
10     C      END
```

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

```
1      FUNCTION COMP(I)
2      C
3      C      KRONOS/6600 AND CROS/7600 ROUTINE TO COMPLEMENT A WORD
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      C      INTEGER COMP
8      C      COMP=NOT,I
9      C      RETURN
10     C      END
```

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

```
6      C
7      INTEGER COMPR
8      COMPR=NUT,I
9      RETURN
10     END
```

```
-----  
1          IDENT GETIT
2
3      *
4      *      CROS/7600 AND KRONOS/6600 FUNCTION GETIT(IADD) TO PICK UP THE
5      *      CONTENTS OF ABSOLUTE LOCATION IADD
6      *
7          ENTRY GETIT
8          VFD   42/0HGETIT,18/1
9          GETIT DATA  0
10         SA1   R1           PICK UP THE ADDRESS IN X1
11         SA2   X1           PICK UP C(IADD)
12         BX6   X2           PUT INTO X6 FOR FUNCTION
13         JP    GETIT
14         END
```

```
-----  
1          SUBROUTINE LCBUFF(FWA,NHRDS,IFILE,IFLAG,IRET,IERROR)
2          C
3          C      KRONOS/6600 AND CROS/7600 ROUTINE TO READ OR WRITE LCM FROM
4          C      OR TO A DISK FILE
5          C
6          C      FWA      = FIRST LCM ADDRESS
7          C      NHRDS    = NO. OF WORDS TO TRANSFER
8          C      IFILE     = LOGICAL UNIT NO. OF DISK FILE
9          C      IFLAG     = READ OR WRITE FLAG
10         C          - 0 = READ DISK
11         C          - 1 = WRITE DISK
12         C      IRET      = RETURN FLAG
13         C          - 2 = RETURN IMMEDIATELY AFTER ISSUING THE I/O REQUEST
14         C          - 1 = WAIT UNTIL I/O IS COMPLETED BEFORE RETURNING
15         C          (NOT FUNCTIONAL ON CROS/7600 OR KRONOS/6600)
16         C      IERROR    = ERROR FLAG
17         C          - 0 = NO ERROR
18         C          - 1 = ERROR
19         C          -1 = END-OF-FILE ON INPUT
20         C
21         C      WRITTEN BY J.L.MORTON, LASL T-3, 1975
22         C
23         C      ----- BEGIN COMDECK PARAM -----  
CCMNON/PCom/NSCP1,ITARP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
```

```

25      1 ALCP3,ALCP4,IFLMSZ
26      * ----- END COMDECK PARAM -----
27      COMMON/YSC1/AASC(1)
28      INTEGER FWA,FNAP
29      C
30      C      CLEAR ERROR FLAG
31      C
32      C      TERROR=0
33      C
34      C      FWAP IS THE BEGINNING LCM ADDRESS OF THE PORTION OF LCM BEING
35      C      READ OR WRITTEN
36      C
37      C      FNAP=FWA
38      C
39      C      NWRSO IS THE TOTAL NO. OF WORDS OF LCM THAT HAVE BEEN READ
40      C      OR WRITTEN
41      C
42      C      NWRSDE=0
43      C
44      C      LOOP FOR READING TO OR WRITING FROM LCM FROM OR TO DISK IN BLOCKS
45      C
46      10 CONTINUE
47      C
48      C      NW IS THE NO. OF LCM WORDS TO TRANSFER THIS TIME. IT IS EQUAL
49      C      TO THE SIZE OF THE SCM BUFFER UNLESS THE NO. OF LCM WORDS
50      C      LEFT TO TRANSFER IS LESS THAN THE BUFFER SIZE. IN THE LATTER
51      C      CASE, NW IS JUST SET TO THE NO. OF REMAINING WORDS.
52      C
53      C      NW=NSCP1
54      C      NWTEST=NWRDSC+NW
55      C      IF(NWTEST.GT.NWRSO) NW=NWRSO-NWRDSD
56      C
57      C      SEE WHETHER REQUEST IS READ INTO OR WRITE FROM LCM
58      C
59      C      IF(IFLAG.NE.0) GO TO 80
60      C
61      C      REQUEST IS WRITE LCM (READ DISK)
62      C
63      C      BUFFER INC(IFILE,1)(AASC(1),AASC(NW))
64      C
65      C      WAIT FOR I/O TO COMPLETE
66      C
67      20 IF(UNIT,IFILE) 20,50,40,30
68      C
69      C      ERROR OCCURRED IN DISK TRANSFER (EITHER UNEXPECTED EOF, PARITY
70      C      ERROR, OR RECORD SHORTER THAN EXPECTED). SET THE ERROR FLAG
71      C      AND RETURN.
72      C
73      30 CONTINUE
74      C      IERROR=1
75      C      RETURN
76      40 CONTINUE
77      C      IERROR=-1
78      C      RETURN
79      50 CONTINUE
80      C      CALL ECWR(AASC,FWAP,NW,IDUM)
81      C

```

```

82      C      BLOCK TRANSFER COMPLETED. UPDATE NARDSD AND SEE IF THERE IS
83      C      MORE DATA LEFT TO TRANSFER.
84      C
85      62 CONTINUE
86      NARDSD=NARDSD+NW
87      IF(IFLAG,NF,0) GO TO 70
88      NLENGTH(IFILE)
89      IF(N,NE,NW) GO TO 34
90      70 CONTINUE
91      IF(NARDSD,GF,NWRCS) RETURN
92      F+AP=FWAP+NW
93      GO TO 10
94      C
95      C      REQUEST IS READ LCM (WRITE DISK)
96      C
97      80 CONTINUE
98      CALL FCRD(AASC,FWAP,NW,1DUM)
99      BUFFER OUT(TFILE,1)(AASC(1),AASC(NW))
100     C
101     C      WAIT FOR I/O TO COMPLETE
102     C
103     22 IF(UNIT,IFILE) 9H,6H,4H,3H
104     END

```

```

1      SUBROUTINE MCODE(NC,IFORM,INTAB,NIN,OUTTAB)
2      C
3      C      KRDOS/6600 AND CROS/7600 ROUTINE TO SIMULATE THE CDC ENCODE
4      C      STATEMENT
5      C
6      C      ENCODE(NC,IFORM,OUTTAB) (INTAB(I),I=1,NIN)
7      C
8      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
9      C
10     DIMENSION INTAB(NIN)
11     INTEGER OUTTAB
12     ENCODE(NC,IFORM,OUTTAB) INTAB
13     RETURN
14     END

```

```

1      FUNCTION OR(I,J)
2      C
3      C      KRDOS/6600 AND CROS/7600 ROUTINE TO DO BOOLEAN UNION
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C

```

```
7      INTEGER OP
8      OR=I,OR,J
9      RETURN
10     END
```

```
1      FUNCTION ORR(I,J)
2      C
3      C      KRONOS/6600 AND CROS/7600 ROUTINE TO DO BOOLEAN UNION
4      C      AND RETURN THE RESULT IN A FLOATING POINT VARIABLE
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
7      C
8      INTEGER ORR
9      ORR=I,OR,J
10     RETURN
11     END
```

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

```
1      SUBROUTINE SCRUFF(FWA,NWDS,TFILE,TFLAG,IRET,IERROR)
2      C
3      C      KRONOS/6600 AND CROS/7600 ROUTINE TO READ OR WRITE SCM FROM
4      C      OR TO A DISK FILE
5      C
6      C      SEE LCBUFF FOR ARGUMENT DOCUMENTATION
7      C
8      C      FWA = BEGINNING OF SCM BLOCK TO BE WRITTEN
9      C
10     C      WRITTEN BY J.L.NORTON,LASL T-3,1975
11     C
```

```

12      DIMENSION FWA(1)
13      IERROR=0
14      IF(IFLAG.NE.0) GO TO 10
15      BUFFER IN(IFILE,1)(FWA(1),FWA(NWRDS))
16      GO TO 20
17      10 CONTINUE
18      BUFFER OUT(IFILE,1)(FWA(1),FWA(NWRDS))
19      20 CONTINUE
20      IF(TRET.EQ.0) RETURN
21      30 IF(UNIT,IFILE) 30,40,50,60
22      40 CONTINUE
23      IF(IFLAG.NE.0) RETURN
24      N=LENGTH(IFILE)
25      IF(N.NE.NWRDS) GO TO 60
26      RETURN
27      50 CONTINUE
28      IERROR=-1
29      RETURN
30      60 CONTINUE
31      IERROR=1
32      RETURN
33      END

```

```

1      IDENT SHIFT
2
3
4
5      ENTRY SHIFT
6
7      *      KRONOS/66MA AND CROS/76MA ROUTINE TO REPLACE THE LAST RUN
8      *      COMPILER IN-LINE SHIFT FUNCTION. THIS IS NECESSARY BECAUSE
9      *      OF THE NEED TO INCLUDE SHIFT IN AN INTEGER STATEMENT FOR
10     *      LTSS COMPATIBILITY. HOWEVER, DOING SO FORCES A CALL TO AN
11     *      EXTERNAL FUNCTION.
12     *
13     *      WRITTEN BY J.L.NORTON,LASL T-3,1975
14     *
15      VFD    42/UHSHIFT,18/2
16      SHIFT   DATA    0
17          SAI    B2
18          SA2    B1
19          ZR     X1,ZERO
20          PL     X1,LEFT
21          PX3   -X1
22          S43   X3
23          AX6   R3,X2
24          JP     SHIFT
25      LEFT    SB3   X1
26          LX6   R3,X2
27          JP     SHIFT
28      ZERO   BX0   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 81 PICK UP IWORD IN X1
11 SA2 82 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 DROPPFILE
5 C
6 C WRITTEN BY J.L.NORTON,LASL T-3,1975
7 C
8 IFLAG=0
9 RETURN
10 END

E. System-Independent Code

```
1      SUBROUTINE YAQUI
2
3      C
4      C      YAQUI IS A TWO-DIMENSIONAL FLUID DYNAMICS CODE THAT COMBINES
5      C          THE ICE (IMPLICIT CONTINUOUS EULERIAN) AND ALE (ARBITRARY
6      C          LAGRANGIAN-EULERIAN) METHODS, ALLOWING CALCULATION OF FLOWS
7      C          AT ALL SPEEDS.
8
9      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,HANS RUPPEL,AND C.W.HIRT,LA8L T=3
10     C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LA8L T=3,1974
11
12     *      ----- BEGIN COMDECK PARAM -----
13     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
14     1 NLCP3,NLCP4,IFLMSZ
15     *      ----- END COMDECK PARAM -----
16     C      ***** LCM IS SET UP IN THE FOLLOWING FASHION *****
17     C          BLOCK 1,NLCPI WORDS LONG,ARRAY DATA
18     C          BLOCK 2,NLCP2 WORDS LONG,PARTICLE POSITIONS AND MASSES
19     C          BLOCK 3,NLCP3 WORDS LONG,TIME-DEPENDENT PARTICLE POSITIONS
20     C          BLOCK 4,IFLMSZ WORDS LONG,FILE BUFFER FOR LTSS
21     C          BLOCK 5,NLCP4 WORDS LONG,SCRATCH AREA
22     C      *****
23
24     *      ----- BEGIN COMDECK YAQSC -----
25     LOGICAL RESTRT,FILM,PAPER,TURB
26     REAL LAM,MU
27     C
28     COMMON/YSC1/AABC(NSCP1)
29     COMMON/YSC1/AASC(9600)
30     COMMON/YSC2/AA(1),ANC,A0,ABFAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
31     1 DTO10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
32     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,PREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
33     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
34     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
35     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
36     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
37     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
38     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
39     COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
40     1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
41     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
42     3 XCNVLD,FIXRO,FIXL,IXRO,IXLO,ISVN,JSVN,QMN,QMX,WMAX,JNM,T2,TLIM,
43     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTR,TMASS,DTVSAV,DTCSAV,IDTV
44     5 ,JDTV,IDTC,JDTA,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMX,ITM,JTM,ITG,JTG
45     6 ,TMASSV,WMAXEF,RHINEF,TSTRTD
46     C
47     COMMON/YSC4/ITAB(ITABP)
48     COMMON/YSC4/ITAB(1000)
49     *      ----- END COMDECK YAQSC -----
50     COMMON/FTABC/FTAB(2)
51     INTEGER AA1
52     EXTERNAL YEXIT
53     DATA FTAB/6,12/
54     DATA NSCP1/9600/,ITABP/1000/,ITABXP/101/,ITABYP/151/
55     DATA IPFB/200/,NP1/10/,NP2/200/,NLCPI/72300/,NLCP2/5000/
56     DATA NLCP3/2000/,NLCP4/1000/,IFLMSZ/4000/
57     DATA TP/0./
```

```

58      C      DO ANY NECESSARY SYSTEM INITIALIZATION
59      C
60      C      CALL SYINIT
61      C
62      C      INITIALIZE THE ERROR RECOVERY ROUTINE
63      C
64      C      CALL TRAP(YEXIT)
65      C
66      C      GET THE CP TIME AT JOB STARTUP
67      C
68      C
69      C      CALL SECOND(TP)
70      C      WRITE(S9,30) TP
71      C
72      C      INITIALIZE
73      C
74      C      CALL YINIT
75      C
76      C      CHECK TYPE OF RUN
77      C
78      C      IF(RESTRT) GO TO 10
79      C
80      C      RUN IS A NEW PROBLEM, GO GENERATE IT,
81      C
82      C      CALL YASET
83      C      GO TO 20
84      C
85      C      RUN IS A PROBLEM RESTART, READ THE DUMP TAPE,
86      C
87      C      10 CONTINUE
88      C      CALL YARSRT
89      C
90      C      GET RID OF THE DUMP FILE
91      C
92      C      CALL CLOSIT(7)
93      C      20 CONTINUE
94      C
95      C      EXECUTE THE MAIN CODE
96      C
97      C      CALL YAQUI2
98      C
99      C      RUN TERMINATION, EXIT,
100     C
101     C      CALL UNCLE(2,SHYQUI,15,15HRUN TERMINATION)
102     C
103     C      30 FORMAT(1H ,29HBEGIN CODE EXECUTION AT CP # ,F10.4)
104     C      END

```

```

1      C      SUBROUTINE AIR
2      C
3      C      SEMI-PHYSICAL FIT TO THE EQUATION OF STATE OF AIR

```

```

4 C DENSITIES FROM 10**2 TO 10**(-7) NORMAL DENSITY
5 C PRESSURE=(GAMMA-1.)*RHO*E, WHERE GAMMA IS A FUNCTION OF
6 C DENSITY AND ENERGY
7 C RHO=MATERIAL DENSITY
8 C EJLN=MATERIAL SPECIFIC INTERNAL ENERGY
9 C GMONE=GAMMA=1.
10 C CONCJ=RELATIVE CONCENTRATION, SEE NOTE BELOW.
11 C
12 C ALL UNITS ARE CGS FOR INPUT QUANTITIES
13 C
14 C ORIGINALLY OBTAINED FROM THE AIR FORCE WEAPONS LAB
15 C MODIFIED BY J.L.NORTON, LASL T=3, 1974
16 C
17 C COMMON/EQNBT/RHO,EJLN,GMONE,CONCJ
18 C
19 C IN THIS VERSION OF THE CODE, TWO EQNS OF STATE ARE BEING USED, AIR
20 C AND METHANE. THE FINAL VALUE OF GMONE IS DETERMINED BY
21 C THE RELATION GMONE(FINAL)=CONCJ*GMONE(METHANE)
22 C + (1.-CONCJ)*GMONE(AIR).
23 C
24 C THUS, CONCJ=1, YIELDS A METHANE GAMMA AND CONCJ=0., AIR,
25 C ANYTHING IN BETWEEN USES A LINEAR COMBINATION OF THE TWO.
26 C
27 C CHECK TO MAKE SURE CONCJ DOES NOT EXCEED 1, IF IT DOES, SET IT TO 1
28 C
29 C IF( CONCJ.GT.1.) CONCJ=1.
30 C
31 C IFLAG=1 SIGNALS AIR TO DO THE GAMMA LINEAR COMBINATION, IF
32 C CONCJ=0., THIS IS UNNECESSARY.
33 C
34 C IFLAG=0
35 C
36 C IF CONCJ=0., SKIP THE METHANE CALCULATION ENTIRELY
37 C
38 C IF( CONCJ.EQ.0.) GO TO 10
39 C
40 C GO GET THE METHANE GAMMA
41 C
42 C CALL MTHANE
43 C
44 C IF CONCJ=1., WE ARE ALL DONE
45 C
46 C IF( CONCJ.EQ.1.) RETURN
47 C
48 C TURN ON THE LINEAR COMBINATION FLAG, SAVE THE METHANE GMONE, AND
49 C CONTINUE ON WITH THE AIR EOS CALCULATION
50 C
51 C IFLAG=1
52 C GMME=GMONE
53 C
54 C CALCULATE GAMMA=1. FOR AIR
55 C
56 C 10 CONTINUE
57 C
58 C RHOZ IS THE NORMAL AIR DENSITY
59 C
60 C RHOZ=1.293E-3

```

```

61      C      RHOT IS THE COMPRESSION
62      C
63      C      RHOT=RHO/RHOZ
64
65      C      MAKE E POSITIVE AND CONVERT TO SI UNITS (TJ/MG)
66      C
67      C      E=1, E=10+ABS(EJLN)
68
69      C      THE ENERGY AT WHICH OXYGEN AND NITROGEN DISSOCIATE IS A
70      C      FUNCTION OF DENSITY
71
72      C
73      C      E1=(8.5*E)/.975
74
75      C      THE FERMI-DIRAC FUNCTON IS ONLY COMPUTED WITHIN 5.*DELTA E OF
76      C      EACH TRANSITION, OTHERWISE IT IS ZERO OR ONE.
77
78      C      IF(ABS(E1)=5.)50,20,20
79      20 IF(E1) 40,40,30
80      30 F0=EXP(-E/4.46)
81      F0=N0,
82      W8=1,
83      GO TO 60
84      40 F0=N0,
85      F0=EXP(-E/6.63)
86      W8=0,
87      GO TO 60
88      50 DE1=.975*RHOT**,.05
89      EE1=8.5+.357* ALOG10(RHOT)
90      E1=(EE1-E)/DE1
91      W8=1./EXP(-E1)+1.,
92      F0=EXP(-E/4.46)*W8
93      F0=EXP(-E/6.63)*(1.-W8)
94
95      C      THE DENSITY DEPENDENCE ONLY OCCURS ABOVE E=1., AND IT IS OF
96      C      THE FORM (RHO/RHOZ**((CONSTANT*LOG(E)))). THE CONSTANT
97      C      MAKES A TRANSITION FROM .048 TO .029 AS THE OXYGEN
98      C      DISSOCIATES AND THE DENSITY SPREAD BECOMES CONSTANT BEYOND
99      C      THE SECOND PEAK.
100
101      60 IF(E=1,)70,70,80
102      70 BETAN=0,
103      GO TO 90
104      80 BETAN=.048*W8+.032*(1.-W8)*ALOG10(E)
105      90 E2=(E=40.)/3,
106      100 IF(ABS(E2)=5.)130,100,100
107      100 IF(E2) 110,110,120
108      110 FN=0,
109      W8=0,
110      GO TO 140
111      120 FN=EXP(-E/25.5)
112      W8=1,
113      GO TO 140
114      130 DE2=44.*RHOT**,.085
115      EE2=45.*RHOT**,.0157
116      E2=(E=EE2)/DE2
117      W8=1./EXP(-E2)+1.,

```

```

118      FN=EXP(-E/25.5)*WS
119      140 E3=(E-160.)/6,
120      BETA=BETAA*(1.-WS)+.045*WS
121      IF(E3+S.)150,160,160
122      150 FE=0,
123      GO TO 170
124      160 FE=1./EXP(-E3)+1.
125      170 RHOFACTOR**BETA
126      GMONE=(,161+,255*FO+,280*FON+,137*FN+,050*FE)*RHOFACTOR
127      C
128      C      IF IFLAG IS ZERO, ONLY AIR IS DESIRED AND WE ARE ALL DONE
129      C
130      C      IF(IFLAG.EQ.0) RETURN
131      C
132      C      CALCULATE THE GAMMA=1, LINEAR COMBINATION
133      C
134      C      GMONE=CONCJ*GMM+(1.-CONCJ)*GMONE
135      C      RETURN
136      C      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,VUTIL
7      C          * 2 = UL,VL
8      C          * 3 = UP,VP
9
10     C
11     C      WRITTEN BY J.L.NORTON,LASL T-3,1975
12     *
13     *      ***** BEGIN COMDECK YSTORE      *****
14     *      ***** BEGIN COMDECK YAQDIM      *****
15     *      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC(
16     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC80(1),E(1),ETIL(1),RVOL
17     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
18     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DL8ROI(1),DE8ROQ(1),CAPGAM(1),TUQ
19     4 (1),8IG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUGVEC(1),MTIL(1),
20     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GR8V(1),GZSV(1),X13K(1),X24K(1),
21     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
22     *
23     *      ***** END COMDECK YAQDIM      *****
24     *      ***** BEGIN COMDECK YAQBC      *****
25     C      LOGICAL RESTRT,FILM,PAPER,TURB
26     C      REAL LAM,MU
27     C      COMMON/YSC1/AA8C(N8CP1)
28     C      COMMON/YSC1/AA8C(9600)
29     C      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
30     1 DTO(10),DTOC(10),DTO2,DTOB,DTPO8,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
31     2 FIPYB,FIPYT,FIYB,FIYL,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
32     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISCC2,ISCC3,ITV,
33     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR

```

```

32      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFD2,KXI,LAM,LPB,MU,NAME(8),
33      1 NCYC,NLC,NPS,NPT,NQ,NDI,NGIB,NGI2,N8C,NUMIT,ZORIG,OM,OMCYL,PXCONV
34      2 ,PXL,PXR,PYB,PYCONV,PYT,ROT,REZRON,REZZIE,REZY0,RIBAR,RIBJB,
35      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36      COMMON/YSC2/TUQI,TUSI,NCG,TNEG,TNEG8V,TUSV,TURB,PTOP,PRITE,PBOTM,
37      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,8AV1,QLEVEL,TQ,IST,
38      2 VV,XCONV,XL,XR,YB,YCNY,YT,PTPOLD,DT8V,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRO,FXLQ,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVS8V,DTC8AV,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASS8V,WMAXEF,RMINEF,TSTRTD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTRT,PILM,PAPER,IPD,IFD
47      *     END COMDECK YAQSC      *****
48      *     BEGIN COMDECK YAQE0      *****
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RC8Q,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),8IE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUG),(AASC(19),SIG),(AASC(20),TUB),(AASC(
56      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DL8ROI,Y13K),(AASC(24),GZ8V
57      7 ),(AASC(25),DL8R0Q,VG),(AASC(26),GR8V),(AASC(27),GRROP,TU8VEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVX8V,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      *     END COMDECK YAQE0      *****
63      *     END COMDECK YSTORE      *****
64      C SET UP THE LOOP OVER CELLS
65      C
66      CALL START
67      C
68      C SET THE BOTTOM BOUNDARY VELOCITIES
69      C
70      IJ8V=IJ
71      DO 40 I=1,IP1
72      GO TO (10,20,30),IFLAG
73      10 CONTINUE
74      VTIL(IJ)=0.
75      GO TO 40
76      20 CONTINUE
77      VL(IJ)=0.
78      GO TO 40
79      30 CONTINUE
80      VP(IJ)=0.
81      40 IJ=IJ+NQ
82      IJ=IJ8V
83      C
84      C SET THE LEFT AND RIGHT BOUNDARIES
85      C
86      DO 90 J=2,JP1
87      GO TO (50,60,70),IFLAG
88      50 CONTINUE

```

```

89      UTIL(IJ)=0,
90      UTIL(IJ+NQIB)=0,
91      GO TO 80
92 60 CONTINUE
93      UL(IJ)=0,
94      UL(IJ+NQIB)=0,
95      GO TO 80
96 70 CONTINUE
97      UP(IJ)=0,
98      UP(IJ+NQIB)=0,
99 80 CONTINUE
100     CALL LOOP
101 90 CONTINUE
102 C
103 C      SET THE TOP BOUNDARY
104 C
105     DO 130 I=1,IP1
106     GO TO (100,110,120),IFLAG
107 100 CONTINUE
108      VTIL(IJ)=0.
109      GO TO 130
110 110 CONTINUE
111      VL(IJ)=0.
112      GO TO 130
113 120 CONTINUE
114      VP(IJ)=0.
115 130 IJ=IJ+NQ
116 C
117 C      SET THE UPPER RIGHHAND CORNER U
118 C
119      IJ=IJ+NQ
120      GO TO (140,150,160),IFLAG
121 140 CONTINUE
122      UTIL(IJ)=0.
123      GO TO 170
124 150 CONTINUE
125      UL(IJ)=0.
126      GO TO 170
127 160 CONTINUE
128      UP(IJ)=0.
129 170 CONTINUE
130      CALL DONE
131      RETURN
132 END

```

```

1      SUBROUTINE CINIT
2      C
3      C      ROUTINE TO CALCULATE QUANTITIES FOR THE CYCLE PRINT AND
4      C      INITIALIZE FOR THE NEXT CYCLE
5      C
6      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3

```

```

7   C MODIFIED AND DOCUMENTED BY J.L.NORTON, LASL T=3, 1978
8   C
9   *
10  *      ---- BEGIN COMDECK PARAM      -----
11  COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
12  NLCPS,NLCP4,IFLMSZ
13  *
14  *      ---- END COMDECK PARAM      -----
15  *      ---- BEGIN COMDECK YSTORE      -----
16  DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC
17  1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC5Q(1),E(1),ETIL(1),RVOL
18  2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
19  3 ,VL(1),ROL(1),AVXSV(1),AVY8V(1),DLSROI(1),DL8R0Q(1),CAPGAM(1),TUQ
20  4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
21  5 CONC(1),CTEMP(1),ANCU(1),ANGV(1),GR8V(1),GZ8V(1),X13K(1),X24K(1),
22  6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
23  *
24  *      ---- END COMDECK YAGDIM      -----
25  *      ---- BEGIN COMDECK YAGSC      -----
26  LOGICAL RESTRT,FILM,PAPER,TURB
27  REAL LAM,MU
28  C COMMON/Y3C1/AASC(NSCP1)
29  COMMON/Y3C1/AASC(9600)
30  COMMON/Y3C2/AA(1),ANC,AB,ABFAC,ABM,B0,COLAMU,CYL,DR,DT,DTG,DTFAC,
31  1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EP5,FIPXL,FIPXR,
32  2 FIPYB,FIPYT,FXL,FXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
33  3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
34  4 IUNP,IXL,IXR,IYB,IYT,J,JBAR
35  COMMON/Y3C2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
36  1 NCYC,NLC,NPB,NPT,NQ,NQ1,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
37  2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
38  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
39  COMMON/Y3C2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TU8V,TURB,PTOP,PRITE,PBOTM,
40  1 ILNG,NILNG,TP3,TUPOT,TQGSBV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,T0,IST,
41  2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DT8V,DTLAST,FIYBO,IYBO,YCNVLD,
42  3 XCNVLD,FIXR0,FXL0,IXR0,IXLO,ISVH,JSVH,GMN,GMX,WMAX,JNM,T2,TLIM,
43  4 ROMFXR,ROMFYT,ROMFYB,JUMP,TWTHR,DTR,THMASS,DTVS,DTCSA,DTCSA,IDTV
44  5 ,JDTV,JDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOD,TMAX,TGMX,ITM,JTM,ITG,JTG
45  6 ,THMASSV,WMAXEF,RMINEF,T8TRTD
46  C COMMON/Y3C2/ZZ
47  COMMON/Y3C4/ITAB(ITABP)
48  COMMON/Y3C4/ITAB(1000)
49  *
50  *      ---- END COMDECK YAGSC      -----
51  *      ---- BEGIN COMDECK YAGEQ      -----
52  EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
53  1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RC5Q,MP),(AASC
54  1 (8),E,ETIL,AREA,XR13K),
55  2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
56  3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
57  4 ,UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
58  5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
59  6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DL8R0I,Y13K),(AASC(24),GZSV
60  7 ),(AASC(25),DL8R0Q,VG),(AASC(26),GR8V),(AASC(27),GRROP,TUQVEC,
61  8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
62  9 AASC(31),ANCU),(AASC(32),ANGV),(AASC(33),AVXSV,X13K),(AASC(34),
63  1 AVY8V,X24K)
64  REAL M,MP,MPAR,MTIL
65  *      ---- END COMDECK YAGEQ      -----

```

```

64      *      ===== END COMDECK YSTORE =====
65      C      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
66      C
67      C      INITIALIZE LOOP VARIABLES
68      C
69      POTE=0,
70      TK=0,
71      TI=0,
72      UMOM=0,
73      VMOM=0,
74      TMASSV=TMASS
75      TMASS=0,
76      TUGENG=0,
77      CIRC=0,
78      TMAX=0,
79      TGMX=0,
80      ITM=0
81      JTG=0
82      JTM=0
83      JTG=0
84      C
85      C      LOOP OVER ALL REAL ZONES^IN THE MESH
86      C
87      CALL START
88      DO 60 J=2,JP1
89      DO 50 I=1,IBAR
90      IPJ=IJ+NQ
91      IPJP=IJP+NQ
92      C
93      C      TMAX IS THE MAXIMUM SIE OF ANY CELL IN THE MESH, IT IS FOUND
94      C      IN CELL ITM,JTM.
95      C
96      IF(SIE(IJ).LE.TMAX) GO TO 10
97      ITM=I
98      JTM=J
99      TMAX=SIE(IJ)
100     C
101     C      TGMX IS THE MAXIMUM SIE GRADIENT AND (ITG,JTG) IS THE CELL
102     C      IN WHICH IT OCCURS
103     C
104     10 CONTINUE
105     SAVA=(X(IPJ)-X(IJ))**2+(Y(IPJ)-Y(IJ))**2
106     SAVB=(X(IJP)-X(IJ))**2+(Y(IJP)-Y(IJ))**2
107     SAVA=ABS(SIE(IJ)-SIE(IPJ))/SQRT(SAVA)
108     SAVB=ABS(SIE(IJ)-SIE(IJP))/SQRT(SAVB)
109     SAV=AMAX1(SAVA,SAVB)
110     IF(SAV.LT.TGmx) GO TO 80
111     ITG=I
112     JTG=J
113     TGmx=SAV
114     20 CONTINUE
115     C
116     C      CALCULATE THE CIRCULATION AROUND THE PROBLEM BOUNDARIES
117     C
118     IF(I.EQ.1) CIRC=CIRC+0.5*(V(IJ)+V(IJP))*(Y(IJP)-Y(IJ))
119     IF(I.EQ.1M1) CIRC=CIRC+0.5*(V(IJ)+V(IJP))*(Y(IJP)-Y(IJ))
120     IF(J.EQ.3) CIRC=CIRC+0.5*(U(IJ)+U(IPJ))*(X(IPJ)-X(IJ))

```

```

121 IF(J.EQ.JBAR) CIRC=CIRC+0.5*(U(IJ)+U(IPJ))*(X(IPJ)=X(IJ))
122 C
123 C XMSENG IS THE MASS IN CELL I,J
124 C TMASS IS THE TOTAL PROBLEM MASS
125 C
126 C XMSENG=RO(IJ)/RVOL(IJ)
127 C TMASS=TMASS+XMSENG
128 C
129 C SPENGK IS THE SPECIFIC KINETIC ENERGY OF CELL I,J
130 C
131 C SPENGK=0.125*(U(IPJ)**2+U(IPJP)**2+U(IJP)**2+U(IJ)**2+V(IPJ)**2+V(
132 C 1 IPJP)**2+V(IJP)**2+V(IJ)**2)
133 C
134 C TK IS THE TOTAL KINETIC ENERGY IN THE PROBLEM
135 C
136 C TK=TK+SPENGK*XMSENG
137 C
138 C TI IS THE TOTAL INTERNAL ENERGY IN THE PROBLEM
139 C
140 C TI=TI+XMSENG*SIE(IJ)
141 C
142 C UMOM AND VMOM ARE THE RADIAL AND AXIAL TOTAL MOMENTA,
143 C RESPECTIVELY
144 C
145 C UMOM=UMOM+0.25*XMSENG*(U(IPJ)+U(IPJP)+U(IJP)+U(IJ))
146 C VMOM=VMOM+0.25*XMSENG*(V(IPJ)+V(IPJP)+V(IJP)+V(IJ))
147 C
148 C POTE IS THE TOTAL GRAVITATIONAL POTENTIAL ENERGY (WITH G=1.).
149 C NOTE THAT VERTEX Y#S AND MASSES ARE USED.
150 C
151 C POTE=POTE+Y(IJ)/RM(IJ)
152 C
153 C ADD THE J=JP2 VERTEX INTO THE POTENTIAL ENERGY
154 C
155 C IF(J.EQ.JP1) POTE=POTE+V(IJP)/RM(IJP)
156 C IF(TURB) TUGENG=TUGENG+TUG(IJ)*XMSENG
157 C
158 C NEW PRESSURE MUST BE CALCULATED, P ARRAY CURRENTLY HOLDS
159 C VELOCITIES DUE TO EQUIVALENCING OF P WITH UP.
160 C
161 C PUT A LOWER BOUND OF ZERO ON THE SIE
162 C
163 C SIE=MAX1(SIE(IJ),0.)
164 C
165 C GO GET THE CELL GAMMA=1
166 C
167 C ROTMP=RO(IJ)
168 C ETMP=SIE
169 C CONCJ=CONC(IJ)
170 C CALL AIR
171 C GM1=GMONE
172 C
173 C COMPUTE PRESSURE
174 C
175 C P(IJ)=GM1*ROTMP*SIE
176 C
177 C SET PRESSURE OF BOTTOM FICTITIOUS ROW USING GAMMA OF CELLS ABOVE

```

```

178 C IF(J.EQ.2) P(IJM)=GM1*RO(IJM)*SIE(IJM)
179 C SET PRESSURE OF TOP FICTITIOUS ROW USING GAMMA OF CELLS BELOW
180 C
181 C IF(J.EQ.JP1) P(IJP)=GM1*RO(IJP)*SIE(IJP)
182 C
183 C SEE IF WE ARE PROCESSING THE RIGHTMOST REAL COLUMN
184 C
185 C IF(I,NE,IBAR) GO TO 40
186 C
187 C YES,
188 C
189 C SET PRESSURE OF RIGHT FICTITIOUS COLUMN USING GAMMA OF CELLS
190 C TO THE LEFT
191 C
192 C P(IPJ)=GM1*RO(IPJ)*SIE(IPJ)
193 C
194 C SEE IF WE ARE PROCESSING THE BOTTOM REAL ROW
195 C
196 C IF(J,NE,2) GO TO 30
197 C
198 C YES,
199 C
200 C SET PRESSURE OF BOTTOM CELL IN RIGHT FICTITIOUS COLUMN USING
201 C GAMMA OF CELL (IBAR,2)
202 C
203 C IPJM=IJM+NQ
204 C P(IPJM)=GM1*RO(IPJM)*SIE(IPJM)
205 C GO TO 40
206 C 30 CONTINUE
207 C
208 C NO. SEE IF WE ARE PROCESSING THE TOP REAL ROW.
209 C
210 C IF(J,NE,JP1) GO TO 40
211 C
212 C YES,
213 C
214 C SET PRESSURE OF UPPER CELL IN RIGHT FICTITIOUS COLUMN USING
215 C GAMMA OF CELL (IBAR,JP1)
216 C
217 C P(IPJP)=GM1*RO(IPJP)*SIE(IPJP)
218 C
219 C 40 CONTINUE
220 C IJ=IPJ
221 C IJP=IJP+NQ
222 C IJM=IJM+NQ
223 C 50 CONTINUE
224 C
225 C ADD THE I=IP1 VERTEX INTO THE POTENTIAL ENERGY
226 C POTE=POTE+Y(IJ)/RM(IJ)
227 C
228 C ADD THE (IP1,JP2) VERTEX INTO THE POTENTIAL ENERGY
229 C
230 C IF(J.EQ.JP1) POTE=POTE+Y(IJP)/RM(IJP)
231 C CALL LOOP
232 C
233 C 60 CONTINUE
234 C

```

```
235      CALL DONE
236      RETURN
237      END
```

```
1      SUBROUTINE CONTOUR(ILOG,ITITLE,NWT)
2      C
3      C      ROUTINE FOR DOING CONTOUR PLOTS IN YAQUI
4      C
5      C      ILOG IS 0 IF LINEAR CONTOUR INCREMENTS ARE TO BE USED
6      C      ILOG IS 1 FOR LOGARITHMIC CONTOUR INCREMENTS
7      C      ITITLE IS THE HOLLERITH TITLE TO BE USED ON THE PLOT
8      C      NWT IS THE NO. OF COMPUTER WORDS IN THE TITLE
9
10     C      THE QUANTITY TO BE PLOTTED IS FOUND IN CQ
11     C
12     C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
13     C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1975
14     C
15     *      ----- BEGIN COMDECK PARAM -----
16     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCPI2,
17     1 NLCPI3,NLCPI4,IFLM8Z
18     *      ----- END COMDECK PARAM -----
19     *      ----- BEGIN COMDECK YSTORE -----
20     *      ----- BEGIN COMDECK YAQDIM -----
21     DIMENSION X(),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
22     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC8Q(1),E(1),ETIL(1),RVOL
23     2 (1),M(1),RM(1),VP(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
24     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DL8R0Q(1),CAPGAM(1),TUQ
25     4 (1),SIG(1),TU$(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
26     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
27     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
28     *      ----- END COMDECK YAQDIM -----
29     *      ----- BEGIN COMDECK YAQSC -----
30     LOGICAL RESTRT,FILM,PAPER,TURB
31     REAL LAM,MU
32     C      COMMON/YSC1/AASC(NSCP1)
33     COMMON/YSC1/AA8C(9600)
34     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
35     1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EP8,FIPXL,FIPXR,
36     2 FIPYB,FIPYT,FIYL,FIYR,FIYB,FIYT,FREZXR,GR,GRVEL,GZ,GZP,I,IBAR,
37     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
38     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
39     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
40     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NBC,NUMIT,ZORIG,OM,OMCYL,PXCONV
41     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
42     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWPIN
43     COMMON/YSC2/TUQI,TUSI,NCG,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
44     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
45     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DT8V,DTLAST,FIYB0,IYB0,YCNVLD,
46     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,J8VW,QMN,GMX,WMAX,JNM,T2,TLIM,
47     4 ROMFXR,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMA88,DTVSAY,DTCSAV,IDTV
```

```

48      5 ,JDTV, IDTC, JDTG, CIRC, TIS, POTE, UMOM, VMOM, TMAX, TGDX, ITM, JTG, ITG, JTG
49      6 ,TMASSV, WMAXEF, RMINEF, TSTRTD
50      COMMON/YSC2/ZZ
51      C      COMMON/YSC4/ITAB(ITABP)
52      COMMON/YSC4/ITAB(1000)
53      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
54      *      **** END COMDECK YAGSC ****
55      *      **** BEGIN COMDECK YAGEQ ****
56      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
57      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
58      1 (8),E,ETIL,AREA,XR13K),
59      2 (AASC(15),SIE),(AASC(16),PM0,DKL8M,RMP),(AASC(9
60      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
61      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
62      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
63      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DL8ROI,Y13K),(AASC(24),GZSV
64      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
65      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
66      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVX8V,X13K),(AASC(34),
67      1 AVYSV,X24K)
68      REAL M,MP,MPAR,MTIL
69      *      **** END COMDECK YAGEQ ****
70      *      **** END COMDECK YSTORE ****
71      *      **** BEGIN COMDECK ASTORE ****
72      COMMON/ASTC/AT(100),FT(100)
73      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
74      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
75      1 ),(AT(9),YCO),(FT,CON)
76      *      **** END COMDECK ASTORE ****
77      *      **** BEGIN COMDECK PCALL ****
78      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
79      *      **** END COMDECK PCALL ****
80      DIMENSION BCD(2),ITITLE(1)
81      DIMENSION ILABEL(26),IPLBL(26)
82      DATA ILABEL/1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK,1HL,1HM,
83      1 1HN,1HO,1HP,1HQ,1HR,1HS,1HT,1HU,1HV,1HW,1HX,1HY,1HZ/
84      DATA IPLBL/17,18,19,20,21,22,23,24,25,33,34,35,36,37,38,39,40,41,
85      1 50,51,52,53,54,55,56,57/
86      DATA BCD/1H /
87      C      FIND THE MINIMUM AND MAXIMUM (QMN AND QMX) OF THE QUANTITY
88      C      TO BE PLOTTED
89      C
90      C      QMN#1,E30
91      C      QMX#-QMN
92      C      CALL START
93      C      DO 20 J=2,JP1
94      C      DO 10 I=1,IBAR
95      C      QMN#AMINI(C0(IJ),QMN)
96      C      QMX#AMAXI(C0(IJ),QMX)
97      10 IJ=IJ+NQ
98      C      CALL LOOP
99      20 CONTINUE
100     C
101     C      SET THE CONTOUR INCREMENT. THE BASIC ALGORITHM IS BASED ON
102     C      THE DESIRE TO HAVE THE CONTOUR INCREMENT AN INTEGRAL POWER
103     C      OF TWO AND THE CONTOURS THEMSELVES INTEGRAL MULTIPLES OF
104     C

```

```

105      C      THIS POWER OF TWO, THESE FACTS INSURE THAT TWO PLOTS
106      C      VERY NEARLY THE SAME WILL HAVE EXACTLY THE SAME CONTOUR
107      C      VALUES. FIRST MAKE AN INITIAL GUESS FOR THE CONTOUR
108      C      INCREMENT (DQ) SO THAT THERE ARE AT LEAST TEN CONTOURS,
109      C
110      C      DQ=(QMX-QMN)/10,
111      C
112      C      IF THE INCREMENT IS ZERO OR NEGATIVE, BYPASS THE PLOT
113      C
114      C      IF(DQ.LE.0.) RETURN
115      C
116      C      FIND A POWER OF TWO JUST LESS THAN DQ BUT AT LEAST 2**=33
117      C      (WHICH IS ABOUT 10**=10)
118      C
119      C      TEST=2,**(-32)
120      30 CONTINUE
121      C      IF(TEST.GE.DQ) GO TO 40
122      C      TEST=2.*TEST
123      C      GO TO 30
124      40 CONTINUE
125      C
126      C      SET THE CONTOUR INCREMENT TO IT
127      C
128      C      DQ=.5*TEST
129      C
130      C      NOW DETERMINE THE CONTOUR VALUES. K WILL BE THE FINAL NO.
131      C      OF CONTOURS. THE FIRST CONTOUR IS THE NEAREST INTEGRAL
132      C      MULTIPLE OF DQ LESS THAN QMN.
133      C
134      50 CONTINUE
135      C      ITEST=QMN/DQ
136      C      IF(ITEST.LT.0) ITEST=ITEST+1
137      C      CON(1)=FLOAT(ITEST)*DQ
138      C      DO 60 K=2,27
139      C      IF(CDN(K-1).GT.QMX) GO TO 70
140      C      IF(K.EQ.27) GO TO 60
141      C      CON(K)=CON(K-1)+DQ
142      60 CONTINUE
143      C
144      C      TOO MANY CONTOURS. GO INCREASE DQ AND TRY AGAIN.
145      C
146      C      GO TO 80
147      70 CONTINUE
148      C      K=K+1
149      C
150      C      IF THERE ARE MORE THAN 20 CONTOURS, INCREASE DQ AND GO CALCULATE
151      C      THE CONTOURS AGAIN.
152      C
153      C      IF(K.LE.20) GO TO 90
154      80 CONTINUE
155      C      DQ=2.*DQ
156      C      GO TO 50
157      90 CONTINUE
158      C
159      C      ADVANCE THE FILM TO THE NEXT FRAME
160      C
161      C      CALL ADV(1)

```

```

162 C      SET THE HEIGHT OF THE FIRST LINE OF THE CONTOUR LABELS
163 C
164 C
165 C      CALL LINCNT(6)
166 C
167 C      LIST THE CONTOURS ON THE PLOT
168 C
169 C      DO 100 KK=1,K
170 C      WRITE(IPFD,250) ILABEL(KK),CON(KK)
171 C      100 CONTINUE
172 C      WRITE(IPFD,260) QMN,QMX
173 C
174 C      LABEL THE TYPE OF PLOT
175 C
176 C      CALL LINCNT(58)
177 C      WRITE(IPFD,270)(ITITLE(I),I=1,NWT)
178 C      WRITE(IPFD,290) JNM,NAME,T,NCYC
179 C
180 C      PUT CONTOUR INFORMATION OUT TO CYCLE SUMMARY
181 C
182 C      IPDX=6
183 C      WRITE(IPDX,280)(ITITLE(I),I=1,NWT)
184 C      WRITE(IPDX,300) QMN,QMX,CON(1),CON(K=1),DQ
185 C
186 C      LOOP TO DO THE ACTUAL PLOT
187 C
188 C      CALL START PULLS IN ROWS 1,2,AND 3 AND LEAVES IJ POINTING AT
189 C      ROW 2
190 C
191 C      CALL START
192 C      DO 240 J=2,JBAR
193 C
194 C      CALL LOOP AT THIS POINT SHIFTS THE ROWS IN SCM UP ONE SO THAT
195 C      IJ POINTS AT ROW J+1 INSTEAD OF ROW J
196 C
197 C      CALL LOOP
198 C      DO 230 I=1,IM1
199 C      IPJ=IJ+NO
200 C      IPJM=IJM+NO
201 C
202 C      N#0 SIGNIFIES THAT XCO AND YCO HAVE NOT BEEN COMPUTED FOR THIS
203 C      I AND J
204 C
205 C      N#0
206 C
207 C      LOOP OVER ALL OF THE CONTOUR VALUES
208 C
209 C      DO 220 KK=1,K
210 C
211 C      SET FLAGS.
212 C
213 C      CONSIDER THE FOUR CELLS (I,J),(I+1,J),(I,J+1), AND (I+1,J+1)
214 C      AND LET THEM BE DENOTED CELLS 1,2,3,AND 4,RESPECTIVELY,
215 C      THEN,KN IS ZERO IF THE QUANTITY TO BE PLOTTED IN CELL N
216 C      ISGREATER THAN THE CURRENT CONTOUR VALUE, AND 1 IF
217 C      THE QUANTITY TO BE PLOTTED IS LESS THAN OR EQUAL TO
218 C      THE CURRENT CONTOUR VALUE.

```

```

219      C
220      C      NOTE THAT WHAT IS BEING CONSIDERED IS THE QUADRILATERAL FORMED BY
221      C      CONNECTING THE CENTERS OF THE FOUR CELLS IN THE ORDER
222      C      1=2=4=3=1. LET THIS BE KNOWN AS THE CURRENT CONTOUR AREA.
223      C
224      K1#0
225      K2#0
226      K3#0
227      K4#0
228      IF(CQ(IJM),LE,CON(KK)) K1=1
229      IF(CQ(IPJM),LE,CON(KK)) K2=1
230      IF(CQ(IJ),LE,CON(KK)) K3=1
231      IF(CQ(IPJ),LE,CON(KK)) K4=1
232      C
233      C      IF ALL THE FLAGS ARE 1 OR 0, THEN THE CURRENT CONTOUR IS EITHER
234      C      GREATER THAN OR LESS THAN OR EQUAL TO THE PLOT QUANTITIES
235      C      AT ALL FOUR CORNERS OF THE CURRENT CONTOUR AREA. THUS, THE
236      C      CURRENT CONTOUR DOES NOT CROSS THE CURRENT CONTOUR AREA,
237      C      GO ON TO THE NEXT CONTOUR,
238      C
239      C      IF(K1*K2*K3*K4,NE,0,OR,K1+K2+K3+K4,EQ,0) GO TO 220
240      C
241      C      CALCULATE THE CENTERS OF CELLS 1,2,3, AND 4 (THE VERTICES OF THE
242      C      CURRENT CONTOUR AREA) IF THIS HAS NOT ALREADY BEEN DONE.
243      C
244      C      IF(N,GT,0) GO TO 130
245      IJB=IJM
246      IJA=IJ
247      DO 120 JJ=1,2
248      DO 110 II=1,2
249      IPJB=IJB+NQ
250      IPJA=IJA+NQ
251      N=N+1
252      XCO(N)=.25*(X(IPJB)+X(IPJA)+X(IJA)+X(IJB))
253      YCO(N)=.25*(Y(IPJB)+Y(IPJA)+Y(IJA)+Y(IJB))
254      IJA=IPJA
255      110 IJB=IPJB
256      IJB=IJ
257      120 IJA=IJP
258      130 CONTINUE
259      C
260      C      DETERMINE WHICH SIDES OF THE CURRENT CONTOUR AREA THE CURRENT
261      C      CONTOUR CROSSES.
262      C      LL COUNTS HOW MANY SIDES THAT HAVE BEEN FOUND TO BE CROSSED
263      C
264      LL#0
265      C
266      C      SEE IF THE LEFT SIDE IS CROSSED
267      C
268      C      IF(K1+K3,NE,1) GO TO 140
269      C
270      C      YES, IC1 AND IC2 ARE THE VERTEX NUMBERS OF THE CURRENT CONTOUR
271      C      AREA THAT BOUND THE SIDE CROSSED, IJ1 AND IJ2 ARE THE INDICES
272      C      OF THE MESH CELLS CONTAINING THESE VERTICES,
273      C
274      IC1=1
275      IC2#3

```

```

276      IJ1=IJM
277      IJ2=IJ
278      C
279      C      GO SET UP PLOT COORDINATES FOR THIS CROSSING
280      C
281      C      GO TO 170
282      C
283      C      SEE IF THE BOTTOM IS CROSSSED 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 CROSSSED
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 CROSSSED
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

```

```

333      IF(IX1(LL).LT.IXL.OR.(IX1(LL).GT.IXR)) GO TO 210
334      C
335      C      IF ONLY ONE INTERSECTION HAS BEEN LOCATED, GO BACK AND LOOK
336      C      FOR ANOTHER
337      C
338      C      IF(LL.GE.2) GO TO 200
339      180 CONTINUE
340      C
341      C      IF IC1=2, RIGHT SIDE WAS INTERSECTED LAST, GO BACK AND GET
342      C      THE TOP SIDE. IF IC1=3, TOP SIDE WAS INTERSECTED LAST,
343      C      ALL SIDES HAVE BEEN CHECKED, GO ON TO NEXT CONTOUR,
344      C
345      C      GO TO (190,160,220),IC1
346      C
347      C      IC1=1, EITHER LEFT OR BOTTOM WAS LAST INTERSECTED. IF BOTTOM
348      C      (IC2=2), GO BACK AND CHECK THE RIGHT. IF LEFT (IC2=3),
349      C      GO BACK AND CHECK THE BOTTOM.
350      C
351      190 CONTINUE
352      IF(IC2.EQ.2) GO TO 150
353      GO TO 140
354      200 CONTINUE
355      C
356      C      TWO SIDES HAVE BEEN CROSSED, CONNECT THE POINTS OF
357      C      INTERSECTION.
358      C
359      C      CALL DRV(IX1,IY1,IX2,IY2)
360      C
361      C      PLOT THE LABEL ON THE FIRST INTERSECTION POINT
362      C
363      C      CALL PLT(IX1,IY1,IPLBL(KK))
364      210 CONTINUE
365      C
366      C      START ON A NEW INTERSECTION PAIR IF BOTTOM WAS LAST SIDE
367      C      CHECKED, OTHERWISE, THERE ARE NOT TWO POSSIBLE INTERSECTIONS
368      C      LEFT SO WE ARE DONE.
369      C
370      LL=0
371      IF(IJ2.EQ.IPJM) GO TO 180
372      220 CONTINUE
373      IJM=IPJM
374      IJ=IPJ
375      230 IJP=IJP+NQ
376      240 CONTINUE
377      C
378      C      DRAW THE PLOT FRAME AND LABEL IT
379      C
380      C      CALL TICBOX
381      RETURN
382      C
383      250 FORMAT(100XA1,2X1PE10.3)
384      260 FORMAT(98X3HGMN,2X1PE10.3/98X3HQMX,2XE10.3)
385      270 FORMAT(1H ,8A10)
386      280 FORMAT(1H ,16HCONTOUR PLOT OF ,6A10)
387      290 FORMAT(1H ,4XA10,8A10,3X2HT#,1PE12.3,1X6HCYCLES,1S)
388      300 FORMAT(5H MIN=1PE12.5,5H MAX=E12.5,3H L=E12.5,3H H=E12.5,/1H ,16X
389      1 4H DQ=E12.5)

```

```

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,NLCPI,NLCP2,
9      1 NLCP3,NLCP4,IFLMSZ
10     *      ===== END COMDECK PARAM =====
11     *      ===== BEGIN COMDECK YAGSC =====
12     LOGICAL RESTRT,FILM,PAPER,TURB
13     REAL LAM,MU
14     C
15     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),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,PIPXR,
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,DMCYL,PXCONV
24     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
25     3 FREZYT,FREZYB,ROMPR,T,THIRD,NCLST,TOUT,TWFIN
26     COMMON/YSC2/TUQI,TUSI,NCG,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
27     1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
28     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
29     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVH,JSVH,QMN,QMX,WMAX,JNM,T2,TLIM,
30     4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
31     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
32     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
33     C
34     COMMON/YSC4/ITAB(ITABP)
35     COMMON/YSC4/ITAB(1000)
36     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
37     *      ===== END COMDECK YAGSC =====
38     C
39     MU=0,
40     LAM=.6
41     C
42     C      OM IS THE PHASE TWO ITERATION RELAXATION PARAMETER OMEGA
43     C
44     C      OM=.1.
45     C
46     C      EPS IS THE PRESSURE ITERATION CONVERGENCE CRITERION
47     C
48     C      EPS=1,E=5
49     C

```

50 C GR IS THE RADIAL COMPONENT OF THE BODY FORCE FELT BY THE ENTIRE
51 C PROBLEM
52 C
53 C GR=0.
54 C
55 C GZ IS THE AXIAL COMPONENT OF THE BODY FORCE FELT BY THE ENTIRE
56 C PROBLEM (USUALLY GRAVITY IN THE -Z DIRECTION)
57 C
58 C GZ=-980.
59 C
60 C FREZXR IS THE RATIO OF ZONE DR=8 AS ONE MOVES AWAY FROM
61 C THE REGION OF CONSTANT ZONING TOWARD THE RIGHT BOUNDARY
62 C
63 C FREZXR=1.
64 C
65 C FREZYT IS THE RATIO OF ZONE DZ=8 AS ONE MOVES AWAY FROM
66 C THE REGION OF CONSTANT ZONING TOWARD THE TOP BOUNDARY
67 C
68 C FREZYT=1.
69 C
70 C FREZYB IS THE RATIO OF ZONE DZ=8 AS ONE MOVES AWAY FROM
71 C THE REGION OF CONSTANT ZONING TOWARD THE BOTTOM BOUNDARY
72 C
73 C FREZYB=1.
74 C
75 C YB IS THE ACTUAL BOTTOM OF THE GRID
76 C
77 C YB=8.
78 C
79 C REZY0 IS THE CENTER OF THE REGION OF UNIFORM ZONING, EITHER YB OR
80 C REZYB MAY BE SPECIFIED.
81 C
82 C REZYB=0.
83 C
84 C REZRON IS THE INITIAL DENSITY OF THE AMBIENT ATMOSPHERE AT Z=REZY0
85 C
86 C REZRON=.001
87 C
88 C REZSIE IS THE SPECIFIC INTERNAL ENERGY OF THE AMBIENT ATMOSPHERE
89 C
90 C REZSIE=2.E10
91 C
92 C IBAR IS THE NUMBER OF REAL ZONES IN THE RADIAL DIRECTION
93 C
94 C IBAR=0
95 C
96 C JBAR IS THE NUMBER OF REAL ZONES IN THE AXIAL DIRECTION
97 C
98 C JBAR=0
99 C
100 C IUNF IS THE NUMBER OF UNIFORM ZONES IN THE RADIAL DIRECTION
101 C
102 C IUNF=0
103 C
104 C JUNF IS THE NUMBER OF UNIFORM ZONES IN THE AXIAL DIRECTION
105 C
106 C JUNF=0

```

107 C
108 C      JCEN IS THE J VALUE ABOVE AND BELOW WHICH THERE ARE AN EQUAL
109 C      NUMBER OF UNIFORM ZONES
110 C
111 C      JCEN=0
112 C
113 C      DR IS THE UNIFORM RADIAL MESH SPACING
114 C
115 C      DR=0.
116 C
117 C      DZ IS THE UNIFORM AXIAL MESH SPACING
118 C
119 C      DZ=0,
120 C
121 C      CYL IS THE GEOMETRY INDICATOR, CYL=1, FOR CYLINDRICAL GEOMETRY
122 C          #0, FOR PLANE (SLAB) GEOMETRY
123 C
124 C      CYL=1,
125 C
126 C      GRDVEL IS THE REZONE INDICATOR, GRDVEL#0, FOR EULERIAN
127 C          #1, FOR LAGRANGIAN
128 C          #2, FOR ALE
129 C
130 C      GRDVEL=2,
131 C
132 C      ANC IS THE NODE COUPLER CONSTANT
133 C
134 C      ANC=.05
135 C
136 C      ADFAC IS A FACTOR USED IN CALCULATING THE COURANT Timestep
137 C
138 C      ADFAC=.2
139 C
140 C      A0,A0M,AND B0 ARE REZONE COEFFICIENTS WHICH DETERMINE THE TYPE OF
141 C      DIFFERENCING USED IN CALCULATING THE FLUXING TERMS, A0 APPLIES
142 C      TO THE MOMENTUM EQN,A0M TO THE MASS AND ENERGY EQNS,AND
143 C      B0 TO ALL THREE, SOME EXAMPLES OF THEIR SIGNIFICANCE ARE
144 C          A0=0.,B0=0., CENTERED DIFFERENCING
145 C          A0=1.,B0=0., FULL DONOR CELL
146 C          A0=0.,B0=2., INTERPOLATED DONOR CELL.
147 C      NOTE THAT THE EQNS ARE UNSTABLE IN THE FIRST CASE.
148 C
149 C      A0=1
150 C      A0M=1,
151 C      B0=0.
152 C
153 C      KXI IS A PARAMETER GOVERNING THE TREATMENT OF VISCOSITY,
154 C          KXI#1 ALLOWS MU AND LAM TO BE TREATED AS NUMERICAL
155 C          VISCOSITY COEFFICIENTS
156 C          KXI#0 ALLOWS MU AND LAM TO BE TREATED AS TRUE PHYSICAL
157 C          VISCOSITY COEFFICIENTS
158 C          KXI#1 CAUSES TO CODE TO CALCULATE ITS OWN NUMERICAL
159 C          VISCOSITY COEFFICIENTS BASED ONLY UPON THE
160 C          RESTRICTION THAT THE RATIO OF THE MU AND LAM
161 C          COMPUTED INTERNALLY BY THE CODE IS EQUAL TO
162 C          THE RATIO OF THE MU AND LAM READ IN
163 C

```

```

164      KX1=1
165      C
166      C GZP IS THE BODY FORCE FELT BY THE PARTICLES, IT IS NOT NECESSARILY
167      C EQUAL TO GZ,
168      C
169      C GZP=980.
170      C
171      C ZORIO IS THE NO. OF FIREBALL RADII TO HOLD THE OUTER BOUNDARY
172      C AWAY FROM THE FIREBALL
173      C
174      C ZORIG=6,
175      C
176      C T IS THE PROBLEM START TIME
177      C
178      C T=0,
179      C
180      C DT IS THE INITIAL TIMESTEP
181      C
182      C DT=.001
183      C
184      C NCLST IS THE CYCLE NO. AT WHICH TO HALT THE RUN
185      C
186      C NCLST=99999
187      C
188      C TWFIN IS THE REAL TIME AT WHICH TO TERMINATE THE RUN
189      C
190      C TWFIN=1,E30
191      C
192      C DTO AND DTOC ARE OUTPUT FREQUENCY CONTROL ARRAYS. IF T.LE.DTOC(1),
193      C EDITS AND PLOTS OCCUR EVERY DTOC(1) PROBLEM SECONDS. IF
194      C DTOC(I=1),LT.T,LE,DTOC(I), THEN OUTPUT OCCURS EVERY DTOC(I)
195      C PROBLEM SECONDS. A MAXIMUM OF 10 DTOC,DTOC PAIRS MAY BE
196      C SPECIFIED.
197      C
198      C DTO(1)=1.
199      C DTOC(1)=1,E30
200      C DO 10 II=2,10
201      C DTOC(II)=0,
202      C 10 DTOC(II)=0.
203      C
204      C NCQ IS THE CYCLE ON WHICH TURBULENCE IS TO BE SEEDED, IF NCQ IS
205      C LESS THAN 0, THERE IS NO TURBULENCE.
206      C
207      C NCQ=1
208      C
209      C IF TURB IS ,FALSE,, THE TURBULENCE IS CURRENTLY OFF
210      C
211      C TURB=.FALSE.
212      C
213      C QLEVEL,TUQI,TUSI,AND TQ ARE ALL TURBULENCE QUANTITIES
214      C
215      C QLEVEL=.02
216      C TUQI=0,
217      C TUSI=0,
218      C TQ=0,
219      C
220      C A PARTICLE CAN BE MOVED NO MORE THAN

```

```

221 C      SQRT(2.)*WMAXEF*SQRT(2.*SIGMA*DT) DUE TO THE TURBULENT
222 C      DIFFUSION EFFECT DURING ANY ONE CYCLE
223 C
224 C      WMAXEF#2.
225 C
226 C      ANY PARTICLE WITH XJ,LE,RMINEF WILL NOT BE SUBJECT
227 C      TO TURBULENT DIFFUSION
228 C
229 C      RMINEF#50.
230 C
231 C      TSTRTD IS THE TIME AT WHICH TO START TURBULENT DIFFUSION
232 C
233 C      TSTRTD#1.
234 C
235 C      IST IS THE NUMBER OF PARTICLES TO FOLLOW AS A FUNCTION OF TIME.
236 C      IF IST.LT.0, NO PARTICLES WILL BE FOLLOWED.
237 C
238 C      IST#1
239 C
240 C      JDUMP IS THE FREQUENCY OF DUMP CYCLES
241 C
242 C      JDUMP#999999
243 C      RETURN
244 C      END

```

```

1      SUBROUTINE DMP(FWA,LWA,IFILE)
2      C
3      C      ROUTINE TO DUMP SCM FROM FWA TO LWA WITH THE OUTPUT DIRECTED
4      C      TO LOGICAL UNIT IFILE
5      C
6      C      WRITTEN BY J.L.NORTON, LASL T=3, 1975
7      C
8      C      INTEGER FWA,GETIT
9      C      DIMENSION IDUMP(4)
10     C
11     C      CHECK FOR ERRORS
12     C
13     C      IF(FWA,LE,LWA) GO TO 10
14     C
15     C      YES, PRINT MESSAGE AND QUIT.
16     C
17     C      WRITE(IFILE,100) FWA,LWA
18     C      RETURN
19     C      10 CONTINUE
20     C
21     C      ALL O.K., PRINT DUMP HEADER.
22     C
23     C      WRITE(IFILE,110) FWA,LWA
24     C
25     C      IFW IS THE ADDRESS OF THE FIRST WORD TO BE PRINTED ON THE
26     C      CURRENT LINE

```

```

27 C ILW IS THE ADDRESS OF THE WORD CURRENTLY BEING PROCESSED
28 C
29 C IFW=FWA
30 C ILW=FWA
31 C
32 C ICLW IS THE CONTENTS OF THE WORD WHOSE ADDRESS IS ILW
33 C
34 C ICLW=GETIT(ILW)
35 C
36 C IDUMP CONTAINS THE WORDS TO BE PRINTED ON THE CURRENT LINE
37 C (A MAXIMUM OF FOUR).
38 C ISUB IS THE NO. OF THE LAST LOCATION OF IDUMP THAT WAS FILLED,
39 C
40 C ISUB=1
41 C IDUMP(1)=ICLW
42 C 20 CONTINUE
43 C
44 C EXAMINE THE NEXT WORD
45 C
46 C ILW=ILW+1
47 C
48 C SEE IF WE ARE DONE
49 C
50 C IF(ILW,LE,LWA) GO TO 30
51 C
52 C YES, FLUSH IDUMP IF NECESSARY AND QUIT.
53 C
54 C IF(ISUB,LE,0) RETURN
55 C WRITE(IFILE,120) IFW,(IDUMP(I),I#1,ISUB)
56 C RETURN
57 C 30 CONTINUE
58 C
59 C NO, GAVE THE CONTENTS OF THE NEXT-TO-THE-LAST WORD AND
60 C GO GET THE CONTENTS OF THE CURRENT WORD.
61 C
62 C ICNLW=ICLW
63 C ICLW=GETIT(ILW)
64 C
65 C SEE IF THE LAST WORD AND THE NEXT-TO-THE-LAST WORD ARE THE SAME
66 C
67 C IF(ICLW,NE,ICNLW) GO TO 80
68 C
69 C YES, WE WILL GO INTO REPETITION MODE, FIRST FLUSH IDUMP
70 C IF NECESSARY.
71 C
72 C ISUB=ISUB-1
73 C IF(ISUB,LE,0) GO TO 40
74 C WRITE(IFILE,120) IFW,(IDUMP(I),I#1,ISUB)
75 C ISUB=0
76 C 40 CONTINUE
77 C
78 C IFW IS NOW THE ADDRESS OF THE FIRST WORD OF THE REPETITION GROUP
79 C
80 C IFW=ILW-1
81 C
82 C SCAN FORWARD UNTIL THE END OF THE REPETITION GROUP IS LOCATED
83 C

```

```

84      IFWP=IFW+2
85      DO 50 I=IFWP,LWA
86      ICLW=GETIT(I)
87      IF(ICLW,NE,ICNLW) GO TO 60
88      50 CONTINUE
89      C      ALL THE REST OF THE REQUESTED DUMP REGION LIES WITHIN THE
90      C      REPETITION GROUP
91      C
92      C      ILW=LWA
93      C      GO TO 70
94      C      60 CONTINUE
95      C
96      C      ILW IS THE ADDRESS OF THE LAST WORD IN THE REPETITION GROUP
97      C
98      C      ILW=I=1
99      C      70 CONTINUE
100     C
101     C      WRITE OUT THE REPETITION GROUP
102     C
103     C      WRITE(IFILE,130) IFW,ILW,ICNLW
104
105     C      GO BACK AND CONTINUE PROCESSING UNLESS WE ARE DONE
106     C
107     C      ILW=ILW+1
108     C      IF(ILW,GT,LWA) RETURN
109     C      IFW=ILW
110     C      ISUB=1
111     C      IDUMP(1)=ICLW
112     C      GO TO 20
113     C
114     C      80 CONTINUE
115     C
116     C      LAST AND NEXT-TO-THE-LAST WORD ARE DIFFERENT. SEE IF THE IDUMP
117     C      BUFFER IS FULL.
118     C
119     C      IF(ISUB,LT,4) GO TO 90
120     C
121     C      YES, FLUSH IT,
122     C
123     C      WRITE(IFILE,120) IFW,(IDUMP(I),I=1,4)
124     C      IFW=ILW
125     C      ISUB=0
126     C
127     C      90 CONTINUE
128     C
129     C      ADD THE CURRENT WORD TO THE IDUMP BUFFER AND CONTINUE
130     C
131     C      ISUB=ISUB+1
132     C      IDUMP(ISUB)=ICLW
133     C
134     C      100 FORMAT(1H ,50ERROR IN DMP ARGUMENTS, DMP BYPASSED, FWA AND LWA ,
135     C      1 4HARE ,2I10)
136     C      110 FORMAT(1H1,17HDUMP OF 8CH FROM ,06,4H TO ,06//)
137     C      120 FORMAT(1H ,06,5X4025)
138     C      130 FORMAT(1H ,06,6H THRU ,06,13H ALL CONTAIN ,020)
139     C      END

```

```

1      SUBROUTINE DMPPK(N,PACK)
2      C
3      C      ROUTINE TO DUMP EXCHANGE PACKAGE
4      C
5      C      WRITTEN BY LARRY RUDSINSKI AND JERRY MELANDEZ,LASL C=4
6      C      MODIFIED TO CORRECTLY PICK UP C(A0) - C(A7) BY J.L.NORTON,LASL TD3
7      C
8      DIMENSION PACK(1),PARCEL(4)
9      DIMENSION ISAVE(8)
10     DIMENSION NPC(8),NAC(8),NBC(8),NXC(8)
11     DIMENSION IAA(9),IA1(8),IA3(2)
12     INTEGER PACK,PARCEL
13     INTEGER PACK1
14     INTEGER GETIT,SHIFT,AND
15     DATA NP$/1HP,3HRA8,3HFL8,3HP8D,3HRAL,3HFLL,3HNEA,3HEEA/
16     DATA NAC/2HA0,2HA1,2HA2,2HA3,2HA4,2HA5,2HA6,2HA7/
17     DATA NBC/3HBPA,2HB1,2HB2,2HB3,2HB4,2HB5,2HB6,2HB7/
18     DATA NXC/2HX0,2HX1,2HX2,2HX3,2HX4,2HX5,2HX6,2HX7/
19     DATA IA1/5HC(A0),5HC(A1),5HC(A2),5HC(A3),5HC(A4),5HC(A5),5HC(A6),
20     ! 5HC(A7)/
21     DATA IA3/7H OUT OF,6H RANGE/
22     IF(N.EQ.0) NR6
23
24     C      PRINT CAUSE OF ABORT
25     C
26     IARG=SHIFT(PACK(4),-36)
27     CALL MODE(IARG,N)
28
29     C      GET THE FIELD LENGTH
30     C
31     IAA(9)=SHIFT(PACK(3),-36)
32
33     C      IF FIELD LENGTH IS GARBAGED,SET TO MAXIMUM
34     C
35     IF(IAA(9).GT.150077B) IAA(9)=150077B
36
37     C      PICK UP AND PRINT REGISTERS,ETC.
38     C
39     DO 30 I=1,8
40
41     C      PICK UP B REGISTER I=1
42     C
43     NB=AND(PACK(I),777777B)
44
45     C      PICK UP A REGISTER I=1
46     C
47     NA=AND(SHIFT(PACK(I),-18),777777B)
48
49     C      GET C(AN)
50     C
51     IF(NA.GE.IAA(9)) GO TO 10
52     ISAVE(I)=GETIT(NA)
53     10 CONTINUE
54

```

```

55      C      PICK UP DATA IN THE TOP OF EXCHANGE PACKAGE WORDS
56      C
57      C      NPNSHIFT(PACK(I),=36)
58      C
59      C      PICK UP X REGISTER I=1 AND PREPARE TO PRINT IT IN 4 PIECES
60      C
61      K=4
62      DO 20 J=1,4
63      PARCEL(K)=AND(SHIFT(PACK(I+8),=(J-1)*15),77777B)
64      KK=1
65      20 CONTINUE
66      IAA(I)=NA
67      WRITE(N,70) NPC(I),NP,NAC(I),NA,NBC(I),NB,NXC(I),PARCEL(1),PARCEL(
68      1 2),PARCEL(3),PARCEL(4),PACK(I+8),PACK(I+8)
69      30 CONTINUE
70      C
71      C      PRINT C(AN)
72      C
73      DO 60 I=1,8
74      C
75      C      CHECK FOR A REGISTER OUT OF RANGE
76      C
77      IF(IAA(I).GE.IAA(9)) GO TO 50
78      PACK1=ISAVE(I)
79      K=4
80      DO 40 J=1,4
81      PARCEL(K)=AND(SHIFT(PACK1,=(J-1)*15),77777B)
82      KK=1
83      40 CONTINUE
84      WRITE(N,80) IA1(I),PARCEL(1),PARCEL(2),PARCEL(3),PARCEL(4),PACK1,
85      1 PACK1
86      GO TO 60
87      C
88      C      A REGISTER IS OUT OF RANGE
89      C
90      50 WRITE(N,90) NAC(I),IA3(1),IA3(2)
91      60 CONTINUE
92      WRITE(N,100)
93      RETURN
94      C
95      70 FORMAT(1H ,A4,2X,08,2X,A2,2X,06,2X,A3,2X,06,7X,A2,2X,05,2X,05,2X,
96      1 05,2X,05,E25,13,8X,A10)
97      80 FORMAT(45X,A5,3(1X,05,1X),1X,05,E25,13,8X,A10)
98      90 FORMAT(47X,A2,1X,A7,A6)
99      100 FORMAT(1H ,//)
100     END

```

```

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

```

```

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
8
9   *      **** BEGIN COMDECK PARAM      ****
10  COMMON/P/COM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCPI2,
11  1 NLCPI3,NLCPI4,IFLMSZ
12  *      **** END COMDECK PARAM      ****
13  *      **** BEGIN COMDECK YSTORE      ****
14  *      **** BEGIN COMDECK YAQDIM      ****
15  DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
16  1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCGQ(1),E(1),ETIL(1),RVOL
17  2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
18  3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSRQ(1),CAPGAM(1),TUG
19  4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
20  5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
21  6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
22  *      **** END COMDECK YAQDIM      ****
23  *      **** BEGIN COMDECK YAQSC      ****
24  LOGICAL RESTRT,FILM,PAPER,TURB
25  REAL LAM,MU
26  C      COMMON/YSC1/AASC(NSCP1)
27  COMMON/YSC1/AASC(9600)
28  COMMON/YSC2/AA(1),ANC,A0,AOFAC,ABM,BR,COLAMU,CYL,DR,DT,DTA,DTFAC,
29  1 DTA(10),DTA(10),DTA2,DTA8,DTP09,DTV,DZ,EM10,EP,FIPXL,FIPXR,
30  2 FIPYB,FIPYT,FXL,FXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
31  3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
32  4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
33  COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
34  1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
35  2 ,PXL,PXR,PYB,PYCONV,PYT,ROT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
36  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
37  COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUBV,TURB,PTOP,PRITE,PBOTM,
38  1 ILNG,NILNG,TP3,TUPOF,TDQSAV,TK,TL,TUGENG,EP1,SAV1,LEVEL,TQ,IST,
39  2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
40  3 XCNVLD,FIXRO,FXL,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
41  4 ROMFXR,ROMFYT,ROMFYB,JUMP,TWTHR,TE,DTR,TMASS,DTVASV,DTCSAV,IDTV
42  5 ,JDTV,IDA,IDA(1),IDC,IDE,IDE(1),IDC(1),IDC(2),IDC(3),IDC(4),IDC(5),
43  6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
44
45  C      COMMON/YSC2/ZZ
46  COMMON/YSC4/ITAB(ITABP)
47  COMMON/YSC4/ITAB(1000)
48  COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IPD
49  *      **** END COMDECK YAQSC      ****
50  *      **** BEGIN COMDECK YAQE0      ****
51  EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
52  1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCGQ,MP),(AASC
53  1 (8),E,ETIL,AREA,XR13K),
54  2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
55  3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
56  4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
57  5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(21),
58  6 ),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
59  7 ),(AASC(25),DLSRQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
60  8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
61  9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
62  1 AVYSV,X24K)

```

```

62      REAL M,MP,MPAR,MTIL
63      *      END COMDECK YAGEQ      ****
64      *      END COMDECK YSTORE      ****
65      C      DR(Z)MIN(MAX) IS THE MINIMUM (MAXIMUM) OF THE MAGNITUDES OF
66      C      CELL SIDES IN THE R (Z) COORDINATE DIRECTION
67      C
68      DRMIN=1.E+20
69      DZMIN=1.E+20
70      DRMAX=0.
71      DZMAX=0.
72      C
73      C      VMAX IS THE MAXIMUM GRID VELOCITY MAGNITUDE ALONG EITHER
74      C      COORDINATE AXIS
75      C
76      VMAX=0.
77      IVM=0
78      JVM=0
79      CALL START
80      C
81      C      LOOP OVER ALL REAL ZONES
82      C
83      DO 30 J=2,JP1
84      DO 20 I=1,IBAR
85      IPJ=IJ+NQ
86      IPJP=IJP+NQ
87      VMAXP=AMAX1(VMAX,ABS(U(IJ)),ABS(V(IJ)))
88      IF(VMAXP,EQ,VMAX) GO TO 10
89      IVM=I
90      JVM=J
91      VMAX=VMAXP
92      10 CONTINUE
93      C
94      C      DETERMINE THE FOUR VERTICES OF CELL (I,J)
95      C
96      C      (X1,Y1) IS VERTEX (I+1,J)      (VERTEX 1)
97      C      (X2,Y2) IS VERTEX (I+1,J+1)    (VERTEX 2)
98      C      (X3,Y3) IS VERTEX (I,J+1)      (VERTEX 3)
99      C      (X4,Y4) IS VERTEX (I,J)        (VERTEX 4)
100     C
101     X1=X(IPJ)
102     X2=X(IPJP)
103     X3=X(IJP)
104     X4=X(IJ)
105     Y1=Y(IPJ)
106     Y2=Y(IPJP)
107     Y3=Y(IJP)
108     Y4=Y(IJ)
109     C
110     C      DETERMINE THE SQUARE OF THE LENGTH OF EACH SIDE
111     C
112     C      X(Y)NM IS THE SQUARE OF THE LENGTH OF THE SIDE BOUNDED BY
113     C      VERTICES N AND M
114     C
115     X14=(X1-X4)**2+(Y1-Y4)**2
116     X23=(X2-X3)**2+(Y2-Y3)**2
117     Y21=(X2-X1)**2+(Y2-Y1)**2
118     Y34=(X3-X4)**2+(Y3-Y4)**2

```

```

119      DRMIN=AMIN1(DRMIN,X14,X23)
120      DRMAX=AMAX1(DRMAX,X14,X23)
121      DZMIN=A MIN1(DZMIN,Y21,Y34)
122      DZMAX=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      DZMIN=SQRT(DZMIN)
130      DZMAX=SQRT(DZMAX)
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

```

```

37      FJ=J=1
38      D=X***(2-J)
39      U=D*X
40
41      C      TWO SEPARATE RATIONAL APPROXIMATIONS ARE USED, BOTH HAVE BEEN
42      C      DERIVED USING A PROGRAM BASED ON MAEHLY'S SECOND DIRECT
43      C      METHOD DESCRIBED IN JOURNAL OF THE ACM, VOL. 10, NO. 3
44
45      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
46      1 *(P(6,J)+U*(P(7,J)))))))/(1.+U*(Q(1,J)+U*(Q(2,J)+U*(Q(3,J)+U*(Q(4,
47      2 J)+U*(Q(5,J)+U*(Q(6,J)))))))+FJ)*SIGN(1.,Z)
48      RETURN
49      END

```

```

1      SUBROUTINE FILMCO
2
3      C      ROUTINE TO UPDATE GRID LIMITS AND PARTICLE QUANTITIES
4
5      C      ORIGINALLY WRITTEN BY A.A. AMSDEN, LASL T=3
6      C      MODIFIED AND DOCUMENTED BY J.L. NORTON, LASL T=3, 1974
7
8      *      ***** BEGIN COMDECK PARAM ***** *
9      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
10     1 NLCP3,NLCP4,IFLMSZ
11     *      ***** END COMDECK PARAM ***** *
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),DELSMC(
15     1 1),VG(1),R0(1),SIE(1),MP(1),RMP(1),RC3Q(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 ,VU(1),ROL(1),AVXSV(1),AVYSV(1),DL8ROI(1),DL8R0Q(1),CAPGAM(1),TUQ
18     4 (1),SIG(1),TUS(1),GRROR(1),GRR0Z(1),GRR0P(1),TUQVEC(1),MTIL(1),
19     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GR8V(1),GZ8V(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
26     C      COMMON/YSC1/AASC(NSCP1)
27     COMMON/YSC1/AASC(9600)
28     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTCA,DTFAC,
29     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
30     2 FIPYB,FIPYT,FIXL,FIYR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
31     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
32     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
33     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
34     1 NCYC,NLC,NPS,NPT,NG,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
35     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
36     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
37     COMMON/YSC2/TUGI,TUSI,NCG,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
1 ILNG,NILNG,TP3,TUPOT,TDQS4V,TK,TI,TUGENG,EP1,SAV1,QLLEVEL,TQ,IST,
```

```

38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRO,IXLO,IXRO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLM,
40      4 ROMFXR,ROMFYR,JDUMP,TWTHR,DTR,TMASS,DTVSAR,DTCSAV,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXEF,RMINEF,TSTRD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      C COMMON/YSC4/ITAB(1000)
46      C COMMON/YSC5/RESTRRT,FILM,PAPER,IPD,IFD
47      * ===== END COMDECK YAQEQ =====
48      * ===== BEGIN COMDECK YAGEQ =====
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),8IE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSRQ,VG),(AASC(26),GRSV),(AASC(27),GRRDP,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ===== END COMDECK YAQEQ =====
63      * ===== END COMDECK YSTORE =====
64      C FIND THE GRID LIMITS. MAX(X)=XR,MAX(Y)=YT,MIN(X)=XL=0.,MIN(Y)=YB,
65      C
66      XL=0.0
67      YB=1.E+20
68      XR=YT=YB
69      CALL START
70      DO 20 J=2,JP2
71      DO 10 I=1,IP1
72      XR=AMAX1(XR,X(IJ))
73      YB=AMIN1(YB,Y(IJ))
74      YT=AMAX1(YT,Y(IJ))
75      10 IJ=IJ+NG
76      CALL LOOP
77      20 CONTINUE
78
79      C VV IS USED IN SCALING VELOCITY VECTOR PLOTS,
80      C VV = .9*MAX(X)/IBAR
81
82      C VV=.9*XR*RIBAR
83
84      C FIYB IS THE LOCATION OF MIN(Y) IN RASTER COUNTS
85
86      C FIYB=916.0
87
88      C XD IS THE RATIO OF GRID WIDTH TO HEIGHT
89
90      C XD=XR/(YT-YB)
91
92      C ONE WISHES TO MAKE THE PLOTS FILL THE FILM FRAME AS NEARLY AS
93      C POSSIBLE. AT MOST THE PLOT CAN BE 1022 RASTER POINTS WIDE AND
94      C 900 RASTER POINTS TALL (LEAVING ROOM FOR LABELS AT THE BOTTOM

```

```

95      C      AND A 16 POINT MARGIN AT THE TOP,) AN IDEAL GRID, THEN, WOULD
96      C      HAVE  $XD=1022/900=1.13556$ . IF  $XD > 1.13556$ , THE GRID IS DEFINED
97      C      AS WIDER THAN HIGH. IF  $XD \leq 1.13556$ , THE GRID IS DEFINED AS
98      C      HIGHER THAN WIDE. IN THE FORMER CASE, THE X COORDINATE RASTER
99      C      BOUNDS ARE SET TO (0,1022) AND THE Y COORDINATE RASTER BOUNDS
100     C      TO (FIYT,916) WHERE FIYT IS DETERMINED SUCH THAT THE X AND Y
101     C      SCALES ARE THE SAME. IN THE LATTER CASE, THE Y COORDINATE
102     C      RASTER BOUNDS ARE SET TO (16,916) AND THE X COORDINATE RASTER
103     C      BOUNDS ARE DETERMINED SUCH THAT THE X AND Y SCALES ARE EQUAL
104     C      AND THE LEFT AND RIGHT MARGINS ARE THE SAME.
105     C
106     C      IF(XD,LE,1.13556) GO TO 30
107     C
108     C      GRID WIDER THAN HIGH, THE X RASTER BOUNDS ARE (FIXL, FIXR), THE
109     C      Y RASTER BOUNDS ARE (FIYT, FIYB),
110     C
111     C      FIXL#0,
112     C      FIXR#1022,
113     C      FIYT#916,-1022,/XD
114     C      GO TO 40
115     C
116     C      GRID HIGHER THAN WIDE
117     C
118     C      30 CONTINUE
119     C      FIXL#AMAX1(0,,511,-450,*XD)
120     C      FIXR#511,+450,*XD
121     C      FIYT#16,
122     C      40 CONTINUE
123     C
124     C      XCONV AND YCONV ARE FACTORS TO CONVERT FROM X AND Y CARTESIAN
125     C      COORDINATES TO RASTER COORDINATES
126     C
127     C      XCONV=(FIXR-FIXL)/(XR-XL)
128     C      YCONV=(FIYT-FIYB)/(YT-YB)
129     C
130     C      PROVIDE FIXED POINT VALUES OF THE RASTER BOUNDS
131     C
132     C      IXL#FIXL
133     C      IXR#FIXR
134     C      IYB#FIYB
135     C      IYT#FIYT
136     C
137     C      IF THERE ARE NO PARTICLES, WE ARE DONE
138     C
139     C      IF(NPT,EQ,0) RETURN
140     C
141     C      THERE ARE PARTICLES, CALCULATE THE CARTESIAN PARTICLE BOUNDS
142     C      FROM THE GRID BOUNDS,
143     C
144     C      PXL#0,0
145     C      PYB#YB
146     C      PXR#XR
147     C      PYT#YT
148     C      FIPYB#916,0
149     C      FIPXL#FIXL
150     C      FIPXR#FIXR
151     C      FIPYT#FIYT

```

```

152 C
153 C CONVERSION FACTORS FROM PARTICLE COORDINATES TO RASTER COORDINATES
154 C
155 PXCONV=(FIPXR-FIPXL)/(PXR-PXL)
156 PYCONV=(FIPYT-FIPYB)/(PYT-PYB)
157 C
158 C PROVIDE FIXED POINT RASTER BOUNDS
159 C
160 IPXL=FIPXL
161 IPXR=FIPXR
162 IPYB=FIPYB
163 IPYT=FIPYT
164 RETURN
165 END

```

```

1 SUBROUTINE GETOMG
2 C
3 C ROUTINE TO CALCULATE VORTICITY
4 C
5 C ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
6 C MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
7 C
8 *----- BEGIN COMDECK PARAM -----*
9 COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCM1,NLCM2,
10 NLCM3,NLCM4,IFLMSZ
11 *----- END COMDECK PARAM -----*
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),DELSMC(
15 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC3Q(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),DLSSROQ(1),CARGAM(1),TUQ
18 4(1),SIG(1),TUS(1),GRRDR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
19 5,CONC(1),CTEMP(1),ANCUC(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 YAGSC -----*
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,DTA,DTFAC,
28 1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29 2 FIPYB,FIPYT,PIXL,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
32 COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33 1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
35 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36 COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,

```

```

37      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRO,FXL0,IXRO,IXL0,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMDM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
47      * ===== END COMDECK YAQSC =====
48      * ===== BEGIN COMDECK YAQEQ =====
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),RD),(AASC(7),DELSM,RC8G,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRRDR),(AASC(22),GRROZ),(AASC(23),DL8ROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRR0P,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANGV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ===== END COMDECK YAQEQ =====
63      * ===== END COMDECK YSTORE =====
64      10 CALL START
65      WMAX=EM10
66      QMN=1,E30
67      GMX=QMN
68      DO 30 J=2,JP1
69      DO 20 I=1,IBAR
70      IPJ=IJ+NQ
71      IPJP=IJP+NQ
72      IF(J.EQ.2) CQ(IJM)=0.
73      IF(J.EQ.JP1) CQ(IJP)=0.
74      IF(I.EQ.IBAR) CQ(IPJ)=0,
75      X1=X(IPJ)
76      Y1=Y(IPJ)
77      U1=U(IPJ)
78      V1=V(IPJ)
79      X2=X(IPJP)
80      Y2=Y(IPJP)
81      U2=U(IPJP)
82      V2=V(IPJP)
83      X3=X(IJP)
84      Y3=Y(IJP)
85      U3=U(IJP)
86      V3=V(IJP)
87      X4=X(IJ)
88      Y4=Y(IJ)
89      U4=U(IJ)
90      V4=V(IJ)
91      R1=.125*RVOL(IJ)*(R(IPJ)+R(IPJP)+R(IJP)+R(IJ))
92      CQ(IJ)=R1*((U1+U4)*(X1-X4)+(V1+V4)*(Y1-Y4)+(U2+U1)*(X2-X1)+(V2+V1)
93      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,10HSQM DIRECT,
12     2 6H RANGE,10HLCM DIRECT,6H RANGE,10HSQM BLOCK ,5HRANGE,
13     3 10HLCM BLOCK ,5HRANGE,10HSQM 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 $1HTHE FOLLOWING CONDITION EXISTED AT THE TIME OF THE ,
29     1 7HABORT=,,2A10)
30     END

```

```

1      SUBROUTINE MSHMKR
2
3      C
4      C      ROUTINE TO GENERATE THE INITIAL PROBLEM MESH AND SET THE
5      C      INITIAL QUANTITIES
6
7      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
8      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
9
10     *      **** BEGIN COMDECK PARAM      ****
11     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
12     1 NLCP3,NLCP4,IFLMSZ
13     *      **** END COMDECK PARAM      ****
14     C      **** BEGIN COMDECK LCMATM      ****
15     COMMON/YSC3/UFIREF(IPFB),EFIREF(IPFB),RHOFIR(IPFB),XFIRE(IPFB)
16     COMMON/YSC3/UFIREF(200),EFIREF(200),RHOFIR(200),XFIRE(200)
17     *      **** END COMDECK LCMATM      ****
18     COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
19     *      **** BEGIN COMDECK YSTORE      ****
20     *      **** BEGIN COMDECK YAQDIM      ****
21     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
22     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC8Q(1),E(1),ETIL(1),RVOL
23     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CO(1),VTIL(1)
24     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSRQQ(1),CAPGAM(1),TUQ
25     4 (1),SIG(1),TU8(1),GRROR(1),GRR0Z(1),GRDROP(1),TUQVEC(1),HTIL(1),
26     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
27     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
28     *      **** END COMDECK YAQDIM      ****
29     *      **** BEGIN COMDECK YAQSC      ****
30     LOGICAL RESTRT,FILM,PAPER,TURB
31     REAL LAM,MU
32     C      COMMON/YSC1/AASC(NSCP1)
33     COMMON/YSC1/AASC(9600)
34     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
35     1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
36     2 FIPYB,FIPYT,FXL,FXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
37     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
38     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
39     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
40     1 NCYC,NLC,NPS,NPT,NG,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
41     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
42     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
43     COMMON/YSC2/TUQI,TU8I,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
44     1 ILNG,NILNG,TP3,TUPOT,TDQS4V,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
45     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
46     3 XCNVLD,FIXRO,FXL,IXRO,IXL,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
47     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DT,TMAS,DTVS4V,DTC4V,DTV
48     5 ,JDTV,JDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
49     6 ,TMAS4V,WMAXEF,RMINEF,T8TRTD
50     C      COMMON/YSC2/ZZ
51     COMMON/YSC4/ITAB(ITABP)
52     COMMON/YSC4/ITAB(1000)
53     *      **** END COMDECK YAQSC      ****
54     *      **** BEGIN COMDECK YAQEQ      ****
55     EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),
56     1 (AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RC8Q,MP),(AASC

```

```

57      1 (8),E,ETIL,AREA,XR13K),
58      2 (AA8C(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
59      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
60      4 UL,PMX,PU),(AABC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CO,ROL),(AASC(17
61      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
62      6 21),GRRQR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
63      7 ),(AASC(25),DL3R0Q,VG),(AASC(26),GRSV),(AASC(27),GRR0P,TUGVEC,
64      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
65      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
66      1 AVYSV,X24K)
67      REAL M,MP,MPAR,MTIL
68      *      ===== END COMDECK YAQE0 =====
69      *      ===== END COMDECK YSTORE =====
70      DIMENSION MESH(3,4)
71      LOGICAL FBFILE
72      DATA (MESH(IJ),IJ=1,12)/12*0/
73      DATA IEFLAG/0/
74      DATA PJLN/0./
75      C
76      C      SET UP THE NAMELIST INPUT TABLE
77      C
78      ASSIGN 530 TO IERRT
79      CALL TABDEF(MESH,4HFIREF,4,IERRT)
80      CALL TABSET(MESH,4HNRAD,NRAD,IEFLAG,0,0,0,0)
81      CALL TABSET(MESH,3HNTH,NTH,IEFLAG,0,0,0,0)
82      CALL TABSET(MESH,6HFBFILE,FBFILE,IEFLAG,0,0,0,0)
83      C
84      C      INITIALIZE. NQIM IS THE NO. OF WORDS OF MEMORY NEEDED TO STORE
85      C      ONE ROW OF DATA (LESS 1).
86      C
87      NQIM=NQI=1
88      IF(.NOT.TURB) GO TO 10
89      TNEG=0.
90      TNEGSV=0.
91      10 CONTINUE
92      C
93      C      DEFINE THE CELL VERTICES AS THOUGH THE ENTIRE MESH WERE UNIFORM
94      C
95      XX=0.0
96      YY=YB
97      CALL START
98      DO 50 J=2,JP2
99      DO 40 I=1,IP1
100     X(IJ)=XX
101     Y(IJ)=YY
102     R(IJ)=XXX*CYL+OMCYL
103     U(IJ)=0.
104     V(IJ)=0.
105     IF(J.NE.2) GO TO 20
106     Y(IJM)=YY+DZ
107     X(IJM)=XX
108     R(IJM)=R(IJ)
109     U(IJM)=0.
110     V(IJM)=0.
111     20 IF(J.NE.JP2) GO TO 30
112     Y(IJP)=YY+DZ
113     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      C 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      C JCEN=JCEN+2
137      C
138      C JT0P AND JB0T ARE THE J-LINES AT THE TOP AND BOTTOM OF THE
139      C UNIFORM REGION
140      C
141      C JT0P=JCEN+JUNFO2
142      C JB0T=JCEN-JUNFO2
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      C TJ=FLOAT(JUNFO2)*DZ
148      C
149      C LOOP TO SET THE NON-UNIFORM VERTICES
150      C
151      C CALL START
152      C DO 110 J=2,JP2
153      C DO 100 I=1,IP1
154      C IMJ=IJ-NQ
155      C IF(FREZXR.EQ.1,) GO TO 60
156      C
157      C SEE IF WE ARE WITHIN THE UNIFORM X REGION
158      C
159      C IF(I.LE.IUNF+1) GO TO 60
160      C
161      C NO, GENERATE THE NON-UNIFORM X USING GEOMETRIC PROGRESSION,
162      C
163      C X(IJ)=X(IMJ)+FREZXR*(X(IMJ)-X(IMJ=NQ))
164      C R(IJ)=X(IJ)*CYL+OMCYL
165      C 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      C JDT=J-JTOP

```

```

171 C
172 C      JDB IS THE NO. OF J LINES THAT THE CURRENT J IS BELOW THE BOTTOM
173 C      OF THE UNIFORM REGION
174 C
175 C      JDB#JBOT#J
176 C
177 C      SEE IF THE CURRENT J IS BELOW THE UNIFORM REGION
178 C
179 C      IF(JDB.GT.0) GO TO 70
180 C
181 C      NO. SEE IF IT IS ABOVE THE UNIFORM REGION,
182 C
183 C      IF(JDT.GT.0) GO TO 80
184 C
185 C      NO. J IS IN UNIFORM REGION, SET IT AS SUCH.
186 C
187 C      Y(IJ)#REZY0+FLOAT(J=JCEN)*DZ
188 C      GO TO 90
189 C
70 CONTINUE
190 C
191 C      J IS BELOW THE UNIFORM REGION, CALCULATE ITS POSITION USING
192 C      FORMULA FOR SUM OF GEOMETRIC PROGRESSION, THE FORMULA IS
193 C      SUM=A*(1-F**N)/(1-F)
194 C      WHERE A IS THE FIRST TERM, N IS THE NO. OF TERMS, AND F IS
195 C      THE RATIO OF THE MTH TERM TO THE (M-1)TH TERM, IN THIS CASE
196 C      ANDZ*FREZ, N=JDB, AND F=FREZ, THE POSITION OF JBOT IS
197 C      AT Y=REZY0-TJ, J IS SUM BELOW THIS OR
198 C      Y(J)=REZY0-TJ=DZ*FREZ*(1-FREZ**JDB)/(1-FREZ),
199 C
200 C      IF(FREZYB.GT.1.) Y(IJ)=REZY0-TJ=DZ*FREZYB*(1,-FREZYB**JDB)*ROMFYB
201 C      GO TO 90
202 C
80 CONTINUE
203 C
204 C      J IS ABOVE THE UNIFORM REGION, USE GEOMETRIC PROGRESSION SUM TO
205 C      CALCULATE ITS VALUE ALSO,
206 C
207 C      IF(FREZYT.GT.1.) Y(IJ)=REZY0+TJ+DZ*FREZYT*(1,-FREZYT**JDT)*ROMFYT
208 C
90 CONTINUE
209 C
210 C      SEE IF J IS 2
211 C
212 C      IF(J.NE.2) GO TO 100
213 C
214 C      YES, SET YB (BOTTOM Y).
215 C
216 C      YB=Y(IJ)
217 C
218 C      MAKE THE FICTITIOUS CELLS AT THE BOTTOM AS THOUGH THERE WERE
219 C      ANOTHER REAL ROW OF CELLS
220 C
221 X(IJM)=X(IJ)
222 IJM#IJM+NQ
223 IJP#IJP+NQ
224 100 IJ#IJ+NQ
225 CALL LOOP
226 110 CONTINUE
227 CALL DONE

```

```

228      C
229      CC   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      CC   MESH IS GENERATED
241      C
242      130 CONTINUE
243      C
244      CC   ****
245      CC   GENERATE AN INITIAL FIREBALL AND ITS AMBIENT ATMOSPHERE
246      CC   ****
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=REZRDN
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=REZRDN*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)

```

```

285      C      SET AMBIENT QUANTITIES IN ROWS J=1 AND 2
286      C
287      C      DO 140 I=1,IP1
288      C      RO(IJ)=ROSAV
289      C      RO(IJM)=ROJ1
290      C      E(IJ)=REZSIE
291      C      SIE(IJ)=REZSIE
292      C      E(IJM)=REZ8IE
293      C      SIE(IJM)=REZ8IE
294      C      IJ=IJ+NQ
295      C      IJM=IJM+NQ
296      140 IJM=IJM+NQ
297      C
298      C      BRING IN THE NEXT ROW
299      C
300      C      CALL LOOP
301      C
302      C      LOOP OVER ALL THE OTHER REAL ROWS. BOOTSTRAP DENSITIES UPWARD
303      C      FROM ROW J=2.
304      C
305      DO 160 J=3,JP1
306      FDEN=(Y(IJP)-Y(IJ))*YY
307      FNUM=(Y(IJ)-Y(IJM))*YY
308      ROSAV=ROSAV+(XX=FNUM)/(XX+FDEN)
309      DO 150 I=1,IP1
310      RO(IJ)=ROSAV
311      E(IJ)=REZSIE
312      SIE(IJ)=REZSIE
313      150 IJ=IJ+NQ
314      CALL LOOP
315      160 CONTINUE
316      C
317      C      SET THE TOP FICTITIOUS ROW
318      C
319      FNUM=FNUM+FREZYT
320      FDEN=FDEN+FREZYT
321      ROJP2=ROSAV*(XX=FNUM)/(XX+FDEN)
322      DO 170 I=1,IP1
323      RO(IJ)=ROJP2
324      E(IJ)=REZSIE
325      SIE(IJ)=REZSIE
326      170 IJ=IJ+NQ
327      CALL DONE
328      C
329      C      THE AMBIENT ATMOSPHERE IS NOW SET. READ IN THE DATA TO GENERATE
330      C      THE FIREBALL. ISUB IS THE DATA POINT SUBSCRIPT. INTERPOLATION
331      C      TABLES ARE READ STARTING WITH ENTRY 2.
332      C
333      ISUB=1
334      C
335      C      READ INFORMATION ABOUT THE FINENESS OF THE FIREBALL
336      C      INTERPOLATION AND FROM WHERE THE FIREBALL INPUT DATA
337      C      WILL COME
338      C
339      IBFILE=5
340      NRAD=5
341      NTH=180

```

```

342      FBFILE=.FALSE.
343      CALL NAMLST(MESH,5,IEFLAG)
344      IF(IEFLAG.NE.0) CALL UNCLE(4,6HMSHMKR,25,
345      1 25HFIRE NAMELIST INPUT ERROR)
346      C
347      C     SEE IF INPUT WILL BE FROM SPECIAL FILE
348      C
349      C     IF(,NOT,FBFILE) GO TO 200
350      C
351      C     YES, PRINT A MESSAGE TELLING THE USER SO.
352      C
353      DO 180 IPX#6,IFD,6
354      180 WRITE(IPX,540)
355      IBFILE#3
356      C
357      C     PRINT THE DATA TABLE HEADER
358      C
359      DO 190 IPX#IPD,IFD,6
360      190 WRITE(IPX,570)
361      C
362      C     LOOP TO READ DATA POINTS
363      C
364      200 CONTINUE
365      ISUB=ISUB+1
366      C
367      C     SEE IF NEXT READ WILL OVERFLOW THE TEMPORARY STORAGE
368      C
369      IF(ISUB.GT.IPPB) CALL UNCLE(4,6HMSHMKR,29,
370      1 29HTOO MANY FIREBALL INPUT CARDS)
371      C
372      C     FIREBALL DATA IS IN SPHERICAL LAGRANGIAN FORM, INPUT IS
373      C     PRESSURE, RADIAL VELOCITY, SPECIFIC INTERNAL ENERGY, AND
374      C     DENSITY, IN CGS UNITS. THE VELOCITY IS DEFINED AT THE GIVEN
375      C     RADIUS. THE SPECIFIC INTERNAL ENERGY AND DENSITY FOR THE
376      C     CELL BETWEEN XFIRE(K-1) AND XFIRE(K) IS READ IN WITH
377      C     XFIRE(K). THE FIREBALL IS ASSUMED TO BE
378      C     CENTERED AT REZY0. THE PRESSURE IS ONLY INFORMATIVE, IT IS
379      C     NOT USED IN THE CALCULATION BUT RATHER IS RECALCULATED FROM
380      C     THE INPUT ENERGY AND DENSITY USING THE EQN=OF=STATE.
381      C
382      READ(IBFILE,580) PJLN,UFIRE(ISUB),EFIRE(ISUB),RHOFIR(ISUB),XFIRE(
383      1 ISUB)
384      C
385      C     CHECK FOR EOF ON INPUT. IF SO, CONSIDER END OF INPUT.
386      C
387      IF(EOF,IBFILE) 230,210
388      210 CONTINUE
389      C
390      C     CALCULATE THE PRESSURE FROM THE EQN=OF=STATE TO BE USED IN THE
391      C     PRINTOUT TO COMPARE WITH THE INPUT PRESSURE. A ZERO DENSITY
392      C     INDICATES THAT ALL THE DATA CARDS HAVE BEEN READ. (A BLANK
393      C     CARD TERMINATES THE INPUT.)
394      C
395      ROTMP=RHOFIR(ISUB)
396      IF(ROTMP.EQ.0,) GO TO 230
397      C
398      C     GO GET GAMMA=1 FROM THE EOS. IF SIE.GE.1.E10, METHANE IS

```

```

399      C      ASSUMED (CONCJ=1.). OTHERWISE, AIR IS USED (CONCJ=0.).
400
401      C      ETMP=EFIRE(ISUB)
402      C      CONCJ=0,
403      C      IF(ETMP.GE.1.E10) CONCJ=1,
404      C      CALL AIR
405      C      PRHO=GMONE*ROTMP*ETMP
406
407      C      PRINT OUT THE INPUT DATA AND THE COMPARISON PRESSURES FOR
408      C      THIS DATA POINT
409      C
410      DO 220 IPX=IPD,IFD,6
411      220 WRITE(IPX,590) UFIRE(ISUB),EFIRE(ISUB),RHOFIR(ISUB),XFIRE(ISUB),
412      1 PJLN,PRHO
413      C
414      C      SEE IF RADIUS IS MONOTONIC INCREASING UNLESS FIRST DATA POINT,
415      C      IN THE LATTER CASE, GO ON AND PROCESS THE NEXT POINT,
416      C
417      IF(ISUB.EQ.2) GO TO 200
418      IF(XFIRE(ISUB).LE.XFIRE(ISUB-1)) CALL UNCLE(4,6HMSHMKR,40,
419      1 40HINPUT RADII ARE NOT MONOTONIC INCREASING)
420
421      C      GO BACK AND PROCESS THE NEXT POINT
422      C
423      C      GO TO 200
424      230 CONTINUE
425      DO 240 IPX=6,IFD,6
426      240 WRITE(IPX,550)
427
428      C      GET RID OF THE FIREBALL INPUT DATA FILE IF ONE EXISTS
429      C
430      IF(FBFILE) CALL CLOSIT(3)
431
432      C      ALL DONE READING INPUT, SET THE DATA POINT COUNT TO THE TRUE NO.
433      C
434      ISUB=ISUB-1
435
436      C      SET THE QUANTITIES AT THE CENTER OF THE SPHERE. THE ENERGY AND
437      C      DENSITY ARE SET TO THOSE AT THE FIRST DATA POINT READ, THE
438      C      VELOCITY IS ZERO, OF COURSE.
439
440      UFIRE(1)=0,
441      EFIRE(1)=EFIRE(2)
442      RHOFIR(1)=RHOFIR(2)
443      XFIRE(1)=0.
444
445      C      LOOP OVER ALL ZONES AND INTERPOLATE THE CELL QUANTITIES
446
447      C      CALL START
448      DO 300 J=2,JP1
449      DO 290 I=1,IBAR
450
451      C      FIND THE VERTICES OF THE CELL OF INTEREST
452
453      VXL=X(IJ)
454      IPJ=IJ+NQ
455      VXR=X(IPJ)

```

```

456      VYB=Y(IJ)
457      VYT=Y(IJP)
458
459      C CONC(IJ)=0, INDICATES NOTHING HAS BEEN SET IN THE
460      C ZONE YET
461
462      C CONC(IJ)=0.
463
464      C LOOP OVER ALL 1-D ZONES
465
466      C DO 280 K=2,ISUB
467
468      C SEE IF ZONE COULD NOT POSSIBLY FALL WITHIN THE YAQUI ZONE
469
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
479      C LOOP OVER ANGULAR INCREMENTS
480
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
490      C LOOP OVER RADIAL INCREMENTS
491
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
500      C SEE IF THE X-COORDINATE OF THE CENTER OF THE LAGRANGIAN
501      C PIECE LIES IN THE YAQUI CELL OF INTEREST
502
503      IF(XCEN,LT,VXL,OR,XCEN,GT,VXR) GO TO 260
504
505      C YES, CHECK THE Y-COORDINATE
506
507      YCEN=CR*COST+REZY0
508      IF(YCEN,LT,VYB,OR,YCEN,GT,VYT) GO TO 260
509
510      C CONSIDER THE PIECE IN THE YAQUI CELL, SEE IF ANY PIECES
511      C HAVE PREVIOUSLY BEEN FOUND.
512

```

```

513      IF( CONC(IJ).NE.0.) GO TO 250
514      C
515      C NO, INITIALIZE,
516      C
517      R0(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*((CDG(THETA1)-CDG(THETA2))*(R2**3-R1**3))
527      CONC(IJ)=CONC(IJ)+VOL
528      XMASS=VOL*RHOFLIR(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      R0(IJ)=R0(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      USE VALUES FROM NEIGHBORING CELL FOR DETERMINING
573      C      CONTRIBUTION OF PART OF CELL OUTSIDE OF THE FIREBALL
574      C
575      C      IF(CONC(IPJ),NE,0,) GO TO 320
576      C
577      C      USE CELL TO RIGHT
578      C
579      C      INDEX=IPJ
580      C      GO TO 360
581      C      320 CONTINUE
582      C      IF(CONC(IJP),NE,0,) GO TO 330
583      C
584      C      USE CELL ABOVE
585      C
586      C      INDEX=IJP
587      C      GO TO 360
588      C      330 CONTINUE
589      C      IF(CONC(IJM),NE,0,) GO TO 340
590      C
591      C      USE CELL BELOW
592      C
593      C      INDEX=IJM
594      C      GO TO 360
595      C      340 CONTINUE
596      C      IPJP=IJP+NQ
597      C      IF(CONC(IPJP),NE,0,) GO TO 350
598      C
599      C      USE CELL TO UPPER RIGHT
600      C
601      C      INDEX=IPJP
602      C      GO TO 360
603      C      350 CONTINUE
604      C      IPJM=IJM+NQ
605      C      IF(CONC(IPJM),NE,0,) CALL UNCLE(1,6HMSHMKR,50,
606      C      1 50HERROR IN PROCESSING CELL WITH LARGE RELATIVE ERROR)
607      C
608      C      USE CELL TO LOWER RIGHT
609      C
610      C
611      C      INDEX=IPJM
612      C      360 CONTINUE
613      C
614      C      ADD IN CONTRIBUTION FROM PART OF CELL OUTSIDE OF THE FIREBALL
615      C
616      C      DV=VOL=CONC(IJ)
617      C      DM=DV*RO(INDEX)
618      C      RO(IJ)=RO(IJ)+DM
619      C      SIE(IJ)=SIE(IJ)+DM*SIE(INDEX)
620      C      370 CONTINUE
621      C      IJ=IJ+NQ
622      C      IJP=IJP+NQ
623      C      IJM=IJM+NQ
624      C      380 CONTINUE
625      C      CALL LOOP
626      C      390 CONTINUE

```

```

627 C
628 C      ALL FIREBALL CELLS ARE NOW COMPLETELY DEFINED,
629 C      GO BACK AND DEFINE THE VELOCITIES, STORE THEM
630 C      TEMPORARILY IN UG,VG,
631 C
632 C      CALL START
633 DO 440 J=2,JP1
634 DO 430 I=1,IBAR
635 UG(IJ)=0.
636 VG(IJ)=0.
637 IPJ=IJ+NQ
638 C
639 C      SEE IF THE CURRENT CELL IS A FIREBALL CELL
640 C
641 IF( CONC(IJ),EQ,0.) GO TO 420
642 C
643 C      YES, ALL NEIGHBORING CELLS OF THE LOWER LEFTHAND VERTEX
644 C      MUST ALSO BE FIREBALL CELLS OR VELOCITY IS SET TO ZERO,
645 C
646 IF(I,EQ,1) GO TO 400
647 IMJ=IJ+NQ
648 IF( CONC(IMJ),EQ,0.) GO TO 420
649 IMJM=IJM=NQ
650 IF( CONC(IMJM),EQ,0.) GO TO 420
651 400 CONTINUE
652 IF( CONC(IJM),EQ,0.) GO TO 420
653 C
654 C      ALL NEIGHBORS ARE FIREBALL CELLS. DIVIDE UP THE MOMENTUM,
655 C
656 XMASS=RO(IJ)+RO(IJM)
657 UG(IJ)=U(IJ)+U(IJM)
658 VG(IJ)=V(IJ)+V(IJM)
659 IF(I,EQ,1) GO TO 410
660 IMJ=IJ+NQ
661 IMJM=IJM=NQ
662 XMASS=XMASS+RO(IMJ)+RO(IMJM)
663 UG(IJ)=UG(IJ)+U(IMJ)+U(IMJM)
664 VG(IJ)=VG(IJ)+V(IMJ)+V(IMJM)
665 410 CONTINUE
666 UG(IJ)=UG(IJ)/XMASS
667 VG(IJ)=VG(IJ)/XMASS
668 420 CONTINUE
669 IJ=IJ+NQ
670 IJP=IJP+NQ
671 IJM=IJM+NQ
672 430 CONTINUE
673 CALL LOOP
674 440 CONTINUE
675 CALL DONE
676 C
677 C      NOW WE HAVE VERTEX VELOCITIES IN UG,VG, STORE THEM IN U,V
678 C      AND CONVERT MASSES TO DENSITIES AND INTERNAL ENERGIES
679 C      TO SPECIFIC INTERNAL ENERGIES,
680 C
681 CALL START
682 DO 470 J=2,JP1
683 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      VOLW,S*(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
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
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
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
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
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(4,6HM8HMKR,34,34HMESH NAMELIST INITIALIZATION ERROR)
743
C      540 FORMAT(1H ,34HFIREBALL INPUT WILL BE FROM FILE 3)
744      550 FORMAT(1H /1H ,28HALL FIREBALL INPUT DATA READ/1H )
745      560 FORMAT(1H ,5HCELL ,2I5,32H HAS A RELATIVE VOLUME ERROR OF ,1PE12.5
746          1,1H.,,2X,46HIT WILL BE TREATED AS A FIREBALL BOUNDARY CELL)
747      570 FORMAT(1H,22HMESHMKR FIREBALL INPUT//1H ,13X1HU,19X1HE,18X3HRHO,
748          1,17X1HX,19X1HP,17X4HPRHO/1H )
749      580 FORMAT(5E15,0)
750      590 FORMAT(1H ,1P6E19,7)
751
752      END

```

```

1      SUBROUTINE MTHANE
2
C      ROUTINE TO CALCULATE THE FACTOR GAMMA=1 FOR METHANE
3
C      UNITS ARE ALL CGS
4
C      ORIGINALLY OBTAINED FROM THE AIR FORCE WEAPONS LAB
5      MODIFIED BY J.L.NORTON,LASL T-3,1974
6
C
7      COMMON/EQNST/RHO,E,GMONE
8
9      DIMENSION A(8)
10
11     DATA A/3,21782E-1,-1.56848E-4,-8.78899E-2,1.25271E-3,1.46612E-2,
12        1,-1.72460E-6,-7.72322E-4,-8.63413E-6/
13        U#E*1,E=10
14        R#ALOG(RHO)
15        GMONE=A(1)+R*(A(2)+U*(A(4)+U*(A(6)+A(8)*U)))+U*(A(3)+U*(A(5)+A(7)*
16        1,U))
17        RETURN
18
19      END

```

```

1      SUBROUTINE NAMLST(TABLE,IF,IEFLAG)
2
C      ROUTINE TO REPLACE COMPILER/SYSTEM DEPENDENT NAMELIST INPUT.
3      HERE,AN ATTEMPT HAS BEEN MADE TO ELIMINATE AS MANY
4      COMPILER- AND/OR SYSTEM-DEPENDENT FEATURES AS POSSIBLE. THOSE
5      STILL REMAINING ARE BOOLEAN ALGEBRA,60 BIT WORD ASSUMPTION,
6      INLINE SHIFT FUNCTION (CAN BE REPLACED BY ASSEMBLY-LANGUAGE
7      FUNCTION IF IT IS TYPED INTEGER),AND TWO ASSEMBLY-LANGUAGE
8      ROUTINES TO FETCH AND STORE WORDS FROM AND INTO ABSOLUTE
9      MEMORY LOCATIONS.
10
11     TABLE = NAMELIST DATA TABLE
12

```

```

13 C IF = INPUT FILESET
14 C IEFLAG = ERROR FLAG
15 C   • 0 = NO ERROR
16 C   • -1 = END-OF-FILE ON FILESET IF
17 C     • N (.NE.0 OR -1) = ERROR EXIT
18 C       (ERROR MESSAGE WILL BE PRINTED BY MESSOT)
19 C       (IF .LT.0,EOF ALSO OCCURRED)
20 C
21 C LIMITATIONS = CURRENTLY, ONLY SMALL CORE VARIABLES MAY BE READ,
22 C ARRAYS OF UP TO FOUR SUBSCRIPTS CAN BE INPUT ALTHOUGH MANY
23 C COMPILERS CAN HANDLE A MAXIMUM OF ONLY THREE SUBSCRIPTS.
24 C
25 C INPUT RULES = DATA MAY BE INPUT IN ANY OF THREE GENERAL FORMS ==
26 C   V=D1
27 C     A=D1,D2,...,DN
28 C     A(SUB)=D1,D2,...,DN
29 C WHERE V IS A NON-SUBSCRIPTED VARIABLE,
30 C   A IS AN ARRAY,
31 C     SUB REPRESENTS FROM 1 TO 4 SUBSCRIPTS (INTEGER CONSTANT)
32 C     AND DN REPRESENTS A DATA ELEMENT,
33 C THE DATA ELEMENTS MAY BE OF A NUMBER OF TYPES ==
34 C   INTEGER = BASE 10 (EXAMPLE,=13569)
35 C     • BASE 8 (FOLLOW CONSTANT WITH B) (77658)
36 C   REAL   • FIXED (13.25)
37 C     • FLOATING (3.265E29)
38 C   COMPLEX = (-13.59,3.14E-7)
39 C   DOUBLE  = 3.14159265358979323846D3
40 C   LOGICAL = .TRUE.,.FALSE.,.T.,.F.
41 C   HOLLERITH = LEFT=JUSTIFIED,BLANK=FILL (5HTHING)
42 C     • LEFT=JUSTIFIED,ZERO=FILL (3LOUT)
43 C     • RIGHT=JUSTIFIED,ZERO=FILL (4RWORD)
44 C MULTIPLIERS ARE ALLOWED, BUT ONLY FOR ONE ELEMENT AT A TIME,
45 C FOR EXAMPLE, 5*23,6*15,3,21*27,3E7,5*4HTEST ARE ALL
46 C LEGAL, 6*(13.,21,7) WOULD WORK, BUT ONLY BECAUSE THE
47 C QUANTITY INSIDE THE PARENTHESES IS INTERPRETED AS A
48 C SINGLE COMPLEX CONSTANT. 22*(1,2,3,4,5,6) WOULD BE
49 C FLAGGED AS AN ERROR. 5*13HABCDEFHIJKLMNOP IS ALSO ILLEGAL
50 C AS THE CONSTANT IS MORE THAN ONE WORD LONG,
51 C INTEGERS WITH MORE THAN 14 DIGITS AND REAL CONSTANTS OF
52 C MORE THAN 14 DIGITS NOT INDICATED AS DOUBLE PRECISION
53 C ARE ILLEGAL. FOR EXAMPLE, PI=3.1415926535897932 IS WRONG
54 C BUT PI=3,1415926535897932D IS O.K.
55 C
56 C *****
57 C NOTE CAREFULLY = NO CONVERSIONS OF DATA TYPE ARE MADE, THIS GIVES
58 C ONE THE CAPABILITY OF STORING REAL CONSTANTS INTO INTEGER
59 C VARIABLES, FOR EXAMPLE, HOWEVER, IF ONE FORGETS TO PUNCH A
60 C DECIMAL POINT (X=.5), THE CONSTANT WILL BE STORED AS AN
61 C INTEGER.
62 C *****
63 C
64 C OTHER CONVENTIONS AND CAPABILITIES ARE AS FOLLOWS ==
65 C   ANY CARD WITH CS IN CC1=2 WILL BE PRINTED OUT AND IGNORED,
66 C   ANY CARD WITH PS IN CC1=2 WILL TURN ON A PRINT SWITCH AND
67 C     EACH CARD OF THE NAMELIST INPUT RECORD (UP TO THE NEXT S)
68 C     WILL BE PRINTED BEFORE PROCESSING,
69 C   AN ISOLATED = IS ILLEGAL, =EF=1,E0, =D=1,DB, =.TRUE.,=.FALSE.

```

```

70      C      AND -.FALSE., .TRUE., .-BZ=QB, A MINUS SIGN IN FRONT OF AN
71      C      OCTAL CONSTANT CAUSES THE CONSTANT TO BE COMPLEMENTED.
72      C      A MINUS SIGN IN FRONT OF A HOLLERITH CONSTANT IS IGNORED.
73      C      AN R HOLLERITH CONSTANT CANNOT BE MORE THAN 10 CHARACTERS
74      C      LONG.
75      C      THE TERMINATION S MUST NOT OCCUR IN CC1 OR 2. IF IT OCCURS
76      C      IN CC1, IT WILL BE IGNORED, AN ERROR WILL BE RETURNED IF
77      C      IT IS IN CC2.
78      C      A NAMELIST RECORD MAY BE OF ANY LENGTH, BLANKS ARE SIGNIFICANT
79      C      AND MAY OCCUR ONLY AROUND NON-NUMERIC CHARACTERS OTHER
80      C      THAN EXPONENTS, FOR EXAMPLE, LEGAL BLANKS ARE X = 5.,
81      C      Y = -10.3, L = .TRUE., C = ( 5., 6.), I = 13,
82      C      ILLEGAL BLANKS WOULD INCLUDE X=5, E 6, L=, T., Y=, 5 .
83      C      THE NAMELIST RECORD IS TREATED AS ONE LONG STRING OF
84      C      CHARACTERS, ALL 80 CHARACTERS OF EACH CARD BEING SCANNED,
85      C      EXCEPT THAT CC2 OF EACH CARD MAY NOT BE A S UNLESS IT IS
86      C      PART OF A HOLLERITH FIELD.
87      C      BECAUSE OF THE UNLIMITED LENGTH OF A NAMELIST RECORD, A
88      C      HOLLERITH CONSTANT MAY BE OF UNLIMITED LENGTH.
89      C      BECAUSE TYPE IS NOT CHECKED, STORING A DOUBLE-PRECISION
90      C      CONSTANT INTO A SINGLE-PRECISION VARIABLE WILL CLOBBER
91      C      THE FOLLOWING LOCATION. (X=5.D WILL STORE A ZERO INTO
92      C      THE LOCATION FOLLOWING X ASSUMING X IS NOT DOUBLE OR
93      C      COMPLEX), A SIMILAR WARNING CAN BE GIVEN FOR X=(1.,2.).
94
95      C*****EXAMPLE OF USAGE *****
96      C
97      C
98      C      SUPPOSE ONE WISHED TO REPLACE THE STATEMENTS
99
100     C      DIMENSION A(20),B(5,3),X(5,10,15)
101     C      NAMELIST/CARDN/I,J,K,A,B,X
102     C      READ(5,CARDN)
103
104     C      WITH SYSTEM-INDEPENDENT INPUT, FURTHERMORE, SUPPOSE ONE WISHED
105     C      TO HAVE PS AND CS CARDS COME OUT ON BOTH PAPER AND FILM
106     C      AND ERROR MESSAGES COME OUT ON FILE 59, THEN ONE WOULD
107     C      NEED THE FOLLOWING ~ ~
108
109     C      DIMENSION A(20),B(5,3),X(5,10,15),ITAB(2),TABLE(3,7)
110
111     C      SET UP FILE TABLE FOR MESSAGES
112
113     C      DATA ITAB/3LOUT,4LFILM/,JTAB/59/
114     C      CALL NAMPRT(2,ITAB)
115     C      CALL ERRPRT(1,JTAB)
116
117     C      DEFINE THE NAMELIST TABLE; IF THERE ARE TO BE N UNIQUE
118     C      INPUT VARIABLE NAMES, THEN 3*(N+1) TABLE LOCATIONS
119     C      ARE NEEDED. THUS, FOR SIX VARIABLES, 3*7 LOCATIONS
120     C      MUST BE SET ASIDE. THE THIRD ARGUMENT IS THE
121     C      SECOND SUBSCRIPT OF TABLE. IF ANY ERRORS OCCUR IN
122     C      SUBSEQUENT CALLS TO TABSET, TRANSFER WILL OCCUR TO
123     C      THE STATEMENT NO. ASSIGNED TO IERRT. THIS ELIMINATES
124     C      ERROR CHECKING AFTER EACH CALL TO TABSET. IEFLAG CAN
125     C      THEN BE EXAMINED TO DETERMINE WHAT TYPE OF ERROR
126     C      OCCURRED.

```

```

127 C      C      ASSIGN 999 TO IERRT
128 C      C      CALL TABDEF(TABLE,5HCARDN,7,IERRT)
129 C      C      C      ENTER EACH VARIABLE INTO THE TABLE. IF THERE IS SOME
130 C      C      C      ERROR, IEFLAG IS RETURNED NON-ZERO.
131 C      C      C      CALL TABSET(TABLE,1HI,I,IEFLAG,0,0,0,0)
132 C      C      C      (THE 0 DENOTES A NON-SUBSCRIPTED VARIABLE)
133 C      C      C      CALL TABSET(TABLE,1HJ,J,IEFLAG,0,0,0,0)
134 C      C      C      CALL TABSET(TABLE,1HK,K,IEFLAG,0,0,0,0)
135 C      C      C      CALL TABSET(TABLE,1HA,A,IEFLAG,1,0,0,0)
136 C      C      C      (THE 1 DENOTES A SUBSCRIPTED VARIABLE BUT THE SUBSCRIPT
137 C      C      C      NEED NOT BE GIVEN FOR A SINGLY-SUBSCRIPTED VARIABLE)
138 C      C      C      CALL TABSET(TABLE,1HB,B,IEFLAG,2,5,0,0)
139 C      C      C      (THE VARIABLE IS DOUBLY-SUBSCRIPTED BUT ONLY THE FIRST
140 C      C      C      SUBSCRIPT NEED BE GIVEN)
141 C      C      C      CALL TABSET(TABLE,1HX,X,IEFLAG,3,5,10,0)
142 C      C      C      (THE VARIABLE IS TRIPLY-SUBSCRIPTED BUT ONLY THE FIRST
143 C      C      C      TWO SUBSCRIPTS NEED BE GIVEN)
144 C      C      C      NOTE THAT TABSET IS CALLED WITH THE FULL NO. OF ARGUMENTS
145 C      C      C      WHETHER THEY ARE USED OR NOT. THIS IS NOT STRICTLY
146 C      C      C      NECESSARY ON MOST COMPILERS. CHECK YOUR LOCAL
147 C      C      C      CONVENTIONS.
148 C      C      C      READ IN A NAMELIST CARD. IF IEFLAG RETURNS 0, ALL WAS OK.
149 C      C      C      IF IEFLAG RETURNS -1, A NORMAL EOF WAS ENCOUNTERED,
150 C      C      C      FOR ALL OTHER VALUES OF IEFLAG, AN ERROR OCCURRED.
151 C      C      C      IF IEFLAG.LT.0, AN EOF ALSO OCCURRED,
152 C      C      C      CALL NAMLST(TABLE,5,IEFLAG)
153 C      C      C      (5 IS THE INPUT FILE NO.)
154 C      C      C      IF(IEFLAG.EQ.0) GO TO 10
155 C      C      C      IF(IEFLAG.EQ.(-1)) GO TO 20
156 C      C      C      A NAMELIST ERROR OCCURRED
157 C      C      C      STOP 7
158 C      C      C      A NAMELIST TABLE INITIALIZATION ERROR OCCURRED
159 C      C      C      999 CONTINUE
160 C      C      C      * * *
161 C      C      C      AN EOF OCCURRED
162 C      C      C      20 CONTINUE
163 C      C      C

```

```

184      C      *
185      C      C      NORMAL EXIT
186      C      C
187      C      10 CONTINUE
188      C      *
189      C      ****
190      C      WRITTEN BY J.L.NORTON, LASL T=3, 1974
191      C
192      C      THIS VERSION RUNS ON A CDC 6600 OR 7600 USING THE RUN COMPILER
193      C
194      C
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      I CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
202      COMMON/ERRORC/IERRT
203      COMMON/CSHIFT/XLEFT,XRIGHT,XDUM
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
215      C      LIST = IF TRUE, EACH NAMELIST CARD INPUT WILL BE LISTED
216      C
217      C      LIST=.FALSE.,
218      C
219      C      BEGINNING OF CODE FOR PROCESSING A NAMELIST RECORD
220      C
221      10 CONTINUE
222
223      C      RESET THE ERROR FLAG
224
225      C      IEFLAG=0
226
227      C      READ A CARD
228
229      C      CALL READIT(IF,IEFLAG)
230
231      C      IF AN END-OF-FILE WAS ENCOUNTERED, QUIT
232
233      C      IF(IEFLAG,LT,0) RETURN
234
235      C      ISUB IS THE POINTER TO THE COLUMN BEING PROCESSED OF THE CARD
236      C      LAST READ. BEGIN LOOKING AT CC2.
237
238      C      ISUB=2
239
240      C      IF CC2 IS NOT A DOLLAR, IGNORE THE CARD AND GO BACK AND READ

```

```

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

```

```

298 C      YES, IF NON-ALPHABETIC,FATAL ERROR.
299 C
300 C      IF(ITEMP.GE.IAF.AND.ITEMP.LE.IAL) GO TO 60
301 C      CALL MESSOT(23)
302 C      IEFLAG=23
303 C      RETURN
304 C      50 CONTINUE
305 C
306 C      IF ANY CHARACTER IS NOT A LETTER OR A NUMBER,FATAL ERROR.
307 C
308 C      IF((ITEMP.GE.INF.AND.ITEMP.LE.INL).OR.(ITEMP.GE.IAF.AND.ITEMP.LE.
309 C      IAL)) GO TO 60
310 C      CALL MESSOT(24)
311 C      IEFLAG=24
312 C      RETURN
313 C      60 CONTINUE
314 C
315 C      STORE THE CHARACTER AND GO LOOK AT THE NEXT
316 C
317 C      IDSTR(IVNUM)=ITEST
318 C      GO TO 30
319 C
320 C      A BLANK CHARACTER HAS BEEN FOUND
321 C
322 C      70 CONTINUE
323 C
324 C      IF IT IS THE FIRST CHARACTER AFTER THE $,FATAL ERROR
325 C
326 C      IF(IVNUM.GT.0) GO TO 80
327 C      CALL MESSOT(11)
328 C      IEFLAG=11
329 C      RETURN
330 C      80 CONTINUE
331 C
332 C      GO PUT THE STRING INTO ONE WORD (INAME),LEFT=JUSTIFIED,BLANK FILL
333 C
334 C      CALL MASH(IDSTR,IVNUM,INAME,IEFLAG)
335 C      IF(IEFLAG.NE.0) RETURN
336 C
337 C      SEE IF IT MATCHES THE NAME IN THE NAMELIST TABLE. IF NOT,GO BACK
338 C      AND READ THE NEXT CARD.
339 C
340 C      IF(TABLE(1,1).NE.INAME) GO TO 10
341 C
342 C      NAME MATCHES. INE IS THE NUMBER OF VARIABLES REPRESENTED IN THE
343 C      TABLE #1, DELIM#,FALSE, INDICATES THAT A DELIMITER (, OR $)
344 C      IS NOT EXPECTED AS THE NEXT CHARACTER, FIRST#=TRUE, INDICATES
345 C      THAT A VARIABLE NAME MUST BE FOUND BEFORE A CONSTANT.
346 C
347 C      INE=TABLE(3,1)+1
348 C      DELIM#,FALSE,
349 C      FIRST#=TRUE,
350 C
351 C      LIST THE CARD IF THE FLAG IS ON
352 C
353 C      IF(LIST) CALL MESSOT(1)
354 C      GO TO 100

```

```

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

```

```

412 C      NO. IF A NAME IS EXPECTED, FATAL ERROR,
413 C
414 C      IF(FIRST) GO TO 120
415 C
416 C      A NAME IS NOT NECESSARILY EXPECTED. LEGAL CHARACTERS ARE NUMBERS,
417 C      +,-,, OR (. IF NONE OF THESE, FATAL ERROR. IF ONE OF THESE,
418 C      PROCEED TO SECTION WHICH DECODES CONSTANTS.
419 C
420 C      IF(ITEMP,GE,INF,AND,ITEMP,LE,INL) GO TO 400
421 C      IF(ITEMP,EQ,1H+,OR,ITEMP,EQ,1H-,OR,ITEMP,EQ,1H,,OR,ITEMP,EQ,1H())
422 C      1 GO TO 400
423 C
424 C      120 CONTINUE
425 C      CALL MESSOT(13)
426 C      IEFLAG#13
427 C      RETURN
428 C
429 C      ****
430 C      SUBLOOP TO DECODE VARIABLE NAMES
431 C      ****
432 C
433 C      130 CONTINUE
434 C
435 C      STORE THE FIRST CHARACTER
436 C
437 C      IVNUM=1
438 C      IDSTR(1)*ITEMP
439 C
440 C      ****
441 C      SUBLOOP TO GET CHARACTERS OF VARIABLE NAME
442 C      ****
443 C
444 C      140 CONTINUE
445 C
446 C      GET THE NEXT CHARACTER AND CHECK FOR ERRORS
447 C
448 C      CALL NXTCOL(IF,IEFLAG)
449 C      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 C      ITEMP=AND SHIFT(ITEMP,-54),778
455 C      IF((ITEMP,GE,IAF,AND,ITEMP,LE,IAL),OR,(ITEMP,GE,INF,AND,ITEMP,LE,
456 C      1 INL)) GO TO 150
457 C      GO TO 170
458 C
459 C      150 CONTINUE
460 C
461 C      CHARACTER IS ALPHANUMERIC. SEE IF IT IS NO. 7. IF SO, FATAL ERROR.
462 C      IF(IVNUM,GE,7) GO TO 160
463 C
464 C      NOT CHARACTER 7. STORE AND GO ON TO THE NEXT.
465 C
466 C      IVNUM=IVNUM+1
467 C      IDSTR(IVNUM)*ITEMP
468 C      GO TO 140

```

```

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

```

```

526 C -----
527 C DETERMINE ANY SUBSCRIPTING INFORMATION
528 C -----
529 C
530 210 CONTINUE
531 C
532 C CHECK THE FIRST CHARACTER AFTER THE VARIABLE NAME, IF IT IS AN
533 C EQUALS, NO SUBSCRIPTING INFORMATION, PROCEED TO CONSTANT SCAN.
534 C OTHER LEGAL CHARACTERS ARE BLANK AND LEFT PAREN, IF NEITHER
535 C OF THESE, FATAL ERROR.
536 C
537 IF(IATEST, EQ, 1H ) GO TO 220
538 IF(IATEST, EQ, 1H() GO TO 230
539 IF(IATEST, EQ, 1H=) GO TO 400
540 CALL MESS80T(5)
541 IEFLAG=5
542 RETURN
543 C
544 C CHARACTER WAS BLANK, FETCH NEXT COLUMN AND CONTINUE.
545 C
546 220 CONTINUE
547 CALL NXTCOL(IF,IEFLAG)
548 IF(IEFLAG,NE,0) RETURN
549 GO TO 210
550 C
551 C CHARACTER WAS LEFT PAREN, BEGIN SUBSCRIPT DECODING.
552 C
553 230 CONTINUE
554 C
555 C INITIALIZE, IVSUM IS THE CURRENT VALUE OF THE NUMBER BEING DECODED
556 C AND IVSIGN IS ITS SIGN (+1 OR -1).
557 C
558 DIMM,TRUE,
559 NSUB=1
560 IVNUM=0
561 IVSUM=0
562 IVSIGN=1
563 DELIMM, FALSE.
564 C
565 C NOSGN IS THE NO. OF SIGNS ENCOUNTERED IN THE FIELD
566 C
567 NOSGN=0
568 C
569 C +++ SUBLOOP TO DECODE A SUBSCRIPT
570 C
571 240 CONTINUE
572 C
573 C GET THE NEXT CHARACTER
574 C
575 CALL NXTCOL(IF,IEFLAG)
576 IF(IEFLAG,NE,0) RETURN
577 C
578 C SEE IF IT IS NUMERIC
579 C
580 ITEMPI=AND(SHIFT(IATEST,-54),778)
581 IF(ITEMPI,GE,INF,AND,ITEMPI,LE,INL) GO TO 250
582 C

```

```

583    C      NO. CHECK FOR BLANK,
584    C      IF(ITEST,EQ,1H ) GO TO 310
585    C
586    C      NO. CHECK FOR COMMA,
587    C      IF(ITEST,EQ,1H,) GO TO 340
588    C
589    C      NO. CHECK FOR RIGHT PAREN,
590    C      IF(ITEST,EQ,1H)) GO TO 360
591    C
592    C      CHECK FOR SIGN OF SUBSCRIPT
593    C      IF(ITEST,EQ,1H+,OR,ITEST,EQ,1H-) GO TO 290
594    C
595    C      NONE OF THESE, FATAL ERROR,
596    C
597    C      CALL MESSOT(6)
598    C      IEFLAG=6
599    C      RETURN
600    C
601    C      CHARACTER WAS NUMERIC, SEE IF A DELIMITER WAS EXPECTED, IF IT WAS,
602    C      FATAL ERROR,
603    C
604    C      250 CONTINUE
605    C      IF(,NOT,DELIM) GO TO 270
606    C
607    C      260 CONTINUE
608    C      CALL MESSOT(7)
609    C      IEFLAG=7
610    C      RETURN
611    C
612    C      270 CONTINUE
613    C
614    C      SEE IF THE SUBSCRIPT IS TOO LONG, IF MORE THAN 5 DIGITS,FATAL
615    C      ERROR,
616    C
617    C      IF(IVNUM,LT,5) GO TO 280
618    C      CALL MESSOT(43)
619    C      IEFLAG=43
620    C      RETURN
621    C
622    C      280 CONTINUE
623    C
624    C      SUBSCRIPT IS OK SO FAR, UPDATE ITS VALUE AND CONTINUE,
625    C
626    C      IVNUM=IVNUM+1
627    C      IVSUM=IVSUM*10+ITEMP-INF
628    C      GO TO 240
629    C
630    C      290 CONTINUE
631    C
632    C      A SIGN HAS BEEN FOUND, MAKE SURE THERE IS ONLY ONE,
633    C
634    C      IF(NOSGN,EQ,0) GO TO 300
635    C      CALL MESSOT(44)
636    C      IEFLAG=44
637    C      RETURN
638    C
639    C      300 CONTINUE

```

```

640 C      SIGN IS OK, UP THE COUNT,SET THE SIGN VARIABLE,AND CONTINUE SCAN,
641 C
642 C      NOSGN#1
643 C      IF(ITEST.EQ.1H-) IVSIGN#=1
644 C      GO TO 240
645 C      310 CONTINUE
646 C
647 C      CHARACTER IS A BLANK, IF NO NUMBERS HAVE BEEN FOUND YET,GO ON TO
648 C      THE NEXT CHARACTER.
649 C
650 C      IF(IVNUM.EQ.0) GO TO 240
651 C
652 C      BLANK CONSIDERED FIELD TERMINATOR, TURN ON THE DELIMITER SWITCH,
653 C
654 C      DELIM#=TRUE,
655 C      320 CONTINUE
656 C
657 C      SUBSCRIPT DECODED, MAKE SURE THERE ARE NO MORE THAN 4,
658 C
659 C      IF(NSUB.GT.4) GO TO 330
660 C
661 C      SUBSCRIPT IS OK, STORE THE SIGNED VALUE IN THE SUBSCRIPT ARRAY,
662 C
663 C      NSUBV(NSUB)=ISIGN(IVSUM,IVSIGN)
664 C
665 C      REINITIALIZE AND GO SEARCH FOR NEXT SUBSCRIPT
666 C
667 C      IVSUM=0
668 C      IVNUM=0
669 C      IVSIGN#1
670 C      NOSGN#0
671 C      NSUB=NSUB+1
672 C      GO TO 240
673 C
674 C      MORE THAN FOUR SUBSCRIPTS, FATAL ERROR.
675 C
676 C      330 CONTINUE
677 C      CALL MESSOT(14)
678 C      IEFLAG#14
679 C      RETURN
680 C      340 CONTINUE
681 C
682 C      COMMA ENCOUNTERED, SEE IF A DELIMITER IS EXPECTED.
683 C
684 C      IF(DELIM) GO TO 350
685 C
686 C      NO. SEE IF A FIELD HAS BEGUN, IF NOT,FATAL ERROR, IF SO,CONSIDER
687 C      FIELD TERMINATED AND GO FINISH PROCESSING FOR THIS SUBSCRIPT,
688 C
689 C      IF(IVNUM.NE.0) GO TO 320
690 C      CALL MESSOT(8)
691 C      IEFLAG#8
692 C      RETURN
693 C
694 C      DELIMITER EXPECTED AND FOUND, TURN OFF FLAG AND CONTINUE SCAN,
695 C
696 C      350 CONTINUE

```

```

697      DELIM=.FALSE.
698      GO TO 240
699
700      360 CONTINUE
701      C      RIGHT PAREN FOUND, TERMINATE SUBSCRIPT PROCESSING, CHECK IF PAREN
702      C      IS FIELD TERMINATOR.
703      C
704      C      IF(IVNUM.NE.0) GO TO 370
705      C
706      C      NO, CHECK FOR ISOLATED SIGN, IF SO, FATAL ERROR.
707      C
708      C      IF(N08GN.EQ.0) GO TO 380
709      C      CALL MESSOT(30)
710      C      IEFLAG=N30
711      C      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      C      NSUBV(NSUB)=ISIGN(IVSUM,IVSIGN)
718      C      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      C      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      C      CALL NXTCOL(IF,IEFLAG)
736      C      IF(IEFLAG.NE.0) RETURN
737      C      IF(ITEST.EQ.1H) GO TO 390
738      C      IF(ITEST.EQ.1H=) GO TO 400
739      C      CALL MESSOT(9)
740      C      IEFLAG=N9
741      C      RETURN
742
743      C      *****-*****-*****-*****-*****-*****-*****-*****-*****-*****-*****-*****-
744      C      SECTION TO DECODE CONSTANTS
745      C      *****-*****-*****-*****-*****-*****-*****-*****-*****-*****-*****-*****-
746      C
747      400 CONTINUE
748      C
749      C      REBET ALL FLAGS
750
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

```

```

754 C          BEEN FOUND
755 C      CPPND = .TRUE. IF THE CLOSING PARENTHESIS OF A COMPLEX CONSTANT
756 C          HAS BEEN FOUND
757 C      DB     = STORAGE FOR DOUBLE PRECISION CONSTANT
758 C      MULT  = .TRUE. IF A MULTIPLIER HAS BEEN FOUND
759 C      DELIM = .TRUE. IF A DELIMITER IS EXPECTED
760 C
761 C      CMPLEX#, FALSE,
762 C      CDONE#, FALSE,
763 C      CPPND#, FALSE,
764 C      DB#0,
765 C      MULT#, FALSE,
766 C      DELIM#, FALSE,
767 C
768 C      ONE RETURNS TO HERE IF THE FIRST NO. OF A COMPLEX CONSTANT HAS
769 C          BEEN PROCESSED AND STORED
770 C
771 C      410 CONTINUE
772 C
773 C      HOLLER = .TRUE. IF A HOLLERITH CONSTANT IS BEING PROCESSED
774 C      IC    = CHARACTER COUNT OF CONSTANT STRING
775 C      DOBLE = .TRUE. IF A DOBLE PRECISION CONSTANT IS BEING PROCESSED
776 C      FIX   = .TRUE. IF A FIXED POINT CONSTANT IS BEING DECODED
777 C      IVSUM = CURRENT VALUE OF FIXED POINT FIELD
778 C      IVNUM = CURRENT NO. OF DIGITS IN THE FIXED POINT FIELD
779 C      EXPN  = .TRUE. IF AN EXPONENT IS BEING DECODED
780 C      XLEFT = VALUE OF FLOATING POINT CONSTANT TO LEFT OF DECIMAL POINT
781 C      XRIGHT = VALUE OF FLOATING POINT CONSTANT TO RIGHT OF DECIMAL
782 C      NOSGN = NO. OF + OR - SIGNS FOUND IN CONSTANT SO FAR
783 C      DONE   = .TRUE. IF A DOLLAR SIGN HAS BEEN FOUND
784 C      IVSIGN = VALUE OF THE LAST SIGN FOUND (+1 OR -1)
785 C      PERIOD = .TRUE. IF THE LAST CHARACTER FOUND WAS AN ISOLATED PERIOD
786 C
787 C      HOLLER#, FALSE,
788 C      IC#0
789 C      DOBLE#, FALSE,
790 C      FIX#, TRUE,
791 C      IVSUM#0
792 C      IVNUM#0
793 C      EXPN#, FALSE,
794 C      XLEFT#0,
795 C      XRIGHT#0,
796 C      NOSGN#0
797 C      DONE#, FALSE,
798 C      IVSIGN#1
799 C      PERIOD#, FALSE,
800 C
801 C      IF THIS IS THE FIRST CONSTANT TO BE PROCESSED AFTER AN = SIGN
802 C          (FIRST#, .TRUE.) OR A COMPLEX CONSTANT IS BEING DECODED,
803 C          ONE MUST FETCH THE NEXT CHARACTER, OTHERWISE, IT HAS PREVIOUSLY
804 C          BEEN READ,
805 C
806 C      IF(,NOT,FIRST,AND,,NOT,CMPLEX) GO TO 430
807 C      420 CONTINUE
808 C
809 C      GET THE NEXT CHARACTER AND CHECK FOR ERRORS
810 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 C      IF(I TEST,EQ,1H ) GO TO 470
822 C
823 C      NO, CHECK FOR A NUMBER,
824 C
825 C      ITEMP=AND(SHIFT(I TEST,-54),77B)
826 C      IF(ITEMP,GE,1NF,AND,ITEMP,LE,INL) GO TO 450
827 C
828 C      NO, CHECK FOR A COMMA,
829 C
830 C      IF(I TEST,EQ,1H,) GO TO 1050
831 C
832 C      NO, CHECK FOR A PERIOD,
833 C
834 C      IF(I TEST,EQ,1H,) GO TO 1140
835 C
836 C      NO, CHECK FOR A PLUS OR MINUS,
837 C
838 C      IF(I TEST,EQ,1H+) GO TO 990
839 C      IF(I TEST,EQ,1H-) GO TO 990
840 C
841 C      NO, CHECK FOR A DOLLAR SIGN,
842 C
843 C      IF(I TEST,EQ,1HS) GO TO 1080
844 C
845 C      NO, CHECK FOR A T OR F,
846 C
847 C      IF(I TEST,EQ,1HT) GO TO 1170
848 C      IF(I TEST,EQ,1HF) GO TO 1170
849 C
850 C      NO, CHECK FOR AN E,
851 C
852 C      IF(I TEST,EQ,1HE) GO TO 1100
853 C
854 C      NO, CHECK FOR AN ASTERISK,
855 C
856 C      IF(I TEST,EQ,1H*) GO TO 680
857 C
858 C      NO, CHECK FOR AN H,
859 C
860 C      IF(I TEST,EQ,1HH) GO TO 770
861 C
862 C      NO, CHECK FOR A B,
863 C
864 C      IF(I TEST,EQ,1HB) GO TO 930
865 C
866 C      NO, CHECK FOR AN L OR AN R,
867 C

```

```

868      IF(ITEMT.EQ.1HL) GO TO 770
869      IF(ITEMT.EQ.1HR) GO TO 770
870
871      C NO. CHECK FOR A D.
872      C IF(ITEMT.EQ.1HD) GO TO 1130
873
874      C NO. CHECK FOR A LEFT PARENTHESIS.
875      C IF(ITEMT.EQ.1H()) GO TO 720
876
877      C NO. CHECK FOR A RIGHT PARENTHESIS.
878      C IF(ITEMT.EQ.1H)) GO TO 750
879
880      C LEGAL CHARACTER NOT FOUND, FATAL ERROR.
881
882      C
883      C 440 CONTINUE
884      C CALL MESS80T(12)
885      C IEFLAG=12
886      C RETURN
887
888      C
889      C *****CHARACTER WAS A NUMBER*****
890      C
891      C CHARACTER WAS A NUMBER
892      C
893
894      C 450 CONTINUE
895
896      C SEE IF A DELIMITER WAS EXPECTED, IF SO,FATAL ERROR.
897
898      C IF(DELIM) GO TO 260
899
900      C RESET ISOLATED PERIOD FLAG
901
902      C PERIOD=.FALSE.
903
904      C UPDATE THE VALUE OF THE NUMERIC FIELD
905
906      C IVSUM=IVSUM*10+ITEMP-INF
907
908      C UPDATE THE DIGIT COUNT
909
910      C IVNUM=IVNUM+1
911
912      C UPDATE THE CHARACTER COUNT
913
914      C IC=IC+1
915
916      C SEE IF THIS IS THE THIRTIETH CHARACTER
917
918      C IF(IC.LT.30) GO TO 460
919
920      C YES, FATAL ERROR.
921
922      C GO TO 1160
923
924      C NO, STORE THE CHARACTER AND GO ON TO THE NEXT,

```

```

925      C
926      460 CONTINUE
927      IDSTR(IC)=ITEST
928      GO TO 420
929      C
930      C-----CHARACTER WAS A BLANK-----C
931      C
932      C-----CHARACTER WAS A BLANK-----C
933      C
934      470 CONTINUE
935      C
936      C     IF THE BEGINNING OF A CONSTANT HAS NOT YET BEEN FOUND, GO ON TO
937      C     THE NEXT CHARACTER
938      C
939      C     IF(FIX,AND,IVNUM,EQ,0) GO TO 420
940      C
941      C     NOT THE BEGINNING OF A CONSTANT, BLANK IS CONSIDERED FIELD
942      C     TERMINATOR. TURN ON THE FLAG WHICH SAYS STORE THE CONSTANT.
943      C
944      C     SAVER,TRUE.
945      C
946      C     SEE IF CURRENT FIELD IS FIXED OR FLOATING
947      C
948      C     IF(,NOT,FIX) GO TO 490
949      C
950      C     FIXED, DO NOT STORE NOW BUT TURN ON DELIMITER FLAG AND CONTINUE
951      C     CHARACTER SCAN.
952      C
953      C     DELIMM,TRUE.
954      C     GO TO 420
955      C
956      C     ENTRY POINT FOR STORING FLOATING POINT NO. IF FIELD IS TERMINATED
957      C     BY A , OR A S
958      C
959      480 CONTINUE
960      C     SAVER,TRUE.
961      490 CONTINUE
962      C
963      C     CHECK FOR ISOLATED PERIOD
964      C
965      C     IF(,NOT,PERIOD) GO TO 500
966      C
967      C     YES, FATAL ERROR.
968      C
969      C     CALL MESSOT(29)
970      C     IEFLAG#29
971      C     RETURN
972      500 CONTINUE
973      C
974      C     NO. SEE IF WE ARE PROCESSING AN EXPONENT.
975      C
976      C     IF(,NOT,EXPX) GO TO 530
977      C
978      C     YES, SEE IF IT IS ZERO.
979      C
980      C     IF(IVSUM,EQ,0) GO TO 630
981      C

```

```

982 C NO. GIVE IT THE PROPER SIGN.
983 C
984 C IVSUM=ISIGN(IVSUM,IVSIGN)
985 C
986 C SEE IF IT IS WITHIN BOUNDS
987 C
988 C IF(IVSUM.GT.IEMIN.AND.IVSUM.LT.IEMAX) GO TO 510
989 C
990 C NO, FATAL ERROR.
991 C
992 C CALL MES80T(26)
993 C IEFLAG=26
994 C RETURN
995 C 510 CONTINUE
996 C
997 C YES, ADJUST THE NO. BASED ON THE EXPONENT AND PROCEED ON
998 C TO THE GENERAL STORING CODE, USE INTERMEDIATE DOUBLE
999 C PRECISION IN ALL CASES.
1000 C
1001 C IF(DOUBLE) GO TO 520
1002 C XLEFT=XLEFT*DCONST**IVSUM
1003 C GO TO 630
1004 C 520 CONTINUE
1005 C DB=DB*DCONST**IVSUM
1006 C GO TO 630
1007 C
1008 C CODE TO PROCESS FLOATING POINT NO. WITH NO EXPONENT
1009 C
1010 C 530 CONTINUE
1011 C
1012 C CHECK FOR FIELD LONGER THAN 15 CHARACTERS BUT NOT DECLARED
1013 C DOUBLE PRECISION. IF SO,FATAL ERROR.
1014 C
1015 C IF(IC.GT.NDSP+1.AND.,NOT.DOUBLE) GO TO 600
1016 C
1017 C CHECK FOR DOUBLE PRECISION FIELD
1018 C
1019 C IF(.NOT.DOUBLE) GO TO 590
1020 C
1021 C YES, CHECK FOR FIELD LONGER THAN 29 CHARACTERS. IF SO,FATAL ERROR.
1022 C
1023 C IF(IC.GT.NDDP+1) GO TO 1160
1024 C
1025 C ALL O.K, INITIALIZE FOR EVALUATION. ICP IS THE NO. OF THE FIELD
1026 C CHARACTER CURRENTLY BEING PROCESSED.
1027 C
1028 C DB=0,
1029 C ICP#0
1030 C
1031 C #####+
1032 C LOOP TO EVALUATE DOUBLE PRECISION CONSTANTS
1033 C #####
1034 C
1035 C INCREMENT THE CHARACTER COUNT
1036 C
1037 C 540 ICP=ICP+1
1038 C

```

```

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

```

```

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

```

```

1153      C
1154          IVNUM=0
1155          IVSUM=0
1156          IVSIGN=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,CMPLX) 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(CPFND) 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          IF THE LAST CHARACTER FOUND WAS NOT A COMMA, TURN ON THE
1203          C      DELIMITER=EXPECTED FLAG
1204      C
1205          IF(IEST,NE,1H,) DELIM=.TRUE.,
1206      C
1207          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      C      IF(,NOT,DOBLE,AND,,NOT,CMPLEX) XSUM=XLEFT
1216      C
1217      C      STORE THE CONSTANT
1218      C
1219      C      CALL STORE(TABLE,IEFLAG)
1220      C
1221      C      IF ANY ERRORS, QUIT RIGHT HERE
1222      C
1223      C      IF(IEFLAG,NE,0) RETURN
1224      C
1225      C      IF THE CLOSING S HAS BEEN FOUND, WE ARE ALL DONE
1226      C
1227      C      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      C      IF(CMPLEX,AND,,NOT,CPFNO) 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      C      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      C      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      C      IF(FIX) GO TO 690
1254      C      CALL MESSOT(17)
1255      C      IEFLAG#17
1256      C      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      C      IF(,NOT,MULT) GO TO 700
1263      C      CALL MESSOT(18)
1264      C      IEFLAG#18
1265      C      RETURN
1266      700 CONTINUE

```

```

1267 C IF PROCESSING OF A COMPLEX NO. HAS BEGUN,FATAL ERROR
1268 C
1269 C
1270 C IF(,NOT,CMPLEX) GO TO 710
1271 C CALL MESSOT(19)
1272 C IEFLAG=19
1273 C RETURN
1274 C 710 CONTINUE
1275 C
1276 C ALL CONDITIONS O.K. TURN ON THE FLAG,SAVE THE MULTIPLIER,
1277 C REINITIALIZE,AND GO ON TO THE NEXT CHARACTER.
1278 C
1279 C MULT=.TRUE.
1280 C MULTSV=IVSUM
1281 C IVNUM=0
1282 C IVSUM=0
1283 C IVSIGN=1
1284 C DELIM=.FALSE.
1285 C GO TO 420
1286 C
1287 C *****
1288 C CHARACTER IS A LEFT PARENTHESIS
1289 C *****
1290 C
1291 C 720 CONTINUE
1292 C
1293 C IF A DELIMITER WAS EXPECTED,FATAL ERROR
1294 C
1295 C IF(DELIM) GO TO 260
1296 C
1297 C SEE IF ANY OTHER FIELD CHARACTERS HAVE BEEN READ. IF SO,FATAL
1298 C ERROR.
1299 C
1300 C IF(IVNUM,EQ,0) GO TO 730
1301 C CALL MESSOT(20)
1302 C IEFLAG=20
1303 C RETURN
1304 C 730 CONTINUE
1305 C
1306 C SEE IF A SIGN HAS BEEN ENCOUNTERED. IF SO,FATAL ERROR.
1307 C
1308 C IF(NOSGN,EQ,0) GO TO 740
1309 C CALL MESSOT(39)
1310 C IEFLAG=39
1311 C RETURN
1312 C 740 CONTINUE
1313 C
1314 C EVERYTHING CHECKS. TURN ON THE COMPLEX FLAG,INDICATE THAT THE
1315 C FIRST HALF OF THE CONSTANT IS BEING PROCESSED,AND GO ON TO
1316 C THE NEXT CHARACTER.
1317 C
1318 C CMPLEX=.TRUE.
1319 C CFRST=.TRUE.
1320 C GO TO 420
1321 C
1322 C *****
1323 C CHARACTER IS A RIGHT PARENTHESIS

```

```

1324 C      ****
1325 C      750 CONTINUE
1326 C      IF NOT PROCESSING A COMPLEX CONSTANT,FATAL ERROR
1327 C
1328 C      IF(,NOT,CMPLEX) GO TO 440
1329 C
1330 C      IF A RIGHT PARENTHESIS HAS PREVIOUSLY BEEN FOUND,FATAL ERROR
1331 C
1332 C      IF(CPFND) GO TO 440
1333 C
1334 C      SEE IF THE COMPLEX CONSTANT HAS ALREADY BEEN STORED
1335 C
1336 C      IF(,NOT,CDONE) GO TO 760
1337 C
1338 C      YES, TURN OFF THE COMPLEX FLAG,TURN ON THE DELIMITER-EXPECTED
1339 C      FLAG,AND GO LOOK FOR THE BEGINNING OF ANOTHER VARIABLE
1340 C      OR CONSTANT,
1341 C
1342 C      CMPLEX=.FALSE.,
1343 C      DELIM=.TRUE.,
1344 C      GO TO 90
1345 C
1346 C      760 CONTINUE
1347 C
1348 C      IF STILL PROCESSING THE FIRST PART,OR EXPECTING A DELIMITER,
1349 C      OR WORKING ON A FIX POINT PART,FATAL ERROR
1350 C
1351 C
1352 C      IF(CFRST,OR,DELIM,OR,FIX) GO TO 640
1353 C
1354 C      ALL CHECKS PASSED, INDICATE CLOSING PARENTHESIS FOUND AND GO
1355 C      STORE THE RESULTS.
1356 C
1357 C      CPFND=.TRUE.,
1358 C      GO TO 480
1359 C
1360 C      ****
1361 C      CHARACTER WAS AN H,AN L,OR AN R
1362 C      ****
1363 C
1364 C      770 CONTINUE
1365 C
1366 C      IF A DELIMITER IS EXPECTED,FATAL ERROR
1367 C
1368 C      IF(DELIM) GO TO 260
1369 C
1370 C      IF PROCESSING A FLOATING POINT NO, OR A COMPLEX CONSTANT,
1371 C      FATAL ERROR
1372 C
1373 C      IF(FIX,AND,,NOT,CMPLEX) GO TO 780
1374 C      CALL MESSOT(34)
1375 C      IEFLAG=34
1376 C      RETURN
1377 C
1378 C      780 CONTINUE
1379 C
1380 C      IF NO CHARACTER COUNT HAS BEEN PREVIOUSLY FOUND,FATAL ERROR

```

```

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 C      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 IDSTR
1415 C
1416      DO 820 I=1,IVNUM
1417      CALL NXTCOL(IF,IEFLAG)
1418      IF(IEFLAG,NE,0) RETURN
1419      IDSTR(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      DELIMR.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 II=1,IVNUM
1473      II=II-1
1474      IIP=IIP-1
1475      880 IDSTR(II)=IDSTR(IIP)
1476      DO 890 II=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      *****CHARACTER IS A B*****
1519      C
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      ++++++LOOP TO DECODE OCTAL NUMBERS+++++
1545      C      LOOP TO DECODE OCTAL NUMBERS
1546      C      ++++++LOOP TO DECODE OCTAL NUMBERS+++++
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      C      ITEMP=AND SHIFT(IDSTR(I),=54),778)
1559      C
1560      C      MAKE SURE IT IS NOT AN 8 OR A 9. IF SO,FATAL ERROR,
1561      C
1562      C      IF(ITEMP.GT.INF+7) GO TO 980
1563      C
1564      C      CONVERT,SHIFT,AND OR
1565      C
1566      C      ISHFT=ISHFT+3
1567      C      ITEMP=SHIFT(ITEMP-INF,ISHFT)
1568      C      IVSUM=OR(IVSUM,ITEMP)
1569      C      960 CONTINUE
1570      C      970 CONTINUE
1571      C
1572      C      ++++++-----+
1573      C
1574      C      TURN ON THE DELIMITER FLAG AND GO STORE THE CONVERTED CONSTANT
1575      C
1576      C      DELIM=.TRUE.
1577      C      GO TO 1070
1578      C      980 CONTINUE
1579      C
1580      C      ILLEGAL CHARACTER IN OCTAL CONSTANT
1581      C
1582      C      CALL MESSOT(41)
1583      C      IEFLAG=41
1584      C      RETURN
1585      C
1586      C      -----
1587      C      CHARACTER IS + OR -
1588      C
1589      C
1590      C      990 CONTINUE
1591      C
1592      C      IF A DELIMITER WAS EXPECTED,FATAL ERROR
1593      C
1594      C      IF(DELIM) GO TO 260
1595      C
1596      C      IF A SIGN HAS ALREADY BEEN FOUND,FATAL ERROR
1597      C
1598      C      IF(NOSGN,NE,0) GO TO 1000
1599      C
1600      C      SEE IF FIXED POINT FLAG IS STILL SET
1601      C
1602      C      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      C      IF(IVNUM,EQ,0) GO TO 1010
1607      C      1000 CONTINUE
1608      C      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 C      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 C      NO8GN=1
1621 C      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 C      IF(FXPN) GO TO 1040
1628 1030 CONTINUE
1629 C      CALL MESSOT(27)
1630 C      IEFLAG=27
1631 C      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 C      IF(IVNUM,NE,0) GO TO 1030
1638 C
1639 C      ALL O.K, GO SET THE SIGN AND CONTINUE,
1640 C
1641 C      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 C      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 C      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 C      IF(IVNUM,GT,NDSP) GO TO 600
1664 C
1665 C      CHECK FOR NO CHARACTERS FOUND, IF SO,FATAL ERROR.

```

```

1666 C
1667 C     IF(IVNUM,NE,0) GO TO 1070
1668 C     CALL MESSOT(30)
1669 C     IEFLAG#30
1670 C     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 C     IF(CMPLEX) GO TO 640
1677 C
1678 C     ALL IS O.K. AFFIX THE SIGN AND STORE THE CONSTANT.
1679 C
1680 C     IF(IVSIGN,LT,0) IVSUM=-IVSUM
1681 C     CALL STORE(TABLE,IEFLAG)
1682 C     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 C     IF(DONE) RETURN
1689 C     GO TO 90
1690 C
1691 C     *****
1692 C     CHARACTER IS S
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 S AS FIELD TERMINATOR AND GO FINISH PROCESSING THE NO.
1699 C     IF FIXED POINT, GO PERFORM CHECKS AND STORE THE CONSTANT.
1700 C
1701 C     DONE=.TRUE.
1702 C     IF(.NOT,FIX) GO TO 480
1703 C     GO TO 1060
1704 C
1705 C     ++++++JUMP TO HERE IF DELIMITER EXPECTED AND FOUND+++++
1706 C
1707 C
1708 C
1709 1090 CONTINUE
1710 C
1711 C     RESET THE DELIMITER-EXPECTED FLAG
1712 C
1713 C     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 C     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      C      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      C      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      C      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      C      EXPN=TRUE,
1748      C      NOSGN=0
1749      C
1750      C      SEE IF THE FIELD CURRENTLY BEING PROCESSED IS FIXED POINT
1751      C
1752      C      IF(,NOT,FIX) GO TO 1120
1753      C
1754      C      YES, SEE IF ANY CHARACTERS HAVE PREVIOUSLY BEEN FOUND,
1755      C
1756      C      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      C      IVNUM=1
1761      C      IVSUM=1
1762      C      IF(IVSIGN,LT,0) IVSUM=-IVSUM
1763      C
1764      C      RESET THE SIGN
1765      C
1766      C      IVSIGN=1
1767      C
1768      C      GO FINISH PROCESSING PART TO LEFT OF THE EXPONENT
1769      C
1770      C      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      C      SAVEN, FALSE,
1777      C      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 C      IF(DELIM) GO TO 260
1788 C
1789 C      SEE IF COMPLEX CONSTANT IS BEING PROCESSED, IF SO, FATAL ERROR,
1790 C
1791 C      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 C      DOBLE=.TRUE.
1797 C      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 C      IF(DELIM) GO TO 260
1808 C
1809 C      IF FLOATING POINT FLAG IS ALREADY SET, FATAL ERROR
1810 C
1811 C      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 C      IF(IVNUM.GT,NDDP) 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 C      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 C      IVNUM=0
1835 C
1836 C      TURN ON THE FLOATING POINT FLAG

```

```

1837 C      FIX#,FALSE,
1838 C
1839 C      STORE THE CHARACTER FOR LATER DOUBLE PRECISION PROCESSING IF
1840 C      NECESSARY
1841 C
1842 C      IC=IC+1
1843 C      IDSTR(IC)=1H.
1844 C
1845 C      IF THE DOUBLE PRECISION FLAG IS ON,STORE PART TO LEFT OF DECIMAL
1846 C      IN DOUBLE PRECISION FORM
1847 C
1848 C      IF(DOBLE) DB=XLEFT
1849 C
1850 C      GO ON AND LOOK AT THE NEXT CHARACTER
1851 C
1852 C      GO TO 420
1853 C
1854 C      FATAL ERROR, TOO MANY DIGITS IN CONSTANT,
1855 C
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 C      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 C      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 ISAVENITEST
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

```

```

1894 C      SET THE DELIMITER-EXPECTED FLAG
1895 C      DELIM=,TRUE,
1896 C
1897 C      PICK UP EITHER A TRUE OR FALSE CONSTANT
1898 C      IVSUM=LCONT
1899 C      IF(ISAVE,EQ,1HF) IVSUM=LCONF
1900 C
1901 C      IF THE CONSTANT IS SIGNED, INTERPRET = TO MEAN COMPLEMENT
1902 C
1903 C      IP(IVSIGN,GE,0) GO TO 1200
1904 C      IF(IVSUM,EQ,LCONT) IVTEMP=LCONT
1905 C      IF(IVSUM,EQ,LCONF) IVTEMP=LCONF
1906 C      IVSUM=IVTEMP
1907 C
1908 C      1200 CONTINUE
1909 C
1910 C
1911 C      STORE THE CONSTANT AND PROCEED ON TO NEXT VARIABLE OR CONSTANT
1912 C
1913 C      CALL STORE(TABLE,IEFLAG)
1914 C      IF(IEFLAG,NE,0) RETURN
1915 C      GO TO 90
1916 C      1210 CONTINUE
1917 C
1918 C      LONG FORM OF LOGICAL CONSTANT BEING USED. CHECK FOR BEGINNING
1919 C      OF TRUE OR FALSE CONSTANT.
1920 C
1921 C      IF(ISAVE,EQ,1HF) GO TO 1250
1922 C
1923 C      TRUE. NEXT CHARACTER BETTER BE R OR FATAL ERROR.
1924 C
1925 C      IF(IEST,EQ,1HR) GO TO 1230
1926 C      1220 CONTINUE
1927 C      CALL MESSOT(10)
1928 C      IEFLAG=10
1929 C      RETURN
1930 C      1230 CONTINUE
1931 C
1932 C      TR FOUND. GET NEXT CHARACTER AND CHECK FOR U. IF NOT, FATAL ERROR.
1933 C
1934 C      CALL NXTCOL(IP,IEFLAG)
1935 C      IF(IEFLAG,NE,0) RETURN
1936 C      IF(IEST,NE,1HU) GO TO 1220
1937 C      1240 CONTINUE
1938 C
1939 C      ALL BUT E. SUCCESSFULLY FOUND. CHECK FOR THESE. IF NOT FOUND,
1940 C      FATAL ERROR. IF FOUND, GO SET UP AND STORE CONSTANT.
1941 C
1942 C      CALL NXTCOL(IP,IEFLAG)
1943 C      IF(IEFLAG,NE,0) RETURN
1944 C      IF(IEST,NE,1HE) GO TO 1220
1945 C      CALL NXTCOL(IP,IEFLAG)
1946 C      IF(IEFLAG,NE,0) RETURN
1947 C      IF(IEST,NE,1H,) GO TO 1220
1948 C      GO TO 1190
1949 C      1250 CONTINUE
1950 C

```

```

1951      C      FALSE BEING PROCESSED, CHECK FOR ALS,
1952      C
1953      IF(IEST,NE,1HA) GO TO 1220
1954      CALL NXTCOL(IF,IEFLAG)
1955      IF(IEFLAG,NE,0) RETURN
1956      IF(IEST,NE,1HL) GO TO 1220
1957      CALL NXTCOL(IF,IEFLAG)
1958      IF(IEFLAG,NE,0) RETURN
1959      IF(IEST,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      C      SUBROUTINE MESSOT(IEFLAG)
2      C      ROUTINE TO DO PRINTING FOR NAMLST
3      C
4      C      IEFLAG IS THE ERROR FLAG (SEE NAMLST)
5      C
6      C      WRITTEN BY J.L.NORTON,LASL T=3,1974
7      C
8      C
9      COMMON/ARRECON/ICHAR(80),ISUB,IEST,NSUB,N8UBV(4),IENTRY,ISBSPT,
10     ICREAL,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
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

```

```

37   C      PICK UP THE NEXT FILE NAME
38   C
39   C      ITST=IFT+I=1
40   C      IPX=GETIT(ITST)
41   C      20 CONTINUE
42   C
43   C      PRINT A ROW OF ASTERISKS UNLESS JUST PRINTING THE CARD
44   C
45   C      IF(IEFLAG,EQ,1) GO TO 30
46   C      WRITE(IPX,970)
47   C      30 CONTINUE
48   C
49   C      PRINT THE CARD ITSELF
50   C
51   C      WRITE(IPX,510) ICHAR
52   C
53   C      ALL DONE IF JUST PRINTING THE CARD
54   C
55   C      IF(IEFLAG,EQ,1) GO TO 500
56   C
57   C      ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++
58   C      SET UP POINTER TO AREA OF CARD CAUSING ERROR
59   C      ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++ ++++++
60   C
61   C      BLANK OUT THE POINTER LINE
62   C
63   C      DO 40 J=1,8
64   C      40 ILINE(J)=1H
65   C
66   C      APPLY 80 AS UPPER BOUND ON CARD COLUMN NO.
67   C
68   C      ISUBX=ISUB
69   C      IF(ISUBX.GT,80) ISUBX=80
70   C
71   C      DETERMINE WORD COUNT (L) AND CHARACTER COUNT WITHIN WORD (K)
72   C      OF CARD WORD CONTAINING THE ERROR
73   C
74   C      K=MOD(ISUBX,10)
75   C      L=ISUBX/10
76   C      IF(K.NE,0) L=L+1
77   C      IF(K,EQ,0) K=10
78   C
79   C      SHIFT THE POINTER TO THE PROPER COLUMN POSITION
80   C
81   C      KX=10-K
82   C      IKX=6*KX
83   C      IC=SHIFT(IPT,IKX)
84   C
85   C      MASK OUT THE PROPER POSITION IN THE LINE AND INSERT THE POINTER
86   C
87   C      IM=SHIFT(77B,IKX)
88   C      ILINE(L)=OR(IC,AND(ILINE(L),COMP(IM)))
89   C
90   C      PRINT THE POINTER LINE
91   C
92   C      WRITE(IPX,520) ILINE
93   C

```

```

94      C      GO PRINT THE PROPER ERROR MESSAGE
95      C
96      GO TO (500,50,60,70,80,90,100,110,120,130,140,150,160,170,180,190,
97      1 200,210,220,230,240,250,260,270,280,290,300,310,320,330,340,350,
98      2 360,370,380,390,400,410,420,430,440,450,460,470,480),IEFLAG
99      50 WRITE(IPX,530)
100     GO TO 490
101     60 WRITE(IPX,540)
102     GO TO 490
103     70 WRITE(IPX,550)
104     GO TO 490
105     80 WRITE(IPX,560)
106     GO TO 490
107     90 WRITE(IPX,570)
108     GO TO 490
109     100 WRITE(IPX,580)
110    GO TO 490
111    110 WRITE(IPX,590)
112    GO TO 490
113    120 WRITE(IPX,600)
114    GO TO 490
115    130 WRITE(IPX,610)
116    GO TO 490
117    140 WRITE(IPX,620)
118    GO TO 490
119    150 WRITE(IPX,630)
120    GO TO 490
121    160 WRITE(IPX,640)
122    GO TO 490
123    170 WRITE(IPX,650)
124    GO TO 490
125    180 WRITE(IPX,660)
126    GO TO 490
127    190 WRITE(IPX,670)
128    GO TO 490
129    200 WRITE(IPX,680)
130    GO TO 490
131    210 WRITE(IPX,690)
132    GO TO 490
133    220 WRITE(IPX,700)
134    GO TO 490
135    230 WRITE(IPX,710)
136    GO TO 490
137    240 WRITE(IPX,720)
138    GO TO 490
139    250 WRITE(IPX,730)
140    GO TO 490
141    260 WRITE(IPX,740)
142    GO TO 490
143    270 WRITE(IPX,750)
144    GO TO 490
145    280 WRITE(IPX,760)
146    GO TO 490
147    290 WRITE(IPX,770)
148    GO TO 490
149    300 WRITE(IPX,780)
150    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 ,36H8SIGN 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,TEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
8      CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
9      LOGICAL MULT,CMPLEX,DIM,DOBLE
10     DIMENSION TABLE(3,1)
11     INTEGER TABLE,SHFT,AND,OR,COMP
12     DATA CREAL/0./
13     C
14     C      GET THE VARIABLE SUBSCRIPT INFORMATION

```

```

15 C
16 C IARENT#TABLE(3,IENTRY)
17 C
18 C GET THE ABSOLUTE ADDRESS OF THE VARIABLE
19 C
20 C IARADD#TABLE(2,IENTRY)
21 C
22 C ISBSPT IS THE MEMORY OFFSET (+1) FOR THE VARIABLE BEING READ. IT
23 C IS 1 UPON THE ENTRY TO STORE FOR A GIVEN VARIABLE NAME. SOME
24 C CHECKING IS DONE ONLY UPON THE FIRST ENTRY. SEE IF THIS IS SO,
25 C
26 C IF(ISBSPT.LE.1) GO TO 10
27 C
28 C NOT THE FIRST ENTRY. SEE IF THE VARIABLE IS SUBSCRIPTED. IF NOT,
29 C FATAL ERROR FOR TRYING TO STORE MORE THAN ONE DATA ELEMENT
30 C INTO A NON-SUBSCRIPTED VARIABLE.
31 C
32 C IF(IARENT.EQ.0) GO TO 60
33 C
34 C EVERYTHING CHECKS, CONTINUE ON WITH THE STORE.
35 C
36 C GO TO 20
37 C 10 CONTINUE
38 C
39 C FIRST ENTRY, SEE IF A MULTIPLIER IS IN EFFECT OR VARIABLE HAS
40 C SUBSCRIPTS.
41 C
42 C IF(MULT,OR,DIM) GO TO 50
43 C
44 C NEITHER CONDITION HOLDS, CONTINUE ON WITH THE STORE.
45 C
46 C 20 CONTINUE
47 C
48 C SET THE MULTIPLIER COUNT. IF NO MULTIPLIER, USE 1.
49 C
50 C IF(,NOT,MULT) MULTSV=1
51 C
52 C LOOP FOR MULTIPLE STORING
53 C
54 C DO 40 I=1,MULTSV
55 C
56 C GET THE ACTUAL ABSOLUTE ADDRESS OF THE LOCATION INTO WHICH THE
57 C STORE IS TO OCCUR
58 C
59 C IADD#IARADD+ISBSPT+1
60 C
61 C SPECIAL HANDLING FOR DOUBLE OR COMPLEX
62 C
63 C IF(CMPLEX,OR,DOBLE) GO TO 30
64 C
65 C ORDINARY ONE ELEMENT VARIABLE, CARRY OUT THE STORE.
66 C
67 C CALL STORIT(IV$UM,IADD)
68 C
69 C INCREMENT THE POINTER AND GO TO THE END OF THE LOOP
70 C
71 C ISBSPT=ISBSPT+1

```

```

72      GO TO 40
73      30 CONTINUE
74      C
75      C      EITHER DOUBLE OR COMPLEX. STORE TWO WORDS.
76      C
77      CALL STORIT(CREAL,IADD)
78      CALL STORIT(CIMAG,IADD+1)
79      ISBSPT=ISBSPT+2
80      40 CONTINUE
81      C
82      C      ALL DONE
83      C
84      RETURN
85      50 CONTINUE
86      C
87      C      VARIABLE HAS SUBSCRIPTS OR CONSTANT HAS A MULTIPLIER. VARIABLE
88      C      MUST BE DIMENSIONED. IF NOT, FATAL ERROR.
89      C
90      IF(IARENT,NE,0) GO TO 70
91      60 CONTINUE
92      CALL MESSOT(15)
93      IEFLAG=15
94      RETURN
95      70 CONTINUE
96      C
97      C      IF VARIABLE IS NOT SUBSCRIPTED, ALL IS O.K., CARRY OUT THE STORE.
98      C
99      IF(.NOT,DIM) GO TO 20
100     C
101     C      VARIABLE IS SUBSCRIPTED. GET THE TABLE DIMENSION INFORMATION.
102     C
103     ISUB1=AND(IARENT,777777B)
104     ISUB2=AND(SHIFT(IARENT,-18),777777B)
105     ISUB3=AND(SHIFT(IARENT,-36),777777B)
106     C
107     C      CHECK FOR SUBSCRIPTING ERRORS. IS THERE A FOURTH SUBSCRIPT BUT
108     C      NO THIRD DIMENSION.
109     C
110     IF(NSUB,EQ,4,AND.ISUB3,EQ,0) GO TO 80
111     C
112     C      IS THERE A THIRD SUBSCRIPT BUT NO SECOND DIMENSION
113     C
114     IF(NSUB,EQ,3,AND.ISUB2,EQ,0) GO TO 80
115     C
116     C      IS THERE A SECOND SUBSCRIPT BUT EITHER NO FIRST DIMENSION OR A
117     C      FIRST DIMENSION INDICATING A SIMPLY-SUBSCRIPTED ARRAY
118     C
119     IF(NSUB,EQ,2,AND.(ISUB1,EQ,1,OR.ISUB1,EQ,0)) GO TO 80
120     C
121     C      IS THERE A SINGLE SUBSCRIPT BUT NO SUBSCRIPTING INFORMATION GIVEN
122     C
123     IF(NSUB,EQ,1,AND.ISUB1,EQ,0) GO TO 80
124     C
125     C      IF N SUBSCRIPTS ARE GIVEN, DIMENSIONAL INFORMATION IS USED FOR
126     C      THE FIRST N-1
127     C
128     IF(NSUBV(4),EQ,0) ISUB3#0

```

```

129      IF(NSUBV(3),EQ,0) ISUB2#0
130      IF(NSUBV(2),EQ,0) ISUB1#0
131      C      CALCULATE THE OFFSET FROM THE SUBSCRIPTING INFORMATION
132      C      ISBSPT=NSUBV(1)+ISUB1*(NSUBV(2)-1+ISUB2*(NSUBV(3)-1+ISUB3*(NSUBV(4
133      C      1 )-1)))
134
135      C      MODIFY THE OFFSET FOR COMPLEX VARIABLES
136      C      IF(CMPLEX) ISBSPT#2*ISBSPT#1
137      C      GO BACK AND DO THE STORE
138      C
139      C      GO TO 20
140      C      80 CONTINUE
141      C
142      C      FATAL SUBSCRIPTING ERROR
143
144      C      CALL MESSOT(16)
145      C      IEFLAG#16
146      C      RETURN
147      C      END
148
149
150
151

```

```

1      SUBROUTINE MASH(IDSTR,IVNUM,I NAME,IEFLAG)
2      C
3      C      ROUTINE TO CONVERT AN ARRAY IDSTR CONTAINING ONE SIX-BIT
4      C      CHARACTER PER WORD, LEFT-JUSTIFIED, INTO ONE WORD, I NAME,
5      C      CONTAINING THE SAME CHARACTERS LEFT TO RIGHT WITH BLANK FILL
6      C      AT THE RIGHT END IF NECESSARY. IVNUM CHARACTERS ARE PROCESSED
7      C      WHERE IVNUM MUST BE LE,10,
8      C
9      C      WRITTEN BY J.L.NORTON,LASL T=3,1974
10     C
11     DIMENSION IDSTR(1)
12     C      INTEGER SHIFT,AND,OR,COMP
13     C      DATA MASK/77000000000000000000000B/
14     C
15     C      CHECK THE CHARACTER COUNT, IF GT,10,FATAL ERROR.
16     C
17     C      IF(IVNUM,LE,10) GO TO 10
18     C      CALL MESSOT(45)
19     C      IEFLAG#45
20     C      RETURN
21     C      10 CONTINUE
22     C
23     C      INITIALIZE THE PACKED WORD
24     C
25     C      I NAME#1H
26     C
27     C      LOOP OVER ALL CHARACTERS

```

```

28      C      DO 20 I=1,IVNUM
29      C      SHIFT MASK INTO PROPER POSITION
30      C      ITEMPM=SHIFT(77B,60=6*I)
31      C      SHIFT CHARACTER INTO PROPER POSITION
32      C      ITEMPC=SHIFT(IDSTR(I),=6*(I-1))
33      C      DROP CHARACTER INTO INAME, FIRST MASK OUT THE BLANK, THEN ISOLATE
34      C      THE CHARACTER AND DROP IT INTO THE PREPARED SLOT.
35      C      20 INAME=OR(AND(INAME,COMP(ITEMPM)),AND(ITEMP,ITEMPM))
36      C      ALL DONE
37      C      RETURN
38      C      END

```

```

1      SUBROUTINE NAMPRT(NFX,FTABX)
2
3      C      ROUTINE TO SET THE FILES ON WHICH THE NAMSLT ROUTINE WILL WRITE
4      C      WHENEVER ITS DOES OUTPUT OTHER THAN ERROR MESSAGES (PS OR CS),
5      C      THE CALLING SEQUENCE IS    CALL NAMPRT(N, TABLE)    WHERE
6      C      N = NO. OF TABLE ENTRIES
7      C      TABLE = ARRAY OF FILE NAMES,EITHER IN THE FORM OF
8      C      LEFT-JUSTIFIED,ZERO-FILLED HOLLERITH
9      C      CONSTANTS OR INTEGERS,
10     C
11     C      FOR EXAMPLE, IF ONE WANTED TO WRITE BOTH ON FILM AND PAPER,
12     C      THE TABLE WOULD APPEAR AS
13     C      DATA TABLE/4LFILM,3LOUT/.
14     C      IF FILM WAS EQUIVALENCED TO FSET12 AND OUT TO FSET6,
15     C      AN ALTERNATE FORM WOULD BE
16     C      DATA TABLE/12,6/
17     C      AND THE CALL WOULD BE
18     C      CALL NAMPRT(2, TABLE).
19
20     C      IF NAMPRT IS NOT CALLED BEFORE, NAMSLT, THE LATTER WILL
21     C      DEFAULT ALL OUTPUT TO OUT.
22
23     C      WRITTEN BY J.L.NORTON, LASL T-3, 1974
24
25     COMMON/IOTAB/NF,IFLOC
26     INTEGER FTABX(1)
27     DATA NF,IFLOC/0,0/
28     IF(NFX,LE,0) RETURN
29     NF=NFX
30     IFLOC=LOCF(FTABX)

```

```
31      RETURN  
32      END
```

```
1      SUBROUTINE ERRPRT(NFX,FTABX)  
2      C  
3      C      ROUTINE ANALOGOUS TO NAMPRT EXCEPT FOR PRINTING ERROR MESSAGES  
4      C  
5      C      WRITTEN BY J.L.NORTON, LASL T=3, 1975  
6      C  
7      COMMON/JOTAB/NF, IFLOC  
8      INTEGER FTABX(1)  
9      DATA NF, IFLOC/0,0/  
10     IF(NFX.LE.0) RETURN  
11     NF=NFX  
12     IFLOC=LOCF(FTABX)  
13     RETURN  
14     END
```

```
1      SUBROUTINE TABDEF(TABLE,NAME,ITSIZE,IERRT)  
2      C  
3      C      ROUTINE TO INITIALIZE A NAMELIST TABLE  
4      C  
5      C      TABLE = ARRAY WHICH WILL BE USED TO STORE NAMELIST DATA NAMES  
6      C      AND RELATED INFORMATION. IT MUST BE 3*(N+1) ELEMENTS  
7      C      LONG WHERE N IS THE NUMBER OF DISTINCT NAMELIST  
8      C      VARIABLES TO BE READ UNDER THIS NAME.  
9      C      NAME = NAMELIST NAME, LEFT=JUSTIFIED, BLANK=FILLED HOLLERITH. IF  
10     C      ONE WERE GOING TO READ CARDS OF THE FORM PSCODEIN ,  
11     C      NAME=6HCODEIN,  
12     C      ITSIZE = SECOND SUBSCRIPT OF THE TABLE ARRAY  
13     C      (MAXIMUM NO. OF TABLE ENTRIES+1)  
14     C      IERRT = VARIABLE SET BY ASSIGNED GO TO AS RETURN POINT IF TABLE  
15     C      INITIALIZATION ENCOUNTERS AN ERROR CONDITION  
16     C  
17     C      WRITTEN BY J.L.NORTON, LASL T=3, 1974  
18     C  
19     INTEGER TABLE  
20     DIMENSION TABLE(3,1)  
21     COMMON/ERRORC/IERRTP  
22     TABLE(1,1)=NAME  
23     TABLE(2,1)=ITSIZE  
24     TABLE(3,1)=0  
25     IERRTP#IERRT  
26     RETURN  
27     END
```

```

1      SUBROUTINE TABSET(TABLE,NAME,LOCA,IEFLAG,NSUB,NSUB1,NSUB2,NSUB3)
2      C
3      C      ROUTINE TO ADD A VARIABLE TO A NAMELIST TABLE
4      C
5      C      TABLE = AN ARRAY WHICH HAS BEEN INITIALIZED BY A CALL TO TABDEF
6      C      NAME = NAME OF VARIABLE TO BE ADDED, AS A HOLLERITH CONSTANT
7      C      LOCA = THE VARIABLE ITSELF
8      C      IEFLAG= ERROR INDICATOR
9      C          • RETURNED ZERO IF NO ERROR WAS FOUND
10     C          • RETURNED WITH THE CONTENTS OF THE VARIABLE =NAME=
11     C              LEFT=JUSTIFIED (LEFT 6 CHARACTERS) AND THE ERROR NO.
12     C              RIGHT=JUSTIFIED IF AN ERROR OCCURRED
13     C      NSUB = NUMBER OF SUBSCRIPTS (0,1,2,3,OR 4)
14     C      NSUB1 = FIRST ARRAY DIMENSION
15     C      NSUB2 = SECOND ARRAY DIMENSION
16     C      NSUB3 = THIRD ARRAY DIMENSION
17     C          • (IF THE NAMELIST VARIABLE HAS N SUBSCRIPTS (N,LE,4),
18     C          THEN NSUB1,NSUB2,...,NSUBN=1 ARE NEEDED)
19
20     C      THE ACTUAL FORM OF THE TABLE IS AS FOLLOWS ==
21     C          WORD 1 = THE TABLE NAME (LEFT=JUSTIFIED,BLANK=FILLED
22     C                      HOLLERITH)
23     C          WORD 2 = UNUSED
24     C          WORD 3 = NO. OF ENTRIES IN THE TABLE
25     C      NEXT FOLLOW TRIPLETS OF WORDS FOR EACH NAMELIST VARIABLE =
26     C          WORD 1 = VARIABLE NAME (SAME AS TABLE NAME)
27     C          WORD 2 = VARIABLE LOCATION (RIGHT JUSTIFIED,RELATIVE
28     C                      TO BEGINNING OF CODE FIELD=LENGTH)
29     C          WORD 3 = SUBSCRIPT INFORMATION
30     C              • IF ZERO,THE VARIABLE IS SINGLY SUBSCRIPTED
31     C              • IF NON-ZERO,THE WORD IS INTERPRETED AS BEING
32     C                  MADE UP OF THREE 18 BIT FIELDS, DENOTING
33     C                  THE RIGHTMOST BIT AS 0, THE THREE FIELDS ARE
34     C                  BITS 0=17,18=35,AND 36=53,RESPECTIVELY,
35     C                  THESE THREE FIELDS CONTAIN NSUB1,NSUB2,AND
36     C                  NSUB3,RESPECTIVELY.
37     C          THE LENGTH OF THE TABLE SHOULD BE 3*(WORD(3)+1) LOCATIONS,
38
39     C      WRITTEN BY J.L.NORTON,LASL T=3,1974
40
41     C      INTEGER TABLE,SHIFT,AND,OR,COMP
42     C      DIMENSION TABLE(3,1),FMT(10)
43     C      COMMON/ERRORC/IERRT
44     C      IEFLAG=0
45
46     C      GET NUMBER OF ENTRIES CURRENTLY IN THE TABLE
47     C
48     C      NENTRY=TABLE(3,1)
49
50     C      COMPUTE SUBSCRIPT FOR NEW TABLE ENTRY
51     C
52     C      ISUB=NENTRY+2
53
54     C      MAKE SURE THERE IS STILL TABLE SPACE LEFT

```

```

55      C
56      C      IF(ISUB,LE,TABLE(2,1)) GO TO 10
57      C
58      C      NO. FATAL ERROR,
59      C
60      C      IEFLAG=5
61      C      GO TO 999
62      C      10 CONTINUE
63      C
64      C      SET UP TABLE ENTRY
65      C
66      C      TABLE(1,ISUB)=NAME
67      C      TABLE(2,ISUB)=LOCF(LOCA)
68      C      TABLE(3,ISUB)=0
69      C
70      C      COMPOSE SUBSCRIPT WORD IF VARIABLE IS AN ARRAY
71      C
72      C      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      C      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      C      TABLE(3,ISUB)=1
82      C      GO TO 30
83      C
84      C      ARRAY HAS MORE THAN ONE SUBSCRIPT
85      C
86      C      20 CONTINUE
87      C
88      C      SEE IF AT LEAST THE FIRST SUBSCRIPT IS SPECIFIED. IF NOT,ERROR,
89      C
90      C      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      C      TABLE(3,ISUB)=NSUB1
96      C      IF(NSUB,EQ,2) GO TO 30
97      C
98      C      YES. THE SECOND SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR,
99      C
100     C      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     C      TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB2,18))
106     C      IF(NSUB,EQ,3) GO TO 30
107     C
108     C      YES. THE THIRD SUBSCRIPT MUST BE SPECIFIED. IF NOT,ERROR,
109     C
110     C      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 C      TABLE(3,ISUB)=OR(TABLE(3,ISUB),SHIFT(NSUB3,36))
116 C      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 C      TABLE(3,1)=NENTRY+1
122 C      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,7777777777700000000B))
144 GO TO IERRT
145 END

```

```

1      SUBROUTINE NXTCOL(IF,IEFLAG)
2
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 C      COMMON/ARRCON/ICHAR(80),ISUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
9 C      CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
10 C      LOGICAL LIST,HOLLER
11 C
12 C      INCREMENT THE COLUMN NO.
13 C
14 C      ISUB=ISUB+1
15 C
16 C      PICK UP THE CHARACTER IN COLUMN ISUB

```

```

17   C      ITEST=ICHAR(ISUB)
18   C
19   C      IF ISUB HAS NOT MOVED ACROSS A CARD BOUNDARY, WE ARE ALL DONE
20   C
21   C      IF (ISUB,LE,80) GO TO 30
22   C
23   C      MUST READ NEXT CARD. IEFLAG WILL BE RETURNED NON-ZERO ONLY IF
24   C      AN EOF WAS ENCOUNTERED.
25   C
26   C      CALL READIT(IF,IEFLAG)
27   C      IF (IEFLAG,EQ,0) GO TO 10
28   C
29   C      AN EOF WAS READ. WE ARE ALL THROUGH.
30   C
31   C      CALL MESSOT(32)
32   C      IEFLAG=32
33   C      RETURN
34   C
35   10 CONTINUE
36   C
37   C      READ WAS SUCCESSFUL. RESET COLUMN AND CHARACTER.
38   C
39   C      ISUB=2
40   C      ITEST=ICHAR(ISUB)
41   C
42   C      IF PROCESSING A HOLLERITH FIELD, NO MORE CHECKING
43   C
44   C      IF (HOLLER) GO TO 20
45   C
46   C      CHECK FOR $ IN CC2. IF NOT, ALL FINISHED CHECKING.
47   C
48   C      IF (ITEST,NE,1HS) GO TO 20
49   C
50   C      S IN CC2. PROBABLY BEGINNING OF NEXT NAMELIST STATEMENT. MUST
51   C      HAVE HAD MISSING TERMINAL S ON PREVIOUS CARD. GO BACK AND
52   C      READ THE CARD IN ERROR.
53   C
54   C      BACKSPACE IF
55   C      BACKSPACE IF
56   C      CALL READIT(IF,IEFLAG)
57   C
58   C      SET THE COLUMN POINTER TO CC80, GO PRINT ERROR MESSAGE, AND QUIT
59   C
60   C      ISUB=80
61   C      CALL MESSOT(33)
62   C      IEFLAG=33
63   C      RETURN
64   C
65   20 CONTINUE
66   C
67   C      EVERYTHING CHECKED OUT. SET COLUMN TO 1 AND PICK UP THE
68   C      CHARACTER IN CC1.
69   C
70   C      ISUB=1
71   C      ITEST=ICHAR(ISUB)
72   C
73   C      IF FLAG IS SET, PRINT THE LAST CARD READ

```

```
74      IF(LIST) CALL MESSOT(1)
75 30 CONTINUE
76 RETURN
77 END
```

```
1      SUBROUTINE READIT(IF,IEFLAG)
2      C
3      C      ROUTINE TO READ THE NEXT CARD ON FILE IF
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1974
6      C
7      C      COMMON/ARRCON/ICHAR(80),IBUB,ITEST,NSUB,NSUBV(4),IENTRY,ISBSPT,
8      C      CREAL,CIMAG,CMPLEX,DIM,MULTSV,IVSUM,MULT,LIST,HOLLER,DOBLE
9      C
10     C      READ THE NEXT CARD
11    C
12    C      READ(IF,30) ICHAR
13    C
14    C      CHECK FOR END-OF-FILE
15    C
16    C      IF(EOF,IF) 10,20
17    C
18    C      YES, SET THE FLAG.
19    C
20    10 CONTINUE
21    IEFLAG=-1
22    20 CONTINUE
23    RETURN
24    C
25    30 FORMAT(80A1)
26    END
```

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

```

14      DATA IOP/2HPS,2HRL,2HWL,2HMJ,2HRX,2HWX,2HRI,2HTB,2HIB,2HRO,2HOB,
15      1 2HRJ,2HJP,2HZR,2HNZ,2HPL,2HNG,2HIR,2HOR,2HDF,2HID,2HEQ,2HNE,2HGE,
16      2 2HLT,2HBX,2HLX,2MAX,2HMX,2HNX,2HZX,2HUX,2HPX,2HFX,2HDX,2HRX,2HIX,
17      3 2HNO,2HCX,2HSA,2HSB,2HSX/
18      DATA IDNAME/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7/
19      C
20      C      IF FWA,GT,LWA,IGNORE THE CALL
21      C
22      C      IF(FWA,GT,LWA) RETURN
23      C      FWAP=FWA
24      C
25      C      LOOP OVER ALL WORDS TO BE DUMPED
26      C
27      10 CONTINUE
28      C
29      C      GET THE CONTENTS OF THE CURRENT ADDRESS (FWAP) TO BE DUMPED
30      C
31      C      IWORD=GETIT(FWAP)
32      C
33      C      BREAK THE WORD UP INTO INDIVIDUAL DIGITS
34      C
35      DO 20 I=1,20
36      IDIGIT(I)=AND SHIFT(IWORD,3*I),7B
37      20 CONTINUE
38      C
39      C      COMPUTE HOLLERITH EQUIVALENT OF DIGITS
40      C
41      DO 30 I=1,20
42      ISUB=IDIGIT(I)+1
43      30 IHQR(I)=IDNAME(ISUB)
44      C
45      C      IDN IS THE CURRENT DIGIT NO. OF THE WORD BEING PROCESSED
46      C
47      C      IDN#1
48      C
49      C      IPART IS THE NO. OF THE INSTRUCTION (1=4) BEING PROCESSED
50      C      FOR THE CURRENT WORD
51      C
52      C      IPART#1
53      C
54      C      INITIALIZE THE OUTPUT FIELDS
55      C
56      DO 50 I=1,4
57      ICODE(I)=2H
58      IREG(I)=1H
59      DO 40 J=1,13
60      IOPRN(J,I)=1H
61      40 CONTINUE
62      50 CONTINUE
63      C
64      C      LOOP OVER ALL DIGITS
65      C
66      60 CONTINUE
67      C
68      C      IOPNO IS THE NO. OF THE CHARACTER OF THE OPERAND FIELD CURRENTLY
69      C      BEING FILLED
70      C

```

```

71      IOPNO=1
72      IGOT=IDIGIT(IDN)+1
73      GO TO (70,360,440,500,520,620,620,620),IGOT
74      C
75      C      FIRST DIGIT OF OP IS 0
76      C
77      70 CONTINUE
78      IDN=IDN+1
79      IGOT=IDIGIT(IDN)+1
80      GO TO (80,190,330,340,350,350,350,350),IGOT
81      C
82      C      OP IS PS (00)
83      C
84      80 CONTINUE
85      ITYPE=1
86      C
87      C      CHECK FOR ERRORS
88      C
89      IF(IDN.LE.12) GO TO 100
90      C
91      C      SEE IF THIS IS END OF WORD PADDED WITH ZEROS. IF SO,
92      C      PROCESS LIKE NO.
93      C
94      IF(IDN.EQ.17) GO TO 600
95      C
96      C      WORD IS IN ERROR, MUST NOT BE INSTRUCTION, PRINT MESSAGE AND GO ON
97      C      TO NEXT WORD.
98      C
99      90 CONTINUE
100     WRITE(IFILE,760) FWAP,IWORD
101     GO TO 180
102     100 CONTINUE
103     ITYPE=1
104     C
105     C      PROCESS INSTRUCTION OF THE FORM OP K
106     C
107     IDN=IDN+2
108     C
109     C      PROCESS K
110     C
111     110 CONTINUE
112     ISUM=0
113     DO 120 I=1,6
114     IDN=IDN+1
115     ISUM=OR(ISUM,SHIFT(IDIGIT(IDN),3*(6-I)))
116     120 CONTINUE
117     ISUM=SHIFT(SHIFT(ISUM,42),-42)
118     IF(ISUM.GE.0) GO TO 130
119     ISUM=-ISUM
120     IOPRN(IOPNO,IPART)=1H-
121     GO TO 140
122     130 CONTINUE
123     IOPRN(IOPNO,IPART)=1H+
124     140 CONTINUE
125     IOPNO=IOPNO+1
126     DO 150 I=1,6
127     ID=AND(SHIFT(ISUM,-3*(6-I)),7B)

```

```

128      IOPRN(IOPNO,IPART)=IDNAME(ID+1)
129      IOPNO=IOPNO+1
130      150 CONTINUE
131
132      C      FINISH PARCEL
133
134      160 CONTINUE
135      ICODE(IPART)=IOP(ITYPE)
136      IF(IDN,GE,20) GO TO 170
137      IPART=IPART+1
138      IDN=IDN+1
139      IF(IDN,GT,16) GO TO 90
140      GO TO 60
141
142      C      WORD IS FINISHED. PRINT THE LINE.
143
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
148      C      SEE IF THIS WAS THE LAST WORD
149
150      180 CONTINUE
151      IF(FWAP,GE,LWA) RETURN
152
153      C      NO. CONTINUE.
154
155      FWAP=FWAP+1
156      GO TO 10
157
158      C      FIRST TWO DIGITS OF OP ARE 01
159
160      190 CONTINUE
161      IDN=IDN+1
162      IGOT=IDIGIT(IDN)+1
163      GO TO (200,210,210,230,240,240,250,310),IGOT
164
165      C      OP IS RJ (010)
166
167      200 CONTINUE
168      ITYPE=12
169      IDN=IDN+1
170      GO TO 110
171
172      C      OP IS 011 OR 012
173
174      210 CONTINUE
175      ITYPE=IGOT
176
177      C      PROCESS INSTRUCTION OF THE FORM    OP   BJ+K
178
179      220 CONTINUE
180      IF(IDN,GT,13) GO TO 90
181      IDN=IDN+1
182      IOPRN(1,IPART)=1HB
183      IOPRN(2,IPART)=IHEQ(IDN)
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(IDN+1),NE,0,OR, IDIGIT(IDN+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#1GOT
200      C
201      C      PROCESS INSTRUCTION OF THE FORM    OPJ    XK
202      C
203      IDN=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(IDN+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(IDN)
222      GO TO 160
223      280 CONTINUE
224      IF(IDIGIT(IDN+2),NE,0) GO TO 290
225      ITYPE#8
226      C
227      C      PROCESS INSTRUCTION OF THE FORM    OPJ
228      C
229      IDN=IDN+1
230      IREG(IPART)=IHEQ(IDN)
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      IDN=IDN+1
240      IREG(IPART)=IHEQ(IDN)
241      IDN=IDN+1

```

```

242      GO TO 270
243      C
244      C      OP IS 017
245      C
246      310 CONTINUE
247      IF(IDIGIT(IDN+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(IDN.GT.12) GO TO 90
262      IDN=IDN+1
263      IOPRN(1,IPART)=1HB
264      IOPRN(2,IPART)=IHEQ(IDN)
265      IDN=IDN+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(IDN.GT.12) GO TO 90
273      IDN=IDN+1
274      IGOT=IDIGIT(IDN)
275      ITYPE=IGOT+14
276      C
277      C      PROCESS INSTRUCTION OF THE FORM   OP    XJ,K
278      C
279      IDN=IDN+1
280      IOPRN(1,IPART)=1HX
281      IOPRN(2,IPART)=IHEQ(IDN)
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(IDN.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      IDN=IDN+1
295      IOPRN(1,IPART)=1HB
296      IOPRN(2,IPART)=IHEQ(IDN)
297      IOPRN(3,IPART)=1H,
298      IOPRN(4,IPART)=1HB

```

```

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

```

```

413      IOPRN(1,IPART)=IHEQ(IDN)
414      IDN=IDN+1
415      IOPRN(2,IPART)=IHEQ(IDN)
416      IOPRN(3,IPART)=IHB
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      IDN=IDN+1
436      IGOTP=IDIGIT(IDN)
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      IDN=IDN+1
449      IGOT=IDIGIT(IDN)+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,1)
491      1 GO TO 610
492      IDN=IDN+2
493      GO TO 160
494      610 CONTINUE
495      IOPRN(1,IPART)=IHEQ(IDN)
496      IOPRN(2,IPART)=IHEQ(IDN+1)
497      IOPRN(3,IPART)=IHEQ(IDN+2)
498      IOPRN(4,IPART)=1HB
499      IDN=IDN+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      IDN=IDN+1
507      IGOT=IDIGIT(IDN)+1
508      IF(IGOT,LT,4,AND,1)
509      IDN=IDN+1
510      IREG(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      760 FORMAT(1H ,06,3X020,3X8(1H=),
555          1 54HWORD CONTAINS ILLEGAL INSTRUCTION(8), MUST BE DATA OR ,
556          2 10HCLOBBED,,8(1H=))
557      770 FORMAT(1H ,06,3X020,4(3XA2,A1,3X13A1))
558
559      END

```

```

1      SUBROUTINE PABORT
2
3      C      ROUTINE TO HANDLE ABORT EXCHANGE PACKAGE DUMPS
4
5      C      WRITTEN BY J.L.NORTON,LASL T=3,1974
6
7      C      INTEGER SHIFT
8      C      LOGICAL FILM
9      C      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
10     C      COMMON/IEDMP/IDUMP(16)
11
12    C      PRINT CP TIME AT ABORT SO A TIME LIMIT ABORT IS EASILY
13    C      DISTINGUISHABLE
14
15    C      CALL SECOND(TJLN)
16    C      DO 10 IPX=6,IFD,6
17    10 WRITE(IPX,30) TJLN

```

```

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

```

```

1      SUBROUTINE PARPLT
2      C      ROUTINE TO PLOT PARTICLES
3      C
4      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
5      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1975
6
7      C      ***** BEGIN COMDECK PARAM *****  

8      *      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
9

```

```

10      1 NLCP3,NLCP4,IFLMSZ
11      * ===== END COMDECK PARAM =====
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),DLSRQ(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),GR8V(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,DTPOS,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
32      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNP,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,N8C,NUMIT,ZORIG,OM,OMCYL,PXCONV
34      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
35      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TG,IST,
38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DT8V,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMPYT,ROMFYB,JDUMP,TWTHR,DTR,MASS,DTVSAV,DTCSAV,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IPD
47      * ===== END COMDECK YAQSC =====
48      * ===== BEGIN COMDECK YAQEQ =====
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),
50      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSRQ,VG),(AASC(26),GR8V),(AASC(27),GRROP,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ===== END COMDECK YAQEQ =====
63      * ===== END COMDECK YSTORE =====
64      * ===== BEGIN COMDECK PCALL =====
65      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
66      * ===== END COMDECK PCALL =====

```

```

67      COMMON/XTENC/XTEN,YTEN
68      DATA BCD/1H /
69      C
70      C      ADVANCE FILM TO NEXT FRAME
71      C
72      C      CALL ADV(1)
73      C
74      C      INITIALIZE PLOT VALUES
75      C
76      IPXL=65
77      IPXR=660
78      IPYB=900
79      FIPXL=65.
80      FIPXR=660.
81      FIPYB=900.
82      YUP=PTOP+2,*PRITE
83      YLB=PROTM-3,*PRITE
84      PYCNVP=FLOAT(IPYT-IPYB)/(YUP-YLB)
85      PXCONV=PXC0NV*PYCNVP/PYCONV
86      PYCONV=PYCNVP
87      C
88      C      SET UP CALL TO TICBOX
89      C
90      XCONVP=PXCONV
91      YCONVP=PYCONV
92      IXLSV=IXL
93      IXRSV=IXR
94      IYBSV=IYB
95      IYTSV=IYT
96      IXL=IPXL
97      IXR=IPXR
98      IYB=IPYB
99      IYT=IPYT
100     FIYBSV=FIYB
101     FIXLSV=FIXL
102     FIYB=FIOPYB
103     FIXL=FIPXL
104     C
105     C      DRAW AND LABEL THE PLOT FRAME
106     C
107     C      CALL TICBOX
108     C
109     C      LABEL THE PARTICLE PLOT
110     C
111     CALL LINCNT(60)
112     WRITE(IFD,100) PXR,PYR,PYT
113     WRITE(IFD,110) JNM,NAME,T,NCYC
114     C
115     C      LOOP OVER ALL THE PARTICLES AND PLOT THEM
116     C
117     C      IECP IS THE LCM INDEX
118     C      NPPT IS THE PARTICLE COUNTER
119     C
120     IECP=1
121     NPPT=0
122     C
123     C      BRING A BUFFER=LOAD OF PARTICLE DATA FROM LCM INTO SCM

```

```

124      C   10 CALL ECRD(AASC,NLCP1+IECP=1,LPB, IDUM)
125      C   KP IS THE SCM INDEX
126      C   KP=1
127      C   SKIP THE PARTICLE IF ITS X=COORDINATE IS NEGATIVE
128      C
129      C   20 IF(XPAR(KP),LT.0.) GO TO 30
130      C
131      C   CALCULATE THE RASTER COORDINATES OF THE PARTICLE
132      C
133      C   IX1=FIPXL+(XPAR(KP)-PXL)*PXCONV
134      C   IY1=FIPYB+(YPAR(KP)-YLB)*PYCONV
135      C
136      C   PLOT IT UNLESS IT IS OUTSIDE OF THE PLOTTING RECTANGLE
137      C
138      C   IF(IX1.GT.IXR.OR.IX1.LT.IXL) GO TO 30
139      C   IF((IY1.GT.IYH).OR.(IY1.LT.IYT)) GO TO 30
140      C   CALL PLT(IX1,IY1,42)
141      C   CALL PLT(IX1,IY1,42)
142      C   CALL PLT(IX1,IY1,42)
143      C
144      C   INCREMENT THE PARTICLE COUNTER AND SEE IF ALL HAVE BEEN PLOTTED
145      C
146      C   30 NPPT=NPPT+1
147      C   IF(NPPT,EQ.NPT) GO TO 40
148      C
149      C   NO. INCREMENT THE SCM POINTER AND SEE IF THE BUFFER NEEDS TO
150      C   BE REFILLED.
151      C
152      C   153 KP=KP+3
153      C   IF(KP,LT,LPB) GO TO 20
154      C
155      C   YFS. INCREMENT THE LCM INDEX AND GO REFILL THE SCM BUFFER,
156      C
157      C   159 IECP=IECP+LPB
158      C   GO TO 10
159      C
160      C   ALL PARTICLES HAVE BEEN PLOTTED, SEE IF ANY TIME=DEPENDENT
161      C   PLOTTING IS DESIRED.
162      C
163      C
164      C   164 40 CONTINUE
165      C   IF(IST,LE,0) GO TO 80
166      C
167      C   YES. SEE IF THERE HAVE BEEN AT LEAST TWO TIME PERIODS SAVED,
168      C
169      C   170 IF(NILNG,LT,2) GO TO 80
170      C
171      C   YES. PREPARE TO DO THE TIME=DEPENDENT PARTICLE PLOT.
172      C
173      C   174 CALL ADV(1)
174      C   CALL TICBOX
175      C   NIST=NPT/IST
176      C   NTOT=NPT/NIST
177      C   DO 70 I=1,NTOT

```

```

181      DO 60 J=1,NILNG
182      C
183      C      SAVE THE LAST POSITION IN TIME
184      C
185      IX3=IX2
186      IY3=IY2
187      IOK3=IOK2
188      IBEGIN=NLCP1+NLCP2+2*(NP1*(J=1)+I=1)
189      CALL ECRD(XTEN,IBEGIN,2,JDUM)
190      C      IX2=FIPXL+(PAX(I,J)=XL)*PXCONV
191      C      IX2=FIPXL+(XTEN-XL)*PXCONV
192      C      IY2=FIPYB+(PAY(T,J)=YLB)*PYCONV
193      C      IY2=FIPYB+(YTEN-YLB)*PYCONV
194      C      IOK2=0
195      IF(IX2.GT.IXR.OR.IX2.LT.IXL.OR.IY2.GT.IYB.OR.IY2.LT.IYT) IOK2=1
196      C
197      C      MARK THE INITIAL POSITION WITH A STAR
198      C
199      IF(J.NE.1) GO TO 50
200      IF(IOK2.EQ.0) CALL PLT(IX2,IY2,44)
201      GO TO 60
202      50 CONTINUE
203      C
204      C      MARK PARTICLES WITH A DOT
205      C
206      IF(IOK2.EQ.0) CALL PLT(IX2,IY2,27)
207      C
208      C      CONNECT THE PARTICLE WITH ITS NEIGHBOR IN TIME
209      C
210      IF(IOK2.EQ.0.AND.IOK3.EQ.0) CALL DRV(IX2,IY2,IX3,IY3)
211      60 CONTINUE
212      70 CONTINUE
213      C
214      C      LABEL THE PLOT
215      C
216      CALL LINCNT(60)
217      WRITE(IFD,90)
218      WRITE(IFD,110) JNM,NAME,T,NCYC
219      80 CONTINUE
220      C
221      C      RESTORE THE PLOT INDICES
222      C
223      IXL=IXLSV
224      IXR=IXRSV
225      IYB=IYBSV
226      IYT=IYTSV
227      FIYB=FIYBSV
228      IFXL=FIXLSV
229      RETURN
230
231      90 FORMAT(25H TIME-DEPENDENT PARTICLES)
232      100 FORMAT(10H PARTICLES/11X5H PXR=E12.5,5H PYB=E12.5,5H PYT=E12.5)
233      110 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLE=,15)
234      END

```

```

1      SUBROUTINE PCHCK
2      C
3      C      ROUTINE TO CHECK THE YAOUI LARGE AND SMALL CORE PARAMETERS
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      *      ----- BEGIN COMDECK PARAM -----
8      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
9      1 NLCPI,NLCP4,IFLMSZ
10     *      ----- END COMDECK PARAM -----
11     *      ----- BEGIN COMDECK YAOSC -----
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,AN,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
17     1 DTOC(10),DTONC(10),DTON8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
18     2 FIPYR,FIPYT,FIIXL,FIIXR,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,REZYB,RIBAR,RIBJB,
24     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
25     COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PHOTM,
26     1 ILNG,NILNG,TP3,TUPOT,TDOSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
27     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
28     3 XCNVLD,FIXRO,FXL0,IXRO,IXL0,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
29     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVS,DTCSAV,DTCSAV,DTVS
30     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
31     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
32     COMMON/YSC2/ZZ
33     C      COMMON/YSC4/ITAR(ITABP)
34     COMMON/YSC4/ITAB(1000)
35     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
36     *      ----- END COMDECK YAOSC -----
37     C      CHECK TO MAKE SURE THE SCM BUFFER WILL BE LARGE ENOUGH
38     C
39     IF(3*NQI.GT,NSCP1) CALL UNCLE(4,5HPCCHK,33,
40     1 33HNOT ENOUGH SMALL CORE FOR BUFFERS)
41     C
42     C      MAKE SURE LCM IS LARGE ENOUGH TO HOLD ALL THE ARRAYS
43     C
44     IF(NLC.GT.NLCPI) CALL UNCLE(4,5HYASET,14,14HNOT ENOUGH LCM)
45     RETURN
46     END

```

```

1      SUBROUTINE PRTGEN

```

```

2      C
3      C      ROUTINE TO GENERATE 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=3,1974
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 YSTORE -----
13     *      ----- BEGIN COMDECK YAQDIM -----
14     DIMENSION X(1),XPAR(1),R(1),YPAR(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),MC(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,AP,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTA,DTFAC,
28     1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
30     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
31     4 JUNF,IXL,IXR,IYB,IYT,J,JBAR
32     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(B),
33     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIH,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
35     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36     COMMON/YSC2/TUQI,TUST,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,LEVEL,TQ,IST,
38     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,UTSV,DTLAST,FIYR0,IYBO,YCNVLD,
39     3 XCNVLD,FIXR0,FIXL0,IXR0,IXL0,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40     4 ROMFXR,ROMFYR,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
41     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43     COMMON/YSC2/ZZ
44     C      COMMON/YSC4/ITAB(ITABP)
45     COMMON/YSC4/ITAB(1000)
46     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
47     *      ----- END COMDECK YAQSC -----
48     *      ----- BEGIN COMDECK YAQD -----
49     EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(A
50     1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
51     1 (B),F,FTIL,AREA,XR13K),
52     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53     3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54     4 ,UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55     5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56     6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57     7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
58     8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(

```

```

59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ----- END COMDECK YAQEQ ----- *
63      * ----- END COMDECK YSTORE ----- *
64      DIMFNSION PARTN(3,7)
65      DATA (PARTN(I),I=1,21)/21*0/
66      DATA IEFLAG/0/,NF/0/
67      INTEGER COMP,AND,OR
68      C
69      C SET UP THE NAMLIST INPUT TABLE
70      C
71      ASSIGN 170 TO IERRT
72      CALL TABDEF(PARTN,5HPARTN,7,IERRT)
73      CALL TABSET(PARTN,5HDPAR,DRPAR,IEFLAG,0,0,0,0)
74      CALL TABSET(PARTN,5HDZPAR,DZPAR,IEFLAG,0,0,V,V)
75      CALL TABSET(PARTN,2HXC,XC,IEFLAG,0,0,0,0)
76      CALL TABSET(PARTN,2HYC,YC,IEFLAG,0,0,V,0)
77      CALL TABSET(PARTN,2HXD,XD,IEFLAG,0,0,0,0)
78      CALL TARSFT(PARTN,2HYD,YD,IEFLAG,V,0,0,0)
79      C
80      C INITIALIZE PARTICLE COUNT
81      C
82      NPT=0
83      C
84      C DEFINE CONSTANT
85      C
86      NQI2=NQI*2
87      C
88      C LPB IS THE LENGTH OF ONE SMALL CORE BUFFER ROUNDED DOWN TO THE
89      C NEAREST MULTIPLE OF THREE. PARTICLE DATA WILL BE STORED IN
90      C A SMALL CORE BUFFER IN THREE WORD BLOCKS.
91      C
92      LPB=NQI/3*3
93      C
94      C KP IS THE SCM BUFFER SUBSCRIPT
95      C
96      KP=1
97      C
98      C IECP IS THE CURRENT LCM READ ADDRESS
99      C
100     IECP=1
101     C
102     C IF THE MESH IS VARIABLE, RECALCULATE THE PARTICLE DR AND DZ
103     C
104     IF(FREZXR,EQ,1.,AND,FREZYT,EQ,1.,AND,FREZYB,EQ,1,) GO TO 10
105     C
106     C IT IS, COMPUTE XMAX,YMAX, AND YMIN BASED ON GEOMETRIC PROGRESSION
107     C RELATIONS. THE FIRST IUNF CELLS ARE OF WIDTH DR AND CONTRIBUTE
108     C IUNF*DR TO XMAX. THIS INCLUDES CELLS I=1 TO I=IUNF. CELL
109     C IUNF+1 HAS WIDTH FREZ*DR, CELL IUNF+2 HAS WIDTH FREZ*(FREZ*DR),
110     C ETC. THE LAST CELL I=IBAR WILL HAVE WIDTH FREZ**((IBAR-IUNF)*DR
111     C AND THE SUM WILL BE THE PROGRESSION
112     C
113     C   FREZ*DR + FREZ**2*DR + ... + FREZ**((IBAR-IUNF)*DR
114     C   = FREZ*(1 + FREZ + FREZ**2 + ... + FREZ**((IBAR-IUNF-1)*DR,
115     C   = FREZ*DR*(1-FREZ**((IBAR-IUNF)))/(1-FREZ),
116     C   THUS, XMAX = IUNF*DR + FREZ*DR*(1-FREZ**((IBAR-IUNF)))/(1-FREZ).

```

```

116      C      YMAX AND YMIN ARE COMPUTED IN A SIMILAR FASHION.
117      C
118      IF(FREZXR.GT.1.) XMAX=(FLOAT(IUNF)+FREZXR*(1.-FREZXR**(IBAR-IUNF))
119      1.*ROMFXR)*DR
120      IF(FREZYT.GT.1.) YMAX=REZY0+(FLOAT(JUNFO2)+FREZYT*(1.-FREZYT**(
121      1.JBAR-JCEN-JUNFO2))*ROMFYT)*DZ
122      IF(FREZYB.GT.1.) YMIN=REZY0-(FLOAT(JUNFO2)+FREZYB*(1.-FREZYB**(
123      1.JCEN-JUNFO2))*ROMFYB)*DZ
124      C
125      C      READ IN QUANTITIES FOR ONE PARTICLE REGION, FIRST SET DEFAULT
126      C      VALUES.
127      C
128      10 CONTINUE
129      C
130      C      DRPAR IS THE R DIMENSION OF THE RECTANGLE IN THE MIDDLE OF WHICH
131      C      ONE PARTICLE WILL BE PLACED
132      C
133      DRPAR=0.
134      C
135      C      DZPAR IS THE Z RECTANGULAR DIMENSION
136      C
137      DZPAR=0.
138      C
139      C      XC,YC,XD, AND YD ARE THE DIMENSIONS OF THE ENTIRE PARTICLE REGION.
140      C      IF THE REGION IS RECTANGULAR,(XC,YC) IS THE LOWER LEFT-HAND
141      C      CORNER AND (XD,YD) IS THE UPPER RIGHT-HAND CORNER. IF THE
142      C      REGION IS CIRCULAR,YD=0.,XC IS UNUSED,YC IS THE Z VALUE OF
143      C      THE CENTER OF THE CIRCLE (ASSUMED TO BE ON THE AXIS OF
144      C      SYMMETRY),AND XD IS THE RADIUS OF THE CIRCLE.
145      C
146      XC=0.
147     YC=0.
148      XD=0.
149      YD=0.
150      C
151      C      DO THE ACTUAL READ
152      C
153      CALL NAMLST(PARTN,5,IEFLAG)
154      C
155      C      CHECK FOR INPUT ERRORS
156      C
157      IF(IEFLAG.NE.0) CALL UNCLE(4,6HPRTGEN,26,
158      1 26HPARTN NAMLIST INPUT ERROR)
159      C
160      C      SEE IF THIS WAS THE LAST PARTICLE REGION, IF SO,WE ARE ALL DONE.
161      C
162      IF(DRPAR.LE.0.) GO TO 140
163      C
164      C      NO. CHECK FOR INPUT ERRORS.
165      C
166      IF(DZPAR.LE.0.) CALL UNCLE(4,6HPRTGEN,14,14HERROR IN DZPAR)
167      IF(YD.EQ.0.,AND,XD.LE.0.) CALL UNCLE(4,6HPRTGEN,11,11HERROR IN XD)
168      C
169      C      NO ERRORS, PRINT OUT THE INPUT VARIABLES.
170      C
171      DO 20 IPX=6,IFD,6
172      20 WRITE(IPX,18N) DRPAR,DZPAR,XC,YC,XD,YD

```

```

173 C SEE IF PARTICLE REGION IS RECTANGULAR OR CIRCULAR
174 C
175 C IF(YD,EQ.0.) GO TO 40
176 C
177 C REGION IS RECTANGULAR, SET PARTICLE REGION BOUNDS. IF VARIABLE
178 C ZONING IS BEING USED, PARTICLE REGION BOUNDS ARE SET TO
179 C PROBLEM BOUNDS.
180 C
181 C IF(FREZXR,EQ.1.,AND,FREZYT,EQ.1.,AND,FREZYB,EQ.1.) GO TO 30
182 C
183 C VARIABLE ZONING
184 C
185 C
186 C PTOP=YMAX
187 C PBOTM=YMIN
188 C PRITE=XMAX
189 C PLEFT=0,
190 C GO TO 50
191 C
192 C UNIFORM ZONING
193 C
194 C 30 CONTINUE
195 C PTOP=YD
196 C PBOTM=YC
197 C PRITE=XD
198 C PLEFT=XC
199 C GO TO 50
200 C
201 C REGION IS CIRCULAR
202 C
203 C 40 CONTINUE
204 C PTOP=YC+XD
205 C PBOTM=YC-XD
206 C PRITE=XD
207 C PLEFT=0,
208 C R2=XD**2
209 C 50 CONTINUE
210 C
211 C INITIALIZE THE OLD PTOP
212 C
213 C PTPOLD=PTOP
214 C
215 C BEGIN LOOP OVER PARTICLES
216 C
217 C INITIALIZE THE Y COORDINATE
218 C
219 C YTE=PBOTM+.5*DZPAR
220 C 60 CONTINUE
221 C
222 C INITIALIZE THE X=COORDINATE
223 C
224 C XTE=PLEFT+.5*DRPAR
225 C 70 CONTINUE
226 C
227 C IF PARTICLE REGION IS CIRCULAR, AN EXTRA CHECK MUST BE MADE TO
228 C SEE IF THE PARTICLE IS IN THE REGION
229 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      STORE THE PARTICLE COORDINATES AND THE PARTICLE MASS
234      C      XPAR(KP)=XTE
235      C      YPAR(KP)=YTE
236      C      MPAR(KP)=0.
237
238      C      INCREMENT THE SCM BUFFER SUBSCRIPT AND THE PARTICLE COUNT
239      C      KP=KP+3
240      C      NPT=NPT+1
241
242      C      SEE IF THE SCM BUFFER NEEDS TO BE FLUSHED
243
244      C      IF(KP,GT,LPB) GO TO 110
245
246      C      NO, INCREMENT THE PARTICLE X-COORDINATE,
247
248      C      90 XTE=XTE+DRPAR
249
250      C      SEE IF WE HAVE GONE OUTSIDE OF THE PARTICLE REGION
251      C      IF(XTE,LE,PRITE) GO TO 70
252
253      C      YES, INCREMENT THE PARTICLE Y-COORDINATE,
254
255      C      100 YTE=YTE+DZPAR
256
257      C      SEE IF Y HAS GONE OUTSIDE THE PARTICLE REGION
258
259      C      IF(YTE,LE,PTOP) GO TO 60
260
261      C      YES, WE ARE DONE WITH THIS PARTICLE REGION, GO READ IN ANOTHER
262      C      CARD.
263
264      C      GO TO 10
265
266      C      FLUSH THE BUFFER TO LCM IF THERE IS ROOM
267
268      C      110 CONTINUE
269      C      IF(IECP+LPB,LF,NLCP2+NLCP1) GO TO 130
270      C      120 CALL UNCLE(4,6HPRTGEN,25,25HPARTICLE ARRAY OVERFLOWED)
271      C      130 CONTINUE
272      C      CALL ECWR(AASC,NLCP1+IECP-1,LPB,IDUM)
273
274      C      INCREMENT THE LCM ADDRESS,RESET THE SCM SUBSCRIPT,AND PROCFED
275
276      C      IECP=IECP+LPB
277      C      KP=1
278      C      GO TO 90
279
280      C      ALL DONE, FLUSH THE SCM BUFFER TO LCM IF THERE IS ROOM,
281
282      C      140 CONTINUE

```

```

287      NW=KP=1
288      IF(NW,EQ,0) GO TO 150
289      IF(IECP+NW.GT,NLCP2+NLCP1) GO TO 120
290      CALL FCWR(AASC,NLCP1+IECP-1,NW,IDUM)
291 150 CONTINUE
292 C
293 C      NPS IS THE NO. OF WORDS OF PARTICLE STORAGE
294 C
295 C      NPS=3*NPT
296 C
297 C      PRINT OUT THE NO. OF PARTICLES GENERATED AND THEIR TOTAL MASS
298 C
299 DO 160 IPX=6,IFD,6
300 160 WRITE(IPX,190) NPT,MT
301      RETURN
302 170 CONTINUE
303      CALL UNCLE(4,6HPRTGEN,35,35HPARTN NAMELIST INITIALIZATION ERROR)
304 C
305 180 FORMAT(8H DRPAR=1PE12.5,8H DZPAR=F12.5,5H XC=E12.5/5H YC=E12.5
306 1 ,5H XD=E12.5,5H YD=E12.5)
307 190 FORMAT(4XI6,38H PARTICLES GENERATED, WITH TOTAL MASS=1PE12.5)
308 END

```

```

1      SUBROUTINE PRTMOV
2      C
3      C      ROUTINE TO MOVE 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,
10     1 NLCP3,NLCP4,IPLMSZ
11     *      ----- END COMDECK PARAM -----
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),AVYSV(1),AVYSV(1),DLSROI(1),DLSROG(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),DKL8M(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,DT,DTFAC,
28     1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EPs,FIPXL,FIPXR,

```

```

29      2 FIPYB,FIPYT,FIYL,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,IVB,IVT,J,JBAR
32      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
35      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37      1 TLNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,GLEVEL,TQ,IST,
38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRD,FIXLD,IXRD,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMFYR,JDUMP,TWTHR,DTR,DMASS,DTVSAT,DTCSAV,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOD,TMAX,TGMAX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXFF,RMINEF,TSTRTD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTRT,FTLM,PAPER,IPD,IFD
47      * ---- END COMDECK YAQSC -----
48      * ---- BEGIN COMDFCK YAQEQ -----
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PHY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ---- END COMDECK YAQEQ -----
63      * ---- END COMDECK YSTORE -----
64      * ---- BEGIN COMDECK PCOM -----
65      COMMON/XTENC/XTEN,YTEN,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
66      EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
67      EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
68      * ---- END COMDECK PCOM -----
69      DATA YDUM/0/
70      PTPOLD=PTOP
71      NIST=NPT/IST
72      C
73      C DECIDE WHETHER TO ADD ANOTHER POINT TO THE TIME-DEPENDENT
74      C PARTICLE DATA
75      C
76      IFLAGP=0
77      IF(NCYC.LE.1) GO TO 10
78      IF(T+EM10.LT.TOUT) GO TO 20
79      10 CONTINUE
80      C
81      C YES
82      C
83      NILNG=NILNG+1
84      IFLAGP=1
85      20 CONTINUE

```

```

86      IECP=1
87      NPPT=0
88      JOLD=0
89      PROT=1.E+10
90      PRITE=1.E10
91      PTOP=1.E10
92      30 CALL ECRD(AASC,NLCP1+IECP=1,LPB,IDUM)
93      KP=1
94      40 NPPT=NPPT+1
95      MPTST=0
96      XTE=XPAR(KP)
97      IF(XTE,LT,0.) GO TO 130
98     YTE=YPAR(KP)
99      ICEL=ITAB(NPPT)
100     J=ICEL/IP1+1
101     I=ICEL-(J-1)*IP1
102     IF(J,EQ,JOLD) GO TO 50
103     JOLD=J
104     IEC=(J-1)*NQI
105     CALL ECRD(AASC(ISC2),IEC,NQI2,IDUM)
106     50 IJ=(I-1)*NQ+ISC2
107     IPJ=IJ+NQ
108     IJP=IJ+NQI
109     IPJP=IPJ+NQI
110     X1=X(IPJ)
111     Y1=Y(IPJ)
112     X2=X(IPJP)
113     Y2=Y(IPJP)
114     X3=X(IJP)
115     Y3=Y(IJP)
116     X4=X(IJ)
117     Y4=Y(IJ)
118     60 XTEN=XTE
119     YTEN=YTE
120     CALL PSUB1(IN)
121     IF(IN,NE,0) GO TO 70
122     XTE=0.25*(X1+X2+X3+X4)
123    YTE=0.25*(Y1+Y2+Y3+Y4)
124     GO TO 160
125     70 U1=U(IPJ)
126     V1=V(IPJ)
127     U2=U(IPJP)
128     V2=V(IPJP)
129     U3=U(IJP)
130     V3=V(IJP)
131     U4=U(IJ)
132     V4=V(IJ)
133     X1Z=X1-XTE
134     X41=X4-X1
135     X21=X2-X1
136     X3421=X3-X4-X21
137     Y1Z=Y1-YTE
138     Y41=Y4-Y1
139     Y21=Y2-Y1
140     Y3421=Y3-Y4-Y21
141     XMU=Y3421*X21-Y21*X3421
142     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-U1*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=YTF
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 PTRBDF(XTEN,YTEN)
171      XPAR(KP)=XTEN
172      YPAR(KP)=YTEN
173
174      C SEE IF TIME-DEPENDENT PARTICLE DATA IS TO BE COLLECTED THIS CYCLE
175      C
176      C IF(IFLAGP,EQ,0) GO TO 120
177      C
178      C YES, SEE IF THE CURRENT PARTICLE IS TO BE SAVED,
179      C
180      C 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,IDUM)
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,IDUM)
197      IECP=IECP+LPB
198      GO TO 30
199      140 CALL ECWR(AASC,NLCP1+IECP=1,LPB,IDUM)

```

```

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  X1=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      JOLD=J
245      CALL ECRD(AASC(ISC2),IEC,NQI2,IDUM)
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      WRITE(IPX,200) 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      WRITE(IPX,270) NPPT,ITAB(NPPT)
273      GO TO 130
274  C   270 FORMAT(5X,8HPARTICLE,I10,2SHOSED OUT, CELL NUMBER,I10)
275 280 FORMAT(5X,14HERROR PARTICLE,IS,6H CELL,IS,12F7.4)
276
277      FND

```

```

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 COMDFCK YSTORE -----
7      *      ----- BEGIN COMDFCK YAQDIM -----
8      DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC(
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),RDL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
12     4 (1),SIG(1),TJS(1),GRROR(1),GRROZ(1),GRROP(1),TURVEC(1),MTIL(1),
13     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GR8V(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,AN,A0FAC,ADM,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
22     1 DTO(10),DTOC(10),DTO2,DTOB,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
23     2 FIPYR,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,IYH,IYT,J,JBAR
26     COMMON/YSC2/JCEK,JP1,JP2,JP4,JUNFO2,KXI,LAM,LPR,MU,NAME(8),
27     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,QMCYL,PXCONV
28     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
29     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN

```

```

30      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
31      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
32      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTBV,DTLAST,FIYBO,IYBO,YCNVLD,
33      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,J8VW,QMN,QMX,WMAX,JNM,T2,TLIM,
34      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
35      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMAX,ITM,JTM,ITG,JTG
36      6 ,TMASSV,WMAXF,F,RMINEF,TSTRTD
37      COMMON/YSC2/ZZ
38      C      COMMON/YSC4/ITAB(ITABP)
39      COMMON/YSC4/ITAB(1000)
40      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
41      *      **** END COMDECK YAQSC ****
42      *      **** BEGIN COMDECK YAQEQ ****
43      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC(8),E,ETIL,AREA,XR13K),
44      2 (AASC(15),SIF),(AASC(16),PM0,DKLSM,RMP),(AASC(9),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
45      3 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(21),GRROR),(AASC(22),GRR0Z),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV),
46      5 ,(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
47      7 ),(AASC(28),MTJL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
48      9 AVYSV,X24K)
49      REAL M,MP,MPAR,MTIL
50      *      **** END COMDECK YAQEQ ****
51      *      **** END COMDECK YSTORE ****
52      *      **** BEGIN COMDECK PCOM ****
53      COMMON/XTEN/XTEN,YTEX,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
54      EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
55      EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
56      *      **** END COMDECK PCOM ****
57      DTIMENSION AMZ(7)
58      DATA PI/3.1415926535897932384626/
59      DATA PI02/1.5707963267948966192313/
60      DATA PI02M1/0.5707963267948966192313/
61      DATA PI2/6.2831853071795864769252/
62      10 AMZMIN=1.E+10
63      DO 60 K=1,4
64      IF(X1(K).EQ.XTEN) GO TO 40
65      XX=(Y1(K)-YTEX)/(X1(K)-XTEN)
66      YY=ABS(XX)
67      IF(YY.GT.1.) GO TO 20
68      SLOPE=PI02M1*XX
69      GO TO 30
70      20 SLOPE=SIGN(PI02-1./YY,XX)
71      30 IF(XTEN.GT.X1(K)) SLOPE=SLOPE+PI
72      IF(XTEN.LT.X1(K).AND.SLOPE.LT.0.) SLOPE=SLOPE+PI2
73      GO TO 50
74      40 SLOPE=SIGN(PI02,Y1(K)-YTEX)
75      50 AMZ(K)=SLOPE
76      60 AMZMIN=AMIN1(AMZMIN,AMZ(K))
77      AMZ(5)=AMZ(1)-AMZMIN
78      AMZ(1)=AMZ(5)
79      AMZ(6)=AMZ(2)-AMZMIN
80      AMZ(2)=AMZ(6)

```

```

87      AMZ(7)=AMZ(3)=AMZMIN
88      AMZ(3)=AMZ(7)
89      AMZ(4)=AMZ(4)=AMZMIN
90      IF(AMZ(1),EQ,0,) K=1
91      IF(AMZ(2),EQ,0,) K=2
92      IF(AMZ(3),EQ,0,) K=3
93      IF(AMZ(4),EQ,0,) K=4
94      INTE=0
95      IN=0
96      IF(AMZ(K+3),GT,AMZ(K+2),AND,AMZ(K+2),GT,AMZ(K+1),AND,AMZ(K+1),GT,
97      1 AMZ(K)) INTE=1
98      IF(INTE,EQ,0) RETURN
99      IN=1
100     DO 100 K=1,4
101     AMZ1=AMZ(K)
102     AMZ2=AMZ(K+1)
103     AMZ3=AMZ(K+2)
104     IF(AMZ1*AMZ2,NE,0,) GO TO 70
105     AMXZ=AMAX1(AMZ1,AMZ2)
106     IF(AMZ3,LT,AMXZ) GO TO 80
107     PHI=ARS(AMZ1=AMZ2)
108     GO TO 90
109     80 PHI=PI2-AMXZ
110     90 IF(PHI,GT,PI) IN=0
111     IF(IN,EQ,0) RETURN
112     100 CONTINUE
113     RETURN
114     END

```

```

1      SUBROUTINE PSUB2(IT,IFLAG)
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),YPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
9      1),V(1),VG(1),RD(1),SIE(1),MP(1),RMP(1),RCSD(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),DKL8M(1),AREA(1)
15     *      END COMDECK YAQDIM      *****
16     *      BEGIN COMDECK YAGSC      *****
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),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,

```

```

23      2 FIPYB,FIPYT,FIYL,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 JUNF,IXL,IXR,IYB,IYT,J,JBAR
26      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,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,REZYB,RIBAR,RIBJB,
29      3 FRFZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
30      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,NEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
31      1 ILNG,NILNG,TP3,TUPOT,TQGSAV,TK,TI,TUGENG,EP1,SAV1,OLEVEL,TQ,IST,
32      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
33      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
34      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
35      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITB,JTG
36      6 ,TMASS,WMAXEF,RMINEF,TSTRTD
37      COMMON/YSC2/ZZ
38      C COMMON/YSC4/ITAB(ITABP)
39      COMMON/YSC4/ITAB(1000)
40      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
41      * **** END COMDECK YAQEQ ****
42      * **** BEGIN COMDECK YAQEQ ****
43      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
44      1 AASC(4),U),(AASC(5),V),(AASC(6),RD),(AASC(7),DELSM,RCSQ,MP),(AASC
45      1 (8),E,ETIL,AREA,XR13K),
46      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
47      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
48      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
49      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
50      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLROI,Y13K),(AASC(24),GZSV
51      7 ),(AASC(25),DLROI,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUGVEC,
52      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
53      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
54      1 AVYSV,X24K)
55      REAL M,MP,MPAR,MTIL
56      * **** END COMDECK YAQEQ ****
57      * **** END COMDECK YSTORE ****
58      * **** BEGIN COMDECK PCOM ****
59      COMMON/XTEN/XTEN,YTEN,XTE,YTE,X1(4),Y1(4),XX1,YY1,XX2,YY2
60      EQUIVALENCE(X1(2),X2),(X1(3),X3),(X1(4),X4)
61      EQUIVALENCE(Y1(2),Y2),(Y1(3),Y3),(Y1(4),Y4)
62      * **** END COMDECK PCOM ****
63      DATA ALF,ALF2/0.001,1.00000001/
64      10 IT=0
65      IFLAG=0
66      IF((YTEN=YY2)*(XTEN=XX1),EQ,(YTEN=YY1)*(XTEN=XX2)) GO TO 30
67      IF(ABS(XTEN-XTE).LT.EM10) GO TO 50
68      PSL=(YTEN-YTE)/(XTEN-XTE)
69      IF(ABS(XX1-XX2).LT.EM10) GO TO 60
70      GSL=(YY2-YY1)/(XX2-XX1)
71      IF(PSL.EQ.GSL) RETURN
72      XINS=(YY1-YTE+PSL*XTE-GSL*XX1)/(PSL-GSL)
73      YINS=YY1+GSL*(XINS-XX1)
74      20 ITA=0
75      IF(((XX1=XINS)*(XX2=XINS),LE.0.),AND,((YY1=YINS)*(YY2=YINS),LE.0.))
76      1 ) ITA=1
77      IF(((XTE=XINS)*(XTEN=XINS),LE.0.),AND,((YTE=YINS)*(YTEN=YINS),LE.
78      1 0.)) ITA=ITA+1
79      IF(ITA.EQ.2) GO TO 70

```

```

80      RETURN
81 30 XTEN=ALF2*Xten+(1.-ALF2)*XTE+ALF
82      YTEN=ALF2*YTEN+(1.-ALF2)*YTE
83      XPAR(KP)=XTEN
84      YPAR(KP)=YTEN
85      IFLAG=1
86      DO 40 IPX=IPD,IFD,6
87      40 WRITE(IPX,100) IFLAG
88      RETURN
89 50 XINS=XTE
90      YINS=YY1+(YY1-YY2)/(XX1-XX2)*(XTE-XX1)
91      GO TO 20
92 60 XINS=XX1
93      YINS=YTE+PSL*(XX1-XTE)
94      GO TO 20
95 70 IF((ABS(XTEN-XINS),LT,1.E-07),AND,(ABS(YTEN-YINS),LT,1.E-07)) GO
96      1 TO 80
97 C     WE NEED TO SCALE ALL ALPHAS
98      XTF=XINS+ALF*(XTEN-XINS)
99      YTE=YINS+ALF*(YTES-YINS)
100     IT=1
101     RETURN
102 80 XPAR(KP)=XTEN-1,E=6*(XINS-XTE)
103      YPAR(KP)=YTEN-1,E=6*(YINS-YTE)
104      IFLAG=2
105      DO 90 IPX=IPD,IFD,6
106      90 WRITE(IPX,100) IFLAG
107 C     IF PARTICLE LANDS ON OR NEAR BOUNDARY, PULL IT BACK TO ORIGINAL
108 C     CELL AND GO ON TO NEXT PARTICLE
109     RETURN
110 C
111 100 FORMAT(5X,7HIFLAG *,IS)
112 END

```

```

1      SUBROUTINE PTRBDF(XJ,YJ)
2      C
3      C     ROUTINE TO ADD TURBULENT DIFFUSION EFFECTS TO THE PARTICLE
4      C     MOTION
5      C
6      C     (XJ,YJ) IS THE POSITION OF THE PARTICLE AFTER
7      C     THE EFFECTS OF CONVECTION HAVE BEEN ADDED. THE POSITION
8      C     WILL BE RETURNED WITH THE TURBULENT DIFFUSION CORRECTION
9      C     ADDED.
10     C
11     C     WRITTEN BY J.L.NORTON, LASL T-3, 1975
12     C
13     *----- BEGIN COMDECK YSTORE -----*
14     *----- BEGIN COMDFCK YADDM -----*
15     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
16     1 1),V(1),VG(1),RO(1),SIF(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
2     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)

```

```

18      3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSRQ(1),CAPGAM(1),TUQ
19      4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
20      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
21      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
22      * ----- END COMDECK YAQDIM -----*
23      * ----- BEGIN COMDECK YAQSC -----*
24      LOGICAL RESTRT,FILM,PAPER,TURB
25      REAL LAM,MU
26      C COMMON/YSC1/AASC(NSCP1)
27      COMMON/YSC1/AASC(9600)
28      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
29      1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
30      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
31      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
32      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
33      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
34      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
35      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
36      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
37      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
38      1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
39      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
40      3 XCNVLD,FIXRO,IXRO,IXLO,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLIM,
41      4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTR,TMASS,DTVS,DTCSAV,DTCSAV,IDTV
42      5 ,JDTV,JDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
43      6 ,TMASSV,WMAXEF,RMINEF,TSTRD
44      COMMON/YSC2/ZZ
45      C COMMON/YSC4/ITAB(1TABP)
46      COMMON/YSC4/ITAB(1000)
47      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
48      * ----- END COMDECK YAQSC -----*
49      * ----- BEGIN COMDECK YAQE0 -----*
50      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
51      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RC5Q,MP),(AASC
52      1 (8),E,ETIL,AREA,XR13K),
53      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
54      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
55      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
56      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
57      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
58      7 ),(AASC(25),DLSRQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
59      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
60      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
61      1 AVYSV,X24K)
62      REAL M,MP,MPAR,MTIL
63      * ----- END COMDECK YAQE0 -----*
64      * ----- END COMDECK YSTORE -----*
65      * ----- BEGIN COMDECK TRBDIF -----*
66      COMMON/CTDIF/FRFV(21),NERFV,DXF
67      * ----- END COMDECK TRBDIF -----*
68      C IPASS=1 IS FOR X (R)
69      C IPASS=2 IS FOR Y (Z)
70      C
71      C IPASS=1
72      C SKIP THE R CORRECTION IF XJ,LE,RMINEF
73      C
74      C

```

```

75      IF(XJ,LE,RMINEF) GO TO 50
76      C
77      C      GET A RANDOM NO. FROM A UNIFORM DISTRIBUTION
78      C      ON THE INTERVAL (0.,1.)
79      C
80      10 CONTINUE
81      XINPUT#0.
82      ETA1=RNUMF(XINPUT)
83      C
84      C      MAP EACH HALF OF THE INTERVAL (0.,1.) INTO (0.,1.)
85      C
86      ETA=ABS(1.-2.*ETA1)
87      C
88      C      FIND THE ERROR FUNCTION BOUNDS ON THE RANDOM NO.
89      C
90      DO 20 L=2,NERFV
91      20 IF(ETA,LE,ERFV(L)) GO TO 30
92      C
93      C      IF NO BOUNDS ARE FOUND, THE RANDOM NO. IS TOO CLOSE TO 1.
94      C      GO BACK AND TRY ANOTHER ONE.
95      C
96      GO TO 10
97      30 CONTINUE
98      C
99      C      BOUNDS HAVE BEEN FOUND, INTERPOLATE THE ERROR FUNCTION INVERSE.
100     C
101     ERFINV=DXF*FLOAT(L=2)*(ETA-ERFV(L=1))/(ERFV(L)-ERFV(L=1))
102     C
103     C      CALCULATE THE CORRECTION
104     C
105     CORR=ERFINV*2.*SQRT(DT*SIG(IJ))
106     C
107     C      ADD ON THE SIGN OF THE CORRECTION
108     C
109     IF(ETA1,GT,.5) CORR=-CORR
110     C
111     C      GO CORRECT THE PROPER COORDINATE
112     C
113     GO TO (40,60),IPASS
114     C
115     C      X (R) COORDINATE
116     C
117     40 CONTINUE
118     XJ=XJ+CORR
119     50 CONTINUE
120     C
121     C      GO BACK AND DO THE Y (Z) COORDINATE
122     C
123     IPASS#2
124     GO TO 10
125     60 CONTINUE
126     C
127     C      Y (Z) COORDINATE
128     C
129     YJ=YJ+CORR
130     RETURN
131     END

```

```

1      SUBROUTINE REZONE
2
3      C ROUTINE TO CALCULATE THE GRID VELOCITIES, NEW GRID POSITIONS,
4      C AND RESTORE THE AMBIENT ATMOSPHERE TO HYDROSTATIC
5      C EQUILIBRIUM
6
7      C ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T=3
8      C MODIFIED BY J.L.NORTON,LASL T=3,1975
9
10     * ----- BEGIN COMDECK PARAM -----
11     COMMON/P/COM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
12     NLCP3,NLCP4,IPLMSZ
13     * ----- END COMDECK PARAM -----
14     * ----- BEGIN COMDECK YSTORE -----
15     * ----- BEGIN COMDECK YAQDIM -----
16     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
17     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSQ(1),E(1),ETIL(1),RVOL
18     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CG(1),VTIL(1)
19     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSR0Q(1),CAPGAM(1),TUQ
20     4 (1),SIG(1),TUS(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
21     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
22     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
23     * ----- END COMDECK YAQDIM -----
24     * ----- BEGIN COMDECK YAQSC -----
25     LOGICAL RESTRT,FILM,PAPER,TURB
26     REAL LAM,MU
27     C COMMON/YSC1/AASC(NSCP1)
28     COMMON/YSC1/AASC(9600)
29     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
30     1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
31     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
32     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
33     4 JUNF,IXL,IXR,IYB,IYT,J,JBAR
34     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
35     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
36     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
37     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
38     COMMON/YSC2/TUAI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
39     1 ILNG,NILNG,TP3,TUPOT,TDGSBV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
40     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYB0,IYB0,YCNVLD,
41     3 XCNVLD,FXR0,FXL0,IXR0,IXL0,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
42     4 ROMFXR,ROMFYB,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAT,DTCSAV,DTV
43     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMON,TMAX,TGHX,ITM,JTM,ITG,JTG
44     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
45     COMMON/YSC2/ZZ
46     C COMMON/YSC4/ITAB(ITABP)
47     COMMON/YSC4/ITAB(1000)
48     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
49     * ----- END COMDECK YAQSC -----
50     * ----- BEGIN COMDECK YAQEQ -----
51     EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),
52     1 (AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
53     1 (8),E,ETIL,AREA,XR13K),
54     2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9

```

```

55      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
56      4 UL,PMX,PU),(AASC(13),VTIL,VL,PHY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
57      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
58      6 21),GRORR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
59      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRRD0P,TUQVEC,
60      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
61      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
62      1 AVYSV,X24K)
63      REAL M,MP,MPAR,MTIL
64      *      ===== END COMDECK YAQEQ =====
65      *      ===== END COMDECK YSTORE =====
66      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
67      FSTF=0.,01
68      ZRS=ZE=SUMRS=SUME=0.,0
69      DMN=1.,0E+20
70      CALL START
71      DO 20 J=2,JP1
72      DO 10 I=1,IBAR
73      DEL=(Y(IJP)-Y(IJ))**2+(X(IJP)-X(IJ))**2
74      DMN=AMIN1(DMN,DEL)
75      RDEL=1.,0/DEL
76      DELE=E(IJ)=0.156
77      SUME=SUME+DELE
78      ZE=ZE+Y(IJ)*DELE
79      SUMRS=SUMRS+RDEL
80      ZRS=ZRS+Y(IJ)*RDEL
81      IJ=IJ+NQ
82      IJP=IJP+NQ
83      10 CONTINUE
84      CALL LOOP
85      20 CONTINUE
86      ZE=ZE/SUME
87      ZRS=ZRS/SUMRS
88      FC3=FCX=FCP2=0.,0
89      CALL START
90      DO 40 J=2,JP1
91      DO 30 I=1,IBAR
92      IF(J,LE,4) FC3=AMAX1(FC3,ABS(VL(IJ)))
93      IF(J,GE,JP1-11) FCP2=AMAX1(FCP2,ABS(VL(IJ)))
94      IF(I,GE,IBAR-3) FCX=AMAX1(FCX,ABS(UL(IJ)))
95      IJ=IJ+NQ
96      30 CONTINUE
97      CALL LOOP
98      40 CONTINUE
99      FCP2=AMAX1(0.,0.,2*RDT*(ZORIG-(Y(IJ)-PTOP)/PRITE)*PRITE)
100     FCX=AMAX1(0.,0.,2*RDT*((ZORIG-X(IJ+NQIB)/PRITE)*PRITE))
101     CALL START
102     DO 150 J=2,JP2
103     DO 140 I=1,IP1
104     IPJ=IJ+NQ
105     IMJ=IJ=NQ
106     IF(I,EQ,1,OR,I,EQ,IP1,OR,J,EQ,2,OR,J,EQ,JP2) GO TO 50
107     XAV=X(IPJ)+X(IMJ)+X(IJP)+X(IJM)
108     UG(IJ)=0.5*(UL(IJ)+U(IJ))+FSTF*(XAV=4,*X(IJ))*RDT
109     YAV=Y(IPJ)+Y(IMJ)+Y(IJP)+Y(IJM)
110     VG(IJ)=0.5*(VL(IJ)+V(IJ))+FSTF*(YAV=4,*Y(IJ))*RDT
111     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(J,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+OMCYL
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     NLCP3,NLCP4,IFLMSZ
11     *     **** END COMDECK PARAM      *****
12     *     **** BEGIN COMDECK YAGSC      *****
13     LOGICAL RESTRT,FILM,PAPER,TURB
14     REAL LAM,MU
15     C
16     COMMON/YSC1/AASC(NSCP1)
17     COMMON/YSC1/AASC(9600)
18     COMMON/YSC2/AA(1),ANG,A0,FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
19     1 DTO(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
20     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,PREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
21     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
22     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
23     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
24     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
25     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
26     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
27     COMMON/YSC2/TUGI,TUSI,NC9,TNEG,TNEGSY,TUSV,TURB,PTOP,PRITE,PBOTM,
28     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
29     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTCSV,DTLABT,FIYBO,IYBO,YCNVLD,
30     3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,J8VW,QMN,QMX,WMAX,JNM,T2,TLM,
31     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAY,DTCSAV,IDTV
32     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOH,TMAX,TGMX,ITM,JTM,ITG,JTG
33     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
34     C
35     COMMON/YSC2/ZZ
36     COMMON/YSC4/ITAB(ITABP)
37     COMMON/YSC4/ITAB(1000)
38     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
39     *     **** END COMDECK YAGSC      *****

```

```

38      C      READ REST OF DUMP, FIRST CHECK THE PARAMETERS.
39      C      CALL PCHK
40      C      READ IN THE LCM ARRAY STORAGE
41      C
42      C      CALL LCBUFF(0,NLC,7,0,1,IERROR)
43      C      IF(IERROR,NE,0) GO TO 20
44      C      GO TO 30
45      C      10 CALL UNCLE(1,6HYARSRT,18,18HSCBUFF INPUT ERROR)
46      C      20 CALL UNCLE(4,6HYARSRT,18,18HLCBUFF INPUT ERROR)
47      C      30 CONTINUE
48      C
49      C      SEE IF THERE ARE ANY PARTICLES
50      C
51      C      IF(NPT,LE,0) GO TO 40
52      C
53      C      YES, READ THEIR RECORDS.
54      C
55      C      CALL LCBUFF(NLCP1,NPS,7,0,1,IERROR)
56      C      IF(IERROR,NE,0) GO TO 20
57      C      CALL SCBUFF(ITAB,ITABP,7,0,1,IERROR)
58      C      IF(IERROR,NE,0) GO TO 10
59      C
60      C      SEE IF THERE IS ANY TIME-DEPENDENT PARTICLE DATA
61      C
62      C      IF(NILNG,LE,0) GO TO 40
63      C
64      C      YES, READ IT IN.
65      C
66      C      CALL LCBUFF(NLCP1+NLCP2,2*NP1*NILNG,7,0,1,IERROR)
67      C      IF(IERROR,NE,0) GO TO 20
68      C      40 CONTINUE
69      C      RETURN
70      C
71      C
72      C

```

```

1      C      SUBROUTINE RIROW
2      C
3      C      ROUTINE TO READ ROW J FROM LCM TO SCM BUFFER ONE
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, 1974
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 RE8TRT,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,THIRD,NCLST,TOUT,TWFIN
26      COMMON/YSC2/TUG1,TUS1,NCG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
27      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
28      2 VV,XCQNV,XL,XR,YB,YCQNV,YT,PTPOLD,DT8V,DTLAST,FIYBQ,IYBQ,YCNVLD,
29      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLM,
30      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,TE,DTR,TMASS,DTVSAV,DTC8AV,IDTV
31      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,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 -----
38      DATA NE/0/
39      IEC=(J=1)*NQI
40      CALL ECRD(AASC,IEC,NQI,IDUM)
41      RETURN
42      END

```

```

1      SUBROUTINE SETIJ
2      C
3      C ROUTINE TO SET THE SCM POINTER TO THE ITH ELEMENT IN ROW J,
4      C ASSUMING ROW J IS IN SCM BUFFER ONE
5      C
6      C ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
7      C MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
8      C
9      * ----- BEGIN COMDECK PARAM -----
10     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
11     1 NLCP3,NLCP4,IFLMSZ
12     * ----- END COMDECK PARAM -----
13     * ----- BEGIN COMDECK YAQSC -----
14     LOGICAL RESTRT,FILM,PAPER,TURB
15     REAL LAM,MU
16     C COMMON/YSC1/AASC(NSCP1)
17     COMMON/YSC1/AASC(9600)
18     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
19     1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
20     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
21     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
22     4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
23     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),

```

```

24      1 NCYC,NLC,NPS,NPT,NG,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
25      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
26      3 FREZYT,FREZBY,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
27      COMMON/YSC2/TUGI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
28      1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,GLEVEL,TQ,IST,
29      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
30      3 XCNVLD,FIXRO,IXRO,IXLO,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLIM,
31      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTR,TMASS,DTVSAV,DTCSAV,IDTV
32      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGDX,ITM,JTM,ITG,JTG
33      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
34      COMMON/YSC2/ZZ
35      C COMMON/YSC4/ITAB(ITABP)
36      C COMMON/YSC4/ITAB(1000)
37      C COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
38      * ----- END COMDECK YAQSC -----*
39      IJ=(I=1)*NQ+1
40      RETURN
41      END

```

```

1      SUBROUTINE START
2      C ROUTINE TO TRANSFER DATA BETWEEN LARGE CORE AND SMALL CORE IN
3      C ORDER TO MINIMIZE SMALL CORE REQUIREMENTS
4      C
5      C ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
6      C MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
7      C
8      C ----- BEGIN COMDECK PARAM -----
9      C 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     C COMMON/YSC1/AASC(9600)
17     C COMMON/YSC2/AA(1),ANC,AN,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,IXL,IXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
20     3 IPTO,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,NG,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
24     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
25     3 FREZYT,FREZBY,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
26     COMMON/YSC2/TUGI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
27     1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,GLEVEL,TQ,IST,
28     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
29     3 XCNVLD,IXRO,IXLO,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLIM,
30     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTR,TMASS,DTVSAV,DTCSAV,IDTV
31     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGDX,ITM,JTM,ITG,JTG
32

```

```

33      6 ,TMASSV,WMAXEF,RMINEF,TSTRTO
34      COMMON/YSC2/ZZ
35      C COMMON/YSC4/ITAB(ITABP)
36      COMMON/YSC4/ITAB(1000)
37      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
38      ★ ***** END COMDECK YAQSC      *****
39      DATA NE/0/
40      C
41      C READ IN THE FIRST THREE ROWS
42      C
43      C IJPS IS THE SCM POINTER TO WHERE THE NEXT ROW IS TO BE READ
44      C INTO SCM. ONE NEEDS TO THINK OF THE SMALL CORE AREA AS BEING
45      C DIVIDED INTO THREE BUFFERS, SAY A,B, AND C. INITIALLY, ROW J=1
46      C IS READ INTO BUFFER A, ROW J=2 INTO BUFFER B, AND ROW J=3 INTO
47      C BUFFER C. WHEN A REQUEST IS MADE FOR ROW J=4, BUFFER A WHICH
48      C CURRENTLY HOLDS ROW J=1 IS WRITTEN OUT TO LCM AND ROW J=4 IS
49      C READ INTO BUFFER A. THIS PROCESS CONTINUES WITH ONE ROW BEING
50      C WRITTEN TO LCM AND ONE ROW REPLACING IT IN SCM SO THAT THREE
51      C ROWS ARE ALWAYS AVAILABLE, ONE IN EACH BUFFER.
52      C IECR IS THE LCM ADDRESS FROM WHICH DATA WILL BE READ TO SCM NEXT
53      C IFCW IS THE LCM ADDRESS TO WHICH DATA WILL BE WRITTEN FROM SCM NXT
54      C
55      IJPS=1
56      IECR=0
57      IECW=0
58      C
59      C READ THE FIRST ROW INTO SCM INTO BUFFER ONE
60      C
61      C AASC IS BEGINNING OF THE THREE SCM BUFFERS
62      C NQI IS THE NUMBER OF WORDS COMPRISING ONE ROW OF DATA
63      C NE IS AN ERROR FLAG WHICH IS UNUSED ON THE 7600
64      C
65      CALL FCRD(AASC(IJPS),IECR,NQI,NE)
66      C
67      C INCREMENT THE LCM READ ADDRESS
68      C
69      IECR=IECR+NQI
70      C
71      C SET THE BUFFER POINTER TO THE SECOND BUFFER, OBVIOUSLY,
72      C ISC2=NQI+1
73      C
74      IJPS=ISC2
75      C
76      C READ IN THE SECOND ROW
77      C
78      CALL FCRD(AASC(IJPS),IECR,NQI,NE)
79      C
80      C INCREMENT THE LCM READ ADDRESS AGAIN
81      C
82      IECR=IECR+NQI
83      C
84      C UPDATE THE POINTERS TO READ THE NEXT ROW INTO BUFFER THREE.
85      C
86      C IJP IS THE POINTER TO THE ELEMENT (1,J+1)
87      C IJ IS THE POINTER TO THE ELEMENT(1,J)
88      C IJM IS THE POINTER TO (1,J=1)
89      C IJMS IS THE SCM ADDRESS TO BE WRITTEN OUT NEXT

```

```

90      C      IBUF IS THE NEXT BUFFER TO BE USED WHEN THE SUBROUTINE IS ENTERED
91      C
92      10 CONTINUE
93          IJP=ISC3
94          IJPS=ISC3
95          IJ=ISC2
96          IJM#1
97          IJM8#1
98          IBUF#1
99      C
100     C      FILL THE THIRD BUFFER
101     C
102     20 CALL ECRD(AASC(IJPS),IECR,NQI,NE)
103     C
104     UPDATE THE LCM POINTER FOR THE NEXT READ
105     C
106     IECR=IECR+NQI
107     RETURN
108     C
109     ****
110     ENTRY LOOP
111     ****
112     C
113     C      READ IN THE NEXT ROW. FIRST, FLUSH THE OLDEST BUFFER.
114     C
115     CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
116     C
117     UPDATE THE LCM POINTER FOR THE NEXT WRITE
118     C
119     IECW=IECW+NQI
120     C
121     PREPARE THE POINTERS FOR THE NEXT READ DEPENDING ON WHICH BUFFER
122     C      IS TO BE FILLED
123     C
124     GO TO (30,40,10),IBUF
125     C
126     PREPARE TO FILL BUFFER ONE
127     C
128     30 CONTINUE
129         IJP#1
130         IJPS#1
131         IJ=ISC3
132         IJM=ISC2
133         IJM8#ISC2
134     C
135     C      INDICATE THAT BUFFER TWO IS TO BE FILLED NEXT TIME
136     C
137     IBUF#2
138     C
139     GO FILL BUFFER ONE
140     C
141     GO TO 20
142     C
143     PREPARE TO FILL BUFFER TWO
144     C
145     40 CONTINUE
146     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      C      IBUF#3
155      C
156      C      GO FILL BUFFER TWO
157      C
158      C      GO TO 20
159      C
160      C      *****
161      C      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      C      CALL ECWR(AASC(IJMS),IECW,NQI,IDUM)
169      C      IECW#IECW+NQI
170      C
171      C      DETERMINE WHICH IS THE OTHER BUFFER TO BE EMPTIED
172      C
173      C      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,

```

```

7      C      IJ POINTS TO CELL (IP1,JP2), SEE DOCUMENTATION IN SUBROUTINE
8      C      START FOR SOME OF THE VARIABLE DEFINITIONS.
9      C
10     C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
11     C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
12     C
13     *      ----- BEGIN COMDECK PARAM -----
14     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
15     1 NLCP3,NLCP4,IFLMSZ
16     *      ----- END COMDECK PARAM -----
17     *      ----- BEGIN COMDECK YAGSC -----
18     LOGICAL RESTRT,FILM,PAPER,TURB
19     REAL LAM,MU
20     C      COMMON/YSC1/AASC(NSCP1)
21     COMMON/YSC1/AASC(9600)
22     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
23     1 DTO(10),DTONC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
24     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
25     3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
26     4 JUNF,IXL,IXR,IYB,IYT,J,JBAR
27     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
28     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
29     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
30     3 PREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
31     COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
32     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
33     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
34     3 XCNVLD,FXRO,FXL0,IXRO,IXL0,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
35     4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DTR,THMASS,DTVS,DTCSAV,IDTV
36     5 ,JDTV,DTDC,DTDC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMAX,ITM,JTM,ITG,JTG
37     6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
38     COMMON/YSC2/ZZ
39     C      COMMON/YSC4/ITAB(ITABP)
40     COMMON/YSC4/ITAB(1000)
41     COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
42     *      ----- END COMDECK YAGSC -----
43     DATA NE/0/
44     C
45     C      SET INITIAL SCM READ ADDRESS (BUFFER TWO)
46     C
47     C      IJMS=ISC2
48     C
49     C      SET NEXT LCM READ AND WRITE ADDRESSES,
50     C
51     C      ITV IS JP1*NQI OR THE LCM ADDRESS OF THE BEGINNING OF ROW JP2
52     C
53     C      IECR=ITV
54     C      IECW=ITV
55     C
56     C      READ ROW JP2 INTO BUFFER TWO
57     C
58     C      CALL ECRO(AASC(IJMS),IECR,NQI,IDUM)
59     C
60     C      DECREMENT THE READ ADDRESS
61     C
62     C      IECR=IECR-NQI
63     C

```

```

64      C      PREPARE TO READ IN ROW J=1 INTO BUFFER ONE
65      C
66      C      IJM POINTS TO (IP1,J=1) (THE END OF BUFFER ONE)
67      C
68      10     CONTINUE
69      C      IJMS=ISC2=NQI
70      C      IJMS=1
71      C
72      C      IBUF IN THIS ROUTINE IS THE LAST BUFFER THAT WAS FILLED
73      C
74      C      IBUF=1
75      C
76      C      IJ=IJM+NQI (THE LAST CELL OF BUFFER TWO)
77      C
78      C      IJ=IJM+NQI
79      C      IJPS=ISC2
80      C
81      C      CHECK TO SEE IF THE BOTTOM OF THE MESH HAS BEEN REACHED
82      C
83      20     IF(IECR,LT,0) GO TO 40
84      C
85      C      NO, GO AHEAD AND READ IN THE NEXT BUFFER
86      C
87      C      CALL ECRD(AASC(IJMS),IECR,NQI,JDUM)
88      C
89      C      DECREMENT THE READ ADDRESS
90      C
91      C      IECR=IECR-NQI
92      30     RETURN
93      C
94      C      BOTTOM OF THE MESH HAS BEEN REACHED, SET A FLAG (IBUF=3),GO
95      C      WRITE OUT ROW J=2,AND QUIT
96      C
97      40     IBUF=3
98      C      GO TO 50
99      C
100     C      ****
101     C      ENTRY LOOPD
102     C      ****
103     C
104     C      ENTRY TO CYCLE THROUGH ONE MORE ROW OF THE MESH IN A DESCENDING
105     C      FASHION AS DESCRIBED ABOVE
106     C
107     C      WRITE OUT ROW J AND READ ROW J=2 INTO ITS BUFFER
108     C
109     50     CONTINUE
110     C      CALL ECWR(AASC(IJPS),IECW,NQI,JDUM)
111     C
112     C      DECREMENT THE WRITE ADDRESS
113     C
114     C      IECW=IECW-NQI
115     C
116     C      SET UP THE POINTERS FOR THE NEXT READ DEPENDING ON WHICH BUFFER
117     C      IS TO BE FILLED, IF IBUF=3,ONLY A RETURN IS EXECUTED SINCE
118     C      THE MESH IS FINISHED.
119     C
120     C      GO TO (60,10,30),IBUF

```

```
121      C      PREPARE POINTERS TO FILL BUFFER TWO
122      C
123      C
124      60 CONTINUE
125          IBUF=2
126          IJ=ISC2-NQ
127          IJPS=1
128          IJMS=IJ+NQI
129          IJMS=ISC2
130      C
131      C      GO DO THE ACTUAL READ
132      C
133          GO TO 20
134          END
```

```

1 SUBROUTINE TICBOX
2
3 C ROUTINE TO DRAW A BOX AROUND THE GRID AREA DISPLAYED
4 C AND LABEL THE AXES
5
6 C ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
7 C MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1975
8
9 *
10 *----- BEGIN COMDECK PARAM -----*
11 COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPF8,NP1,NP2,NLCPI,NLCPI2,
12 1 NLCPI3,NLCPI4,IFLMSZ
13 *----- END COMDECK PARAM -----*
14 *----- BEGIN COMDECK YAGSC -----*
15 LOGICAL RESTRT,FILM,PAPER,TURB
16 REAL LAM,MU
17 C COMMON/YSC1/AASC(NSCP1)
18 COMMON/YSC1/AASC(9600)
19 COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTG,DTFAC,
20 1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
21 2 FIPYB,FIPYI,FIXL,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
22 3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISCI3,ITV,
23 4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
24 COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
25 1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
26 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
27 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
28 COMMON/YSC2/TUGI,TUSI,NCO,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
29 1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
30 2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
31 3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
32 4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DTI,MASS,DTVSAV,DTCSAV,IDTV
33 5 ,JDTV,ITC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
34 6 ,TMASSV,WMAXEF,RHINEF,TSTRTD
35 C COMMON/YSC2/ZZ
36 COMMON/YSC4/ITAB(ITABP)
COMMON/YSC4/ITAB(1000)

```

```

37      COMMON/YSCS/RESTRT,FILM,PAPER,IPD,IFD
38      *      **** END COMDECK YAQSC ****
39      *      **** BEGIN COMDECK PCALL ****
40      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
41      *      **** END COMDECK PCALL ****
42      COMMON/NCODEC/IMSIGN,XMAN,IESIGN,IEXP
43      DATA BCD/1H /
44
45      C      DRAW THE PLOT FRAME
46      C
47      CALL DRV(IXL,IYT,IXR,IYT)
48      CALL DRV(IXR,IYT,IXR,IYB)
49      CALL DRV(IXR,IYB,IXL,IYB)
50      CALL DRV(IXL,IYB,IXL,IYT)
51
52      C      IX2,IX3,IY2, AND IY3 ARE THE RASTER COORDINATES OF THE ENDS OF
53      C      THE TIC MARKS TO BE DRAWN ON THE PLOTTING RECTANGLE
54
55      C      IX2=IXL+8
56      C      IX3=IXR-8
57      C      IY2=IYB-8
58      C      IY3=IYT+8
59
60      C      ESTABLISH TOP AND BOTTOM FOR PLOT INCREMENTS. ROUND TOP AND
61      C      BOTTOM GRID VALUES TO THE NEAREST POWER OF 10. (YUPI,YLBI).
62
63      C      IXX=ALOG10(YUP)
64      DTIC=10,**IXX
65
66      C      DETERMINE THE Y EXTREMA TO USE TO SET DTIC, THE TIC INCREMENT.
67      C      THE Y MINIMUM, YLBI, IS SET SUCH THAT IT IS LESS THAN YLB
68      C      AND EQUAL TO AN INTEGRAL MULTIPLE OF THE POWER OF TEN
69      C      NEAREST TO BUT LESS THAN YUP. THE Y MAXIMUM, YUPI, IS
70      C      DETERMINED SIMILARLY BUT SO THAT IT IS GREATER THAN YUP.
71
72      C      ASSUME THAT YLB IS .GE.0
73
74      C      YLBI=0,
75      IF(DTIC,GE,YLB) GO TO 20
76      DO 10 I=1,9
77      YLBI=YLBI+DTIC
78      IF(YLBI,LT,YLB) GO TO 10
79      YLBI=YLBI-DTIC
80      GO TO 20
81      10 CONTINUE
82      CALL UNCLE(1,6HTICBOX,24,24HCOULD NOT DETERMINE YLBI)
83      20 CONTINUE
84      YUPI=DTIC
85      DO 30 I=2,10
86      YUPI=YUPI+DTIC
87      IF(YUPI,GE,YUP) GO TO 40
88      30 CONTINUE
89      CALL UNCLE(1,6HTICBOX,24,24HCOULD NOT DETERMINE YUPI)
90      40 CONTINUE
91
92      C      BOUNDS ESTABLISHED. DETERMINE THE LABELLING INCREMENT BASED
93      C      ON 10 INCREMENTS.

```

```

94      C
95      DTIC=(YUPI=YLB)*,1
96      50 CONTINUE
97      C
98      COMPUTE HOW MANY TICS WILL BE DRAWN
99      C
100     ITIC=0
101     YTICT=YLB
102     60 CONTINUE
103     IF(YTICT,GE,YLB) GO TO 70
104     YTICT=YTICT+DTIC
105     GO TO 60
106     70 CONTINUE
107     YLB=YTICT-OTIC
108     80 CONTINUE
109     IF(YTICT,GE,YUP) GO TO 90
110     ITIC=ITIC+1
111     YTICT=YTICT+DTIC
112     GO TO 80
113     90 CONTINUE
114     C
115     MUST BE AT LEAST 6 TICS OR DTIC WILL BE REDUCED
116     C
117     IF(ITIC,GE,6) GO TO 100
118     DTIC=DTIC/2.
119     GO TO 50
120     100 CONTINUE
121     C
122     YTIC IS THE Y-COORDINATE OF THE LABEL
123     C
124     YTIC=YLB
125     110 CONTINUE
126     IF(YTIC,GE,YLB) GO TO 120
127     YTIC=YTIC+DTIC
128     GO TO 110
129     C
130     COMPUTE RASTER COORDINATE FOR YTIC
131     C
132     120 CONTINUE
133     IY1=FIYB+(YTIC-YLB)*YCONVP
134     C
135     IF THIS LABEL WOULD PUT US OUTSIDE OF THE GRID, CONSIDER THE
136     Y-LABELLING FINISHED
137     C
138     IF(IY1,LT,IYT) GO TO 210
139     C
140     DRAW TICS ON THE LEFT AND RIGHT SIDES OF THE FRAME
141     C
142     CALL DRV(IXL,IY1,IX2,IY1)
143     CALL DRV(IX3,IY1,IXR,IY1)
144     C
145     PREPARE AND OUTPUT THE LABEL ITSELF
146     C
147     IMSIGN=1H
148     IF(YTIC,GT,0.) GO TO 140
149     IF(YTIC,LT,0.) GO TO 130
150     IMSIGN=2H0.

```

```

151      CALL NCODE(8,6H(6XA2),IMSIGN,1,BCD)
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) GU 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,BCD)
179      GO TO 200
180 190  CONTINUE
181      CALL NCODE(8,15H(A1,F4,2,A1,I2),IMSIGN,4,BCD)
182 200  CONTINUE
183      CALL TSP(1,IY1,8,BCD)
184
185 C      INCREMENT THE TIC VALUE AND PROCEED
186 C
187      YTIC=YTIC+DTIC
188      GO TO 120
189
190 C      SAME PROCEDURE AS FOR Y-AXIS
191
192 210  CONTINUE
193
194 C      USE THE SAME LABELLING INCREMENT FOR THE X-AXIS
195
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      IRITER=IX1-29
204      IMSIGN=1H
205      IF(XTIC,GT,0.) GO TO 240
206      IF(XTIC,LT,0.) GO TO 230
207      IMSIGN=2H0.

```

```

208      CALL NCODE(8,9H(3XA2,3X),IMSIGN,1,BCD)
209      GO TO 300
210 230 CONTINUE
211      XLOG=ALOG10(ABS(XTIC))
212      IMSIGN=1H-
213      GO TO 250
214 240 CONTINUE
215      XLOG=ALOG10(XTIC)
216 250 CONTINUE
217      IEXP=XLOG
218      XMAN=XLOG-FLOAT(IEXP)
219      IF(XMAN,GE,0,) GO TO 260
220      XMAN=XMAN+1,
221      IEXP=IEXP+1
222 260 CONTINUE
223      XMAN=1.000001*XMAN
224      IF(XMAN,LT,1.) GO TO 270
225      XMAN=XMAN+1,
226      IEXP=IEXP+1
227 270 CONTINUE
228      XMAN=10,**XMAN
229      IESIGN=1H+
230      IF(IEXP,GE,0) GO TO 280
231      IEXP=-IEXP
232      IESIGN=1H-
233 280 CONTINUE
234      IF(IEXP,GE,10) GO TO 290
235      CALL NCODE(8,17H(1XA1,F4,2,A1,I1),IMSIGN,4,BCD)
236      GO TO 300
237 290 CONTINUE
238      CALL NCODE(8,15H(A1,F4,2,A1,I2),IMSIGN,4,BCD)
239 300 CONTINUE
240      CALL TSP(IRITE,JRITE,8,BCD)
241      XTIC=XTIC+DTIC
242      GO TO 220
243      END

```

```

1      SUBROUTINE TRBCUR
2
3      C      ROUTINE TO CALCULATE TURBULENCE CORRECTIONS
4
5      C      ORIGINALLY WRITTEN BY HANS RUPPEL,LASL T-3
6      C      MODIFIED BY J.L.NORTON,LASL T-3,1975
7
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 YSTORE ****
13     *      **** BEGIN COMDECK YAQDIM ****
14     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC

```

```

15      1 1),VG(1),RG(1),SIE(1),MP(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),DT02,DT08,DTPOS,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
32      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNPO2,KXI,LAM,LPB,MU,NAME(8),
33      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,DMCYL,PXCONV
34      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
35      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,GLEVEL,TQ,IST,
38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVV,QMN,QMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DT,TMASS,DTVS,DTCSAV,DTVSAV,IDTV
41      5 ,JDTV,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43      COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
47      * ----- END COMDECK YAQSC -----*
48      * ----- BEGIN COMDECK YAQE -----*
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GR6V),(AASC(27),GRROP,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
1 AVYSV,X24K)
60      REAL M,MP,MPAR,MTIL
61      * ----- END COMDECK YAQE -----*
62      * ----- END COMDECK YSTORE -----*
63      CALL START
64      DO 20 J=2,JP1
65      DO 10 I=1,IBAR
66      IPJP=IJP+NQ
67      IPJ=IJ+NQ
68      IMJ=IJ+NQ
69      X1=X(IPJ)
70      Y1=Y(IPJ)
71

```

```

72      X2=X(IPJP)
73      Y2=Y(IPJP)
74      X3=X(IJP)
75      Y3=Y(IJP)
76      X4=X(IJ)
77      Y4=Y(IJ)
78      Y21=Y2-Y1
79      Y32=Y3-Y2
80      Y43=Y4-Y3
81      Y14=Y1-Y4
82      X12=X1-X2
83      X23=X2-X3
84      X34=X3-X4
85      X41=X4-X1
86      AVX8V(IJ)=.25*(X1+X2+X3+X4)
87      AVY8V(IJ)=.25*(Y1+Y2+Y3+Y4)
88      R12=R(IPJ)+R(IPJP)
89      R23=R(IJP)+R(IPJP)
90      R34=R(IJP)+R(IJ)
91      R41=R(IJ)+R(IPJ)
92      C=CONC(IJ)
93      SN=SIG(IJ)
94      RC=RO(IJ)
95      RCQ=RC*TUQ(IJ)
96      RCS=RC*SIE(IJ)
97      AC=AREA(IJ)
98      IF(I.EQ.1) IMJ=IJ
99      IF(I.EQ.IBAR) IPJ=IJ
100     IF(J.EQ.2) IJM=IJ
101     IF(J.EQ.JP1) IJP=IJ
102     Z1=R12*(Y21**2+X12**2)*(SIG(IPJ)+SN)/(AREA(IPJ)+AC)
103     Z2=R23*(Y32**2+X23**2)*(SIG(IJP)+SN)/(AREA(IJP)+AC)
104     Z3=R34*(Y43**2+X34**2)*(SIG(IMJ)+SN)/(AREA(IMJ)+AC)
105     Z4=R41*(Y14**2+X41**2)*(SIG(IJM)+SN)/(AREA(IJM)+AC)
106     CTEMP(IJ)=0,5*((CONC(IPJ)=C)*Z1+(CONC(IJP)=C)*Z2+(CONC(IMJ)=C)*Z3+
107     1*(CONC(IJM)=C)*Z4)
108     DLSROI(IJ)=0,
109     DLSROQ(IJ)=0,
110     R1=RO(IPJ)
111     R2=RO(IJP)
112     R3=RO(IMJ)
113     R4=RO(IJM)
114     ROL(IJ)=0,5*((R1=RC)*Z1+(R2=RC)*Z2+(R3=RC)*Z3+(R4=RC)*Z4)
115     DLSROQ(IJ)=.5*((R1*TUQ(IPJ)=RCQ)*Z1+(R2*TUQ(IJP)=RCQ)*Z2+(R3*TUQ(
116     1 IMJ)=RCQ)*Z3+(R4*TUQ(IJM)=RCQ)*Z4)
117     DLSROI(IJ)=0,5*((R1*SIE(IPJ)=RCS)*Z1+(R2*SIE(IJP)=RCS)*Z2+(R3*SIE(
118     1 IMJ)=RCS)*Z3+(R4*SIE(IJM)=RCS)*Z4)
119     IJ=IJ+NQ
120     IJM=IJM+NQ
121     IJP=IJP+NQ
122     10 CONTINUE
123     CALL LOOP
124     20 CONTINUE
125     CALL DONE
126     CALL START
127     DO 80 J=2,JP1
128     DO 70 I=1,IBAR

```

```

129      IPJ=IJ+NQ
130      IMJ=IJ-NQ
131      X1=AVXSV(IPJ)
132      X2=AVXSV(IJP)
133      X3=AVXSV(IMJ)
134      X4=AVXSV(IJM)
135      Y1=AVYSV(IPJ)
136      Y2=AVYSV(IJP)
137      Y3=AVYSV(IMJ)
138      Y4=AVYSV(IJM)
139      P1=P(IPJ)
140      P2=P(IJP)
141      P3=P(IMJ)
142      P4=P(IJM)
143      R01=R0(IPJ)
144      R02=R0(IJP)
145      R03=R0(IMJ)
146      R04=R0(IJM)
147      IF(I,NE,1) GO TO 30
148      X3=X1
149      Y3=Y1
150      P3=P1
151      R03=R01
152      IF(J,NE,2) GO TO 40
153      X4=X2
154      Y4=Y2
155      P4=P2
156      R04=R02
157      IF(I,NE,IBAR) GO TO 50
158      X1=X3
159      Y1=Y3
160      P1=P3
161      R01=R03
162      IF(J,NE,JP1) GO TO 60
163      X2=X4
164      Y2=Y4
165      P2=P4
166      R02=R04
167      60 CONTINUE
168      RA=1./(X1-X3)*(Y2-Y4)-(X2-X4)*(Y1-Y3)
169      Y31=(Y3-Y1)*RA
170      Y42=(Y4-Y2)*RA
171      X13=(X1-X3)*RA
172      X24=(X2-X4)*RA
173      DPDZ=(P2-P4)*Y31+(P3-P1)*Y42
174      DPDZ=(P2-P4)*X13+(P3-P1)*X24
175      DR0DR=(R02-R04)*Y31+(R03-R01)*Y42
176      DR0DZ=(R02-R04)*X13+(R03-R01)*X24
177      GRR0P(IJ)=DR0DR*DPDR+DR0DZ*DPDZ
178      GRR0R(IJ)=DR0DR
179      GRR0Z(IJ)=DR0DZ
180      IJ=IPJ
181      IJM=IJM+NQ
182      IJP=IJP+NQ
183      70 CONTINUE
184      CALL LOOP
185      80 CONTINUE

```

```

186      CALL DONE
187      CALL START
188      CALL LOOP
189      DO 120 J=3,JP1
190      DO 110 I=1,IBAR
191      IMJ=IJ=NQ
192      IMJM=IJM=NQ
193      IF(I.EQ.1) IMJ=IJ
194      IF(I.EQ.1) IMJM=IJM
195      SIG=0.25*(SIG(IJ)+SIG(IMJ)+SIG(IJM))
196      VVA=0.25*(1./RVOL(IJ)+1./RVOL(IMJ)+1./RVOL(IJM))*RMC
197      1 IJ)
198      X1=AVXSV(IJM)
199      X2=AVXSV(IJ)
200      X3=AVXSV(IMJ)
201      X4=AVXSV(IMJM)
202      Y1=AVYSV(IJM)
203      Y2=AVYSV(IJ)
204      Y3=AVYSV(IMJ)
205      Y4=AVYSV(IMJM)
206      IF(I.NE.1) GO TO 90
207      X3=X2
208      X4=X1
209      Y3=Y2
210      Y4=Y1
211      90 CONTINUE
212      RXXA=1./((X1-X3)*(Y2-Y4)+(X2-X4)*(Y3-Y1))
213      XXA=((RO(IJM)-RO(IMJ))*(Y2-Y4)+(RO(IJ)-RO(IMJM))*(Y3-Y1))*RXXA*SIJ
214      YYA=((RO(IJ)-RO(IMJM))*(X1-X3)+(RO(IMJ)-RO(IJM))*(X2-X4))*RXXA*SIJ
215      XXA=XXA*VVA
216      YYA=YYA*VVA
217      IF(NCYC.EQ.NCQ+1) GO TO 100
218      UTIL(IJ)=UTIL(IJ)+XXA=GRSV(IJ)
219      VTIL(IJ)=VTIL(IJ)+YYA=GZSV(IJ)
220      100 CONTINUF
221      GRSV(IJ)=XXA
222      GZSV(IJ)=YYA
223      IJ=IJ+NQ
224      IJM=IJM+NQ
225      110 CONTINUE
226      CALL LOOP
227      120 CONTINUE
228      CALL DONE
229      CALL START
230      DO 140 J=2,JP1
231      DO 130 I=1,IBAR
232      IPJ=IJ+NQ
233      IPJP=IPJ+NQ
234      SIJ=SIG(IJ)
235      DALF=.25*DT*SIJ
236      X13=X(IPJ)-X(IPJP)
237      X24=X(IPJP)-X(IJ)
238      Y13=Y(IPJ)-Y(IPJP)
239      Y24=Y(IPJP)-Y(IJ)
240      U13=U(IPJ)+U(IPJP)
241      U24=U(IPJP)+U(IJ)
242      V13=V(IPJ)+V(IPJP)

```

```

243      V24=V(IPJP)+V(IJ)
244      R13=(R(IPJ)+R(IPJP))*DALF
245      R24=(R(IPJP)+R(IJ))*DALF
246      ZZ=GRR0Z(IJ)
247      ZR=GRR0R(IJ)
248      H13=0.5*(U13*ZZ+V13*ZR)
249      H24=0.5*(U24*ZZ+V24*ZR)
250      H1=(H24*Y24-V24*ZZ*X24)*R24
251      H2=(-H13*Y13+V13*ZZ*X13)*R13
252      H3=(-H24*X24+U24*ZR*Y24)*R24
253      H4=(H13*X13-U13*ZR*Y13)*R13
254      UTIL(IPJP)=UTIL(IPJ)-H3*RM(IPJ)
255      UTIL(IPJP)=UTIL(IPJP)-H4*RM(IPJP)
256      UTIL(IJP)=UTIL(IJP)+H3*RM(IJP)
257      UTIL(IJ)=UTIL(IJ)+H4*RM(IJ)
258      VTIL(IJ)=VTIL(IJ)+H2*RM(IJ)
259      VTIL(IJP)=VTIL(IJP)+H1*RM(IJP)
260      VTIL(IPJP)=VTIL(IPJP)-H1*RM(IPJP)
261      VTIL(IPJP)=VTIL(IPJP)-H2*RM(IPJP)
262      *****TURBULENCE ENERGY EQUATION*****
263      RIJ=RO(IJ)
264      RECRHO=1./RIJ
265      D=DELSM(IJ)
266      XPR1=2.*SIJ/(TUS(IJ)**2*QLEVEL)
267      TUQ(IJ)=RIJ*TUQ(IJ)+DT*(SIJ*(CAPGAM(IJ)*RIJ-GRR0P(IJ)*RECRHO)+*
268      1 DLSR0Q(IJ)*RVOL(IJ))
269      TUQ(IJ)=(TUQ(IJ)-TWTHRD*D*D*SIJ*RIJ*DT)/((1.+(XPR1+TWTHRD*D)*DT)*
270      1 RIJ)
271      IF(TUQ(IJ),LT,0.) TNEG=TNEG+TUQ(IJ)*RIJ/RVOL(IJ)
272      IF(TUQ(IJ),LT,0.) TUQ(IJ)=0.
273      *****INTERNAL ENERGY EQUATION*****
274      SIE(IJ)=SIE(IJ)+DT*(DLSROI(IJ)*RVOL(IJ)*RECRHO+XPR1*TUQ(IJ))
275      IJ=IJ+NQ
276      IJM=IJM+NQ
277      IJP=IJP+NQ
278      130 CONTINUE
279      CALL LOOP
280      140 CONTINUE
281      CALL DONE
282      RETURN
283      END

```

```

1      SUBROUTINE TRBERF
2
3      C      ROUTINE TO CALCULATE THE ERROR FUNCTION VALUES TO BE USED
4      C      FOR INTERPOLATION TO COMPUTE THE PARTICLE TURBULENT
5      C      DIFFUSION MOTION
6
7      C      WRITTEN BY J.L.NORTON, LASL T=3, 1975
8
9      *      ***** BEGIN COMDECK TRBDIF      *****

```

```

10      COMMON/CTDIF/ERFV(21),NERFV,DXEF
11      *      ===== END COMDECK TRBDIF =====
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,GROVEL,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,REZYB,RIBAR,RIBJB,
25      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWPIN
26      COMMON/YSC2/TUGI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PROTM,
27      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
28      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
29      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLM,
30      4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTR,MASS,DTVSAV,DTCSAV,IDTV
31      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,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 =====
38      C      FIRST,INITIALIZE THE VARIOUS PARAMETERS
39      C
40      C      NERFV IS THE NO. OF VALUES OF ERF IN THE INTERPOLATION TABLES
41      C
42      C      NERFV=21
43      C
44      C      DXEF IS THE SPACING IN X BETWEEN THE ERF(X) TABLE ENTRIES
45      C
46      C      DXEP=WMAXEF/FLOAT(NERFV-1)
47      C
48      C      COMPUTE THE TABLE VALUES
49      C
50      XEF=DXEF
51      DO 10 I=1,NERFV
52      XEF=XEF+DXEF
53      ERFV(I)=ERF(XEF)
54 10 CONTINUE
55      IF(ERFV(NERFV).LE.0.) CALL UNCLE(4,6HTRBERF,26,
56      1 26ERRORS IN CALCULATING ERFV)
57      RETURN
58      END

```

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

```

2      C
3      C      ERROR PROCESSING ROUTINE
4      C
5      C      ISFLAG = TYPE OF CALL
6      C          • 1 = FATAL ERROR,DUMP ARRAYS (NOT IMPLEMENTED)
7      C          • 2 = NORMAL EXIT
8      C          • 3 = DUMP ARRAYS AND RETURN (NOT IMPLEMENTED)
9      C          • 4 = FATAL ERROR,NO DUMP
10     C      RNAME = NAME OF CALLING ROUTINE
11     C      NCHAR = NO. OF CHARACTERS IN ERROR MESSAGE
12     C      CHAR = ERROR MESSAGE
13     C
14     C      WRITTEN BY J.L.NORTON,LASL T=3,1974
15     C
16     *      ----- BEGIN COMDECK PARAM -----
17     *      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
18     *      NLCP3,NLCP4,IFLMSZ
19     *      ----- END COMDECK PARAM -----
20     *      ----- BEGIN COMDECK YAGSC -----
21     *      LOGICAL RESTRT,FILM,PAPER,TURB
22     *      REAL LAM,MU
23     C      COMMON/YSC1/AASC(NSCP1)
24     C      COMMON/YSC1/AASC(9600)
25     C      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTA,DTFAC,
26     1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,PIPXL,PIPXR,
27     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
28     3 IDTO,IJ,IJM,IJM,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
29     4 IUNF,IXL,IXR,IXB,IXT,J,JBAR
30     COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
31     1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
32     2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
33     3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLBT,TOUT,TWFIN
34     COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
35     1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
36     2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DT8V,DTLAST,FIYBO,IYBO,YCNVLD,
37     3 XCNVLD,FIXRO,IXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
38     4 ROMFXR,ROMFYB,JDUMP,TWTHRD,TE,DTR,THMASS,DTVSAV,DTCSAV,IDTV
39     5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOD,TMAX,TGMX,ITM,JTM,ITG,JTG
40     6 ,THMASSV,WMAXEF,RMINEF,TSTRTD
41     COMMON/YSC2/ZZ
42     C      COMMON/YSC4/ITAB(ITABP)
43     C      COMMON/YSC4/ITAB(1000)
44     C      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
45     *      ----- END COMDECK YAGSC -----
46     DIMENSION CHAR(1)
47     DATA TP/0./
48     DO 10 IPX=6,IFD,6
49     10 WRITE(IPX,100)
50     GO TO (60,20,40,60),ISFLAG
51     20 CONTINUE
52     DO 30 IPX=6,IFD,6
53     30 WRITE(IPX,150)
54     GO TO 80
55     40 CONTINUE
56     DO 50 IPX=6,IFD,6
57     50 WRITE(IPX,130)
58     GO TO 80

```

```

59      60 CONTINUE
60      DO 70 IPX=6,IFD,6
61      70 WRITE(IPX,110)
62      80 CONTINUE
63      NWDS=NCHAR/10
64      IF(MOD(NCHAR,10).NE.0) NWDS=NWDS+1
65      DO 90 IPX=6,IFD,6
66      WRITE(IPX,120) RNAME,(CHAR(II),II#1,NWDS)
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 ,25H CODE TERMINATION AT CP # ,F10,4)
75      110 FORMAT(1H ,20HFATAL ERROR IN YAQUI)
76      120 FORMAT(1H ,24H UNCLE CALLED BY ROUTINE ,A6,10X9A10)
77      130 FORMAT(1H ,16H START UNCLE DUMP)
78      140 FORMAT(1H ,60(1H=))
79      150 FORMAT(1H ,25HNORMAL EXIT THROUGH UNCLE)
80      END

```

```

1      SUBROUTINE VELPLT(VMAX,IFLAG)
2
3      C      ROUTINE TO DO VELOCITY PLOTS
4
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
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
14     *      ***** BEGIN COMDECK PARAM      *****
15     COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,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),CG(1),VTIL(1)
23     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSRQO(1),CAPGAM(1),TUQ
24     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUGVEC(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),DKLSM(1),AREA(1)
27     *      ***** END COMDECK YAQDIM      *****
28     *      ***** BEGIN COMDECK YAQSC      *****

```

```

29      LOGICAL RESTRT,FILM,PAPER,TURB
30      REAL LAM,MU
31      C COMMON/YSC1/AASC(NSCP1)
32      C COMMON/YSC1/AASC(9600)
33      C COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
34      C 1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
35      C 2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
36      C 3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
37      C 4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
38      C COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNP,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
39      C 1 NCYC,NLC,NPC,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
40      C 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
41      C 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
42      C COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,NEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
43      C 1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
44      C 2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
45      C 3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
46      C 4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVS,DTCSAV,DTVS,DTCSAV,DTVS,
47      C 5 ,JDTV,JDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
48      C 6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
49      C COMMON/YSC2/ZZ
50      C COMMON/YSC4/ITAB(ITABP)
51      C COMMON/YSC4/ITAB(1000)
52      C COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
53      * ----- END COMDECK YAQSC -----
54      * ----- BEGIN COMDECK YAQE -----
55      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC(8),E,ETIL,AREA,XR13K),(AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(21),GRORR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),AVYSV,X24K)
56      REAL M,MP,MPAR,MTIL
57      * ----- END COMDECK YAQE -----
58      * ----- END COMDECK YSTORE -----
59      * ----- BEGIN COMDECK PCALL -----
60      C COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
61      * ----- END COMDECK PCALL -----
62      DIMENSION ITITLE(5,3)
63      DIMENSION IJL2(5),IJL3(5)
64      EQUIVALENCE(ITITLE(6),IJL2),(ITITLE(11),IJL3)
65      DATA ITITLE/49HFLUID VELOCITY VECTORS SCALED TO MAXIMUM VELOCITY/
66      DATA IJL2/31HUNScaled FLUID VELOCITY VECTORS,1H /
67      DATA IJL3/30HVELOCITY OF GRID REL. TO FLUID,
68      1 20H SCALED TO MAX. VEL./
69      C
70      C SET UP THE VELOCITY VECTOR SCALING FACTOR.
71      C VV IS .9 TIMES AN AVERAGE DR. THIS IS USED TO SCALE THE LENGTH
72      C OF THE VELOCITY VECTOR SO IT IS NOT LONGER THAN THE EXTENT
73      C OF AN AVERAGE CELL.

```

```

86      DROU=VV/VMAX
87      C
88      C      SPACE FORWARD TO THE NEXT FILM FRAME
89      C
90      C      CALL ADV(1)
91      C
92      C      LOOP OVER ALL REAL ZONES
93      C
94      C      CALL START
95      DO 20 J=2,JP2
96      DO 10 I=1,IP1
97      IF(IFLAG.EQ.1) DROU=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     C      IY1=FIYB+(Y(IJ)-YLB)*YCONVP
104     IF(IY1.GT.IYB.OR.IY1.LT.IYT) GO TO 10
105     IY2=FIYB+(Y(IJ)+Y(IJ)*DROU-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)*DROU)*XCONVP
110     IF(IX2.GT.IXR) GO TO 10
111     C
112     C      DRAW THE VECTOR
113     C
114     C      CALL DRV(IX1,IY1,IX2,IY2)
115     C
116     C      PLOT A PLUS (+) AT THE VERTEX POSITION
117     C
118     C      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     C      CALL LINCNT(60)
126     IFLGP=IFLAG+1
127     WRITE(IFD,40)(ITITLE(I,IFLGP),I=1,5),VMAX
128     WRITE(IFD,30) JNM,NAME,T,NCYC
129     RETURN
130     C
131     30 FORMAT(1H ,4XA10,8A10,3X2HT=,1PE12.5,1X6HCYCLES=,I5)
132     40 FORMAT(1H ,5A10/18X5HVMAX=,1PE12.5)
133     END

```

```

1      SUBROUTINE W1ROW
2      C

```

```

3   C      ROUTINE TO WRITE ROW J FROM SCM BUFFER ONE TO LCM
4   C
5   C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
6   C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
7   C
8   *      ----- BEGIN COMDECK PARAM -----
9   COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
10  1 NLCP3,NLCP4,IPLMSZ
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,DTA,DTFAC,
18  1 DTO(10),DTOC(10),DT02,DT08,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,QM,OMCYL,PXCONV
24  2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
25  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
26  COMMON/YSC2/TUGI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
27  1 ILNG,NILNG,TP3,TUPOT,TDQ3AV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
28  2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
29  3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
30  4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
31  5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMON,TMAX,TGMAX,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 -----
38  DATA NE/B/
39  IEC=(J-1)*NQI
40  CALL ECWR(AASC,IEC,NQI,IDUM)
41  RETURN
42  END

```

```

1      SUBROUTINE YAQUI2
2
3      C      MAIN ROUTINE FOR RUNNING CODE PHYSICS
4
5      C      ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T=3
6      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
7
8      *      ----- BEGIN COMDECK PARAM -----
9      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCPI,NLCP2,
10     1 NLCP3,NLCP4,IPLMSZ

```

```

11   *      ===== END COMDECK PARAM =====
12   *      ===== BEGIN COMDECK YSTORE =====
13   *      ===== BEGIN COMDECK YAQDIM =====
14   *      DIMENSION X(1),XPAR(1),R(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),DLSRQQ(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   C      COMMON/YSC1/AASC(9600)
27   C      COMMON/YSC2/AA(1),ANC,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
28   *      1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29   *      2 FIPYB,FIPYT,FIYL,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
32   C      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33   *      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34   *      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZZIE,REZY0,RIBAR,RIBJB,
35   *      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36   C      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37   *      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
38   *      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,PIYBO,IYBO,YCNVLD,
39   *      3 XCNVLD,FIXRO,FXLQ,IXR0,IXL0,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLIM,
40   *      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
41   *      5 ,JDTV,IDTC,JDTA,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMAX,ITH,JTM,ITG,JTG
42   *      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43   C      COMMON/YSC2/ZZ
44   C      COMMON/YSC4/ITAB(ITABP)
45   C      COMMON/YSC4/ITAB(1000)
46   C      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
47   *      ===== END COMDECK YAQSC =====
48   *      ===== BEGIN COMDECK YAQEQ =====
49   *      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50   *      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
51   *      1 (8),E,ETIL,AREA,XR13K),
52   *      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53   *      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54   *      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),G,CQ,ROL),(AASC(17
55   *      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56   *      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57   *      7 ),(AASC(25),DLSRQQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
58   *      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59   *      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60   *      1 AVYSV,X24K)
61   *      REAL M,MP,MPAR,MTIL
62   *      ===== END COMDECK YAQEQ =====
63   *      ===== END COMDECK YSTORE =====
64   *      ===== BEGIN COMDECK ASTORE =====
65   *      COMMON/ASTC/AT(100),FT(100)
66   *      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
67   *      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO

```

```

68      1 ),(AT(9),YCO),(FT,CON)
69      ★      ***** END COMDECK ASTORE      *****
70      ★      ***** BEGIN COMDECK PCALL      *****
71      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
72      ★      ***** END COMDECK PCALL      *****
73      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
74      LOGICAL EDIT, LAST, DUMP
75      REAL LAMD
76      DATA IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX/0,0,4*0./
77      DATA ISWTCH/-1/
78      C      INITIALIZE ON STARTUP OR RESTART
79      C
80      C      NUMIT=0
81      C      TEG=0.
82      C      DTR=0.
83      C      TMASS=0.
84      C      DTVSAV=0.
85      C      DTCSAV=0.
86      C
87      C      INITIALIZE IF THE TURBULENCE IS ON
88      C
89      C      IF(.NOT.TURB) GO TO 10
90      C      TNEG=0.
91      C      TP3=0.
92      C      SAV1=0.
93      C      TUQENG=0.
94      C      TDQSAV=1.
95      C
96      10 CONTINUE
97      C
98      C      DO CERTAIN INITIALIZATION ONLY DURING RESTART
99      C
100     C      IF(.NOT.RESTRT) GO TO 20
101     C      CALL CINIT
102     C      IF(TURB) CALL TRBERF
103     C      GO TO 30
104     C
105     C      20 CONTINUE
106     C
107     C      STARTUP, CONTINUE INITIALIZATION.
108     C      DTV=0.
109     C      IDTV=0
110     C      JDTV=0
111     C      DTC=0,
112     C      IDTC=0
113     C      JDTCA=0
114     C
115     C      -----
116     C      BEGIN LOOP OVER CYCLES
117     C      -----
118     C
119     C      30 CONTINUE
120     C
121     C      INITIALIZE TURBULENCE QUANTITIES
122     C
123     C      IF(.NOT.TURB) GO TO 40
124     C      TNEGSV=TNEGSV+TNEG

```

```

125      TNEG=0,
126      EP1=TK+TI+TP3+TUDENG
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      C      TOLD=T2
133      C
134      C      IF JUST FINISHING INITIALIZATION, GO DO AN EDIT FIRST
135      C
136      C      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      C      NCYC=NCYC+1
142      C
143      C      SET THE NEW TIMESTEP, SAVE THE OLD ONE IN DTLAST.
144      C
145      C      DTLAST=DT
146      C
147      C      ON THE FIRST CYCLE, SET DEFAULT DTC AND DTV
148      C
149      C      IF(NCYC.NE.1) GO TO 60
150      C      DTC=DT
151      C      DTV=DT
152      60 CONTINUE
153      C
154      C      ON THE SECOND CYCLE, BOOST DT BY A FACTOR OF 10
155      C
156      C      IF(NCYC.NE.2) GO TO 70
157      C      DT=10.*DT
158      C      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      C      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      C      DTFAC=1.25
169      C      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. DTP05 IS DT BEFORE
173      C      ANY SUCH ADJUSTMENT. DTSV IS THE TIMESTEP SAVED FOR FUTURE
174      C      REFERENCE.
175      C
176      C      DTP05=DT
177      C      DTSV=DT
178      C      IF(T+DT.GT.TOUT) DT=TOUT-T
179      C      IF(DT.LT.1.E-8) DT=1.E-8
180      C
181      C      UPDATE T

```

```

182 C T=T+DT
183 C IF(DT,NE,DTPOS) T=TOUT
184 C
185 C *****TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS*****
186 C *****TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS*****
187 C *****TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS*****
188 C *****TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS*****
189 C *****TEMPORARY PATCH FOR MYLAR BALLOON CALCULATIONS*****
190 C IF(T,GT,.6) ANC=.05
191 C
192 C ILNG IS ONE IF THIS IS AN EDIT CYCLE
193 C
194 C ILNG=0
195 C IF(ABS(TOUT-T),LT,EM10) ILNG=1
196 C
197 C SET QUANTITIES DERIVED FROM THE TIMESTEP
198 C
199 C RDT#1./DT
200 C DT02=.5*DT
201 C DT08=.125*DT
202 C
203 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
204 C PHASE 1 = EXPLICIT LAGRANGIAN PART
205 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
206 C
207 C CALL YPH1
208 C
209 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
210 C PHASE 2 = PRESSURE ITERATION
211 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
212 C
213 C CALL YPH2
214 C
215 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
216 C PHASE 3,PART I = CONTINUOUS REZONE
217 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
218 C
219 C CALL YPH3
220 C
221 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
222 C PHASE 3,PART II = FLUXING
223 C $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
224 C
225 C 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 C DTR#1.E30
233 C VMAX#1.E10
234 C CALL START
235 C DO 110 J#2,JP2
236 C DO 100 I#1,IP1
237 C U(IJ)=UG(IJ)=U(IJ)
238 C V(IJ)=VG(IJ)=V(IJ)

```

```

239      VMAX=AMAX1(VMAX,ABS(U(IJ)),ABS(V(IJ)))
240      IF(J,NE,JTM,OR,I,NE,ITM) GO TO 80
241      IPJ=IJ+NQ
242      IPJP=IJP+NQ
243      XTMAX=.25*(X(IJ)+X(IPJ)+X(IJP)+X(IPJP))
244      YTMAX=.25*(Y(IJ)+Y(IPJ)+Y(IJP)+Y(IPJP))
245      80 CONTINUE
246      IF(I,EQ,IP1) GO TO 90
247      IPJ=IJ+NQ
248      IF(U(IJ),NE,0.,AND,V(IJ),NE,0.) DTR=AMIN1(DTR,ABS((X(IPJ)-X(IJ))/U
249      I(IJ)),ABS((Y(IJP)-Y(IJ))/V(IJ)))
250      90 CONTINUE
251      IJ=IJ+NQ
252      IJP=IJP+NQ
253      100 CONTINUE
254      CALL LOOP
255      110 CONTINUE
256      CALL DONE
257      DTR=.1*DTR
258      C
259      C SEE IF THIS IS AN EDIT CYCLE
260      C
261      IF(ILNG,NE,1) GO TO 120
262      C
263      C YES, PLOT VELOCITY OF GRID RELATIVE TO THE FLUID.
264      C
265      FIYB=FIYB0
266      IYB=IYB0
267      YCONVP=YCNVLD
268      XCONVP=XCNVLD
269      FIXR=FIXR0
270      FIXL=FIXL0
271      IXR=IXR0
272      IXL=IXL0
273      YLB=YB
274      CALL ADV(1)
275      CALL VELPLT(VMAX,2)
276      120 CONTINUE
277      C
278      C STORE U,V,AND RM INTO THEIR FINAL LOCATIONS AND ZERO RMP
279      C
280      CALL START
281      TP3=0,
282      DO 140 J=2,JP2
283      DO 130 I=1,IP1
284      U(IJ)=UP(IJ)
285      V(IJ)=VP(IJ)
286      RM(IJ)=RMP(IJ)
287      TP3=TP3-GZ*Y(IJ)/RM(IJ)
288      RMP(IJ)=0,
289      130 IJ=IJ+NQ
290      CALL LOOP
291      140 CONTINUE
292      CALL DONE
293      C
294      C MOVE THE PARTICLES IF THERE ARE ANY
295      C

```

```

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

```

```

353      NCQ=NCYC
354      C
355      C      GO GET THE VORTICITY
356      C
357      C      CALL GETOMG
358      C
359      C      INITIALIZE TUQ AND TUSV
360      C
361      C      TUSV=0.
362      C      CUTOFF=-.1*QMN*TUQI
363      C      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      C      CALL TRBERF
380      C
381      C      INDICATE THAT THE TURBULENCE HAS BEEN SEEDED
382      C
383      DO 230 IPX=6,IFD,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)=LEVEL*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.

```

```

410      C
411      C      EDIT#,TRUE,
412      C      TOUT=TOUT+DTO(IDTO)
413      C      IF(T+EM10.LT.DTOC(IDTO)) GO TO 280
414      C      TOUT=DTOC(IDTO)+DTO(IDTO+1)
415      C      IDTO=IDTO+1
416      C      280 CONTINUE
417      C
418      C      IF TURBULENCE HAS BEEN SEEDED,DO AN EDIT
419      C
420      C      IF(NCYC.EQ.NCA) EDIT#=,TRUE,
421      C
422      C      IF THIS IS STARTUP OR THE FIRST CYCLE,DO AN EDIT
423      C
424      C      IF(NCYC.LE,1) EDIT#=,TRUE,
425      C
426      C      IF THE ITERATION COUNT HAS BEEN EXCEEDED,DO AN EDIT FOR
427      C      DIAGNOSTIC PURPOSES
428      C
429      C      IF(NUMIT.GE,500) EDIT#=,TRUE,
430      C
431      C      SEE IF THIS IS THE LAST CYCLE
432      C
433      C      LAST#=,FALSE,
434      C
435      C      QUERY TTY
436      C
437      C      CALL TTYTST(IFLAG)
438      C      IF(IFLAG.NE,0) LAST#=,TRUE,
439      C      IF(T.GE.TWFIN) LAST#=,TRUE,
440      C      IF(T2.GE.TLIM) LAST#=,TRUE,
441      C      IF(NCYC.GE,NCLST) LAST#=,TRUE,
442      C      IF(LAST,AND,IFLAG.EQ,0) EDIT#=,TRUE,
443      C
444      C      SEE IF IT IS TIME TO DUMP
445      C
446      C      DUMP#=,FALSE,
447      C      IF(LAST) DUMP#=,TRUE,
448      C      IF(MOD(NCYC,JDUMP).EQ,0,AND,NCYC.NE,0) DUMP#=,TRUE,
449      C      IF(EDIT,AND,NCYC.GT,1) DUMP#=,TRUE,
450      C
451      C      DO THE EDIT IF REQUIRED
452      C
453      C      IF(.NOT.EDIT) GO TO 290
454      C      IF(FILM) CALL YPLOT
455      C      CALL YEDIT
456      C      290 CONTINUE
457      C
458      C      DO THE DUMP IF REQUIRED
459      C
460      C      IF(DUMP) CALL YDUMP
461      C
462      C      QUIT IF THIS IS THE LAST CYCLE
463      C
464      C      IF(LAST) RETURN
465      C
466      C      IF THIS IS THE EDIT AFTER INITIALIZATION, IMMEDIATELY BEGIN

```

```

467      C      CYCLE 1
468      C
469      IF(NCYC.EQ.0) GO TO 50
470      GO TO 30
471      C
472      300 FORMAT(1H ,20X24HTOTAL INTERNAL ENERGY = ,1PE14.7/1H ,20X
473          1 23HTOTAL KINETIC ENERGY = ,E14.7/1H ,20X
474          2 31HTOTAL GRAV. POTENTIAL ENERGY = ,E14.7/1H ,20X
475          3 24HTOTAL RADIAL MOMENTUM = ,E14.7/1H ,20X
476          4 23HTOTAL AXIAL MOMENTUM = ,E14.7)
477      310 FORMAT(1H ,20X7HVMAX * ,1PE12.5,11H AT VERTEX ,2I5)
478      320 FORMAT(1H ,20X020,025)
479      330 FORMAT(/1H ,27HTURBULENCE SEEDED ON CYCLE ,I4,13H WITH TUQI = ,E12
480          1 .5,IX1H./1H ,5X33HTOTAL TURBULENCE ENERGY SEEDED = ,E12.5)
481      340 FORMAT(1H ,6(1H*),7H CYCLE ,I5,4H, T=,1PE12.5,5H, DT=,E12.5,
482          1 5H, CP=,E12.5/1H ,20X7HGRINDS=,E12.5,8H, NUMIT=,I3,7H, CIRC=,E12
483          2 .5/1H ,20X4HDTV=,E12.5,7H, IDTV=,I3,7H, JDTV=,I3/1H ,20X4HDTC=,
484          3 E12.5,7H, IDTC=,I3,7H, JDTCE=,I3/1H ,20X5HTMAX=,E12.5,6H, ITM=,I3,
485          4 6H, JTME=,I3,8H, XTMAX=,E10.3,8H, YTMAX=,E10.3/1H ,20X5HTGMX=,E12
486          5 .5,6H, ITGE=,I3,6H, JTGS=,I3/1H ,20X5HPRIT=,E10.3,7H, PTOP=,E10.3,
487          6 7H, PBOT=,E10.3,8H, PDIAM=,E10.3,8H, PAVHT=,E10.3)
488      END

```

```

1      SUBROUTINE YARSRT
2
3      C      ROUTINE TO RESTART A YAQUI PROBLEM
4
5      C      WRITTEN BY J.L.NORTON,LA8L T=3,1975
6
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 YSTORE -----
12     *      ----- BEGIN COMDECK YAQDIM -----
13     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC
14     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC8Q(1),E(1),ETIL(1),RVOL
15     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
16     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROG(1),CAPGAM(1),TUQ
17     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRRDP(1),TUQVEC(1),MTIL(1),
18     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
19     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
20     *      ----- END COMDECK YAQDIM -----
21     *      ----- BEGIN COMDECK YAQSC -----
22     LOGICAL RESTRT,FILM,PAPER,TURB
23     REAL LAM,MU
24     C      COMMON/YSC1/AASC(NSCP1)
25     COMMON/YSC1/AASC(9600)
26     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0H,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
27     1 DTO(10),DTOC(10),DTO2,DTDB,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
28     2 FIPYB,FIPYT,FIXL,FIYR,FIYT,FRZXR,GR,GRDVEL,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,JUNFO2,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,THIRD,NCLST,TOUT,TWFIN
35      COMMON/YSC2/TUQI,TUSI,NCG,TNEG,NEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
36      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
37      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
38      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
39      4 ROMFXR,ROMFYR,ROMFYB,JDUMP,TWTHR,DTR,MASS,DTVSAV,DTCSAV,IDTV
40      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMX,ITM,JTM,ITG,JTG
41      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
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),RD),(AASC(7),DELSM,RCSSQ,MP),(AASC
50      1 (8),E,ETIL,AREA,XR13K),
51      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,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),GROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
56      7 ),(AASC(25),DLSR0Q,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(IJ),IJ=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      CINTCYC=1
80      CTAPE=0
81      CCALL NAMLST(RCYCLE,5,IEFLAG)
82      C
83      C CHECK FOR INPUT ERRORS
84      C
85      C IF(IEFLAG,NE,0) CALL UNCLE(4,6HYARSRT,27,

```

```

86      1 27HRCYCLE NAMELIST INPUT ERROR)
87      C
88      C      OPEN THE INPUT DUMP FILE
89      C
90      C      CALL OPENIT(7,1)
91      C
92      C      NO ERRORS. CALL ROUTINE TO GET TAPE IF TAPE NO. WAS READ,
93      C      (ONLY FUNCTIONAL ON CROS/7600)
94      C
95      C      IF(TAPE,NE,0) CALL GETTPE(TAPE)
96      C      REWIND 7
97      C      JNSC=LOCF(ZZ)-LOCF(AA)+1
98      C
99      C      READ THE NEXT CYCLE ON THE TAPE
100     C
101     10 CONTINUE
102     CALL SCBUFF(AA,JNSC,7,0,1,IERROR)
103     C
104     C      CHECK FOR ERRORS
105     C
106     IF(IERROR,GE,0) GO TO 30
107     20 CALL UNCLE(1,6HYARSRT,18,18HSCBUFF INPUT ERROR)
108     30 CONTINUE
109     IF(IERROR,EQ,0) GO TO 40
110     C
111     C      SEE IF TERMINAL DUMP RECORD WAS FOUND
112     C
113     IF(AA(1),EQ,666) GO TO 70
114     GO TO 20
115     40 CONTINUE
116     C
117     C      NO. PRINT CYCLE NO. OF DUMP LAST READ,
118     C
119     DO 50 IPX=6,IFD,6
120     50 WRITE(IPX,130) NCYC
121     C
122     C      SEE IF LAST CYCLE IS DESIRED
123     C
124     IF(INTCYC,EQ,(-1)) GO TO 60
125     C
126     C      NO. SEE IF WE HAVE FOUND THE CORRECT CYCLE,
127     C
128     IF(NCYC=INTCYC) 60,80,110
129     60 CONTINUE
130     C
131     C      CORRECT CYCLE NOT FOUND YET. SKIP REST OF DUMP AND GO READ NEXT,
132     C
133     CALL RTAPE
134     GO TO 10
135     70 CONTINUE
136     C
137     C      DUMP TAPE TERMINATION FOUND. FATAL UNLESS INTCYC IS -1.
138     C
139     IF(INTCYC,NE,(-1)) CALL UNCLE(4,6HYARSRT,17,17HEOF ON INPUT TAPE)
140     C
141     C      O.K. DUMP WAS THE LAST ONE READ.
142     C

```

```

143      GO TO 90
144      80 CONTINUE
145      CALL RTAPE
146      90 CONTINUE
147      C
148      C      PRINT THE CYCLE NO., PROBLEM NAME, AND PROBLEM TIME
149      C
150      DO 100 IPX=6,IFD,6
151      WRITE(IPX,140) NCYC
152      100 WRITE(IPX,150) NAME,T
153      C
154      C      RESTART COMPLETED. SEE IF THERE ARE CHANGES TO INPUT VARIABLES.
155      C
156      CALL YINPUT
157      TLIM=TLIMSV
158      C
159      C      CLOSE THE INPUT DUMP FILE
160      C
161      CALL CLOSIT(7)
162      C
163      C      REFRESH THE CP TIME
164      C
165      CALL SECOND(T2)
166      RETURN
167      C
168      C      DUMP NOT ON TAPE. FATAL ERROR.
169      C
170      110 CONTINUE
171      CALL UNCLE(4,6HYARSRT,23,23HRESTART CYCLE NOT FOUND)
172      120 CONTINUE
173      CALL UNCLE(4,6HYARSRT,36,36HRCYCLE NAMELIST INITIALIZATION ERROR)
174      C
175      130 FORMAT(1H ,6HCYCLE ,15,6H FOUND)
176      140 FORMAT(1H .21HRESTARTING FROM CYCLE,15)
177      150 FORMAT(1H ,8A10/1H ,2HT=,1PE12.5/1H )
178      END

```

```

1      SUBROUTINE YASET
2      C
3      C      ROUTINE TO GENERATE A NEW PROBLEM
4      C
5      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T=3
6      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T=3,1974
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 YSTORE *****
13     *      ***** BEGIN COMDECK YAQDIM *****
14     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSMC

```

```

15      1 1),VG(1),RG(1),SIE(1),MP(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),GRRDR(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      C COMMON/YSC1/AASC(9600)
27      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,OTC,DTFAC,
28      1 DTO(10),DTOC(10),DTO2,DTO8,DTP09,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29      2 FIPYB,FIPYT,FIXL,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
32      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
35      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36      COMMON/YSC2/TUQI,TUSI,NCG,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,OLEVEL,TQ,IST,
38      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      3 XCNVLD,PIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,TE,DTR,TMASS,DTVS,DTCSA,DTCSA,IDTV
41      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43      C COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      C COMMON/YSC4/ITAB(1000)
46      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
47      * ----- END COMDECK YAQSC -----
48      * ----- BEGIN COMDECK YAQEQ -----
49      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(AASC
50      1 AASC(4),U),(AASC(5),V),(AASC(6),RD),(AASC(7),DELSM,RCSQ,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRRDR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ----- END COMDECK YAQEQ -----
63      * ----- END COMDECK YSTORE -----
64      * ----- BEGIN COMDECK CTAB -----
65      C COMMON/CTABC/XTAB(ITABXP),YTAB(ITABYP)
66      C COMMON/CTABC/XTAB(101),YTAB(151)
67      * ----- END COMDECK CTAB -----
68      DATA NE/0/
69      C READ THE PROBLEM TITLE
70      C
71      C
```

```

72      READ(5,320) NAME
73      C
74      C      INITIALIZE THE PROBLEM VARIABLES
75      C
76      C      CALL DEFINE
77      C
78      C      READ THE INPUT VARIABLES
79      C
80      C      CALL YINPUT
81      C
82      C      OUTPUT THE PROBLEM TITLE
83      C
84      DO 10 IPX=6,IFD,6
85      10 WRITE(IPX,320) NAME
86      C
87      C      IF NCQ,LT,0, THERE WILL BE NO TURBULENCE
88      C
89      C      IF(NCQ,LT,0) GO TO 70
90      C
91      C      THERE WILL BE TURBULENCE, PRINT OUT THE INPUT QUANTITIES
92      C
93      DO 20 IPX=6,IFD,6
94      20 WRITE(IPX,290) QLEVEL,TUQI,TUSI,NCQ,TQ,TSTRTD,WMAXEF,RMINEF
95      C
96      C      GO TO 50
97      C
98      C      THERE WILL BE NO TURBULENCE, INDICATE SUCH,
99      C
100     30 CONTINUE
101     DO 40 IPX=6,IFD,6
102     40 WRITE(IPX,300)
103     50 CONTINUE
104     C
105     C      PRINT GENERAL INPUT VARIABLES
106     DO 60 KT=6,IFD,6
107     WRITE(KT,330) IBAR,JBAR,IUNF,JUNF,JCEN,DR,DZ,CYL,GRDVEL,A0,A0M,B0,
108     1 KXI
109     WRITE(KT,340) MU,LAM,OM,EP8,GR,GZ
110     WRITE(KT,350) FREZXR,FREZYT,FREZYB,ZORIG,YB,REZY0,REZRON,REZSIE
111     WRITE(KT,360) GZP
112     WRITE(KT,370) T,DT,NCLST,TWFIN,PAPER,FiLM
113     WRITE(KT,260) ANC,A0FAC
114     WRITE(KT,310) IST
115     WRITE(KT,380)(DTO(N),N=1,10)
116     WRITE(KT,390)(DTOC(N),N=1,10)
117     60 CONTINUE
118     C
119     C      CALCULATE AND STORE PROBLEM CONSTANTS
120     C
121     IM1=IBAR-1
122     IP1=IBAR+1
123     IP2=IBAR+2
124     JP1=JBAR+1
125     JP2=JBAR+2
126     JP4=JBAR+4
127     RIBAR=1./FLOAT(IBAR)
128     RIBJB=1./FLOAT(IBAR*JBAR)

```

```

129      NQIB=NQ*IBAR
130      DMCYL#1.=CYL
131      EM10W1,E=10
132      IF(FREZXR,NE.1,) ROMFXR#1,/(1,-FREZXR)
133      IF(FREZYT,NE.1,) ROMFYT#1,/(1,-FREZYT)
134      IF(FREZYB,NE.1,) ROMFYB#1,/(1,-FREZYB)
135
136      C      NQI IS THE NO. OF WORDS OF DATA NEEDED TO STORE ONE ROW. IT IS
137      C      THE SIZE OF A SMALL CORE BUFFER.
138      C
139      C      NQI=NQ*IP1
140      C
141      C      SEE ROUTINE =START= FOR DEFINITIONS
142      C
143      ISC2=NQI+1
144      ISC3=ISC2+NQI
145
146      C      SEE ROUTINE =STARTD= FOR DEFINITIONS
147      C
148      ITV#JP1*NQI
149
150      C      NSC IS THE NO. OF WORDS OF SMALL CORE IN COMMON YSC2
151      C
152      NSC=LOCF(ZZ)=LOCF(AA)+1
153
154      C      NLC IS THE NO. OF WORDS OF LARGE CORE USED IN COMMON YLC1. IT
155      C      SHOULD NOT EXCEED THE PARAMETER VALUE.
156
157      C      NLC#JP4*NQI
158
159      C      GO CHECK THE LARGE AND SMALL CORE PARAMETERS
160
161      C      CALL PCHK
162
163      C      IDTO IS THE SUBSCRIPT IN THE DTOC ARRAY SUCH THAT
164      C      DTOC(IDTO=1),LT,T,LE,DTOC(IDTO)
165
166      C      IDTO=1
167
168      C      TOUT IS THE TIME AT WHICH OUTPUT SHOULD OCCUR NEXT
169
170      C      TOUT#DTOC(1)
171      C      IF(TOUT,GT,T) GO TO 90
172      70 CONTINUE
173      C      TOUT=TOUT+DTOC(IDTO)
174      C      IF(TOUT,LE,DTOC(IDTO)) GO TO 80
175      C      TOUT#DTOC(IDTO)
176      C      IDTO=IDTO+1
177      80 IF(TOUT,GT,T) GO TO 90
178      C      GO TO 70
179      90 CONTINUE
180
181      C      THE TIMESTEP FOR THE FIRST CYCLE IS REDUCED BY A FACTOR OF 10
182      C      FROM THE SPECIFIED INITIAL DT
183
184      C      DT#,1*DT
185

```

```

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 C DTPOS=DT
191 C
192 C DTSV IS THE TIMESTEP FROM THE LAST CYCLE
193 C
194 C DTSV=DT
195 C
196 C NCYC IS THE CYCLE NO.
197 C
198 C NCYC=0
199 C
200 C COLAMU IS 1/(2/3*(2*MU+LAM))
201 C
202 C 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 C NILNG=0
208 C
209 C IUNF MUST BE AT LEAST 1
210 C
211 C IUNF=MAX0(IUNF,1)
212 C
213 C JUNF MUST BE AT LEAST 2
214 C
215 C JUNF=MAX0(JUNF,2)
216 C JUNFO2=JUNF/2
217 C
218 C IF JCEN IS ZERO, SET IT TO JBAR/2
219 C
220 C IF(JCEN,EQ.0) JCEN=JBAR/2
221 C
222 C GO GENERATE THE PARTICLES
223 C
224 C CALL PRGEN
225 C
226 C GO GENERATE THE MESH
227 C
228 C CALL MSHMKR
229 C
230 C GO SET THE PLOT QUANTITIES
231 C
232 C CALL FILMCO
233 C
234 C ****LOCATE WHICH CELL EACH PARTICLE IS IN IF THERE ARE ANY PARTICLES****
235 C ****LOCATE WHICH CELL EACH PARTICLE IS IN IF THERE ARE ANY PARTICLES****
236 C ****LOCATE WHICH CELL EACH PARTICLE IS IN IF THERE ARE ANY PARTICLES****
237 C
238 C 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.

```

```

243      C
244      IF(JP2.GT.ITABYP) CALL UNCLE(4,SHYASET,21,21HYTAB ARRAY OVERFLOWED
245      1 )
246      IF(IP1.GT.ITABXP) CALL UNCLE(4,SHYASET,21,21HXTAB ARRAY OVERFLOWED
247      1 )
248      CALL START
249      DO 120 J=2,JP2
250      YTAB(J)=Y(IJ)
251      IF(J.GT.2) GO TO 110
252      DO 100 I=1,IP1
253      XTAB(I)=X(IJ)
254      100 IJ=IJ+NQ
255      110 CALL LOOP
256      120 CONTINUE
257      C
258      C   INITIALIZE THE LCM ADDRESS AND THE PARTICLE COUNT
259      C
260      IECP=1
261      NPPT=0
262      C
263      C   BRING IN A BUFFER-FULL OF PARTICLE DATA
264      C
265      C   LPB IS SET IN PARTGEN
266      C
267      130 CALL ECRD(AASC,NLCP1+IECP-1,LPB,IDUM)
268      KP=1
269      140 CONTINUE
270      C
271      C   LOCATE THE I AND J OF THE CELL CONTAINING THE PARTICLE IN QUESTION
272      C
273      DO 150 J=2,JP2
274      IF(YTAB(J).GT.YPAR(KP)) GO TO 170
275      150 CONTINUE
276      DO 160 IPX=6,IFD,6
277      160 WRITE(IPX,270) YTAB(JP2),YPAR(KP),KP
278      CALL UNCLE(1,SHYASET,23,23HJ OF PARTICLE NOT FOUND)
279      170 DO 180 I=1,IP1
280      IF(XTAB(I).GT.XPAR(KP)) GO TO 200
281      180 CONTINUE
282      DO 190 IPX=6,IFD,6
283      190 WRITE(IPX,280) XTAB(IP1),XPAR(KP),KP
284      CALL UNCLE(1,SHYASET,23,23HI OF PARTICLE NOT FOUND)
285      200 CONTINUE
286      C
287      C   INCREMENT THE PARTICLE COUNT
288      C
289      NPPT=NPPT+1
290      C
291      C   I AND J OF CELL FOUND. CODE IT AND STORE IN ITAB. FIRST MAKE SURE
292      C   THAT STORAGE WILL NOT BE OVERFLOWED.
293      C
294      IF(NPPT.GT.ITABP) CALL UNCLE(4,SHYASET,21,21HITAB ARRAY OVERFLOWED
295      1 )
296      ITAB(NPPT)=(J-2)*IP1+I-1
297      C
298      C   SEE IF ALL PARTICLES HAVE BEEN PROCESSED
299      C

```

```

300      IF(NPPT,GE,NPT) GO TO 210
301      C
302      C      NO, SEE IF THE BUFFER NEEDS TO BE RELOADED.
303      C
304      KP=KP+3
305      IF(KP,LT,LPB) GO TO 140
306      C
307      C      YES, INCREMENT THE LCM POINT AND REFILL THE BUFFER
308      C
309      IECPR=IECP+LPB
310      GO TO 130
311      210 CONTINUE
312      C
313      C      *****
314      C      CALCULATE THE CELL-CENTERED VOLUMES (RECIPROCALS),MASSES,AND
315      C      TOTAL ENERGIES FOR ALL CELLS
316      C      *****
317      C
318      CALL START
319      DO 230 J=2,JP1
320      DO 220 I=1,IBAR
321      IPJ=IJ+NQ
322      IPJP=IJP+NQ
323      X1=X(IPJ)
324      Y1=Y(IPJ)
325      R1=R(IPJ)
326      X2=X(IPJP)
327      Y2=Y(IPJP)
328      R2=R(IPJP)
329      X3=X(IJP)
330      Y3=Y(IJP)
331      R3=R(IJP)
332      X4=X(IJ)
333      Y4=Y(IJ)
334      R4=R(IJ)
335      RVOL(IJ)=8.0/((R1+R2+R3+R4)*((X1-X3)*(Y2-Y4)-(Y1-Y3)*(X2-X4)))
336      M(IJ)=RO(IJ)/RVOL(IJ)
337      GRROR(IJ)=0.
338      GRROR(IPJ)=0.
339      GRROR(IPJP)=0.
340      GRROR(IJP)=0.
341      GRROR(IJ)=0.
342      GRROR(IPJP)=0.
343      GRROR(IJP)=0.
344      GRROR(IPJ)=0.
345      E(IJ)=SIE(IJ)+.125*(U(IPJ)**2+U(IPJP)**2+U(IJP)**2+U(IJ)**2+V(IPJ)
346      1.**2+V(IPJP)**2+V(IJP)**2+V(IJ)**2)
347      IJ=IPJ
348      220 IJP=IPJP
349      CALL LOOP
350      230 CONTINUE
351      CALL DONE
352      C
353      C      *****
354      C      COMPUTE THE VERTEX MASSES,EXCEPT FOR BOUNDARY VERTICES,THE VERTEX
355      C      MASS IS JUST 1/4 OF THE MASSES OF ALL THE CELLS HAVING THE
356      C      VERTEX AS A CORNER.

```

```

357      C ****
358      C
359      CALL STARTD
360      DO 250 JJ=2,JP2
361      J#JP#*JJ
362      DO 240 II=1,IP1
363      I*IP2-II
364      IMJ#IJ=NQ
365      IMJM#IJM=NQ
366      XX=0.0
367      IF(I,NE,IP1,AND,J,NE,2) XX=M(IJM)
368      IF(I,NE,IP1,AND,J,NE,JP2) XX=XX+M(IJ)
369      IF(I,NE,1,AND,J,NE,JP2) XX=XX+M(IMJ)
370      IF(I,NE,1,AND,J,NE,2) XX=XX+M(IMJM)
371      RM(IJ)=4./XX
372      IJ=IMJ
373      240 IJM=IMJM
374      CALL LOOPD
375      250 CONTINUE
376      RETURN
377      C
378      260 FORMAT(1H ,3X4HANC#,1PE12.5/1H ,1X6HA0FAC#,E12.5)
379      270 FORMAT(1H,12HYTAB,YPAR,KP,2E20.7,I10)
380      280 FORMAT(1H,12HXTAB,XPAR,KP,2E20.7,I10)
381      290 FORMAT(1H ,24HTHERE WILL BE TURBULENCE/1X7HQLEVEL#,1PE12.5/3X
382      1 SHTUQI#,E12.5/3X5HTUSI#,E12.5/4X4HNCQ#,IS/5X3HTQ#,E12.5/1H ,7HTST
383      1 E12.5/1H ,7HWMAXEF#,E12.5/1H ,7HRMINEF#,E12.5)
384      300 FORMAT(15H NO TURBULENCE)
385      310 FORMAT(1H ,3X,4HIST#,I4)
386      320 FORMAT(8A10)
387      330 FORMAT(3X,SHIBAR#I4/3X,5HJBAR#I4/3X,5HIUNF#I4/3X,5HJUNF#I4/3X,
388      1 SHJCEN#I4/5X,3HDR#1PE12.5/5X,3HDZ#E12.5/4X,4HCYL#E12.5/8H GRDVEL#
389      2 E12.5/SX,3HA0#E12.5/4X,4HA0M#E12.5/SX,3HB0#E12.5/4X,4HKX#I#I3)
390      340 FORMAT(5X,3HMU#1PE12.5/4X,4HLAM#E12.5/SX,3HOM#E12.5/4X,4HEPS#E12.5
391      1 /5X,3HGR#E12.5/5X,3HGZ#E12.5)
392      350 FORMAT(1H ,7HFREZXR#,1PE12.5/1H ,7HFREZY#E12.5/1H ,7HFREZYB#,E12
393      1 ,5/1H ,1X6HZORIG#,E12.5/1H ,4X3HYB#,E12.5/1H ,1X6HREZY#E12.5/
394      2 1H ,7HRE2RON#,E12.5/1H ,7HRE2SIE#,E12.5)
395      360 FORMAT(4X,4HGZP#E12.5)
396      370 FORMAT(1H ,5X2HT#,1PE12.5/1H ,4X3HDT#,E12.5/1H ,1X6HNCLST#,I6/1H ,
397      1 1X6HTWFIN#,E12.5/1H ,1X6HPAPER#,I1/1H ,2X5HFILM#,I1)
398      380 FORMAT(12H DTOC(I=10)#4(1PE14.5)/(12X,4E14.5))
399      390 FORMAT(12H DTOC(I=10)#4(1PE14.5)/(12X,4E14.5))
400      END

```

```

1      SUBROUTINE YDUMP
2
3      C ROUTINE TO DO A YAQUI DUMP
4
5      C WRITTEN BY J.L.NORTON,LASL T=3,1975
6

```

```

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 YSTORE      *****
12  *      ***** BEGIN COMDECK YAQDIM      *****
13  DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
14  1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RCSO(1),E(1),ETIL(1),RVOL
15  2 (1),M(1),RM(1),VP(1),PL(1),UP(1),UTIL(1),UL(1),CG(1),VTIL(1)
16  3 ,VL(1),ROL(1),AVXSV(1),AVY8V(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
17  4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
18  5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
19  6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
20  *      ***** END COMDECK YAQDIM      *****
21  *      ***** BEGIN COMDECK YAQSC      *****
22  LOGICAL RESTRT,FILM,PAPER,TURB
23  REAL LAM,MU
24  C COMMON/YSC1/AASC(NSCP1)
25  COMMON/YSC1/AASC(9600)
26  COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
27  1 DTO(10),DTOC(10),DTO2,DTO8,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
28  2 FIPYB,FIPYT,FIXL,FIXR,FIYT,FREZXR,GR,GRDVEL,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,JUNFO2,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,REZYD,RIBAR,RIBJB,
34  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
35  COMMON/YSC2/TUQI,TUSI,NCO,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
36  1 ILNG,NILNG,TP3,TUPOT,TDOSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
37  2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
38  3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
39  4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
40  5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
41  6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
42  COMMON/YSC2/ZZ
43  C COMMON/YSC4/ITABP(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),RO),(AASC(7),DELSM,RCSO,MP),(AASC
50  1 (8),E,ETIL,AREA,XR13K),
51  2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,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,CG,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),DLSROI,Y13K),(AASC(24),GZSV
56  7 ),(AASC(25),DLSROQ,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 AVY8V,X24K)
60  REAL M,MP,MPAR,MTIL
61  *      ***** END COMDECK YAQEQ      *****
62  *      ***** END COMDECK YSTORE      *****
63  INTEGER AA

```

```

64      DATA TP/0./
65      C
66      C      PRINT TIMING FIGURE
67      C
68      C      CALL SECOND(TP)
69      C      WRITE(59,70) TP,T,NCYC
70      C
71      C      PRINT THE TIME AND CYCLE BEING DUMPED
72      C
73      C      DO 10 IPX=6,IFD,6
74      10 WRITE(IPX,90) T,NCYC
75      C      CALL OPENIT(8,1)
76      C
77      C      BACKSPACE OVER THE TRAILER RECORD WRITTEN BY THE LAST DUMP
78      C
79      C      BACKSPACE 8
80      C
81      C      WRITE OUT THE SCM COMMON
82      C
83      C      JNSC=LDCF(ZZ)-LDCF(AA)+1
84      C      CALL SCBUFF(AA,JNSC,8,1,1,IERROR)
85      C      IF(IERROR,EQ,0) GO TO 30
86      20 CALL UNCLE(4,5HYDUMP,19,19HSCBUFF OUTPUT ERROR)
87      C
88      C      I/O SUCCESSFULLY COMPLETED
89      C
90      30 CONTINUE
91      C
92      C      WRITE OUT THE ARRAYS FROM LCM
93      C
94      C      CALL LCBUFF(0,NLC,8,1,1,IERROR)
95      C
96      C      CHECK FOR ERRORS IN LCBUFF
97      C
98      C      IF(IERROR,EQ,0) GO TO 50
99      C
100     C      LCBUFF ERROR, KILL THE RUN,
101     C
102     40 CALL UNCLE(4,5HYDUMP,19,19HLCBUFF OUTPUT ERROR)
103     C
104     C      NO ERRORS, CONTINUE
105     C
106     50 CONTINUE
107     C
108     C      SEE IF THERE ARE ANY PARTICLES
109     C
110     C      IF(NPT,LE,0) GO TO 60
111     C
112     C      YES, WRITE OUT THE PARTICLE ARRAYS FROM LCM,
113     C
114     C      CALL LCBUFF(NLCP1,NPS,8,1,1,IERROR)
115     C      IF(IERROR,NE,0) GO TO 40
116     C
117     C      WRITE OUT THE ITAB ARRAY FROM SCM
118     C
119     C      CALL SCBUFF(ITAB,ITABP,8,1,1,IERROR)
120     C      IF(IERROR,NE,0) GO TO 20

```

```

121 C      SEE IF THERE IS ANY TIME-DEPENDENT PARTICLE DATA
122 C
123 C      IF(NILNG.LE.0) GO TO 60
124 C
125 C      YES, WRITE IT OUT FROM LCM,
126 C
127 C      CALL LCBUFF(NLCP1+NLCP2,2*NP1*NILNG,8,1,1,IERROR)
128 C      IF(IERROR.NE.0) GO TO 40
129 C
130 60 CONTINUE
131 C
132 C      TERMINATE THE DUMP WITH A SPECIAL TRAILER RECORD
133 C
134 C      IJUNK=666
135 C      CALL SCBUFF(IJUNK,1,8,1,1,IERROR)
136 C      IF(IERROR.NE.0) GO TO 20
137 C      CALL SECOND(TP)
138 C
139 C      PRINT TIMING FIGURE
140 C
141 C      WRITE(59,80) TP
142 C      RETURN
143 C
144 70 FORMAT(1H ,20HBEGIN YDUMP AT CP = ,F10.4,1H,,5X15HPROBLEM TIME = ,
145 1 1PE12.5,1H,,5X8HCYCLE = ,I5)
146 80 FORMAT(1H ,18HEND YDUMP AT CP = ,F10.4)
147 90 FORMAT(1H ,15HTAPE DUMP AT T=,1PE12.5,8H, CYCLE=,I5)
148 END

```

```

1      SUBROUTINE YEDIT
2
3      C      ROUTINE TO PRINT FULL LISTING OF YAQUI MESH QUANTITIES
4
5      C      ORIGINALLY WRITTEN BY A.A.AMSDEN,LASL T-3
6      C      MODIFIED AND DOCUMENTED BY J.L.NORTON,LASL T-3,1974
7
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 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),RDL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSROQ(1),CAPGAM(1),TUQ
18     4 (1),SIG(1),TUS(1),GRROR(1),GRRDZ(1),GRRQP(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      C COMMON/YSC1/AASC(9600)
27      C COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTCA,DTFAC,
28      C 1 DTO(10),DTOC(10),DTO2,DT08,DTP08,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
29      C 2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
30      C 3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
31      C 4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
32      C COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33      C 1 NCYC,NLC,NPS,NPT,NQ,NQI,NGIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34      C 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRDN,REZSIE,REZY0,RIBAR,RIBJB,
35      C 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN.
36      C COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37      C 1 ILNG,NILING,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TG,IST,
38      C 2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
39      C 3 XCNVLD,FIXRO,IXL,IXR,ISVW,JSVW,QMN,GMX,WMAX,JNM,T2,TLM,
40      C 4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
41      C 5 ,JDTV,IDTC,JDTA,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42      C 6 ,TMASSV,WMAXEF,RMINEF,TSTRD
43      C COMMON/YSC2/ZZ
44      C COMMON/YSC4/ITAB(ITABP)
45      C COMMON/YSC4/ITAB(1000)
46      C COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
47      * ***** END COMDECK YAQ8C *****
48      * ***** BEGIN COMDECK YAQEQQ *****
49      EQUivalence(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RC8Q,MP),(AASC
51      1 (8),E,ETIL,AREA,XR13K),
52      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56      6 21),GRROR),(AASC(22),GR0Z),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRR0P,TUQVEC,
58      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
60      1 AVYSV,X24K)
61      REAL M,MP,MPAR,MTIL
62      * ***** END COMDECK YAQEQQ *****
63      * ***** END COMDECK YSTORE *****
64      EQUIVALENCE(TEST,TEST)
65      DATA TP/0,/
66      DATA INDEF/177700000000007777777B/
67      CALL SECOND(TP)
68      WRITE(59,120) TP
69      C ALLOW 60 LINES PER PAGE OF DATA. SET THE COUNT TO FORCE A
70      C TOP-OF-PAGE HEADING FIRST THING
71      C
72      C LINESF#99
73      C ***** TEMPORARY PATCH FOR OUTPUTTING VORTICITY *****
74      C ***** IF(.NOT,TURB) CALL GETOMG *****
75      C ***** CALL START *****
76      C DO 100 J=1,JP2

```

```

80      DO 90 I=1,IP1
81      C
82      C     SEE IF A HEADER NEEDS TO BE PRINTED
83      C
84      C     IF(LINESF.LT.60) GO TO 30
85      C
86      C     YES, RESET THE LINE COUNT,
87      C
88      C     LINESF=0
89      C
90      C     PRINT THE HEADER
91      C
92      DO 20 IPX=IPD,IFD,6
93      WRITE(IPX,140) JNM,NAME,T,NCYC
94      IF(TURB) GO TO 10
95      WRITE(IPX,150)
96      GO TO 20
97      10 CONTINUE
98      WRITE(IPX,160)
99      20 CONTINUE
100     30 CONTINUE
101     IPJM=IJM+NQ
102     IPJ=IJ+NQ
103     D=0,
104     PRM=0,
105     PRV=0,
106     IF(TURB) GO TO 60
107     TEST=RM(IJM)
108     IF(TEST.EQ.INDEF) GO TO 40
109     IF(TEST.NE.0.) PRM=1./TEST
110     40 CONTINUE
111     TEST=RVOL(IJM)
112     IF(TEST.EQ.INDEF) GO TO 50
113     IF(TEST.NE.0.) PRV=1./TEST
114     50 CONTINUE
115     D=CONC(IJM)
116     C ***** TEMPORARY PATCH FOR OUTPUTTING VORTICITY *****
117     C
118     C ***** TEMPORARY PATCH FOR OUTPUTTING VORTICITY *****
119     PRM=CQ(IJM)
120     GO TO 70
121     60 PRV=TU9(IJM)
122     D=CONC(IJM)
123     PRM=TUQ(IJM)
124     70 CONTINUE
125     DO 80 IPX=IPD,IFD,6
126     WRITE(IPX,170) I,J,X(IJM),Y(IJM),U(IJM),V(IJM),SIE(IJM),RO(IJM),
127     1 PRV,D,PRM,P(IJM)
128     LINESF=LINESF+1
129     IJ=IPJ
130     90 IJM=IPJM
131     CALL LOOP
132     100 CONTINUE
133     IF(.NOT.FILM) GO TO 110
134     PIYB=PIYBO
135     IYB=IYBO
136     YCONV=YCNVLD

```

```

137      XCONVB=XCNVLD
138      FIXR=FIXRD
139      FIXL=FIXLD
140      IXR=IXRD
141      IXL=IXLD
142      CALL ADV(1)
143      !10 CONTINUE
144      CALL SECOND(TP)
145      WRITE(59,130) TP
146      RETURN
147      C
148      120 FORMAT(1H ,20HBEGIN YEDIT AT CP = ,F10.4)
149      130 FORMAT(1H ,18HEND YEDIT AT CP = ,F10.4)
150      140 FORMAT(1H1,A10,8A10,3X2HT*,1PE12.5,1X6HCYCLE*,I5)
151      150 FORMAT(1H ,2X1HI,3X1HJ,6X1HX,10X1HY,10X1HU,10X1HV,9X3HSIE,8X3HRHO,
152      1 8X3HVOL,9X1HC,9X3HVTC,8X1HP)
153      160 FORMAT(1H ,2X1HI,3X1HJ,6X1HX,10X1HY,10X1HU,10X1HV,9X3HSIE,8X3HRHO,
154      1 8X4HSCAL,9X1HC,9X1HQ,10X1HP)
155      170 FORMAT(1X,I3,1H,I3,10(1X,1PE10.3))
156      END

```

```

1      SUBROUTINE YEXIT(IABORT)
2      C
3      C      YAQUI ERROR RECOVERY ROUTINE
4      C
5      C      WRITTEN BY J.L.NORTON,LASL T-3,1975
6      C
7      CALL EXIT
8      END

```

```

1      SUBROUTINE YFLUX
2      C
3      C      ROUTINE TO FLUX MASS,MOMENTUM,AND ENERGY IF THE GRID VELOCITY IS
4      C      NOT EQUAL TO THE FLUID VELOCITY
5      C
6      C      ORIGINALLY WRITTEN BY A.A,AMSDEN AND HANS RUPPEL,LASL T-3
7      C      MODIFIED BY J.L.NORTON,LASL T-3,1975
8      C
9      *      ----- BEGIN COMDECK YSTORE      *****
10     *      ----- BEGIN COMDECK YAQDIM      *****
11     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
12     1 1),V(1),VG(1),RO(1),8IE(1),MP(1),RMP(1),RC80(1),E(1),ETIL(1),RVOL
13     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
14     3 ,VL(1),ROL(1),AVX8V(1),AVY8V(1),DL8ROI(1),DL8ROQ(1),CAPGAM(1),TUQ
15     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),

```

```

16      5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GR8V(1),GZSV(1),X13K(1),X24K(1),
17      6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKL8M(1),AREA(1)
18      *      ===== END COMDECK YAQDIM =====
19      *      ===== BEGIN COMDECK YAQSC =====
20      LOGICAL RESTRT,FILM,PAPER,TURB
21      REAL LAM,MU
22      C      COMMON/YSC1/AASC(NSCP1)
23      COMMON/YSC1/AASC(9600)
24      COMMON/YSC2/AA(1),ANC,A0,AOFAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
25      1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
26      2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GROVEL,GZ,GZP,I,IBAR,
27      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
28      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
29      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KX1,LAM,LPB,MU,NAME(8),
30      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
31      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
32      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
33      COMMON/YSC2/TUQI,TUSI,NCQ,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
34      1 ILNG,NILNG,TP3,TUPOT,TDQBAV,TK,TI,TUGENG,EP1,SAV1,QLEVEL,TQ,IST,
35      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
36      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
37      4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
38      5 ,JDTV,IDTC,JDTG,CIRC,TIS,POTE,UMOM,VHOM,TMAX,TGMX,ITM,JTM,ITG,JTG
39      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
40      COMMON/YSC2/ZZ
41      C      COMMON/YSC4/ITAB(ITABP)
42      COMMON/YSC4/ITAB(1000)
43      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
44      *      ===== END COMDECK YAQSC =====
45      *      ===== BEGIN COMDECK YAQEQ =====
46      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
47      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC
48      1 (8),E,ETIL,AREA,XR13K),
49      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
50      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
51      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
52      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
53      6 21),GROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
54      7 ),(AASC(25),DL8R00,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
55      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
56      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVX8V,X13K),(AASC(34),
57      1 AVYSV,X24K)
58      REAL M,MP,MPAR,MTIL
59      *      ===== END COMDECK YAQEQ =====
60      *      ===== END COMDECK YSTORE =====
61      *      ===== BEGIN COMDECK PARAM =====
62      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
63      1 NLCP3,NLCP4,IPLMSZ
64      *      ===== END COMDECK PARAM =====
65      *      ===== BEGIN COMDECK ASTORE =====
66      COMMON/ASTC/AT(100),FT(100)
67      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
68      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
69      1 ),(AT(9),YCO),(FT,CON)
70      *      ===== END COMDECK ASTORE =====
71      DTC=1,E30
72      CALL START

```

```

73      DO 50 J=2,JP!
74      DO 40 I=i,IBAR
75      IMJ=IJ+NQ
76      IPJ=IJ+NQ
77      IPJP=IPJ+NQ
78      X1=X(IPJ)
79      Y1=Y(IPJ)
80      R1=R(IPJ)
81      X2=X(IPJP)
82      Y2=Y(IPJP)
83      R2=R(IPJP)
84      X3=X(IJP)
85      Y3=Y(IJP)
86      R3=R(IJP)
87      X4=X(IJ)
88      Y4=Y(IJ)
89      R4=R(IJ)
90      UL1=UL(IPJ)
91      VL1=VL(IPJ)
92      UL2=UL(IPJP)
93      VL2=VL(IPJP)
94      UL3=UL(IJP)
95      VL3=VL(IJP)
96      UL4=UL(IJ)
97      VL4=VL(IJ)
98      UD1=UG(IPJ)=0.5*(UL1+U(IPJ))
99      VD1=VG(IPJ)=0.5*(VL1+V(IPJ))
100     UD2=UG(IPJP)=0.5*(UL2+U(IPJP))
101     VD2=VG(IPJP)=0.5*(VL2+V(IPJP))
102     UD3=UG(IJP)=0.5*(UL3+U(IJP))
103     VD3=VG(IJP)=0.5*(VL3+V(IJP))
104     UD4=UG(IJ)=0.5*(UL4+U(IJ))
105     VD4=VG(IJ)=0.5*(VL4+V(IJ))
106     X12=X1=X2
107     X23=X2=X3
108     X34=X3=X4
109     X41=X4=X1
110     Y21=Y2=Y1
111     Y32=Y3=Y2
112     Y43=Y4=Y3
113     Y14=Y1=Y4
114     Y31=Y3=Y1
115     R12=R1+R2
116     R23=R2+R3
117     R34=R3+R4
118     R41=R4+R1
119     U12=UL1+UL2
120     U23=UL2+UL3
121     U34=UL3+UL4
122     U41=UL4+UL1
123     V12=VL1+VL2
124     V23=VL2+VL3
125     V34=VL3+VL4
126     V41=VL4+VL1
127     D=.25*RVOL(IJ)*(R12*(U12*Y21+V12*X12)+R23*(U23*Y32+V23*X23)+R34*(
128     U34*Y43+V34*X34)+R41*(U41*Y14+V41*X41))
129     VOLR=VOLT=VOLC=1./RVOL(IJ)

```

```

130      IF(I,NE,IBAR) VOLR#1./RVOL(IPJ)
131      IF(J,NE,JP1) VOLT#1./RVOL(IJP)
132      IF(I,EQ,1) GO TO 60
133      FL#=FR
134      AL#=AR
135      10 IF(J,EQ,2) GO TO 70
136      FB#=FT(I)
137      AB#=AT(I)
138      20 FR=DT08*R12*((UD1+UD2)*Y21+(VD1+VD2)*X12)
139      AR#A0M*SIGN(1.,FR)+B0*4.*FR/(VOLR+VOLC)
140      FT(I)=DT08*R23*((UD2+UD3)*Y32+(VD2+VD3)*X23)
141      AT(I)=A0M*SIGN(1.,FT(I))+B0*4.*FT(I)/(VOLT+VOLC)
142      XX#AMAX1(ABS(FB),ABS(FR),ABS(FT(I)),ABS(FL))
143      DTC#AMIN1(DTC,DTP03*A0PAC/(XX*RVOL(IJ)+DTP03*ABS(D)+EM10))
144      IF(DTCSAV,NE,DTC) IDTC#I
145      IF(DTCSAV,NE,DTC) JDTCA#J
146      DTCSAV=DTC
147      MP(IJ)=MTIL(IJ)*VOLC+FR*((1,-AR)*ROL(IJ)+(1,+AR)*ROL(IPJ))+FT(I)*(
148      1,(1,-AT(I))*ROL(IJ)+(1,+AT(I))*ROL(IJP))+FL*((1,-AL)*ROL(IJ)+(1,
149      2,AL)*ROL(IMJ))+FB*((1,-AB)*ROL(IJ)+(1,+AB)*ROL(IJM))
150      ROE#RO(IJ)*ETIL(IJ)
151      8IE(IJ)=1./MP(IJ)*(ROE+VOLC+FR*((1,-AR)*ROE+(1,+AR)*RO(IPJ)*ETIL(
152      1,IPJ))+FT(I)*((1,-AT(I))*ROE+(1,+AT(I))*RO(IJP)*ETIL(IJP))+FL*((1,
153      2,AL)*ROE+(1,+AL)*RO(IMJ)*ETIL(IMJ))+FB*((1,-AB)*ROE+(1,+AB)*RO(
154      3,IJM)*ETIL(IJM)))
155      IF(,NOT,TURB) GO TO 30
156      ROQ#RO(IJ)*TUQ(IJ)
157      TUQVEC(IJ)=1./MP(IJ)*(ROQ+VOLC+FR*((1,-AR)*ROQ+(1,+AR)*RO(IPJ)*TUQ
158      1,(IPJ))+FT(I)*((1,-AT(I))*ROQ+(1,+AT(I))*RO(IJP)*TUQ(IJP))+FL*((1,
159      2,AL)*ROQ+(1,+AL)*RO(IMJ)*TUQ(IMJ))+FB*((1,-AB)*ROQ+(1,+AB)*RO(IJM
160      3)*TUQ(IJM)))
161      30 CONTINUE
162      ROQ#CONC(IJ)
163      CTEMP(IJ)=VOLC+ROQ+FR*((1,-AR)*ROQ+(1,+AR)*CONC(IPJ))+FT(I)*((1,-
164      1,AT(I))*ROQ+(1,+AT(I))*CONC(IJP))+FL*((1,-AL)*ROQ+(1,+AL)*CONC(IMJ
165      2))+FB*((1,-AB)*ROQ+(1,+AB)*CONC(IJM))
166      CTEMP(IJ)=CTEMP(IJ)*RVOL(IJ)
167      RVOL(IJ)=8./((R1+R2+R3+R4)*((X1-X3)*(Y2-Y4)-(Y1-Y3)*(X2-X4)))
168      IJ#IPJ
169      IJP#IPJP
170      40 IJMA#IJM+NQ
171      CALL LOOP
172      50 CONTINUE
173      CALL DONE
174      GO TO 80
175      60 FL#DT08*R34*((UD3+UD4)*Y43+(VD3+VD4)*X34)
176      AL#A0M*SIGN(1.,FL)+B0*2.*FL*RVOL(IJ)
177      GO TO 10
178      70 FB#DT08*R41*((UD4+UD1)*Y14+(VD4+VD1)*X41)
179      AB#A0M*SIGN(1.,FB)+B0*2.*FB*RVOL(IJ)
180      GO TO 20
181      80 CALL START
182      DO 110 J#2,JP1
183      DO 100 I#1,IBAR
184      RO(IJ)=MP(IJ)*RVOL(IJ)
185      CONC(IJ)=CTEMP(IJ)
186      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      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(IJJ))
264 VL24=0.5*(VL2+VL4+V(IPJP)+V(IJJ))
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=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 H=1,
288 W=((Y3=YC)*UGCUC+(X3=X3)*VGCVC)/((Y3=Y2)*UGCUC+(X2=X3)*VGCVC)
289 GO TO 240
290 H=1,
291 W=((YC=Y1)*UGCUC+(X1=XC)*VGCVC)/((Y2=Y1)*UGCUC+(X1=X2)*VGCVC)
292 GO TO 240
293 H=0,
294 W=((YC=Y4)*UGCUC+(X4=XC)*VGCVC)/((Y3=Y4)*UGCUC+(X4=X3)*VGCVC)
295 GO TO 240
296 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

```

```

301      VB=W*H*VL2+OMW*H*VL3+W*OMH*VL1+OMW*OMH*VL4
302      ALTE=A0+B0*(ABS(F13)+ABS(F24))*RVOL(IJ)/ROL(IJ)
303      XX=(1,-ALTE)*UC+ALTE*UB
304      UP(IPJ)=UP(IPJ)-FM1*XX
305      UP(IJP)=UP(IJP)+FM3*XX
306      UP(IPJP)=UP(IPJP)-FM2*XX
307      UP(IJ)=UP(IJ)+FM4*XX
308      XX=(1,-ALTE)*VC+ALTE*VB
309      VP(IPJ)=VP(IPJ)-FM1*XX
310      VP(IJP)=VP(IJP)+FM3*XX
311      VP(IPJP)=VP(IPJP)-FM2*XX
312      VP(IJ)=VP(IJ)+FM4*XX
313      IJ=IPJ
314      250 IJP=IPJP
315      CALL LOOP
316      260 CONTINUE
317      CALL DONE
318      C
319      C      GO SET THE BOUNDARY CONDITIONS
320      C
321      CALL BC(3)
322      RETURN
323      END

```

```

1      SUBROUTINE YINIT
2      C
3      C      ROUTINE TO INITIALIZE THE CODE
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,NLCR1,NLCR2,
9      1 NLCR3,NLCR4,IFLMSZ
10     *      **** END COMDECK PARAM      ****
11     *      **** BEGIN COMDECK YSTORE      ****
12     *      **** BEGIN COMDECK YAQDIM      ****
13     DIMENSION X(1),XPAR(1),R(1),YPAR(1),Y(1),MPAR(1),U(1),UG(1),DELSM(
14     1 1),V(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC8Q(1),E(1),ETIL(1),RVOL
15     2 (1),M(1),RM(1),VP(1),P(1),PL(1),UP(1),UTIL(1),UL(1),CQ(1),VTIL(1)
16     3 ,VL(1),ROL(1),AVXSV(1),AVYSV(1),DLSROI(1),DLSR0Q(1),CAPGAM(1),TUQ
17     4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
18     5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
19     6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
20     *      **** END COMDECK YAQDIM      ****
21     *      **** BEGIN COMDECK YAQSC      ****
22     LOGICAL RESTRT,FILM,PAPER,TURB
23     REAL LAM,MU
24     C      COMMON/YSC1/AASC(NSCP1)
25     COMMON/YSC1/AASC(9600)
26     COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
27     1 DT0(10),DT0C(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,

```

```

28      2 FIPYB,FIPYT,FIYL,FIYR,FIYB,FIYT,FREZXR,GR,GRDVEL,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,JUNFO2,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,REZBIE,REZYB,RIBAR,RIBJB,
34      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
35      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,NEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
36      1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUGENG,EP1,SAV1,OLEVEL,TQ,IST,
37      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
38      3 XCNVLD,FIXRO,FXL0,IXRO,IXL0,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
39      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAT,DTCSAV,IDTV
40      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMM,ITM,JTM,ITG,JTG
41      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
42      COMMON/YSC2/ZZ
43      C COMMON/YSC4/ITAB(1TABP)
44      COMMON/YSC4/ITAB(1000)
45      COMMON/YSC5/RESTART,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),R0),(AASC(7),DELSM,RCSQ,MP),(AASC
50      1 (8),E,ETIL,AREA,XR13K),
51      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,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),GR0Z),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
56      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRR0P,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      COMMON/FTABC/FTAB(2)
64      DIMENSION START(3,5)
65      DATA (START(II),II=1,15)/15*0/
66      DATA II, IDUM/0,0/
67      DATA IEFLAG/0/
68      DATA RESTART,PAPER,FILM,WRAPUP/,TRUE,,FALSE,,TRUE,,20,/
69
70      C GET THE CP TIME AT JOB START
71      C
72      C CALL SECOND(T2)
73      C
74      C MAKE SURE ENOUGH LCM IS AVAILABLE
75      C
76      INEED=NLCP1+NLCP2+NLCP3+IFLMSZ+NLCP4+1
77      CALL GETLCM(ILSIZE)
78      IF(INEED.GT.ILSIZE) CALL UNCLE(4,5HYINIT,27,
79      1 27HNOT ENOUGH LCM IS AVAILABLE)
80
81      C NO IS THE NO. OF DISTINCT LCM ARRAYS ALLOCATED. ARRAYS CAN BE
82      C ADDED OR DELETED AS LONG AS NO IS CHANGED CONSISTENTLY.
83      C
84      NO#34

```

```

85      C
86      C      READ THE JOB PARAMETERS. FIRST SET UP FOR NAMELIST INPUT.
87      C
88      C      ASSIGN 30 TO IERRT
89      C      CALL TABDEF(START,5HSTART,5,IERRT)
90      C      CALL TABSET(START,6HRESTART,RESTART,IEFLAG,0,0,0,0)
91      C      CALL TABSET(START,4HFILM,FILM,IEFLAG,0,0,0,0)
92      C      CALL TABSET(START,5HPAPER,PAPER,IEFLAG,0,0,0,0)
93      C      CALL TABSET(START,6HWRAPUP,WRAPUP,IEFLAG,0,0,0,0)
94      C
95      C      DO THE ACTUAL READ
96      C
97      C          RESTART = IF .TRUE., THE CODE WILL BE RESTARTED FROM A
98      C              DUMP TAPE (DEFAULT)
99      C              = IF .FALSE., THE PROBLEM WILL BE GENERATED FROM INPUT
100     C              DATA TO FOLLOW
101     C          PAPER = IF .TRUE., OUTPUT WILL OCCUR ON PAPER (DEFAULT=.FALSE.)
102     C          FILM = IF .TRUE., OUTPUT WILL OCCUR ON FILM (DEFAULT=.TRUE.)
103     C          (IF PAPER AND FILM ARE BOTH FALSE, FILM IS SET TO .TRUE.)
104     C          WRAPUP = NO. OF CP SECONDS TO ALLOW FOR PROBLEM TERMINATION
105     C              AFTER THE LAST CYCLE (DEFAULT=20.)
106     C
107     C      CALL NAMLST(START,5,IEFLAG)
108     C
109     C      CHECK FOR ERRORS
110     C
111     C      IF(IEFLAG.NE.0) CALL UNCLE(4,5HYINIT,23,23ERROR IN START NAMELIST
112     C      )
113     C
114     C      NO ERRORS. SET THE OUTPUT DO LOOP INDICES.
115     C
116     C      IPD#6
117     C      IFD#12
118     C      IF(.NOT.PAPER.AND,.NOT.FILM) FILM=.TRUE.
119     C      IF(.NOT.PAPER) IPD#12
120     C      IF(.NOT.FILM) IFD#6
121     C
122     C      SET UP THE CONSTANT 1/3
123     C
124     C      THIRD#1./3.
125     C
126     C      SET UP THE CONSTANT 2/3
127     C
128     C      TWTHRD#2.*THIRD
129     C
130     C      GET THE JOB ID
131     C
132     C      CALL GETJOB(JNM)
133     C
134     C      GET THE JOB TIME LIMIT
135     C
136     C      CALL GETJTL(TLIM)
137     C
138     C      GIVE A LITTLE TIME FOR WRAPUP
139     C
140     C      TLIM=TLIM-WRAPUP
141     C

```

```

142      IF FILM IS TRUE,SET UP OUTPUT FROM NAMELIST ROUTINES TO GO TO
143      FILM AND PAPER BOTH
144
145      IF(FILM) CALL NAMPRT(2,FTAB)
146
147      SET LCM TO INDEFINITES
148
149      I#0
150      ILAST=NLCP1+NLCP2+NLCP3+1
151      10 CONTINUE
152      CALL ECWR(177700000000000777777B,I,1,1DUM)
153      IF(I.EQ.ILAST) GO TO 20
154      I#I+1
155      GO TO 10
156      20 CONTINUE
157      RETURN
158      30 CONTINUE
159      CALL UNCLE(4,5HYINIT,35,35H$TART NAMELIST INITIALIZATION ERROR)
160      END

```

```

1 SUBROUTINE YINPUT
2
3 C ROUTINE TO READ YAQUI INPUT VARIABLES
4
5 C WRITTEN BY J.L.NORTON, LASL T-3, 1975
6
7 C ***** BEGIN COMDECK YSTORE *****
8 C ***** 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),RCRG(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),DLSROI(1),DLSRQ(1),CAPGAM(1),TUQ
13 4 (1),SIG(1),TUS(1),GRROR(1),GRROZ(1),GRRDP(1),TUQVEC(1),MTIL(1),
14 5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZSV(1),X13K(1),X24K(1),
15 6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKLSM(1),AREA(1)
16 C ***** END COMDECK YAQDIM *****
17 C ***** BEGIN COMDECK YAQSC *****
18 LOGICAL RESTRT,FILM,PAPER,TURB
19 REAL LAM,MU
20 C COMMON/YSC1/AASC(NSCP1)
21 C COMMON/YSC1/AASC(9600)
22 C COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTA,DTFAC,
23 1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
24 2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
25 3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
26 4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
27 C COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
28 1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
29 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZYB,RIBAR,RIBJB,
30 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
31 C COMMON/YSC2/TUQI,TUSI,NQD,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
```

```

32      1 ILNG,NILNG,TP3,TUPOT,TDQ8AV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
33      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
34      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,HMAX,JNM,T2,TLIM,
35      4 ROMFXR,ROMFYT,ROMFYB,JUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
36      5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
37      6 ,THASSV,WMAXEF,RMINEF,TSTRTD
38      COMMON/YSC2/2Z
39      C COMMON/YSC4/ITAB(ITABP)
40      COMMON/YSC4/ITAB(1000)
41      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
42      * ===== END COMDECK YAGEQ =====
43      * ===== BEGIN COMDECK YAGEQ =====
44      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),
45      1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RC8Q,MP),(AASC
46      1 (8),E,ETIL,AREA,XR13K),
47      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
48      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
49      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
50      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
51      6 21),GRROR),(AASC(22),GRR0Z),(AASC(23),DLSROI,Y13K),(AASC(24),GZ8V
52      7 ),(AASC(25),DLSR0Q,VG),(AASC(26),GRSV),(AASC(27),GRR0P,TUQVEC,
53      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),
54      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
55      1 AVYSV,X24K)
56      REAL M,MP,MPAR,MTIL
57      * ===== END COMDECK YAGEQ =====
58      * ===== END COMDECK YSTORE =====
59      DIMENSION CARDN(3,49)
60      DATA (CARDN(II),II=1,147)/147*0/
61      DATA IEFLAG/0/
62      C
63      C SET UP THE NAMELIST INPUT TABLE
64      C
65      ASSIGN 40 TO IERRT
66      CALL TABDEF(CARDN,SHCARDN,49,IERRT)
67      CALL TABSET(CARDN,2HMU,MU,IEFLAG,0,0,0,0)
68      CALL TABSET(CARDN,3HLAM,LAM,IEFLAG,0,0,0,0)
69      CALL TABSET(CARDN,2HOM,OM,IEFLAG,0,0,0,0)
70      CALL TABSET(CARDN,3HEPS,EPS,IEFLAG,0,0,0,0)
71      CALL TABSET(CARDN,2HGR,GR,IEFLAG,0,0,0,0)
72      CALL TABSET(CARDN,2HGZ,GZ,IEFLAG,0,0,0,0)
73      CALL TABSET(CARDN,6HFREZXR,FREZXR,IEFLAG,0,0,0,0)
74      CALL TABSET(CARDN,2HYB,YB,IEFLAG,0,0,0,0)
75      CALL TABSET(CARDN,5HREZY0,REZY0,IEFLAG,0,0,0,0)
76      CALL TABSET(CARDN,6HREZRON,REZRON,IEFLAG,0,0,0,0)
77      CALL TSET(CARDN,6HREZSIE,REZSIE,IEFLAG,0,0,0,0)
78      CALL TABSET(CARDN,3HGZP,GZP,IEFLAG,0,0,0,0)
79      CALL TABSET(CARDN,1HT,T,IEFLAG,0,0,0,0)
80      CALL TABSET(CARDN,2HDT,DT,IEFLAG,0,0,0,0)
81      CALL TABSET(CARDN,5HNCLST,NCLST,IEFLAG,0,0,0,0)
82      CALL TSET(CARDN,5HTWFIN,TWFIN,IEFLAG,0,0,0,0)
83      CALL TABSET(CARDN,3HNCQ,NCQ,IEFLAG,0,0,0,0)
84      CALL TABSET(CARDN,6HQLEVEL,QLEVEL,IEFLAG,0,0,0,0)
85      CALL TABSET(CARDN,4HTUQI,TUQI,IEFLAG,0,0,0,0)
86      CALL TABSET(CARDN,4HTUSI,TUSI,IEFLAG,0,0,0,0)
87      CALL TABSET(CARDN,2HTQ,TQ,IEFLAG,0,0,0,0)
88      CALL TABSET(CARDN,3HIST,IST,IEFLAG,0,0,0,0)

```

```

89      CALL TABSET(CARDN,4HIBAR,IBAR,IEFLAG,0,0,0,0)
90      CALL TABSET(CARDN,4HJBAR,JBAR,IEFLAG,0,0,0,0)
91      CALL TABSET(CARDN,4HIUNF,IUNF,IEFLAG,0,0,0,0)
92      CALL TABSET(CARDN,4HJUNF,JUNF,IEFLAG,0,0,0,0)
93      CALL TABSET(CARDN,4HJCEN,JCEN,IEFLAG,0,0,0,0)
94      CALL TABSET(CARDN,2HDR,DR,IEFLAG,0,0,0,0)
95      CALL TABSET(CARDN,2HDZ,DZ,IEFLAG,0,0,0,0)
96      CALL TABSET(CARDN,3HCYL,CYL,IEFLAG,0,0,0,0)
97      CALL TABSET(CARDN,6HGRDVEL,GRDVEL,IEFLAG,0,0,0,0)
98      CALL TABSET(CARDN,2HA0,A0,IEFLAG,0,0,0,0)
99      CALL TABSET(CARDN,3HA0M,A0M,IEFLAG,0,0,0,0)
100     CALL TABSET(CARDN,2HB0,B0,IEFLAG,0,0,0,0)
101     CALL TABSET(CARDN,3HKXI,KXI,IEFLAG,0,0,0,0)
102     CALL TABSET(CARDN,3HANC,ANC,IEFLAG,0,0,0,0)
103     CALL TABSET(CARDN,5HA0FAC,A0FAC,IEFLAG,0,0,0,0)
104     CALL TABSET(CARDN,3HDTO,DTO,IEFLAG,1,0,0,0)
105     CALL TABSET(CARDN,4HDTOC,DTOC,IEFLAG,1,0,0,0)
106     CALL TABSET(CARDN,6HFREZYT,FREZYT,IEFLAG,0,0,0,0)
107     CALL TABSET(CARDN,6HFREZYB,FREZYB,IEFLAG,0,0,0,0)
108     CALL TABSET(CARDN,5HZORIG,ZORIG,IEFLAG,0,0,0,0)
109     CALL TABSET(CARDN,5HJDUMP,JDUMP,IEFLAG,0,0,0,0)
110     CALL TABSET(CARDN,4HNAME,NAME,IEFLAG,1,0,0,0)
111     CALL TABSET(CARDN,6HWMAXEF,WMAXEF,IEFLAG,0,0,0,0)
112     CALL TABSET(CARDN,6HRMINEF,RMINEF,IEFLAG,0,0,0,0)
113     CALL TABSET(CARDN,6HTSTRTD,TSTRTD,IEFLAG,0,0,0,0)
114     CALL TABSET(CARDN,4HIEOF,IEOF,IEFLAG,0,0,0,0)

115     C
116     C      READ THE INPUT VARIABLES
117     C
118 10    CONTINUE
119     IEOF=0
120     CALL NAMLST(CARDN,5,IEFLAG)
121     C
122     C      CHECK FOR INPUT ERRORS
123     C
124     IF(IEFLAG.NE.0) CALL UNCLE(4,6HYINPUT,26,
125     1,26HCARDN NAMELIST INPUT ERROR)
126     C
127     NO ERRORS. CHECK FOR EOF.
128     C
129     IF(IEOF.NE.0) GO TO 20
130     C
131     NO. CONTINUE READING.
132     C
133     GO TO 10
134 20    CONTINUE
135     C
136     C      CHECK THE VALUES OF THE INPUT PARAMETERS
137     C
138     IF(MU.LT.0.) CALL UNCLE(4,6HYINPUT,8,8HMU,LT.0.)
139     IF(LAM.LT.0.) CALL UNCLE(4,6HYINPUT,9,9HLAM,LT.0.)
140     IF(OM.LT.1..OR.OM.GT.2.) CALL UNCLE(4,6HYINPUT,10,10HILLEGAL OM)
141     IF(EPS.LE.0.) CALL UNCLE(4,6HYINPUT,9,9HEPS,LE.0.)
142     IF(FREZXR.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZXR,LT.1.)
143     IF(FREZYT.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZYT,LT.1.)
144     IF(FREZYB.LT.1.) CALL UNCLE(4,6HYINPUT,12,12HFREZYB,LT.1.)
145     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(DT,LE,0.) CALL UNCLE(4,6HYINPUT,8,8HDT,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.DTOC(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,NF,0.,AND.GRDVEL,NE,1.,AND.GRDVEL,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,35,35HCARDN 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 *
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 ),VG(1),VG(1),RO(1),SIE(1),MP(1),RMP(1),RC3Q(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),DLSROQ(1),CAPGAM(1),TUQ
14 4 (1),SIG(1),TUS(1),GRRDR(1),GRROZ(1),GRROP(1),TUQVEC(1),MTIL(1),
15 5 CONC(1),CTEMP(1),ANCU(1),ANCV(1),GRSV(1),GZ8V(1),X13K(1),X24K(1),
16 6 Y13K(1),Y24K(1),XR13K(1),XR24K(1),DKL8M(1),AREA(1)
17 * ===== END COMDECK YAQDIM =====
18 * ===== BEGIN COMDECK YAQSC =====
19 LOGICAL RESTRT,FILM,PAPER,TURB
20 REAL LAM,MU
20 C COMMON/Y8C1/AASC(NSCP1)

```

```

21 COMMON/YSC1/AASC(9600)
22 COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTA,DTFAC,
23 1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
24 2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
25 3 IOTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
26 4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
27 COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
28 1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
29 2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
30 3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
31 COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,NEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
32 1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
33 2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
34 3 XCNVLD,PIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
35 4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVS,DTCSAV,DTCSAV,DTVS
36 5 ,JDTV,DTDC,DTDC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
37 6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
38 COMMON/YSC2/ZZ
39 C COMMON/YSC4/ITAB(ITABP)
40 COMMON/YSC4/ITAB(1000)
41 COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
42 * **** END COMDECK YAQSC ****
43 * **** BEGIN COMDECK YAQEQ ****
44 EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
45 1 AASC(4),U),(AASC(5),V),(AASC(6),R0),(AASC(7),DELSM,RCSQ,MP),(AASC
46 1 (8),E,ETIL,AREA,XR13K),
47 2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
48 3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
49 4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
50 5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
51 6 21),GRROR),(AASC(22),GRR0Z),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
52 7 ),(AASC(25),DLSR00,VG),(AASC(26),GRSV),(AASC(27),GRR0P,TUQVEC,
53 8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
54 9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
55 1 AVYSV,X24K)
56 REAL M,MP,MPAR,MTIL
57 * **** END COMDECK YAQEQ ****
58 * **** END COMDECK YSTORE ****
59 * **** BEGIN COMDECK PARAM ****
60 COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
61 1 NLCP3,NLCP4,IFLMSZ
62 * **** END COMDECK PARAM ****
63 * **** BEGIN COMDECK ASTORE ****
64 COMMON/ASTC/AT(100),FT(100)
65 DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
66 EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
67 1 ),(AT(9),YCO),(FT,CON)
68 * **** END COMDECK ASTORE ****
69 REAL LAM0,MU02
70 C
71 C APPLY THE NODE COUPLER TO ALL VERTICES
72 C
73 CALL START
74 Y1=ANC*RDT
75 DO 100 J=2,JP2
76 DO 90 I=1,IP1
77 IMJ=IJ=NQ

```

```

78      IPJ=IJ+NQ
79
80      C   XX IS ZERO IF I IS 1 OR IP1, 1 OTHERWISE
81      C   YY IS ZERO IF J IS 2 OR JP2, 1 OTHERWISE
82      C
83      C   XX=1.
84      C   YY=1.
85      C
86      C   U1,V1 ARE VELOCITIES AT VERTEX (I=1,J) UNLESS I=1 IN WHICH CASE
87      C       VERTEX (I,J) IS USED
88      C
89      C   IF(I,EQ,1) GO TO 10
90      C   U1=U(IPJ)
91      C   V1=V(IPJ)
92      C   GO TO 20
93      10  XX=0.0
94      C   U1=U(IJ)
95      C   V1=V(IJ)
96      C
97      C   U2,V2 ARE VELOCITIES AT VERTEX (I+1,J) UNLESS I=IP1 IN WHICH CASE
98      C       VERTEX (I,J) IS USED
99      C
100     20  IF(I,EQ,IP1) GO TO 30
101     U2=U(IPJ)
102     V2=V(IPJ)
103     GO TO 40
104     30  U2=U(IJ)
105     V2=V(IJ)
106     XX=0.0
107     C
108     C   U3,V3 ARE VELOCITIES AT VERTEX (I,J=1) UNLESS J=2 IN WHICH CASE
109     C       VERTEX (I,J) IS USED
110     C
111     40  IF(J,EQ,2) GO TO 50
112     U3=U(IJM)
113     V3=V(IJM)
114     GO TO 60
115     50  U3=U(IJ)
116     V3=V(IJ)
117     YY=0.0
118     C
119     C   U4,V4 ARE VELOCITIES AT VERTEX (I,J+1) UNLESS J=JP2 IN WHICH CASE
120     C       VERTEX (I,J) IS USED
121     C
122     60  IF(J,EQ,JP2) GO TO 70
123     U4=U(IJP)
124     V4=V(IJP)
125     GO TO 80
126     70  YY=0.0
127     U4=U(IJ)
128     V4=V(IJ)
129     C
130     C   U5,V5 ARE VELOCITIES AT VERTEX (I,J)
131     C
132     80  US=U(IJ)
133     VS=V(IJ)
134     C

```

```

135 C THE NODE COUPLER IS APPLIED TO THE U (V) VELOCITY IF FLAGU (V)
136 C IS 1.
137 C
138 C FLAGU=0,
139 C 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 C IF(U5,EQ,AMAX1(U1,U2,U5),OR,U5,EQ,AMIN1(U1,U2,U5)) FLAGU=1,0
148 C IF(V5,EQ,AMAX1(V1,V2,V5),OR,V5,EQ,AMIN1(V1,V2,V5)) FLAGV=1,0
149 C IF(U5,EQ,AMAX1(U3,U4,U5),OR,U5,EQ,AMIN1(U3,U4,U5)) FLAGU=1,0
150 C 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 C UAV=0,25*(U1+U2+U3+U4)=U5
157 C 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 C ANCX=Y1*FLAGU*UAV
163 C ANCY=Y1*FLAGV*VAV
164 C AX=GR+ANCX
165 C 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 C UTIL(IJ)=(U(IJ)+DT*AX)*XX
171 C 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 C ANCU(IJ)=ANCX*XX*DT
176 C ANCV(IJ)=ANCY*YY*DT
177 C IJ=IPJ
178 C IJP=IJP+NQ
179 C 90 IJM=IJM+NQ
180 C CALL LOOP
181 C 100 CONTINUE
182 C 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 C CALL START
188 DO 160 J=2,JPI
189 DO 150 I=1,IBAR
190 ROL(IJ)*0,
191 IPJ=IJ+NQ

```

```

192      IPJP=IJP+NQ
193      C
194      C   VERTEX (I+1,J) IS LABELLED 1, (I+1,J+1) IS 2, (I,J+1) IS 3,
195      C   AND (I,J) IS 4
196      C
197      X1=X(IPJ)
198      Y1=Y(IPJ)
199      R1=R(IPJ)
200      U1=U(IPJ)
201      V1=V(IPJ)
202      X2=X(IPJP)
203      Y2=Y(IPJP)
204      R2=R(IPJP)
205      U2=U(IPJP)
206      V2=V(IPJP)
207      X3=X(IJP)
208      Y3=Y(IJP)
209      R3=R(IJP)
210      U3=U(IJP)
211      V3=V(IJP)
212      X4=X(IJ)
213      Y4=Y(IJ)
214      R4=R(IJ)
215      U4=U(IJ)
216      V4=V(IJ)
217      C
218      C   X(Y)NM = X(Y)N=X(Y)M
219      C
220      X24=X2=X4
221      Y24=Y2=Y4
222      X31=X3=X1
223      Y31=Y3=Y1
224      C
225      C   UOR IS THE AVERAGE U VELOCITY DIVIDED BY THE AVERAGE X-POSITION
226      C   OF THE VERTICES, FOR CYLINDRICAL GEOMETRY. FOR SLAB GEOMETRY
227      C   UOR IS ZERO.
228      C
229      UOR=(U1+U2+U3+U4)/(R1+R2+R3+R4)*CYL
230      C
231      C   HRMN IS .5*(RM+RN)
232      C
233      HR13=.5*(R1+R3)
234      HR24=.5*(R2+R4)
235      C
236      C   DTO2MN = DTO2/(MASS OF VERTEX N)
237      C
238      DTO2M1=DTO2*RM(IPJ)
239      DTO2M2=DTO2*RM(IPJP)
240      DTO2M3=DTO2*RM(IJP)
241      DTO2M4=DTO2*RM(IJ)
242      C
243      C   XY IS THE CROSS PRODUCT OF THE DIAGONALS OF THE CELL
244      C
245      XY=X24*Y31-X31*Y24
246      C
247      C   CAREA IS THE CELL AREA WHICH IS JUST HALF OF THE CROSS PRODUCT
248      C   OF THE DIAGONALS

```

```

249 C
250 C      CAREA=.5*XY
251 C
252 C      SAVE THE AREA FOR USE IN THE TURBULENCE CALCULATION
253 C      IN TRBCOR
254 C
255 C      AREA(IJ)=CAREA
256 C
257 C      RXY IS THE RECIPROCAL OF XY (HALF THE RECIPROCAL OF THE CELL AREA)
258 C
259 C      RXY=1./XY
260 C
261 C      U(V)IJ=U(V)I=U(V)J
262 C
263 C      U24=U2=U4
264 C      U31=U3=U1
265 C      V24=V2=V4
266 C      V31=V3=V1
267 C
268 C      DU(DR) 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 C      DUDR=RXY*(U24*Y31-U31*Y24)
273 C      DUDY=RXY*(X24*U31-X31*U24)
274 C      DVDR=RXY*(V24*Y31-V31*Y24)
275 C      DVDY=RXY*(X24*V31-X31*V24)
276 C
277 C      COMPUTE DIV(VELOCITY) = VOLUME DILATATION
278 C
279 C      D=(DUDR+DVDR)*(1.+UOR*DT)+UOR
280 C
281 C      SAVE THE VELOCITY DIVERGENCE
282 C
283 C      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 C      XXA=2./(Y24**2+Y31**2)*CAREA**2
295 C      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 C      DKLSM(IJ)=2.*DT*(XXA+YYA)/(XXA*YYA)
301 C
302 C      XX AND YY ARE THE ACTUAL AVX AND AVY
303 C
304 C      XX=SQRT(XXA)
305 C      YY=SQRT(YYA)

```

```

306 C      CALCULATE THE EFFECTIVE VISCOSITY COEFFICIENTS.
307 C
308 C      LAMBDA(MU) EFFECTIVE = LAM(MU) INPUT TIMES AK WHERE THE LATTER
309 C          IS DEPENDENT ON THE VALUE OF KXI. IF KXI IS 0 OR 1,
310 C          AK=1. OR RHO(IJ),RESPECTIVELY, IF KXI IS -1, AK IS
311 C          DETERMINED FROM NUMERICAL STABILITY REQUIREMENTS,
312 C
313 C
314 C      KP=KXI+2
315 C      GO TO (130,110,120),KP
316 C      110 CONTINUE
317 C          AK=1.
318 C          GO TO 140
319 C      120 CONTINUE
320 C          AK=R0(IJ)
321 C          GO TO 140
322 C      130 CONTINUE
323 C          VELIJ=U4*U4+V4*V4
324 C          VELMX=0.7*AMAX1(ABS(U4*XX),ABS(V4*YY))
325 C          AK=R0(IJ)*COLAMU*(DT02*VELIJ+VELMX)+EM10
326 C      140 CONTINUE
327 C
328 C      DETERMINE LAMBDA*DIVERGENCE(VELOCITY)=THIS IS LAMD.
329 C          NOTE THAT LAMD IS ALWAYS .LE.,0.
330 C
331 C          ROSIG=0.
332 C          IF(TURB) ROSIG=R0(IJ)*SIG(IJ)
333 C          LAMD=A MIN1(D,0.)*AK*LAM-TWTHRD*ROSIG*D
334 C
335 C      MU02 IS HALF MU EFFECTIVE
336 C
337 C      MU02=0.5*(AK*MU+ROSIG)
338 C      IF(TURB) CAPGAM(IJ)=2.*((DUDR**2+DVDY**2+.5*(DUDY+DVDR)**2+UOR**2)
339 C
340 C      PIXX,PIYY,PIXY,AND PITHTA ARE COMPONENTS OF THE VISCOSITY
341 C          STRESS TENSOR
342 C
343 C      PIIXX=4.*MU02*DUDR+LAMD
344 C      PIYY=4.*MU02*DVDY+LAMD
345 C      PIXY=2.*MU02*(DUDY+DVDR)
346 C      PITHTA=4.*MU02*UOR+LAMD*CYL
347 C
348 C      PITH IS 1/4 OF THE CELL AREA * PITHTA
349 C
350 C      PITH=.25*XY*PITHTA
351 C
352 C      CALCULATE THE VELOCITY CORRECTIONS FOR THE FOUR CELL VERTICES
353 C          USING THE EXPLICIT LAGRANGIAN FORM OF THE MOMENTUM EQN.
354 C          NOTE THAT THE TILDE VELOCITIES WERE INITIALIZED ABOVE IN
355 C          SUBROUTINE BODY FORCES AND THE EFFECTS OF THE
356 C          TUQYY=P(IJ)
357 C          IF(TURB) TUQYY=TUQYY+TWTHRD*R0(IJ)*TUQ(IJ)
358 C          YY=Y24*TUQYY
359 C          XX=HR24*(PIXY*X24+PIXX*Y24)
360 C          UTIL(IPJ)=UTIL(IPJ)+DT02M1*(XX+R1*YY=PITH)
361 C
362 C

```

```

363 UTIL(IJP)=UTIL(IJP)=DT02M3*(XX+R3*YY+PITH)
364 XX=HR13*(PIXY*X31-PIXX*Y31)
365 YY=Y31*TUQYY
366 UTIL(IPJP)=UTIL(IPJP)+DT02M2*(XX+R2*YY-PITH)
367 UTIL(IJ)=UTIL(IJ)=DT02M4*(XX+R4*YY+PITH)
368 PYYMP#PIYY=TUQYY
369 XX=HR24*(PYYMP*X24-PIXY*Y24)
370 VTIL(IPJ)=VTIL(IPJ)+DT02M1*XX
371 VTIL(IJP)=VTIL(IJP)=DT02M3*XX
372 XX=HR13*(PYYMP*X31-PIXY*Y31)
373 VTIL(IPJP)=VTIL(IPJP)+DT02M2*XX
374 VTIL(IJ)=VTIL(IJ)=DT02M4*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 !),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),UTILE(1),UL(1),CQ(1),VTIL(1)
13     3 ,VL(1),ROL(1),AVX8V(1),AVYSV(1),DLSROI(1),DLSROQ(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),DT02,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
25     2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,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

```

```

28      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
29      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,DM,OMCYL,PXCONV
30      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
31      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
32      COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEG8V,TUSV,TURB,PTOP,PRITE,PBOTM,
33      1 ILNG,NILNG,TP3,TUPOT,TDG8AV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
34      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
35      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,JSVW,GMN,QMX,WMAX,JNM,T2,TLIM,
36      4 ROMFXR,ROMFYT,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVS,DTCSAV,IDTV
37      5 ,JDTV,IDTC,JOTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMAX,ITM,JTM,ITG,JTG
38      6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
39      COMMON/YSC2/ZZ
40      C COMMON/YSC4/ITAB(ITABP)
41      COMMON/YSC4/ITAB(1000)
42      COMMON/YSC5/RESTART,FILM,PAPER,IPD,IFD
43      * ***** END COMDECK YAQSC *****
44      * ***** BEGIN COMDECK YAGEQ *****
45      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(AASC
46      1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSH,RCSQ,MP),(AASC
47      1 (8),E,ETIL,AREA,XR13K),
48      2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
49      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
50      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
51      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
52      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
53      7 ),(AASC(25),DLSROQ,VG),(AASC(26),GR8V),(AASC(27),GRRDP,TUQVEC,
54      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(A
55      9 AASC(31),ANCU),(AASC(32),ANCY),(AASC(33),AVXSV,X13K),(AASC(34),
56      1 AVYSV,X24K)
57      REAL M,MP,MPAR,MTIL
58      * ***** END COMDECK YAGEQ *****
59      * ***** END COMDECK YSTORE *****
60      * ***** BEGIN COMDECK PARAM *****
61      COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
62      1 NLCP3,NLCP4,IFLMSZ
63      * ***** END COMDECK PARAM *****
64      * ***** BEGIN COMDECK ASTORE *****
65      COMMON/ASTC/AT(100),FT(100)
66      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
67      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
68      1 ),(AT(9),YCO),(FT,CON)
69      * ***** END COMDECK ASTORE *****
70      COMMON/EQNST/ROTMP,ETMP,GMONE,CONCJ
71      C
72      C IF THE TURBULENCE IS ON, GO COMPUTE CORRECTIONS DUE TO IT
73      C
74      IF(TURB) CALL TRBCOR
75      CALL START
76      DO 20 J=2,JP1
77      DO 10 I=1,IBAR
78      IMJ=IJ+NQ
79      IPJ=IJ+NQ
80      IPJP=IJP+NQ
81      ROTMP=RO(IJ)
82      ETMP=SIE(IJ)
83      CONCJ=CONC(IJ)
84      CALL AIR

```

```

85      GM1=GM0NE
86      GGM1=GM1*(GM1+1.0)
87      C      ***** SET DENSITY FOR START OF PRESSURE ITERATION *****
88      MTIL(IJ)=RO(IJ)+ROL(IJ)*RVOL(IJ)*DT
89      ROL(IJ)=MTIL(IJ)
90      RCSQ(IJ)=1./(EM10+GGM1*AMAX1(SIE(IJ),0.))
91      IF(TURB.AND.T.GT.TSTRTD) CONC(IJ)=CONC(IJ)+CTEMP(IJ)*RVOL(IJ)*DT
92      UG(IJ)=1./(R(IJ)+R(IPJ)+R(IPJP)+R(IPJ))*CYL
93      X13=X(IPJ)-X(IJ)
94      X13K(IJ)=X13
95      X24=X(IPJP)-X(IJ)
96      X24K(IJ)=X24
97      Y13=Y(IPJ)-Y(IJ)
98      Y13K(IJ)=Y13
99      Y24=Y(IPJP)-Y(IJ)
100     Y24K(IJ)=Y24
101     VG(IJ)=1./(X13+Y24-X24+Y13)
102     XR13K(IJ)=0.5*(R(IPJ)+R(IJP))*X13
103     XR24K(IJ)=0.5*(R(IPJP)+R(IJ))*X24
104     IJ=IPJ
105     IPJ=IPJP
106     10 IJM=IJM+NQ
107     CALL LOOP
108     20 CONTINUE
109     CALL DONE
110     NUMIT=0
111     MUSTIT=1
112     PLMAX=EM10
113     30 CALL START
114     DO 60 J=2,JP1
115     DO 50 I=1,IBAR
116     IPJ=IJ+NQ
117     IPJP=IP+NQ
118     X1=X(IPJ)
119     Y1=Y(IPJ)
120     R1=R(IPJ)
121     U1=UL(IPJ)
122     V1=VL(IPJ)
123     X2=X(IPJP)
124     Y2=Y(IPJP)
125     R2=R(IPJP)
126     U2=UL(IPJP)
127     V2=VL(IPJP)
128     X3=X(IJP)
129     Y3=Y(IJP)
130     R3=R(IJP)
131     U3=UL(IJP)
132     V3=VL(IJP)
133     X4=X(IJ)
134     Y4=Y(IJ)
135     R4=R(IJ)
136     U4=UL(IJ)
137     V4=VL(IJ)
138     U0R=(U1+U2+U3+U4)*UG(IJ)
139     RAR=VG(IJ)
140     DUDX=((U1-U3)*(Y2-Y4)-(U2-U4)*(Y1-Y3))*RAR
141     DVDY=((V2-V4)*(X1-X3)-(V1-V3)*(X2-X4))*RAR

```

```

142      D=(DUDX+DVDY)*(1.+UOR*DT)+UOR
143      S=RDT*(ROL(IJ)-MTIL(IJ))+ROL(IJ)*D
144      RA=RC8Q(IJ)*(RDT+D)+DKL8M(IJ)
145      DP=DM*S/RA
146      ROL(IJ)=ROL(IJ)+RC8Q(IJ)*DP
147      PLMAX=AMAX1(PLMAX,ABS(PL(IJ)))
148      IF(ABS(DP),LE.,EPS*PLMAX) GO TO 40
149      MUSTIT#1
150      PL(IJ)=PL(IJ)+DP
151      Y24=Y2=Y4
152      Y31=Y3=Y1
153      XR13=.5*(R1+R3)*(X1-X3)
154      XR24=.5*(R2+R4)*(X2-X4)
155      XX=DT02*DP
156      DT02M1=XX*RM(IPJ)
157      DT02M2=XX*RM(IPJP)
158      DT02M3=XX*RM(IJP)
159      DT02M4=XX*RM(IJ)
160      UL(IPJ)=U1+DT02M1*R1*Y24
161      UL(IPJP)=U2+DT02M2*R2*Y31
162      UL(IJP)=U3=DT02M3*R3*Y24
163      UL(IJ)=U4=DT02M4*R4*Y31
164      VL(IPJ)=V1=DT02M1*XR24
165      VL(IJ)=V4=DT02M4*XR13
166      VL(IPJP)=V2=DT02M2*XR13
167      VL(IJP)=V3=DT02M3*XR24
168      40 IJ=IPJ
169      50 IJP=IPJP
170      CALL LOOP
171      60 CONTINUE
172      CALL DONE
173      C
174      C      GO SET THE BOUNDARY CONDITIONS
175      C
176      CALL BC(2)
177      NUMIT=NUMIT+1
178      IF(MUSTIT,EQ,0) GO TO 80
179      MUSTIT=0
180      IF(NUMIT,LT,500) GO TO 30
181      DO 70 IPX=IPD,IFD,6
182      70 WRITE(IPX,90)
183      80 CONTINUE
184      RETURN
185      C
186      90 FORMAT(42H ITERATION LIMIT EXCEEDED - RUN MAY ABORT.)
187      END

```

```

1      SUBROUTINE YPH3
2      C
3      C      ROUTINE TO FINISH PHASE 1 AND CALCULATE GRID VELOCITIES
4      C

```

```

5   C  ORIGINALLY WRITTEN BY A.A.AMSDEN AND HANS RUPPEL,LASL T-3
6   C  MODIFIED BY J.L.NORTON,LASL T-3,1975
7
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),DL8R0Q(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),GZ9V(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,DTA,DTFAC,
24  1 DTO(10),DTOC(10),DT02,DT08,DTPOS,DTV,DZ,EM$0,EPS,FIPXL,FIPXR,
25  2 FIPYB,FIPYT,FIXL,FIXR,FIYB,FIYT,FREZXR,GR,GRDVEL,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
28  COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
29  1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
30  2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
31  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
32  COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
33  1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,QLEVEL,TQ,IST,
34  2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
35  3 XCNVLD,FIXRD,FIXLO,IXRD,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
36  4 ROMFXR,ROMFYB,JDUMP,TWTHRD,TE,DTR,TMASS,DTVSAV,DTCSAV,DTV,
37  5 ,JDTV,IDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
38  6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
39  COMMON/YSC2/ZZ
40  C  COMMON/YSC4/ITAB(ITABP)
41  COMMON/YSC4/ITAB(1000)
42  COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
43  *  ***** END COMDECK YAQSC      *****
44  *  ***** BEGIN COMDECK YAQEQ      *****
45  EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
46  1 AASC(4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
47  1 (8),E,ETIL,AREA,XR13K),
48  2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
49  3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
50  4 ,UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
51  5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
52  6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZ9V
53  7 ),(AASC(25),DL8R0Q,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
54  8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
55  9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
56  1 AVYSV,X24K)
57  REAL M,MP,MPAR,MTIL
58  *  ***** END COMDECK YAQEQ      *****
59  *  ***** END COMDECK YSTORE      *****
60  *  ***** BEGIN COMDECK PARAM      *****
61  COMMON/PCOM/NSCP1,ITABP,ITABXP,ITABYP,IPFB,NP1,NP2,NLCP1,NLCP2,
```

```

62      1 NLCP3,NLCP4,IFLMSZ
63      *      ===== END COMDECK PARAM =====
64      *      ===== BEGIN COMDECK ASTORE =====
65      COMMON/ASTC/AT(100),FT(100)
66      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
67      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
68      1),(AT(9),YCO),(FT,CON)
69      *      ===== END COMDECK ASTORE =====
70      REAL LAMD,MU02
71      DTV=1,E30
72      CALL START
73      DO 40 J=2,JP1
74      DO 30 I=1,IBAR
75      IMJ=IJ+NQ
76      IPJ=IJ+NQ
77      IPJP=IJP+NQ
78      X1=X(IPJ)
79      Y1=Y(IPJ)
80      R1=R(IPJ)
81      U1L=UL(IPJ)
82      U1=U(IPJ)
83      V1L=VL(IPJ)
84      V1=V(IPJ)
85      X2=X(IPJP)
86      Y2=Y(IPJP)
87      R2=R(IPJP)
88      U2L=UL(IPJP)
89      U2=U(IPJP)
90      V2L=VL(IPJP)
91      V2=V(IPJP)
92      X3=X(IJP)
93      Y3=Y(IJP)
94      R3=R(IJP)
95      U3L=UL(IJP)
96      U3=U(IJP)
97      V3L=VL(IJP)
98      V3=V(IJP)
99      X4=X(IJ)
100     Y4=Y(IJ)
101     R4=R(IJ)
102     U4L=UL(IJ)
103     U4=U(IJ)
104     V4L=VL(IJ)
105     V4=V(IJ)
106     X12=X1-X2
107     X23=X2-X3
108     X34=X3-X4
109     X41=X4-X1
110     X24=X24K(IJ)
111     X31=X13K(IJ)
112     Y24=Y24K(IJ)
113     Y31=Y13K(IJ)
114     Y21=Y2=Y1
115     Y32=Y3=Y2
116     Y43=Y4=Y3
117     Y14=Y1=Y4
118     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      U0R#(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+DVDX)*(1.+U0R*D7)+U0R
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      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      ALAM#AK*LAM
154      AMU#AK*MU
155      LAMD#AMIN1(D,0.)*ALAM
156      MU02#5*AMU
157      FMU02#4.*MU02
158      PIXX#FMU02*DUDX+LAMD
159      PIYY#FMU02*DVDY+LAMD
160      PIXY#2.*MU02*(DUDY+DVDX)
161      PITH#0.25*XY*(FMU02*U0R+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) IDTV#I
174      IF(DTVSAV.NE.DTV) JDTV#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      ! 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      ! )=(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*RR0
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      TNEG=TNEG
197      TK2=0.
198      TI2=0.
199      TP2=0.
200      TQ2=0.
201      TUPO=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      FFE=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      I(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

```

```

233      E(IMJ)=E(IJM)+2,*VTEMP*RVOL(IMJ)/RO(IMJ)
234      E(IMJM)=E(IMJM)+2,*VTEMP*RVOL(IMJM)/RO(IMJM)
235      GO TO 110
236      90 IF(J,EO,2) GO TO 100
237      E(IMJM)=F(IMJM)+2,*VTEMP*RVOL(IMJM)/RO(IMJM)
238      E(IJM)=E(IJM)+2,*VTEMP*RVOL(IJM)/RO(IJM)
239      GO TO 110
240      100 E(IJ)=E(IJ)+2,*VTEMP*RVOL(IJ)/RO(IJ)
241      E(IMJ)=E(IMJ)+2,*VTEMP*RVOL(IMJ)/RO(IMJ)
242      110 CONTINUE
243      IJ=IPJ
244      IJP=IJP+NQ
245      120 IJM=IJM+NQ
246      CALL LOOP
247      130 CONTINUE
248      CALL DONE
249      CALL START
250      IFAKE=IJ
251      DO 190 J=2,JP1
252      DO 180 I#1,IBAR
253      IPJ=IJ+NQ
254      IPJP=IJP+NQ
255      IF(,NOT,TURB) GO TO 140
256      FF=RO(IJ)/RVOL(IJ)
257      TI2=TI2+E(IJ)*FF
258      TQ2=TQ2+FF*TUQ(IJ)
259      IMJ=IJ+NQ
260      SS=SIG(IJ)/RVOL(IJ)
261      TUPOT=TUPOT-,25*SS*(GRROZ(IJ)*(U(IJ)+U(IPJ)+U(IPJP)+U(IJP))+GRROZ(IJ)*(V(IJ)+V(IPJ)+V(IPJP)+V(IJP)))
262      TUPOTA=TUPOTA-SS*GZ*DT*GRROZ(IJ)
263      140 CONTINUE
264      IF(J,NE,2) GO TO 150
265      C ***** SET BOTTOM FICTICIOUS ROW *****
266      ROL(IJM)=RO(IJM)
267      ETIL(IJM)=ETIL(IJ)
268      IF(TURB) TUQ(IJM)=TUQ(IJ)
269      CONC(IJM)=CONC(IJ)
270      IF(I,NE,IBAR) GO TO 150
271      C ***** SET LOWER RIGHT FICTICIOUS CORNER *****
272      C ROL(IJM+NQ)=RO(IJM+NQ)
273      C ETIL(IJM+NQ)=ETIL(IFAKE)
274      C IF(TURB) TUQ(IJM+NQ)=TUQ(IFAKE)
275      C CONC(IJM+NQ)=CONC(IFAKE)
276      C GO TO 160
277      150 IF(J,NE,JP1) GO TO 160
278      C ***** SET TOP FICTICIOUS ROW *****
279      C ROL(IJP)=RO(IJP)
280      C IF(TURB) TUQ(IJP)=TUQ(IJ)
281      C CONC(IJP)=CONC(IJ)
282      C 160 IF(I,NE,IBAR) GO TO 170
283      C ***** SET RIGHT HAND FICTICIOUS COLUMN *****
284      C ROL(IPJ)=RO(IPJ)
285      C ETIL(IPJ)=ETIL(IJ)
286      C IF(TURB) TUQ(IPJ)=TUQ(IJ)
287      C CONC(IPJ)=CONC(IJ)
288      C 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      TDQ=TQ2-TUGFNG
297      TE=TK+TI+TUGFNG
298      DELTAE=EP1+TUPOT+TUPOTA
299      DELEWR=DELTAE
300      FIDEL=DELEWR=SAV1
301      IF(TDQSAV.NE.0.) ERATIO=FIDEL/TDQSAV
302      TDQSAV=TDQ
303      SAV1=DELTAE
304      200 CONTINUE
305      IF(GRDVEL.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*GRDVEL*(UL(IJ)+U(IJ))
310      VG(IJ)=0.5*GRDVEL*(VL(IJ)+V(IJ))
311      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      IJ=IJ+NQ
322      CALL LOOP
323      240 CONTINUE
324      CALL DONE
325      GO TO 260
326      250 CALL REZONE
327      260 CONTINUE
328      RETURN
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,IPFB,NP1,NP2,NLCP1,NLCP2,
10      NLCP3,NLCP4,IFLMSZ

```

```

11      *      ===== END COMDECK PARAM =====
12      *      ===== BEGIN COMDECK ASTORE =====
13      COMMON/A8TC/AT(100),FT(100)
14      DIMENSION IX1(1),IY1(1),IX2(1),IY2(1),XCO(1),YCO(1),CON(1)
15      EQUIVALENCE(AT,IX1),(AT(2),IX2),(AT(3),IY1),(AT(4),IY2),(AT(5),XCO
16      ),(AT(9),YCO),(FT,CON)
17      *      ===== END COMDECK ASTORE =====
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),AVYBV(1),DLSROI(1),DLSRQ(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),DKLSM(1),AREA(1)
27      *      ===== END COMDECK YAQDIM =====
28      *      ===== BEGIN COMDECK YAQSC =====
29      LOGICAL RESTRT,FILM,PAPER,TURB
30      REAL LAM,MU
31      C      COMMON/YSC1/AASC(NSCP1)
32      COMMON/YSC1/AASC(9600)
33      COMMON/YSC2/AA(1),ANC,A0,A0FAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
34      1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,DTV,DZ,EM10,EPS,FIPXL,FIPXR,
35      2 FIPYB,FIPYT,FXL,FXIR,FIYB,FIYT,FREZXR,GR,GRDVEL,GZ,GZP,I,IBAR,
36      3 IDTO,IJ,IJM,IJP,IM1,IPXL,IPXR,IPYB,IPYT,IP1,IP2,ISC2,ISC3,ITV,
37      4 IUNF,IXL,IXR,IYB,IYT,J,JBAR
38      COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
39      1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
40      2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
41      3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
42      COMMON/YSC2/TUQ1,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
43      1 ILNG,NILNG,TP3,TUPOT,TDGSBV,TK,TI,TUGENG,EP1,SAV1,OLEVEL,TQ,IST,
44      2 VV,XCONV,XL,XR,YB,YCONV,YT,PTPOLD,DTSV,DTLAST,FIYBO,IYBO,YCNVLD,
45      3 XCNVLD,FIXRO,FIXLO,IXRO,IXLO,ISVW,J8VW,QMN,QMXY,WMAX,JNM,T2,TLIM,
46      4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTR,MASS,DTVS,DTCSA,DTCSA,IDTV
47      5 ,JDTV,JDTC,CIRC,TIS,POTE,UMOM,VMMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
48      6 ,TMASSV,WMAXEF,RMINFF,TSTRTD
49      COMMON/YSC2/ZZ
50      C      COMMON/YSC4/ITAB(ITABP)
51      COMMON/YSC4/ITAR(1000)
52      COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
53      *      ===== END COMDECK YAQSC =====
54      *      ===== BEGIN COMDECK YAQEQ =====
55      EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(AASC
56      1 (4),U),(AASC(5),V),(AASC(6),RO),(AASC(7),DELSM,RCSQ,MP),(AASC
57      1 (8),F,ETIL,AREA,XR13K),
58      2 ,(AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
59      3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
60      4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
61      5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
62      6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
63      7 ),(AASC(25),DLSRQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
64      8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
65      9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
66      1 AVYSV,X24K)
67      REAL M,MP,MPAR,MTIL

```

```

68      *      ===== END COMDECK YAGEQ =====
69      *      ===== END COMDECK YSTORE =====
70      *      ===== BEGIN COMDECK PCALL =====
71      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
72      *      ===== END COMDECK PCALL =====
73      LOGICAL EULER
74      DIMENSION BCD(2)
75      DIMENSION ITITLE(6,8),INWT(8)
76      DIMENSION IJL1(6),IJL2(6),IJL3(6),IJL4(6),IJL5(6),IJL6(6)
77      DIMENSION IJL7(6),IJL8(6)
78      EQUIVALENCE(ITITLE,IJL1),(ITITLE(7),IJL2),(ITITLE(13),IJL3),
79      1 ITITLE(19),IJL4),(ITITLE(25),IJL5),(ITITLE(31),IJL6),(ITITLE(37),
80      2 IJL7),(ITITLE(43),IJL8)
81      DATA IJL1/7HDENSITY,5*1H /
82      DATA IJL2/24HSPECIFIC INTERNAL ENERGY,3*1H /
83      DATA IJL3/9HVORTICITY,5*1H /
84      DATA IJL4/18HVELOCITY MAGNITUDE,4*1H /
85      DATA IJL5/26HSPECIFIC TURBULENCE ENERGY,3*1H /
86      DATA IJL6/30HRATIO OF SPEC. TURB. ENERGY TO,
87      1 21H SPEC. KINETIC ENERGY/
88      DATA IJL7/28HTURBULENCE VISCOISITY (SIGMA),3*1H /
89      DATA IJL8/22HCONCENTRATION X RADIUS,3*1H /
90      DATA INWT/1,3,1,2,3,6,3,3/
91      DATA BCD/1H /
92      DATA VMAX,DRMIN,DZMIN,DRMAX,DZMAX/5*0.,/IVM,JVM/0,0/
93      DATA TP/0./
94      CALL SECOND(TP)
95      WRITE(59,210) TP
96      C
97      C      IF THERE ARE ANY PARTICLES, GO PLOT THEM
98      C
99      C      IF(NPT.GT.0) CALL PARPLT
100     C
101     C      Euler is true if the calculation is Eulerian
102     C
103     C      Euler=.FALSE.,
104     C      IF(GRDVFL.EQ.0.) Euler=.TRUE.
105     C
106     C      Initialize coordinate-to-raster conversion factors in common
107     C
108     C      YCONVP=YCONV
109     C      XCONVP=XCONV
110     C
111     C      Initialize minimum plot y value in common
112     C
113     C      YLB=YB
114     C
115     C      Go get the maximum velocity absolute value along the coordinate
116     C      directions and the minimum and maximum zone sizes
117     C
118     C      CALL DVMM(VMAX,IVM,JVM,DRMIN,DZMIN,DRMAX,DZMAX)
119     C      VTEST=.05*VMAX
120     C
121     C      If doing an Eulerian calculation, only do zone plots upon
122     C      startup. Otherwise, do them every output cycle.
123     C
124     C      IF(EULER,AND,NCYC,NE,0) GO TO 20

```

```

125      CALL ADV(1)
126      CALL ZONPLT
127
128      C PUT LABELS ON THE ZONE PLOT AND OUT TO THE CYCLE SUMMARY PRINT
129      C
130      CALL LINCNT(59)
131      WRITE(IFD,220)
132      DO 10 IPX=6,IFD,6
133      10 WRITE(IPX,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
134      WRITE(IFD,260) JNM,NAME,T,NCYC
135      20 CONTINUE
136
137      C SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO,
138      C OTHERWISE, PLOT VELOCITIES SCALED TO THE MAXIMUM VELOCITY.
139      C
140      C 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
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      IYB0=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      IXR0#IXR
165      FIXL#65.
166      IXL#65
167      FIXR#660.
168      IXR#660
169      YCNVL0#YCONV
170      YCONV#YCONVP
171      XCNVL0#XCONV
172      XCONV#XCONVP
173
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
180      C PUT LABELS ON THE ZONE PLOT
181      C

```

```

182      CALL LINCNT(59)
183      WRITE(IFD,230)
184      WRITE(IFD,250) DRMIN,DRMAX,DZMIN,DZMAX,XR,YB,YT
185      WRITE(IFD,260) JNM,NAME,T,NCYC
186      C
187      C      DRAW THE PLOTTING RECTANGLE AND LABEL IT
188      C
189      C      CALL TICBOX
190      30 CONTINUE
191      C
192      C      SKIP VELOCITY PLOTS IF VELOCITIES ARE ESSENTIALLY ZERO
193      C
194      C      IF(VMAX.LT.EM10) GO TO 40
195      C
196      C      PLOT VELOCITIES SCALED TO MAXIMUM VELOCITY
197      C
198      C      CALL VELPLT(VMAX,0)
199      C      CALL TICBOX
200      C
201      C      PLOT UNSCALED VELOCITIES
202      C
203      C      CALL VELPLT(VMAX,1)
204      C      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 (ISOHERMS)
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      C      LP#8
221      C      IF(,NOT,TURB) LP=4
222      C
223      C      PREPARE THE PLOTTING ARRAYS
224      C
225      C      DO 200 L=1,LP
226      C
227      C      CHECK FOR VORTICITY PLOT
228      C
229      C      IF(L.NE,3) GO TO 50
230      C
231      C      YES, SPECIAL SUBROUTINE PREPARES VORTICITY,
232      C
233      C      CALL GETOMG
234      C      GO TO 190
235      50 CONTINUE
236      C      CALL START
237      C      DO 180 J=2,JP1
238      C      DO 170 I=1,IBAR

```

```

239      GO TO (60,70,200,80,80,80,140,80),L
240      60 CONTINUE
241      C
242      C      DENSITY
243      C
244      C      CQ(IJ)=RQ(IJ)
245      C      GO TO 160
246      C      70 CONTINUE
247      C
248      C      SPECIFIC INTERNAL ENERGY
249      C
250      C      CQ(IJ)=SIE(IJ)
251      C      GO TO 160
252      C      80 CONTINUE
253      C      IPJ=IJ+NQ
254      C      IPJP=IJP+NQ
255      C      IF(L.EQ.5.OR.L.EQ.8) GO TO 90
256      C      XXA=(0.25*(U(IJ)+U(IPJ)+U(IPJP)+U(IJP)))**2+(0.25*(V(IJ)+V(IPJ)+V(
257      C      1 IPJP)+V(IJP)))**2
258      C      GO TO 100
259      C      90 CONTINUE
260      C      RAV=0.25*(R(IJ)+R(IPJ)+R(IPJP)+R(IJP))
261      C      100 CONTINUE
262      C      LPX=L-3
263      C      GO TO (110,120,130,140,150),LPX
264      C      110 CONTINUE
265      C
266      C      VELOCITY MAGNITUDE
267      C
268      C      CQ(IJ)=SQRT(XXA)
269      C      GO TO 160
270      C      120 CONTINUE
271      C
272      C      SPECIFIC TURBULENCE ENERGY
273      C
274      C      CQ(IJ)=TUQ(IJ)
275      C      GO TO 160
276      C      130 CONTINUE
277      C
278      C      RATIO OF SPEC. TURB. ENERGY TO SPEC. KINETIC ENERGY
279      C
280      C      CQ(IJ)=TUQ(IJ)/(XXA+EM10)*2,
281      C      TEST=AMAX1(ABS(U(IJ)),ABS(V(IJ)))
282      C      IF(TEST.LT.VTEST) CQ(IJ)=0.
283      C      GO TO 160
284      C      140 CONTINUE
285      C
286      C      TURBULENCE VISCOSITY (SIGMA)
287      C
288      C      CQ(IJ)=SIG(IJ)
289      C      GO TO 160
290      C      150 CONTINUE
291      C
292      C      CONCENTRATION X RADIUS
293      C
294      C      CQ(IJ)=CONC(IJ)*RAV
295      C      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
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      ILLOG#1
309      IF(L.GT.2) ILLOG#0
310
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 CONTOUR(ILLOG,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=YCNVLD
324      XCONV=XCNVLD
325      FIXR=FIXRD
326      FIXL=FIXLD
327      IXR=IXRD
328      IXL=IXLD
329      CALL SECOND(TP)
330      WRITE(59,240) TP
331      RETURN
332
333 C      210 FORMAT(1H ,20HBEGIN YPLOT AT CP = ,F10,4)
334 C      220 FORMAT(10H ALL ZONES)
335 C      230 FORMAT(29H ZONES IN THE FIREBALL REGION)
336 C      240 FORMAT(1H ,18HEND YPLOT AT CP = ,F10,4)
337 C      250 FORMAT(7H DRMIN#1PE12.5,7H DRMAX#E12.5,7H DZMIN#E12.5/7H DZMAX#E12
338 C           1 .5,4H XR#E12.5,4H YB#E12.5,4H YT#E12.5)
339 C      260 FORMAT(1H ,4XA10,8A10,3X2HT#,1PE12.5,1X6HCYCLE#,IS)
340      END

```

```

1      SUBROUTINE ZONPLT
2
3      C      ROUTINE TO DO FULL AND PARTIAL ZONE PLOTS
4
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,NLCPI,NLCP2,
10  NLCP3,NLCP4,IFLMSZ
11  *      ----- END COMDECK PARAM -----
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),RD(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,AP,AOFAC,A0M,B0,COLAMU,CYL,DR,DT,DTC,DTFAC,
28  1 DTO(10),DTOC(10),DTO2,DT08,DTPOS,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,IYR,IYT,J,JBAR
32  COMMON/YSC2/JCEN,JP1,JP2,JP4,JUNF,JUNFO2,KXI,LAM,LPB,MU,NAME(8),
33  1 NCYC,NLC,NPS,NPT,NQ,NQI,NQIB,NQI2,NSC,NUMIT,ZORIG,OM,OMCYL,PXCONV
34  2 ,PXL,PXR,PYB,PYCONV,PYT,RDT,REZRON,REZSIE,REZY0,RIBAR,RIBJB,
35  3 FREZYT,FREZYB,ROMFR,T,THIRD,NCLST,TOUT,TWFIN
36  COMMON/YSC2/TUQI,TUSI,NCQ,TNEG,TNEGSV,TUSV,TURB,PTOP,PRITE,PBOTM,
37  1 ILNG,NILNG,TP3,TUPOT,TDQSAV,TK,TI,TUQENG,EP1,SAV1,OLEVEL,TQ,IST,
38  2 VV,XCONV,XL,XR,YR,YCONV,YT,PTPOLD,DTSLV,DTLAST,FIYBO,IYBO,YCNVLD,
39  3 XCNVLD,FIXRO,FIXL,IXRO,IXLO,ISVW,JSVW,QMN,QMX,WMAX,JNM,T2,TLIM,
40  4 ROMFXR,ROMFYB,JDUMP,TWTHR,DTE,DTR,TMASS,DTVSAV,DTCSAV,IDTV
41  5 ,JDTV,JDTC,JDTC,CIRC,TIS,POTE,UMOM,VMOM,TMAX,TGMX,ITM,JTM,ITG,JTG
42  6 ,TMASSV,WMAXEF,RMINEF,TSTRTD
43  COMMON/YSC2/ZZ
44  C      COMMON/YSC4/ITAB(ITABP)
45  COMMON/YSC4/ITAB(1000)
46  COMMON/YSC5/RESTRT,FILM,PAPER,IPD,IFD
47  *      ----- END COMDECK YAQSC -----
48  *      ----- BEGIN COMDECK YAQEQ -----
49  EQUIVALENCE(AASC(1),X,XPAR),(AASC(2),R,YPAR),(AASC(3),Y,MPAR),(
50  1 AASC(4),U),(AASC(5),V),(AASC(6),RD),(AASC(7),DELSM,RCSQ,MP),(AASC
51  1 (8),E,ETIL,AREA,XR13K),
52  2 (AASC(15),SIE),(AASC(16),PM0,DKLSM,RMP),(AASC(9
53  3 ),RVOL),(AASC(10),M,RM,VP),(AASC(11),P,PL,EP,UP),(AASC(12),UTIL,
54  4 UL,PMX,PU),(AASC(13),VTIL,VL,PMY,PV),(AASC(14),Q,CQ,ROL),(AASC(17
55  5 ),CAPGAM,UG),(AASC(18),TUQ),(AASC(19),SIG),(AASC(20),TUS),(AASC(
56  6 21),GRROR),(AASC(22),GRROZ),(AASC(23),DLSROI,Y13K),(AASC(24),GZSV
57  7 ),(AASC(25),DLSROQ,VG),(AASC(26),GRSV),(AASC(27),GRROP,TUQVEC,
58  8 Y24K),(AASC(28),MTIL),(AASC(29),CONC),(AASC(30),CTEMP,XR24K),(
59  9 AASC(31),ANCU),(AASC(32),ANCV),(AASC(33),AVXSV,X13K),(AASC(34),
1 AVYSV,X24K)
60  REAL M,MP,MPAR,MTIL
61  *      ----- END COMDECK YAQEQ -----

```

```

63      *      ***** END COMDECK YSTORE      *****
64      *      ***** BEGIN COMDECK PCALL      *****
65      COMMON/PCALLC/XCONVP,YCONVP,YUP,YLB
66      *      ***** END COMDECK PCALL      *****
67      C      LOOP OVER ALL REAL ZONES
68      C
69      JB=0
70      CALL START
71      DO 60 J=2,JP1
72      DO 50 I=1,IBAR
73      IPJ=IJ+NO
74      IPJP=IJP+NO
75      C
76      C      DETERMINE THE FOUR VERTICES OF CELL (I,J)
77      C
78      C      (X1,Y1) IS VERTEX (I+1,J)      (VERTEX 1)
79      C      (X2,Y2) IS VERTEX (I+1,J+1)    (VERTEX 2)
80      C      (X3,Y3) IS VERTEX (I,J+1)      (VERTEX 3)
81      C      (X4,Y4) IS VERTEX (I,J)        (VERTEX 4)
82      C
83      X1=X(IPJ)
84      X2=X(IPJP)
85      X3=X(IJP)
86      X4=X(IJ)
87      Y1=Y(IPJ)
88      Y2=Y(IPJP)
89      Y3=Y(IJP)
90      Y4=Y(IJ)
91      C
92      C      (IXN,IYN) ARE THE PLOT RASTER COORDINATES OF VERTEX N
93      C
94      IX1=FIXL+(X1-XL)*XCONVP
95      IX2=FIXL+(X2-XL)*XCONVP
96      IX3=FIXL+(X3-XL)*XCONVP
97      IX4=FIXL+(X4-XL)*XCONVP
98      IY1=FIYB+(Y1-YLB)*YCONVP
99      IY2=FIYB+(Y2-YLB)*YCONVP
100     IY3=FIYB+(Y3-YLB)*YCONVP
101     IY4=FIYB+(Y4-YLB)*YCONVP
102     C
103     C      IF VERTEX 1 OR 2 IS ABOVE THE TOP OR BELOW THE BOTTOM OF THE
104     C      PLOT REGION,DO NOT PLOT THIS ZONE
105     C
106     IF(IY1,GT,IYB,OR,IY1,LT,IYT) GO TO 40
107     IF(IY2,GT,IYB,OR,IY2,LT,IYT) GO TO 40
108     C
109     C      IF VERTEX 1 OR 2 IS TO THE RIGHT OF THE PLOT REGION,SKIP
110     C      THIS ZONE
111     C
112     IF(IX1,GT,IXR) GO TO 40
113     IF(IX2,GT,IXR) GO TO 40
114     C
115     C      IF THIS ZONE IS ON THE SYMMETRY AXIS,DRAW THE LEFT SIDE
116     C
117     IF(I,EQ,1) CALL DRV(IX3,IY3,IX4,IY4)
118     C
119     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 IJEIPJ
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 Simulation Technique (GEST) Experiments," Los Alamos Scientific Laboratory report LA-6154-MS (December 1975).