LA-5680

UC-32, 78, and 79
Reporting Date: July 1974
Issued: December 1974

KACHINA: an Eulerian Computer Program for Multifield Fluid Flows

by

Anthony A. Amsden
Francis H. Harlow

los alamos
scientific laboratory
of the University of California
LOS ALAMOS, NEW MEXICO 87544

UNITED STATES
ATOMIC ENERGY COMMISSION
CONTRACT W-7405-ENG. 36

Work partially supported by the US AEC Division of Reactor Safety Research.

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, VA 22151
Price: Printed Copy $5.45 Microfiche $2.25

CONTENTS

Page

Abstract 1

I. Basic Description of the Computing Method 1
  - A. Introduction 1
  - B. The Variables and the Computing Mesh 2
  - C. Phase 1 — Explicit Calculations 4
  - D. Phase 2 — Implicit Calculations 10
  - E. Boundary Conditions 15
  - F. Sudden Incompressibility 17

II. The KACHINA Computing Program 18
  - A. General Structure 18
  - B. The Indexing Notation 19
  - C. Storage of Cell Data 20
  - D. An Optional Three-Row Buffering Scheme 23
  - E. Marker Particles 25
  - F. The Input Data 27
  - G. Output — Plots, Prints, and Motion Pictures 31
  - H. Tape Dump and Restart 33
  - I. The Common Block KSC 33
  - J. Microfilm Plots — Scaling and Subroutine CALLs 37
  - K. Miscellaneous System Subroutine CALLs 39

III. References 40

Appendix A. Flow Diagram for the KACHINA Program (June 19, 1974 Status) 41

Appendix B. FORTRAN IV Index Listing of the KACHINA Program (June 19, 1974 Status) 65

Appendix C. Modifications for LCM Storage of Cell Data 132

KACHINA: AN EULERIAN COMPUTER PROGRAM FOR MULTIFIELD FLUID FLOWS

by

Anthony A. Amsden and Francis H. Harlow

ABSTRACT

Many fluid flow problems of interest involve the presence of bubbles, droplets, or chunks in a fluid. Because of the resulting likelihood for relative motion, more than one set of field variables is required to describe the dynamics. Until recently, numerical techniques for studying such flows have had very limited usefulness. This report presents a new computing program, named KACHINA, that significantly advances the ability to handle multifield flows. KACHINA uses the powerful new Implicit, Multifield (IMF) computing method for handling different material fields, in which the multifield treatment is coupled with an implicit formulation of the equations, permitting calculation of fully interpenetrating flows that at any instant may have both supersonic and far subsonic or incompressible regions in the domain of interest. Although IMF-KACHINA development is a continuing project, the results have already proved its usefulness, even in its present state. This report includes the current KACHINA flow diagram and program listing.

I. BASIC DESCRIPTION OF THE COMPUTING METHOD

A. Introduction

The Implicit Continuous-fluid Eulerian (ICE) method[1] has become well known and widely accepted since its introduction, as it was the first technique to afford a means for numerical solution of multidimensional flows in which the Mach number might range from zero (the incompressible limit) to greater than unity (the supersonic regime). The ICE concept has been recast in various formulations; for example, in the YAQUI program,[2] ICE was combined with the Arbitrary Lagrangian-Eulerian (ALE) technique,[3] to provide the additional capability of fully variable zoning and rezoning. Other programs based on the ICE concept have included effects such as reactive chemistry,[4] magnetohydrodynamics,[5] and multiple fluids.[6] This ability to calculate a wide variety of flows has contributed significantly to an understanding of atmospheric explosion phenomenology, laser design theory, plasma physics in the CTR program, many flows in biological systems, and even advanced astrophysical concepts, to name just a few.

Development of numerical techniques still has a long way to go, however, in providing man an understanding of all the fluid dynamic processes that interest him. One common type of flow has defied really successful numerical modeling. It involves the presence of bubbles, droplets, or chunks in a fluid, implying that relative motions must be considered, and a complex set of field variables is required to describe the dynamics with any accuracy. Examples of such flows are:

- Ordinary snow, rain, or hail falling through the atmosphere.
- Cavitation or flashing flow, in which bubbles of vapor are formed from the fluid itself. (Visualize the formation of steam by boiling,

or the propagation of a flame front through a confined explosive such as encased gunpowder.)

- Liquid or vapor rising through a bed of solid grains in a fluidized dust bed.
- Jet entrainment, in which immiscible or mutually diffusing liquid droplets are carried along or mixed with another liquid.

Until recently, numerical techniques for examining such multifield processes were extremely limited. A powerful new computing technique,[7] known as the Implicit, Multifield (IMF) method and based, once again, on ICE, is now available to help overcome these limitations and thus significantly advance the art of modeling multifield flows. This report discusses a program named KACHINA, which embodies the IMF methodology. Our treatment is based upon an implicit formulation of the coupled set of differential equations for multidimensional, multifield flow. Because of the program's Eulerian aspects, it can follow completely interpenetrating material motions over long periods of time, and because of the implicit treatment of mass convection and the equation of state, the flows at any instant may have both supersonic and far subsonic regions in the domain of interest. Further, the implicit coupling of the fields allows forces to range from negligibly weak to strong enough to tie the fields together completely. The program also can pile up a particulate field into a close-packed region with a variable boundary position, and possibly reopen such a region later.

Development of IMF and KACHINA is an expanding and continuing project. In its present basic state, KACHINA is still fairly limited and does not take into account a number of physical processes that will be required for future applications. It has, however, proved its usefulness even at this point, and has permitted meaningful calculations of a variety of one- and two-dimensional two-field flow situations that could not have been made using previous techniques. This report includes the current version of the KACHINA flow diagram and program listing. The derivation of the technique is fully described in Ref. 7, which also includes examples of a variety of test calculations illustrating some of KACHINA's capabilities. We therefore omit such aspects here, and concentrate on the solution procedure and the equations as they appear in KACHINA.

B. The Variables and the Computing Mesh

For simplicity, we presently limit our consideration to two primary material fields, although the IMF principle is not restricted to these two, but will be developed further into a three-field model, including full treatment of phase transitions among the fields. We label these two fields "vapor" and "droplets." The vapor field may be considered to be a gas in bubble form or with dispersed droplets in it, and the droplet field to be a fluid or an aggregate of solid particles. The components of each field have constituents and properties that can vary in space and time, but pressure is assumed to be in local equilibrium between the two fields. The pressure is related directly to the equation of state of the vapor when the droplet field is disperse, or to the maintenance of incompressibility when the droplet field is close-packed.

At present, the coupled fields are represented on a two-dimensional axisymmetric grid of fixed Eulerian cells through which the fluid moves. Cells have uniform dimensions δr and δz, measured in the radial (r) and axial (z) directions, respectively, and they are labeled by indices located at their centers, with i counted in the r direction and j in the z direction. The mesh of cells is $\bar{I}$ cells wide by $\bar{J}$ cells high, as shown in Fig. 1.

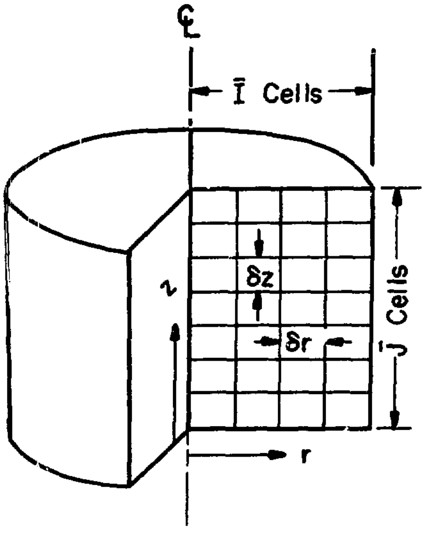


Fig. 1. The KACHINA axisymmetric two-dimensional computing mesh.

Variables may be defined either at cell centers or at cell edges, in which case they are labeled by half-integer indices in the finite-difference notation. The location of the principal KACHINA variables about a cell (i,j) is illustrated in Fig. 2, where u and v are velocity components in the radial and axial directions, respectively, ρ is the density, and I is the specific internal energy. Each of these four variables has both vapor and droplet components, denoted by the subscripts "v" and "d," respectively. The pressure, p, has only a single value in the cell because of the local equilibrium between the two fields.

The treatment of such a mixture of droplets and vapor, and the procedure for applying the equation of state to the vapor, requires a knowledge of the proportions of vapor and droplets within any given cell volume. For this purpose, we use the void fraction, θ, defined as the volume per unit total volume occupied by vapor. Consequently, $(1-\theta)$ is the volume per unit total volume occupied by droplets.

In addition, we allow two components within each field, designating them by subscripts 1 and 2. One must be able to describe the varying relative proportions of the two components in a way that ensures the separate conservation of each. For either field,

$$\theta_1 = \theta_2 \quad ,$$

and

$$\rho' \equiv \rho'_1 + \rho'_2 \quad ,$$

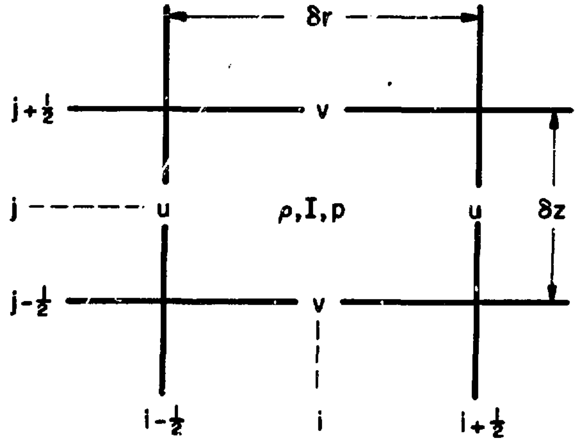


Fig. 2. Location of variables and indices about a KACHINA cell.

in which the prime signifies the mass per unit total volume. For the droplets, we use conservative explicit equations to transport the two densities ρ'_d and ρ'_{d1}, and get the value of ρ'_{d2} simply by subtraction when necessary. Then, with the corresponding normal material densities ρ_1 and ρ_2 defined, microscopic incompressibility allows derivation of an expression for the region's effective θ.

For the vapor, an equation for ρ'_v, the sum of the two vapor densities, is required for the implicit coupling with the momentum equations. After this advanced-time density and the corresponding advanced-time velocities have been determined, ρ'_{v1} can be obtained by means of a second implicit solution. As in the case of the droplets, the value of ρ'_{v2} is available by subtraction when needed.

Thus, it is seen that the complete arrays of four densities must be stored and maintained, two for each field, and that we have chosen these to be ρ'_{d1} and ρ'_d for the droplets, and ρ'_{v1} and ρ'_v for the vapor.

The input data used to create the droplet and vapor fields specify the initial values of ρ'_{d1}, ρ'_{d2}, ρ_{v1}, ρ_{v2}, I_v, and I_d for all cells of the mesh, where ρ_{v1} and ρ_{v2} are the actual microscopic (partial) densities of the two vapor components. Using this information, one may place I_v, I_d, and ρ'_{d1} directly in cell storage, but generation of the initial fields of ρ'_d, ρ'_{v1}, and ρ'_v, along with θ, requires some preliminary calculation.

- The field of ρ'_d is formed simply from the sum of ρ'_{d1} and ρ'_{d2}.
- With the normal material densities ρ_1 and ρ_2 of the droplet field components also specified in the input data, the initial void fraction is obtained from the relationship

$$\theta = 1 - \frac{\rho'_{d1}}{\rho_1} - \frac{\rho'_{d2}}{\rho_2} \quad .$$

- The values of ρ_{v1} and ρ_{v2} are then combined with θ to obtain the fields of ρ'_{v1} and ρ'_v:

$$\rho'_{v1} = \theta\rho_{v1} \quad ,$$

$$\rho'_v = \theta\left(\rho_{v1} + \rho_{v2}\right) \quad .$$

Care must be taken to ensure that the input data are not specified incorrectly so that $\theta < 0$ results.

In the present code, the dynamics are assumed to arise from pressure gradients from internal heat sources, or as the result of externally applied boundary conditions and/or gravitational effects. Therefore, all four velocity components, u_v, u_d, v_v, and v_d are initially set to zero throughout the interior of the mesh.

Because the vapor and droplet fields are bound by forces that can create conditions ranging from a complete tying together to the allowance of freely independent motion, a drag function, K, is used to relate the momentum exchange between the fields.

Other cell quantities will be introduced in the discussion of the full calculation cycle, but those described so far may be considered the principal KACHINA variables.

Solution of the dynamics evolves through a sequence of cycles, or time steps, each of duration δt. For each time step, the full coupled set of equations is solved to get the new values of all the field variables at a time δt later. This solution uses the results of the previous cycle or the set of initial conditions, and it is stored so as to allow the processing to be repeated in the next cycle.

Each cycle in KACHINA is composed of two distinct phases. The explicit calculations are performed in Phase 1, and all implicit aspects are then handled in Phase 2.

C. Phase 1 — Explicit Calculations

In Phase 1, we have collected all the explicit calculations for the cycle. There are two major parts in this phase, the first concerned with calculating new values of scalar variables defined at cell centers, and the second with calculating new values of the cell-edge momenta. Performance of the first part is the responsibility of a sweep over all the interior cells of the mesh, which solves the equations for ${}^{n+1}\left(\rho'_d\right)^j_i$, ${}^{n+1}\left(\rho'_{d1}\right)^j_i$, ${}^{n+1}\theta^j_i$, A^j_i, $\tilde{p}^j_i$, K^j_i, ${}^{n+1}I_v$, and ${}^{n+1}I_d$. In our notation, the superscript "n" indicates the old value of a quantity at the beginning of the cycle, and "n+1" indicates the new value at the end of the cycle. Immediately at the beginning of a cycle, we can calculate the final values in that cycle of all these quantities, except the pressure $\tilde{p}$, as they are not subject to further modification in Phase 2. The subsidiary quantity A^j_i has the dimensions of an internal energy I_v; it appears in Phase 1 to supply the energy term for the equation-of-state pressure for the vapor, as will be described below. It is subscripted and saved for later use as a necessary coefficient in Phase 2.

The droplet density equations are

$$
\begin{aligned}
{}^{n+1}\left(\rho'_d\right)^j_i &= {}^{n}\left(\rho'_d\right)^j_i \\
&+ \frac{\delta t}{r_i\,\delta r}\left[{}^{n}\left\langle u_d\, r\rho'_d\right\rangle^j_{i-\frac{1}{2}} - {}^{n}\left\langle u_d\, r\rho'_d\right\rangle^j_{i+\frac{1}{2}}\right] \\
&+ \frac{\delta t}{\delta z}\left[{}^{n}\left\langle v_d\, \rho'_d\right\rangle^{j-\frac{1}{2}}_i - {}^{n}\left\langle v_d\, \rho'_d\right\rangle^{j+\frac{1}{2}}_i\right] ,
\end{aligned}
$$

and

$$
\begin{aligned}
{}^{n+1}\left(\rho'_{d1}\right)^j_i &= {}^{n}\left(\rho'_{d1}\right)^j_i \\
&+ \frac{\delta t}{r_i\,\delta r}\left[{}^{n}\left\langle u_d\, r\rho'_{d1}\right\rangle^j_{i-\frac{1}{2}} - {}^{n}\left\langle u_d\, r\rho'_{d1}\right\rangle^j_{i+\frac{1}{2}}\right] \\
&+ \frac{\delta t}{\delta z}\left[{}^{n}\left\langle v_d\, \rho'_{d1}\right\rangle^{j-\frac{1}{2}}_i - {}^{n}\left\langle v_d\, \rho'_{d1}\right\rangle^{j+\frac{1}{2}}_i\right] ,
\end{aligned}
$$

where the angular brackets $\langle\ \rangle$ indicate a partial donor-cell treatment of the convective flux of the enclosed quantity. The use of the donor-cell differencing facilitates automatic mitigation of truncation-error effects[8] without requiring an explicit artificial diffusion. According to our formulation, the convective flux of some cell-centered quantity, Q^j_i, at cell boundary $(i+\frac{1}{2}, j)$ would be given by

$$
\left\langle urQ\right\rangle^j_{i+\frac{1}{2}} = (ur)^j_{i+\frac{1}{2}}\left[\left(\tfrac{1}{2}+\xi\right)Q^j_i + \left(\tfrac{1}{2}-\xi\right)Q^j_{i+1}\right] .
$$

where

$$
\xi = \beta_o\left(\frac{u^j_{i+\frac{1}{2}}\,\delta t}{\delta r}\right) + \alpha_o \ \ \mathrm{sign}\left(u^j_{i+\frac{1}{2}}\right) ,
$$

and α_o and β_o are input coefficients. For $\alpha_o = 0$ and $\beta_o = \frac{1}{2}$, ξ will give an automatically interpolated donor-cell form. More commonly, however, we use

$\beta_o = 0$ and $0 < \alpha_o \leq \frac{1}{2}$. With $\beta_o = 0$, use of $\alpha_o = 0$ would give pure space-centered differencing, which is numerically unstable in the absence of a mitigating diffusive process, whereas use of $\alpha_o = \frac{1}{2}$ would give pure donor-cell differencing, which offers the greatest smoothing. The formulation of ξ assumes that $|u_{max}|\,\delta t/\delta r < \frac{1}{2}$ everywhere, ordinarily a reasonable upper limit on δt for accuracy.

The above transport equations for ${}^{n+1}\left(\rho'_d\right)^j_i$ and ${}^{n+1}\left(\rho'_{d1}\right)^j_i$ allow us to rigorously conserve the masses of the two components separately, and with their values known, ${}^{n+1}\theta^j_i$ is calculated from

$$ {}^{n+1}\theta^j_i = 1 - \frac{{}^{n+1}\left(\rho'_{d1}\right)^j_i}{\rho_1} - \frac{\left[{}^{n+1}\left(\rho'_d\right)^j_i - {}^{n+1}\left(\rho'_{d1}\right)^j_i\right]}{\rho_2} \; . $$

With ${}^{n+1}\theta^j_i$ now available, we can determine the energy term A^j_i. As an example, assume that we are using the polytropic gas equation of state, as we, in fact, do in this version of KACHINA. In this case, the pressure ordinarily would be given by

$$ p = \left(\gamma_v - 1\right) \rho_v I_v \; , $$

in which γ_v is the ratio of specific heats in the vapor. This basic form, however, is insensitive to variations in the proportions of components within the vapor, and it is unsatisfactory for direct use. To overcome this deficiency, we first define A^j_i, a necessary coefficient for Phase 2 iteration purposes, as

$$ A^j_i = \frac{{}^n\left(I_v\right)^j_i}{{}^{n+1}\theta^j_i} \left[\frac{{}^n\left(\rho'_{v1}\right)^j_i b_{v1} \left(\gamma_1 - 1\right) + {}^n\left(\rho'_{v2}\right)^j_i b_{v2} \left(\gamma_2 - 1\right)}{{}^n\left(\rho'_{v1}\right)^j_i b_{v1} + {}^n\left(\rho'_{v2}\right)^j_i b_{v2}} \right] , $$

in which γ's and the specific heats, b_v's, are specified separately for each of the two components, and are provided in the input data. With A^j_i determined, the KACHINA equation-of-state pressure is then calculated directly as

$$ p_{EOS} = A^j_i \, {}^n\left(\rho'_v\right)^j_i \; . $$

The initializing pressure, $\tilde{p}$, actually stored in Phase 1 for a given cell after the first cycle is some mixture of p_{EOS} and the pressure ${}^n p$ left over from the Phase 2 pressure iteration of the previous cycle. This treatment is required to allow the code to account for large variations in the flow Mach number from place to place or as a function of time. We use a function f of the local flow Mach number M to determine the exact proportions of this mixture, where

$$ \tilde{p} = f \, {}^n p + (1 - f)\, p_{EOS} \; , $$

in which the desired limits are $f = 1$ when $M \to 0$ and $f = 0$ when $M \to \infty$. We have found that relating f to the *square* of the local Mach number,

$$ f = \frac{1}{1 + 10\left(\frac{M}{M_o}\right)^2} \; , $$

gives satisfactory results, with $M_o = 0.5$ and the coefficient 10 chosen to allow the equation of state to exert a strong enough influence. The Mach number is calculated using the standard $M = u/c$ form appropriate for a polytropic gas.[9] Choosing a squared Mach number function makes $(1 - f)$ vary inversely in proportion to the vapor internal energy as that quantity becomes large, consistent with the dependence of p_{EOS} on that same variable. The choice also allows us to avoid taking a square root in the calculation of the sound speed, c. For calculations in which the sound speed becomes very large, the exponent should be greater than 2.0, in order that the equation-of-state effects disappear in that limit. Thus we have

$$ \left(\frac{M}{M_o}\right)^2 = \frac{U^2}{c^2 M_o^2} \; , $$

where the square of an appropriate average velocity for the cell (i,j) is calculated as

$$ U^2 = \frac{1}{4}\left\{\left[{}^n\left(u_v\right)^j_{i+\frac{1}{2}}\right]^2 + \left[{}^n\left(u_v\right)^j_{i-\frac{1}{2}}\right]^2\right. $$

$$+\left[{}^{n}\left(v_v\right)_i^{j+\frac{1}{2}}\right]^2+\left[{}^{n}\left(v_v\right)_i^{j-\frac{1}{2}}\right]^2\Bigg\} \quad ,$$

and the square of a sufficiently accurate approximation to the sound speed can be shown to be

---

$$c^2=\frac{{}^{n}I_v\left\{\gamma_{v1}\left(\gamma_{v1}-1\right){}^{n}\left(\frac{\rho'_{v1}}{\rho'_v}\right)+\gamma_{v2}\left(\gamma_{v2}-1\right)\left[1-{}^{n}\left(\frac{\rho'_{v1}}{\rho'_v}\right)\right]\right\}}{\left({}^{n+1}\theta_i^j\right)^2\left[\dfrac{{}^{n}\rho'_d}{{}^{n}\rho'_v}+1\right]} \; .$$

To allow greater generality, it is desirable to replace U^2 by $\left(U^2+p/\rho\right)$, in which p/ρ is given by ${}^{n}p/\rho_v={}^{n}p\theta/\rho'_v$. A flow starting from rest and driven by its own internal pressure will have an initial p/ρ comparable to c^2, and later will develop U^2 comparable to c^2, thus tending to make M^2 always comparable to unity. This formulation also automatically handles far subsonic flows, as in such cases, p/ρ values and U^2 values will be much smaller than c^2, resulting in a small M^2 and f of order unity.

The initializing vapor density, $\tilde{\rho}'_v$, for the cell is also stored at this time. It is based similarly on f,

$$\tilde{\rho}'_v=f\,\theta\rho_{vo}+(1-f)\,\frac{\tilde{p}}{A} \; ,$$

where ρ_{vo} may be a specified constant for completely incompressible flow, or may be allowed to vary in case of buoyancy effects.

The next consideration is the drag function, K_i^j, which we calculate as

$$K_i^j=\frac{3\;{}^{n}\left(\rho'_v\right)_i^j\left(1-{}^{n+1}\theta_i^j\right)}{2\left({}^{n+1}\theta_i^j\right)^2 r_p{}^2}\left[3\nu_v+\left(\frac{r_p c_{DR}}{4}\right)\left|\vec{u}_v-\vec{u}_d\right|\right] ,$$

where

$$\left|\vec{u}_v-\vec{u}_d\right|\equiv\frac{1}{2}\Bigg\{\left[{}^{n}\left(u_v\right)_{i+\frac{1}{2}}^j-{}^{n}\left(u_d\right)_{i+\frac{1}{2}}^j+{}^{n}\left(u_v\right)_{i-\frac{1}{2}}^j-{}^{n}\left(u_d\right)_{i-\frac{1}{2}}^j\right]^2$$

$$+\left[{}^{n}\left(v_v\right)_i^{j+\frac{1}{2}}-{}^{n}\left(v_d\right)_i^{j+\frac{1}{2}}+{}^{n}\left(v_v\right)_i^{j-\frac{1}{2}}-{}^{n}\left(v_d\right)_i^{j-\frac{1}{2}}\right]^2\Bigg\}^{\frac{1}{2}} ,$$

ν_v is the coefficient of kinematic viscosity for the vapor, r_p is the mean linear dimension of a droplet, and c_{DR} is a drag coefficient. In the present version of KACHINA, these three quantities are simply read-in constants. In a future version, r_p will be vastly generalized to include droplet growth or decrease in size, by such processes as evaporation, condensation, rupture, and coalescence. This generalization will be accomplished by means of a transport equation for r_p. At first, the generalization will involve the cell quantity r_i^j, and perhaps, later, an r_p distribution that can vary with position and time. Our present constant value for r_p is a useful first approximation, however, and it allows us to derive information about dependence of the results on the choice of droplet scale.

In theory, the θ_i^j's appearing in the K_i^j equation could be either ${}^{n}\theta$'s or ${}^{n+1}\theta$'s, but our specific choice of ${}^{n+1}\theta$ is based on computational requirements for suddenly incompressible flow. This point will be clarified in Sec. I.F.

The final quantities calculated in the first sweep in Phase 1 are the specific internal energies ${}^{n+1}I_v$ and ${}^{n+1}I_d$.

$${}^{n+1}\left(I_v\right)_i^j={}^{n}\left(I_v\right)_i^j$$

$$+\,\delta t\left\{\frac{1}{r_i\,\delta r}\left[\left\langle u_v I_v r\right\rangle_{i-\frac{1}{2}}^j-\left\langle u_v I_v r\right\rangle_{i+\frac{1}{2}}^j\right]\right.$$

$$\left.+\,\frac{1}{\delta z}\left[\left\langle v_v I_v\right\rangle_i^{j-\frac{1}{2}}-\left\langle v_v I_v\right\rangle_i^{j+\frac{1}{2}}\right]\right\}$$

$$+ \frac{\delta t}{{}^{n}\left(\rho_v'\right)_i^j} \Big\{ R_i^j \Big[\left(T_d\right)_i^j - \left(T_v\right)_i^j \Big]$$

$$+ K_i^j \Big[\left(u_{di}^{\,j} - u_{vi}^{\,j}\right)^2 + \left(v_{di}^{\,j} - v_{vi}^{\,j}\right)^2 \Big]$$

$$+ \frac{k_v}{r_i \delta r^2} \Big[\theta_{i+\frac{1}{2}}^j \, r_{i+\frac{1}{2}} \left(T_{vi+1}^{\,j} - T_{vi}^{\,j}\right)$$

$$- \theta_{i-\frac{1}{2}}^j \, r_{i-\frac{1}{2}} \left(T_{vi}^{\,j} - T_{vj-1}^{\,j}\right) \Big]$$

$$+ \frac{k_v}{\delta z^2} \Big[\theta_i^{j+\frac{1}{2}} \left(T_{vi}^{\,j+1} - T_{vi}^{\,j}\right)$$

$$- \theta_i^{j-\frac{1}{2}} \left(T_{vi}^{\,j} - T_{vi}^{\,j-1}\right) \Big] \Big\}$$

$$- \frac{\tilde{p}_i^j \delta t}{{}^{n}\left(\rho_v'\right)_i^j} \Bigg(\frac{1}{r_i \delta r} \Big\{ r_{i+\frac{1}{2}} \Big[\theta u_v + (1-\theta) u_d \Big]_{i+\frac{1}{2}}^j$$

$$- r_{i-\frac{1}{2}} \Big[\theta u_v + (1-\theta) u_d \Big]_{i-\frac{1}{2}}^j \Big\}$$

$$+ \frac{1}{\delta z} \Big\{ \Big[\theta v_v + (1-\theta) v_d \Big]_i^{j+\frac{1}{2}}$$

$$- \Big[\theta v_v + (1-\theta) v_d \Big]_i^{j-\frac{1}{2}} \Big\} \Bigg)$$

$$+ {}^{n}\left(I_v\right)_i^j \, \delta t \Big\{ \frac{1}{r_i \delta r} \Big[u_{vi+\frac{1}{2}}^{\,j} \, r_{i+\frac{1}{2}} - u_{vi-\frac{1}{2}}^{\,j} \, r_{i-\frac{1}{2}} \Big]$$

$$+ \frac{1}{\delta z} \Big[v_{vi}^{\,j+\frac{1}{2}} - v_{vi}^{\,j-\frac{1}{2}} \Big] \Big\} \quad ,$$

and

$${}^{n+1}\left(I_d\right)_i^j = {}^{n}\left(I_d\right)_i^j$$

$$+ \delta t \Big\{ \frac{1}{r_i \delta r} \Big[\left\langle u_d I_d r \right\rangle_{i-\frac{1}{2}}^j - \left\langle u_d I_d r \right\rangle_{i+\frac{1}{2}}^j \Big]$$

$$+ \frac{1}{\delta z} \Big[\left\langle v_d I_d \right\rangle_i^{j-\frac{1}{2}} - \left\langle v_d I_d \right\rangle_i^{j+\frac{1}{2}} \Big] \Big\}$$

$$+ \frac{\delta t}{{}^{n}\left(\rho_d'\right)_i^j} \Big\{ R_i^j \Big[\left(T_v\right)_i^j - \left(T_d\right)_i^j \Big] + E_i^j$$

$$+ \frac{k_d}{r_i \delta r^2} \Big[\left(1 - \theta_{i+\frac{1}{2}}^j\right) r_{i+\frac{1}{2}} \left(T_{di+1}^{\,j} - T_{di}^{\,j}\right)$$

$$- \left(1 - \theta_{i-\frac{1}{2}}^j\right) r_{i-\frac{1}{2}} \left(T_{di}^{\,j} - T_{di-1}^{\,j}\right) \Big]$$

$$+ \frac{k_d}{\delta z^2} \Big[\left(1 - \theta_i^{j+\frac{1}{2}}\right) \left(T_{di}^{\,j+1} - T_{di}^{\,j}\right)$$

$$- \left(1 - \theta_i^{j-\frac{1}{2}}\right) \left(T_{di}^{\,j} - T_{di}^{\,j-1}\right) \Big] \Big\}$$

$$+ {}^{n}\left(I_d\right)_i^j \, \delta t \Big\{ \frac{1}{r_i \delta r} \Big[u_{di+\frac{1}{2}}^{\,j} \, r_{i+\frac{1}{2}} - u_{di-\frac{1}{2}}^{\,j} \, r_{i-\frac{1}{2}} \Big]$$

$$+ \frac{1}{\delta z} \Big[v_{di}^{\,j+\frac{1}{2}} - v_{di}^{\,j-\frac{1}{2}} \Big] \Big\} \quad .$$

Time level n is assumed for all quantities in the right-hand sides of both of the above equations, except for K_i^j and $\tilde{p}_i^j$, which are coefficients for the drag and work terms, respectively, in ${}^{n+1}\left(I_v\right)_i^j$.

These two quantities are those just obtained, as described above. Several new quantities appear in ${}^{n+1}\left(I_v\right)_i^j$ and ${}^{n+1}\left(I_d\right)_i^j$, and they require discussion.

R_i^j is an exchange function that controls the heat transferred between the two fields per unit volume per unit time, as a result of surface conduction. In general, R_i^j will be a cell variable, although now it is a constant and no array is stored. It is a coefficient that multiplies the local temperature difference, in which the temperature T is now given simply by the I = bT relationship, and is calculated as

$${}^{n}\left(T_v\right)_i^j = \frac{{}^{n}\left(I_v \rho_v'\right)_i^j}{{}^{n}\left(\rho_{v1}'\right)_i^j \, b_{v1} + {}^{n}\left(\rho_{v2}'\right)_i^j \, b_{v2}}$$

and

$$ {}^{n}\left(T_d\right)_i^j = \frac{{}^{n}\left(I_d \rho'_d\right)_i^j}{{}^{n}\left(\rho'_{d1}\right)_i^j b_{d1} + {}^{n}\left(\rho'_{d2}\right)_i^j b_{d2}} . $$

Because the four specific heat coefficients that appear are presently constants, we are relieved of having to store the two T arrays. The constants k_v and k_d are heat conduction coefficients. E_i^j represents the energy contribution from some optional heat source, such as chemical or nuclear processes.

The velocities appearing in the drag term in the ${}^{n+1}\left(I_v\right)_i^j$ equation are calculated as

$$\left(u_{di}^j - u_{vi}^j\right)^2 \equiv \left[\frac{1}{2}\left(u^j_{di+\frac{1}{2}} + u^j_{di-\frac{1}{2}} - u^j_{vi+\frac{1}{2}} - u^j_{vi-\frac{1}{2}}\right)\right]^2$$

and

$$\left(v_{di}^j - v_{vi}^j\right)^2 \equiv \left[\frac{1}{2}\left(v_{di}^{j+\frac{1}{2}} + v_{di}^{j-\frac{1}{2}} - v_{vi}^{j+\frac{1}{2}} - v_{vi}^{j-\frac{1}{2}}\right)\right]^2 .$$

When it becomes available, the new ${}^{n+1}I_v$ value is used to adjust the A for use in Phase 2,

$$A_i^j = A_i^j \, {}^{n+1}\left(I_v\right)_i^j \Big/ {}^{n}\left(I_v\right)_i^j .$$

The second major pass through the mesh in Phase 1 is concerned with calculating a set of momentum fluxes, comprised of the four arrays $\overline{\left(\rho'_v u_v\right)}$, $\overline{\left(\rho'_v v_v\right)}$, $\overline{\left(\rho'_d u_d\right)}$, and $\overline{\left(\rho'_d v_d\right)}$. These fluxes are defined at the same cell-edge positions as the velocities u_v, v_v, u_d, and v_d, respectively. While dealing with momentum, it is convenient simply to replace each velocity in computer storage by the corresponding momentum, which will be reconverted to a velocity in Phase 2. The four equations used are:

$$\begin{aligned}
\overline{\left(\rho'_v u_v\right)}^j_{i+\frac{1}{2}} = {} & {}^{n}\left(\rho'_v u_v\right)^j_{i+\frac{1}{2}} \\
& + \frac{\delta t}{r_{i+\frac{1}{2}}\delta r}\left[{}^{n}\left\langle \rho'_v u_v^{\,2} r\right\rangle^j_i - {}^{n}\left\langle \rho'_v u_v^{\,2} r\right\rangle^j_{i+1}\right] \\
& + \delta t \, {}^{n}\left(F_{vr}\right)^j_{i+\frac{1}{2}} \\
& + \frac{\delta t}{\delta z}\left[{}^{n}\left\langle \rho'_v u_v v_v\right\rangle^{j-\frac{1}{2}}_{i+\frac{1}{2}} - {}^{n}\left\langle \rho'_v u_v v_v\right\rangle^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right] ,
\end{aligned}$$

$$\begin{aligned}
\overline{\left(\rho'_v v_v\right)}^{j+\frac{1}{2}}_{i} = {} & {}^{n}\left(\rho'_v v_v\right)^{j+\frac{1}{2}}_{i} \\
& + \frac{\delta t}{r_i\delta r}\left[{}^{n}\left\langle \rho'_v u_v v_v r\right\rangle^{j+\frac{1}{2}}_{i-\frac{1}{2}} - {}^{n}\left\langle \rho'_v u_v v_v r\right\rangle^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right] \\
& + \delta t \, {}^{n}\left(F_{vz}\right)^{j+\frac{1}{2}}_{i} \\
& + \frac{\delta t}{\delta z}\left[{}^{n}\left\langle \rho'_v v_v^{\,2}\right\rangle^{j}_{i} - \left\langle \rho'_v v_v^{\,2}\right\rangle^{j+1}_{i}\right] + \left(\rho'_v\right)^{j+\frac{1}{2}}_i g\delta t ,
\end{aligned}$$

$$\begin{aligned}
\overline{\left(\rho'_d u_d\right)}^j_{i+\frac{1}{2}} = {} & {}^{n}\left(\rho'_d u_d\right)^j_{i+\frac{1}{2}} \\
& + \frac{\delta t}{r_{i+\frac{1}{2}}\delta r}\left[{}^{n}\left\langle \rho'_d u_d^{\,2} r\right\rangle^j_i - {}^{n}\left\langle \rho'_d u_d^{\,2} r\right\rangle^j_{i+1}\right] \\
& + \delta t \, {}^{n}\left(F_{dr}\right)^j_{i+\frac{1}{2}} \\
& + \frac{\delta t}{\delta z}\left[{}^{n}\left\langle \rho'_d u_d v_d\right\rangle^{j-\frac{1}{2}}_{i+\frac{1}{2}} - {}^{n}\left\langle \rho'_d u_d v_d\right\rangle^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right] ,
\end{aligned}$$

and

$$\begin{aligned}
\overline{\left(\rho'_d v_d\right)}^{j+\frac{1}{2}}_{i} = {} & {}^{n}\left(\rho'_d v_d\right)^{j+\frac{1}{2}}_{i} \\
& + \frac{\delta t}{r_i\delta r}\left[{}^{n}\left\langle \rho'_d u_d v_d r\right\rangle^{j+\frac{1}{2}}_{i-\frac{1}{2}} - {}^{n}\left\langle \rho'_d u_d v_d r\right\rangle^{j+\frac{1}{2}}_{i+\frac{1}{2}}\right] \\
& + \delta t \, {}^{n}\left(F_{dz}\right)^{j+\frac{1}{2}}_{i} \\
& + \frac{\delta t}{\delta z}\left[{}^{n}\left\langle \rho'_d v_d^{\,2}\right\rangle^{j}_{i} - {}^{n}\left\langle \rho'_d v_d^{\,2}\right\rangle^{j+1}_{i}\right] + \left(\rho'_d\right)^{j+\frac{1}{2}}_i g\delta t .
\end{aligned}$$

The donor-cell formulations in the convective flux terms are analogous to those previously described for the quantities in the first sweep, but, because the centerings of the variables involved are different, we include several representative examples. In the $\overline{\rho u}$ equations:

$$\langle \rho u^2 r \rangle_i^j = (ur)_i^j \left[(\tfrac{1}{2} + \xi)(\rho u)_{i-\frac{1}{2}}^j + (\tfrac{1}{2} - \xi)(\rho u)_{i+\frac{1}{2}}^j \right] ,$$

where

$$\xi = \beta_o \left(\frac{u_i^j \delta t}{\delta r} \right) + \alpha_o \text{ sign } \left(u_i^j \right) ,$$

and

$$u_i^j = \tfrac{1}{2} \left(u_{i+\frac{1}{2}}^j + u_{i-\frac{1}{2}}^j \right) .$$

$$\langle \rho uv \rangle_{i+\frac{1}{2}}^{j-\frac{1}{2}} = v_{i+\frac{1}{2}}^{j-\frac{1}{2}} \left[(\tfrac{1}{2} + \xi)(\rho u)_{i+\frac{1}{2}}^{j-1} + (\tfrac{1}{2} - \xi)(\rho u)_{i+\frac{1}{2}}^j \right] ,$$

where

$$\xi = \beta_o \left(\frac{v_{i+\frac{1}{2}}^{j-\frac{1}{2}} \delta t}{\delta z} \right) + \alpha_o \text{ sign } \left(v_{i+\frac{1}{2}}^{j-\frac{1}{2}} \right) ,$$

and

$$v_{i+\frac{1}{2}}^{j-\frac{1}{2}} = \tfrac{1}{2} \left(v_i^{j-\frac{1}{2}} + v_{i+1}^{j-\frac{1}{2}} \right) .$$

In the $\overline{\rho v}$ equations:

$$\langle \rho uvr \rangle_{i-\frac{1}{2}}^{j+\frac{1}{2}} = u_{i-\frac{1}{2}}^{j+\frac{1}{2}} \, r_{i-\frac{1}{2}} \left[(\tfrac{1}{2} + \xi)(\rho v)_{i-1}^{j+\frac{1}{2}} + (\tfrac{1}{2} - \xi)(\rho v)_i^{j+\frac{1}{2}} \right] ,$$

where

$$\xi = \beta_o \left(\frac{u_{i-\frac{1}{2}}^{j+\frac{1}{2}} \delta t}{\delta r} \right) + \alpha_o \text{ sign } \left(u_{i-\frac{1}{2}}^{j+\frac{1}{2}} \right) ,$$

and

$$u_{i-\frac{1}{2}}^{j+\frac{1}{2}} = \tfrac{1}{2} \left(u_{i-\frac{1}{2}}^j + u_{i-\frac{1}{2}}^{j+1} \right) .$$

In addition to the convective flux terms, the $\overline{\rho v}$ equations contain terms for including the effect of gravitational acceleration, with g being constant.

In many KACHINA applications, the flows to be studied will involve interaction with some confining and at least partially nonfailing structural elements, which may not be practical to include in the computer model in complete detail. Such interactions may be significantly nonisotropic in orientation. To properly incorporate their effects will require interaction terms with material strength effects capable of representing added inertia, nonisotropic drag, energy dissipation, and elastic-plastic deformation. In the present KACHINA version, the F_r and F_z terms in the momentum equations represent a simple preliminary approach to including nonisotropic effects. These terms contribute a dissipationless turning that tends to constrict motion primarily to the axial direction. The direction of the force is orthogonal to the velocity, and the strength is proportional to the departure of the velocity from the axial direction. Thus, if

$$\vec{u} = \hat{i}u + \hat{j}v \quad ,$$

and

$$\vec{F} = \hat{i}a + \hat{j}b \quad ,$$

then

$$au + bv = 0 \quad ,$$

and

$$a^2 + b^2 = \varepsilon^2 u^2 / \left(u^2 + v^2 \right) \quad ,$$

with magnitude proportional to $|u|$. Therefore,

$$a = \pm \varepsilon uv / \left(u^2 + v^2 \right) \quad ,$$

and

$$b = \mp \varepsilon u^2 / \left(u^2 + v^2 \right) \quad ,$$

or

$$a = - \varepsilon uv \text{ sign } (v) / \left(u^2 + v^2 \right) \quad ,$$

and

$$b = + \varepsilon u^2 \text{ sign } (v) / \left(u^2 + v^2 \right) \quad .$$

Because v sign (v) is always positive ($\equiv |v|$), a is always directed against u, whereas b works to increase positive v or to decrease negative v. Our form for the nonisotropic term in this version of KACHINA is

$$\vec{F} = - \frac{\varepsilon_\ell u_\ell \text{ sign } (v_\ell)}{u_\ell^2 + v_\ell^2} \left(\hat{i} v_\ell - \hat{j} u_\ell\right) ,$$

where ℓ = d or v, thereby requiring a different ε for the term in a droplet equation from that in a vapor equation. With $F_r = a$ and $F_z = b$, the equations used are:

$${}^n\left(F_{vr}\right)_{i+\frac{1}{2}}^{j} = - \frac{\varepsilon_v \, {}^n\left[u_v v_v \text{ sign } \left(v_v\right)\right]_{i+\frac{1}{2}}^{j}}{{}^n\left(u_v^2 + v_v^2\right)_{i+\frac{1}{2}}^{j}} ,$$

$${}^n\left(F_{dr}\right)_{i+\frac{1}{2}}^{j} = - \frac{\varepsilon_d \, {}^n\left[u_d v_d \text{ sign } \left(v_d\right)\right]_{i+\frac{1}{2}}^{j}}{{}^n\left(u_d^2 + v_d^2\right)_{i+\frac{1}{2}}^{j}} ,$$

$${}^n\left(F_{vz}\right)_{i}^{j+\frac{1}{2}} = + \frac{\varepsilon_v \, {}^n\left[u_v^2 \text{ sign } \left(v_v\right)\right]_{i}^{j+\frac{1}{2}}}{{}^n\left(u_v^2 + v_v^2\right)_{i}^{j+\frac{1}{2}}} ,$$

and

$${}^n\left(F_{dz}\right)_{i}^{j+\frac{1}{2}} = + \frac{\varepsilon_d \, {}^n\left[u_d^2 \text{ sign } \left(v_d\right)\right]_{i}^{j+\frac{1}{2}}}{{}^n\left(u_d^2 + v_d^2\right)_{i}^{j+\frac{1}{2}}} .$$

Generally, $\varepsilon_\ell(r, z, t)$ may be prescribed in its r-z variations, and ε_ℓ may decay during the course of a calculation, representing the loss of integrity of the confining structure. At present, however, we simply specify ε_v and ε_d as constants. Note that if ε_v or ε_d is large, the corresponding term may require an implicit treatment to ensure that all motions are constrained to the desired trajectory.

Further, this nonisotropic force cannot be incorporated into the drag term, because it includes the effects of droplets colliding with other droplets that are constrained into axial channels by material strength, and such collisions are precluded by our two-field approach.

Note that this version of the momentum equations omits all viscous terms because they are required neither for numerical stability (because of our partial donor-cell convection treatment), nor for the physical processes that we wish to represent in this initial version of the code. Their later inclusion will be accomplished entirely by addition of the appropriate stress terms to those equations defining the $\overline{\rho u}$ and $\overline{\rho v}$ quantities.[7]

This completes the explicit part of the calculation cycle. In summary, at the end of Phase 1, we have in computer storage the n+1 values of ρ_d', ρ_{dl}', θ, I_v, and I_d, along with the values of ${}^n\rho_v'$, ${}^n\rho_{vl}'$, A, $\tilde{p}$, K, $(\overline{\rho_v' u_v})$, $(\overline{\rho_v' v_v})$, $(\overline{\rho_d' u_d})$, and $(\overline{\rho_d' v_d})$.

D. Phase 2 — Implicit Calculations

The Phase 2 calculations start with conversion of the momenta from Phase 1 back into velocity components $\tilde{u}_v$, $\tilde{v}_v$, $\tilde{u}_d$, and $\tilde{v}_d$. These are the tentative final velocities for the cycle, as symbolized by the tildes, and they include the first-guess effects of pressure acceleration and drag, both of which will be corrected as the tilde velocities converge to their final values for the cycle. The velocities are given by the (explicit) expressions:

$$\left(\tilde{u}_v\right)_{i+\frac{1}{2}}^{j} = \frac{{}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^{j} \not{C}_{i+\frac{1}{2}}^{j} + \delta t \; {}^{n}K_{i+\frac{1}{2}}^{j} \left(\not{U}_{i+\frac{1}{2}}^{j} + \not{C}_{i+\frac{1}{2}}^{j}\right)}{{}^{n}\rho_{vi+\frac{1}{2}}'^{\,j} \left[{}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^{j} + \delta t \; {}^{n}K_{i+\frac{1}{2}}^{j}\right] + \delta t \; {}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^{j} \; {}^{n}K_{i+\frac{1}{2}}^{j}} ,$$

$$\left(\tilde{v}_v\right)_{i}^{j+\frac{1}{2}} = \frac{{}^{n+1}\left(\rho_d'\right)_{i}^{j+\frac{1}{2}} \not{V}_{i}^{j+\frac{1}{2}} + \delta t \; {}^{n}K_{i}^{j+\frac{1}{2}} \left(\not{U}_{i}^{j+\frac{1}{2}} + \not{V}_{i}^{j+\frac{1}{2}}\right)}{{}^{n}\rho_{vi}'^{\,j+\frac{1}{2}} \left[{}^{n+1}\left(\rho_d'\right)_{i}^{j+\frac{1}{2}} + \delta t \; {}^{n}K_{i}^{j+\frac{1}{2}}\right] + \delta t \; {}^{n+1}\left(\rho_d'\right)_{i}^{j+\frac{1}{2}} \; {}^{n}K_{i}^{j+\frac{1}{2}}} ,$$

$$\left(\tilde{u}_d\right)^j_{i+\frac{1}{2}} = \frac{{}^n\left(\rho'_v\right)^j_{i+\frac{1}{2}} \not{B}^j_{i+\frac{1}{2}} + \delta t \; {}^nK^j_{i+\frac{1}{2}} \left(\not{B}^j_{i+\frac{1}{2}} + \not{C}^j_{i+\frac{1}{2}}\right)}{{}^n\rho'^{\,j}_{v i+\frac{1}{2}} \left[{}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}} + \delta t \; {}^nK^j_{i+\frac{1}{2}}\right] + \delta t \; {}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}} \; {}^nK^j_{i+\frac{1}{2}}} \quad ,$$

$$\left(\tilde{v}_d\right)^{j+\frac{1}{2}}_i = \frac{{}^n\left(\rho'_v\right)^{j+\frac{1}{2}}_i \not{D}^{j+\frac{1}{2}}_i + \delta t \; {}^nK^{j+\frac{1}{2}}_i \left(\not{D}^{j+\frac{1}{2}}_i + \not{E}^{j+\frac{1}{2}}_i\right)}{{}^n\rho'^{\,j+\frac{1}{2}}_{vi} \left[{}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i + \delta t \; {}^nK^{j+\frac{1}{2}}_i\right] + \delta t \; {}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i \; {}^nK^{j+\frac{1}{2}}_i} \quad ,$$

where:

$$\not{B}^j_{i+\frac{1}{2}} = \overline{\left(\rho'_d u_d\right)}^j_{i+\frac{1}{2}} + \left(1 - {}^{n+1}\theta^j_{i+\frac{1}{2}}\right) \frac{\delta t}{\delta r} \left(\tilde{p}^j_i - \tilde{p}^j_{i+1}\right) ,$$

$$\not{C}^j_{i+\frac{1}{2}} = \overline{\left(\rho'_v u_v\right)}^j_{i+\frac{1}{2}} + {}^{n+1}\theta^j_{i+\frac{1}{2}} \frac{\delta t}{\delta r} \left(\tilde{p}^j_i - \tilde{p}^j_{i+1}\right) ,$$

$$\not{D}^{j+\frac{1}{2}}_i = \overline{\left(\rho'_d v_d\right)}^{j+\frac{1}{2}}_i + \left(1 - {}^{n+1}\theta^{j+\frac{1}{2}}_i\right) \frac{\delta t}{\delta z} \left(\tilde{p}^j_i - \tilde{p}^{j+1}_i\right) \quad ,$$

$$\not{E}^{j+\frac{1}{2}}_i = \overline{\left(\rho'_v v_v\right)}^{j+\frac{1}{2}}_i + {}^{n+1}\theta^{j+\frac{1}{2}}_i \frac{\delta t}{\delta z} \left(\tilde{p}^j_i - \tilde{p}^{j+1}_i\right).$$

An implicit treatment is now required to help increase computational stability, in particular to eliminate the usual Courant-like restriction on high sound speed, by allowing signals to traverse more than one cell per time step. This is accomplished by iterating the quantities $\tilde{p}$, $\tilde{\rho}'_v$, $\tilde{u}_v$, $\tilde{v}_v$, $\tilde{u}_d$, and $\tilde{v}_d$ to obtain new velocities that have been accelerated with time-advanced pressure gradients. The new velocities depend on the new pressures and densities, which, in turn, depend on the velocities; therefore the technique is implicit. It is best solved by an iterative process to provide the generality of initial and boundary conditions that usually are precluded by direct solution techniques.

The final (n+1) values of p are accumulated from the tilde values, in which each iteration contributes an increment to $\tilde{p}$, designated by δp.

The source term for the iteration is labeled $\tilde{D}$; it is composed of a $\partial\rho'_v/\partial t$ term plus a $\nabla \cdot (\rho'\vec{u})$ term, both of which use the most recently updated tilde values:

$$\tilde{D}^j_i = \frac{1}{\delta t}\left[\left(\tilde{\rho}'_v\right)^j_i - {}^n\left(\rho'_v\right)^j_i\right]$$

$$+ \frac{1}{r_i \delta r}\left[\left\langle\tilde{\rho}'_v\tilde{u}_v r\right\rangle^j_{i+\frac{1}{2}} - \left\langle\tilde{\rho}'_v\tilde{u}_v r\right\rangle^j_{i-\frac{1}{2}}\right]$$

$$+ \frac{1}{\delta z}\left[\left\langle\tilde{\rho}'_v\tilde{v}_v\right\rangle^{j+\frac{1}{2}}_i - \left\langle\tilde{\rho}'_v\tilde{v}_v\right\rangle^{j-\frac{1}{2}}_i\right] \quad .$$

Note that two levels of the total vapor density ρ'_v must be maintained throughout this iterative process, as indicated in the first term of $\tilde{D}$. The donor-cell formulation of the second and third terms of $\tilde{D}$ is calculated in a manner similar to that used in Phase 1. For example,

$$\left\langle\tilde{\rho}'_v\tilde{u}_v r\right\rangle^j_{i+\frac{1}{2}} = \left(u_v r\right)^j_{i+\frac{1}{2}} \left[\left(\tfrac{1}{2} + \xi\right)\left(\tilde{\rho}'_v\right)^j_i + \left(\tfrac{1}{2} - \xi\right)\left(\tilde{\rho}'_v\right)^j_{i+1}\right] ,$$

in which

$$\xi = \beta_o \left[\frac{\left(\tilde{u}_v\right)^j_{i+\frac{1}{2}} \delta t}{\delta r}\right] + \alpha_o \left[\operatorname{sign}\left(\tilde{u}_v\right)^j_{i+\frac{1}{2}}\right] \quad .$$

The necessary pressure change for the cell is given by

$$\delta p^j_i = - \omega_p \left(\beta_p \tilde{D}\right)^j_i \quad ,$$

in which ω_p is an under- or overrelaxation coefficient of order unity. Straight relaxation is given by $\omega_p = 1$, but, because we are using a relaxation procedure based on Jacobi's method, the iteration will converge only if $0 < \omega_p \leq 1$. Also,

$$\frac{1}{\beta_{pi}^{\ j}} = \frac{\partial D_i^j}{\partial p_i^j} = \frac{1}{A_i^j \delta t} + \frac{\delta t}{r_i \delta r}\left\{\frac{r_{i+\frac{1}{2}}\ {}^{n+1}\theta_{i+\frac{1}{2}}^j + r_{i-\frac{1}{2}}\ {}^{n+1}\theta_{i-\frac{1}{2}}^j}{\delta r} + r_{i+\frac{1}{2}}\ K_{i+\frac{1}{2}}^j \left[\frac{\partial \left(u_d\right)_{i+\frac{1}{2}}^j}{\partial p_i^j} - \frac{\partial \left(u_v\right)_{i+\frac{1}{2}}^j}{\partial p_i^j}\right]\right.$$

$$\left. - r_{i-\frac{1}{2}}\ K_{i-\frac{1}{2}}^j \left[\frac{\partial \left(u_d\right)_{i-\frac{1}{2}}^j}{\partial p_i^j} - \frac{\partial \left(u_v\right)_{i-\frac{1}{2}}^j}{\partial p_i^j}\right]\right\} + \frac{\delta t}{\delta z}\left\{\frac{{}^{n+1}\theta_i^{j+\frac{1}{2}} + {}^{n+1}\theta_i^{j-\frac{1}{2}}}{\delta z} + K_i^{j+\frac{1}{2}}\left[\frac{\partial \left(v_d\right)_i^{j+\frac{1}{2}}}{\partial p_i^j} - \frac{\partial \left(v_v\right)_i^{j+\frac{1}{2}}}{\partial p_i^j}\right]\right.$$

$$\left. - K_i^{j-\frac{1}{2}}\left[\frac{\partial \left(v_d\right)_i^{j-\frac{1}{2}}}{\partial p_i^j} - \frac{\partial \left(v_v\right)_i^{j-\frac{1}{2}}}{\partial p_i^j}\right]\right\} \quad ,$$

in which the partial derivatives are calculated by means of the following equations. Note that the second and fourth equations are not obtained merely by index changes from the first and third equations.

$$\left[\frac{\partial \left(u_d\right)_{i+\frac{1}{2}}^j}{\partial p_i^j} - \frac{\partial \left(u_v\right)_{i+\frac{1}{2}}^j}{\partial p_i^j}\right] = + \frac{\tilde{\rho}'^{\,j}_{vi+\frac{1}{2}}\left(1 - {}^{n+1}\theta_{i+\frac{1}{2}}^j\right)\frac{\delta t}{\delta r} - {}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^j\left(\frac{{}^{n+1}\theta_{i+\frac{1}{2}}^j \delta t}{\delta r} - \frac{\left(\tilde{u}_v\right)_{i+\frac{1}{2}}^j}{2A_i^j}\right)}{\tilde{\rho}'^{\,j}_{vi+\frac{1}{2}}\left[{}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^j + \delta t\ {}^{n}K_{i+\frac{1}{2}}^j\right] + \delta t\ {}^{n+1}\left(\rho_d'\right)_{i+\frac{1}{2}}^j\ {}^{n}K_{i+\frac{1}{2}}^j} \quad ,$$

$$\left[\frac{\partial \left(u_d\right)_{i-\frac{1}{2}}^j}{\partial p_i^j} - \frac{\partial \left(u_v\right)_{i-\frac{1}{2}}^j}{\partial p_i^j}\right] = - \frac{\tilde{\rho}'^{\,j}_{vi-\frac{1}{2}}\left(1 - {}^{n+1}\theta_{i-\frac{1}{2}}^j\right)\frac{\delta t}{\delta r} - {}^{n+1}\left(\rho_d'\right)_{i-\frac{1}{2}}^j\left(\frac{{}^{n+1}\theta_{i-\frac{1}{2}}^j \delta t}{\delta r} + \frac{\left(\tilde{u}_v\right)_{i-\frac{1}{2}}^j}{2A_i^j}\right)}{\tilde{\rho}'^{\,j}_{vi-\frac{1}{2}}\left[{}^{n+1}\left(\rho_d'\right)_{i-\frac{1}{2}}^j + \delta t\ {}^{n}K_{i-\frac{1}{2}}^j\right] + \delta t\ {}^{n+1}\left(\rho_d'\right)_{i-\frac{1}{2}}^j\ {}^{n}K_{i-\frac{1}{2}}^j} \quad ,$$

$$\left[\frac{\partial \left(v_d\right)_i^{j+\frac{1}{2}}}{\partial p_i^j} - \frac{\partial \left(v_v\right)_i^{j+\frac{1}{2}}}{\partial p_i^j}\right] = + \frac{\tilde{\rho}'^{\,j+\frac{1}{2}}_{vi}\left(1 - {}^{n+1}\theta_i^{j+\frac{1}{2}}\right)\frac{\delta t}{\delta z} - {}^{n+1}\left(\rho_d'\right)_i^{j+\frac{1}{2}}\left(\frac{{}^{n+1}\theta_i^{j+\frac{1}{2}} \delta t}{\delta z} - \frac{\left(\tilde{v}_v\right)_i^{j+\frac{1}{2}}}{2A_i^j}\right)}{\tilde{\rho}'^{\,j+\frac{1}{2}}_{vi}\left[{}^{n+1}\left(\rho_d'\right)_i^{j+\frac{1}{2}} + \delta t\ {}^{n}K_i^{j+\frac{1}{2}}\right] + \delta t\ {}^{n+1}\left(\rho_d'\right)_i^{j+\frac{1}{2}}\ {}^{n}K_i^{j+\frac{1}{2}}} \quad ,$$

$$\left[\frac{\partial \left(v_d\right)_i^{j-\frac{1}{2}}}{\partial p_i^j} - \frac{\partial \left(v_v\right)_i^{j-\frac{1}{2}}}{\partial p_i^j}\right] = - \frac{\tilde{\rho}'^{\,j-\frac{1}{2}}_{vi}\left(1 - {}^{n+1}\theta_i^{j-\frac{1}{2}}\right)\frac{\delta t}{\delta z} - {}^{n+1}\left(\rho_d'\right)_i^{j-\frac{1}{2}}\left(\frac{{}^{n+1}\theta_i^{j-\frac{1}{2}} \delta t}{\delta z} + \frac{\left(\tilde{v}_v\right)_i^{j-\frac{1}{2}}}{2A_i^j}\right)}{\tilde{\rho}'^{\,j-\frac{1}{2}}_{vi}\left[{}^{n+1}\left(\rho_d'\right)_i^{j-\frac{1}{2}} + \delta t\ {}^{n}K_i^{j-\frac{1}{2}}\right] + \delta t\ {}^{n+1}\left(\rho_d'\right)_i^{j-\frac{1}{2}}\ {}^{n}K_i^{j-\frac{1}{2}}} \quad .$$

We have found that it is sufficient to calculate an array of β_p's for all cells and store them before entering the iterations and hold them invariant throughout the iterations to enhance computer efficiency.

With δp calculated, the next step is to update $\tilde{p}$ and $\tilde{\rho}_v'$.

$${}^{new}(\tilde{p})_i^j = {}^{old}(\tilde{p})_i^j + \delta p_i^j \quad ,$$

$${}^{new}\left(\tilde{\rho}_v'\right)_i^j = f_i^j\ \theta_i^j\ \rho_{vo} + \left(1 - f_i^j\right)\frac{{}^{new}(\tilde{p})_i^j}{A_i^j} \quad ,$$

thus allowing direct calculation of the new velocity values:

$${}^{new}\left(\tilde{u}_v\right)^j_{i+\frac{1}{2}} = \frac{{}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}}\,\mathcal{R}^j_{i+\frac{1}{2}} + \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\left(\mathcal{R}^j_{i+\frac{1}{2}} + \mathcal{U}^j_{i+\frac{1}{2}}\right)}{{}^{new}\left(\tilde{\rho}'_v\right)^j_{i+\frac{1}{2}}\left[{}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}} + \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\right] + \delta t\ {}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}}\ {}^{n}K^j_{i+\frac{1}{2}}} ,$$

$${}^{new}\left(\tilde{u}_d\right)^j_{i+\frac{1}{2}} = \frac{{}^{new}\left(\tilde{\rho}'_v\right)^j_{i+\frac{1}{2}}\,\mathcal{U}^j_{i+\frac{1}{2}} + \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\left(\mathcal{R}^j_{i+\frac{1}{2}} + \mathcal{U}^j_{i+\frac{1}{2}}\right)}{{}^{new}\left(\tilde{\rho}'_v\right)^j_{i+\frac{1}{2}}\left[{}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}} + \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\right] + \delta t\ {}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}}\ {}^{n}K^j_{i+\frac{1}{2}}} ,$$

$${}^{new}\left(\tilde{v}_v\right)^{j+\frac{1}{2}}_i = \frac{{}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i\,\mathcal{S}^{j+\frac{1}{2}}_i + \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\left(\mathcal{S}^{j+\frac{1}{2}}_i + \mathcal{V}^{j+\frac{1}{2}}_i\right)}{{}^{new}\left(\tilde{\rho}'_v\right)^{j+\frac{1}{2}}_i\left[{}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i + \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\right] + \delta t\ {}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i\ {}^{n}K^{j+\frac{1}{2}}_i} ,$$

$${}^{new}\left(\tilde{v}_d\right)^{j+\frac{1}{2}}_i = \frac{{}^{new}\left(\tilde{\rho}'_v\right)^{j+\frac{1}{2}}_i\,\mathcal{V}^{j+\frac{1}{2}}_i + \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\left(\mathcal{S}^{j+\frac{1}{2}}_i + \mathcal{V}^{j+\frac{1}{2}}_i\right)}{{}^{new}\left(\tilde{\rho}'_v\right)^{j+\frac{1}{2}}_i\left[{}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i + \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\right] + \delta t\ {}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i\ {}^{n}K^{j+\frac{1}{2}}_i} ,$$

where

$$\mathcal{R}^j_{i+\frac{1}{2}} = {}^{old}\left(\tilde{\rho}'_v\tilde{u}_v\right)^j_{i+\frac{1}{2}} + \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\ {}^{old}\left(\tilde{u}_v - \tilde{u}_d\right)^j_{i+\frac{1}{2}}$$

$$+ \frac{\delta t\left(\delta p^j_i - \delta p^j_{i+1}\right)}{\delta r}\ {}^{n+1}\theta^j_{i+\frac{1}{2}} ,$$

$$\mathcal{U}^j_{i+\frac{1}{2}} = {}^{n+1}\left(\rho'_d\right)^j_{i+\frac{1}{2}}\ {}^{old}\left(\tilde{u}_d\right)^j_{i+\frac{1}{2}} - \delta t\ {}^{n}K^j_{i+\frac{1}{2}}\ {}^{old}\left(\tilde{u}_v - \tilde{u}_d\right)^j_{i+\frac{1}{2}}$$

$$+ \frac{\delta t\left(\delta p^j_i - \delta p^j_{i+1}\right)}{\delta r}\left(1 - {}^{n+1}\theta^j_{i+\frac{1}{2}}\right) ,$$

$$\mathcal{S}^{j+\frac{1}{2}}_i = {}^{old}\left(\tilde{\rho}'_v\tilde{v}_v\right)^{j+\frac{1}{2}}_i + \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\ {}^{old}\left(\tilde{v}_v - \tilde{v}_d\right)^{j+\frac{1}{2}}_i$$

$$+ \frac{\delta t\left(\delta p^j_i - \delta p^{j+1}_i\right)}{\delta z}\ {}^{n+1}\theta^{j+\frac{1}{2}}_i ,$$

$$\mathcal{V}^{j+\frac{1}{2}}_i = {}^{n+1}\left(\rho'_d\right)^{j+\frac{1}{2}}_i\ {}^{old}\left(\tilde{v}_d\right)^{j+\frac{1}{2}}_i - \delta t\ {}^{n}K^{j+\frac{1}{2}}_i\ {}^{old}\left(\tilde{v}_v - \tilde{v}_d\right)^{j+\frac{1}{2}}_i$$

$$+ \frac{\delta t\left(\delta p^j_i - \delta p^{j+1}_i\right)}{\delta z}\left(1 - {}^{n+1}\theta^{j+\frac{1}{2}}_i\right) .$$

Each iteration consists of two sweeps over the entire mesh; the first sweep provides ${}^{new}\tilde{p}$ for all cells, then the second sweep calculates updated values of $\tilde{\rho}'_v$, $\tilde{u}_v$, $\tilde{u}_d$, $\tilde{v}_v$, and $\tilde{v}_d$ for all cells. This two-sweep-per-iteration procedure is required because the neighboring values of δp^j_{i+1} and δp^{j+1}_i must be available for the $\mathcal{R}$, $\mathcal{U}$, $\mathcal{S}$, and $\mathcal{V}$ equations.

The iteration procedure is repeated until

$$\left|D^j_i\right| \le \epsilon_1\left\{\left[\left(\frac{|{}^{u}v|_R + |{}^{u}v|_L}{2\delta r} + \frac{|{}^{v}v|_T + |{}^{v}v|_B}{2\delta z}\right)\rho'_v\right]_{max} + \epsilon_2\right\}$$

is satisfied for all cells, at which time the fields of $\tilde{p}$, $\tilde{\rho}'_v$, $\tilde{u}_v$, $\tilde{u}_d$, $\tilde{v}_v$, and $\tilde{v}_d$ are considered to have become the final (n+1) values for the cycle. In the above test, the fineness of the convergence is governed by the input constant ϵ_1, which typically ranges from 10^{-4} to 10^{-3}. The ϵ_2 provides a cutoff minimum at the beginning of a problem, when the velocity field has yet to be established; it, too, is problem dependent and has the dimensions of 1/time. We have found that a value $\epsilon_2 = 10^{-6}$ suffices for most problems run on the CDC 6600 or 7600, which carry about 14 or 15 significant digits in floating-point numbers. Thus, if $\epsilon_1 = 10^{-4}$, the convergence requirement with all zero velocities is $(10^{-4})(10^{-6}) = 10^{-10}$, which is certainly within accuracy standards, but if ε_2 were reduced to 10^{-10}, however, the resulting 10^{-14} would border on machine significance, and convergence could not be obtained. The $\left[\;\right]_{max}$ portion of the convergence test above represents the magnitude of the largest initial tilde velocity times ρ'_v product in the entire mesh, found upon an examination of all cells that was performed concurrently with the β_p calculation, back before the iteration began.

In practice, it is wise to specify some maximum allowable number of iterations per cycle, simply terminating the iterative process if this number is ever reached, and considering the current values of $\tilde{p}$, $\tilde{\rho}'_v$, $\tilde{u}_v$, $\tilde{u}_d$, $\tilde{v}_v$, and $\tilde{v}_d$ to be satisfactory. In our test runs, this procedure has worked well with a cutoff of 100 iterations. If the cutoff is encountered because the solution is diverging, obviously the calculation is in serious trouble, but the cutoff will occasionally terminate an iteration that is converging properly. In such instances, computer time is saved by this termination, and the current pressure, density, and velocity values are accurate enough that the iteration can be expected to converge more rapidly in the next cycle. Typical runs encounter this cutoff only very rarely, the usual number of iterations per cycle seldom exceeding 10.

At the end of this iterative solution, we have in computer storage the n+1 values of ρ'_d, ρ'_{d1}, θ, I_v, I_d, ρ'_v, p, u_v, u_d, v_v, and v_d, along with A, K, and ${}^n\rho'_{v1}$. Because the new velocities are now available, we can solve for the one remaining unknown field variable, ${}^{n+1}\rho'_{v1}$. We use a similar Newton-Raphson iteration scheme again, but this time based on the Gauss-Seidel method. The first guess for $\tilde{\rho}'_{v1}$ is simply ${}^n\rho'_{v1}$, and the changes are accumulated from the relationship

$$\delta\left(\tilde{\rho}'_{v1}\right)^j_i = -\,\omega_p\left(\beta_\rho Q\right)^j_i \quad .$$

Here, ω_ρ is a relaxation coefficient lying in the range $0<\omega_\rho<2$. The term β_ρ remains constant throughout this iteration, and is given by

$$\left(\beta_\rho\right)^j_i = \cfrac{1}{\cfrac{1}{\delta t} + \cfrac{1}{2r_i\delta r}\left[{}^{n+1}\left(ru_v\right)^j_{i+\frac{1}{2}} - {}^{n+1}\left(ru_v\right)^j_{i-\frac{1}{2}}\right] + \cfrac{1}{2\delta z}\left[{}^{n+1}\left(v_v\right)^{j+\frac{1}{2}}_i - {}^{n+1}\left(v_v\right)^{j-\frac{1}{2}}_i\right]} \quad .$$

The denominator in the β_ρ equation will not vanish if $|u_v|_{max}\,\delta t/\delta r < 0.5$ and $|v_v|_{max}\,\delta t/\delta z < 0.5$. The source term Q is continually recalculated using the latest values of $\tilde{\rho}'_{v1}$:

$$Q^j_i = \frac{\left(\tilde{\rho}'_{v1}\right)^j_i - {}^n\left(\rho'_{v1}\right)^j_i}{\delta t}$$

$$+\,\frac{1}{r_i\delta r}\left[\left\langle{}^{n+1}u_v r\tilde{\rho}'_{v1}\right\rangle^j_{i+\frac{1}{2}} - \left\langle{}^{n+1}u_v r\tilde{\rho}'_{v1}\right\rangle^j_{i-\frac{1}{2}}\right]$$

$$+\,\frac{1}{\delta z}\left[\left\langle{}^{n+1}v_v\tilde{\rho}'_{v1}\right\rangle^{j+\frac{1}{2}}_i - \left\langle{}^{n+1}v_v\tilde{\rho}'_{v1}\right\rangle^{j-\frac{1}{2}}_i\right] \quad .$$

The $\tilde{\rho}'_{v1}$ values are iterated until $Q\approx 0$ for every cell, at which time the current $\tilde{\rho}'_{v1}$ values are considered to have become the ${}^{n+1}\rho'_{v1}$ values. In practice, we have found that Q can be tested against the same convergence term used for the $\tilde{D}$ test. Although Q is analogous to $\tilde{D}$ of the first iteration, and many of the same comments apply equally, one should note that the n+1 u's and v's result solely from the first iteration and remain unchanged through this second iteration. The second and third terms in Q^j_i are written in a manner like that used in the $\tilde{D}$ equation. For example,

$$\left\langle{}^{n+1}u_v r\tilde{\rho}'_{v1}\right\rangle^j_{i+\frac{1}{2}}$$

$$= {}^{n+1}\left(u_v\right)^j_{i+\frac{1}{2}}\, r_{i+\frac{1}{2}}\left[\left(\tfrac{1}{2}+\xi\right)\left(\tilde{\rho}'_{v1}\right)^j_i + \left(\tfrac{1}{2}-\xi\right)\left(\tilde{\rho}'_{v1}\right)^j_{i+1}\right] ,$$

in which

$$\xi = \frac{\beta_o\delta t}{\delta r}\left[{}^{n+1}\left(u_v\right)^j_{i+\frac{1}{2}}\right] + \alpha_o\ \mathrm{sign}\left[{}^{n+1}\left(u_v\right)^j_{i+\frac{1}{2}}\right] \quad .$$

The solution of the second iteration completes the calculations associated with Phase 2, the implicit half of the cycle.

E. Boundary Conditions

A variety of boundary conditions have been successfully tested in the KACHINA code. Figure 3 illustrates the currently available boundary options for the bottom, right, and top edges of the computing mesh, as specified by the input data for each particular problem. In all instances, the left boundary of the mesh serves as the axis of cylindrical symmetry. Typical configurations we have used include a box with three rigid free-slip walls, or the other extreme of three continuative outflow boundaries, in which the rigid section of the right-hand wall has been reduced to zero height. A set of studies of fluidized dust beds used a specified inflow bottom boundary, a rigid right wall, and outflow along the top. On the right boundary, the transition point from rigid free slip to outflow, noted in Fig. 3, can lie at any desired cell boundary from bottom to top, and it allows the extremes of an all-rigid free-slip boundary or an all-outflow boundary.

As Fig. 4 shows, the boundary conditions considered here are described in relation to the bottom boundary, the treatment being entirely analogous at the other boundaries. These conditions are more easily applied if the computing mesh shown in Fig. 1 is surrounded on all four sides by a belt of fictitious or outside cells. These cells provide convenient exterior storage locations for functions of the neighboring inside cell variables, chosen so that when they are referenced in the equations the desired boundary condition is satisfied automatically without any testing of the boundary type.

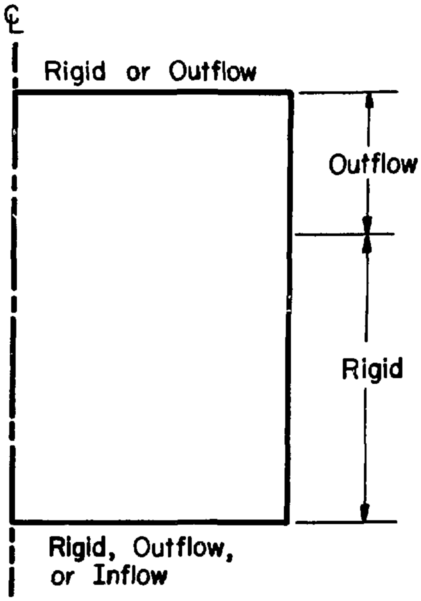


Fig. 3. Boundary conditions available in KACHINA.

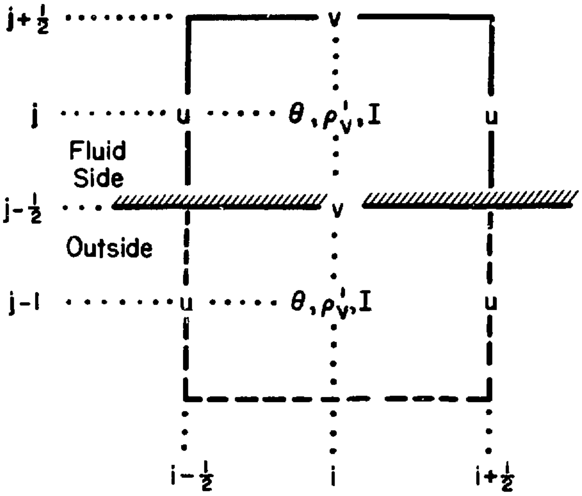


Fig. 4. Quantities involved in boundary conditions at the bottom of the KACHINA mesh.

(1) RIGID (FREE-SLIP): A rigid free-slip boundary represents an axis or plane of symmetry, or a nonadhering surface that exerts no drag upon the fluid. The normal component of velocity vanishes at the wall, and there is no gradient in scalar variables across the boundary. If the boundary is insulated:

$$\theta_i^{j-1} = \theta_i^j \quad ,$$

$$\left(v_v\right)_i^{j-\frac{1}{2}} = \left(v_d\right)_i^{j-\frac{1}{2}} = 0 \quad ,$$

$$\left(\rho'_{v1}\right)_i^{j-1} = \left(\rho'_{v1}\right)_i^j \quad ,$$

$$\left(\rho'_v\right)_i^{j-1} = \left(\rho'_v\right)_i^j \quad ,$$

$$\left(I_v\right)_i^{j-1} = \left(I_v\right)_i^j \quad .$$

(2) INFLOW (Specified): The inflow boundary allows vapor, only, to move into the system at a prescribed rate that, in principle, can vary with position and time, although in the present KACHINA it is constant.

$$\theta_i^{j-1} = \theta_{\text{specified}} \quad ,$$

$$\left(v_v\right)_i^{j-\frac{1}{2}} = v_{\text{specified}} \quad ,$$

$$\left(v_d\right)_i^{j-\frac{1}{2}} = 0 \quad ,$$

$$\left(\rho'_{v1}\right)_i^{j-1} = \theta_{\text{specified}} \left(\rho_{v1}\right)_{\text{specified}} \quad ,$$

$$\left(\rho'_v\right)_i^{j-1} = \theta_{\text{specified}} \left(\rho_{v1} + \rho_{v2}\right)_{\text{specified}} \quad ,$$

$$\left(I_v\right)_i^{j-1} = \left(I_v\right)_{\text{specified}} \quad .$$

(3) OUTFLOW (Continuative): A continuative outflow boundary allows fluid to leave the system at its own chosen rate, hopefully with minimal upstream flow disturbance whether the flow is subsonic or supersonic. Nothing precludes an outflow boundary from becoming a continuative inflow boundary (without specification), should the velocity field at the boundary become reversed. An example is shown in the final sample calculation discussed in Ref. 7:

$$\theta_i^{j-1} = \theta_i^j \quad ,$$

$$\left(v_v\right)_i^{j-\frac{1}{2}} = \left(v_v\right)_i^{j+\frac{1}{2}} \quad ,$$

$$\left(v_d\right)_i^{j-\frac{1}{2}} = \left(v_d\right)_i^{j+\frac{1}{2}} \quad ,$$

$$\left(\rho'_{v1}\right)_i^{j-1} = \left(\rho'_{v1}\right)_i^j \quad ,$$

$$\left(\rho'_v\right)_i^{j-1} = \left(\rho'_v\right)_i^j \quad ,$$

$$\left(I_v\right)_i^{j-1} = \left(I_v\right)_i^j \quad .$$

For all three types of boundary conditions, there is no gradient in droplet densities or internal energies across the boundary:

$$\left(\rho'_{d1}\right)_i^{j-1} = \left(\rho'_{d1}\right)_i^j \quad ,$$

$$\left(\rho'_d\right)_i^{j-1} = \left(\rho'_d\right)_i^j \quad ,$$

$$\left(I_d\right)_i^{j-1} = \left(I_d\right)_i^j \quad ,$$

nor is there any gradient in the tangential component of velocity across the boundary:

$$\left(u_v\right)_{i-\frac{1}{2}}^{j-1} = \left(u_v\right)_{i-\frac{1}{2}}^j \quad ,$$

$$\left(u_d\right)_{i-\frac{1}{2}}^{j-1} = \left(u_d\right)_{i-\frac{1}{2}}^j \quad ,$$

$$\left(u_v\right)_{i+\frac{1}{2}}^{j-1} = \left(u_v\right)_{i+\frac{1}{2}}^j \quad ,$$

$$\left(u_d\right)_{i+\frac{1}{2}}^{j-1} = \left(u_d\right)_{i+\frac{1}{2}}^j \quad .$$

This specification of tangential velocity is required only for marker particle movement, discussed in Sec. II.E. Because there is no shear viscosity in this version of the code, the external tangential velocities are not otherwise referenced.

The momentum components in Phase 1 have values at the boundary that are based on the density and velocity of vapor or droplets, as appropriate, at that boundary position, in accordance with the above treatments.

The boundary conditions are initially set in the problem setup. The exterior values of the explicit variables are reset in Phase 1 as the neighboring fluid-side n+1 values become available. During the pressure iteration in Phase 2, the exterior values of $\tilde{p}$, $\tilde{\rho}'_v$, and the wall velocities are updated continuously to keep them appropriate to the continuously changing interior values, and ρ'_{v1} is treated similarly in the second iteration.

Note that KACHINA requires no special pressure boundary conditions. This is a direct benefit of the Chorin-Hirt method[10] chosen for the Phase 2 iteration procedure, which also contributes to efficiency and simplification of the solution process.

F. Sudden Incompressibility

In certain flow situations, the vapor can be almost completely extruded from some cell or number of cells. We conclude that the droplets have come into rigid contact with one another, so that the $\vec{u}_d$ field has suddenly become essentially incompressible. An analogous situation is that of a set of billiard balls on a table, which may be moved about with great freedom until they are drawn together and racked into the triangle, whereupon they become tightly packed. The billiard balls may subsequently be separated and resume their previous freedom of movement; similarly, the flow that became so suddenly incompressible may open up again at some later time.

Sudden incompressibility must be allowed for and treated in a special manner in a multifield computing model; otherwise, the calculation will almost surely break down sooner or later. Fortunately, the void fraction will forewarn of the situation if it is carefully monitored on a cell-by-cell basis, as in such instances θ will become small. We test whether each ${}^{n+1}\theta < \theta_o$, where the value $\theta_o = 0.02$ has been found appropriate, at least for the test problems we have run. Usually, of course, ${}^{n+1}\theta > \theta_o$, and the calculational procedure is the standard one described in the preceding sections. For those cells in which ${}^{n+1}\theta < \theta_o$, in principle the only modification required would be simply to force the velocity divergence to vanish by reducing the $\tilde{D}$ equation in Phase 2 to a $\nabla \cdot \vec{u}_d = 0$ expression. In KACHINA, we begin by setting ${}^{n+1}\theta \equiv 0$, principally to make such cells highly visible in the numerical printout. Alternatively, we could leave θ at its calculated value, which would be the required procedure if θ_o were somewhat larger, as for example in a relatively porous bed of close-packed granules. In practice, there are actually several places in each of the two phases at which we change the computational procedure. In Phase 1:

(1) We set

$$ {}^{n+1}\theta_i^j = 0 \quad , $$

$$ \tilde{p}_i^j = {}^n p_i^j \quad , $$

$$ A_i^j = A_\infty \quad , $$

$$ K_i^j = K_\infty \quad , $$

where A_∞ and K_∞ are some large numbers, say about 10^{10} times the ordinarily expected magnitudes of these quantities. The choice of time level n+1 for the θ's appearing in the *standard* K equation is dictated by the fact that when a suddenly incompressible region opens up again, the restored condition ${}^{n+1}\theta > \theta_o$ will allow passing of this test and reestablishment of the standard procedure for the cell. However, ${}^n\theta = 0$ because of the *previous* state of the cell, and use of this old flag as ${}^n\theta$ in the K equation would cause the computer to try a division by zero.

(2) In the ${}^{n+1}I_v$ equations, we eliminate two terms. The drag term is omitted on the assumption that $\vec{u}_v$ and $\vec{u}_d$ must be very closely tied together, and the work term is omitted because no work can be done on the vapor in such cells.

(3) Neither the momentum equations nor the initialization of the tilde velocities at the beginning of Phase 2 requires changing, because with $K = K_\infty$, the initializations force $\vec{u}_v \equiv \vec{u}_d$.

In Phase 2:

(4) Those cells with ${}^{n+1}\theta < \theta_o$ require different expressions for both β_p and $\tilde{D}$:

$$ \left(\beta_p\right)_i^j \equiv \frac{\delta r^2 \delta z^2}{2\delta t\left(\delta r^2 + \delta z^2\right)} \left(\rho'_d\right)_i^j \quad , $$

$$ \tilde{D}_i^j = \frac{1}{r_i \delta r}\left[r_{i+\frac{1}{2}}\left(\tilde{u}_d\right)_{i+\frac{1}{2}}^j - r_{i-\frac{1}{2}}\left(\tilde{u}_d\right)_{i-\frac{1}{2}}^j\right] $$

$$ + \frac{1}{\delta z}\left[\left(\tilde{v}_d\right)_i^{j+\frac{1}{2}} - \left(\tilde{v}_d\right)_i^{j-\frac{1}{2}}\right] \quad . $$

Note that no consideration of donor-cell formulations arises in this $\nabla \cdot \vec{u}_d$ form of the $\tilde{D}$ equation.

(5) Neither the intermediate nor final values of pressure, density ρ'_v, or velocity in Phase 2 require changing. As in (3) above, the equations will automatically provide ${}^{new}\vec{u}_v \equiv {}^{new}\vec{u}_d$, and they arrive at values that ensure $\nabla \cdot \vec{u}_d = 0$ for those particular cells.

(6) The ρ'_{v1} iteration requires no changes.

II. THE KACHINA COMPUTING PROGRAM

A. General Structure

KACHINA was written for the CDC-7600, to provide a tool for several specific studies and for further methodology development. It embodies a number of features to make efficient use of computer storage and time, and it follows the better programming concepts, ill-defined but popularly called "structured programming," that have received so much attention recently. The basic KACHINA will be extended in several directions by a number of investigators, and its modular form has already worked successfully in other recent computing programs. The physical arrangement and the top-to-bottom flow in the coding correspond to the logical sequence of the computing cycle to the greatest degree practicable. The efficiency loss that results from writing the entire code in a higher level language rather than in machine language is hopefully counterbalanced by increased readability for most users and the simplification of adapting it in the future for use at other installations and for computers other than the CDC-6000/7000 series. Computing efficiency can be increased substantially by carefully rewriting the iteration sections in machine language, which we strongly recommend to anyone doing a significant amount of calculation with a FORTRAN program containing any iterative solutions.

As depicted in Fig. 5, KACHINA is built in an overlay fashion to minimize the use of Small Core Memory (SCM), the fast memory on the CDC-7600. The main overlay (0,0) always resides in SCM, and it contains the main controlling program, KACHINA. Subservient to it are the longer programs in the two primary overlays, (1,0) and (2,0), which reside on disk storage. KASET is the setup program, and KACHYDR performs the two-phase hydrodynamics described in Sec. I.

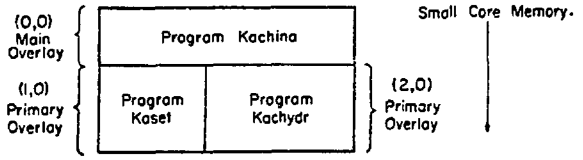


Fig. 5. The KACHINA three-program overlay structure.

The structure within each of these three programs is further detailed in Fig. 6, which introduces the UPDATE notation used in the actual code.

In addition to the main program, KACHINA, the (0,0) overlay contains the common KSC, which is the SCM portion of the information written on tape for restarting purposes and is therefore the natural repository for all the SCM data that must be retained from cycle to cycle. Any subroutines that will be referenced by the primary overlays should also be placed in (0,0) to ensure that they are always resident in SCM and directly accessible by all programs. At present, LOOP is the only such subroutine; its function is described in Secs. II.C and II.D.

To set up a calculation from initial input data, the main program calls the (1,0) overlay program KASET from the disk and surrenders control to it. This overlay is placed in SCM immediately following the (0,0) overlay. KASET itself is only a two-instruction program; it prints "SETUP" to indicate that control has reached (1,0), and then immediately calls subroutine SETUP to perform the actual setup, creating the computing mesh with its initial cell quantities, and generating marker particles if they

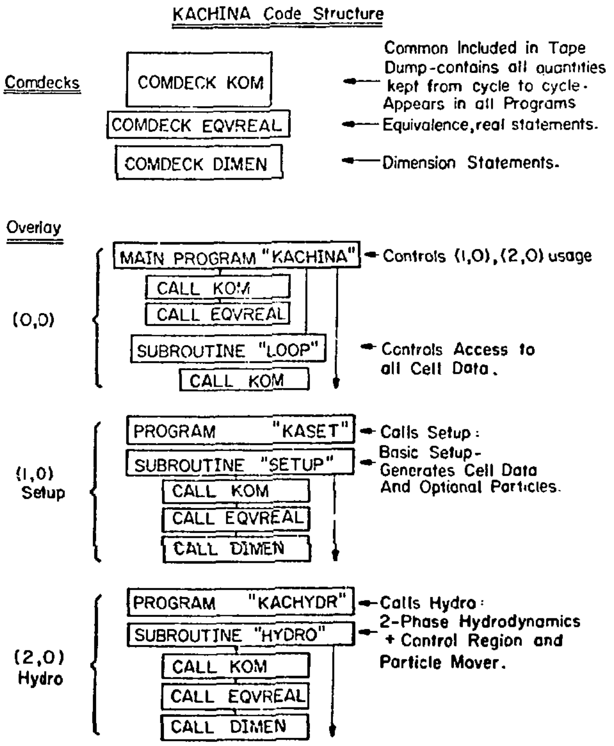


Fig. 6. The KACHINA overlays, showing the functions of all sections and the UPDATE nomenclature.

are specified. SETUP also sets boundary conditions for the edges of the mesh and determines parameters for microfilm plotting. When the problem setup is completed, SETUP returns control to the (0,0) main overlay program.

To calculate after setting up, the main program calls the (2,0) primary overlay KACHYDR from the disk and surrenders control to it. Because this overlay is of the same level as (1,0), it covers the image of (1,0) in SCM, as it is read in to the same locations following the (0,0) overlay and thus allows reuse of the SCM space. Like KASET, KACHYDR is a two-instruction program: it prints "HYDRO" and immediately calls subroutine HYDRO. Should the job abort because of an unexpected error, the printed message allows the user to ascertain quickly which program he is in, which might otherwise be difficult inasmuch as the range of instruction addresses for both the (1,0) and (2,0) overlays starts at the same point.

HYDRO is the largest section of code in the computer program. It contains the two-phase hydrodynamics, the calculational cycles of which are repeated continuously under the direction of a "control region." This region is strategically placed at the beginning of the subroutine, at which point in the cyclic process the quantities of greatest interest representing the solution at a given instant in problem time, are available. The control region provides all microfilm plots and numerical listings of cell data. It also increments the problem time t by the current δt, performs tape dumps and tape restarts, and senses problem completion or an impending operating system time limit. In the latter two events, it returns control to the main program, which, in turn, always searches the input queue for further tasks. If there are none, the job is ended.

To restart a calculation from a tape dump, the main program bypasses the (1,0) overlay and calls (2,0) instead. HYDRO senses the restart condition immediately, and the control region reads the information from tape into memory and turns control over to the point in the calculation cycle that will continue the problem from where it left off when the tape dump was made.

B. The Indexing Notation

Figures 2 and 4 show that some variables are defined at cell centers and some at cell edges, as is typical of a number of Eulerian computing methods. In FORTRAN, one can represent p_i^j simply by "P(I,J)," but $u_{i+\frac{1}{2}}^j$ cannot be represented by a "half-integer" index, so our convention is that "U(I,J)" denotes this particular velocity. Thus the indices I and J denote a quantity located at the center of cell (i,j), at the right edge $(i+\frac{1}{2},j)$, or at the top edge $(i,j+\frac{1}{2})$, depending on where the quantity is defined to be by the difference equations. In KACHINA, "(I,J)" is replaced simply by "(IJ)," as only single subscripts are used for computer efficiency. In the KACHINA subscript notation, the letter "P" stands for "+," and "M" stands for "-." Thus, we write

IJ = (i,j) ,

IMJ = (i-1,j) ,

IJM = (i,j-1) ,

IPJP = (i+1,j+1) ,

etc.

Such a notation permits easy reading of programmed difference equations in the code. Figure 7 shows the single subscripts used to define cell quantities about a cell (i,j).

As the number of cell edges in either direction is one greater than the number of cells, it is

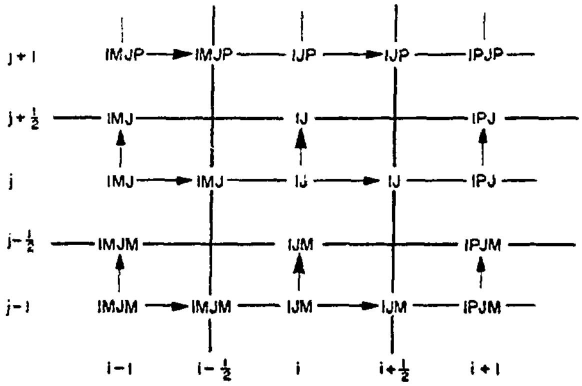


Fig. 7. Single-subscript notation for cell and cell-edge quantities about cell (i,j).

apparent that the grid in computer storage must be at least ($\bar{I}$+1) by ($\bar{J}$+1) in size. Because our indexing refers to cell centers and right and top edges, one extra column of storage on the left and one extra row along the bottom are provided. KACHINA also includes an extra row of cells across the top and an extra column up the right, giving a mesh that is ($\bar{I}$+2) by ($\bar{J}$+2) in extent. As described in Sec. I.E, these exterior zones are known as outside or fictitious cells, and surrounding the mesh with them helps in treating the boundary conditions.

An example of the actual KACHINA mesh for the virtual mesh of Fig. 1 is shown in Fig. 8, from which it is evident that double DO loops in FORTRAN to sweep all cell centers would have the limits J = 2 to JP1 and I = 2 to IP1. Similarly, DO loops with limits J = 2 to JP1 and I = 2 to IBAR will access all interior u velocity components, and those with limits of J = 2 to JBAR and I = 2 to IP1 will access all interior v velocity components. Boundary velocities and exterior values of the cell-centered variables receive special treatment and are not normally included within the limits of the DO loops.

C. Storage of Cell Data

Although the present modest size of the code certainly doesn't warrant overlay construction, the basic KACHINA of this report has been built to allow for considerable expansion in two respects. First, calculations will become more finely resolved, implying use of several thousand computing cells. Second, more coding will be added to deal with other physical phenomena. This coding will include such features as phase transitions, a third field, more realistic equations of state, chemical and nuclear processes, and better isotropic force treatments, to name only a few. Both these considerations will greatly increase the demands on SCM space, and it is far better to allow for such growth in the initial architecture of a program than to have to add it later in some fashion that would require substantial rewriting to retain reasonable efficiency.

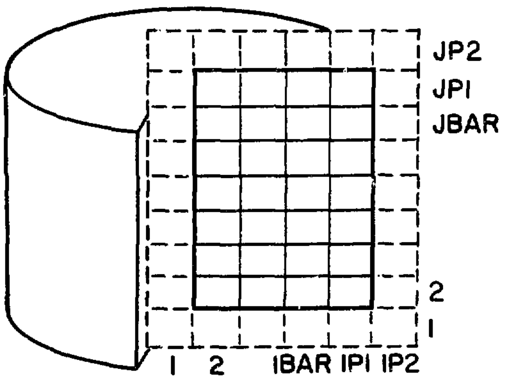


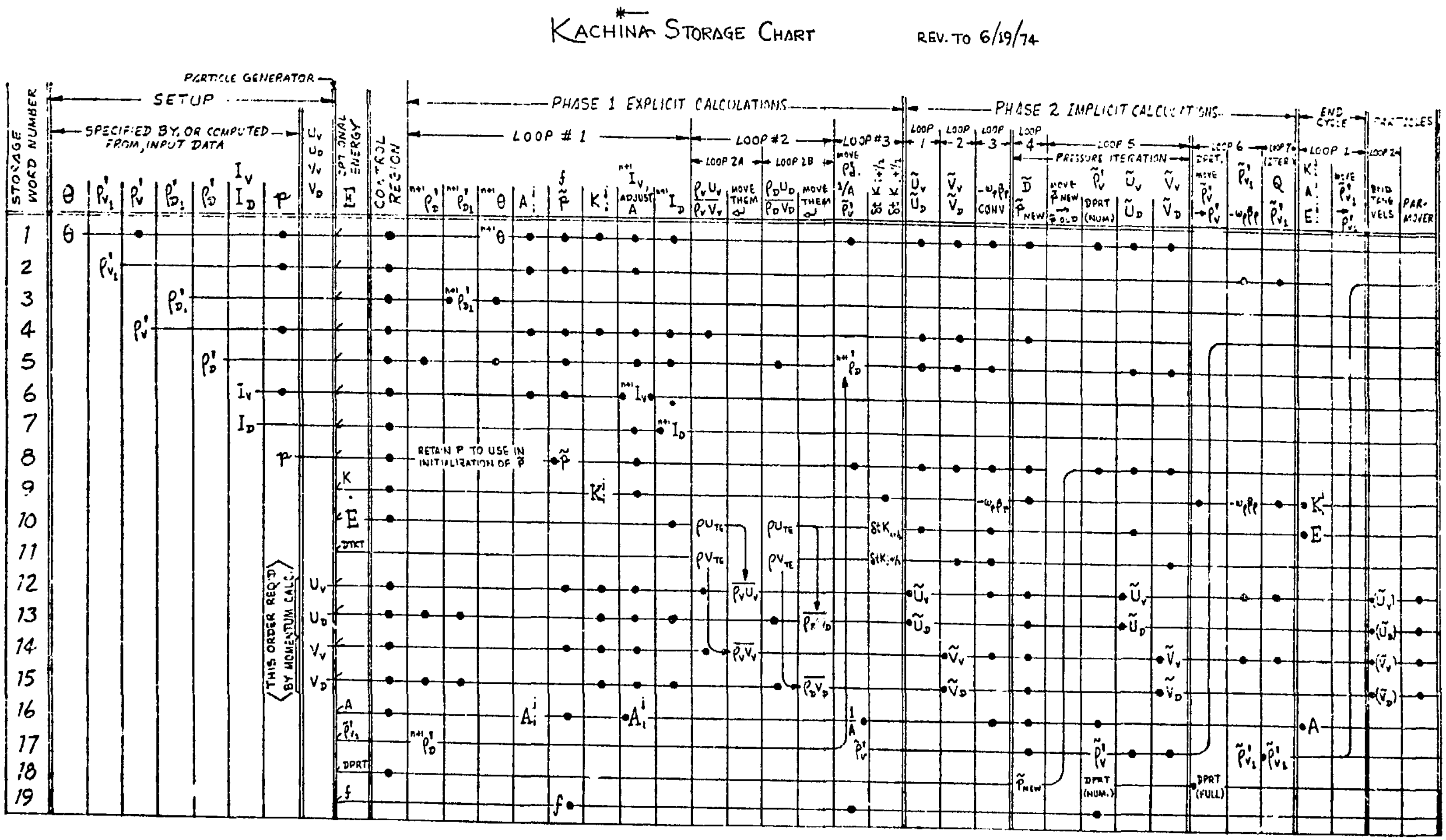
Fig. 8. KACHINA indexing scheme for the virtual mesh of Fig. 1, showing an example with many fewer cells than ordinarily are used for a calculation.

In addition to the overlay structure, KACHINA has several provisions for storage of cell data to allow efficient use of SCM. SCM requirements are significantly reduced by our use of only 19 storage words per cell, although the full calculation cycle requires 32 variables. Use of so few storage words is made possible by retaining quantities during a cycle only as long as they are needed, and then using their storage words for other quantities. Figure 9 shows the allocation of the 19 storage words for a KACHINA cell in the (1,0) and (2,0) overlays. The ordering from left to right corresponds to the actual order in which quantities are calculated in the code. A black dot indicates that the quantity currently in the given storage word is referenced to calculate the quantity specified at the top of the column.

Note that before the iterations, the cell edge quantities $\delta t K^{j}_{i+½}$ and $\delta t K^{j+½}_{i}$ and the cell quantity $\left(-\omega_p\beta_p\right)^{j}_{i}$ are formed, and that A^{j}_{i} is converted to its reciprocal. Because all these quantities remain invariant in the iterative process, it is expedient to compute them throughout the mesh beforehand to avoid needless repetitive calculation within the iteration itself.

During the pressure iteration, ${}^{new}\tilde{p}$ replaces ${}^{old}\tilde{p}$ as soon as there is no further need to keep δp available in memory. As the iterations are completed, the field values of $\tilde{D}$, K, and A are retrieved in case they are to be listed numerically in an output routine in the next pass through the control region. Also, $\tilde{\rho}'_v$ replaces ${}^{n}\rho'_v$ in word 4, becoming ${}^{n+1}\rho'_v$ by doing so, and similarly $\tilde{\rho}'_{v1}$ replaces ${}^{n}\rho'_{v1}$ in word 2, becoming ${}^{n+1}\rho'_{v1}$.

In the Phase 1 explicit calculations, the convection equations require neighboring values (on all four sides of the cell) of the variable being solved for. To maintain the correct time level (n) of these



WORD #:	EQUIV:	NAME:
1	θ	TH
2	ρ'_{V_1}	RØVPR1
3	ρ'_{D_1}	RØDPR1
4	ρ'_V	RØVPR
5	ρ'_D	RØDPR
6	I_V	SIEV
7	I_D	SIED
8	p	P
9	$K_j^i, -\omega_f\beta_f, \omega_f\beta_f$	KIJ, ØMBETA
10	$E, \rho U_{TE}, \delta t K_{i+1/2}$	E, RØUTE, DTKR
11	$\rho V_{TE}, \delta t K_{j+1/2}$	RØVTE, DTKT
12	$U_V, \overline{\rho_V U_V}$	UV, RØUV
13	$U_D, \overline{\rho_D U_D}$	UD, RØUD
14	$V_V, \overline{\rho_V V_V}$	VV, RØVV
15	$V_D, \overline{\rho_D V_D}$	VD, RØVD
16	$A, 1/A$	A, RA
17	$CQ, \tilde{\rho}'_V, \rho'_D, \tilde{\rho}'_{V_1}$	CQ, RØVPRT, RØDPRT, RØVPR1T
18	$\tilde{p}_{NEW}, \mathcal{D}$	PP, DPRT
19	f	F

Fig. 9. The storage of cell data in KACHINA, showing how the 19 words per cell are allocated among 32 named variables during the cycle.

referenced variables, the new (n+1) values could be stored in completely separate arrays and later transferred, just as ρ'_v and ρ'_{vl} are handled. We have chosen, instead, to temporarily store aside $\bar{I}$-length vectors of the (n+1) values, transferring them at the end of each row to relieve the demands on cell storage.

It appears that using fewer than 19 words per cell would complicate the computer logic significantly while providing very little gain, and using more than 19 would begin to waste space. Our various storage treatments have been governed by a balance between these considerations.

The contour quantity (CQ) in the control region denotes the field of some chosen cell variable for which a contour plot is drawn on microfilm. The present choices are all of the quantities in words 1 through 10, which are automatically plotted in sequence by placing the complete field of each quantity in CQ as its plotting turn occurs. This transfer to CQ could be avoided by simple indexing through words 1 to 10, but intermediate storage in CQ allows values to be adjusted specifically for plotting purposes, as is sometimes necessary.

Charts such as Fig. 9 have proven extremely useful in initially planning the storage before a code is written, but they are equally useful thereafter in visualizing the quantities available at any given point in the calculation cycle, and the locations of storage vacancies. The storage layout changes frequently during the development of the code, and the version in Fig. 9 is certainly not final for KACHINA.

Reducing the number of storage words per cell, as discussed above, will help to allow larger meshes to fit in SCM, but eventually SCM space is exhausted. The next step is to transfer all the cell data to Large Core Memory (LCM), reading only some part of the mesh at a time into an SCM buffer for processing, and then rewriting it back out to LCM. The optimum procedure is that which requires the minimum number of read/write references to LCM, and one could, indeed, tailor the logic for each problem to do this. The procedure we have chosen, however, simply deals with three j-rows of cells at a time in SCM processing, without regard to overall mesh dimensions. To facilitate inclusion of the LCM routine described in Appendix C, the cell variables are stored "interleaved" with all the variables for a given cell stored contiguously, followed by all the variables for the next cell, etc. (Contrast this with the other method of storing cell variables in individual $\bar{I}$ by $\bar{J}$ blocks for each variable. That scheme is competitive only when the computing code is designed for smaller meshes that will always fit in SCM.)

In the SCM version of this report, the storage block AASC contains all cell data. At present, it has a dimension of 26 676 words, allowing any combination of IP2 by JP2 $\leq$ 1404 cells, at 19 words per cell. A 25 by 50 logical mesh, requiring an actual 27 by 52 mesh, dictated this particular choice. With the transfer of cell data to LCM, however, it is no longer appropriate to retain AASC within the main SCM common KSC, because KSC should contain only the tape restart information. Therefore, AASC is defined in a separate SCM common that is never written on tape, and the tape dump/restart routines are modified to write/read the entire LCM storage block of complete cell data. These and related matters are discussed in detail in Appendix C.

Next, we must consider how the code actually accesses cell data.

The single-subscript index notation described in Sec. II.B can remain the same, whether the cell data are stored in SCM or LCM, because the actual location of the data is transparent to the primary overlays. Subroutine LOOP in the (0,0) overlay is of crucial importance here, as it has complete control over all references to cell data and relieves the primary overlays of any direct concern with cell data transfers.

In essence, LOOP's responsibility is simply to have three rows of the mesh, j, j+1, and j-1, available in SCM for processing, and to have the corresponding indices IJ, IJP, and IJM set properly to the column i = 2 cells to begin the processing of each row. At the end of each row, LOOP must step up the one row in j and reset the three indices accordingly.

In the SCM version of the program, this process is trivial because no LCM logic is involved. Sweeps over the mesh always begin at the lower left corner and move across to the right edge, then up by rows of cells. To begin a typical sweep, the calling program CALLS the "START" entry in LOOP, which merely sets IJ to reference the lower left cell (2,2), IJP to reference the cell (2,3) above it, and IJM to reference the cell (2,1) below. Control is then RETURNed to the calling program, which then initiates an ordinary double DO loop.

Secondary indices are often needed to reference cells located in columns to the left or right of the column identified by IJ, IJP, and IJM. These indices are easily obtained by applying increments or decrements of the variable NQ, the number-of-quantities, that is, the number-of-storage-words-per-cell, to the three primary indices. In this manner, reference can be made to any neighbor of cell IJ shown in Fig. 7.

Similarly, the calling program can progress across the row from left to right by adding NQ to each index, to advance one column at a time. At the end of the row, the I DO loop falls through, leaving IJ, IJP, and IJM referring to cells in column IP1. A CALL is then issued to the "LOOP" entry point in LOOP, which increments the three indices over two additional columns by adding (2*NQ) to each index. This actually sets the indices back to the left and up through column 1 of the next row, stopping at column 2. Control is again RETURNed to the calling program, and this process is repeated until the entire mesh has been swept, as indicated by the fact that the outer DO loop on J falls through.

The number of storage words per cell in this KACHINA version is seen to be NQ = 19, as per Fig. 9. This number may be increased very simply by adding the new variables to the EQUIVALENCE and DIMENSION statements in the Comdecks EQVREAL and DIMEN, and redefining NQ in the (0,0) main program at one place only.

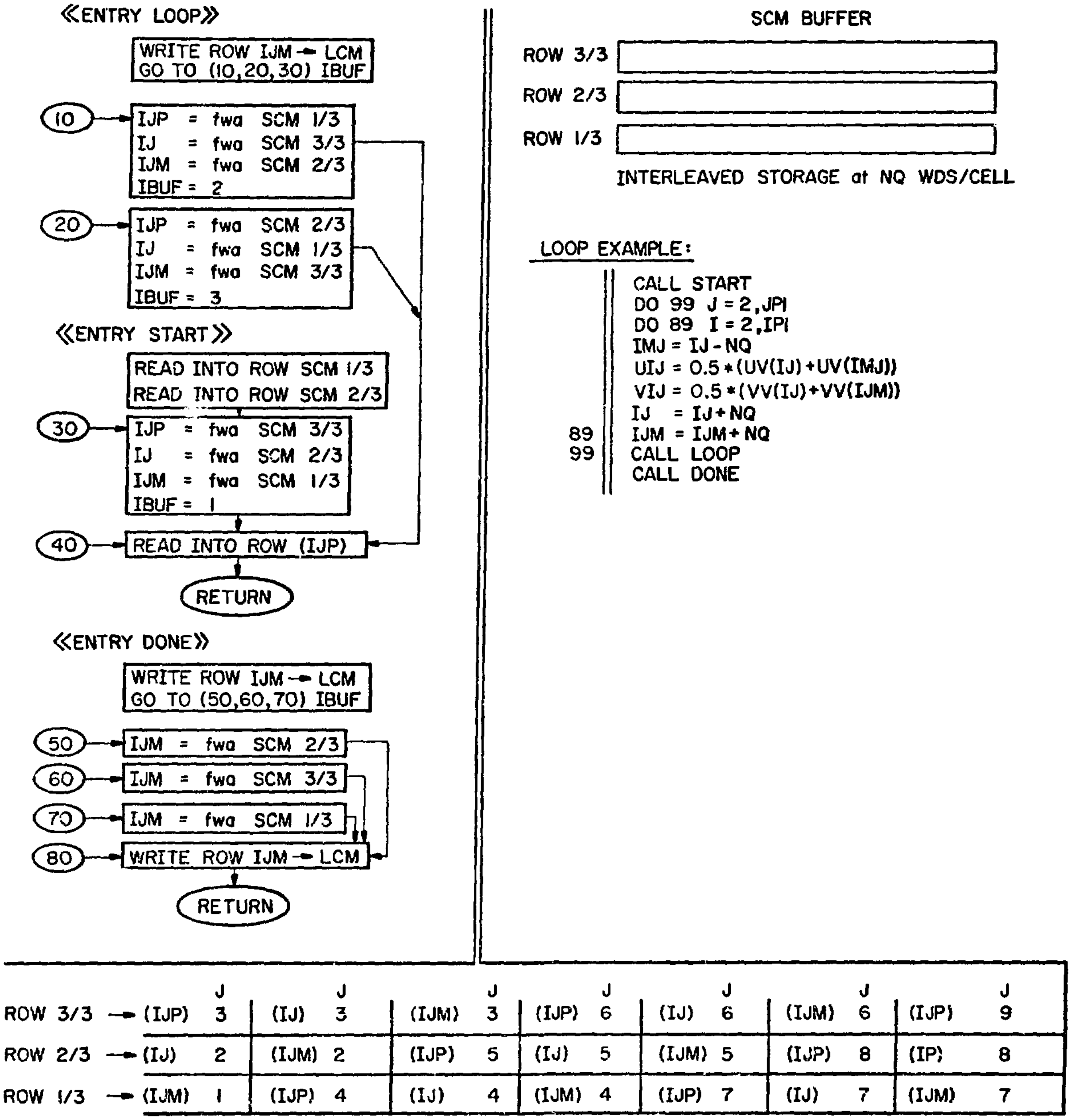
D. An Optional Three-Row Buffering Scheme

When placing all the cell data in LCM, the most significant modification to the code logic is to replace subroutine LOOP by an expanded version that will shuttle the cell data between the large LCM array and the small SCM buffer where it is operated on. These changes are detailed in Appendix C. Generally, LOOP will keep three complete rows of the mesh in SCM at any one time: the row being processed and the rows above and below, again referred to by IJ, IJP, and IJM. All calculations affecting cell data are actually performed directly on the current contents of the buffer. The merit of interleaving the cell data now becomes evident, as all quantities pertaining to the three rows of cells are instantly available. The schematic flow diagram and sample FORTRAN DO loop in Fig. 10 enlarge on the previous discussion to show how buffering can be added.

(1) As in the SCM version, the "START" entry of LOOP is CALLed before the double DO loops are initiated. But now, START reads in the entire contents of the bottommost three rows of the mesh from LCM to the SCM buffer, placing row j=1 in the buffer section designated "row 1/3;" likewise, row j=2 is read into "row 2/3," and row j=3 is read into "row 3/3." Rows 1/3, 2/3, and 3/3 are contiguous in SCM, and like their counterparts in LCM, each contains NQI = NQ*IP2 words. With the three rows read in, the calling program needs to know how to access data in the buffer. As before, this information is provided by setting IJM, IJ, and IJP to point to the first words of the i=2 column of cells in each row. Thus, IJM points to an address NQ words into SCM row 1/3, as do IJ in row 2/3 and IJP in row 3/3. Note the indicator IBUF, which is set to 1; it will control the subsequent reading and writing of individual rows and the resetting of the three indices. With the first three rows of cells read in and the basic indices set, control can be RETURNed to the calling program.

(2) The double DO loops are initiated, and processing is performed exactly as previously described for the SCM version. In the example shown in Fig. 10, we calculate the average u_v and v_v at the center of cell (i,j). The terminal statement of the inner DO loop, which again counts columns within each row, is statement No. 89. Note how the primary indices IJ and IJM are advanced to the next column in the row. The inner loop on I is repeated until the row is completed, at which time control passes to the "CALL LØØP" statement.

(3) The LØØP entry immediately writes row IJM back into LCM, and, depending on the value of IBUF, goes to statement No. 10, 20, or 30. Because IBUF was initially set to 1, control passes to statement No. 10 in our example. Note that now the indices IJP, IJ, and IJM are reset to point to different SCM rows -- IJP to the vacated row 1/3, IJ to 3/3, and IJM to 2/3. IBUF is reset to 2 to control the next entry to LOOP, and control passes to statement No. 40 which will read the new IJP row, row j=4, into



LOOP EXAMPLE:

```
      CALL START
      DO 99 J = 2,JP1
      DO 89 I = 2,IP1
      IMJ = IJ - NQ
      UIJ = 0.5*(UV(IJ)+UV(IMJ))
      VIJ = 0.5*(VV(IJ)+VV(IJM))
      IJ  = IJ+NQ
89    IJM = IJM+NQ
99    CALL LOOP
      CALL DONE
```

		J		J		J		J		J		J		J
ROW 3/3 →	(IJP)	3	(IJ)	3	(IJM)	3	(IJP)	6	(IJ)	6	(IJM)	6	(IJP)	9
ROW 2/3 →	(IJ)	2	(IJM)	2	(IJP)	5	(IJ)	5	(IJM)	5	(IJP)	8	(IP)	8
ROW 1/3 →	(IJM)	1	(IJP)	4	(IJ)	4	(IJM)	4	(IJP)	7	(IJ)	7	(IJM)	7

Fig. 10. KACHINA three-row buffer.

SCM row 1/3. Observe that there has been no unnecessary shuffling of data in SCM: row j-1 was read out and replaced by row j+1, and the three indices were reset to point to the locations of rows j+1, j, and j-1. As shown at the bottom of Fig. 10, the grid rows in SCM are in their actual logical order only every third row.

(4) LOOP returns to the calling program, advancing the outer DO-loop index J, and rows are processed similarly until all those specified by index J have been processed. Then control passes to the "CALL DONE" statement, an entry point that simply RETURNed in the SCM version.

(5) DONE is really only a cleaning-up operation. Because no further LCM reads are required, it merely writes the final two rows, j and j+1 (JP1 and JP2, respectively) back out into LCM. CALL DONE can be omitted on those loops, such as certain output routines, that reference but do not alter cell data.

Not indicated in the flow of Fig. 10 is the incrementing of the relative address indices for reading and writing LCM. These indices are initially set to 0 and incremented by NQI as processing progresses up the mesh.

Complete information on converting KACHINA to an LCM version with the three-row buffer routine is provided in Appendix C, which lists the changes required to convert the basic SCM version of KACHINA provided in Appendixes A and B.

The SCM version has been built so that the (1,0) and (2,0) overlays are completely compatible with LCM usage and require no modification in the conversion.

Recall that the SCM buffer AASC becomes separated from the SCM common block KSC. A dimension of only 5814_{10} words (=3 rows * 19 words per cell * 102 columns) will allow any $\overline{I} \leqslant 100$. Considering the reduction from the present 26 676 words, and also the great capacity of LCM (on the order of $400K_{10}$ words are available), it becomes evident that very large problems can be run with the LCM version of KACHINA. Further, the goal of freeing a considerable amount of SCM space is achieved.

E. Marker Particles

Marker particles are a purely optional feature in KACHINA, as they do not influence the flow, but are simply carried along with it. Microfilm plots of particle coordinates are often a very useful form of visual output, as they not only distinguish readily the regions occupied by the various components, but also indicate relative proportions of the interpenetrating components within each region. Marker particle plots also aid in the location of shock fronts, rarefaction waves, and regions that have become incompressible. These benefits become especially evident when motion pictures are generated by drawing a plot each calculation cycle. For these reasons, marker particles are a valuable part of most KACHINA studies.

Although the markers are moved by fluid velocities in the usual fashion, which will be described at the end of this section, the concept of our multifield KACHINA model introduces some novel aspects of marker creation.

First, it is appropriate to have a separate set of particles for each field component. This is necessary to allow accurate particle movement in our interpenetrating fluid model and to provide the proper visual distinction between the fields and their components. Each particle is tagged to indicate the component with which it is associated. This tag not only specifies whether the particle is to be moved according to u_v and v_v velocities or u_d and v_d velocities, but also indicates the plotting symbol chosen to identify the component.

The second novel aspect is the initial particle distribution. In the typical fluid-dynamics computer code, the density and spacing of particles is determined by specifying the total number of particles per *cell* in each direction, but in KACHINA we specify the total number of particles per unit *area*, called NPUA, expressed in cell units. In combination with the initial void fraction and the densities in each particular area, NPUA is subdivided into appropriate parts to determine the *effective* number of particles per unit area (PNEFF) for each type of particle. Therefore, a region with a large void fraction will contain a higher ratio of vapor particles to droplet particles than will a region with a smaller void fraction. The larger θ is, the smaller the number of droplet particles dispersed over the region, and at the limit $\theta = 1$, no droplet particles at all will be generated. The situation reverses as $\theta \to 0$, and at $\theta = 0$, no vapor particles are generated.

The densities play a similar role in the initial particle distribution, allowing for the distinction between the components themselves by controlling the number of particles of each component generated. Thus, the particles mirror the proportions of the components, a useful and desirable feature.

Particle generation over some fluid region takes place in KACHINA according to the following method. We assume a rectangular region of the mesh, encompassing some integer number of cells, where the number of cells in the radial direction is denoted by w, and that in the axial direction by h. Also specified for this region are NPUA and the initial values of ρ'_{d1}, ρ'_d, and θ, as we presently allow the creation of two types of droplet particles, but only one of vapor. In this case, three passes rather than four will be made to generate the particles for the region, one pass for each particle type. The major steps repeated for *each* of these three passes are as follows.

(1) PNEFF is calculated, and is given by one of the following equations, depending on which of the three passes is being performed.

(a) For droplet component number 1,

$$\text{PNEFF} = (\text{NPUA}) \left(\rho'_{d1}/\rho'_d\right) (1 - \theta) \ ;$$

(b) For droplet component number 2,

$$\text{PNEFF} = (\text{NPUA}) \left[1 - \left(\rho'_{d1}/\rho'_d\right)\right] (1 - \theta) \ ;$$

(c) For the vapor, $\text{PNEFF} = (\text{NPUA})\, \theta$.

(2) A uniform matrix of particles is to be laid down over the region, the total number to be given by the product of XNP particles in the r direction times YNP particles in the z direction. By this definition,

$$\frac{\text{XNP}}{\text{YNP}} \equiv \frac{w}{h} \ ,$$

and

$$(\text{XNP})\ (\text{YNP}) \equiv wh\ (\text{PNEFF}) \ .$$

These equations are combined and solved for XNP and then YNP:

$$\text{XNP} = w\ (\text{PNEFF})^{\frac{1}{2}} \ ,$$

$$\text{YNP} = (\text{XNP})\ h/w \ ,$$

which are then rounded up to the nearest integers. The product (XNP)*(YNP) is checked to ensure that it is greater than zero before proceeding to Step (3). If the product is zero, the logic should skip ahead to the pass for the next particle type.

(3) With the number of particles in each direction now determined for the type at hand, the spacing (s) between particles is given by

$$x_s = w/\text{XNP} \ ,$$

and

$$y_s = h/\text{YNP} \ ,$$

and the coordinates of the first particle (x_1, y_1) closest to the bottom left corner (c) of the rectangular region are given by

$$x_1 = x_c + \frac{x_s}{2} \ ,$$

and

$$y_1 = y_c + \frac{y_s}{2} \ .$$

(4) This information is next fed into a double DO loop with limits given by XNP and YNP. This loop generates and stores the array of particles and appends to each x coordinate some identifying tag, depending on the number of the pass.

If desired, the variable "ØFFSET" may be changed from its usual 0.0 to 1.0 to uniformly displace the entire particle matrix to the upper right by a distance $x_s/4$ and $y_s/4$. For the use of ØFFSET, refer to the description of a "part card" in Sec. II.F.

A set of three such passes is performed for each discrete fluid region in the mesh, each pass contributing some number of particles, which are accumulated into the total collection.

An example of the above scheme, with actual numbers and complete with drawings and chart, is given on the page labeled 4 in the Flow Diagram in Appendix A, where the complete flow for Marker Particle Generation also appears in detail.

The next particle consideration to be discussed is the scheme for moving them. For this purpose, the final (n+1) velocities resulting from Phase I are sufficient, if the exterior tangential velocities are set, as mentioned in the discussion of Boundary Conditions. The particles are moved by use of a typical area-weighting scheme involving the nearest velocity components. As shown in Fig. 11, the reference cell (shaded) is defined as the cell lying between the lower pair of u velocities when the u of the kth particle (u_k) is calculated. Similarly, the cell lying between the leftmost pair of v velocities is the reference cell when the v of the particle (v_k) is calculated. These velocities are either vapor or droplet velocities, depending upon the type of particle k. The reference cells provide an indexing base for obtaining the four u's and four v's. The donor cell is defined as the cell containing the particle before it is moved, and there is a 50-50 chance that the reference cell is the donor cell for either u_k or v_k. If we denote the reference cell by (IR,JR) and the donor cell by (ID,JD), we can show with the aid of Fig. 11 that it is possible to calculate u_k in every possible case by using ID and JR in the formula

$$u_k = A_2{*}U(ID-1,\ JR+1) + A_1{*}U(ID,\ JR+1)$$

$$+ A_3{*}U(ID-1,\ JR) + A_4{*}U(ID,\ JR)\ ,$$

without ever calculating the quadrant of the cell in which the particle k lies. Similarly, v_k is calculated using IR and JD:

$$v_k = A_2{*}V(IR,\ JD) + A_1{*}V(IR+1,\ JD)$$

$$+ A_3{*}V(IR,\ JD-1) + A_4{*}V(IR+1,\ JD-1)\ .$$

The Flow Diagram and FORTRAN listing in Appendixes A and B show how the partial areas A_1 through A_4 and the single subscript equivalents to IR, JR, ID, and JD are obtained quickly without ever testing the particle's position. After u_k and v_k have been determined, the particles are moved according to

$$r_k^{n+1} = r_k^n + u_k \frac{\delta t}{\delta r}\ ,$$

and

$$z_k^{n+1} = z_k^n + v_k \frac{\delta t}{\delta z}\ .$$

Note that the particles in KACHINA are both generated and stored in units of cell distance rather than problem distance.

KACHINA allows for outflow, automatically destroying any particle that crosses a mesh boundary. This is done by using one index (KP) to pick up particles and a second index (KPN) to store the new coordinates. Both are identical initially, but whenever a particle leaves the system, KPN is not advanced, so the storage of all the remaining unmoved particles is shifted by one position. This shift automatically destroys the affected particle, as the next particle is stored over it.

No provision has been made yet to create particles at an inflow boundary (if one exists), but this would logically be done after all the particles have been moved. Generally, at least one column or row of new particles would be generated each cycle and stored on the index KPN where it was left after the particle movement. See Ref. 11 for details of logic for creation of particles at a typical inflow boundary.

F. The Input Data

BCD data cards punched according to specific formats can provide all the parameters required to set up a KACHINA calculation. The number of cards varies depending upon the complexity of the mesh geometry. However, the following cards must always appear.

Card No. 1: IBAR, JBAR, DR, DZ, AO, BO, ROVO, NUV, CDR, RPAR (Format 2I4, 8F8.3), where:

IBAR = $\bar{I}$, the number of real (interior) zones in the radial direction.

JBAR = $\bar{J}$, the number of real (interior) zones in the axial direction.

DR = δr, the cell size in the radial direction.

DZ = δz, the cell size in the axial direction.

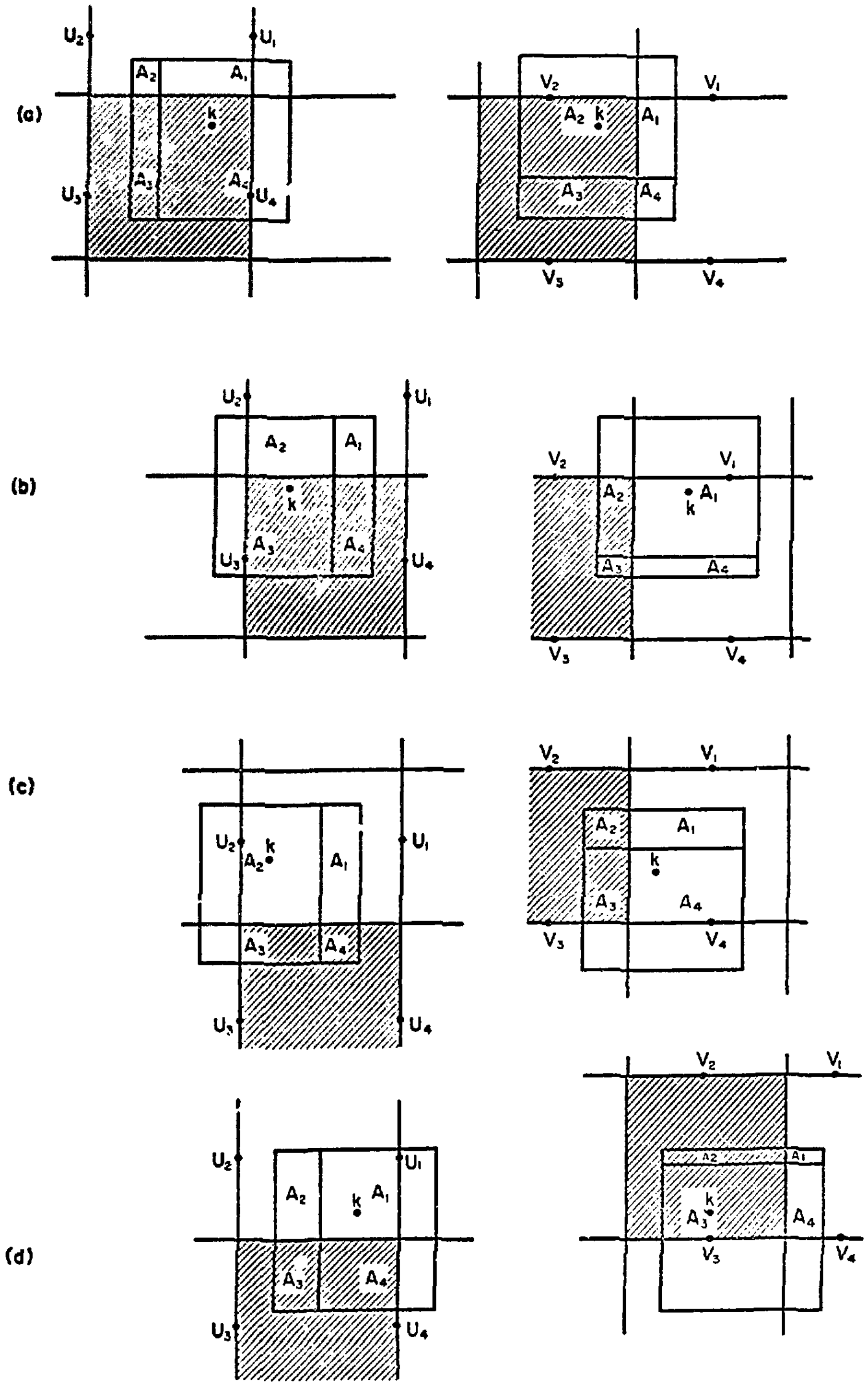


Fig. 11. Area velocity weighting scheme for u_k and v_k, with particle k shown for each of the four quadrants of the cell. The reference cell is shaded in each case. The subscripted u and v velocities are those of either vapor or droplets, depending upon the type of particle k.

AO = α_o constants denoting proportions of donor-cell

BO = β_o convective fluxing. See Secs. I.C. and I.D.

RØVO = ρ_{vo}, specified vapor density for incompressible flow.

NUV = ν_v, kinematic viscosity coefficient for the vapor field.

CDR = C_{DR}, the drag coefficient.

RPAR = r_p, the mean linear dimension of a droplet.

Card No. 2: NAME (Format 8A10), where columns 2-80 are used for problem identification on prints and all film frames.

Card No. 3: IDCØMP (Format 8A10), where columns 2-80 are used to identify material components once at the beginning of the film and printer output. (On both these cards, column 1 is treated as a carriage control, so it should be left blank. If desired, both cards may be left entirely blank, but they must always appear in the input deck.)

Card No. 4: ØMP, ØMRØ, EPS, G, KV, KD, EPV, EPD, R (Format 9F8.3), where:

ØMP = ω_p Phase-2 iteration relaxation parameters.

ØMRØ = ω_ρ See Sec. I.D for description.

EPS = ε_1, the convergence criterion for both iterative solutions in Phase 2. (With reference to Sec. I.D, only ε_1 appears in the input, whereas ε_2 is specified directly in the coding.)

G = g, gravity felt by both vapor and droplets, acting only in the axial direction, which may be + or − to pull up or down, respectively.

KV = k_v Heat conduction coefficients for the vapor and droplet fields.

KD = k_d

EPV = ε_v Epsilons for the Phase-1 nonisotropic force terms,

EPD = ε_d for vapor and droplet fields, respectively.

R = R, the exchange function that describes heat transfer between vapor and droplet fields.

Card No. 5: RØ1, RØ2, GAM1, GAM2, BV1, BV2, BD1, BD2 (Format 8F8.3), where:

RØ1 = ρ_1 The actual microscopic material densities of the two droplet

RØ2 = ρ_2 components.

GAM1 = γ_1 The ratio of specific heats for the two vapor components,

GAM2 = γ_2 appearing in Phase-1 A equation and pressure calculation.

BV1 = b_{v1}

BV2 = b_{v2} Specific heat constants for vapor and droplet components,

BD1 = b_{d1} appearing in Phase-1 A and T equations.

BD2 = b_{d2}

Card No. 6: JRIGID, IBØT, THIN, RØVIN1, RØVIN2, SIEVIN, VVIN, ITØP (Format 2I4, 5F8.3, I4), where:

JRIGID = the integral number of cells up the right boundary that are to be treated as rigid free-slip. Any cells above JRIGID are treated as continuative outflow. $0 \leq \text{JRIGID} \leq \text{JBAR}$. See Sec. I.E.

IBØT = Boundary condition for the (entire) bottom boundary, = 0 for rigid free-slip, = 1 for continuative outflow, = 2 for specified inflow, where:

THIN = θ_{in} Parameters for specified inflow along bottom boundary when

RØVIN1 = $(\rho_{v1})_{in}$ IBØT = 2; θ_{in}, $(\rho_{v1})_{in}$, $(\rho_{v2})_{in}$, and $(I_v)_{in}$ refer to the

RØVIN2 = $(\rho_{v2})_{in}$ center of the outside cell, and $(v_v)_{in}$ is defined on the

SIEVIN = $(I_v)_{in}$ boundary. See Sec. I.E.

VVIN = $(v_v)_{in}$

ITØP Boundary condition for the (entire) top boundary, 0 = rigid free-slip, 1 = continuative outflow.

Card No. 7: T, DT, T2ØMD, TLIMD, TWFIN, LPR, ISPR, (CØLØUR(N), N = 1, 3) (Format 5F8.3, 2I4, 3F4.1), where:

T = t_o, the problem starting time, usually zero.

DT = δt_o, the initial δt. Except in movie runs, δt is chosen automatically after cycle 10, according to the condition $\delta t = 0.1 \min(\delta r/u_{max}, \delta z/v_{max})$, where

$u_{max} = max(|u_v|, |u_d|)$ and
$v_{max} = max(|v_v|, |v_d|)$, in which all velocities over the mesh are considered.

T20MD = 1.0 to force tape dumps every 20 min of central processor (CP) time for restarting, or = 0.0 to bypass this option.

TLIMD = 1.0 to force a tape dump and RETURN to the (0,0) overlay just bfore the CP time limit specified on the JØB card is reached; > 1.0 to force tape dump and RETURN immediately after cycle 0 output; = 0.0 to run out to a full time limit with no tape dump.

TWFIN = problem finish time. When this time (t ≥ TWFIN) is reached, control returns to the (0,0) overlay. (*Upon RETURN to (0,0) for either the TLIMD or TWFIN condition, the (0,0) main program KACHINA searches the input queue for further tasks.*)

LPR = "Long Print" Control, where: 0 = movie option, 1 = cell-data listing on microfilm only, 2 = cell-data listing on both film and printer, 3 = cell-data listing on printer only. These options are described more fully in Sec. II.G.

ISPR = "Short Print" Control, where: the four-line listing of summations of mass, momentum, and energy is provided each cycle if ISPR = 2, or only on those cycles that have the "long print" of cell data if ISPR = 1.

CØLØUR(N),N = 1,3 These parameters are effective only for color microfilm processing, and are intended for movies of particles, where CØLØUR(1) refers to droplet component No. 1, CØLØUR(2) refers to droplet component No. 2, and COLOUR(3) refers to vapor. Seven basic color choices are available: 0.0 = white, 1.0 = red, 1.6 = yellow, 2.0 = green, 2.6 = cyan, 3.0 = blue, and 3.6 = magenta.

Card No. 8: (DTØ(N), N = 1, 10) is used in conjunction with

Card No. 9: (DTØC(N), N = 1, 10) (both are Format 10F8.3), where $DTØ_n$ specifies the problem time output interval for both plots and prints. $DTØC_n$ specifies the time at which the change to $DTØ_{n+1}$. As an example, assume that t is in seconds, and that output is wanted every 1/4 s for the first second, then every 2 s up to 8 s of problem time, then every 1/8 s up to t = 10, then more infrequently again, with output only every 10 s until t = 200. One would use

DTØ (1-10) = 0.25, 2.0, 0.125, 10.0 ,
DTØC (1-10) = 1.0, 8.0, 10.0, 200.0 .

To keep the output time interval fixed throughout a run, specify DTØ(1) = (interval) and DTØC(1) > TWFIN. When an output time is being approached, the automatic δt routine (see *DT* on Card No. 7) will choose a special δt for one cycle so that the output occurs at the precise time desired.

The above nine cards pertain to all KACHINA setups. They have defined a mesh and provided the parameters for its use. What remains to be defined is the contents of this mesh -- fluid regions to fill it in, and associated marker particles if desired. One "part card" is used to define each fluid region and any associated particles. Because the number of fluid regions can vary with problem geometry, the number of part cards also varies accordingly. The present definition of a "part" is limited to a cylindrical annulus, encompassing some rectangular region of cells, and constrained to follow cell boundaries. As shown in Fig. 12, four dimensions are adequate to define each part. A part card contains the following information:

NB, NR, NT, NL, RØDPRI1, RØDPRI2, RØV1, RØV2, SIEVI, SIEDI, NPUA (Format 4I4, 6F8.3, I4), where:

NB, NR, NT, NL } are four dimensions (see Fig. 12), specified in integer numbers of cells to emphasize that the part is constrained to follow cell boundaries. Thus, NL and NB specify how many cells in from the left and up from the bottom to locate

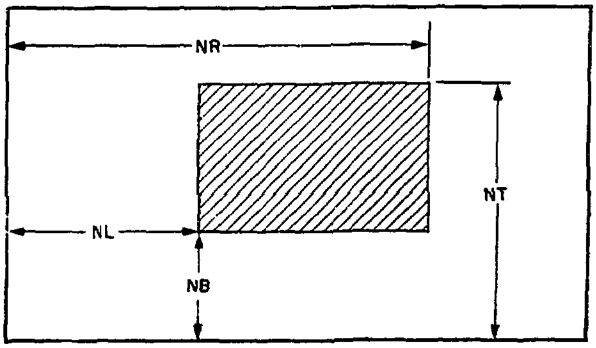


Fig. 12. Part shape available for fluid regions in KACHINA, defined by the integral number of cells over and up to the two corners.

the lower left corner of the region, and NR and NT similarly locate the upper right corner. As an example, if a single region is to cover the entire mesh, set NL = NB = 0, NR = $\overline{I}$, and NT = $\overline{J}$.

RØDPRI1 = $(\rho_{d1})_i$
RØDPRI2 = $(\rho_{d2})_i$
RØV1 = ρ_{v1}
RØV2 = ρ_{v2}
SIEVI = $(I_v)_i$
SIEDI = $(I_d)_i$

Initial values of the variables necessary to completely specify all cells in the fluid region, as described in Sec. I.B. (All velocities are initially set to zero.)

NPUA = number of particles per unit area, as described in Sec. II.E. Typical NPUA values in our test calculations have ranged from 1 to 4. To bypass the particle option, simply set NPUA = 0 for any or all parts. (Recall also the variable ØFFSET appearing in the particle generator, which may be used to displace the particle arrays to the upper right. See Sec. II.E.)

The part cards are processed individually, and the number of fluid regions is unlimited, clear up to the extreme of one region for each cell. If different parts refer to the same zone or zones, the latest information will override any earlier specifications for the cells, but any particles that have been created will be retained, suggesting one possible use for the ØFFSET parameter.

The set of part cards terminates with the final card having NR = 0 and the rest of the card unused. Therefore, at least two part cards must appear in a KACHINA input deck.

This completes the discussion of the input data cards. We see that an input deck must consist of at least 11 cards, but the maximum number is unlimited. The final card normally placed at the end of the input deck is in reality the first card for the next problem. The first quantity on Card No. 1 is IBAR, and it determines the action to be taken by KACHINA. If IBAR > 0, it is valid for use as $\overline{I}$, and the setup is called. The value IBAR = 0 indicates a tape restart, and IBAR < 0 indicates that the end of data has been reached. Thus, a negative IBAR card is the appropriate way to terminate a deck, and hence, the job.

G. Output — Plots, Prints, and Motion Pictures

The KACHINA output is in the usual two forms, visual information on 35-mm microfilm or 16-mm motion-picture film and printed information on microfilm or fanfold paper. Both forms are provided automatically in cycles 0 and 1, and thereafter at intervals specified by DTØ and DTØC in the input data. The microfilm plots are generally the most immediately useful output, and they are made on the III FR-80 or the S-C 4020 computer output microfilm devices. As many as 16 plots are provided in the basic package, four particle plots, velocity vector plots of both fields, and contour plots of θ, ρ'_{v1}, ρ'_{d1}, ρ'_v, ρ'_d, I_v, I_d, p, K, and E.

The particle plots are made by plotting the x and y coordinates of all particles, and they are provided automatically when particles are used. Of the four different particle plots, the first is a composite plot of all particles in which a different plotting symbol or color is used for each type of particle. This is followed by a set of three individual particle plots, for each of which KACHINA sorts through the particles and plots only those tagged to correspond to the currently specified type.

Whereas particle plots are useful in following the motions of the components of each field, they cannot convey complete information on the flow details. Velocity vector plots are useful because they show at a glance both the direction of flow and the relative magnitude of the velocities. They are provided separately for both the vapor and droplet fields. Vectors are plotted as if originating at each cell center, denoted by a "+," and their length and direction are proportional to the velocity components. If (x_1, y_1) are the coordinates of the center of cell (i,j), the coordinates of the vector end points (x_2, y_2) are given by

$$x_2 = x_1 + \left(\frac{u^j_{i+\frac{1}{2}} + u^j_{i-\frac{1}{2}}}{2}\right) \text{DRØU} \quad ,$$

and

$$y_2 = y_1 + \left(\frac{v^{j+\frac{1}{2}}_i + v^{j-\frac{1}{2}}_i}{2}\right) \text{DRØU} \quad ,$$

where u and v refer either to (u_v and v_v) or (u_d and v_d), and DRØU is a scaling coefficient defined as

$DRØU = 0.9\ \delta r\ (VEL_{max})$.

This coefficient is recalculated for each individual velocity-vector plot, and it scales the length of a vector drawn for the largest $|u|$ or $|v|$ velocity in the system at that instant, VEL_{max}, to be 9/10ths of the radial cell dimension δr. This method ensures that the vectors are always of reasonable length, regardless of velocity magnitudes. The plot is omitted if there are no significant velocities $\left(VEL_{max} < 10^{-10}\right)$ in a particular field.

Contour plots are drawn for any cell-centered quantity stored in CQ, and they are composed of connected vector segments joining points of equal value, just as the lines on a contour map join points of equal elevation. At present, the plots are all linear in contour increment.

In addition to the various plots described above, three different types of numerical listed data are provided.

The "long print" is a complete numerical listing of the principal field variables over the entire mesh. Two lines containing the i and j and 16 field quantities are given for each cell. They appear as follows. On the first line: $i, j, \theta_i^j, (u_v)_{i+\frac{1}{2}}^j, (v_v)_i^{j+\frac{1}{2}}, (I_v)_i^j, (\rho'_{v1})_i^j, (\rho'_v)_i^j, D_i^j, p_i^j$. On the second line: $E_i^j, (u_d)_{i+\frac{1}{2}}^j, (v_d)_i^{j+\frac{1}{2}}, (I_d)_i^j, (\rho'_{d1})_i^j, (\rho'_d)_i^j, A_i^j, K_i^j$.

The "short print" is a four-line listing of sums over the mesh of mass(M), momentum (MØM) in the r and z directions, internal energy (IE), kinetic energy (KE), and total energy (E). In addition to first specifying the current problem time and cycle number, the 4 lines provide 20 summations in the following order.

$$\begin{array}{lllll}
\Sigma\theta & \Sigma(MØM_r)_{v+d} & \Sigma(MØM_z)_v & \Sigma(MØM_z)_d & \Sigma(MØM_z)_{v+d}, \\
\Sigma(M_v)_1 & \Sigma(M_d)_1 & \Sigma IE_v & \Sigma IE_d & \Sigma IE_{v+d}, \\
\Sigma(M_v)_2 & \Sigma(M_d)_2 & \Sigma KE_v & \Sigma KE_d & \Sigma KE_{v+d}, \\
\Sigma(M_v)_{1+2} & \Sigma(M_d)_{1+2} & \Sigma E_v & \Sigma E_d & \Sigma E_{v+d}.
\end{array}$$

This short print is provided every cycle if the input variable ISPR = 2, or only on those cycles that have a long print if ISPR = 1. LPR, another variable in the input data, has primary control of the destination of both the long and short prints, where:

LPR = 1 gives prints on microfilm only,
LPR = 2 gives prints on microfilm and fanfold paper,
LPR = 3 gives prints on fanfold paper only.

If LPR = 0, both long and short prints are omitted, and no alphanumeric writing of any kind appears on microfilm. LPR = 0 is intended for motion picture use, and the only microfilm output is the first (complete) particle plot. For movies, KACHINA bypasses the automatic δt that normally would begin after cycle 10, and the user must be sure to choose an input δt that is at least as small as the minimum δt required at any time during the course of the problem, usually determined by a preceding run with normal output, and also to set DTØ = δt and DTØC $\geq$ TWFIN. For color processing, the chosen colors are specified by CØLØUR(N), N = 1, 3 in the input data, as described in Sec. II.F. The code is easily altered to provide some plot other than this particle plot for the movie or to have a frame shared by several different types of plots.

Finally, a one-line print is provided on fanfold paper every cycle, regardless of the LPR setting, and also on microfilm if LPR = 1 or 2. This line contains the following nine quantities.

T	is the current problem time.
CYC	is the current cycle number.
DT	is the current δt.
CP	is the current central processor (CP) clock time.
GRINDS	$= \delta CP/(\bar{I}*\bar{J})$, the elapsed CP time for the cycle just completed, divided by the total number of cells. The CP time per cell per cycle is a useful indicator of the code's computing efficiency.
ITP	is the number of iterations required for convergence in the preceding Phase-2 pressure iteration.
CELLS	is the number of cells that failed to converge in the pressure iteration, and is zero unless the iteration failed to converge and was cut off at ITERS = 100.
ITRØ CELLS	The analogous information from the preceding Phase-2 $\tilde{\rho}'_{v1}$ iteration.

H. Tape Dump and Restart

Tape dumps are staged out as Fileset 8 in the control region under influence of the quantities T20MD and/or TLIMD, as described in Sec. II.F. The variables dumped are the contents of the SCM common KSC and also the LCM block, if cell data have been transferred to LCM storage, as described in Secs. II.C and II.D and Appendix C.

A tape restart is performed by staging in the dump tape as Fileset 7. The input deck consists of an IBAR = 0 data card, where JBAR = the dump number on the tape and is used as a check.

I. The Common Block KSC

The following list provides the names, descriptions, and sources of all quantities in the SCM CØMMØN/KSC/ in the (0,0) overlay. This common is of fundamental importance in communication among the various overlays and their subroutines. By design, it contains all the SCM-based information that must be maintained from cycle to cycle, as it is the SCM portion of the tape-dump data.

The sources in the list are keyed to the following symbols.

I = Supplied as part of the standard input data. The parenthetical symbol that follows I specifies where this quantity is read,
0 = (0,0) Main Program,
L = (0,0) Subroutine LØØP,
S = (1,0) Subroutine SETUP,
H = (2,0) Subroutine HYDRØ.

Multiple sources indicate that the quantity is recalculated.

NAME	DESCRIPTION	SOURCE
AA	Dummy word, always the first word in the CØMMØN.	---
AASC	Cell storage, appears in this CØMMØN for SCM version only.	---
AKINFI	A_∞, K_∞ for sudden incompressibility. See Sec. I.F.	S
AO	α_o, a constant in the convective fluxing. See Secs. I.C and I.D.	I(0)
BDTØDR	$\beta_o\delta t/\delta r$.	S
BDTØDZ	$\beta_o\delta t/\delta z$.	S
BD1	b_{d1}.	I(S)
BD2	b_{d2}. Specific heat constants for the droplet field.	I(S)
BINF	= 1.0 if bottom boundary has specified inflow; = 0.0 otherwise.	S
BØUT	= 1.0 if bottom boundary has continuative outflow; = 0.0 otherwise.	S
BV1	b_{v1}.	I(S)
BV1GM11	$(b_{v1})(\gamma_1 - 1.)$.	S
BV2	b_{v2}.	I(S)
BV2GM12	$(b_{v2})(\gamma_2 - 1.)$.	S
BO	β_o, a constant in the convective fluxing, used in conjunction with α_o.	I(0)
CDR	C_{DR}, drag coefficient appearing in the K equation.	I(0)
CØLØUR	Colors for movie. See Secs. II.F and II.G.	I(S)
C1	The h/min/s on the wall clock when the job began. Printed with D1.	0
DR	δr, the cell size in the radial direction.	I(0)
DRØ2	$\delta r/2$.	S
DRSQ	δr^2.	S
DT	δt, the time step, subject to automatic recalculation.	I(S),H
DTØ	Problem time interval between outputs (plots and prints).	I(S)
DTØC	Problem time at which to change to next DTØ in the set.	I(S)
DTØDR	$\delta t/\delta r$.	S,H
DTØDZ	$\delta t/\delta z$.	S,H
DTØ2	$\delta t/2$.	S,H
DTPØS	δt possible for the cycle, but δt used may be reduced to adjust to output time.	S,H
DZ	δz, the cell size in the axial direction.	I(0)

NAME	DESCRIPTION	SOURCE
DZ02	$\delta z/2.$	S
DZSQ	$\delta z^2.$	S
D1	The month/day/year when the job began. Printed with C1.	0
EM10	$10^{-10}.$	S
EM3	$10^{-3}.$	S
EM6	$10^{-6}.$	S
EPS	ε_1, convergence criterion for both Phase-2 iterations.	I(S)
EPV	ε_v and ε_d, appearing in the F (nonisotropic force) terms.	I(S)
EP9	$10^{9}.$	S
EP10	$10^{10}.$	S
EP20	$10^{20}.$	S
FIBAR	Floating-point equivalent of $\bar{I}$.	S
FIXL	Floating-point frame coordinate for left edge of plots.	S
FIXR	Floating-point frame coordinate for right edge of plots.	S
FIYB	Floating-point frame coordinate for bottom edge of plots.	S
FIYT	Floating-point frame coordinate for top edge of plots.	S
FJBAR	Floating-point equivalent of $\bar{J}$.	S
G	g, gravity felt by both vapor and droplets, ±.	I(S)
GAM1	γ_1. Ratio of specific heats for	I(S)
GAM2	γ_2. the components of the vapor field.	I(S)
GDT	$g\delta t.$	S
GGM11	$(\gamma_{v1})(\gamma_{v1} - 1.)$.	S
GGM12	$(\gamma_{v2})(\gamma_{v2} - 1.)$.	S
GM11	$(\gamma_{v1} - 1.)$.	S
GM12	$(\gamma_{v2} - 1.)$.	S
I	Index i. In C0MM0N because of ENTRY SETIJ in L00P.	---
IABRT	Abort indicator, set to 1 if the setup encounters a $\theta < 0$.	0,S
IALL	JP2*NQI, the total amount of cell storage required.	S
IBAR	$\bar{I}$, the number of interior cells in the r direction.	I(0)
IDT0	Index for DT0 and DT0C tables.	S
IJ	Index for cell (i,j), initialized by L00P.	L
IJM	Index for cell (i,j-1), initialized by L00P.	L
IJP	Index for cell (i,j+1), initialized by L00P.	L
IP1	$\bar{I}$+1, index of rightmost column of interior cells.	S
IP2	$\bar{I}$+2, index of column of exterior cells on the right.	S
ISPR	Short print control, described in Sec. II.G.	I(S)
IXL	Integer frame coordinate for left edge of plots.	S
IXR	Integer frame coordinate for right edge of plots.	S
IYB	Integer frame coordinate for bottom edge of plots.	S
IYT	Integer frame coordinate for top edge of plots.	S
J	Index j, in C0MM0N because of ENTRY R1R0W in L00P.	---
JBAR	$\bar{J}$, the number of interior cells in the z direction.	I(0)
JNM	Job name identification assigned by the operating system.	0
JP1	$\bar{J}$+1, index of the topmost row of interior cells.	S
JP2	$\bar{J}$+2, index of row of exterior cells along the top.	S
JRIGID	Number of rigid cells up the right boundary; continuative above this point.	I(S)

NAME	DESCRIPTION	SOURCE
JTØP	= JP1 if top is rigid, = JP2 if top is outflow; controls $\tilde{V}$ initialization in Phase 2.	S
JX1	1+NQ, first word address (fwa) of col. 2, row j=1 (or 1/3 if LCM), set by LØØP as an index.	S
JX2	J2+NQ, fwa of col. 2, row j=2 (or 2/3 if LCM), set by LØØP as an index.	S
JX3	J3+NQ, fwa of col. 2, row j=3 (or 3/3 if LCM), set by LØØP as an index.	S
J2	1+NQI, fwa of col. 1 of row j=2/3, used by LØØP, LCM version only.	S
J3	J2+NQI, fwa of col. 1 of row j=3/3, used by LØØP, LCM version only.	S
KD	k_d, heat conduction coefficient for the droplet field.	I(S)
KDØDRSQ	$k_d/\delta r^2$.	S
KDØDZSQ	$k_d/\delta z^2$.	S
KV	k_v, heat conduction coefficient for the vapor field.	I(S)
KVØDRSQ	$k_v/\delta r^2$.	S
KVØDZSQ	$k_v/\delta z^2$.	S
LCM	= 1 if LCM is used for cell storage, = 0 if SCM is used for cell storage.	L
LPR	Determines output options on film and printer.	I(S)
MUSTPR	Number of cells failing to converge in pressure iteration.	H
MUSTRØ	Number of cells failing to converge in $\tilde{\rho}'_{v1}$ iteration.	H
NAME	Problem identification from columns 2-80 of input card No. 2.	I(S)
NCYC	Number of calculation cycles completed.	S,H
NLC	Number of words to tape dump from LCM cell storage, if used.	S
NPTØT	Total number of particles in the system at a given instant.	S,H
NQ	Number of quantities, or storage words, per cell.	0
NQI	NQ*IP2, the number of words for one full row of cells.	S
NQI2	NQI + NQI, the number of words for two full rows of cells.	S
NQL	NQ*(1 - LCM), used as index adjustment if SCM version.	S
NQ2	NQ + NQ, the number of words in two cells, used by LØØP, SCM version.	S
NQ2L	NQL + NQL, used as index adjustment if SCM version.	S
NSC	Number of words in this SCM common, for tape dump.	S,H
NUMIT	Number of iterations required for Phase-2 pressure convergence.	H
NUMRØ	Number of iterations required for Phase-2 $\left(\tilde{\rho}'_{v1}\right)$ convergence.	H
NUMTD	Number of the next tape dump.	S,H
NUV	ν_v, kinematic viscosity coefficient for the vapor field	I(0)
NUV3	$3*\nu_v$, appears in K equation in Phase 1.	S
NVAP	= 1 if $\rho_{v1} = 0$ and $\rho_{v2} > 0$, = 2 if $\rho_{v1} > 0$ and $\rho_{v2} = 0$, = 3 if ρ_{v1} and $\rho_{v2} > 0$.	S
ØMBAS	$-\omega_p \delta r^2 \delta z^2 / 2(\delta r^2 + \delta z^2)$, base of complete ØMBSPL.	S
ØMBSPL	ØMBAS/δt, stored for $(-\omega_p \beta_p)_i^j$ when $\theta_i^j < \theta_o$.	S,H
ØMP	ω_p, the Phase-2 pressure iteration relaxation coefficient.	I(S)
ØMRØ	ω_ρ, the Phase-2 $\left(\tilde{\rho}'_{v1}\right)$ iteration relaxation coefficient.	I(S)
R	Exchange function for heat transfer between the two fields.	I(S)
RCØNT	Right boundary table, on index J; = 0.0 where rigid, = 1.0 where outflow.	S
RDR	$1/\delta r$.	S
RDRSQ	$1/\delta r^2$.	S
RDT	$1/\delta t$.	S
RDZ	$1/\delta z$.	S

NAME	DESCRIPTION	SOURCE
RDZSQ	$1/\delta z^2$.	S
RI	Table of r_i's, of length IP2.	S
RIBAR	Reciprocal of $\bar{I}$.	S
RIBJB	Reciprocal of $(\bar{I}*\bar{J})$, used in control region grind calculation.	S
RIP	Table of $r_{i+\frac{1}{2}}$'s, of length IP2.	S
RJBAR	Reciprocal of $\bar{J}$.	S
RØ1	ρ_1, microscopic material density of droplet component 1.	I(S)
RØ2	ρ_2, microscopic material density of droplet component 2.	I(S)
RØVPIN	$(\rho_v')_{j=1}$, for specified inflow bottom boundary, = THIN*(RØVIN1 + RØVIN2).	S
RØVPIN1	$(\rho_{v1}')_{j=1}$, for specified inflow bottom boundary, = THIN*RØVIN1.	S
RØVO	ρ_{vo}, specified vapor density for incompressible flow.	I(O)
RPAR	r_p, the mean linear dimension of a droplet.	I(O)
RPCDR	$r_p C_{DR}$, = RPAR*CDR, appears in K equation.	S
RPCØF	$3/2r_p^2$ = 1.5/RPAR**2, appears in K equation.	S
RRI	Table of $1/r_i$'s, of length IP2.	S
RRIP	Table of $1/r_{i+\frac{1}{2}}$'s, of length IP2.	S
RRIDR	Table of $1/(r_i\delta r)$'s, of length IP2.	S
RRØ1	$1/\rho_1$.	S
RRØ2	$1/\rho_2$.	S
R2DR	$1/(2\delta r)$.	S
R2DZ	$1/(2\delta z)$.	S
SIEVIN	$(I_v)_{j=1}$, for specified inflow on bottom boundary.	I(S)
SQMO	M_o^2 in Mach No. ratio in Phase-1 $\tilde{p}$ initialization, usually = $(0.5)^2$.	S
T	t, the problem time.	I(S),H
THIN	[illegible]$_{j=1}$, for specified inflow on bottom boundary.	I(S)
THO	θ_o, the critical value for sudden incompressibility, usually = 0.02.	S
TLIMD	= 1.0 to force a tape dump and RETURN before time limit.	I(S)
TØP	= 0.0 if top boundary is rigid, or = 1.0 if top is outflow.	S
TØUT	The next problem output time for plots/prints.	S,H
TWFIN	Time-When-To-Finish: calculation completed when $t \geq$ TWFIN.	I(S)
T20MI	= 1.0 to force tape dumps every 20 min of CP time.	I(S)
VØL	Table of V_i's, of length IP2, where $V_i = 2\pi r_i \delta r \delta z$.	S
VØLR	Table of $V_{i+\frac{1}{2}}$'s, of length IP2, where $V_{i+\frac{1}{2}} = 2\pi r_{i+\frac{1}{2}} \delta r \delta z$.	S
VVIN	$(v_v)_{j=1}$, the specified velocity at inflow bottom boundary.	I(S)
XCØNV	Plotting factor, converts x's from problem units to 4020 units.	S
XL	= 0.0, the left edge of the mesh, for plots.	S
XP	Storage block for x coordinates of marker particles.	S,H
XR	= $\bar{I}*\delta r$, the right edge of the mesh, for plots.	S
YB	= 0.0, the bottom edge of the mesh, for plots.	S
YCØNV	Plotting factor, converts y's from problem units to 4020 units.	S
YP	Storage block for y coordinates of marker particles.	S,H
YT	= $\bar{J}*\delta z$, the top edge of the mesh, for plots.	S
ZZ	Dummy word, always the final word in the CØMMØN.	---

J. Microfilm Plots — Scaling and Subroutine CALLs

The microfilm plots discussed in Sec. II.G are generally the most useful form of output from our fluid dynamics codes, as they often can quickly convey information about a flow process that would be less readily grasped by examining numerical listings alone.

Our original Computer Output Microfilm (COM) device is the S-C 4020, which has a matrix of 1024 by 1024 raster points on the CRT face, as is shown in Fig. 13. Usefulness of the 4020 has been increased by addition of a set of three color filters between the tube face and the camera, individually movable under control commands. The III FR-80 COM device has a resolvable matrix of 10K by 10K points which allows more accurate plotting, and it is also a programmable computer in its own right. For our relatively simple plots, whatever COM is assigned to our offline film output is treated as a 4020, the FR-80 becoming a 4020 simulator when used in this manner. Therefore, the following description of our plot scaling is based on the 4020 film frame of Fig. 13.

Note from Fig. 13 that the origin of the x-y coordinate system lies at the upper left corner of the frame and the values of the two integer indices increase to the right and down. This coordinate system obviously does not match that of the fluid dynamics computing mesh shown in Figs. 1 and 8, where the origin is at the lower left corner. A conversion from physical mesh position to a corresponding 4020 frame position is required for all plotting. In KACHINA, the left, bottom, right, and top edges of the physical mesh are specified by:

$XL = YB = 0.0$,

$XR = \bar{I}\delta r$,

$YT = \bar{J}\delta z$,

and their counterparts in 4020 coordinates are given by the integers IXL, IYB, IXR, and IYT, calculated in accordance with the following considerations. First, we reserve areas across the top and bottom of the frame for plot identification, problem time, cycle number, and so forth. These are indicated by the shaded areas in Fig. 14. (The technique for generating alphanumeric information in these regions is discussed below.) The unshaded area that remains is 1024 points wide by 900 points high, although we consider the available width to the 1022 points, to ensure frame separation. Second, within this rectangular region, we maximize the size of the plot that is drawn, while maintaining its true physical proportion of height to width. Thus, if the physical area encompassed by the computing mesh is higher than it is wide, $XR < YT$, the resulting plot occupies the region exemplified by the fine shading in Fig. 15. The coordinates in this case are given by:

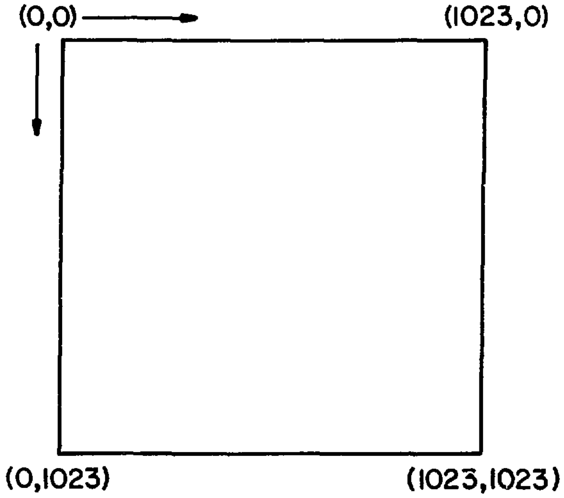


Fig. 13. The 4020 grid is a reflection of the first quadrant, and contains a matrix of 1024 by 1024 raster points, with the origin located at the upper left corner. x increases to the right, and y increases downward.

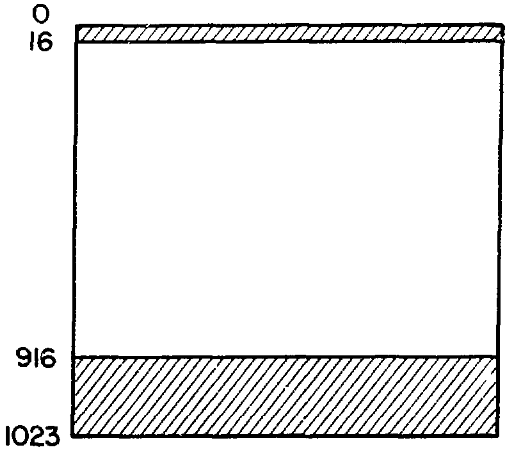


Fig. 14. The area in the center, available for plotting, encompasses 1024 by 900 raster points. The shaded areas are reserved for labeling, allowing two lines at the top and six at the bottom.

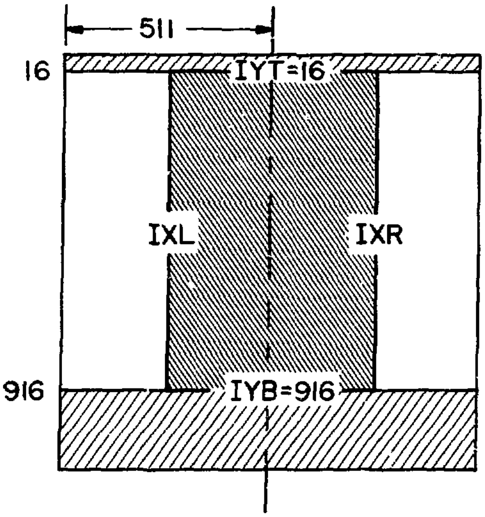


Fig. 15. The plotting area when the computing mesh is higher than it is wide, XR < YT.

$$FIXL = 511. - 450.\left(\frac{XR}{YT - YB}\right) ,$$

$$FIXR = 511. + 450.\left(\frac{XR}{YT - YB}\right) ,$$

$$FIYB = 916. \; ,$$

$$FIYT = 16.$$

Conversely, if the computing area is wider than it is high, XR > YT, the plot occupies the region exemplified by the fine shading in Fig. 16. Here the coordinates are given by:

$$FIXL = 0. \; ,$$

$$FIXR = 1022. \; ,$$

$$FIYB = 916. \; ,$$

$$FIYT = 916. - 1022.\left(\frac{YT - YB}{XR}\right) .$$

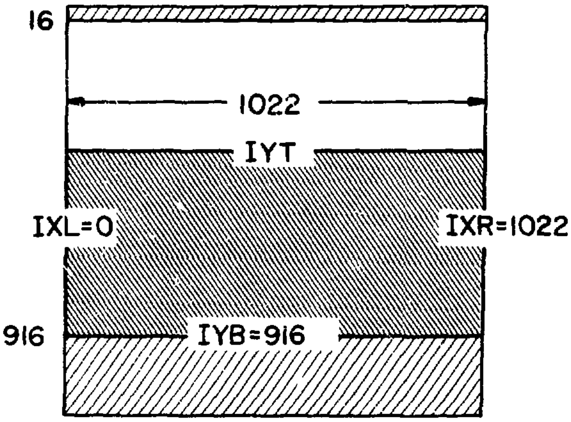


Fig. 16. The plotting area when the computing mesh is wider than it is high, XR > YT.

In either case, the equivalent integers IXL, IXR, IYB, and IYT are then simply set directly from FIXL, FIXR, FIYB, and FIYT. The "4020 Setup" in SETUP calculates these eight quantities, and then calculates the two conversion factors that will be required for translating physical mesh coordinates to 4020 frame coordinates. These are given by the ratios

XCØNV = (FIXR - FIXL)/(XR - XL) ,

and

YCØNV = (FIYT - FIYB)/(YT - YB) .

A physical mesh coordinate is multiplied by the appropriate factor to convert it, and this product is then added to FIXL or FIYB and the sum is converted to an integer to locate the position on the 4020 frame of Fig. 13. A conversion subroutine is available that would handle this task for us, but it is more efficient for KACHINA to do it, thus avoiding recalculation of these two ratios whenever a point is plotted or a vector segment is drawn.

A set of local software subroutines provided by the Computer Sciences and Services Division of the Los Alamos Scientific Laboratory handles the communication between the problem program and the CØM devices by producing 4020-format commands. KACHINA uses a number of these subroutines, which are accessed by the following FØRTRAN calling sequences.

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

CALL FRAME (IXL, IXR, IYB, IYT) draws a rectangular outline of the computing mesh. Two horizontal axes are drawn through IYT and IYB from IXL to IXR, and two vertical axes are drawn through IXL and IXR from IYT to IYB.

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

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

CALL CØLØR (c) controls the filter selection for color processing; c is a floating-point variable with a value in the range 0.0 to 4.0. Single-filter selections are determined as follows:

c = 0.0 No filter (white),
c = 1.6 Yellow filter,
c = 2.6 Cyan filter,
c = 3.6 Magenta filter.

The primary colors are obtained by appropriate filter combinations:

c = 1.0 Red (= yellow + magenta),
c = 2.0 Green (= yellow + cyan),
c = 3.0 Blue (= cyan + magenta).

Several additional subroutines are available for writing alphanumeric information on the frame. Whereas some of these write large characters composed of dot patterns, the basic CRT tube has a set of small alphanumeric characters that can be generated in the "typewriter" command mode, and it is this latter form that is used in KACHINA. In typewriter mode, the 4020 frame of Fig. 13 is composed of 64 lines of 128 characters each. Each character occupies a rectangular region 8 raster points wide by 16 high. The first character of a line is treated as a carriage control, completely analogous to line printer use.

CALL LINCNT (ℓ) locates the first column of line ℓ, where ℓ ranges from 0 (top line of the frame) through $\ell = 63$ (bottom line of frame), and $\ell = 64$ advances the film to the top of the next frame. After LINCNT locates the desired starting line position, ordinary formatted WRITE statements generate the actual alphanumeric information. The FORMAT statements are identical to those appropriate for a line printer, provided the 128 character per line restriction is observed. Line advancement is automatic, as on a line printer, until either line 63 has been written on, after which the film is automatically advanced to the top of the next frame, or another CALL LINCNT is issued to specify any desired line (0-63) of the current frame.

CALL EMPTY ensures that if the film buffer contains any words, they are written on the 4020 tape. EMPTY generally appears as the final command to a sequence of 4020 instructions. Thus no residual commands are left in the film buffer, where they would be susceptible to possible loss in the event of a hardware or system failure.

K. Miscellaneous System Subroutine CALLs

KACHINA uses a number of other CALLs to access various local operating system subroutines not directly related to microfilm usage. These are, briefly, as follows.

CALL GETQ (key, q) (Get Quantity) is available for retrieving a variety of job task parameters from the operating system. Here, key is the task parameter identifier, in left-justified display code, and q is the name of the problem program location to which the value of the quantity is returned. In KACHINA, we use two task parameters: KJBN is the job name, composed of 10 symbols, the first 7 symbols from the job card name field, followed by a one-digit input station ID and a two-character job sequence number. KTLM is the time limit specified on the job card, converted to CDC 7600 clock cycles, where one clock cycle = $27.5 * 10^{-9}$ seconds.

CALL DATE1 (D1) stores the current date in display code in the location specified by D1. The form is eight characters, MM/DD/YY, where MM is the month, DD is the day, and YY is the year.

CALL CLØCK1 (C1) stores the current time of day in display code in the location specified by C1. The form is eight characters, HH·MM·SS, where HH is the hour (00 through 23), MM is the number of minutes, and SS is the number of seconds.

CALL SEC0ND (T1) stores the elapsed central processor time for the job in the location specified by T1. This time is expressed as a real number representing seconds, to the nearest thousandth of a second.

CALL AFSREL ( ), or Active Fileset Release, is used in two ways in KACHINA. First, upon a tape restart, it is called to release LCM space the the operating system has devoted to Fileset 7, as soon as that fileset becomes inactive. Second, it is also called to initiate output processing of active filesets 0UT and FILM at each tape dump, assuming that the dump is before job completion and is an intermediate one.

CALL DATAREL ( ), or Data Release, is used to stage a restart dump from the disk onto a physical tape. Otherwise, tape dumps would be accumulated on disk storage during job execution and not actually staged to tape until job completion, leaving them vulnerable to loss in event of hardware or system failure.

If the descriptions of system subroutines provided in Secs. II.J and II.K are inadequate, contact the Los Alamos Scientific Laboratory Computer Sciences and Services Division.

III. REFERENCES

1. F. H. Harlow and A. A. Amsden, "A Numerical Flui Dynamics Calculation Method for All Flow Speeds, J. Comp. Phys. 8, 197 (1971).

2. 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 (1973).

3. 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 (1974).

4. T. D. Butler, O. A. Farmer, and W. C. Rivard, "A Numerical Technique for Transient, Chemically Reactive Flows of Arbitrary Mach Number," manuscript in preparation.

5. J. U. Brackbill and W. [illegible] Pracht, "An Implicit, Almost-Lagrangian Algo[illegible] for Magnetohydrodynamics," J. Comp. Phys. 1[illegible], 455 (1973).

6. F. H. Harlow and A. A. Amsden, "Multi-Fluid Flow Calculations at All Mach Numbers," J. Comp. Phys., accepted for publication.

7. F. H. Harlow and A. A. Amsden, "Numerical Calculation of Multiphase Fluid Flow," J. Comp. Phys., accepted for publication.

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

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

10. C. W. Hirt and J. L. Cook, "Calculating Three-Dimensional Flows Around Structures and Over Rough Terrain," J. Comp. Phys. 10, 324 (1972).

11. A. A. Amsden and F. H. Harlow, "The SMAC Method: A Numerical Technique for Calculating Incompressible Fluid Flows," Los Alamos Scientific Laboratory report LA-4370 (1970).

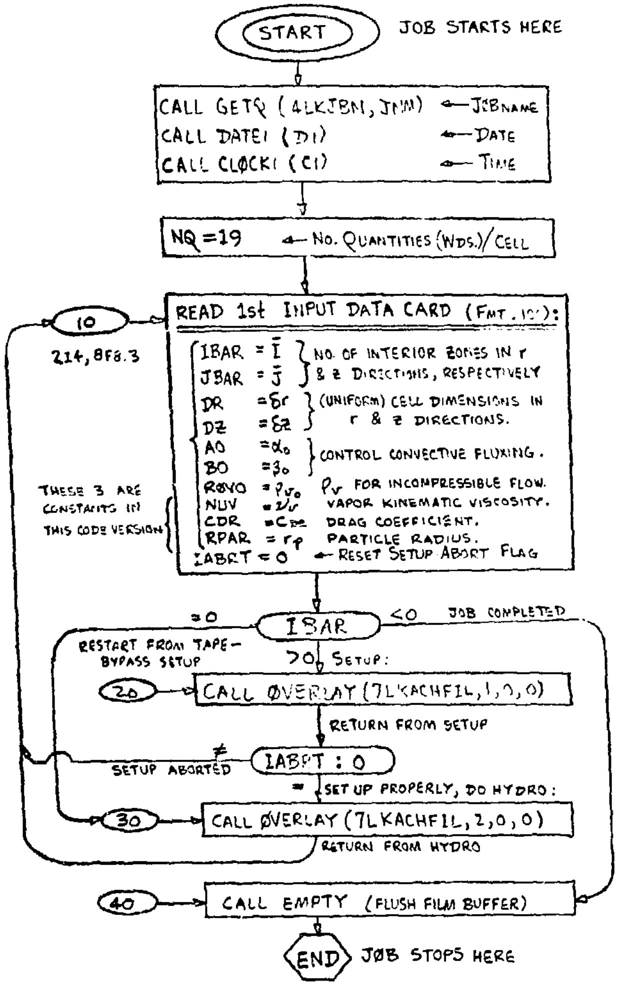
APPENDIX A

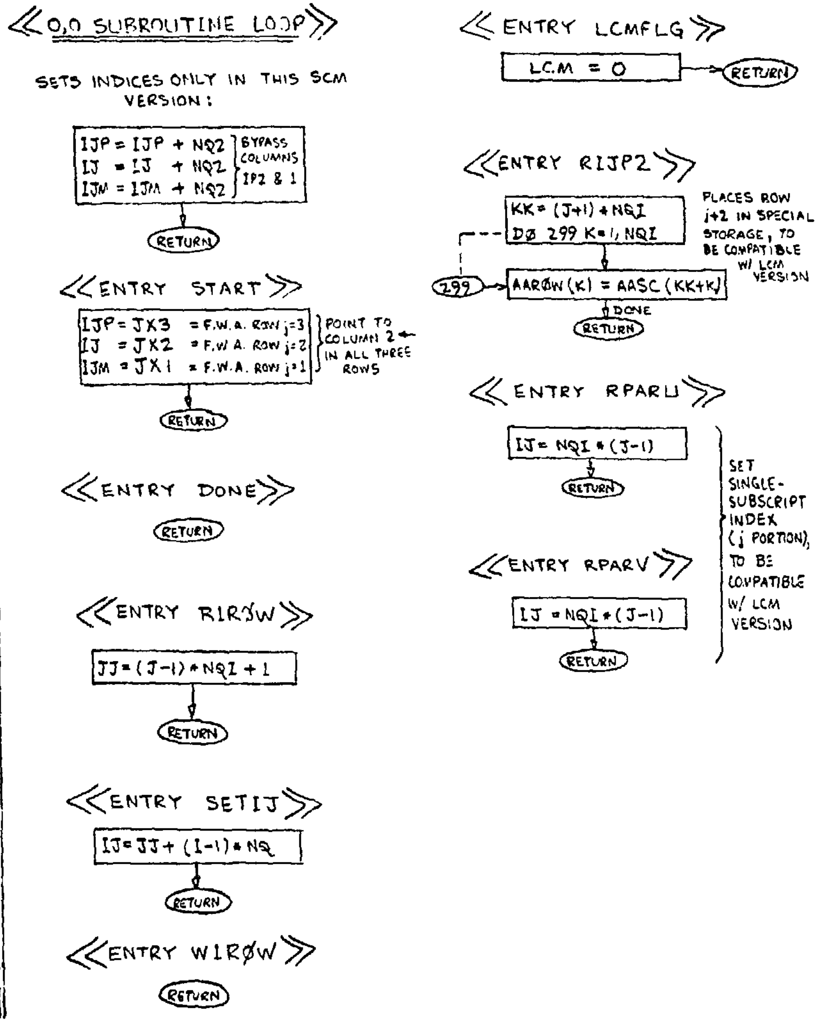
FLOW DIAGRAM FOR THE KACHINA PROGRAM

(June 19, 1974 Status)

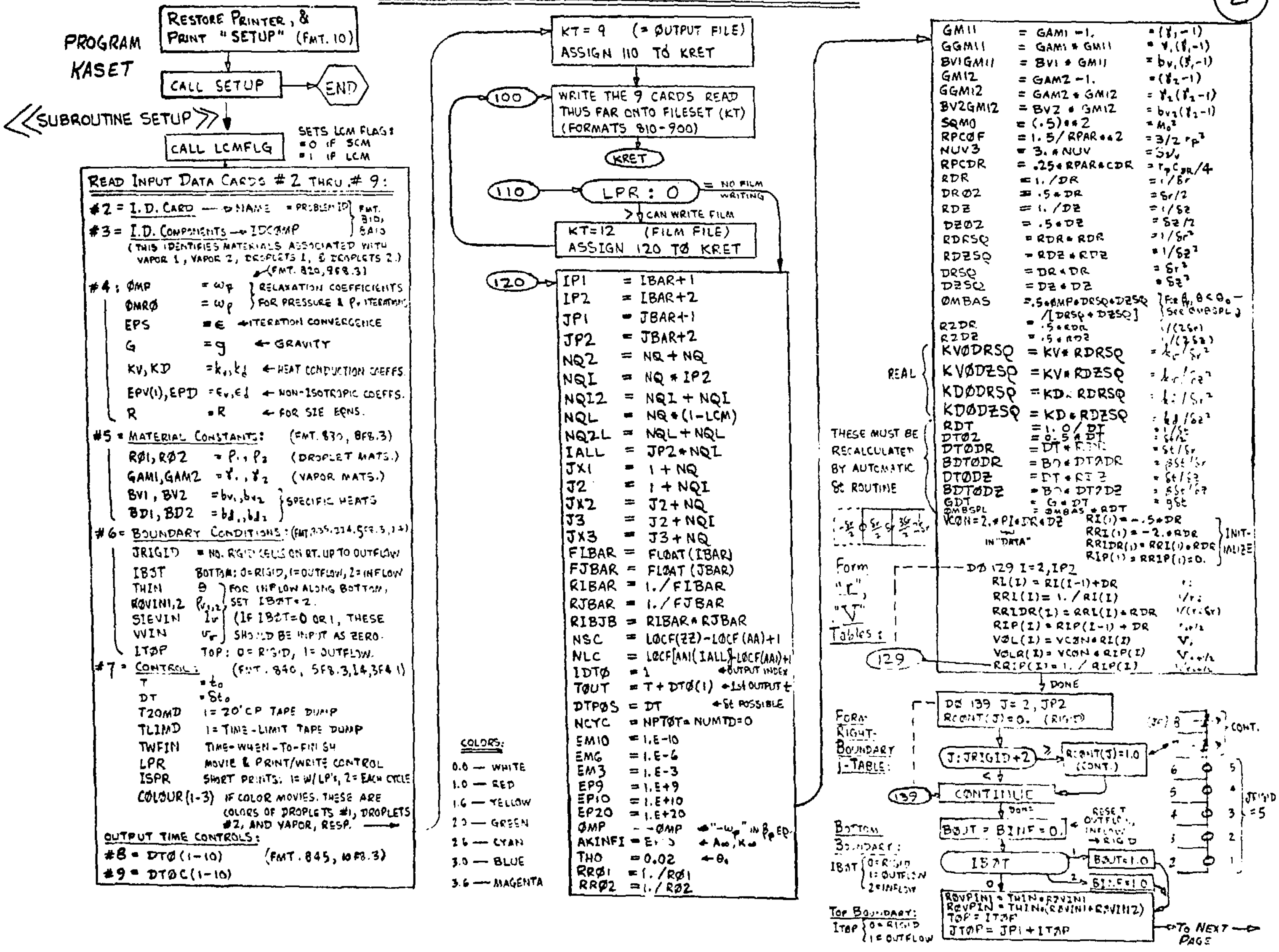
KACHINA — *A MULTI-FIELD HYDRO CODE*

PROGRAM KACHINA – 0,0 OVERLAY:



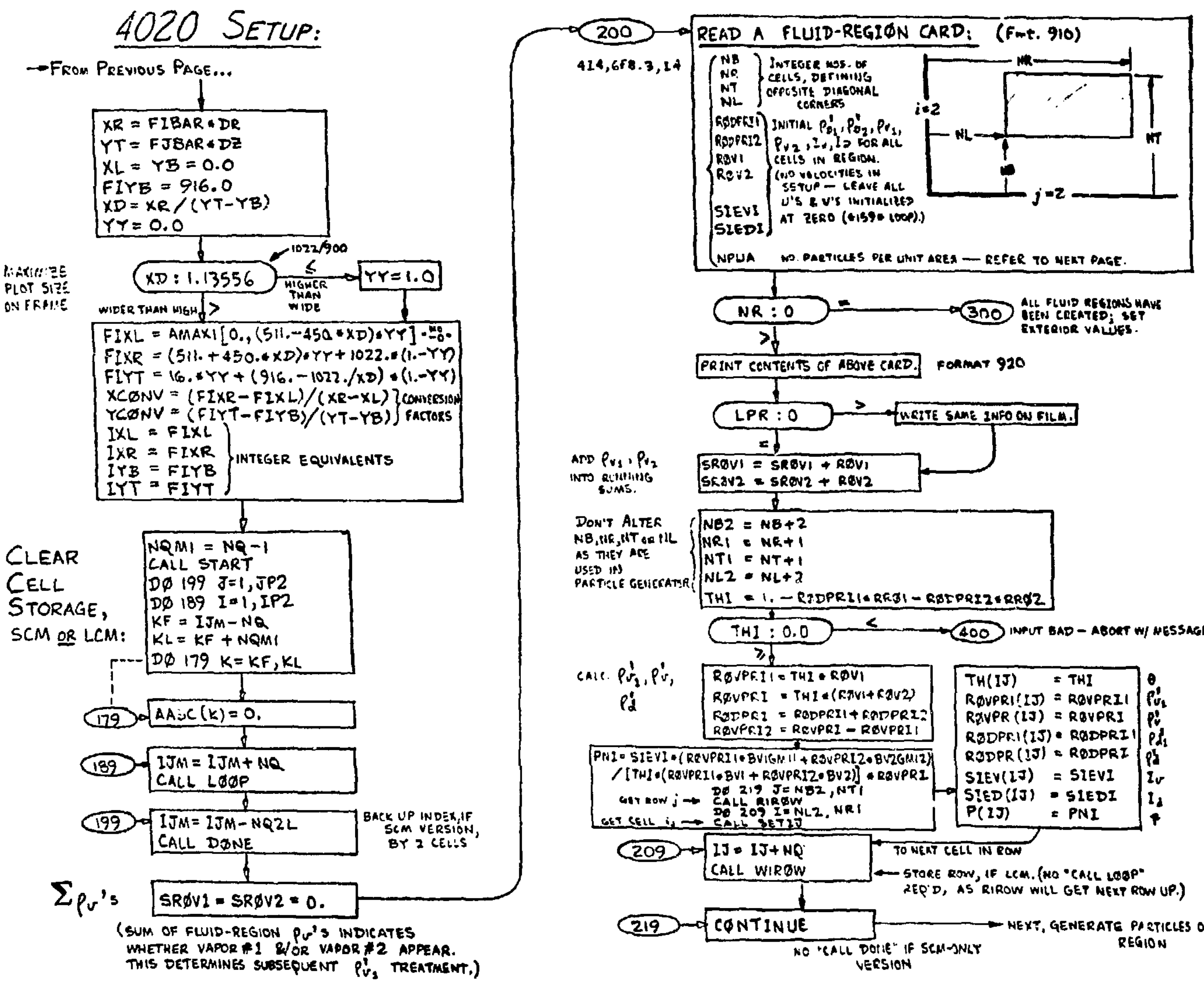


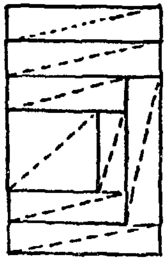
1,0 OVERLAY – THE PROBLEM SETUP:



1,0 OVERLAY - THE PROBLEM SETUP-CONTINUED:

4020 SETUP:

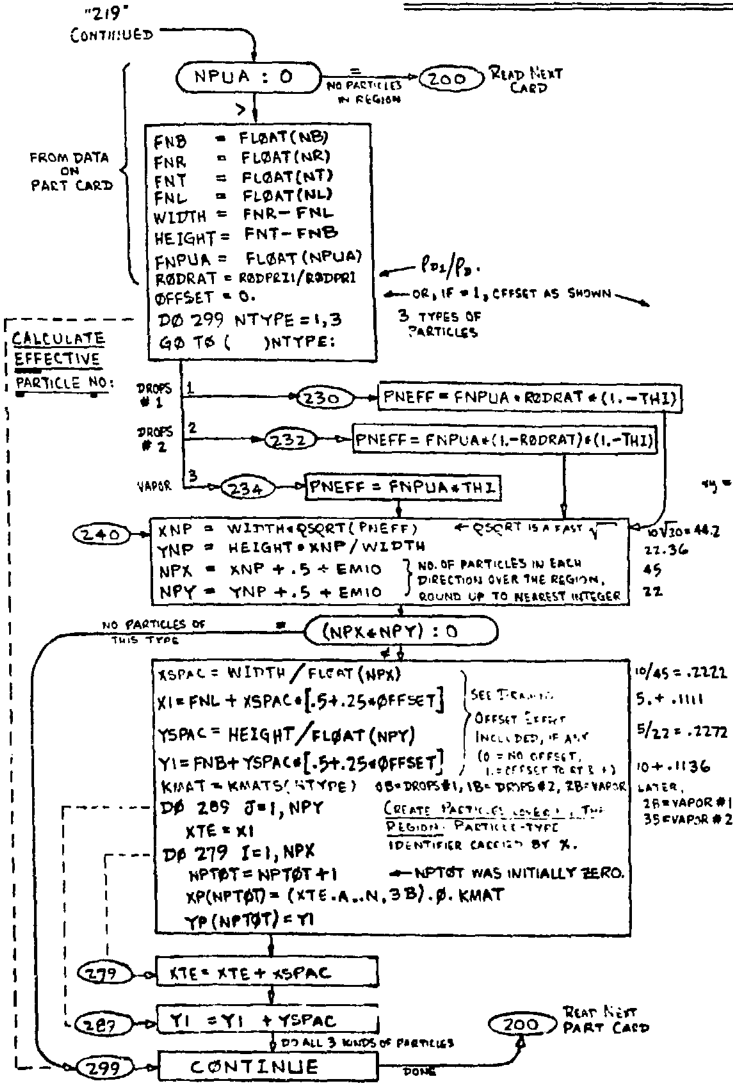




EXAMPLE OF A TYPICAL SETUP — (6 FLUID REGIONS, BUT 8 FLUID-REGION CARDS REQUIRED.)

1,0 OVERLAY—MARKER PARTICLE GENERATION:

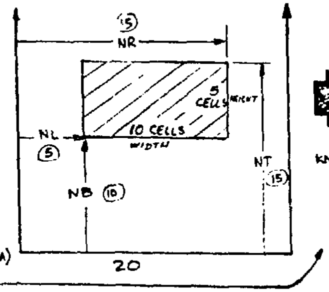
④

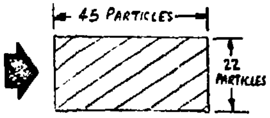


PARTICLE GENERATOR DEALS IN CELL AREA & USES NB-R-T-L

AS AN EXAMPLE, CONSIDER THE REGION ILLUSTRATED.

AREA = 50 CELLS.
IF ~1000 PARTICLES DESIRED IN REGION, THEN SPECIFY NPUA = 20 (PARTICLES/UNIT AREA)





KNOWN: $xy = 1000$, $\frac{10}{5} = \frac{x}{y}$,

$10y = 5x$, $y = 5x/10$. $x\left(\frac{5x}{10}\right) = 1000$

$5x^2 = 10000$, $x^2 = 2000$

$x = 44.72 \rightarrow 45$ $y = \frac{5x}{10} = 22.36 \rightarrow 22$

$45 \times 22 = 990$ particles.

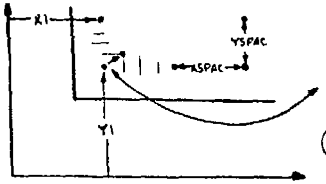
$$\frac{\text{WIDTH}}{\text{HEIGHT}} = \frac{x}{y} \quad \left(= \frac{\text{XNP}}{\text{YNP}}\right)$$

$$y = \frac{x \cdot \text{HEIGHT}}{\text{WIDTH}}$$

$$xy = \frac{x^2 \cdot \text{HEIGHT}}{\text{WIDTH}} = \overset{\text{DESIRED NO. PARTICLES}}{\overline{\text{AREA} \cdot \text{NPUA}}}$$

$$x^2 = \frac{\text{AREA} \cdot \text{WIDTH} \cdot \text{NPUA}}{\text{HEIGHT}}$$

$$x = \sqrt{\frac{\text{WIDTH} \cdot \text{HEIGHT} \cdot \text{WIDTH} \cdot \text{NPUA}}{\text{HEIGHT}}} = \text{WIDTH}\sqrt{\text{NPUA}}$$

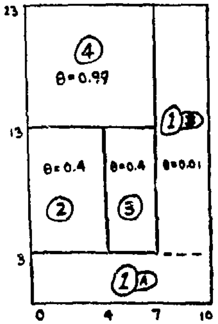


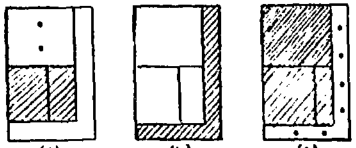
MAY WISH TO OFFSET ONE SET OF PARTICLES. GENERALLY THIS IS UNNECESSARY, AS PNEFF'S GENERALLY DIFFER, AND SEPARATE PLOTS ARE MADE.

(OFFSET = $\frac{\text{XSPAC}}{4}$, $\frac{\text{YSPAC}}{4}$)

TYPE OF PARTICLES:	REGION NO. 1A	1B	2	3	4
DROPS #1:					
PNEFF = $\frac{\rho_{D1}}{\rho_D}(1-\theta)$NPUA	0	0	2.4	2.4	0.04
NPX	0	0	6	5	1
NPY	0	0	15	15	2
DROPS #2:					
PNEFF = $\left(1-\frac{\rho_{D1}}{\rho_D}\right)(1-\theta)$NPUA	3.96	3.96	0	0	0
NPX	20	6	0	0	0
NPY	6	40	0	0	0
VAPOR:					
PNEFF = (θ)·NPUA	0.04	0.04	1.6	1.6	3.96
NPX	2	1	5	4	14
NPY	1	4	13	13	20

TOTAL NO. OF PARTICLES = 930





(a) (b) (c)

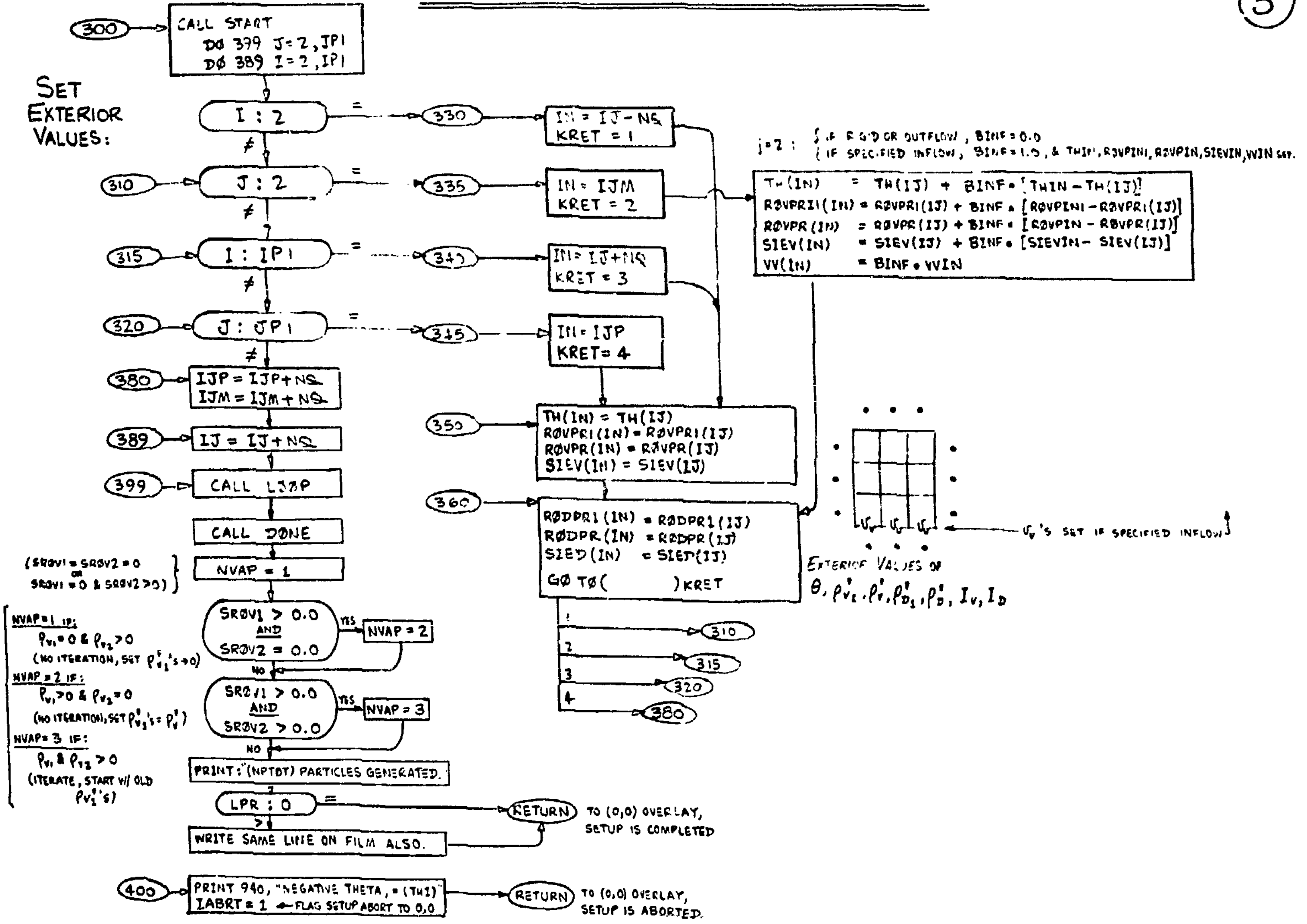
(a) 167 DROPS #1 PARTICLES

(b) 360 DROPS #2 PARTICLES

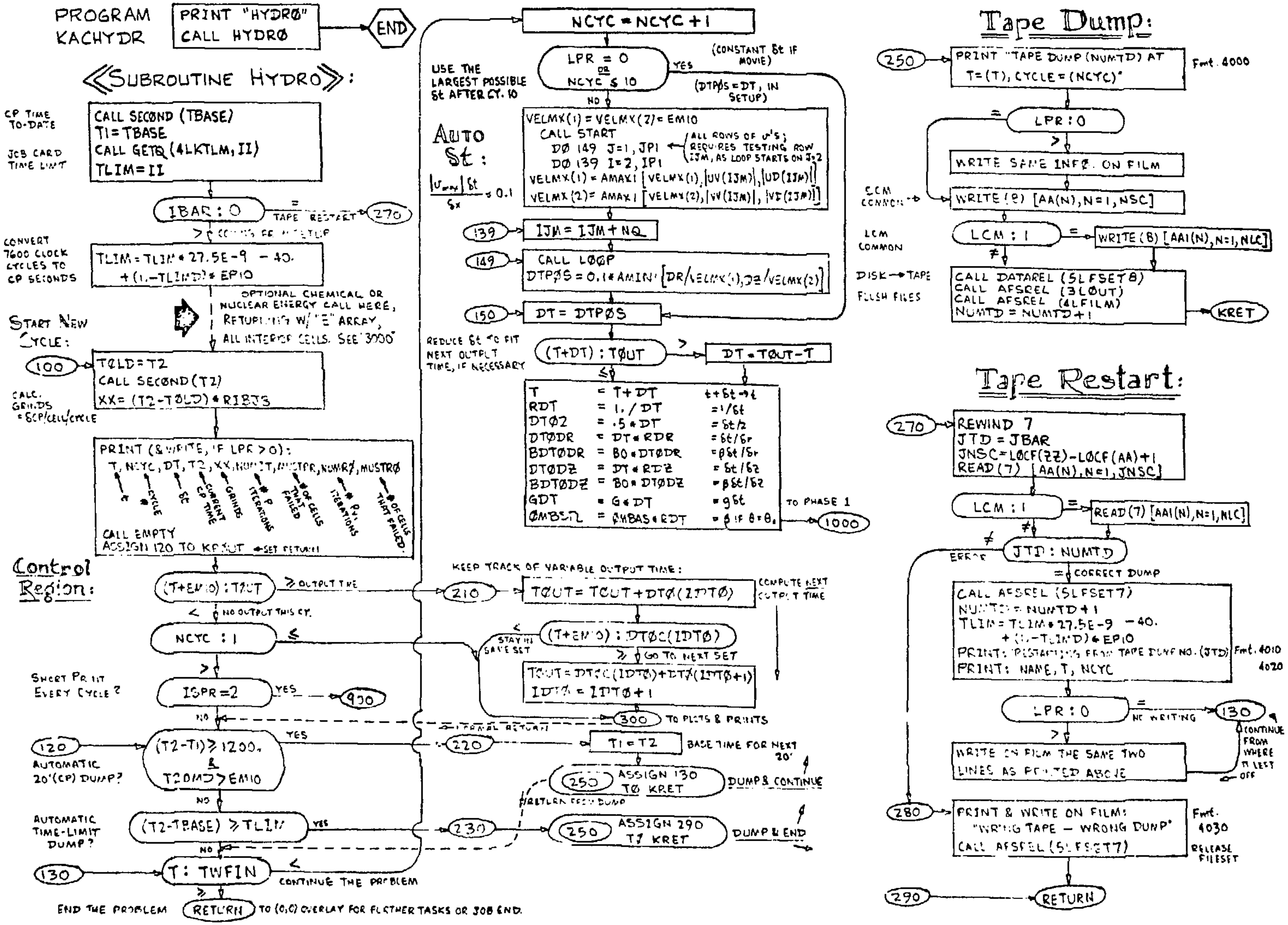
(c) 403 VAPOR PARTICLES

1,0 OVERLAY — COMPLETION OF SETUP:

⑤

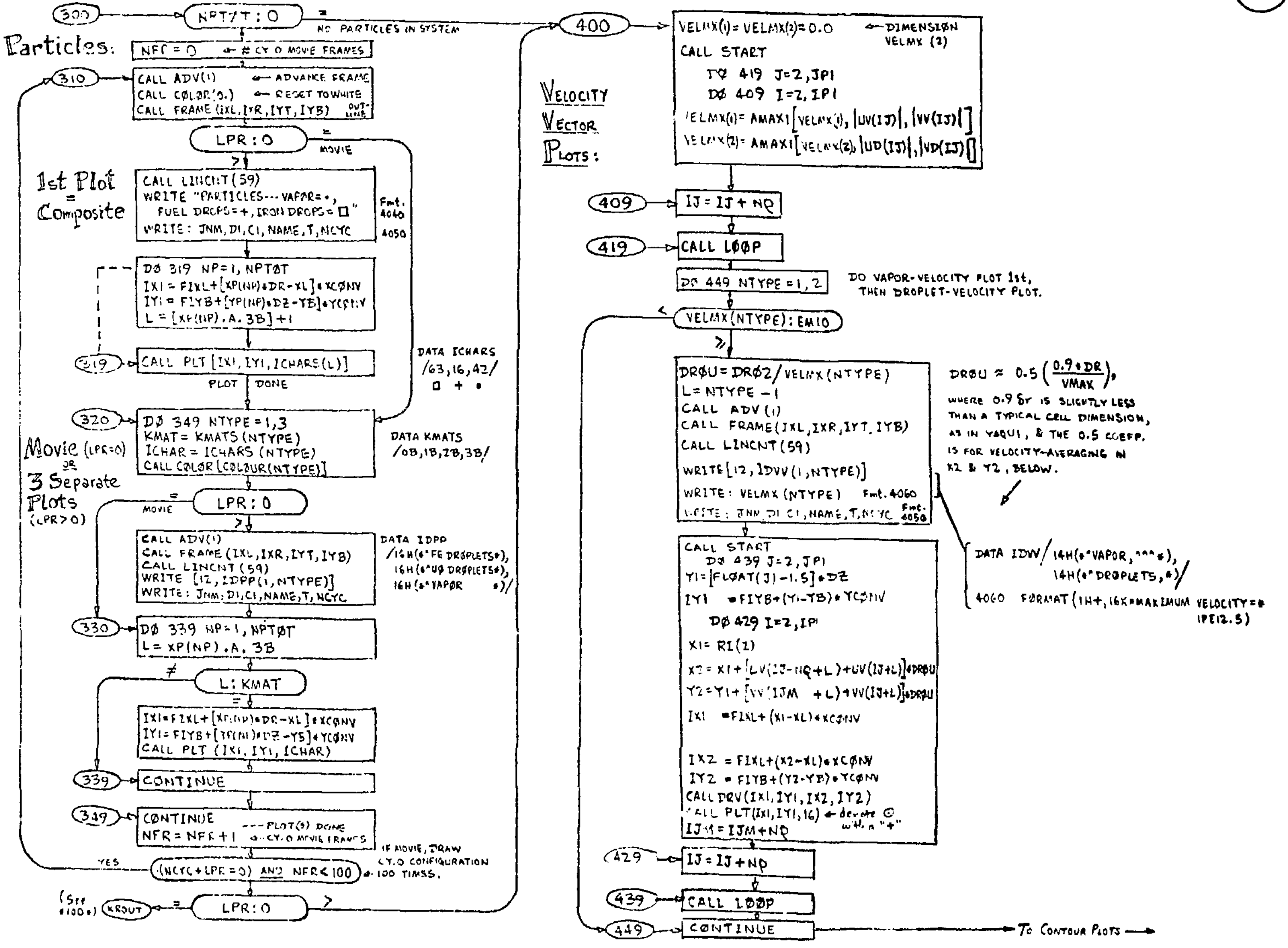


START OF 2,0 OVERLAY — CONTROL REGION:

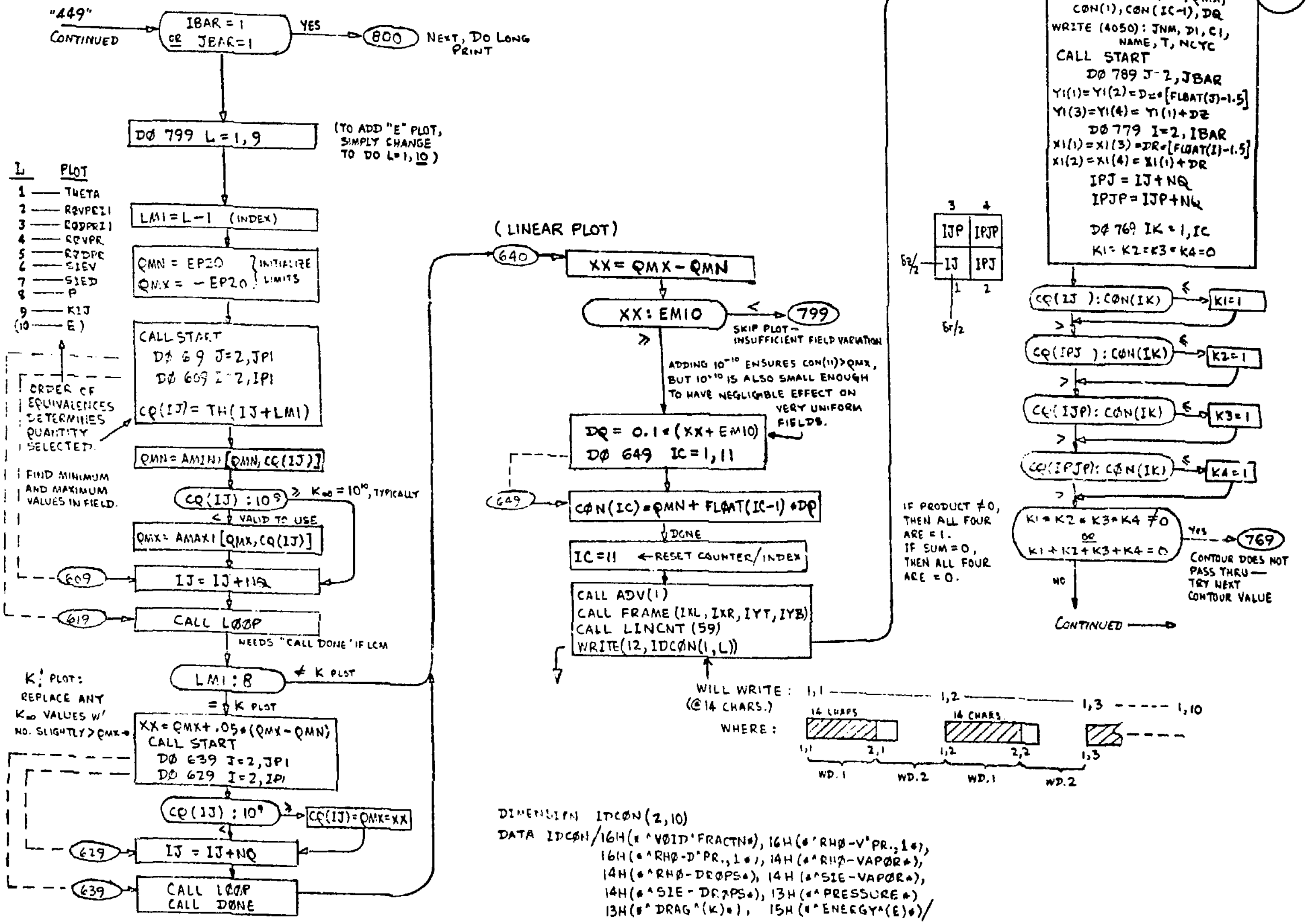


PARTICLE PLOTS & VELOCITY VECTOR PLOTS:

⑦

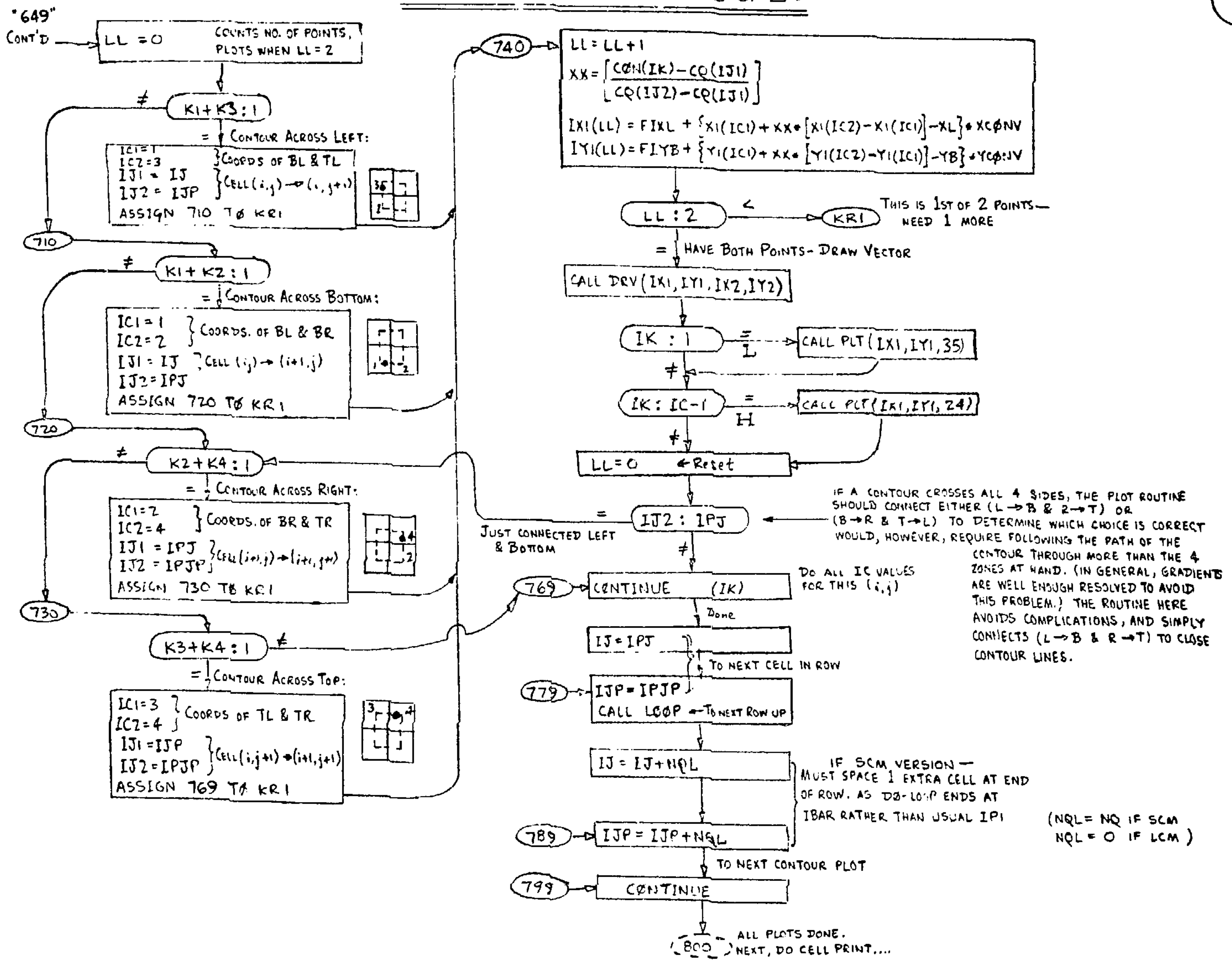


CONTOUR PLOTS – PAGE 1 OF 2:



```
DIMENSION IDCØN(2,10)
DATA IDCØN/16H(*^VØID'FRACTN*), 16H(*'RHØ-V'PR.,1*),
          16H(*^RHØ-D'PR.,1*), 14H(*^RHØ-VAPØR*),
          14H(*^RHØ-DRØPS*), 14H (*^SIE-VAPØR*),
          14H(*^SIE - DRØPS*), 13H (*^PRESSURE*)
          13H(*^DRAG^(K)*),  15H (*^ENERGY^(E)*)/
```

CONTOUR PLOTS — PAGE 2 OF 2:



IF A CONTOUR CROSSES ALL 4 SIDES, THE PLOT ROUTINE SHOULD CONNECT EITHER (L → B & R → T) OR (B → R & T → L) TO DETERMINE WHICH CHOICE IS CORRECT WOULD, HOWEVER, REQUIRE FOLLOWING THE PATH OF THE CONTOUR THROUGH MORE THAN THE 4 ZONES AT HAND. (IN GENERAL, GRADIENTS ARE WELL ENOUGH RESOLVED TO AVOID THIS PROBLEM.) THE ROUTINE HERE AVOIDS COMPLICATIONS, AND SIMPLY CONNECTS (L → B & R → T) TO CLOSE CONTOUR LINES.

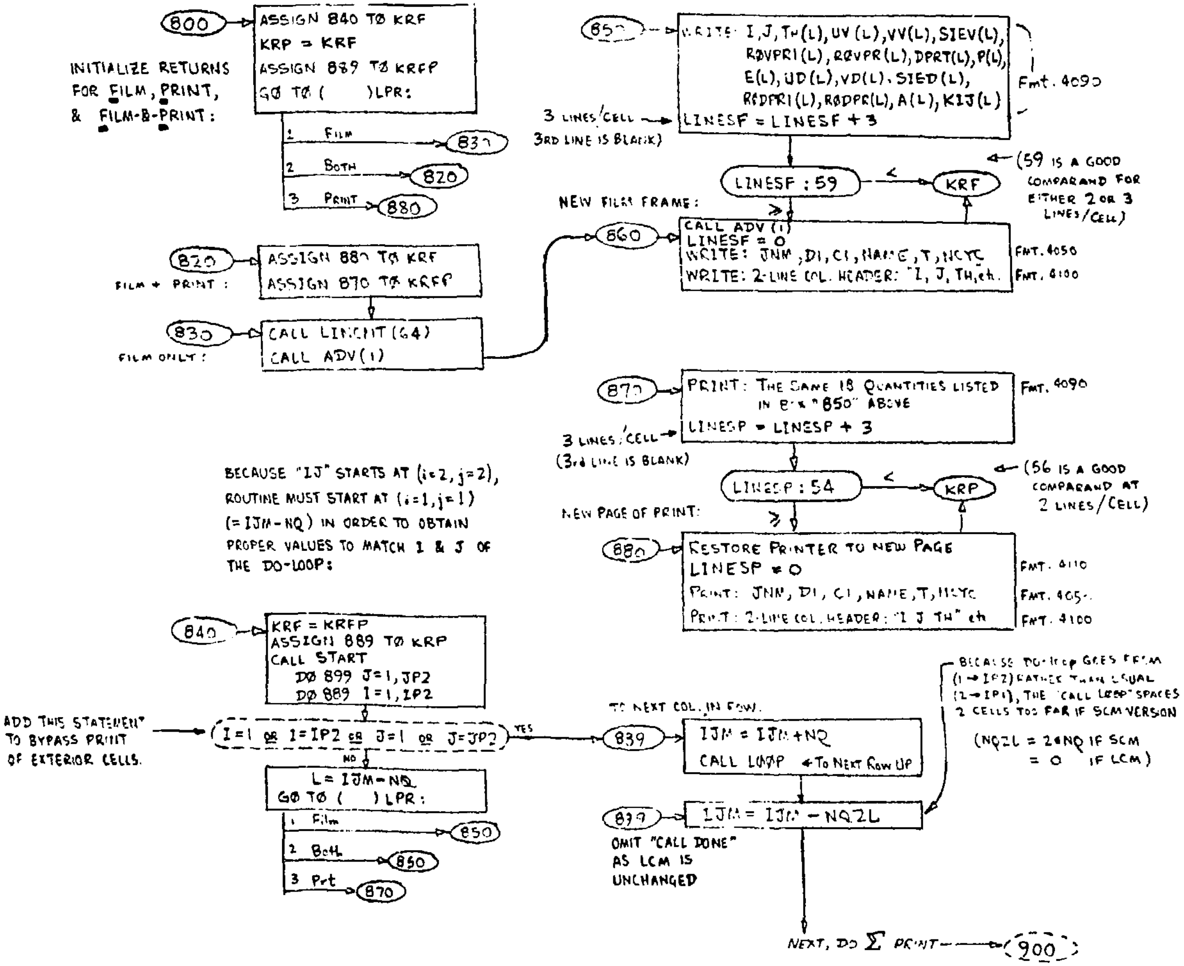
IF SCM VERSION — MUST SPACE 1 EXTRA CELL AT END OF ROW, AS DØ-LOOP ENDS AT IBAR RATHER THAN USUAL IP1

(NQL = NQ IF SCM
NQL = 0 IF LCM)

⑩

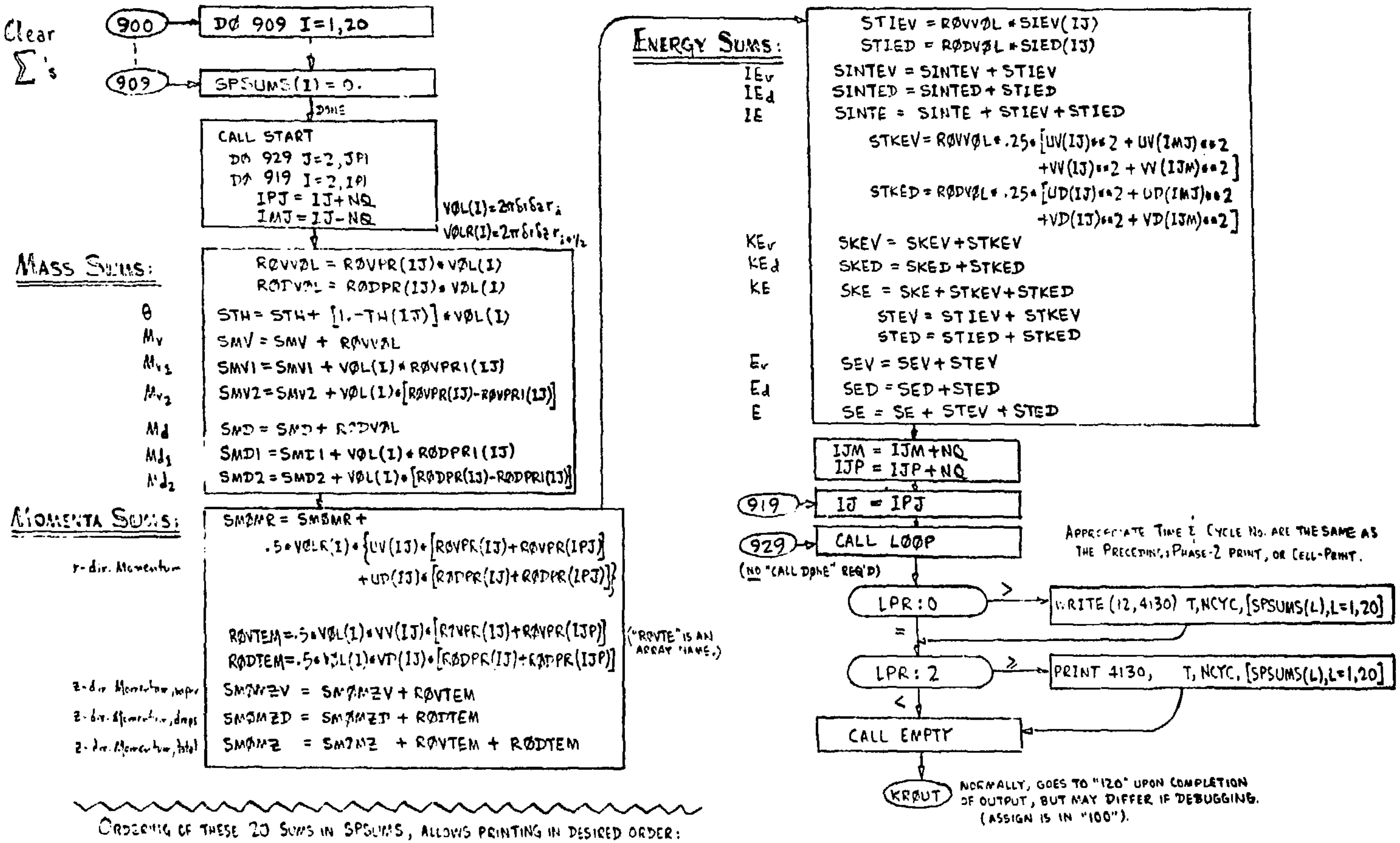
LONG PRINT:

I J θ UV VV SIEV RØVPRI RØVPR D P
E UD VD SIED RØDPRI RØDPR A K } 2-LINE FORMAT



SHORT PRINT:

⑪



```
900  DØ 909 I=1,20
909  SPSUMS(I) = 0.
     CALL START
      DØ 929 J=2,JP1
      DØ 919 I=2,IP1
         IPJ = IJ+NQ
         IMJ = IJ-NQ
```

$VØL(I)=2\pi\delta_1\delta_2 r_i$

$VØLR(I)=2\pi\delta_1\delta_2 r_{i+1/2}$

```
      RØVVØL = RØVPR(IJ)*VØL(I)
      RØDVØL = RØDPR(IJ)*VØL(I)
STH= STH+ [1.-TH(IJ)]*VØL(I)
SMV = SMV + RØVVØL
SMV1 = SMV1 + VØL(I)*RØVPR1(IJ)
SMV2 = SMV2 + VØL(I)*[RØVPR(IJ)-RØVPR1(IJ)]
SMD = SMD + RØDVØL
SMD1 = SMD1 + VØL(I)*RØDPR1(IJ)
SMD2 = SMD2 + VØL(I)*[RØDPR(IJ)-RØDPR1(IJ)]

SMØMR = SMØMR +
     .5*VØLR(I)*{UV(IJ)*[RØVPR(IJ)+RØVPR(IPJ)]
                +UD(IJ)*[RØDPR(IJ)+RØDPR(IPJ)]}

RØVTEM=.5*VØL(I)*VV(IJ)*[RØVPR(IJ)+RØVPR(IJP)]
RØDTEM=.5*VØL(I)*VD(IJ)*[RØDPR(IJ)+RØDPR(IJP)]
SMØMZV = SMØMZV + RØVTEM
SMØMZD = SMØMZD + RØDTEM
SMØMZ  = SMØMZ  + RØVTEM + RØDTEM
```

("RØVTE" IS AN ARRAY NAME.)

```
   STIEV = RØVVØL *SIEV(IJ)
   STIED = RØDVØL *SIED(IJ)
SINTEV = SINTEV + STIEV
SINTED = SINTED + STIED
SINTE  = SINTE + STIEV + STIED
   STKEV = RØVVØL*.25*[UV(IJ)**2 + UV(IMJ)**2
                      +VV(IJ)**2 + VV(IJM)**2]
   STKED = RØDVØL*.25*[UD(IJ)**2 + UD(IMJ)**2
                      +VD(IJ)**2 + VD(IJM)**2]
SKEV = SKEV+STKEV
SKED = SKED+STKED
SKE  = SKE+STKEV+STKED
   STEV = STIEV + STKEV
   STED = STIED + STKED
SEV = SEV+STEV
SED = SED+STED
SE  = SE + STEV + STED

     IJM = IJM+NQ
     IJP = IJP+NQ
919  IJ = IPJ
929  CALL LØØP
     LPR:0   >  WRITE (12,4130) T,NCYC,[SPSUMS(L),L=1,20]
     LPR:2   >  PRINT 4130,    T,NCYC,[SPSUMS(L),L=1,20]
     CALL EMPTY
     KRØUT
```

(NO "CALL DONE" REQ'D)

APPROPRIATE TIME & CYCLE NO. ARE THE SAME AS THE PRECEDING PHASE-2 PRINT, OR CELL-PRINT.

NORMALLY, GOES TO "120" UPON COMPLETION OF OUTPUT, BUT MAY DIFFER IF DEBUGGING. (ASSIGN IS IN "100").

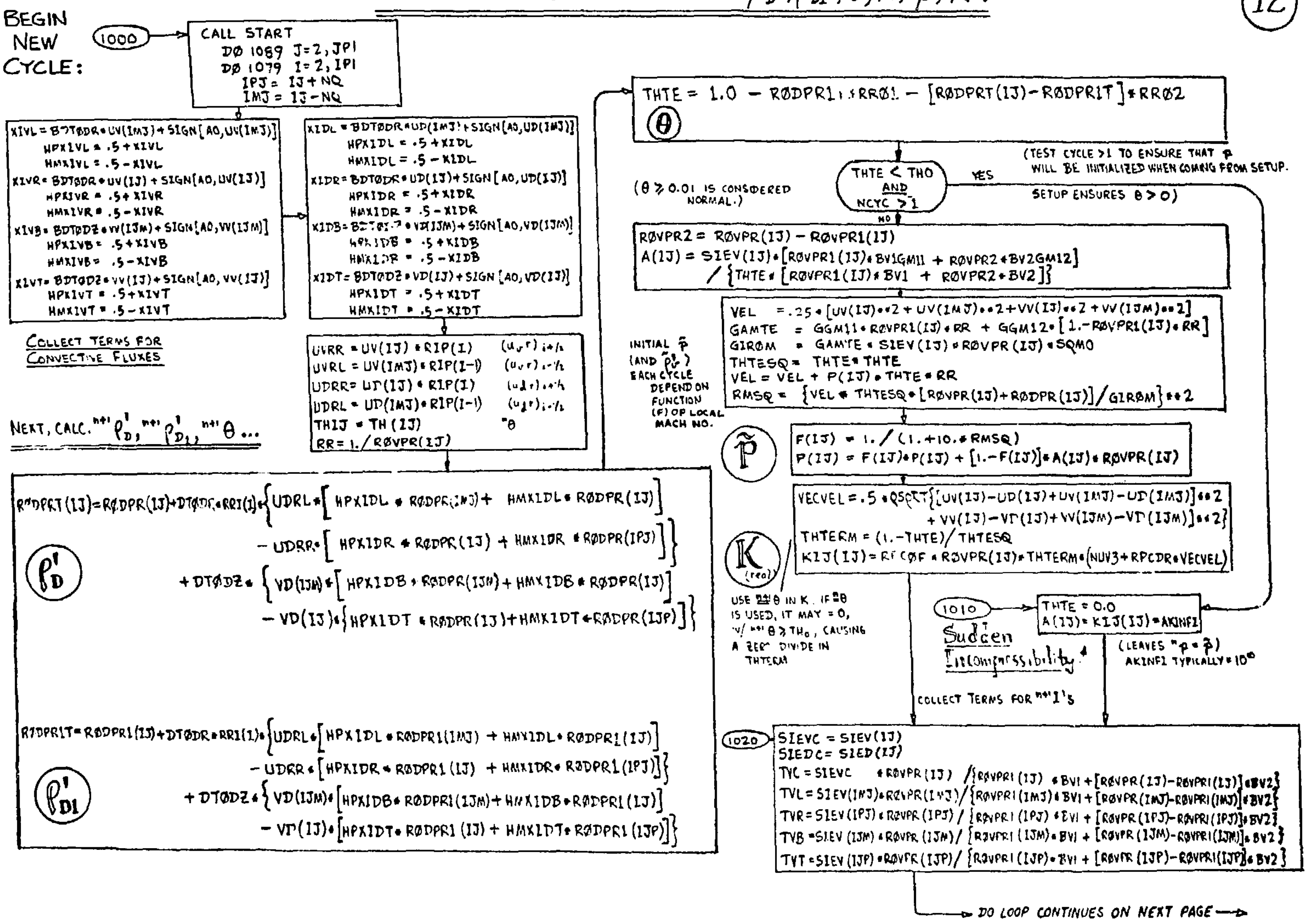
ORDERING OF THESE 20 SUMS IN SPSUMS, ALLOWS PRINTING IN DESIRED ORDER:

WD. 1 = STH	WD. 6 = SMV1	WD. 11 = SMV2	WD. 16 = SMV
2 = SMØMR	7 = SMD1	12 = SMD2	17 = SMD
3 = SMØMZV	8 = SINTEV	13 = SKEV	18 = SEV
4 = SMØMZD	9 = SINTED	14 = SKED	19 = SED
5 = SMØMZ	10 = SINTE	15 = SKE	20 = SE
LINE 1	LINE 2	LINE 3	LINE 4

CHOICE —
DO SHORT PRINT:
— EVERY CYCLE (ISPR=2)
— LONG-PRINT CYCLES ONLY (ISPR=1)

PHASE 1 – P.1 – CALCULATE ρ_D', ρ_{D1}', θ, A, $\tilde{p}$, K:

⑫



```
CALL START
  DØ 1089 J=2,JP1
  DØ 1079 I=2,IP1
    IPJ= IJ+NQ
    IMJ= IJ-NQ
```

```
XIVL=BDTØDR*UV(IMJ)+SIGN[AO,UV(IMJ)]
    HPXIVL = .5+XIVL
    HMXIVL = .5-XIVL
XIVR= BDTØDR*UV(IJ) + SIGN[AO,UV(IJ)]
    HPXIVR = .5+ XIVR
    HMXIVR = .5-XIVR
XIVB= BDTØDZ*VV(IJM)+SIGN[AO,VV(IJM)]
    HPXIVB= .5+XIVB
    HMXIVB= .5-XIVB
XIVT= BDTØDZ*VV(IJ)+SIGN[AO, VV(IJ)]
    HPXIVT = .5+XIVT
    HMXIVT = .5-XIVT
```

COLLECT TERMS FOR CONVECTIVE FLUXES

```
XIDL = BDTØDR*UD(IMJ)+SIGN[AO,UD(IMJ)]
    HPXIDL = .5+XIDL
    HMXIDL = .5-XIDL
XIDR= BDTØDR*UD(IJ)+SIGN[AO,UD(IJ)]
    HPXIDR = .5+XIDR
    HMXIDR = .5-XIDR
XIDB= BDTØDZ*VD(IJM)+SIGN[AO,VD(IJM)]
    HPXIDB = .5+XIDB
    HMXIDB = .5-XIDB
XIDT= BDTØDZ*VD(IJ)+SIGN[AO,VD(IJ)]
    HPXIDT = .5+XIDT
    HMXIDT = .5-XIDT
```

```
UVRR = UV(IJ) *RIP(I)        (uvr)i+½
UVRL = UV(IMJ)*RIP(I-1)      (uvr)i-½
UDRR= UD(IJ) * RIP(I)        (udr)i+½
UDRL = UD(IMJ)*RIP(I-1)      (udr)i-½
THIJ = TH (IJ)               ⁿθ
RR= 1./RØVPR(IJ)
```

NEXT, CALC. $^{n+1}\rho_D'$, $^{n+1}\rho_{D1}'$, $^{n+1}\theta$ ...

```
RØDPRT(IJ)=RØDPR(IJ)+DTØDR*RRI(I)*{UDRL*[ HPXIDL * RØDPR(IMJ)+  HMXIDL* RØDPR(IJ)]
                                  - UDRR*[ HPXIDR * RØDPR(IJ) + HMXIDR * RØDPR(IPJ)]}
                      + DTØDZ* {VD(IJM)*[ HPXIDB * RØDPR(IJM)+ HMXIDB * RØDPR(IJ)]
                                  - VD(IJ)*{HPXIDT * RØDPR(IJ)+HMXIDT*RØDPR(IJP)]}
```

```
RØDPR1T= RØDPR1(IJ)+DTØDR*RRI(I)*{UDRL*[HPXIDL * RØDPR1(IMJ) + HMXIDL* RØDPR1(IJ)]
                                 - UDRR*[HPXIDR * RØDPR1 (IJ)  + HMXIDR* RØDPR1(IPJ)]}
                      + DTØDZ*{VD(IJM)*[HPXIDB* RØDPR1(IJM)+ HMXIDB*RØDPR1(IJ)]
                                 - VD(IJ)*[HPXIDT* RØDPR1 (IJ) + HMXIDT* RØDPR1 (IJP)]}
```

```
THTE = 1.0 - RØDPR1(IJ)*RRØ1 - [RØDPRT(IJ)-RØDPR1T]*RRØ2
```

(θ ≥ 0.01 IS CONSIDERED NORMAL.)

```
THTE < THO
   AND
NCYC > 1
```

(TEST CYCLE >1 TO ENSURE THAT p WILL BE INITIALIZED WHEN COMING FROM SETUP. SETUP ENSURES θ > 0)

```
RØVPR2 = RØVPR(IJ) - RØVPR1(IJ)
A(IJ) = SIEV(IJ)*[RØVPR1(IJ)*BV1GM11 + RØVPR2*BV2GM12]
        / {THTE*[RØVPR1(IJ)*BV1 + RØVPR2*BV2]}
```

```
VEL    =.25*[UV(IJ)**2 + UV(IMJ)**2+VV(IJ)**2 + VV(IJM)**2]
GAMTE  = GGM11*RØVPR1(IJ)*RR + GGM12*[1.-RØVPR1(IJ)*RR]
GIRØM  = GAMTE * SIEV(IJ) * RØVPR (IJ) * SQMO
THTESQ= THTE*THTE
VEL = VEL + P(IJ)*THTE*RR
RMSQ = {VEL * THTESQ*[RØVPR(IJ)+RØDPR(IJ)]/GIRØM}**2
```

INITIAL $\tilde{p}$ (AND $\tilde{\rho}_V'$) EACH CYCLE DEPEND ON FUNCTION (F) OF LOCAL MACH NO.

```
F(IJ) = 1./(1.+10.*RMSQ)
P(IJ) = F(IJ)*P(IJ) + [1.-F(IJ)]*A(IJ)*RØVPR(IJ)
```

```
VECVEL=.5*QSQRT{[UV(IJ)-UD(IJ)+UV(IMJ)-UD(IMJ)]**2
                + VV(IJ)-VD(IJ)+VV(IJM)-VD(IJM)]**2}
THTERM = (1.-THTE)/THTESQ
KIJ(IJ)=RFCØF * RØVPR(IJ)*THTERM*(NUV3+RPCDR*VECVEL)
```

USE $^{n+1}\theta$ IN K. IF $^{n}\theta$ IS USED, IT MAY = 0, W/ $^{n+1}\theta$ ≥ THₒ, CAUSING A ZERO DIVIDE IN THTERM

```
THTE = 0.0
A(IJ)= KIJ(IJ)*AKINF1
```

Sudden Incompressibility!

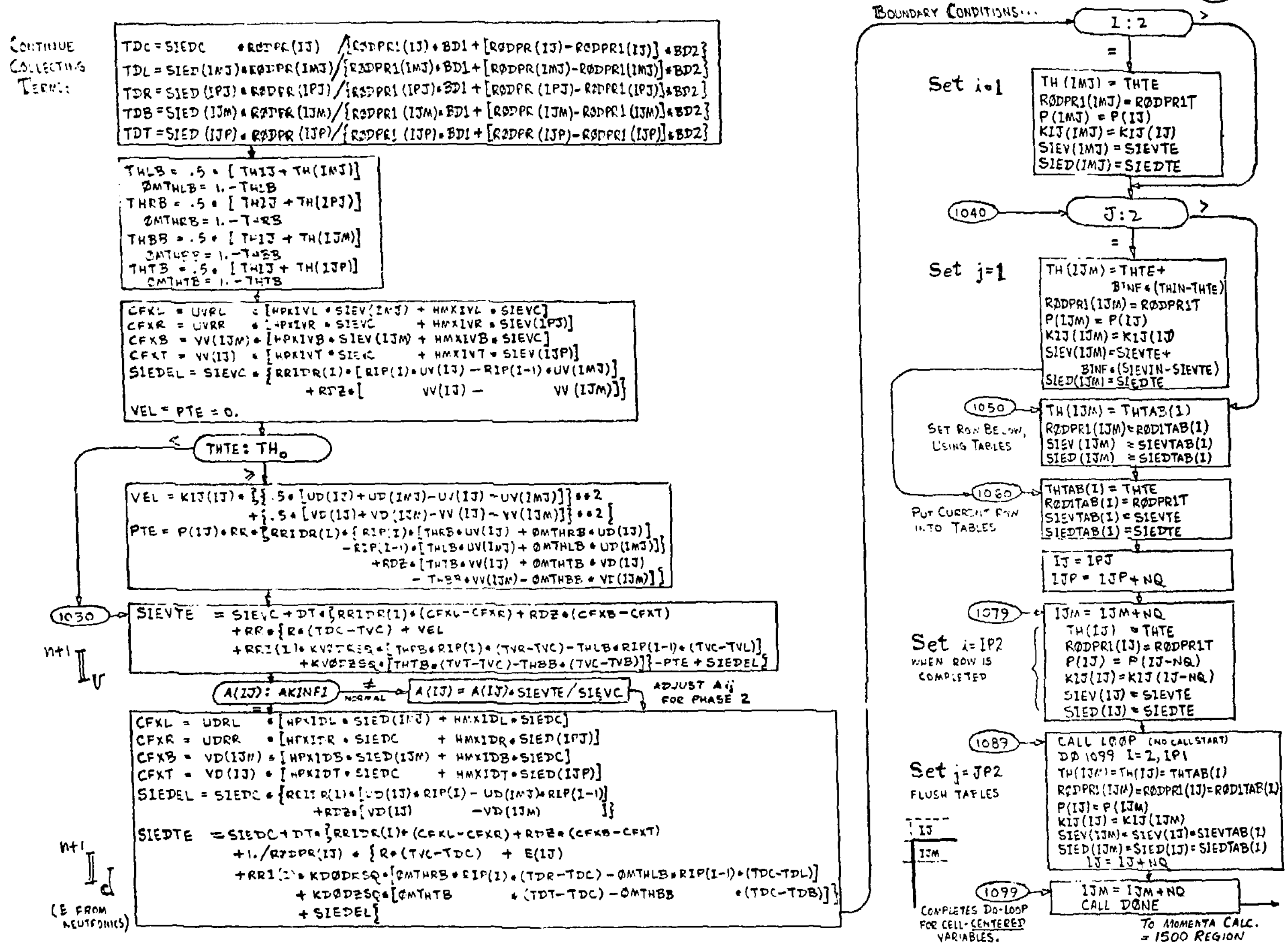
(LEAVES $^{n}p = \tilde{p}$) AKINF1 TYPICALLY = 10⁶

COLLECT TERMS FOR ^{n+1}I's

```
SIEVC = SIEV(IJ)
SIEDC= SIED(IJ)
TVC = SIEVC     *RØVPR(IJ)  /{RØVPR1(IJ)  *BV1 + [RØVPR(IJ)-RØVPR1(IJ)]*BV2}
TVL= SIEV(IMJ)*RØVPR(IMJ)/{RØVPR1(IMJ)*BV1 + [RØVPR(IMJ)-RØVPR1(IMJ)]*BV2}
TVR= SIEV(IPJ)*RØVPR(IPJ)/ {RØVPR1(IPJ) *BV1 + [RØVPR(IPJ)-RØVPR1(IPJ)]*BV2}
TVB =SIEV (IJM)*RØVPR(IJM)/ {RØVPR1(IJM)*BV1 + [RØVPR(IJM)-RØVPR1(IJM)]*BV2}
TVT =SIEV (IJP)*RØVPR(IJP)/ {RØVPR1(IJP)*BV1 + [RØVPR(IJP)-RØVPR1(IJP)]*BV2}
```

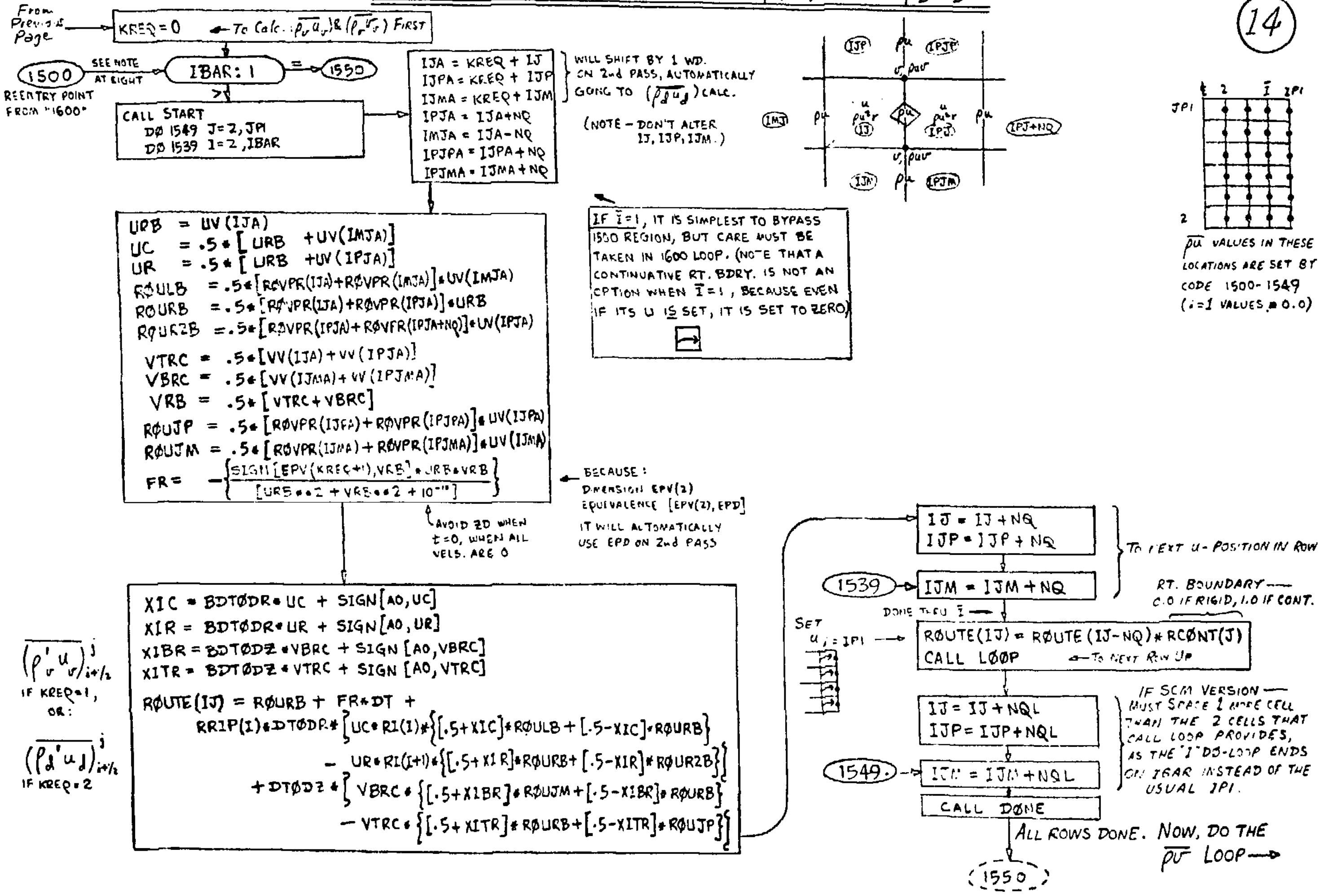
DO LOOP CONTINUES ON NEXT PAGE →

PHASE 1 – P.2 – CALCULATE $^{n+1}I_V$ AND $^{n+1}I_D$:



```
TDC = SIEDC     * RØDPR(IJ)  /{RØDPR1(IJ)*BD1 + [RØDPR(IJ)-RØDPR1(IJ)] *BD2}
TDL = SIED(IMJ) * RØDPR(IMJ) /{RØDPR1(IMJ)*BD1 + [RØDPR(IMJ)-RØDPR1(IMJ)]*BD2}
TDR = SIED(IPJ) * RØDPR(IPJ) /{RØDPR1(IPJ)*BD1 + [RØDPR(IPJ)-RØDPR1(IPJ)]*BD2}
TDB = SIED(IJM) * RØDPR(IJM) /{RØDPR1(IJM)*BD1 + [RØDPR(IJM)-RØDPR1(IJM)]*BD2}
TDT = SIED(IJP) * RØDPR(IJP) /{RØDPR1(IJP)*BD1 + [RØDPR(IJP)-RØDPR1(IJP)]*BD2}

THLB = .5 * [THIJ + TH(IMJ)]
  ØMTHLB = 1.-THLB
THRB = .5 * [THIJ + TH(IPJ)]
  ØMTHRB = 1.-THRB
THBB = .5 * [THIJ + TH(IJM)]
  ØMTHBB = 1.-THBB
THTB = .5 * [THIJ + TH(IJP)]
  ØMTHTB = 1.-THTB

CFXL = UVRL    * [HPXIVL * SIEV(IMJ) + HMXIVL * SIEVC]
CFXR = UVRR    * [HPXIVR * SIEVC     + HMXIVR * SIEV(IPJ)]
CFXB = VV(IJM) * [HPXIVB * SIEV(IJM) + HMXIVB * SIEVC]
CFXT = VV(IJ)  * [HPXIVT * SIEVC     + HMXIVT * SIEV(IJP)]
SIEDEL = SIEVC * {RRIDR(I)*[RIP(I)*UV(IJ) - RIP(I-1)*UV(IMJ)]
                  +RDZ*[      VV(IJ) -         VV(IJM)]}
VEL = PTE = 0.

VEL = KIJ(IJ) * {{.5*[UD(IJ)+UD(IMJ)-UV(IJ) - UV(IMJ)]}**2
                +{.5*[VD(IJ)+VD(IJM)-VV(IJ) - VV(IJM)]}**2}
PTE = P(IJ)*RR*{RRIDR(I)*{RIP(I)*[THRB*UV(IJ) + ØMTHRB*UD(IJ)]
                          -RIP(I-1)*[THLB*UV(IMJ) + ØMTHLB*UD(IMJ)]}
                +RDZ*[THTB*VV(IJ) + ØMTHTB*VD(IJ)
                      - THBB*VV(IJM) - ØMTHBB*VD(IJM)]}

1030 SIEVTE = SIEVC + DT*{RRIDR(I)*(CFXL-CFXR) + RDZ*(CFXB-CFXT)
            +RR*{R*(TDC-TVC) + VEL
            +RRI(I)*KVØDRSQ*[THRB*RIP(I)*(TVR-TVC)-THLB*RIP(I-1)*(TVC-TVL)]
              +KVØDZSQ*[THTB*(TVT-TVC)-THBB*(TVC-TVB)]}-PTE + SIEDEL}

A(IJ) = A(IJ)*SIEVTE/SIEVC

CFXL = UDRL    * [HPXIDL * SIED(IMJ) + HMXIDL * SIEDC]
CFXR = UDRR    * [HPXIDR * SIEDC     + HMXIDR * SIED(IPJ)]
CFXB = VD(IJM) * [HPXIDB * SIED(IJM) + HMXIDB * SIEDC]
CFXT = VD(IJ)  * [HPXIDT * SIEDC     + HMXIDT * SIED(IJP)]
SIEDEL = SIEDC * {RRIDR(I)*[UD(IJ)*RIP(I) - UD(IMJ)*RIP(I-1)]
                  +RDZ*[VD(IJ)       -VD(IJM)       ]}

SIEDTE = SIEDC + DT*{RRIDR(I)*(CFXL-CFXR) + RDZ*(CFXB-CFXT)
        +1./RØDPR(IJ) * {R*(TVC-TDC)  + E(IJ)
        +RRI(I)*KDØDRSQ*[ØMTHRB*RIP(I)*(TDR-TDC) - ØMTHLB*RIP(I-1)*(TDC-TDL)]
           + KDØDZSQ*[ØMTHTB     *(TDT-TDC) - ØMTHBB      *(TDC-TDB)]}
           + SIEDEL}

TH(IMJ) = THTE
RØDPR1(IMJ) = RØDPR1T
P(IMJ) = P(IJ)
KIJ(IMJ) = KIJ(IJ)
SIEV(IMJ) = SIEVTE
SIED(IMJ) = SIEDTE

TH(IJM) = THTE+
          BINF*(THIN-THTE)
RØDPR1(IJM) = RØDPR1T
P(IJM) = P(IJ)
KIJ(IJM) = KIJ(IJ)
SIEV(IJM) = SIEVTE+
          BINF*(SIEVIN-SIEVTE)
SIED(IJM) = SIEDTE

1050 TH(IJM) = THTAB(I)
RØDPR1(IJM) = RØD1TAB(I)
SIEV(IJM) = SIEVTAB(I)
SIED(IJM) = SIEDTAB(I)

1060 THTAB(I) = THTE
RØD1TAB(I) = RØDPR1T
SIEVTAB(I) = SIEVTE
SIEDTAB(I) = SIEDTE

IJ = IPJ
IJP = IJP+NQ

1079 IJM = IJM+NQ
  TH(IJ) = THTE
  RØDPR1(IJ) = RØDPR1T
  P(IJ) = P(IJ-NQ)
  KIJ(IJ) = KIJ(IJ-NQ)
  SIEV(IJ) = SIEVTE
  SIED(IJ) = SIEDTE

1089 CALL LOØP (NO CALL START)
DØ 1099 I=2,IP1
TH(IJM)=TH(IJ)=THTAB(I)
RØDPR1(IJM)=RØDPR1(IJ)=RØD1TAB(I)
P(IJ)=P(IJM)
KIJ(IJ) = KIJ(IJM)
SIEV(IJM)=SIEV(IJ)=SIEVTAB(I)
SIED(IJM)=SIED(IJ)=SIEDTAB(I)
  IJ = IJ+NQ

1099 IJM = IJM+NQ
CALL DØNE
```

PHASE 1 – P. 3 – CALCULATE $(\overline{\rho_V u_V})$ OR $(\overline{\rho_D u_D})$:



From Previous Page

```
KREQ = 0          <- To Calc. (ρ_v u_v) & (ρ_r v_r) FIRST
```

(1500) SEE NOTE AT RIGHT → IBAR : 1 — = → (1550); > ↓

REENTRY POINT FROM "1600"

```
CALL START
   DØ 1549  J=2,JP1
   DØ 1539  I=2,IBAR
```

```
IJA = KREQ + IJ
IJPA = KREQ + IJP
IJMA = KREQ + IJM
IPJA = IJA+NQ
IMJA = IJA-NQ
IPJPA = IJPA+NQ
IPJMA = IJMA+NQ
```

WILL SHIFT BY 1 WD. ON 2nd PASS, AUTOMATICALLY GOING TO $(\overline{\rho_d u_d})$ CALC.

(NOTE – DON'T ALTER IJ, IJP, IJM.)

IF $\bar{I}$=1, IT IS SIMPLEST TO BYPASS 1500 REGION, BUT CARE MUST BE TAKEN IN 1600 LOOP. (NOTE THAT A CONTINUATIVE RT. BDRY. IS NOT AN OPTION WHEN $\bar{I}$=1, BECAUSE EVEN IF ITS U IS SET, IT IS SET TO ZERO)

$\overline{\rho u}$ VALUES IN THESE LOCATIONS ARE SET BY CODE 1500-1549 (i=1 VALUES = 0.0)

```
URB    = UV(IJA)
UC     = .5*[URB  +UV(IMJA)]
UR     = .5*[URB  +UV(IPJA)]
RØULB  = .5*[RØVPR(IJA)+RØVPR(IMJA)]*UV(IMJA)
RØURB  = .5*[RØVPR(IJA)+RØVPR(IPJA)]*URB
RØUR2B = .5*[RØVPR(IPJA)+RØVPR(IPJA+NQ)]*UV(IPJA)
VTRC  = .5*[VV(IJA)+VV(IPJA)]
VBRC  = .5*[VV(IJMA)+VV(IPJMA)]
VRB   = .5*[VTRC+VBRC]
RØUJP = .5*[RØVPR(IJPA)+RØVPR(IPJPA)]*UV(IJPA)
RØUJM = .5*[RØVPR(IJMA)+RØVPR(IPJMA)]*UV(IJMA)
```

$$FR = -\left\{\frac{SIGN[EPV(KREQ+1),VRB]*URB*VRB}{[URB**2+VRB**2+10^{-10}]}\right\}$$

AVOID ZD WHEN t=0, WHEN ALL VELS. ARE 0

BECAUSE:
DIMENSION EPV(2)
EQUIVALENCE [EPV(2), EPD]
IT WILL AUTOMATICALLY USE EPD ON 2nd PASS

$(\overline{\rho'_v u_v})^j_{i+1/2}$ IF KREQ=1, OR:

$(\overline{\rho'_d u_d})^j_{i+1/2}$ IF KREQ=2

```
XIC  = BDTØDR*UC + SIGN[A0,UC]
XIR  = BDTØDR*UR + SIGN[A0,UR]
XIBR = BDTØDZ*VBRC + SIGN[A0,VBRC]
XITR = BDTØDZ*VTRC + SIGN[A0,VTRC]

RØUTE(IJ) = RØURB + FR*DT +
     RRIP(I)*DTØDR*{UC*RI(I)*{[.5+XIC]*RØULB+[.5-XIC]*RØURB}
                   - UR*RI(I+1)*{[.5+XIR]*RØURB+[.5-XIR]*RØUR2B}}
     +DTØDZ*{VBRC*{[.5+XIBR]*RØUJM+[.5-XIBR]*RØURB}
             - VTRC*{[.5+XITR]*RØURB+[.5-XITR]*RØUJP}}
```

```
IJ = IJ+NQ
IJP = IJP+NQ
```

```
(1539) IJM = IJM+NQ
```

To NEXT u-POSITION IN ROW

DONE THRU $\bar{I}$ →

SET $u_{j=IP1}$ →

RT. BOUNDARY — 0.0 IF RIGID, 1.0 IF CONT.

```
RØUTE(IJ) = RØUTE(IJ-NQ)*RCØNT(J)
CALL LØØP          <- To NEXT Row UP
```

```
IJ = IJ+NQL
IJP = IJP+NQL
```

```
(1549) IJM = IJM+NQL
```

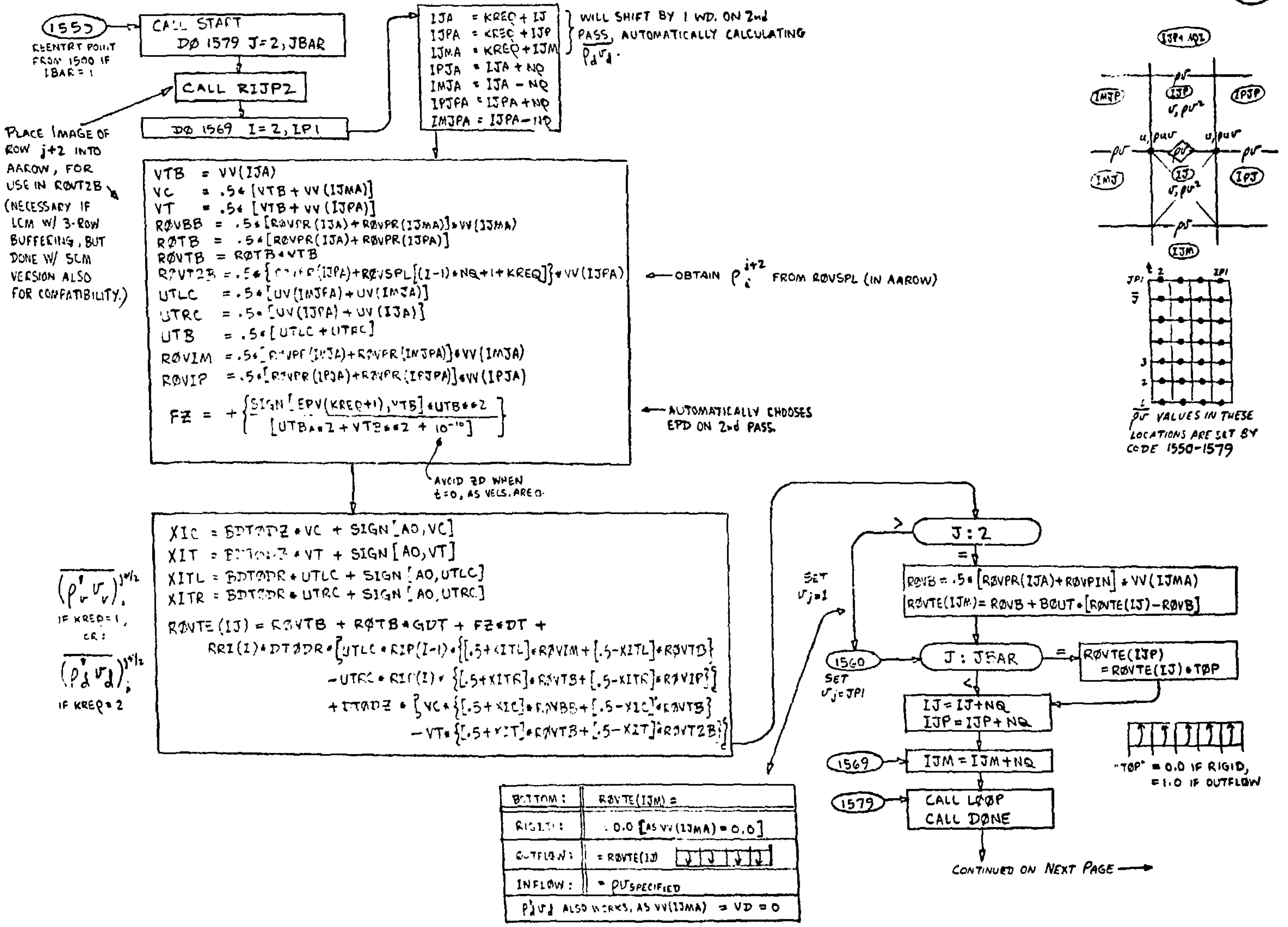
IF SCM VERSION — MUST SPACE 1 MORE CELL THAN THE 2 CELLS THAT CALL LOOP PROVIDES, AS THE "I" DO-LOOP ENDS ON IBAR INSTEAD OF THE USUAL IP1.

```
CALL DØNE
```

ALL ROWS DONE. NOW, DO THE $\overline{\rho v}$ LOOP →

(1550)

PHASE 1 – P. 4 – CALCULATE $(\rho_v V_v)$ OR $(\rho_D V_D)$:



```
CALL START
  DØ 1579 J=2,JBAR

CALL RIJP2

DØ 1569 I=2,IP1
```

```
IJA   = KREQ+IJ
IJPA  = KREQ+IJP
IJMA  = KREQ+IJM
IPJA  = IJA+NQ
IMJA  = IJA-NQ
IPJPA = IJPA+NQ
IMJPA = IJPA-NQ
```

```
VTB    = VV(IJA)
VC     = .5*[VTB+VV(IJMA)]
VT     = .5*[VTB+VV(IJPA)]
RØVBB  = .5*[RØVPR(IJA)+RØVPR(IJMA)]*VV(IJMA)
RØTB   = .5*[RØVPR(IJA)+RØVPR(IJPA)]
RØVTB  = RØTB*VTB
RØVT2B = .5*{RØVPR(IJPA)+RØVSPL[(I-1)*NQ+1+KREQ]}*VV(IJPA)
UTLC   = .5*[UV(IMJPA)+UV(IMJA)]
UTRC   = .5*[UV(IJPA)+UV(IJA)]
UTB    = .5*[UTLC+UTRC]
RØVIM  = .5*[RØVPR(IMJA)+RØVPR(IMJPA)]*VV(IMJA)
RØVIP  = .5*[RØVPR(IPJA)+RØVPR(IPJPA)]*VV(IPJA)
```

$$FZ = +\left\{\frac{SIGN\left[EPV(KREQ+1),VTB\right]*UTB**2}{\left[UTB**2+VTB**2+10^{-10}\right]}\right\}$$

```
XIC  = BDTØDZ*VC + SIGN[AO,VC]
XIT  = BDTØDZ*VT + SIGN[AO,VT]
XITL = BDTØDR*UTLC + SIGN[AO,UTLC]
XITR = BDTØDR*UTRC + SIGN[AO,UTRC]

RØVTE(IJ) = RØVTB + RØTB*GDT + FZ*DT +
     RRI(I)*DTØDR*{UTLC*RIP(I-1)*{[.5+XITL]*RØVIM+[.5-XITL]*RØVTB}
                  -UTRC*RIP(I)*{[.5+XITR]*RØVTB+[.5-XITR]*RØVIP}}
                  +DTØDZ*{VC*{[.5+XIC]*RØVBB+[.5-XIC]*RØVTB}
                       -VT*{[.5+XIT]*RØVTB+[.5-XIT]*RØVT2B}}
```

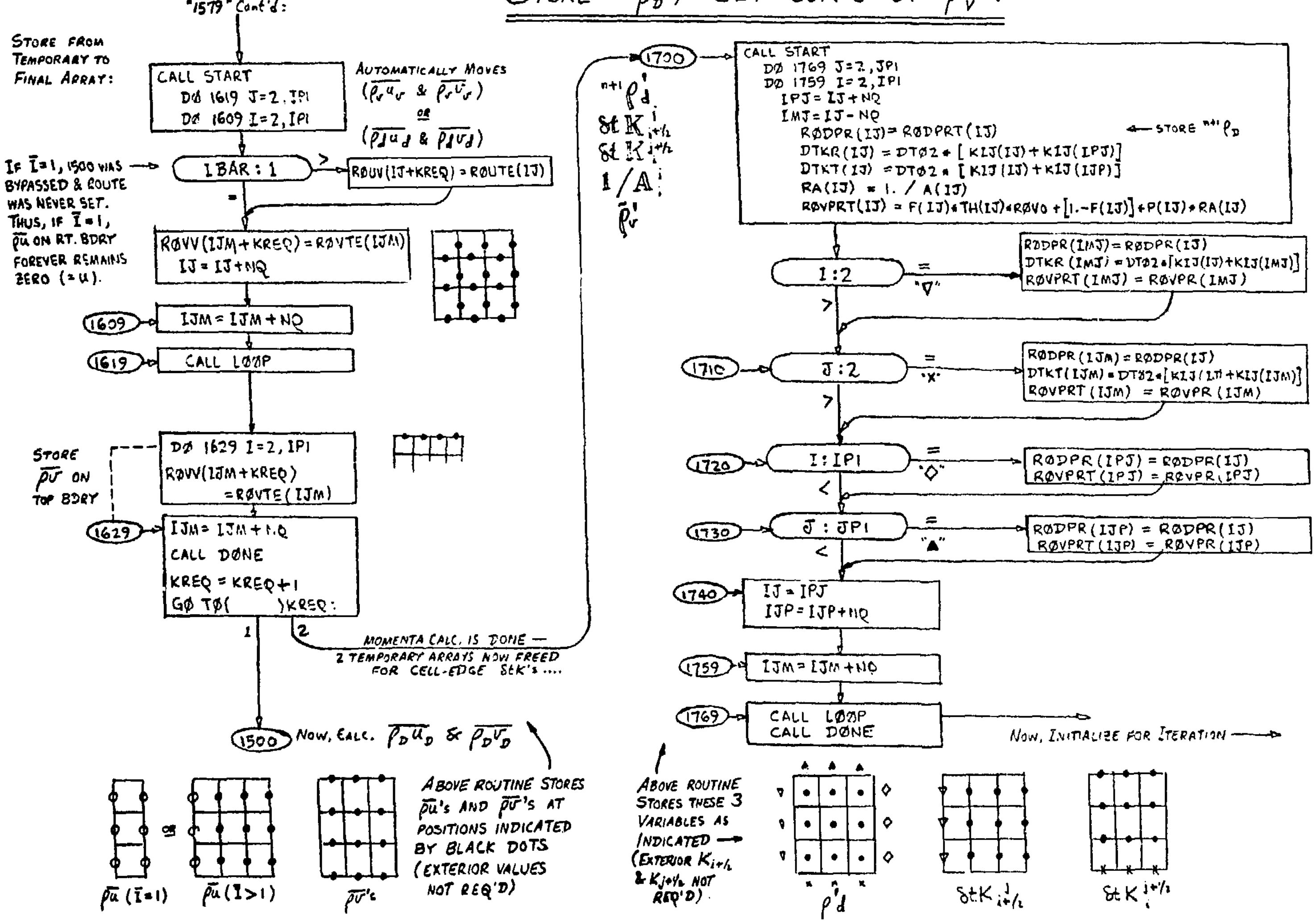
BOTTOM:	RØVTE(IJM) =
RIGID:	= 0.0 [AS VV(IJMA) = 0.0]
OUTFLOW:	= RØVTE(IJ)
INFLOW:	= $\rho v_{SPECIFIED}$
$\overline{\rho_d' v_d}$ ALSO WORKS, AS VV(IJMA) = VD = 0	

```
RØVB = .5*[RØVPR(IJA)+RØVPIN]*VV(IJMA)
RØVTE(IJM) = RØVB + BØUT*[RØVTE(IJ)-RØVB]

RØVTE(IJP)
   = RØVTE(IJ)*TØP

IJ = IJ+NQ
IJP = IJP+NQ

IJM = IJM+NQ

CALL LØØP
CALL DØNE
```

CONTINUED ON NEXT PAGE ⟶

PHASE 1 - P.5 - COMPLETION OF MOMENTUM CALCULATION,

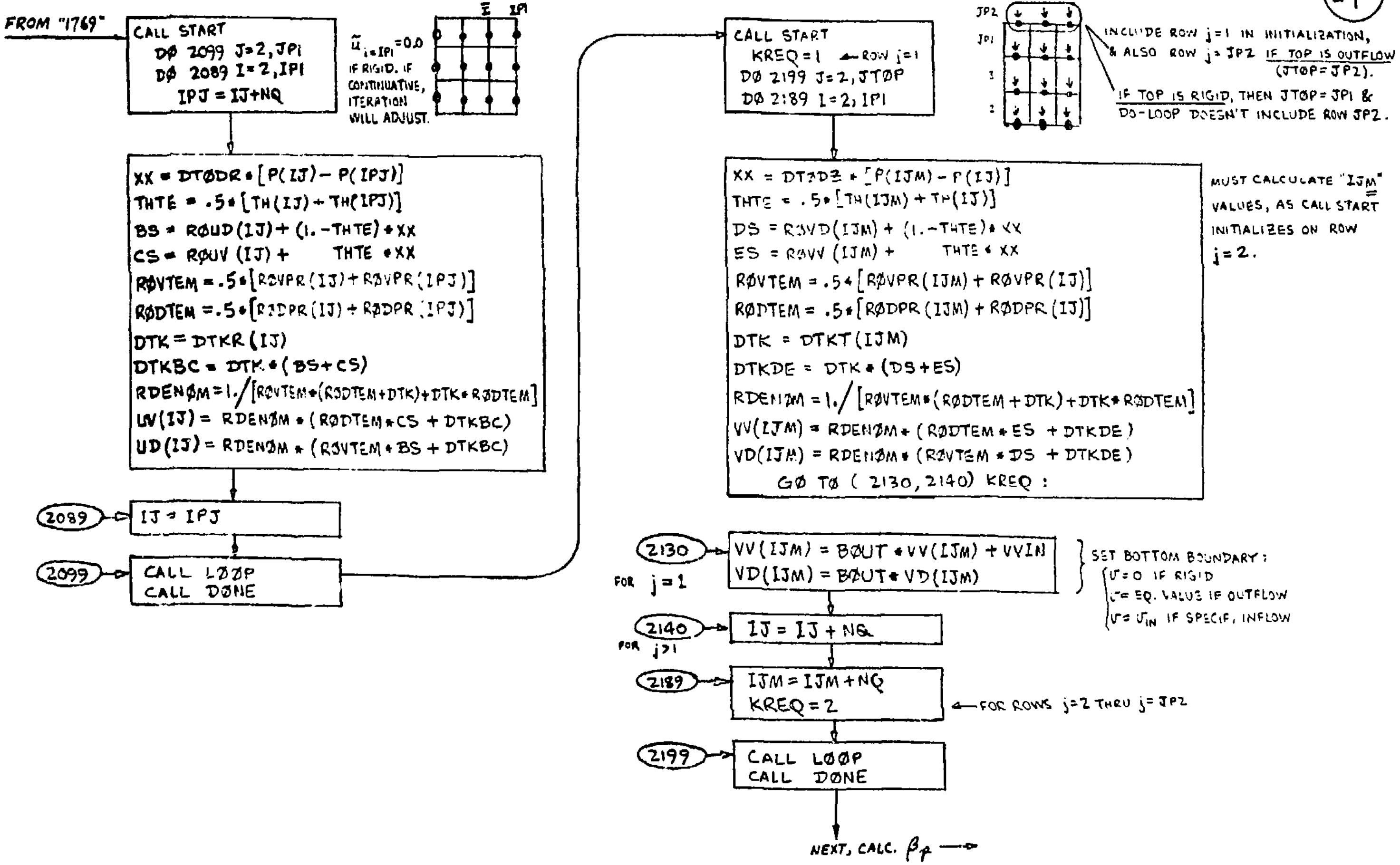
STORE $^{n+1}\rho_D'$, SET $\delta t K$'s & $\tilde{\rho}_V'$:



```
"1579" Cont'd:
      CALL START
        DØ 1619 J=2,JPI
        DØ 1609 I=2,IPI
      IBAR : 1
        >  RØUV(IJ+KREQ) = RØUTE(IJ)
        =
      RØVV(IJM+KREQ) = RØVTE(IJM)
        IJ = IJ+NQ
1609  IJM = IJM+NQ
1619  CALL LØØP
      DØ 1629 I=2,IPI
      RØVV(IJM+KREQ)
           =RØVTE(IJM)
1629  IJM = IJM+NQ
      CALL DØNE
      KREQ = KREQ+1
      GØ TØ(        )KREQ:
        1 -> 1500
        2 -> 1700

1700  CALL START
        DØ 1769 J=2,JPI
        DØ 1759 I=2,IPI
          IPJ = IJ+NQ
          IMJ = IJ-NQ
            RØDPR(IJ) = RØDPRT(IJ)
            DTKR(IJ) = DTØ2 * [KIJ(IJ)+KIJ(IPJ)]
            DTKT(IJ) = DTØ2 * [KIJ(IJ)+KIJ(IJP)]
            RA(IJ) = 1. / A(IJ)
            RØVPRT(IJ) = F(IJ)*TH(IJ)*RØV0 + [1.-F(IJ)]*P(IJ)*RA(IJ)
      I:2
        =  "∇"  RØDPR(IMJ) = RØDPR(IJ)
                DTKR(IMJ) = DTØ2*[KIJ(IJ)+KIJ(IMJ)]
                RØVPRT(IMJ) = RØVPR(IMJ)
        >
1710  J:2
        =  "x"  RØDPR(IJM) = RØDPR(IJ)
                DTKT(IJM) = DTØ2*[KIJ(IJ)+KIJ(IJM)]
                RØVPRT(IJM) = RØVPR(IJM)
        >
1720  I:IPI
        =  "◇"  RØDPR(IPJ) = RØDPR(IJ)
                RØVPRT(IPJ) = RØVPR(IPJ)
        <
1730  J:JPI
        =  "▲"  RØDPR(IJP) = RØDPR(IJ)
                RØVPRT(IJP) = RØVPR(IJP)
        <
1740  IJ = IPJ
      IJP = IJP+NQ
1759  IJM = IJM+NQ
1769  CALL LØØP
      CALL DØNE
```

PHASE 2 – P.1 – INITIALIZE TILDE VELOCITIES:

17



FROM "1769"

```
CALL START
  DØ 2099 J=2,JP1
  DØ 2089 I=2,IP1
    IPJ = IJ+NQ
```

$\tilde{u}_{i=IP1}=0.0$ IF RIGID. IF CONTINUATIVE, ITERATION WILL ADJUST.

```
XX = DTØDR*[P(IJ)-P(IPJ)]
THTE = .5*[TH(IJ)+TH(IPJ)]
BS = RØUD(IJ)+(1.-THTE)*XX
CS = RØUV (IJ) +     THTE *XX
RØVTEM =.5*[RØVPR(IJ)+RØVPR(IPJ)]
RØDTEM =.5*[RØDPR(IJ)+RØDPR(IPJ)]
DTK = DTKR(IJ)
DTKBC = DTK*(BS+CS)
RDENØM=1./[RØVTEM*(RØDTEM+DTK)+DTK*RØDTEM]
UV(IJ) = RDENØM*(RØDTEM*CS + DTKBC)
UD(IJ) = RDENØM*(RØVTEM*BS + DTKBC)
```

```
2089  IJ = IPJ
2099  CALL LØØP
      CALL DØNE
```

```
CALL START
  KREQ=1    ← ROW j=1
  DØ 2199 J=2,JTØP
  DØ 2189 I=2,IP1
```

INCLUDE ROW j=1 IN INITIALIZATION, & ALSO ROW j=JP2 IF TOP IS OUTFLOW (JTØP=JP2).

IF TOP IS RIGID, THEN JTØP=JP1 & DO-LOOP DOESN'T INCLUDE ROW JP2.

```
XX = DTØDZ*[P(IJM)-P(IJ)]
THTE = .5*[TH(IJM)+TH(IJ)]
DS = RØVD(IJM) + (1.-THTE)*XX
ES = RØVV (IJM) +      THTE * XX
RØVTEM = .5*[RØVPR(IJM) + RØVPR(IJ)]
RØDTEM = .5*[RØDPR(IJM) + RØDPR(IJ)]
DTK = DTKT(IJM)
DTKDE = DTK*(DS+ES)
RDENØM = 1./[RØVTEM*(RØDTEM+DTK)+DTK*RØDTEM]
VV(IJM) = RDENØM*(RØDTEM*ES + DTKDE)
VD(IJM) = RDENØM*(RØVTEM*DS + DTKDE)
    GØ TØ ( 2130,2140) KREQ :
```

MUST CALCULATE "IJM" VALUES, AS CALL START INITIALIZES ON ROW j=2.

```
2130  VV(IJM) = BØUT*VV(IJM) + VVIN
      VD(IJM) = BØUT*VD(IJM)
```

FOR j=1

SET BOTTOM BOUNDARY:
$\tilde{v}$=0 IF RIGID
$\tilde{v}$= EQ. VALUE IF OUTFLOW
$\tilde{v}=\tilde{v}_{IN}$ IF SPECIF. INFLOW

```
2140  IJ = IJ + NQ
```

FOR j>1

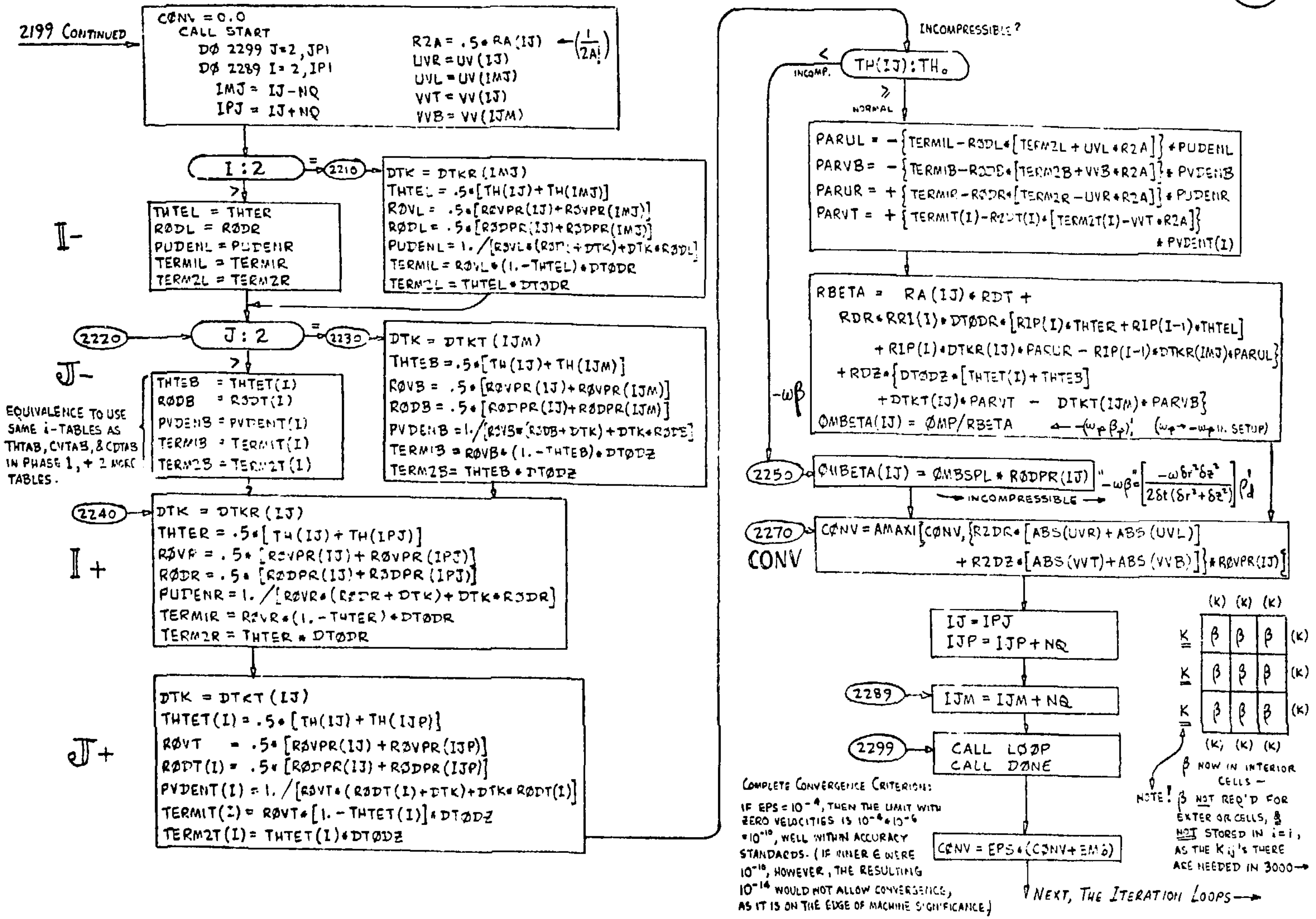
```
2189  IJM = IJM + NQ
      KREQ = 2
```

← FOR ROWS j=2 THRU j=JP2

```
2199  CALL LØØP
      CALL DØNE
```

NEXT, CALC. β_p →

PHASE 2 – P. 2 – CALCULATE β AND CONVERGENCE :



```
CØNV = 0.0
  CALL START
    DØ 2299 J=2,JP1
    DØ 2289 I= 2,IP1
      IMJ = IJ-NQ
      IPJ = IJ+NQ
R2A = .5*RA(IJ)   <-(1/2A_i)
UVR = UV(IJ)
UVL = UV(IMJ)
VVT = VV(IJ)
VVB = VV(IJM)

I:2  >
THTEL  = THTER
RØDL   = RØDR
PUDENL = PUDENR
TERM1L = TERM1R
TERM2L = TERM2R

I:2  = -> 2210
DTK = DTKR(IMJ)
THTEL = .5*[TH(IJ)+TH(IMJ)]
RØVL = .5*[RØVPR(IJ)+RØVPR(IMJ)]
RØDL = .5*[RØDPR(IJ)+RØDPR(IMJ)]
PUDENL = 1./[RØVL*(RØDL+DTK)+DTK*RØDL]
TERM1L = RØVL*(1.-THTEL)*DTØDR
TERM2L = THTEL*DTØDR

2220  J:2  >
THTEB  = THTET(I)
RØDB   = RØDT(I)
PVDENB = PVDENT(I)
TERM1B = TERM1T(I)
TERM2B = TERM2T(I)

J:2  = -> 2230
DTK = DTKT(IJM)
THTEB = .5*[TH(IJ)+TH(IJM)]
RØVB = .5*[RØVPR(IJ)+RØVPR(IJM)]
RØDB = .5*[RØDPR(IJ)+RØDPR(IJM)]
PVDENB = 1./[RØVB*(RØDB+DTK)+DTK*RØDB]
TERM1B = RØVB*(1.-THTEB)*DTØDZ
TERM2B = THTEB*DTØDZ

2240
DTK = DTKR(IJ)
THTER = .5*[TH(IJ)+TH(IPJ)]
RØVR = .5*[RØVPR(IJ)+RØVPR(IPJ)]
RØDR = .5*[RØDPR(IJ)+RØDPR(IPJ)]
PUDENR = 1./[RØVR*(RØDR+DTK)+DTK*RØDR]
TERM1R = RØVR*(1.-THTER)*DTØDR
TERM2R = THTER*DTØDR

DTK = DTKT(IJ)
THTET(I) = .5*[TH(IJ)+TH(IJP)]
RØVT     = .5*[RØVPR(IJ)+RØVPR(IJP)]
RØDT(I) = .5*[RØDPR(IJ)+RØDPR(IJP)]
PVDENT(I) = 1./[RØVT*(RØDT(I)+DTK)+DTK*RØDT(I)]
TERM1T(I) = RØVT*[1.-THTET(I)]*DTØDZ
TERM2T(I) = THTET(I)*DTØDZ

INCOMPRESSIBLE?
TH(IJ):TH0    < INCOMP. -> 2250    >= NORMAL
PARUL = -{TERM1L-RØDL*[TERM2L+UVL*R2A]}*PUDENL
PARVB = -{TERM1B-RØDB*[TERM2B+VVB*R2A]}*PVDENB
PARUR = +{TERM1R-RØDR*[TERM2R-UVR*R2A]}*PUDENR
PARVT = +{TERM1T(I)-RØDT(I)*[TERM2T(I)-VVT*R2A]}
                                     *PVDENT(I)

RBETA =  RA(IJ)*RDT +
   RDR*RR1(I)*DTØDR*[R1P(I)*THTER+R1P(I-1)*THTEL]
      +R1P(I)*DTKR(IJ)*PARUR - R1P(I-1)*DTKR(IMJ)*PARUL}
   +RDZ*{DTØDZ*[THTET(I)+THTEB]
      +DTKT(IJ)*PARVT  -  DTKT(IJM)*PARVB}
ØMBETA(IJ) = ØMP/RBETA

2250  ØMBETA(IJ) = ØMBSPL*RØDPR(IJ)

2270  CØNV = AMAX1[CØNV,{R2DR*[ABS(UVR)+ABS(UVL)]
                 +R2DZ*[ABS(VVT)+ABS(VVB)]}*RØVPR(IJ)]

IJ = IPJ
IJP = IJP+NQ

2289  IJM = IJM+NQ

2299  CALL LØØP
      CALL DØNE

CØNV = EPS*(CØNV+EM6)
```

$\leftarrow -(\omega_\phi \beta_\phi)'_i \quad (\omega_\phi \rightarrow -\omega_\phi \text{ in SETUP})$

$$-\omega\beta'' = \left[\frac{-\omega\delta r^2\delta z^2}{2\delta t(\delta r^2+\delta z^2)}\right]\rho'_d$$

EQUIVALENCE TO USE SAME i-TABLES AS THTAB, CVTAB, & CDTAB IN PHASE 1, + 2 MORE TABLES.

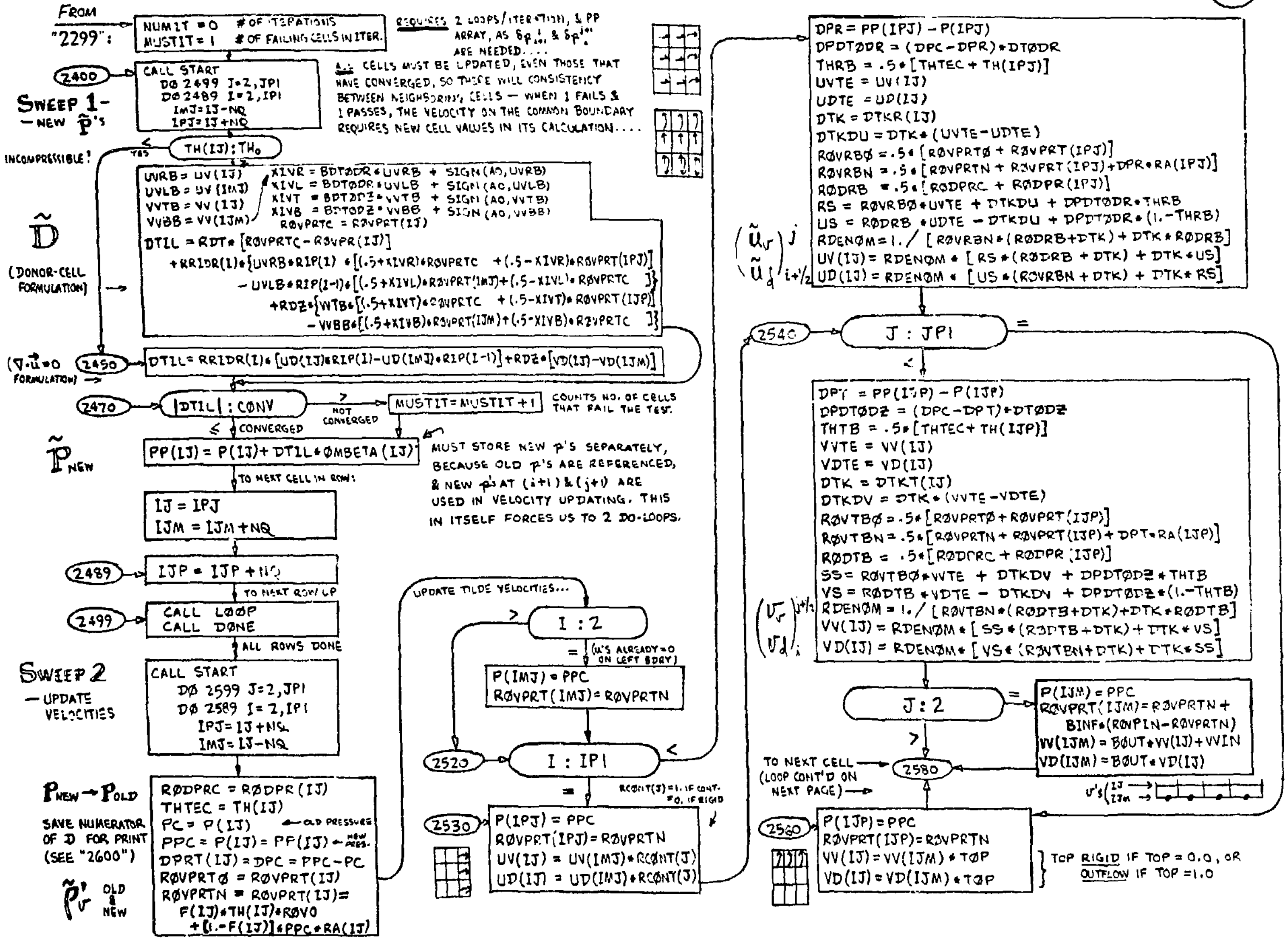
β NOW IN INTERIOR CELLS –

NOTE! β NOT REQ'D FOR EXTERIOR CELLS, & NOT STORED IN i=i, AS THE K_{ij}'s THERE ARE NEEDED IN 3000 →

COMPLETE CONVERGENCE CRITERION:
IF EPS = 10^{-4}, THEN THE LIMIT WITH ZERO VELOCITIES IS $10^{-4}*10^{-6} = 10^{-10}$, WELL WITHIN ACCURACY STANDARDS. (IF INNER ε WERE 10^{-10}, HOWEVER, THE RESULTING 10^{-14} WOULD NOT ALLOW CONVERGENCE, AS IT IS ON THE EDGE OF MACHINE SIGNIFICANCE.)

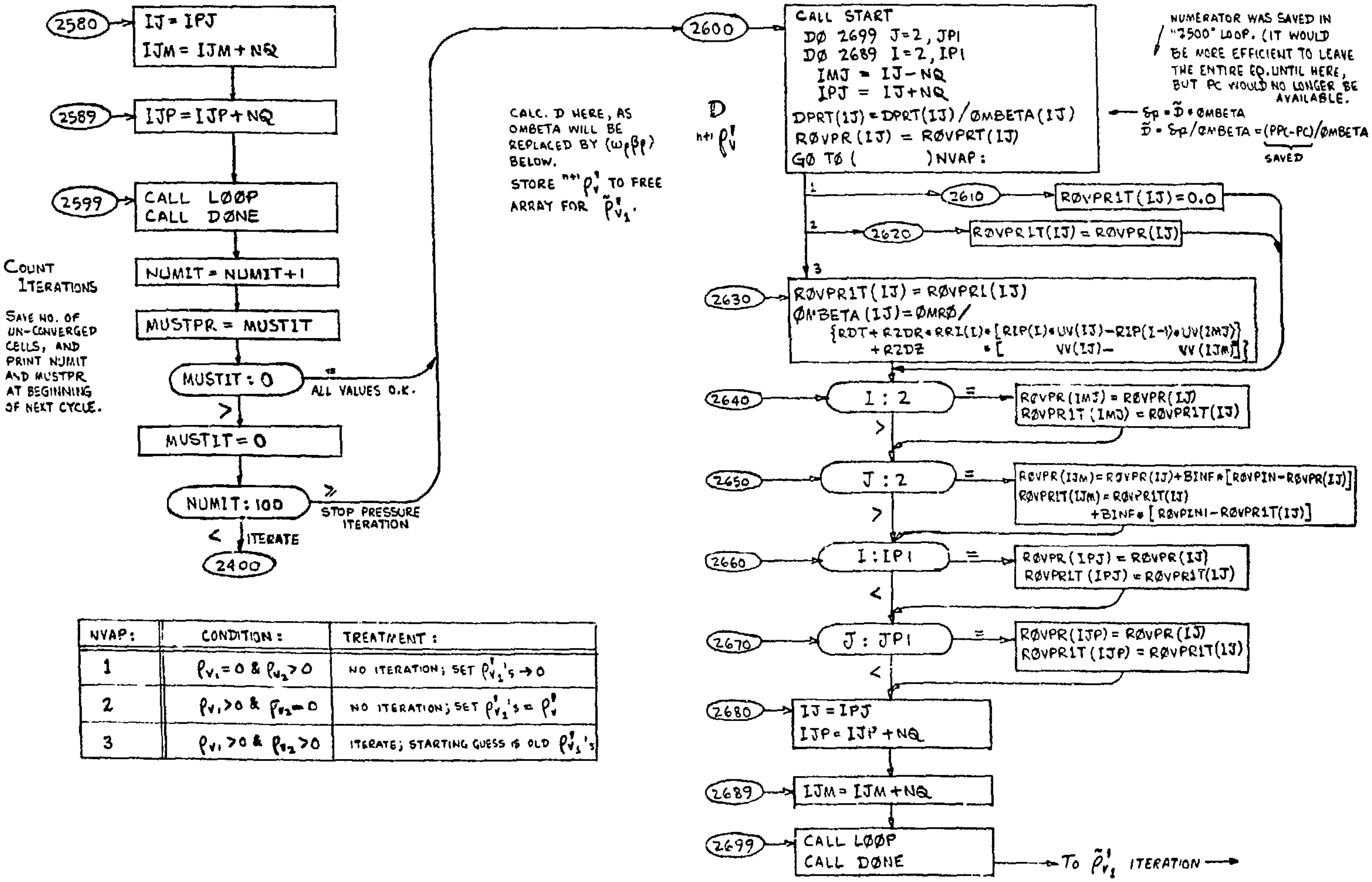
NEXT, THE ITERATION LOOPS →

PHASE 2 - p. 3 - PRESSURE ITERATION:



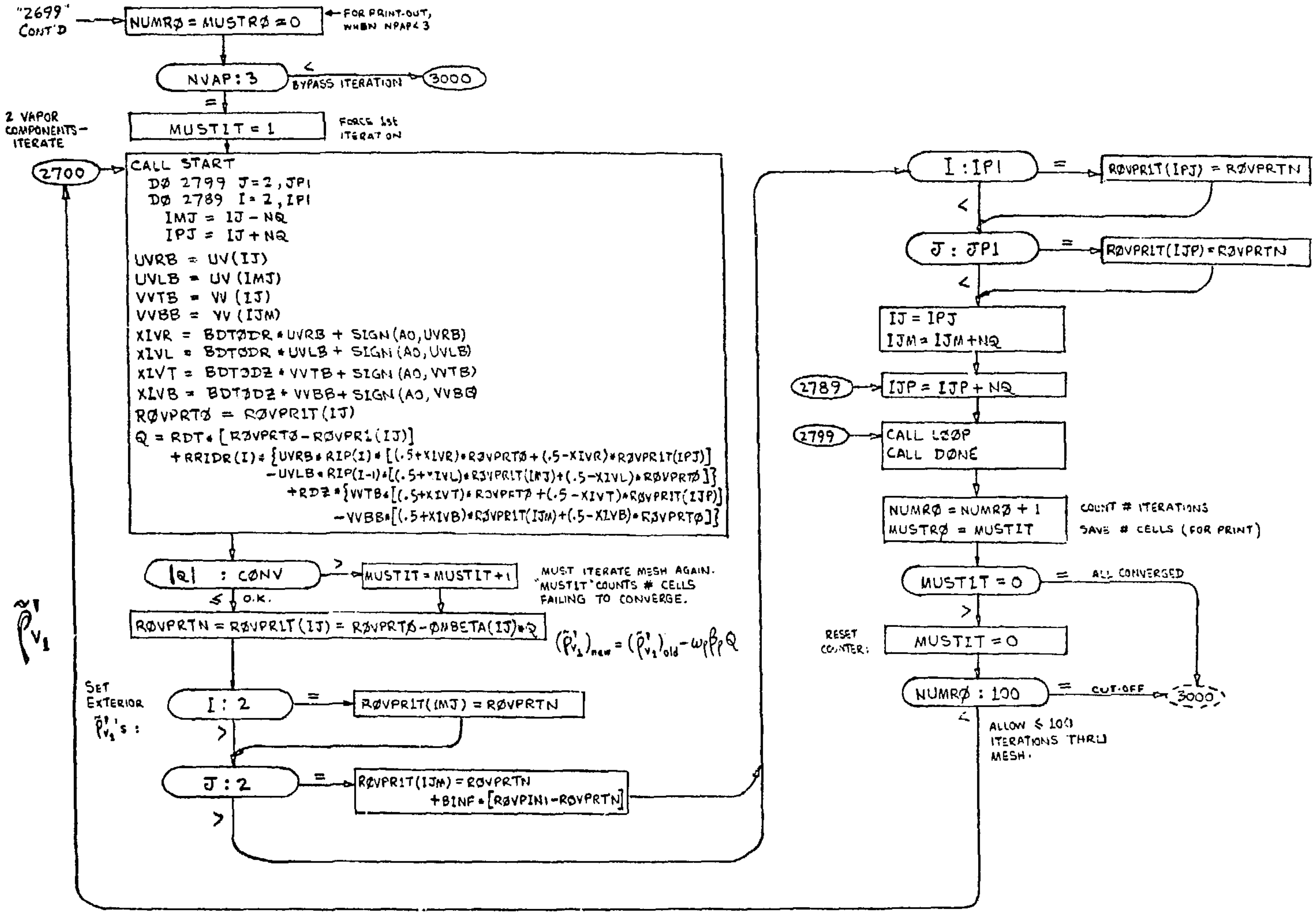
PHASE 2 – p. 4 – END $\tilde{p}$ ITERATION. STORE D, $^{n+1}\rho_v'$, AND $\tilde{\rho}_{v1}'$:

⑳



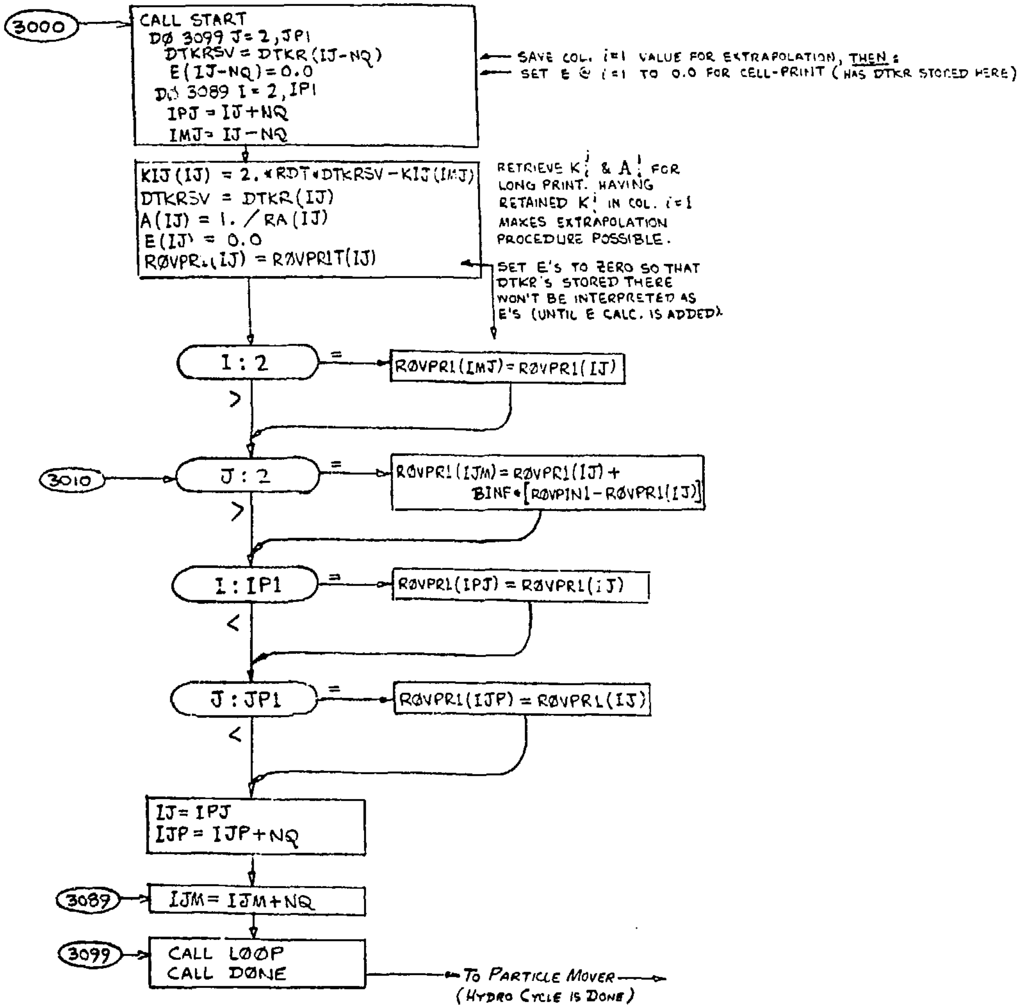
NVAP:	CONDITION:	TREATMENT:
1	$\rho_{v1} = 0$ & $\rho_{v2} > 0$	NO ITERATION; SET ρ_{v1}''s → 0
2	$\rho_{v1} > 0$ & $\rho_{v2} = 0$	NO ITERATION; SET ρ_{v1}''s = ρ_v'
3	$\rho_{v1} > 0$ & $\rho_{v2} > 0$	ITERATE; STARTING GUESS IS OLD ρ_{v1}''s

PHASE 2 – P.5 — $\tilde{\rho}'_{v1}$ ITERATION:



```
CALL START
  DØ 2799 J=2,JP1
  DØ 2789 I=2,IP1
    IMJ = IJ - NQ
    IPJ = IJ + NQ
UVRB = UV(IJ)
UVLB = UV (IMJ)
VVTB = VV (IJ)
VVBB = VV (IJM)
XIVR = BDTØDR*UVRB + SIGN(A0,UVRB)
XIVL = BDTØDR*UVLB + SIGN(A0,UVLB)
XIVT = BDTØDZ*VVTB + SIGN(A0,VVTB)
XIVB = BDTØDZ*VVBB + SIGN(A0,VVBB)
RØVPRTØ = RØVPRIT(IJ)
Q = RDT*[RØVPRTØ-RØVPR1(IJ)]
    +RRIDR(I)*{UVRB*RIP(I)*[(.5+XIVR)*RØVPRTØ+(.5-XIVR)*RØVPRIT(IPJ)]
                -UVLB*RIP(I-1)*[(.5+XIVL)*RØVPRIT(IMJ)+(.5-XIVL)*RØVPRTØ]}
                  +RDZ*{VVTB*[(.5+XIVT)*RØVPRTØ+(.5-XIVT)*RØVPRIT(IJP)]
                       -VVBB*[(.5+XIVB)*RØVPRIT(IJM)+(.5-XIVB)*RØVPRTØ]}
```

PHASE 2 – P. 6 — STORE K, A, AND $^{n+1}\rho'_{VI}$:



```
CALL START
 DØ 3099 J= 2,JP1
   DTKRSV = DTKR(IJ-NQ)
    E(IJ-NQ)=0.0
  DØ 3089 I = 2,IP1
   IPJ = IJ+NQ
   IMJ= IJ-NQ
```

← SAVE COL. i=1 VALUE FOR EXTRAPOLATION, THEN:
← SET E @ i=1 TO 0.0 FOR CELL-PRINT (HAS DTKR STORED HERE)

```
KIJ(IJ) = 2.*RDT*DTKRSV-KIJ(IMJ)
DTKRSV = DTKR(IJ)
A(IJ) = 1./RA(IJ)
E(IJ) = 0.0
RØVPR1(IJ) = RØVPR1T(IJ)
```

RETRIEVE K_i^j & A_i^j FOR LONG PRINT. HAVING RETAINED K_i^j IN COL. i=1 MAKES EXTRAPOLATION PROCEDURE POSSIBLE.

SET E'S TO ZERO SO THAT DTKR'S STORED THERE WON'T BE INTERPRETED AS E'S (UNTIL E CALC. IS ADDED).

```
RØVPR1(IMJ)=RØVPR1(IJ)
```

```
RØVPR1(IJM)=RØVPR1(IJ)+
        BINF*[RØVPIN1-RØVPR1(IJ)]
```

```
RØVPR1(IPJ) = RØVPR1(IJ)
```

```
RØVPR1(IJP) = RØVPR1(IJ)
```

```
IJ= IPJ
IJP= IJP+NQ
```

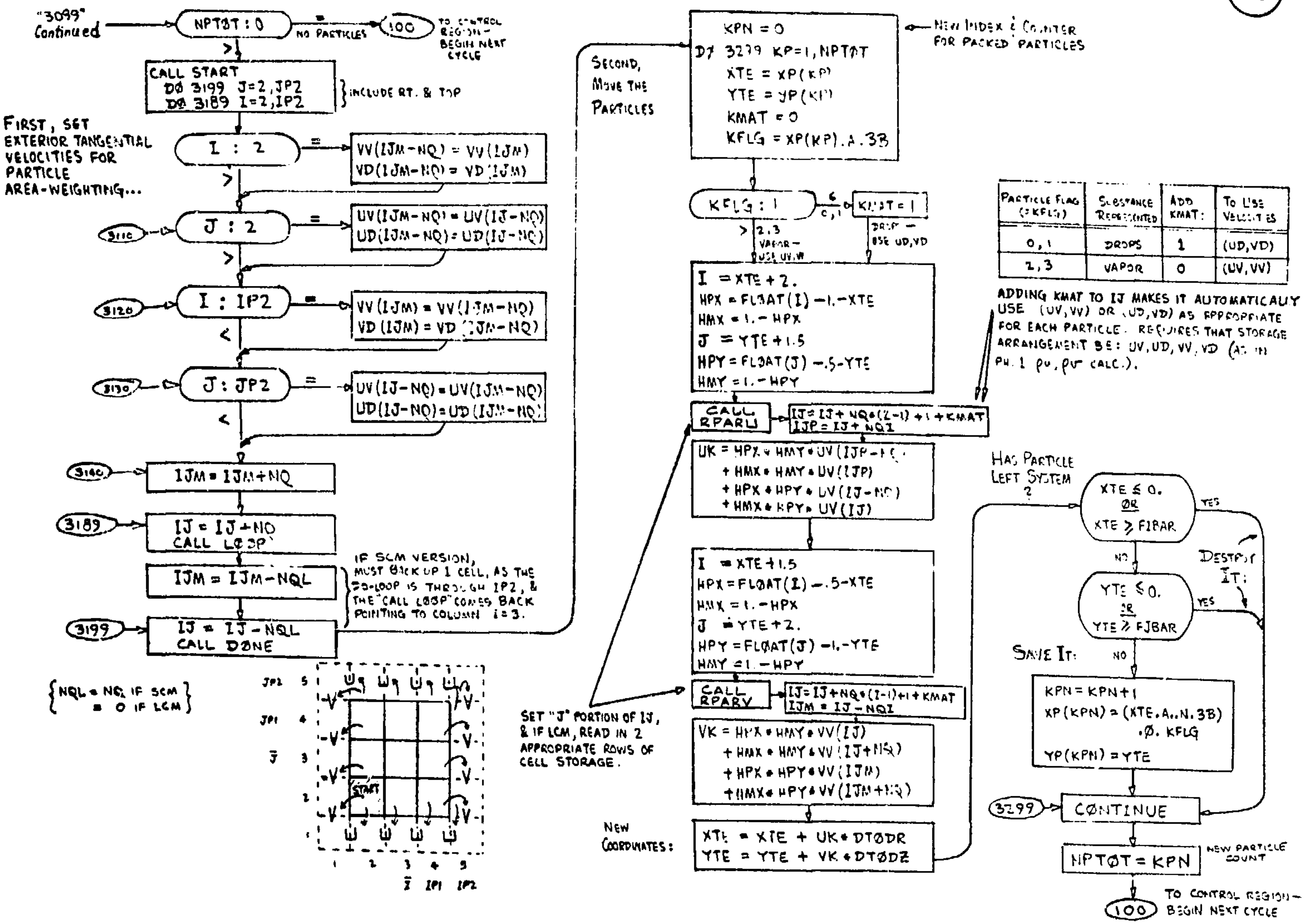
```
IJM= IJM+NQ
```

```
CALL LØØP
CALL DØNE
```

→ TO PARTICLE MOVER →
(HYDRO CYCLE IS DONE)

PARTICLE MOVER:

㉓



PARTICLE FLAG (=KFLG)	SUBSTANCE REPRESENTED	ADD KMAT:	TO USE VELOCITIES
0,1	DROPS	1	(UD,VD)
2,3	VAPOR	0	(UV,VV)

ADDING KMAT TO IJ MAKES IT AUTOMATICALLY USE (UV,VV) OR (UD,VD) AS APPROPRIATE FOR EACH PARTICLE. REQUIRES THAT STORAGE ARRANGEMENT BE: UV,UD,VV,VD (AS IN PH. 1 ρu, ρv CALC.).

APPENDIX B

FORTRAN IV INDEX LISTING OF THE KACHINA PROGRAM

(June 19, 1974 Status)

LASL identification: LP-0335

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

INDEX COMPILED ON 06/11/74
INDEX START TIME IS      2.629 SECONDS

************************************************************************************************************************



   VARIABLES ARE DESCRIBED BY TYPE
     FOR ARRAYS TYPE IS PRECEDED BY ()

          CODE       MEANING OF CODE ASSOCIATED WITH LINE REFERENCES
           •         STATEMENT NUMBER DEFINED
           *         VARIABLE APPEARS ON LEFT OF = SIGN
          AG         ARGUMENT IN SUBROUTINE, FUNCTION OR CALL
          AS         ASSIGN STATEMENT
          BI         BUFFER IN
          BO         BUFFER OUT
          CN         NAME OF LABELLED COMMON
          CO         VARIABLE IN COMMON
          CX         COMPLEX
          DA         DATA STATEMENT
          DP         DOUBLE PRECISION
          DC         DECODE
          DI         DIMENSION
          DO         DO LOOP
          EC         ENCODE
          EN         ENTRY POINT
          EQ         EQUIVALENCE
          F          FUNCTION
          IN         INTEGER
          LA         LARGE
          LC         VARIABLE IN LCM
          LG         LOGICAL
          LX         LEXT ROUTINE
          NM         NAMELIST
          PA         PARAMETER
          PR         PRINT
          PU         PUNCH
          RD         READ
          RL         REAL
          SI         SMALL IN
          SM         SMALL
          SO         SMALL OUT
          SU         SUBROUTINE
          TI         TINY IN
          TO         TINY OUT
          WR         WRITE
```

```
INDEX  01/00/75                     OVERLAY (KACHFIL,0,0)                              PAGE    1
        1          OVERLAY (KACHFIL,0,0)                                              KACHINA    2

INDEX  01/00/75  PROGRAM KACHINA (INP,OUT,FILM,FSET9=OUT,FSET12=FILM,FSET7,FSET8)      PAGE    2
        1          PROGRAM KACHINA (INP,OUT,FILM,FSET9=OUT,FSET12=FILM,FSET7,FSET8)   KACHINA    3
        2          COMMON /KSB/ AA1(1),AAROW(988)                                     KOM        2
        3          COMMON /KSC/ AA(1),AASC(26676),AKINFI,A0,BDTODR,BDTODZ,            KOM        3
                  1            B01,B02,BINF,BOUT,BV1,BV1GM11,BV2,BV2GM12,              KOM        4
                  2            B0,CDR,COLOUR(3),C1,DR,DR02,DRSQ,DT,DT0(10),            KOM        5
                  2            DTOC(10),DTODR,DTODZ,DT02,DTPOS,DZ,DZ02,                KOM        6
                  3            DZSQ,D1,EM10,EM3,EM6,EPS,EPV(2),EP9,EP10,EP20,          KOM        7
                  3            FIBAR,FIXL,FIXR,FIYB,FIYT,FJBAR,G,GAM1,GAM2,GDT,        KOM        8
                  4            GGM11,GGM12,GM11,GM12,I,IABRT,IALL,IBAR,IOTO,IJ,        KOM        9
                  5            IJM,IJP,IP1,IP2,ISPR,IXL,IXR,IYB,IYT,J,JBAR,JNM,JP1,    KOM       10
                  6            JP2,JRIGID,JTOP,JX1,JX2,JX3,J2,J3,KD,KDODRSQ,           KOM       11
                  7            KDODZSQ,KV,KVODRSQ,KVODZSQ,LCM,LPR,MUSTPR,             KOM       12
                  8            MUSTRO,NAME(8),NCYC,NLC,NPTOT,NQ,NQI,NQI2,              KOM       13
                  9            NQL,NQ2,NQ2L,NSC,NUMIT,NUMRO,NUMTD,NUV,                 KOM       14
                  1            NUV3,NVAP,OMBAS,OMBSPL,OMP,OMRO,R,RCONT(66),RDR,        KOM       15
                  1            RDRSQ,RDT,RDZ,RDZSQ,RI(34),RIBAR,RIBJB,RIP(34),         KOM       16
                  2            RJBAR,R01,R02,ROVPIN,ROVPIN1,ROV0,RPAR,RPCDR,RPCOF,     KOM       17
                  3            RRI(34),RRIP(34),RRTDR(34),RR01,RR02,R2DR,R2DZ,         KOM       18
                  4            SIEVIN,SQM0,T,THIN,TH0,TLIMD,TOP,TOUT,TWFIN,            KOM       19
                  5            T20MD,VOL(34),VOLR(34),VVIN,XCONV,XL,XP(4000),XR,       KOM       20
                  6            YB,YCONV,YP(4000),YT,ZZ                                 KOM       21
        4          EQUIVALENCE (AASC(1),TH),(AASC(2),ROVPR1),(AASC(3),RODPR1),        EQVREAL    2
                  1           (AASC(4),ROVPR),(AASC(5),RODPR),(AASC(6),SIEV),          EQVREAL    3
                  2           (AASC(7),SIED),(AASC(8),P),(AASC(9),KIJ,OMBETA),         EQVREAL    4
                  3           (AASC(10),E,ROUTE,DTKR),(AASC(11),ROVTE,DTKT),           EQVREAL    5
                  4           (AASC(12),UV,ROUV),(AASC(13),UD,ROUD),                   EQVREAL    6
                  5           (AASC(14),VV,ROVV),(AASC(15),VD,ROVD),                   EQVREAL    7
                  6           (AASC(16),A,RA),(AASC(17),CQ,ROVPRT,RODPRT,ROVPR1T),     EQVREAL    8
                  7           (AASC(18),PP,DPRT),(AASC(19),F)                          EQVREAL    9
        5          EQUIVALENCE (AAROW(4),ROVSPL)                                      EQVREAL   10
        6          EQUIVALENCE (EPV(2),EPD)                                           EQVREAL   11
        7          REAL KD,KDODRSQ,KDODZSQ,KIJ,KV,KVODRSQ,KVODZSQ,NUV,NUV3            EQVREAL   12
        8          CALL GETQ (4LKJBN,JNM)                                             KACHINA    6
        9          CALL DATE1 (D1)                                                    KACHINA    7
       10          CALL CLOCK1 (C1)                                                   KACHINA    8
       11          NQ = 19                                                            KACHINA    9
       12     10   READ 100, IBAR,JBAR,DR,DZ,A0,B0,ROV0,NUV,CDR,RPAR                  KACHINA   10
       13          IABRT = 0                                                          KACHINA   11
       14          IF (IBAR) 40,30,20                                                 KACHINA   12
       15     20   CALL OVERLAY (7LKACHFIL,1,0,0)                                     KACHINA   13
       16          IF (IABRT.NE.0) GO TO 10                                           KACHINA   14
       17     30   CALL OVERLAY (7LKACHFIL,2,0,0)                                     KACHINA   15
       18          GO TO 10                                                           KACHINA   16
       19     40   CALL EMPTY                                                         KACHINA   17
              C                                                                       KACHINA   18
       20    100   FORMAT (2I4,8F8.3)                                                 KACHINA   19
       21          END                                                                KACHINA   20
```

```
INDEX  01/00/75  PROGRAM KACHINA (INP,OUT,FILM,FSET9=OUT,FSET12=FILM,FSET7,FSET8)          PAGE    3
SINGLY REFERENCED VARIABLES
 A        -R   4EQ   DTODZ   -R   3CO   G       -R   3CO   JTOP     -I   3CO   OMBAS   -R   3CO   ROUV     -R   4EQ   TH      -R   4EQ
 AA      ()R   3CO   DTOZ    -R   3CO   GAM1    -R   3CO   JX1      -I   3CO   OMBETA  -R   4EQ   ROVD     -R   4EQ   THIN    -R   3CO
 AA1     ()R   2CO   DTPOS   -R   3CO   GAM2    -R   3CO   JX2      -I   3CO   OMBSPL  -R   3CO   ROVPIN   -R   3CO   THO     -R   3CO
 AKINF1   -R   3CO   DZOZ    -R   3CO   GUT     -R   3CO   JX3      -I   3CO   OMP     -R   3CO   ROVPIN1  -R   3CO   TLIMD   -R   3CO
 BDTODR   -R   3CO   DZSQ    -R   3CO   GETQ    -    8SU   J2       -I   3CO   OMRO    -R   3CO   ROVPR    -R   4EQ   TOP     -R   3CO
 BDTODZ   -R   3CO   E       -R   4EQ   GGM11   -R   3CO   J3       -I   3CO   P       -R   4EQ   ROVPRT   -R   4EQ   TOUT    -R   3CO
 BD1      -R   3CO   EMPTY   -   19SU   GGM12   -R   3CO   KACHINA  -    1SU   PP      -R   4EQ   ROVPR1   -R   4EQ   TWFIN   -R   3CO
 BD2      -R   3CO   EM10    -R   3CO   GM11    -R   3CO   KSB      -    2CN   R       -R   3CO   ROVPR1T  -R   4EQ   T2OMD   -R   3CO
 BINF     -R   3CO   EM3     -R   3CO   GM12    -R   3CO   KSC      -    3CN   RA      -R   4EQ   ROVSPL   -R   5EQ   UD      -R   4EQ
 BOUT     -R   3CO   EM6     -R   3CO   I       -I   3CO   LCM      -I   3CO   RCONT  ()R   3CO   ROVTE    -R   4EQ   UV      -R   4EQ
 BV1      -R   3CO   EPD     -R   6EQ   IALL    -I   3CO   LPR      -I   3CO   RDR     -R   3CO   ROVV     -R   4EQ   VD      -R   4EQ
 BV1GM11  -R   3CO   EPS     -R   3CO   IDTO    -I   3CO   MUSTPR   -I   3CO   RDRSQ   -R   3CO   RD1      -R   3CO   VOI    ()R   3CO
 BV2      -R   3CO   EP10    -R   3CO   IJ      -I   3CO   MUSTRO   -I   3CO   RDT     -R   3CO   RO2      -R   3CO   VOLR   ()R   3CO
 BV2GM12  -R   3CO   EP20    -R   3CO   IJM     -I   3CO   NAME    ()I   3CO   RDZ     -R   3CO   RPCDR    -R   3CO   VV      -R   4EQ
 CLOCK1   -   10SU   EP9     -R   3CO   IJP     -I   3CO   NCYC     -I   3CO   RDZSQ   -R   3CO   RPCOF    -R   3CO   VVIN    -R   3CO
 COLOUR  ()R   3CO   F       -R   4EQ   INP     -I   1AG   NLC      -I   3CO   READ    -   12F   RRI     ()R   3CO   XCONV   -R   3CO
 CQ       -R   4EQ   FIBAR   -R   3CO   IP1     -I   3CO   NPTOT    -I   3CO   REAL    -    7F   RRIDR   ()R   3CO   XL      -R   3CO
 DATE1    -    9SU   FIXL    -R   3CO   IP2     -I   3CO   NQI      -I   3CO   RI     ()R   3CO   RRIP    ()R   3CO   XP     ()R   3CO
 DPRT     -R   4EQ   FIXR    -R   3CO   ISPR    -I   3CO   NQI2     -I   3CO   RIBAR   -R   3CO   RRO1     -R   3CO   XR      -R   3CO
 DRO2     -R   3CO   FIYB    -R   3CO   IXL     -I   3CO   NQL      -I   3CO   RIBJB   -R   3CO   RRO2     -R   3CO   YA      -R   3CO
 DRSQ     -R   3CO   FIYT    -R   3CO   IXR     -I   3CO   NQ2      -I   3CO   RIP    ()R   3CO   R2DR     -R   3CO   YCONV   -R   3CO
 DT       -R   3CO   FJBAR   -R   3CO   IYB     -I   3CO   NQ2I     -I   3CO   RJBAR   -R   3CO   R2DZ     -R   3CO   YP     ()R   3CO
 DTKQ     -R   4EQ   FORMAT  -    20F   IYT     -I   3CO   NSC      -I   3CO   RODPR   -R   4EQ   SIED     -R   4EQ   YT      -R   3CO
 DTKT     -R   4EQ   FSET12  -R   1AG   J       -I   3CO   NUMTT    -I   3CO   RODPRT  -R   4EQ   SIEV     -R   4EQ   ZZ      -R   3CO
 DTO     ()R   3CO   FSET7   -R   1AG   JP1     -I   3CO   NUMRO    -I   3CO   RODPR1  -R   4EQ   SIEVIN   -R   3CO
 DTOC    ()R   3CO   FSET8   -R   1AG   JP2     -I   3CO   NUMTD    -I   3CO   ROUD    -R   4EQ   SOMO     -R   3CO
 DTODR    -R   3CO   FSET9   -R   1AG   JRIGID  -I   3CO   NVAP     -I   3CO   ROUTE   -R   4EQ   T        -R   3CO
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
MULTIPLY-REFERENCED VARIABLES
     10 -     12*     16      18
     20 -     14      15*
     30 -     14      17*
     40 -     14      19*
    100 -     12RD    20*
AARO*    ()R   2CO    5EQ
AASC     ()R   3CO    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ
               4EQ    4EQ    4EQ
A0        -R   3CO    12RD
B0        -R   3CO    12RD
CDR       -R   3CO    12RD
COMMON    -    2F     3F
C1        -R   3CO    10AG
DR        -R   3CO    12RD
DZ        -R   3CO    12RD
D1        -R   3CO    9AG
FPV      ()R   3CO    6EQ
EQUIVAL   -    4F     5F     6F
FILM      -R   1AG    1AG
IABRT     -I   3CO    13=    16
IBAR      -I   3CO    12RD   14
JBAR      -I   3CO    12RD
JNM       -I   3CO    8AG
KD        -R   3CO    7RL
KDODRSQ   -R   3CO    7RL
KDODZSQ   -R   3CO    7RL
```

```
    INDEX  01/08/75  PROGRAM KACHINA (INP.OUT,FILM,FSET9=OUT,FSET12=FILM,FSET7,FSET8)          PAGE    4
KIJ      -R     4EQ    7RL
KV       -R     3CO    7RL
KVODRSQ  -R     3CO    7RL
KVODZSQ  -R     3CO    7RL
NQ       -I     3CO   11=
NUV      -R     3CO    7RL   12RD
NUV3     -R     3CO    7RL
OUT      -R     1AG    1AG
OVERLAY  -     15SU   17SU
ROVO     -R     3CO   12RD
RPAR     -R     3CO   12RD
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

```
      SUBROUTINE LOOP                                                   KACHINA   21
      COMMON /KSB/ AA1(1),AAROW(988)                                    KOM        2
      COMMON /KSC/ AA(1),AASC(25676),AKINFI,A0,BDTODR,BDTODZ,           KOM        3
     1             BD1,BD2,BINF,BOUT,BV1,BV1GM11,BV2,BV2GM12,           KOM        4
     2             B0,CDR,COLOUR(3),C1,DR,DRO2,DRSQ,DT,DTO(10),         KOM        5
     2             DTOC(10),DTODR,DTODZ,DTO2,DTPOS,DZ,DZO2,             KOM        6
     3             DZSQ,DI,EM10,EM3,EM6,EPS,EPV(2),EP9,EP10,EP20,       KOM        7
     3             FIBAR,FIXL,FIXR,FIYB,FIYT,FJBAR,G,GAM1,GAM2,GDT,     KOM        8
     4             GGM11,GGM12,GM11,GM12,I,IABRT,IALL,IBAR,IDTO,IJ,     KOM        9
     5             IJM,IJP,IP1,IP2,ISPR,IXL,IXR,IYB,IYT,J,JBAR,JNM,JP1, KOM       10
     6             JP2,JRIGID,JTOP,JX1,JX2,JX3,J2,J3,KD,KDODRSQ,        KOM       11
     7             KDODZSQ,KV,KVODRSQ,KVODZSQ,LCM,LPR,MUSTPR,           KOM       12
     8             MUSTHO,NAME(8),NCYC,NLC,NPTOT,NQ,NQI,NQI2,           KOM       13
     9             NQL,NQ2,NQ2L,NSC,NUMIT,NUMRO,NUMTD,NUV,              KOM       14
     1             NUV3,NVAP,OMBAS,OMBSPL,OMP,OMRO,R,RCONT(66),RDR,     KOM       15
     1             RDRSQ,RDT,RDZ,RDZSQ,RI(34),RIBAR,RIBJB,RIP(34),      KOM       16
     2             RJBAR,RO1,RO2,ROVPIN,ROVPIN1,ROVQ,RPAR,RPCDR,RPCOF,  KOM       17
     3             RRI(34),RRIP(34),RRIDR(34),RRO1,RRO2,R2DR,R2DZ,      KOM       18
     4             SIEVIN,SQMO,T,THIN,THO,TLIMD,TOP,TOUT,TWFIN,         KOM       19
     5             T20MD,VOL(34),VOLR(34),VVIN,XCONV,XL,XP(4000),XR,    KOM       20
     6             YB,YCONV,YP(4000),YT,ZZ                              KOM       21
      IJP = IJP + NQ2                                                   KACHINA   23
      IJ = IJ + NQ2                                                     KACHINA   24
      IJM = IJM + NQ2                                                   KACHINA   25
      RETURN                                                            KACHINA   26
      ENTRY START                                                       KACHINA   27
      IJP = JX3                                                         KACHINA   28
      IJ  = JX2                                                         KACHINA   29
      IJM = JX1                                                         KACHINA   30
      RETURN                                                            KACHINA   31
      ENTRY DONE                                                        KACHINA   32
      RETURN                                                            KACHINA   33
      ENTRY R1ROW                                                       KACHINA   34
      JJ = (J-1)*NQI + 1                                                KACHINA   35
      RETURN                                                            KACHINA   36
      ENTRY SETIJ                                                       KACHINA   37
      IJ = JJ + (I-1)*NQ                                                KACHINA   38
      RETURN                                                            KACHINA   39
      ENTRY W1ROW                                                       KACHINA   40
      RETURN                                                            KACHINA   41
      ENTRY LCMFLG                                                      KACHINA   42
      LCM = 0                                                           KACHINA   43
      RETURN                                                            KACHINA   44
      ENTRY RIJP2                                                       KACHINA   45
      KK = (J+1)*NQI                                                    KACHINA   46
      DO 299 K=1,NQI                                                    KACHINA   47
  299 AAROW(K) = AASC(KK,K)                                             KACHINA   48
      RETURN                                                            KACHINA   49
      ENTRY RPARU                                                       KACHINA   50
      IJ = NQI*(J-1)                                                    KACHINA   51
      RETURN                                                            KACHINA   52
      ENTRY RPARV                                                       KACHINA   53
      IJ = NQI*(J-1)                                                    KACHINA   54
      RETURN                                                            KACHINA   55
      END                                                               KACHINA   56
```

```
  INDEX  01/00/75                          SUBROUTINE LOOP
                                                                                   PAGE    6
 SINGLY REFERENCED VARIABLES
  AA      ()R  3C0   DT0     ()R  3C0   FJBAR   -R   3C0   JP2     -I   3C0   NQL     -I   3C0   RIJP2   -    26EN   SQM0    -R   3C0
  AA1     ()R  2C0   DTOC    ()R  3C0   G       -R   3C0   JRIGID  -I   3C0   NQ2L    -I   3C0   RIP     ()R  3C0    START   -    8EN
  AKINFI  -R   3C0   DTODR   -R   3C0   GAM1    -R   3C0   JTOP    -I   3C0   NSC     -I   3C0   [illegible]IBAR -R 3C0 T     -R   3C0
  A0      -R   3C0   DTODZ   -R   3C0   GAM2    -R   3C0   J2      -I   3C0   NUMIT   -I   3C0   RL[illegible]IN -R 3C0 THIN  -R   3C0
  BDTODR  -R   3C0   DT02    -R   3C0   GDT     -R   3C0   J3      -I   3C0   NUMRO   -I   3C0   ROV[illegible]NI -R 3C0 THO  -R   3C0
  BDTODZ  -R   3C0   DTPOS   -R   3C0   GGM11   -R   3C0   KD      -I   3C0   NUMTD   -I   3C0   ROV0    -R   3C0    TLIMD   -R   3C0
  BD1     -R   3C0   DZ      -R   3C0   GGM12   -R   3C0   KDODRSQ -I   3C0   NUV     -I   3C0   R01     -R   3C0    TOP     -R   3C0
  BD2     -R   3C0   DZ02    -R   3C0   GM11    -R   3C0   KDODZSQ -I   3C0   NUV3    -I   3C0   R02     -R   3C0    TOUT    -R   3C0
  BINF    -R   3C0   DZSQ    -R   3C0   GM12    -R   3C0   KSB     -    2CN   NVAP    -I   3C0   RPAR    -R   3C0    TWFIN   -R   3C0
  BOUT    -R   3C0   D1      -R   3C0   IABRT   -I   3C0   KSC     -    3CN   OMBAS   -R   3C0   RPARU   -    31EN   T20MD   -R   3C0
  BV1     -R   3C0   EM10    -R   3C0   IALL    -I   3C0   KV      -I   3C0   OMBSPL  -R   3C0   RPARV   -    34EN   VOL     ()R  3C0
  BV1GM11 -R   3C0   EM3     -R   3C0   IBAR    -I   3C0   KVODRSQ -I   3C0   OMP     -R   3C0   RPCDR   -R   3C0    VOLR    ()R  3C0
  BV2     -R   3C0   EM6     -R   3C0   IDT0    -I   3C0   KVODZSQ -I   3C0   OMRO    -R   3C0   RPCOF   -R   3C0    VVIN    -R   3C0
  BV2GM12 -R   3C0   EPS     -R   3C0   IP1     -I   3C0   LCMFLG  -    23EN  R       -R   3C0   RRI     ()R  3C0    W1ROW   -    21EN
  B0      -R   3C0   EPV     ()R  3C0   IP2     -I   3C0   LOOP    -    1SU   RCONT   ()R  3C0   RRIDR   ()R  3C0    XCONV   -R   3C0
  CDR     -R   3C0   EP10    -R   3C0   ISPR    -I   3C0   LPR     -I   3C0   ROR     -R   3C0   RRIP    ()R  3C0    XL      -R   3C0
  COLOUR  ()R  3C0   EP20    -R   3C0   IXL     -I   3C0   MUSTPR  -I   3C0   RDRSQ   -R   3C0   RR01    -R   3C0    XP      ()R  3C0
  CI      -R   3C0   EP9     -R   3C0   IXR     -I   3C0   MUSTRO  -I   3C0   RDT     -R   3C0   RR02    -R   3C0    XR      -R   3C0
  DONE    -    13EN  FIBAR   -R   3C0   IYB     -I   3C0   NAME    ()I  3C0   RDZ     -R   3C0   R1ROW   -    15EN   YR      -R   3C0
  DR      -R   3C0   FIXL    -R   3C0   IYT     -I   3C0   NCYC    -I   3C0   RDZSQ   -R   3C0   R2DR    -R   3C0    YCONV   -R   3C0
  DRO2    -R   3C0   FIXR    -R   3C0   JBAR    -I   3C0   NLC     -I   3C0   RI      ()R  3C0   R2DZ    -R   3C0    YP      ()R  3C0
  DRSG    -R   3C0   FIYB    -R   3C0   JNM     -I   3C0   NPTOT   -I   3C0   RIBAR   -R   3C0   SETIJ   -    18EN   YY      -R   3C0
  DT      -R   3C0   FIYT    -R   3C0   JP1     -I   3C0   NQI2    -I   3C0   RIBJB   -R   3C0   SIEVIN  -R   3C0    Z7      -R   3C0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 MULTIPLY-REFERENCED VARIABLES
    299 -     28D0    29=
 AAROW   ()R   2C0    29=
 AASC    ()R   3C0    29
 COMMON  -     2F      3F
 ENTRY   -     8F     13F     15F     18F     21F     23F     26F     31F     34F
 I       -I    3C0    19
 IJ      -I    3C0     5=     5       10=     19=     32=     35=
 IJM     -I    3C0     6=     6       11=
 IJP     -I    3C0     4=     4        9=
 J       -I    3C0    16      27      32      35
 JJ      -I   16=     19
 JX1     -I    3C0    11
 JX2     -I    3C0    10
 JX3     -I    3C0     9
 K       -I   28D0    29      29
 KK      -I   27=     29
 LCM     -I    3C0    24=
 NQ      -I    3C0    19
 NQI     -I    3C0    16      27      28D0    32      35
 NQ2     -I    3C0     4       5       6
 RETURN  -     7F     12F     14F     17F     20F     22F     25F     30F     33F     36F
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
INDEX  01/00/75                  OVERLAY (KACHFIL,1,0)                              PAGE    7
        1          OVERLAY (KACHFIL,1.0)                                           KASET       2



INDEX  01/00/75                  PROGRAM KASET                                      PAGE    8
        1          PROGRAM KASET                                                   KASET       3
        2          PRINT 10                                                        KASET       4
        3          CALL SETUP                                                      KASET       5
        4     10   FORMAT (1H1* SETUP*)                                            KASET       6
        5          END                                                             KASET       7



INDEX  01/00/75                  PROGRAM KASET                                      PAGE    9
SINGLY REFERENCED VARIABLES
 FORMAT  -    4F    KASET   -     1SU   PRINT   -    2F    SETUP   -    3SU
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
MULTIPLY-REFERENCED VARIABLES
    10 -     2PR    4*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
      SUBROUTINE SETUP                                                  KASET     8
      COMMON /KSB/ AA1(1),AAROW(988)                                    KOM       2
      COMMON /KSC/ AA(1),AASC(26676),AKINFI,A0,BDTODR,BDTODZ,           KOM       3
     1             BD1,BD2,BINF,BOUT,BV1,BV1GM11,BV2,BV2GM12,           KOM       4
     2             B0,CDR,COLOUR(3),C1,DR,DRO2,DRSQ,DT,DTO(10),         KOM       5
     2             DTOC(10),DTODR,DTODZ,DTO2,DTPDS,DZ,DZO2,             KOM       6
     3             DZSQ,D1,EM10,EM3,EM6,EPS,EPV(2),EP9,EP10,EP20,       KOM       7
     3             FIBAR,FIXL,FIXR,FIYB,FIYT,FJBAR,G,GAM1,GAM2,GDT,     KOM       8
     4             GGM11,GGM12,GM11,GM12,I,IABRT,IALL,IBAR,IDTO,IJ,     KOM       9
     5             IJM,IJP,IP1,IP2,ISPR,IXL,IXR,IYB,IYT,J,JBAR,JNM,JP1, KOM      10
     6             JP2,JRIGID,JTOP,JX1,JX2,JX3,J2,J3,KD,KDODRSQ,        KOM      11
     7             KDODZSQ,KV,KVODRSQ,KVODZSQ,LCM,LPR,MUSTPR,           KOM      12
     8             MUSTRO,NAME(8),NCYC,NLC,NPTOT,NQ,NQI,NQI2,           KOM      13
     9             NQL,NQ2,NQ2L,NSC,NUMIT,NUMRO,NUMTD,NUV,              KOM      14
     1             NUV3,NVAP,OMBAS,OMBSPL,OMP,OMRO,R,RCONT(66),RDR,     KOM      15
     1             RDRSQ,RDT,RDZ,RDZSQ,RI(34),RIBAR,RIBJB,RIP(34),      KOM      16
     2             RJBAR,RO1,RO2,ROVPIN,ROVPIN1,ROV0,RPAR,RPCDR,RPCOF,  KOM      17
     3             RRI(34),RRIP(34),RRIDR(34),RRO1,RRO2,R2DR,R2DZ,      KOM      18
     4             SIEVIN,SOM0,T,THIN,THO,TLIMD,TOP,TOUT,TWFIN,         KOM      19
     5             T20MD,VOL(34),VOLR(34),VVIN,XCONV,XL,XP(4000),XR,    KOM      20
     6             YB,YCONV,YP(4000),YT,ZZ                              KOM      21
      EQUIVALENCE (AASC(1),TH),(AASC(2),ROVPR1),(AASC(3),RODPR1),       EQVREAL   2
     1            (AASC(4),ROVPR),(AASC(5),RODPR),(AASC(6),SIEV),       EQVREAL   3
     2            (AASC(7),SIED),(AASC(8),P),(AASC(9),KIJ,OMBETA),      EQVREAL   4
     3            (AASC(10),E,ROUTE,DTKR),(AASC(11),ROVTE,DTKT),        EQVREAL   5
     4            (AASC(12),UV,ROUV),(AASC(13),UD,ROUD),                EQVREAL   6
     5            (AASC(14),VV,ROVV),(AASC(15),VD,ROVD),                EQVREAL   7
     6            (AASC(16),A,RA),(AASC(17),CQ,ROVPRT,RODPRT,ROVPRIT),  EQVREAL   8
     7            (AASC(18),PP,DPRT),(AASC(19),F)                       EQVREAL   9
      EQUIVALENCE (AAROW(4),ROVSPL)                                     EQVREAL  10
      EQUIVALENCE (EPV(2),EPD)                                          EQVREAL  11
      REAL KD,KDODRSQ,KDODZSQ,KIJ,KV,KVODRSQ,KVODZSQ,NUV,NUV3           EQVREAL  12
      DIMENSION TH(1),ROVPR1(1),RODPR1(1),ROVPR(1),RODPR(1),SIEV(1),    DIMEN     2
     1          SIED(1),P(1),KIJ(1),OMBETA(1),E(1),ROUTE(1),DTKR(1),    DIMEN     3
     2          ROVTE(1),DTKT(1),UV(1),ROUV(1),UD(1),ROUD(1),VV(1),     DIMEN     4
     3          ROVV(1),VD(1),ROVD(1),A(1),RA(1),CG(1),ROVPRT(1),       DIMEN     5
     4          RODPRT(1),ROVPRIT(1),PP(1),DPRT(1),F(1)                 DIMEN     6
      DIMENSION ROVSPL(1)                                               DIMEN     7
      DATA PI / 3.1415 92653 58979 32384 626 /                          KASET    12
      DIMENSION IDCOMP(8),KMATS(4)                                      KASET    13
      DATA KMATS /0B,1B,2B,3B/                                          KASET    14
      CALL LCMFLG                                                       KASET    15
      READ 810, NAME                                                    KASET    16
      READ 810, IDCOMP                                                  KASET    17
      READ 820, OMP,OMRO,EPS,G,KV,KD,EPV(1),EPD,R                       KASET    18
      READ 830, RO1,RO2,GAM1,GAM2,BV1,BV2,BD1,BD2                       KASET    19
      READ 835, JRIGID,IBOT,THIN,ROVIN1,ROVIN2,SIEVIN,VVIN,ITOP         KASET    20
      READ 840, T,DT,T20MD,TLIMD,TWFIN,LPR,ISPR,(COLOUR(N),N=1,3)       KASET    21
      READ 845, (DTO (N),N=1,10)                                        KASET    22
      READ 845, (DTOC(N),N=1,10)                                        KASET    23
      KT = 9                                                            KASET    24
      ASSIGN 110 TO KRET                                                KASET    25
  100 WRITE (KT,810) NAME                                               KASET    26
      WRITE (KT,810) IDCOMP                                             KASET    27
      WRITE (KT,850) IBAR,JBAR,DR,DZ,A0,B0,ROV0,NUV,CDR,RPAR            KASET    28
      WRITE (KT,860) OMP,OMRO,EPS,G,KV,KD,EPV(1),EPD,R                  KASET    29
```

```
            WRITE (KT,870) RO1,RO2,GAM1,GAM2,BV1,BV2,BD1,BD2                   KASET    30
            WRITE (KT,875) JRIGID,IBOT,THIN,ROVIN1,ROVIN2,SIEVIN,VVIN,ITOP     KASET    31
            WRITE (KT,880) T,DT,T20MD,TLIMD,TWFIN,LPR,ISPR,(COLOUR(N),N=1,3)   KASET    32
            WRITE (KT,890) (DTO(N),N=1,10)                                     KASET    33
            WRITE (KT,900) (DTOC(N),N=1,10)                                    KASET    34
            GO TO KRET                                                         KASET    35
      110   IF (LPR.EQ.0) GO TO 120                                            KASET    36
            KT = 12                                                            KASET    37
            ASSIGN 120 TO KRET                                                 KASET    38
            GO TO 100                                                          KASET    39
      120   IP1 = IBAR + 1                                                     KASET    40
            IP2 = IBAR + 2                                                     KASET    41
            JP1 = JBAR + 1                                                     KASET    42
            JP2 = JBAR + 2                                                     KASET    43
            NQ2 = NQ + NQ                                                      KASET    44
            NQI = NQ * IP2                                                     KASET    45
            NQI2 = NQI + NQI                                                   KASET    46
            NQL = NQ*(1-LCM)                                                   KASET    47
            NQ2L = NQL + NQL                                                   KASET    48
            IALL = JP2 * NQI                                                   KASET    49
            JX1 =  1 + NQ                                                      KASET    50
            J2  =  1 + NQI                                                     KASET    51
            JX2 = J2 + NQ                                                      KASET    52
            J3  = J2 + NQI                                                     KASET    53
            JX3 = J3 + NQ                                                      KASET    54
            FIBAR = FLOAT(IBAR)                                                KASET    55
            FJBAR = FLOAT(JBAR)                                                KASET    56
            RIBAR = 1. / FIBAR                                                 KASET    57
            RJBAR = 1. / FJBAR                                                 KASET    58
            RIBJB = RIBAR * RJBAR                                              KASET    59
            NSC = LOCF(ZZ) - LOCF(AA) + 1                                      KASET    60
            NLC = LOCF(AA1(JP2*NQI)) - LOCF(AA1) + 1                           KASET    61
            IDTO = 1                                                           KASET    62
            TOUT = T + DTO(1)                                                  KASET    63
            DTPOS = DT                                                         KASET    64
            NCYC = NPTOT = NUMTD = 0                                           KASET    65
            EM10 = 1.E-10                                                      KASET    66
            EM6 = 1.E-6                                                        KASET    67
            EM3 = 1.E-3                                                        KASET    68
            EP9 = 1.E+9                                                        KASET    69
            EP10 = 1.E+10                                                      KASET    70
            EP20 = 1.E+20                                                      KASET    71
            OMP = -OMP                                                         KASET    72
            AKINF1 = EP10                                                      KASET    73
            THO = 0.02                                                         KASET    74
            RRO1 = 1. / RO1                                                    KASET    75
            RRO2 = 1. / RO2                                                    KASET    76
            GM11  = GAM1 - 1.                                                  KASET    77
            GGM11 = GAM1 * GM11                                                KASET    78
            BV1GM11 = BV1 * GM11                                               KASET    79
            GM12  = GAM2 - 1.                                                  KASET    80
            GGM12 = GAM2 * GM12                                                KASET    81
            BV2GM12 = BV2 * GM12                                               KASET    82
            SQMO = (.5)**2                                                     KASET    83
            RPCOF = 1.5/RPAR**2                                                KASET    84
            NUV3 = 3. * NUV                                                    KASET    85
```

```
      RPCDR = .25 * RPAR * CDR                                          KASET  86
      RDR = 1. / DR                                                     KASET  87
      DR02 = .5 * DR                                                    KASET  88
      RDZ = 1. / DZ                                                     KASET  89
      DZ02 = .5 * DZ                                                    KASET  90
      RDRSQ = RDR * RDR                                                 KASET  91
      RDZSQ = RDZ * RDZ                                                 KASET  92
      DRSQ = DR * DR                                                    KASET  93
      DZSQ = DZ * DZ                                                    KASET  94
      OMBAS = .5*OMP*DRSQ*DZSQ / (DRSQ+DZSQ)                            KASET  95
      R2DR = .5 * RDR                                                   KASET  96
      R2DZ = .5 * RDZ                                                   KASET  97
      KVODRSQ = KV * RDRSQ                                              KASET  98
      KVODZSQ = KV * RDZSQ                                              KASET  99
      KDODRSQ = KD * RDRSQ                                              KASET 100
      KDODZSQ = KD * RDZSQ                                              KASET 101
      RDT = 1. / DT                                                     KASET 102
      DT02 = .5 * DT                                                    KASET 103
      DTODR = DT * RDR                                                  KASET 104
      BDTODR = B0 * DTODR                                               KASET 105
      DTODZ = DT * RDZ                                                  KASET 106
      BDTODZ = B0 * DTODZ                                               KASET 107
      GDT = G * DT                                                      KASET 108
      OMBSPL = OMBAS * RDT                                              KASET 109
      VCON = 2.*PI * DR*DZ                                              KASET 110
      RI(1) = -.5 * DR                                                  KASET 111
      RRI(1) = -2. * RDR                                                KASET 112
      RRIDR(1) = RRI(1) * RDR                                           KASET 113
      RIP(1) = RRIP(1) = 0.                                             KASET 114
      DO 129 I=2,IP2                                                    KASET 115
      RI(I) = RI(I-1) + DR                                              KASET 116
      RRI(I) = 1. / RI(I)                                               KASET 117
      RRIDR(I) = RRI(I) * RDR                                           KASET 118
      RIP(I) = RIP(I-1) + DR                                            KASET 119
      VOL(I) = VCON*RI(I)                                               KASET 120
      VOLR(I) = VCON*RIP(I)                                             KASET 121
  129 RRIP(I) = 1. / RIP(I)                                             KASET 122
      DO 139 J=2,JP2                                                    KASET 123
      RCONT(J) = 0.                                                     KASET 124
      IF (J.GE.JRIGID+2) RCONT(J) = 1.                                  KASET 125
  139 CONTINUE                                                          KASET 126
      BOUT = BINF = 0.                                                  KASET 127
      IF (IBOT.EQ.1) BOUT = 1.                                          KASET 128
      IF (IBOT.EQ.2) BINF = 1.                                          KASET 129
      ROVPIN1 = THIN * ROVIN1                                           KASET 130
      ROVPIN  = THIN * (ROVIN1+ROVIN2)                                  KASET 131
      TOP = ITOP                                                        KASET 132
      JTOP = JP1 + ITOP                                                 KASET 133
      XR = FIBAR * DR                                                   KASET 134
      YT = FJBAR * DZ                                                   KASET 135
      XL = YB = 0.                                                      KASET 136
      FIYB = 916.                                                       KASET 137
      XD = XR / (YT-YB)                                                 KASET 138
      YY = 0.                                                           KASET 139
      IF (XD.LE.1.13556) YY = 1.                                        KASET 140
      FIXL = AMAX1(0.,(511.-450.*XD)*YY)                                KASET 141
```

```
          FIXR = (511.+450.*XD)*YY + 1022.*(1.-YY)                             KASET   142
          FIYT = 16.*YY + (916.-1022./XD)*(1.-YY)                              KASET   143
          XCONV = (FIXR-FIXL) / (XR-XL)                                        KASET   144
          YCONV = (FIYT-FIYB) / (YT-YB)                                        KASET   145
          IXL = FIXL                                                           KASET   146
          IXR = FIXR                                                           KASET   147
          IYB = FIYB                                                           KASET   148
          IYT = FIYT                                                           KASET   149
          NQM1 = NQ - 1                                                        KASET   150
          CALL START                                                           KASET   151
          DO 199 J=1,JP2                                                       KASET   152
          DO 189 I=1,IP2                                                       KASET   153
          KF = IJM - NQ                                                        KASET   154
          KL = KF + NQM1                                                       KASET   155
          DO 179 K=KF,KL                                                       KASET   156
  179     AASC(K) = 0.                                                         KASET   157
  189     IJM = IJM + NQ                                                       KASET   158
          CALL LOOP                                                            KASET   159
  199     IJM = IJM - NQ2L                                                     KASET   160
          CALL DONE                                                            KASET   161
          SROV1 = SROV2 = 0.                                                   KASET   162
  200     READ 910, NB,NR,NT,NL,RODPRI1,RODPRI2,ROV1,ROV2,SIEVI,SIEDI,NPUA     KASET   163
          IF (NR.EQ.0) GO TO 300                                               KASET   164
          PRINT 920, NB,NR,NT,NL,RODPRI1,RODPRI2,ROV1,ROV2,SIEVI,SIEDI,NPUA    KASET   165
          IF (LPR.GT.0) WRITE (12,920) NB,NR,NT,NL,RODPRI1,RODPRI2,            KASET   166
         1                            ROV1,ROV2,SIEVI,SIEDI,NPUA               KASET   167
          SROV1 = SROV1 + ROV1                                                 KASET   168
          SROV2 = SROV2 + ROV2                                                 KASET   169
          NB2 = NB + 2                                                         KASET   170
          NR1 = NR + 1                                                         KASET   171
          NT1 = NT + 1                                                         KASET   172
          NL2 = NL + 2                                                         KASET   173
          THI = 1. - RODPRI1*RRO1 - RODPRI2*RRO2                               KASET   174
          IF (THI.LT.0.) GO TO 400                                             KASET   175
          ROVPRI1 = THI * ROV1                                                 KASET   176
          ROVPRI = THI * (ROV1+ROV2)                                           KASET   177
          RODPRI = RODPRI1 + RODPRI2                                           KASET   178
          ROVPRI2 = ROVPRI - ROVPRI1                                           KASET   179
          PNI = SIEVI*(ROVPRI1*BV1GM11 + ROVPRI2*BV2GM12)                      KASET   180
         1        / (THI*(ROVPRI1*BV1    + ROVPRI2*BV2)) * ROVPRI              KASET   181
          DO 219 J=NB2,NT1                                                     KASET   182
          CALL RIROW                                                           KASET   183
          DO 209 I=NL2,NR1                                                     KASET   184
          CALL SETIJ                                                           KASET   185
          TH(IJ) = THI                                                         KASET   186
          ROVPR1(IJ) = ROVPRI1                                                 KASET   187
          ROVPR(IJ) = ROVPRI                                                   KASET   188
          RODPR1(IJ) = RODPRI1                                                 KASET   189
          RODPR(IJ) = RODPRI                                                   KASET   190
          SIEV(IJ) = SIEVI                                                     KASET   191
          SIED(IJ) = SIEDI                                                     KASET   192
          P(IJ) = PNI                                                          KASET   193
  209     IJ = IJ + NQ                                                         KASET   194
          CALL WIROW                                                           KASET   195
  219     CONTINUE                                                             KASET   196
          IF (NPUA.EQ.0) GO TO 200                                             KASET   197
```

```
            FNB = FLOAT (NB)                                        KASET    198
            FNR = FLOAT (NR)                                        KASET    199
            FNT = FLOAT (NT)                                        KASET    200
            FNL = FLOAT (NL)                                        KASET    201
            WIDTH = FNR - FNL                                       KASET    202
            HEIGHT = FNT - FNB                                      KASET    203
            FNPUA = FLOAT(NPUA)                                     KASET    204
            RODRAT = RODPRI1 / RODPRI                               KASET    205
            OFFSET = 0.                                             KASET    206
            DO 299 NTYPE=1,3                                        KASET    207
            GO TO (230,232,234) NTYPE                               KASET    208
   230      PNEFF = FNPUA * RODRAT*(1.-THI)                         KASET    209
            GO TO 240                                               KASET    210
   232      PNEFF = FNPUA * (1.-RODRAT)*(1.-THI)                    KASET    211
            GO TO 240                                               KASET    212
   234      PNEFF = FNPUA * THI                                     KASET    213
   240      XNP = WIDTH*QSQRT(PNEFF)                                KASET    214
            YNP = HEIGHT*XNP/WIDTH                                  KASET    215
            NPX = XNP+.5+EM10                                       KASET    216
            NPY = YNP+.5+EM10                                       KASET    217
            IF (NPX*NPY.EQ.0) GO TO 299                             KASET    218
            XSPAC = WIDTH / FLOAT(NPX)                              KASET    219
            X1 = FNL + XSPAC*(.5+.25*OFFSET)                        KASET    220
            YSPAC = HEIGHT / FLOAT(NPY)                             KASET    221
            Y1 = FNB + YSPAC*(.5+.25*OFFSET)                        KASET    222
            KMAT = KMATS(NTYPE)                                     KASET    223
            DO 289 J=1,NPY                                          KASET    224
            XTE = X1                                                KASET    225
            DO 279 I=1,NPX                                          KASET    226
            NPTOT = NPTOT + 1                                       KASET    227
            XP(NPTOT) = (XTE.A..N.3B) .O. KMAT                      KASET    228
            YP(NPTOT) = Y1                                          KASET    229
   279      XTE = XTE + XSPAC                                       KASET    230
   289      Y1 = Y1 + YSPAC                                         KASET    231
   299      CONTINUE                                                KASET    232
            GO TO 200                                               KASET    233
   300      CALL START                                              KASET    234
            DO 399 J=2,JP1                                          KASET    235
            DO 389 I=2,IP1                                          KASET    236
            IF (I.EQ.2) GO TO 330                                   KASET    237
   310      IF (J.EQ.2) GO TO 335                                   KASET    238
   315      IF (I.EQ.IP1) GO TO 340                                 KASET    239
   320      IF (J.EQ.JP1) GO TO 345                                 KASET    240
            GO TO 380                                               KASET    241
   330      IN = IJ - NQ                                            KASET    242
            KRET = 1                                                KASET    243
            GO TO 350                                               KASET    244
   335      IN = IJM                                                KASET    245
            KRET = 2                                                KASET    246
            TH(IN) = TH(IJ) + BINF*(THIN-TH(IJ))                    KASET    247
            ROVPR1(IN) = ROVPR1(IJ) + BINF*(ROVPIN1-ROVPR1(IJ))     KASET    248
            ROVPR(IN) = ROVPR(IJ) + BINF*(ROVPIN-ROVPR(IJ))         KASET    249
            SIEV(IN) = SIEV(IJ) + BINF*(SIEVIN-SIEV(IJ))            KASET    250
            VV(IN) = BINF*VVIN                                      KASET    251
            GO TO 360                                               KASET    252
   340      IN = IJ + NQ                                            KASET    253
```

```
           KRET = 3
           GO TO 350                                                    KASET     254
     345   IN = IJP                                                     KASET     255
           KRET = 4                                                     KASET     256
     350   TH(IN) = TH(IJ)                                              KASET     257
           ROVPR1(IN) = ROVPR1(IJ)                                      KASET     258
           ROVPR(IN) = ROVPR(IJ)                                        KASET     259
           SIEV(IN) = SIEV(IJ)                                          KASET     260
     360   RODPR1(IN) = RODPR1(IJ)                                      KASET     261
           RODPR(IN) = RODPR(IJ)                                        KASET     262
           SIED(IN) = SIED(IJ)                                          KASET     263
           GO TO (310,315,320,380) KRET                                 KASET     264
     380   IJP = IJP + NQ                                               KASET     265
           IJM = IJM + NQ                                               KASET     266
     389   IJ = IJ + NQ                                                 KASET     267
     399   CALL LOOP                                                    KASET     268
           CALL DONE                                                    KASET     269
           NVAP = 1                                                     KASET     270
           IF (SROV1.GT.0. .A. SROV2.EQ.0.) NVAP = 2                    KASET     271
           IF (SROV1.GT.0. .A. SROV2.GT.0.) NVAP = 3                    KASET     272
           PRINT 930, NPTOT                                             KASET     273
           IF (LPR.GT.0) WRITE (12,930) NPTOT                           KASET     274
           RETURN                                                       KASET     275
     400   PRINT 940, THI                                               KASET     276
           IABRT = 1                                                    KASET     277
           RETURN                                                       KASET     278
     C                                                                  KASET     279
     810   FORMAT (8A10)                                                KASET     280
     820   FORMAT (9F8.3)                                               KASET     281
     830   FORMAT (8F8.3)                                               KASET     282
     835   FORMAT (2I4,5F8.3,I4)                                        KASET     283
     840   FORMAT (5F8.3,2I4,3F4.1)                                     KASET     284
     845   FORMAT (10F8.3)                                              KASET     285
     850   FORMAT (3X*IBAR=*I4/3X*JBAR=*I4/5X*DR=*1PE12.5/5X*DZ=*E12.5/5X*A0= KASET     286
          1*E12.5/5X*B0=*E12.5/3X*ROV0=*E12.5/4X*NUV=*E12.5/4X*CDR=*E12.5/3X* KASET     287
          2RPAR=*E12.5)                                                 KASET     288
     860   FORMAT (4X*OMP=*1PE12.5/3X*OMRO=*E12.5/4X*EPS=*E12.5/6X*G=*E12.5/ KASET     289
          15X*KV=*E12.5/5X*KD=*E12.5/4X*EPV=*E12.5/4X*EPD=*E12.5/6X*R=*E12.5) KASET     290
     870   FORMAT (4X*RO1=*1PE12.5/4X*RO2=*E12.5/3X*GAM1=*E12.5/3X*GAM2=* KASET     291
          1E12.5/4X*BV1=*E12.5/4X*BV2=*E12.5/4X*BD1=*E12.5/4X*BD2=*E12.5) KASET     292
     875   FORMAT (* JRIGID=*I4/3X*IBOT=*I4/3X*THIN=*1PE12.5/* ROVIN1=*E12.5/ KASET     293
          1* ROVIN2=*E12.5/* SIEVIN=*E12.5/3X*VVIN=*E12.5/3X*ITOP=*I4)  KASET     294
     880   FORMAT (6X*T=*1PE12.5/5X*DT=*E12.5/*  T20MD=*E12.5/*  TLIMD=*E12.5 KASET     295
          1/*  TWFIN=*E12.5/4X*LPR=*I4/3X*ISPR=*I4/* COLOUR(1-3)=*      KASET     296
          2 3(0PF3.1,  ))                                               KASET     297
     890   FORMAT (*  DTO(1-10)=*5(1PE12.5,2X)/12X,5(E12.5,2X))         KASET     298
     900   FORMAT (* DTOC(1-10)=*5(1PE12.5,2X)/12X,5(E12.5,2X))         KASET     299
     910   FORMAT (4I4,6F8.3,I4)                                        KASET     300
     920   FORMAT (*  NB=*I3*  NR=*I3*  NT=*I3*  NL=*I3*  RODPR11=*1PE12.5* KASET     301
          1RODPR12=*E12.5*  ROV1=*E12.5*  ROV2=*E12.5/36X*SIEV1=*E12.5,4X*SIE KASET     302
          2V2=*E12.5*  NPUA=*I4)                                        KASET     303
     930   FORMAT (I10* PARTICLES GENERATED. SETUP COMPLETED.*)         KASET     304
     940   FORMAT (10X*NEGATIVE THETA,=*1PE12.5)                        KASET     305
           END                                                          KASET     306
                                                                        KASET     307
```

```
   INDEX  01/30/75                       SUBROUTINE SETUP                                         PAGE   16

 SINGLY REFERENCED VARIABLES
  AMAX1     -  139SU    JNM      -I    3CO    LCMFLG  -    13SU    NUMIT    -I    3CO    REAL     -     7F    SETUP    -     1SU
  C1        -R   3CO    KSB      -     2CN    MUSTPR  -I    3CO    NUMRO    -I    3CO    RIROW    -  179SU    WIROW    -   191SU
  D1        -R   3CO    KSC      -     3CN    MUSTRO  -I    3CO    QSORT    -  210SU    SETIJ    -  181SU
*****************************************************************************************************************************
 MULTIPLY-REFERENCED VARIABLES
     100 -     24*     37
     110 -     23AS    34*
     120 -     34      36AS    38*
     129 -    113DO   120*
     139 -    121DO   124*
     179 -    154DO   155*
     189 -    151DO   156*
     199 -    150DO   158*
     200 -    161*    193     229
     209 -    180DO   190*
     219 -    178DO   192*
     230 -    204     205*
     232 -    204     207*
     234 -    204     209*
     240 -    206     208     210*
     279 -    222DO   226*
     289 -    224DO   227*
     299 -    203DO   214     228*
     300 -    152     230*
     310 -    234*    261
     315 -    235*    261
     320 -    236*    261
     330 -    233     238*
     335 -    234     241*
     340 -    235     249*
     345 -    236     257*
     350 -    240     251     254*
     360 -    248     250*
     380 -    237     261     262*
     389 -    232DO   264*
     399 -    231DO   265*
     400 -    172     273*
     810 -      14RD    15RD    24WR    25WR   276*
     820 -      16RD   277*
     830 -      17RD   278*
     835 -      18RD   279*
     840 -      19RD   280*
     845 -      20RD    21RD   281*
     850 -      26WR   282*
     860 -      27WR   283*
     870 -      28WR   284*
     875 -      29WR   285*
     880 -      30WR   286*
     890 -      31WR   287*
     900 -      32WR   288*
     910 -     161RD   289*
     920 -     163RD   164WR   290*
     930 -     270RD   271WR   291*
     940 -     273RD   292*
 A       0R     4EQ     8DI
```

```
     INDEX  01/00/75                             SUBROUTINE SETUP
                                                                                    PAGE   17
AA       ()R     3C0   58
AAROW    ()R     2C0    5EQ
AASC     ()R     3C0    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ    4EQ
                 4EQ    4EQ    4EQ  155=
AA1      ()R     2C0   59     59
AKINFI    -R     3C0   71=
ASSIGN    -     23F    36F
A0        -R     3C0   26WR
BDTODR    -R     3C0  103=
BDTODZ    -R     3C0  105=
BD1       -R     3C0   17RD   28WR
BD2       -R     3C0   17RD   28WR
BINF      -R     3C0  125=   127=   243    244    245    246    247
BOUT      -R     3C0  125=   126=
BV1       -R     3C0   17RD   28WR   77    177
BV1GM11   -R     3C0   77=   177
BV2       -R     3C0   17RD   28WR   80    177
BV2GM12   -R     3C0   80=   177
B0        -R     3C0   26WR  103    105
COR       -R     3C0   26WR   84
COLOUR   ()R     3C0   19RD   30WR
COMMON    -      2F     3F
CQ       ()R     4EQ    8DI
DATA      -     10F    12F
DIMENSI   -      8F     9F    11F
DONE      -   1595U  2665U
DPRT     ()R     4EQ    8DI
DR        -R     3C0   26WR   85     86     91     91    108    109    114    117    132
DR02      -R     3C0   86=
DRSQ      -R     3C0   91=    93     93
DT        -R     3C0   19RD   30WR   62    100    101    102    104    106
DTKR     ()R     4EQ    8DI
DTKT     ()R     4EQ    8DI
DTO      ()R     3C0   20RD   31WR   61
DTOC     ()R     3C0   21RD   32WR
DTODR     -R     3C0  102=   103
DTODZ     -R     3C0  104=   105
DTO2      -R     3C0  101=
DTPOS     -R     3C0   62=
DZ        -R     3C0   26WR   87     88     92     92    108    133
DZ02      -R     3C0   88=
DZSQ      -R     3C0   92=    93     93
E        ()R     4EQ    8DI
EM10      -R     3C0   64=   212    213
EM3       -R     3C0   66=
EM6       -R     3C0   65=
EPD       -R     6EQ   16RD   27WR
EPS       -R     3C0   16RD   27WR
EPV      ()R     3C0    6EQ   16RD   27WR
EP10      -R     3C0   68=    71
EP20      -R     3C0   69=
EP9       -R     3C0   67=
EQUIVAL   -      4F     5F     6F
F        ()R     4EQ    8DI
FIBAR     -R     3C0   53=    55    132
FIXL      -R     3C0  139=   142    144
```

```
FIXR     -R     3CO   140=   142    145
FIYB     -R     3CO   135=   143    146
FIYT     -R     3CO   141=   143    147
FJBAR    -R     3CO    54=    56    133
FLOAT    -     53SU   54SU   194SU  195SU  196SU  197SU  200SU  215SU  217SU
FNB      -R   194=    199    218
FNL      -R   197=    198    216
FNPU4    -R   290=    205    207    209
FNR      -R   195=    198
FNT      -R   196=    199
FORMAT   -    276F    277F   278F   279F   2[illegible]  [illegible]81F   282F   283F   284F   285F   286F   287F   288F   289F   290F   291F   292F
G        -R     3CO    16RD   27WR  106
GAM1     -R     3CO    17RD   20WR   75     1
GAM2     -R     3CO    17RD   28WR   78     79
GDT      -R     3CO   104=
GGM11    -R     3CO    76=
GGM12    -R     3CO    79=
GM11     -R     3CO    75=    76     77
GM12     -R     3CO    78=    79     80
HEIGHT   -R   199=    211    217
I        -I     3CO  11300   114    114    115    115    116    116    117    117    118    118    119    119    120    120    15100
              18000  22200  23200   233    235
IABRT    -I     3CO   274=
IALL     -I     3CO    47=
IBAR     -I     3CO    26WR   38     39     53
IBOT     -I    18RD    29WR  126    127
IOCOMP   ()I   11DI    15RD   25WR
IOTO     -I     3CO    60=
IJ       -I     3CO   182    183    184    185    186    187    188    189    190=   190    238    243    243    244    244    245
              245     246    246    249    254    255    256    257    258    259    260    264=   264
IJM      -I     3CO   152    156=   156    158=   158    241    263=   263
IJP      -I     3CO   252    262=   262
IN       -I   238=    241=   243    244    245    246    247    249=   252=   254    255    256    257    258    259    260
IP1      -I     3CO    38=  23200   235
IP2      -I     3CO    39=    63  11300  15100
ISPR     -I     3CO    19RD   30WR
ITOP     -I    18RD    29WR  130    131
IXL      -I     3CO   144=
IXR      -I     3CO   145=
IYB      -I     3CO   146=
IYT      -I     3CO   147=
J        -I     3CO  12100   122    123    123  15000  17800  22000  23100   234    236
JBAR     -I     3CO    26WR   40     41     54
JP1      -I     3CO    40=   131  23100   236
JP2      -I     3CO    41=    47     59  12100  15000
JRIGID   -I     3CO    18RD   29WR  123
JTOP     -I     3CO   131=
JX1      -I     3CO    48=
JX2      -I     3CO    50=
JX3      -I     3CO    52=
J2       -I     3CO    49=    50     51
J3       -I     3CO    51=    52
K        -I   15400   155
KD       -R     3CO     7RL   16RD   27WR    98     99
KOODRSQ  -R     3CO     7RL   98=
KOODZSQ  -R     3CO     7RL   99=
```

```
     INDEX  01/00/75                                    SUBROUTINE SETUP                                         PAGE   19
KF       -I   152=    153     15400
KIJ      ()R    4EQ     7RL     8DI
KL       -I   153=    15400
KMAT     -I   219=    224
KMATS    ()I    11DI    12DA   219
KRET     -I   23AS.   33      36AS   239=    242=    250=    253=    261
KT       -I   22=     24WR    25WR    26WR    27WR    28WR    29WR    30WR    31WR    32WR    35=
KV       -R     3CO     7RL    16RD    27WR    96      97
KVODRSQ  -R     3CO     7RL    96=
KVOD2SQ  -R     3CO     7RL    97=
LCH      -I     3CO    45
LOCF     -     58SU    58SU    59SU    59SU
LOOP     -    157SU   265SU
LPR      -I     3CO    19RD    30WR    34      154     271
N        -I    19RD    19RD    20RD    20RD    21RD    21RD    30WR    30WR    31WR    31WR    32WR    32WR
NAME     ()I    3CO    14RD    24WR
NB       -I   161RD   163PR   164WR   167     194
NB2      -I   167=    178DO
NCYC     -I     3CO    63=
NL       -I   161RD   163PR   164WR   170     197
NLC      -I     3CO    59=
NL2      -I   170=    180DO
NPTOT    -I     3CO    63=    223=    223     224     225     270PR   271WR
NPUA     -I   161RD   163PR   164WR   193     200
NPX      -I   212=    214     215     222DO
NPY      -I   213=    214     217     220DO
NQ       -I     3CO    42      42      43      45      44      50      52      148     152     156     190     238     249     262     263     264
NQ1      -I     3CO    43=     64      44      47      49      51      59
NQ12     -I     3CO    44=
NQL      -I     3CO    45=     46      46
NQH1     -I   148=    151
NQ2      -I     3CO    42=
NQ2L     -I     3CO    46=    158
NR       -I   161RD   162     163PR   164WR   168     195
NR1      -I   168=    180DO
NSC      -I     3CO    56=
NT       -I   161RD   163PR   164WR   169     196
NTYPE    -I   203DO   204     219
NT1      -I   169=    178DO
NUMTD    -I     3CO    63=
NUV      -R     3CO     7RL    26WR    83
NUV3     -R     3CO     7RL    83=
NVAP     -I     3CO   267=    268=    269=
OFFSET   -R   207=    216     218
OMBAS    -R     3CO    93=    107
OMBETA   ()R    4EQ     8DI
OMBSPL   -R     3CO   107=
OMP      -R     3CO    16RD    27WR    70=     70      93
OMRO     -R     3CO    16RD    27WR
P        ()R    4EQ     8DI   189=
PI       -R    10DA    10R
PNEFF    -R   205=    207=    209=    210
PN1      -R   177=    189
PP       ()R    4EQ     8DI
PRINT    -    163F    270F    273F
R        -R     3CO    16RD    27WR
```

```
RA       ()R      4EQ    8DI
RCONT    ()R      3CO   122=    123=
RDR      -R       3CO    85=     89      86      94     102     110     111     116
RDRSQ    -R       3CO    84=     96      98
RDI      -R       3CO   100=    107
RD2      -R       3CO    87=     90      93      95     104
RD2SQ    -R       3CO    90=     97      99
READ     -       14F    15F     16F     17F     18F     19F     20F     21F    161F
RETURN   -      272F   275F
RI       ()R      3CO   109=    114=    114     115     118
RIBAR    -R       3CO    55=     57
RIRJB    -R       3CO    57=
RIP      ()R      3CO   112=    117=    117     119     120
RJBAR    -R       3CO    56=     57
RODPR    ()R      4EQ    8DI    186=    259=    259
RODPRI   -R     175=    186     201
RODPHI1  -R     161RD  163PR   164WR   171     175     185     201
RODPHI2  -R     161RD  163PR   164WR   171     175
RODPRT   ()R      4EQ    8DI
RODPR1   ()R      4EQ    8DI    185=    258=    258
RODRAT   -R     201=    205     207
ROLD     ()R      4EQ    8DI
ROUTE    ()R      4EQ    8DI
ROUV     ()R      4EQ    8DI
ROVD     ()R      4EQ    8DI
ROVIN1   -R      18RD   29WR    128     129
ROVIN2   -R      18RD   29WR    129
ROVPIN   -R       3CO   129=    245
ROVPIN1  -R       3CO   128=    244
ROVPR    ()R      4EQ    8DI    184=    245=    245     245     256=    256
ROVPRI   -R     174=    176     177     184
ROVPRI1  -R     173=    176     177     177     183
ROVPRI2  -R     176=    177     177
ROVPRT   ()R      4EQ    8DI
ROVPR1   ()R      4EQ    8DI    183=    244=    244     244     255=    255
ROVPR1T  ()R      4EQ    8DI
ROVSPL   ()R      5EQ    9DI
ROVTE    ()R      4EQ    8DI
ROVV     ()R      4EQ    8DI
ROV0     -R       3CO   26WR
ROV1     -R     161RD  163PR   164WR   165     173     174
ROV2     -R     161RD  163PR   164WR   166     174
RO1      -R       3CO   17RD    28WR    73
RO2      -R       3CO   17RD    28WR    74
RPAR     -R       3CO   26WR    82      86
RPCOR    -R       3CO    84=
RPCOF    -R       3CO    82=
RRI      ()R      3CO   110=    111     115=    116
RRIDR    ()R      3CO   111=    116=
RRIP     ()R      3CO   112=    120=
RRO1     -R       3CO    73=    171
RRO2     -R       3CO    74=    171
R2OR     -R       3CO    94=
R2OZ     -R       3CO    95=
SIED     ()R      4EQ    8DI    188=    260=    260
SIEDI    -R     161RD  163PR   164WR   188
```

```
SIEV     ()R     4EQ     8DI    187=    246=    246     246     257=    257
SIEV1    -R   161RD   163PR   164WR    177     187
SIEVIN   -R     3CO    18RD    29WR    246
SONO     -R     3CO     81=
SROV1    -R    160=    165=    165     268     269
SROV2    -R    160=    166=    166     268     269
START    -    149SU   230SU
T        -R     3CO    19RD    30WR     61
TH       ()R     4EQ     8DI    182=    243=    243     243     254=    254
TH1      -R    171=    172     173     174     177     182     205     207     209     273PR
THIN     -R     3CO    18RD    29WR    128     129     243
THO      -R     3CO     72=
TLIMD    -R     3CO    19RD    30WR
TOP      -R     3CO    130=
TOUT     -R     3CO     61=
TWFIN    -R     3CO    19RD    30WR
T20MD    -R     3CO    19RD    30WR
UD       ()R     4EQ     8DI
UV       ()R     4EQ     8DI
VCON     -R    108=    118     119
VD       ()R     4EQ     8DI
VOL      ()R     3CO    118=
VOLR     ()R     3CO    119=
VV       ()R     4EQ     8DI    247=
VVIN     -R     3CO    18RD    29WR    247
WIDTH    -R    198=    210     211     215
WRITE    -      24F     25F     26F     27F     28F     29F     30F     31F     32F    164F    271F
XCONV    -R     3CO    142=
XD       -R    136=    138     139     140     141
XL       -R     3CO    134=    142
XNP      -R    210=    211     212
XP       ()R     3CO    224=
XR       -R     3CO    132=    136     142
XSPAC    -R    215=    216     226
XTE      -R    221=    224     226=    226
X1       -R    216=    221
YB       -R     3CO    134=    136     143
YCONV    -R     3CO    143=
YNP      -R    211=    213
YP       ()R     3CO    225=
YSPAC    -R    217=    218     227
YT       -R     3CO    133=    136     143
YY       -R    137=    138=    139     140     140     141     141
Y1       -R    218=    225     227=    227
ZZ       -R     3CO     58
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
INDEX  01/30/75                      OVERLAY (KACHFIL,2,0)                                  PAGE   22
       1          OVERLAY (KACHFIL,2,0)                                                    KACHYDR      2
```

```
INDEX  01/00/75                      PROGRAM KACHYDR                                        PAGE   23
       1          PROGRAM KACHYDR                                                          KACHYDR      3
       2          PRINT 10                                                                 KACHYDR      4
       3          CALL HYDRO                                                               KACHYDR      5
       4     10   FORMAT (* HYDRO*)                                                        KACHYDR      6
       5          END                                                                      KACHYDR      7
```

```
INDEX  01/00/75                      PROGRAM KACHYDR                                        PAGE   24
 SINGLY REFERENCED VARIABLES
  FORMAT  -     4F    HYDRO    -     3SU   KACHYDR -     1SU   PRINT    -     2F
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
 MULTIPLY-REFERENCED VARIABLES
    10 -     2PR    4*
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

```
INDEX  01/00/75                          SUBROUTINE HYDRO
                                                                                  PAGE   25
     1           SUBROUTINE HYDRO
     2           COMMON /KSB/ AA1(1),AAROW(988)                                  KACHYDR     8
     3           COMMON /KSC/ AA(1),AASC(26676),AKINFI,A0,BDTODR,BDTODZ,          KOM         2
               1              BD1,BD2,BINF,BOUT,BV1,BV1GM11,BV2,BV2GM12,          KOM         3
               2              B0,CDQ,COLOUR(3),C1,DR,DR02,DRSQ,DT,DTO(10),        KOM         4
               2              DTOC(10),DTODR,DTODZ,DT02,DTPOS,DZ,DZ02,            KOM         5
               3              DZSQ,D1,EM10,EM3,EM6,EPS,EPV(2),EP9,EP10,EP20,      KOM         6
               3              FIBAR,FIXL,FIXR,FIYB,FIYT,FJBAR,G,GAM1,GAM2,GDT,    KOM         7
               4              GGM11,GGM12,GM11,GM12,I,IABRT,IALL,IBAR,IDTO,IJ,    KOM         8
               5              IJM,IJP,IP1,IP2,ISPR,IXL,IXR,IYB,IYT,J,JBAR,JNM,JP1, KOM        9
               6              JP2,JRIGID,JTOP,JX1,JX2,JX3,J2,J3,KD,KDODRSQ,       KOM        10
               7              KDODZSQ,KV,KVODRSQ,KVODZSQ,LCM,LPR,MUSTPR,          KOM        11
               8              MUSTRO,NAME(8),NCYC,NLC,NPTOT,NQ,NQI,NQI2,          KOM        12
               9              NQL,NQ2,NQ2L,NSC,NUMIT,NUMRO,NUMTD,NUV,             KOM        13
               1              NUV3,NVAP,OMBAS,OMBSPL,OMP,OMRO,R,RCONT(66),RDR,    KOM        14
               1              RDRSQ,RDT,RDZ,RDZSQ,RI(34),RIBAR,RIBJB,RIP(34),     KOM        15
               2              RJBAR,RO1,RO2,ROVPIN,ROVPIN1,ROV0,RPAR,RPCDR,RPCOF, KOM        16
               3              RRI(34),RRIP(34),RRIDR(34),RRO1,RRO2,R2DR,R2DZ,     KOM        17
               4              SIEVIN,SOMD,T,THIN,THO,TLIMD,TOP,TOUT,TWFIN,        KOM        18
               5              T2OMD,VOL(34),VOLR(34),VVIN,XCONV,XL,XP(4000),XR,   KOM        19
               6              YB,YCONV,YP(4000),YT,ZZ                             KOM        20
     4           EQUIVALENCE (AASC(1),TH),(AASC(2),ROVPR1),(AASC(3),RODPR1),      KOM        21
               1             (AASC(4),ROVPR),(AASC(5),RODPR),(AASC(6),SIEV),      EQVREAL     2
               2             (AASC(7),SIED),(AASC(8),P),(AASC(9),KIJ,OMBETA),     EQVREAL     3
               3             (AASC(10),E,ROUTE,DTKR),(AASC(11),ROVTE,DTKT),       EQVREAL     4
               4             (AASC(12),UV,ROUV),(AASC(13),UD,ROUD),               EQVREAL     5
               5             (AASC(14),VV,ROVV),(AASC(15),VD,ROVD),               EQVREAL     6
               6             (AASC(16),A,RA),(AASC(17),CQ,ROVPRT,RODPRT,ROVPR1T), EQVREAL     7
               7             (AASC(18),PP,DPRT),(AASC(19),F)                      EQVREAL     8
     5           EQUIVALENCE (AAROW(4),ROVSPL)                                    EQVREAL     9
     6           EQUIVALENCE (EPV(2),EPD)                                         EQVREAL    10
     7           REAL KD,KDODRSQ,KDODZSQ,KIJ,KV,KVODRSQ,KVODZSQ,NUV,NUV3          EQVREAL    11
     8           DIMENSION TH(1),ROVPR1(1),RODPR1(1),ROVPR(1),RODPR(1),SIEV(1),   EQVREAL    12
               1           SIED(1),P(1),KIJ(1),OMBETA(1),E(1),ROUTE(1),DTKR(1),   DIMEN       2
               2           ROVTE(1),DTKT(1),UV(1),ROUV(1),UD(1),ROUD(1),VV(1),    DIMEN       3
               3           ROVV(1),VD(1),ROVD(1),A(1),RA(1),CQ(1),ROVPRT(1),      DIMEN       4
               4           RODPRT(1),ROVPR1T(1),PP(1),DPRT(1),F(1)                DIMEN       5
     9           DIMENSION ROVSPL(1)                                              DIMEN       6
    10           DIMENSION THTAB(34),RODITAB(34),SIEVTAB(34),SIEDTAB(34),         DIMEN       7
               1           THTET(1),RODT(1),PVDENT(1),TERM1T(1),TERM2T(34),       KACHYDR    12
               2           IX1(1),IX2(1),IY1(1),IY2(1),X1(4),Y1(4),CON(11),       KACHYDR    13
               3           SPSUMS(20),KMATS(4),IDPP(2,3),ICHARS(3),VELMX(2),      KACHYDR    14
               4           TOVV(2,2),IDCON(2,10)                                  KACHYDR    15
    11           EQUIVALENCE (THTAB,IX1),(THTAB(2),IX2),(THTAB(3),IY1),           KACHYDR    16
               1             (THTAB(4),IY2),(THTAB(5),X1),(THTAB(9),Y1),          KACHYDR    17
               2             (THTAB(13),CON),(THTAB,THTET),(RODITAB,RODT),        KACHYDR    18
               3             (SIEVTAB,PVDENT),(SIEDTAB,TERM1T)                    KACHYDR    19
    12           EQUIVALENCE (RODT,SPSUMS),(RODT(1),STH),(RODT(2),SMOMR),         KACHYDR    20
               1             (RODT(3),SMOMZV),(RODT(4),SMOMZD),(RODT(5),SMOMZ),   KACHYDR    21
               2             (RODT(6),SMV1),(RODT(7),SMD1),(RODT(8),SINTEV),      KACHYDR    22
               3             (RODT(9),SINTED),(RODT(10),SINTE),(RODT(11),SMV2),   KACHYDR    23
               4             (RODT(12),SMD2),(RODT(13),SKEV),(RODT(14),SKED),     KACHYDR    24
               5             (RODT(15),SKE),(RODT(16),SMV),(RODT(17),SMD),        KACHYDR    25
               6             (RODT(18),SEV),(RODT(19),SED),(RODT(20),SE)          KACHYDR    26
    13           DATA KMATS  /0B,1H,2B,3B/                                        KACHYDR    27
    14           DATA IDPP   /16H(* DROPLETS, 1*),16H(* DROPLETS, 2*)/,           KACHYDR    28
                                                                                  KACHYDR    29
```

```
      1              16H(* VAPOR      *)/                                   KACHYDR   30
       DATA ICHARS /16*63,42/                                               KACHYDR   31
       DATA IDVV   /14H(* VAPOR,    *),14H(* DROPLETS,*)/                   KACHYDR   32
       DATA IDCON  /16H(* VOID FRACTN*),     16H(* RHO-V PR.,1*),           KACHYDR   33
      1              16H(* RHO-D PR.,1*),     16H(* RHO-V PRIME*),          KACHYDR   34
      2              16H(* RHO-D PRIME*),     14H(* SIE-VAPOR*),            KACHYDR   35
      3              14H(* SIE-DROPS*),       13H(* PRESSURE*),             KACHYDR   36
      4              13H(* DRAG (K)*),        15H(* ENERGY (E)*)/           KACHYDR   37
       DATA TOLD,T2 /0.,0./                                                 KACHYDR   38
       CALL SECOND (TBASE)                                                  KACHYDR   39
       T1 = TBASE                                                           KACHYDR   40
       CALL GETQ (4LKTLM,II)                                                KACHYDR   41
       TLIM = II                                                            KACHYDR   42
       IF (IBAR.EQ.0) GO TO 270                                             KACHYDR   43
       TLIM = TLIM*27.5E-9 - 40. + (1.-TLIMD)*EP10                          KACHYDR   44
  100  CONTINUE                                                             KACHYDR   45
C         CALC. OPTIONAL E ARRAY HERE, = CHEMICAL OR NUCLEAR ENERGY.        KACHYDR   46
       TOLD = T2                                                            KACHYDR   47
       CALL SECOND (T2)                                                     KACHYDR   48
       XX = (T2-TOLD)*RIBJR                                                 KACHYDR   49
       PRINT 4120, T,NCYC,DT,T2,XX,NUMIT,MUSTPR,NUMRO,MUSTRO                KACHYDR   50
       IF (LPR.GT.0) WRITE (12,4120) T,NCYC,DT,T2,XX,NUMIT,MUSTPR,NUMRO,    KACHYDR   51
      1                                MUSTRO                               KACHYDR   52
       CALL EMPTY                                                           KACHYDR   53
       ASSIGN 120 TO KROUT                                                  KACHYDR   54
       IF (T+EM10.GE.TOUT) GO TO 210                                        KACHYDR   55
       IF (NCYC.LE.1) GO TO 300                                             KACHYDR   56
       IF (ISPR.EQ.2) GO TO 900                                             KACHYDR   57
  120  IF (T2-T1.GE.1200. .A. T20MD.GT.EM10) GO TO 220                      KACHYDR   58
       IF (T2-TBASE.GE.TLIM) GO TO 230                                      KACHYDR   59
  130  IF (T.GE.TWFIN) RETURN                                               KACHYDR   60
       NCYC = NCYC + 1                                                      KACHYDR   61
       IF (LPR.EQ.0 .O. NCYC.LE.10) GO TO 150                               KACHYDR   62
       VELMX(1) = VELMX(2) = EM10                                           KACHYDR   63
       CALL START                                                           KACHYDR   64
       DO 149 J=1,JP1                                                       KACHYDR   65
       DO 139 I=2,IP1                                                       KACHYDR   66
       VELMX(1) = AMAX1(VELMX(1),ABS(UV(IJM)),ABS(UD(IJM)))                 KACHYDR   67
       VELMX(2) = AMAX1(VELMX(2),ABS(VV(IJM)),ABS(VD(IJM)))                 KACHYDR   68
  139  IJM = IJM + NQ                                                       KACHYDR   69
  149  CALL LOOP                                                            KACHYDR   70
       DTPOS = 0.1*AMIN1(DR/VELMX(1),DZ/VELMX(2))                           KACHYDR   71
  150  DT = DTPOS                                                           KACHYDR   72
       IF (T+DT.GT.TOUT) DT = TOUT - T                                      KACHYDR   73
       T = T + DT                                                           KACHYDR   74
       RDT = 1. / DT                                                        KACHYDR   75
       DT02 = .5 * DT                                                       KACHYDR   76
       DTODR = DT * RDR                                                     KACHYDR   77
       BDTODR = B0 * DTODR                                                  KACHYDR   78
       DTODZ = DT * RDZ                                                     KACHYDR   79
       BDTODZ = B0 * DTODZ                                                  KACHYDR   80
       GDT = G * DT                                                         KACHYDR   81
       OMBSPL = OMBAS * RDT                                                 KACHYDR   82
       GO TO 1000                                                           KACHYDR   83
  210  TOUT = TOUT + DTO(IDTO)                                              KACHYDR   84
       IF (T+EM10.LT.DTOC(IDTO)) GO TO 300                                  KACHYDR   85
```

```
            TOUT = DTOC(IDTO) + DTO(IDTO + 1)                           KACHYDR   86
            IDTO = IDTO + 1                                             KACHYDR   87
            GO TO 300                                                   KACHYDR   88
      220   T1 = T2                                                     KACHYDR   89
            ASSIGN 130 TO KRET                                          KACHYDR   90
            GO TO 250                                                   KACHYDR   91
      230   ASSIGN 290 TO KRET                                          KACHYDR   92
      250   PRINT 4000, NUMTD,T,NCYC                                    KACHYDR   93
            IF (LPR.GT.0) WRITE (12,4000) NUMTD,T,NCYC                  KACHYDR   94
            WRITE (8) (AA(N),N=1,NSC)                                   KACHYDR   95
            IF (LCM.EQ.1) WRITE(8) (AA1(N),N=1,NLC)                     KACHYDR   96
            CALL DATAREL (5LFSET8)                                      KACHYDR   97
            CALL AFSREL (3LOUT)                                         KACHYDR   98
            CALL AFSREL (4LFILM)                                        KACHYDR   99
            NUMTD = NUMTD + 1                                           KACHYDR  100
            GO TO KRET                                                  KACHYDR  101
      270   REWIND 7                                                    KACHYDR  102
            JTD = JBAR                                                  KACHYDR  103
            JNSC = LOCF(ZZ) - LOCF(AA) + 1                              KACHYDR  104
            READ(7) (AA(N),N=1,JNSC)                                    KACHYDR  105
            IF (LCM.EQ.1) READ(7)  (AA1(N),N=1,NLC)                     KACHYDR  106
            IF (JTD.NE.NUMTD) GO TO 280                                 KACHYDR  107
            CALL AFSREL (5LFSET7)                                       KACHYDR  108
            NUMTD = NUMTD + 1                                           KACHYDR  109
            TLIM = TLIM*27.5E-9 - 40. + (1.-TLIMD)*EP10                 KACHYDR  110
            PRINT 4010, JTD                                             KACHYDR  111
            PRINT 4020, NAME,T,NCYC                                     KACHYDR  112
            IF (LPR.EQ.0) GO TO 130                                     KACHYDR  113
            WRITE (12,4010) JTD                                         KACHYDR  114
            WRITE (12,4020) NAME,T,NCYC                                 KACHYDR  115
            GO TO 130                                                   KACHYDR  116
      280   PRINT 4030                                                  KACHYDR  117
            CALL AFSREL (5LFSET7)                                       KACHYDR  118
      290   RETURN                                                      KACHYDR  119
      300   IF (NPTOT.EQ.0) GO TO 400                                   KACHYDR  120
            NFR = 0                                                     KACHYDR  121
      310   CALL ADV (1)                                                KACHYDR  122
            CALL COLOR (0.)                                             KACHYDR  123
            CALL FRAME (IXL,IXR,IYT,IYB)                                KACHYDR  124
            IF (LPR.EQ.0) GO TO 320                                     KACHYDR  125
            CALL LINCNT (59)                                            KACHYDR  126
            WRITE (12,4040)                                             KACHYDR  127
            WRITE (12,4050) JNM,D1,C1,NAME,T,NCYC                       KACHYDR  128
            DO 319 NP=1,NPTOT                                           KACHYDR  129
            IX1 = FIXL + (XP(NP)*DR-XL)*XCONV                           KACHYDR  130
            IY1 = FIYB + (YP(NP)*DZ-YB)*YCONV                           KACHYDR  131
            L = (XP(NP).A.38) + 1                                       KACHYDR  132
      319   CALL PLT (IX1,IY1,ICHARS(L))                                KACHYDR  133
      320   DO 349 NTYPE=1,3                                            KACHYDR  134
            KMAT = KMATS(NTYPE)                                         KACHYDR  135
            ICHAR = ICHARS(NTYPE)                                       KACHYDR  136
            CALL COLOR (COLOUR(NTYPE))                                  KACHYDR  137
            IF (LPR.EQ.0) GO TO 330                                     KACHYDR  138
            CALL ADV (1)                                                KACHYDR  139
            CALL FRAME (IXL,IXR,IYT,IYB)                                KACHYDR  140
            CALL LINCNT (59)                                            KACHYDR  141
```

```
               WRITE (12,IDPP(1,NTYPE))
               WRITE (12,4050) JNM,D1,C1,NAME,T,NCYC                       KACHYDR   142
     330       DO 339 NP=1,NPTOT                                           KACHYDR   143
               L = XP(NP).A.3B                                             KACHYDR   144
               IF (L.NE.KMAT) GO TO 339                                    KACHYDR   145
               IX1 = FIXL + (XP(NP)*DR-XL)*XCONV                           KACHYDR   146
               IY1 = FIYB + (YP(NP)*DZ-YB)*YCONV                           KACHYDR   147
               CALL PLT (IX1,IY1,ICHAR)                                    KACHYDR   148
     339       CONTINUE                                                    KACHYDR   149
     349       CONTINUE                                                    KACHYDR   150
               NFR = NFR + 1                                               KACHYDR   151
               IF (NCYC+LPR.EQ.0 .A. NFR.LT.100) GO TO 310                 KACHYDR   152
               IF (LPR.EQ.0) GO TO KPOUT                                   KACHYDR   153
     400       VELMX(1) = VELMX(2) = 0.                                    KACHYDR   154
               CALL START                                                  KACHYDR   155
               DO 419 J=2,JP1                                              KACHYDR   156
               DO 409 I=2,IP1                                              KACHYDR   157
               VELMX(1) = AMAX1(VELMX(1),ABS(UV(IJ)),ABS(VV(IJ)))          KACHYDR   158
               VELMX(2) = AMAX1(VELMX(2),ABS(UD(IJ)),ABS(VD(IJ)))          KACHYDR   159
     409       IJ = IJ + NQ                                                KACHYDR   160
     419       CALL LOOP                                                   KACHYDR   161
               DO 449 NTYPE=1,2                                            KACHYDR   162
               IF (VELMX(NTYPE).LT.EM10) GO TO 449                         KACHYDR   163
               DROU = DRO2 / VELMX(NTYPE)                                  KACHYDR   164
               L = NTYPE - 1                                               KACHYDR   165
               CALL ADV (1)                                                KACHYDR   166
               CALL FRAME (IXL,IXR,IYT,IYB)                                KACHYDR   167
               CALL LINCNT (59)                                            KACHYDR   168
               WRITE (12,IDVV(1,NTYPE))                                    KACHYDR   169
               WRITE (12,4060) VELMX(NTYPE)                                KACHYDR   170
               WRITE (12,4050) JNM,D1,C1,NAME,T,NCYC                       KACHYDR   171
               CALL START                                                  KACHYDR   172
               DO 439 J=2,JP1                                              KACHYDR   173
               Y1 = (FLOAT(J)-1.5) * DZ                                    KACHYDR   174
               IY1 = FIYB + (Y1-YB)*YCONV                                  KACHYDR   175
               DO 429 I=2,IP1                                              KACHYDR   176
               X1 = RI(I)                                                  KACHYDR   177
               X2 = X1 + (UV(IJ-NQ+L)+UV(IJ+L))*DROU                       KACHYDR   178
               Y2 = Y1 + (VV(IJM  +L)+VV(IJ+L))*DROU                       KACHYDR   179
               IX1 = FIXL + (X1-XL)*XCONV                                  KACHYDR   180
               IX2 = FIXL + (X2-XL)*XCONV                                  KACHYDR   181
               IY2 = FIYB + (Y2-YB)*YCONV                                  KACHYDR   182
               CALL DRV (IX1,IY1,IX2,IY2)                                  KACHYDR   183
               CALL PLT (IX1,IY1,16)                                       KACHYDR   184
               IJM = IJM + NQ                                              KACHYDR   185
     429       IJ = IJ + NQ                                                KACHYDR   186
     439       CALL LOOP                                                   KACHYDR   187
     449       CONTINUE                                                    KACHYDR   188
               IF (IBAR.EQ.1 .O. JBAR.EQ.1) GO TO 800                      KACHYDR   189
               DO 799 L=1,9                                                KACHYDR   190
               LM1 = L - 1                                                 KACHYDR   191
               QMN = EP20                                                  KACHYDR   192
               QMX = -EP20                                                 KACHYDR   193
               CALL START                                                  KACHYDR   194
               DO 619 J=2,JP1                                              KACHYDR   195
               DO 609 I=2,IP1                                              KACHYDR   196
                                                                           KACHYDR   197
```

```
       CQ(IJ) = TH(IJ+LM1)                                          KACHYDR   198
       QMN = AMIN1(QMN,CQ(IJ))                                      KACHYDR   199
       IF (CQ(IJ).LT.EP9) QMX = AMAX1(QMX,CQ(IJ))                   KACHYDR   200
  609  IJ = IJ + NQ                                                 KACHYDR   201
  619  CALL LOOP                                                    KACHYDR   202
       IF (LM1.NE.8) GO TO 640                                      KACHYDR   203
       XX = QMX + .05*(QMX-QMN)                                     KACHYDR   204
       CALL START                                                   KACHYDR   205
       DO 639 J=2,JP1                                               KACHYDR   206
       DO 629 I=2,IP1                                               KACHYDR   207
       IF (CQ(IJ).GE.EP9) CQ(IJ) = QMX = XX                         KACHYDR   208
  629  IJ = IJ + NQ                                                 KACHYDR   209
  639  CALL LOOP                                                    KACHYDR   210
       CALL DONE                                                    KACHYDR   211
  640  XX = QMX - QMN                                               KACHYDR   212
       IF (XX.LT.EM10) GO TO 799                                    KACHYDR   213
       DQ = .1*(XX+EM10)                                            KACHYDR   214
       DO 649 IC=1,11                                               KACHYDR   215
  649  CON(IC) = QMN + FLOAT(IC-1)*DQ                               KACHYDR   216
       IC = 11                                                      KACHYDR   217
       CALL ADV (1)                                                 KACHYDR   218
       CALL FRAME (IXL,IXR,IYT,IYB)                                 KACHYDR   219
       CALL LINCNT (59)                                             KACHYDR   220
       WRITE (12,IDCON(1,L))                                        KACHYDR   221
       WRITE (12,4080) QMN,QMX,CON(1),CON(IC-1),DQ                  KACHYDR   222
       WRITE (12,4050) JNM,D1,C1,NAME,T,NCYC                        KACHYDR   223
       CALL START                                                   KACHYDR   224
       DO 789 J=2,JBAR                                              KACHYDR   225
       Y1(1) = Y1(2) = DZ*(FLOAT(J)-1.5)                            KACHYDR   226
       Y1(3) = Y1(4) = Y1(1) + DZ                                   KACHYDR   227
       DO 779 I=2,IBAR                                              KACHYDR   228
       X1(1) = X1(3) = DR*(FLOAT(I)-1.5)                            KACHYDR   229
       X1(2) = X1(4) = X1(1) + DR                                   KACHYDR   230
       IPJ = IJ + NQ                                                KACHYDR   231
       IPJP = IJP + NQ                                              KACHYDR   232
       DO 769 IK = 1,IC                                             KACHYDR   233
       K1 = K2 = K3 = K4 = 0                                        KACHYDR   234
       IF (CQ(IJ).LE.CON(IK)) K1 = 1                                KACHYDR   235
       IF (CQ(IPJ).LE.CON(IK)) K2 = 1                               KACHYDR   236
       IF (CQ(IJP).LE.CON(IK)) K3 = 1                               KACHYDR   237
       IF (CQ(IPJP).LE.CON(IK)) K4 = 1                              KACHYDR   238
       IF (K1*K2*K3*K4.NE.0 .O. K1+K2+K3+K4.EQ.0) GO TO 769         KACHYDR   239
       LL = 0                                                       KACHYDR   240
       IF (K1+K3.NE.1) GO TO 710                                    KACHYDR   241
       IC1 = 1                                                      KACHYDR   242
       IC2 = 3                                                      KACHYDR   243
       IJ1 = IJ                                                     KACHYDR   244
       IJ2 = IJP                                                    KACHYDR   245
       ASSIGN 710 TO KR1                                            KACHYDR   246
       GO TO 740                                                    KACHYDR   247
  710  IF (K1+K2.NE.1) GO TO 720                                    KACHYDR   248
       IC1 = 1                                                      KACHYDR   249
       IC2 = 2                                                      KACHYDR   250
       IJ1 = IJ                                                     KACHYDR   251
       IJ2 = IPJ                                                    KACHYDR   252
       ASSIGN 720 TO KR1                                            KACHYDR   253
```

```
            GO TO 740                                                   KACHYDR  254
      720   IF (K2+K4.NE.1) GO TO 730                                   KACHYDR  255
            IC1 = 2                                                     KACHYDR  256
            IC2 = 4                                                     KACHYDR  257
            IJ1 = IPJ                                                   KACHYDR  258
            IJ2 = IPJP                                                  KACHYDR  259
            ASSIGN 730 TO KR1                                           KACHYDR  260
            GO TO 740                                                   KACHYDR  261
      730   IF (K3+K4.NE.1) GO TO 769                                   KACHYDR  262
            IC1 = 3                                                     KACHYDR  263
            IC2 = 4                                                     KACHYDR  264
            IJ1 = IJP                                                   KACHYDR  265
            IJ2 = IPJP                                                  KACHYDR  266
            ASSIGN 769 TO KR1                                           KACHYDR  267
      740   LL = LL + 1                                                 KACHYDR  268
            XX = (CON(IK)-CQ(IJ1)) / (CQ(IJ2)-CQ(IJ1))                  KACHYDR  269
            IX1(LL) = FIXL + (X1(IC1)+XX*(X1(IC2)-X1(IC1))-XL)*XCONV    KACHYDR  270
            IY1(LL) = FIYB + (Y1(IC1)+XX*(Y1(IC2)-Y1(IC1))-YB)*YCONV    KACHYDR  271
            IF (LL.LT.2) GO TO KR1                                      KACHYDR  272
            CALL DRV (IX1,IY1,IX2,IY2)                                  KACHYDR  273
            IF (IK.EQ.1) CALL PLT (IX1,IY1,35)                          KACHYDR  274
            IF (IK.EQ.IC-1) CALL PLT (IX1,IY1,24)                       KACHYDR  275
            LL = 0                                                      KACHYDR  276
            IF (IJ2.EQ.IPJ) GO TO 720                                   KACHYDR  277
      769   CONTINUE                                                    KACHYDR  278
            IJ = IPJ                                                    KACHYDR  279
      779   IJP = IPJP                                                  KACHYDR  280
            CALL LOOP                                                   KACHYDR  281
            IJ = IJ + NQL                                               KACHYDR  282
      789   IJP = IJP + NQL                                             KACHYDR  283
      799   CONTINUE                                                    KACHYDR  284
      800   ASSIGN 840 TO KRF                                           KACHYDR  285
            KRP = KRF                                                   KACHYDR  286
            ASSIGN 889 TO KRFP                                          KACHYDR  287
            GO TO (830,820,880) LPR                                     KACHYDR  288
      820   ASSIGN 880 TO KRF                                           KACHYDR  289
            ASSIGN 870 TO KRFP                                          KACHYDR  290
      830   CALL LINCNT (64)                                            KACHYDR  291
            CALL ADV (1)                                                KACHYDR  292
            GO TO 860                                                   KACHYDR  293
      840   KRF = KRFP                                                  KACHYDR  294
            ASSIGN 889 TO KRP                                           KACHYDR  295
            CALL START                                                  KACHYDR  296
            DO 899 J=1,JP2                                              KACHYDR  297
            DO 889 I=1,IP2                                              KACHYDR  298
            L = IJM - NQ                                                KACHYDR  299
            GO TO (850,850,870) LPR                                     KACHYDR  300
      850   WRITE (12,6090) I,J,TH(L),UV(L),VV(L),SIEV(L),ROVPR1(L),ROVPR(L),  KACHYDR  301
           1                DPRT(L),P(L),E(L),UD(L),VD(L),SIED(L),RODPR1(L),   KACHYDR  302
           2                RODPR(L),A(L),KIJ(L)                       KACHYDR  303
            LINESF = LINESF + 3                                         KACHYDR  304
            IF (LINESF.LT.59) GO TO KRF                                 KACHYDR  305
      860   CALL ADV (1)                                                KACHYDR  306
            LINESF = 0                                                  KACHYDR  307
            WRITE (12,4050) JNM,D1,C1,NAME,T,NCYC                       KACHYDR  308
            WRITE (12,6100)                                             KACHYDR  309
```

```
      GO TO KRF                                                         KACHYDR  310
  870 PRINT 4090,        I,J,TH(L),UV(L),VV(L),SIEV(L),ROVPR1(L),ROVPR(L),  KACHYDR  311
     1                   DPRT(L),P(L),E(L),UD(L),VD(L),SIED(L),RODPR1(L),   KACHYDR  312
     2                   RODPR(L),A(L),KIJ(L)                               KACHYDR  313
      LINESP = LINESP + 3                                               KACHYDR  314
      IF (LINESP.LT.54) GO TO KRP                                       KACHYDR  315
  880 PRINT 4110                                                        KACHYDR  316
      LINESP = 0                                                        KACHYDR  317
      PRINT 4050, JNM,D1,C1,NAME,T,NCYC                                 KACHYDR  318
      PRINT 4100                                                        KACHYDR  319
      GO TO KRP                                                         KACHYDR  320
  889 IJM = IJM + NQ                                                    KACHYDR  321
      CALL LOOP                                                         KACHYDR  322
  899 IJM = IJM - NQ2L                                                  KACHYDR  323
  900 DO 909 I=1,20                                                     KACHYDR  324
  909 SPSUMS(I) = 0.                                                    KACHYDR  325
      CALL START                                                        KACHYDR  326
      DO 929 J=2,JP1                                                    KACHYDR  327
      DO 919 I=2,IP1                                                    KACHYDR  328
      IPJ = IJ + NQ                                                     KACHYDR  329
      IMJ = IJ - NQ                                                     KACHYDR  330
          ROVVOL = ROVPR(IJ)*VOL(I)                                     KACHYDR  331
          RODVOL = RODPR(IJ)*VOL(I)                                     KACHYDR  332
      STH = STH + (1.-TH(IJ))*VOL(I)                                    KACHYDR  333
      SMV = SMV + ROVVOL                                                KACHYDR  334
      SMV1 = SMV1 + VOL(I)*ROVPR1(IJ)                                   KACHYDR  335
      SMV2 = SMV2 + VOL(I)*(ROVPR(IJ)-ROVPR1(IJ))                       KACHYDR  336
      SMD = SMD + RODVOL                                                KACHYDR  337
      SMD1 = SMD1 + VOL(I)*RODPR1(IJ)                                   KACHYDR  338
      SMD2 = SMD2 + VOL(I)*(RODPR(IJ)-RODPR1(IJ))                       KACHYDR  339
      SMOMR = SMOMR + .5*VOLR(I)*(UV(IJ)*(ROVPR(IJ)+ROVPR(IPJ))         KACHYDR  340
     1                          +UD(IJ)*(RODPR(IJ)+RODPR(IPJ)))         KACHYDR  341
          ROVTEM = .5*VOL(I)*VV(IJ) * (ROVPR(IJ)+ROVPR(IJP))            KACHYDR  342
          RODTEM = .5*VOL(I)*VD(IJ) * (RODPR(IJ)+RODPR(IJP))            KACHYDR  343
      SMOMZV = SMOMZV + ROVTEM                                          KACHYDR  344
      SMOMZD = SMOMZD + RODTEM                                          KACHYDR  345
      SMOMZ  = SMOMZ  + ROVTEM + RODTEM                                 KACHYDR  346
          STIEV = ROVVOL*SIEV(IJ)                                       KACHYDR  347
          STIED = RODVOL*SIED(IJ)                                       KACHYDR  348
      SINTEV = SINTEV + STIEV                                           KACHYDR  349
      SINTED = SINTED + STIED                                           KACHYDR  350
      SINTE = SINTE + STIEV + STIED                                     KACHYDR  351
          STKEV = ROVVOL*.25*(UV(IJ)**2+UV(IMJ)**2                      KACHYDR  352
     1                       +VV(IJ)**2+VV(IJM)**2)                     KACHYDR  353
          STKED = RODVOL*.25*(UD(IJ)**2+UD(IMJ)**2                      KACHYDR  354
     1                       +VD(IJ)**2+VD(IJM)**2)                     KACHYDR  355
      SKEV = SKEV + STKEV                                               KACHYDR  356
      SKED = SKED + STKED                                               KACHYDR  357
      SKE = SKE + STKEV + STKED                                         KACHYDR  358
          STEV = STIEV + STKEV                                          KACHYDR  359
          STED = STIED + STKED                                          KACHYDR  360
      SEV = SEV + STEV                                                  KACHYDR  361
      SED = SED + STED                                                  KACHYDR  362
      SE = SE + STEV + STED                                             KACHYDR  363
      IJM = IJM + NQ                                                    KACHYDR  364
      IJP = IJP + NQ                                                    KACHYDR  365
```

2

```
 910  IJ = IPJ
 929  CALL LOOP                                                                   KACHYDR   366
      IF (LPR.GT.0) WRITE (12,4130) T,NCYC,(SPSUMS(L),L=1,20)                     KACHYDR   367
      IF (LPR.GE.2) PRINT 4130,     T,NCYC,(SPSUMS(L),L=1,20)                     KACHYDR   368
      CALL EMPTY                                                                  KACHYDR   369
      GO TO KROUT                                                                 KACHYDR   370
1000  CALL START                                                                  KACHYDR   371
      DO 1089 J=2,JP1                                                             KACHYDR   372
      DO 1079 I=2,IP1                                                             KACHYDR   373
      IPJ = IJ + NQ                                                               KACHYDR   374
      IMJ = IJ - NQ                                                               KACHYDR   375
      XIVL = BDTODR*UV(IMJ) + SIGN(A0,UV(IMJ))                                    KACHYDR   376
      HPXIVL = .5 + XIVL                                                          KACHYDR   377
      HMXIVL = .5 - XIVL                                                          KACHYDR   378
      XIVR = BDTODR*UV(IJ)  + SIGN(A0,UV(IJ) )                                    KACHYDR   379
      HPXIVR = .5 + XIVR                                                          KACHYDR   380
      HMXIVR = .5 - XIVR                                                          KACHYDR   381
      XIVB = BDTODZ*VV(IJM) + SIGN(A0,VV(IJM))                                    KACHYDR   382
      HPXIVB = .5 + XIVB                                                          KACHYDR   383
      HMXIVB = .5 - XIVB                                                          KACHYDR   384
      XIVT = BDTODZ*VV(IJ)  + SIGN(A0,VV(IJ) )                                    KACHYDR   385
      HPXIVT = .5 + XIVT                                                          KACHYDR   386
      HMXIVT = .5 - XIVT                                                          KACHYDR   387
      XIDL = BDTODR*UD(IMJ) + SIGN(A0,UD(IMJ))                                    KACHYDR   388
      HPXIDL = .5 + XIDL                                                          KACHYDR   389
      HMXIDL = .5 - XIDL                                                          KACHYDR   390
      XIDR = BDTODR*UD(IJ)  + SIGN(A0,UD(IJ) )                                    KACHYDR   391
      HPXIDR = .5 + XIDR                                                          KACHYDR   392
      HMXIDR = .5 - XIDR                                                          KACHYDR   393
      XIDB = BDTODZ*VD(IJM) + SIGN(A0,VD(IJM))                                    KACHYDR   394
      HPXIDB = .5 + XIDB                                                          KACHYDR   395
      HMXIDB = .5 - XIDB                                                          KACHYDR   396
      XIDT = BDTODZ*VD(IJ)  + SIGN(A0,VD(IJ) )                                    KACHYDR   397
      HPXIDT = .5 + XIDT                                                          KACHYDR   398
      HMXIDT = .5 - XIDT                                                          KACHYDR   399
      UVRR = UV(IJ) *RIP(I)                                                       KACHYDR   400
      UVRL = UV(IMJ)*RIP(I-1)                                                     KACHYDR   401
      UDRR = UD(IJ) *RIP(I)                                                       KACHYDR   402
      UDRL = UD(IMJ)*RIP(I-1)                                                     KACHYDR   403
      THIJ = TH(IJ)                                                               KACHYDR   404
      RR = 1. / ROVPR(IJ)                                                         KACHYDR   405
      RODPRT(IJ) = RODPR(IJ)                                                      KACHYDR   406
     1       +DTODR*RRI(I)*(UDRL*(HPXIDL*RODPR(IMJ)+HMXIDL*RODPR(IJ) )            KACHYDR   407
     2                     -UDRR*(HPXIDR*RODPR(IJ) +HMXIDR*RODPR(IPJ)))           KACHYDR   408
     3          +DTODZ*(VD(IJM)*(HPXIDB*RODPR(IJM)+HMXIDB*RODPR(IJ) )             KACHYDR   409
     4                 -VD(IJ) *(HPXIDT*RODPR(IJ) +HMXIDT*RODPR(IJP)))            KACHYDR   410
      RODPR1T = RODPR1(IJ)                                                        KACHYDR   411
     1     +DTODR*RRI(I)*(UDRL*(HPXIDL*RODPR1(IMJ)+HMXIDL*RODPR1(IJ) )            KACHYDR   412
     2                   -UDRR*(HPXIDR*RODPR1(IJ) +HMXIDR*RODPR1(IPJ)))           KACHYDR   413
     3        +DTODZ*(VD(IJM)*(HPXIDB*RODPR1(IJM)+HMXIDB*RODPR1(IJ) )             KACHYDR   414
     4               -VD(IJ) *(HPXIDT*RODPR1(IJ) +HMXIDT*RODPR1(IJP)))            KACHYDR   415
      THTE = 1. - RODPR1T*RRO1 - (RODPRT(IJ)-RODPR1T)*RRO2                        KACHYDR   416
      IF (THTE.LT.THO .A. NCYC.GT.1) GO TO 1010                                   KACHYDR   417
      ROVPR2 = ROVPR(IJ) - ROVPR1(IJ)                                             KACHYDR   418
      A(IJ) = SIEV(IJ)*(ROVPR1(IJ)*BV1GM11 + ROVPR2*BV2GM12)                      KACHYDR   419
     1         / (THTE*(ROVPR1(IJ)*BV1 + ROVPR2*BV2))                             KACHYDR   420
                                                                                  KACHYDR   421
```

```
          VEL = .25 * (UV(IJ)**2+UV(IMJ)**2+VV(IJ)**2+VV(IJM)**2)          KACHYDR 422
          GAMTE = GGM11*ROVPR1(IJ)*RR + GGM12*(1.-ROVPR1(IJ)*RR)           KACHYDR 423
          GIROM = GAMTE*SIEV(IJ)*ROVPR(IJ)*SQM0                            KACHYDR 424
          THTESQ = THTE * THTE                                             KACHYDR 425
          VEL = VEL + P(IJ)*THTE*RR                                        KACHYDR 426
          RMSQ = (VEL*THTESQ*(ROVPR(IJ)+RODPR(IJ))/GIROM)   **2            KACHYDR 427
          F(IJ) = 1. / (1.+10.*RMSQ)                                       KACHYDR 428
          P(IJ) = F(IJ)*P(IJ) + (1.-F(IJ))*A(IJ)*ROVPR(IJ)                 KACHYDR 429
          VECVEL = .5*QSQRT((UV(IJ)-UD(IJ)+UV(IMJ)-UD(IMJ))**2             KACHYDR 430
         1                      +(VV(IJ)-VD(IJ)+VV(IJM)-VD(IJM))**2)       KACHYDR 431
          THTERM = (1.-THTE) / THTESQ                                      KACHYDR 432
          KIJ(IJ) = RPCOF*ROVPR(IJ)*THTERM*(NUV3+RPCDR*VECVEL)             KACHYDR 433
          GO TO 1020                                                       KACHYDR 434
 1010     THTE = 0.                                                        KACHYDR 435
          A(IJ) = KIJ(IJ) = AKINFI                                         KACHYDR 436
 1020     SIEVC = SIEV(IJ)                                                 KACHYDR 437
          SIEDC = SIED(IJ)                                                 KACHYDR 438
          TVC = SIEVC     *ROVPR(IJ)                                       KACHYDR 439
         1         / (ROVPR1(IJ) *BV1 + (ROVPR(IJ) -ROVPR1(IJ) )*BV2)      KACHYDR 440
          TVL = SIEV(IMJ)*ROVPR(IMJ)                                       KACHYDR 441
         1         / (ROVPR1(IMJ)*BV1 + (ROVPR(IMJ)-ROVPR1(IMJ))*BV2)      KACHYDR 442
          TVR = SIEV(IPJ)*ROVPR(IPJ)                                       KACHYDR 443
         1         / (ROVPR1(IPJ)*BV1 + (ROVPR(IPJ)-ROVPR1(IPJ))*BV2)      KACHYDR 444
          TVB = SIEV(IJM)*ROVPR(IJM)                                       KACHYDR 445
         1         / (ROVPR1(IJM)*BV1 + (ROVPR(IJM)-ROVPR1(IJM))*BV2)      KACHYDR 446
          TVT = SIEV(IJP)*ROVPR(IJP)                                       KACHYDR 447
         1         / (ROVPR1(IJP)*BV1 + (ROVPR(IJP)-ROVPR1(IJP))*BV2)      KACHYDR 448
          TDC = SIEDC     *RODPR(IJ)                                       KACHYDR 449
         1         / (RODPR1(IJ) *BD1 + (RODPR(IJ) -RODPR1(IJ) )*BD2)      KACHYDR 450
          TDL = SIED(IMJ)*RODPR(IMJ)                                       KACHYDR 451
         1         / (RODPR1(IMJ)*BD1 + (RODPR(IMJ)-RODPR1(IMJ))*BD2)      KACHYDR 452
          TDR = SIED(IPJ)*RODPR(IPJ)                                       KACHYDR 453
         1         / (RODPR1(IPJ)*BD1 + (RODPR(IPJ)-RODPR1(IPJ))*BD2)      KACHYDR 454
          TDB = SIED(IJM)*RODPR(IJM)                                       KACHYDR 455
         1         / (RODPR1(IJM)*BD1 + (RODPR(IJM)-RODPR1(IJM))*BD2)      KACHYDR 456
          TDT = SIED(IJP)*RODPR(IJP)                                       KACHYDR 457
         1         / (RODPR1(IJP)*BD1 + (RODPR(IJP)-RODPR1(IJP))*BD2)      KACHYDR 458
          THLB = .5 * (THIJ + TH(IMJ))                                     KACHYDR 459
          OMTHLB = 1. - THLB                                               KACHYDR 460
          THRB = .5 * (THIJ + TH(IPJ))                                     KACHYDR 461
          OMTHRB = 1. - THRB                                               KACHYDR 462
          THBB = .5 * (THIJ + TH(IJM))                                     KACHYDR 463
          OMTHBB = 1. - THBB                                               KACHYDR 464
          THTB = .5 * (THIJ + TH(IJP))                                     KACHYDR 465
          OMTHTB = 1. - THTB                                               KACHYDR 466
          CFXL = UVRL    * (HPXIVL*SIEV(IMJ)+HMXIVL*SIEVC)                 KACHYDR 467
          CFXR = UVRR    * (HPXIVR*SIEVC    +HMXIVR*SIEV(IPJ))             KACHYDR 468
          CFXB = VV(IJM)* (HPXIVB*SIEV(IJM)+HMXIVB*SIEVC)                  KACHYDR 469
          CFXT = VV(IJ)  * (HPXIVT*SIEVC    +HMXIVT*SIEV(IJP))             KACHYDR 470
          SIEDEL = SIEVC*(RRIDR(I)*(RIP(I)*UV(IJ)-RIP(I-1)*UV(IMJ))        KACHYDR 471
         1                 +RDZ*(       VV(IJ)-          VV(IJM)))         KACHYDR 472
          VEL = PTE = 0.                                                   KACHYDR 473
          IF (THTE.LT.TH0) GO TO 1030                                      KACHYDR 474
          VEL = KIJ(IJ) * ((.5*(UD(IJ)+UD(IMJ)-UV(IJ)-UV(IMJ)))**2         KACHYDR 475
         1               +(.5*(VD(IJ)+VD(IJM)-VV(IJ)-VV(IJM)))**2)         KACHYDR 476
          PTE = P(IJ)*RR*(RRIDR(I)*(RIP(I)*(THRB*UV(IJ) +OMTHRB*UD(IJ) )   KACHYDR 477
```

```
     1                          -RIP(I-1)*(THLB*UV(IMJ)+OMTHLB*UD(IMJ)))    KACHYDR 478
     2                             +RDZ*(THTB*VV(IJ) +OMTHTB*VD(IJ)        KACHYDR 479
     3                                -THBB*VV(IJM)-OMTHBB*VD(IJM)))      KACHYDR 480
1030  SIEVTE = SIEVC+DT*(RRIDR(I)*(CFXL-CFXR)+RDZ*(CFXB-CFXT)             KACHYDR 481
     1           +RR*(R*(TDC-TVC)+VEL                                     KACHYDR 482
     2 +RRI(I)*KVODRSQ*(THRB*RIP(I)*(TVR-TVC)-THLB*RIP(I-1)*(TVC-TVL))    KACHYDR 483
     3        +KVODZSQ*(THTB*(TVT-TVC)-THBB*(TVC-TVB))) - PTE + SIEDEL)   KACHYDR 484
      IF (A(IJ).NE.AKINF(I)) A(IJ) = A(IJ)*SIEVTE/SIEVC                   KACHYDR 485
      CFXL = UDRL   * (HPXIDL*SIED(IMJ)+HMXIDL*SIEDC)                     KACHYDR 486
      CFXR = UDRR   * (HPXIDR*SIEDC    +HMXIDR*SIED(IPJ))                 KACHYDR 487
      CFXB = VD(IJM)* (HPXIDB*SIED(IJM)+HMXIDB*SIEDC)                     KACHYDR 488
      CFXT = VD(IJ) * (HPXIDT*SIEDC    +HMXIDT*SIED(IJP))                 KACHYDR 489
      SIEDFL = SIEDC*(RRIDR(I)*(UD(IJ)*RIP(I)-UD(IMJ)*RIP(I-1))           KACHYDR 490
     1                   +RDZ*(VD(IJ)         -VD(IJM)          ))        KACHYDR 491
      SIEDTE = SIEDC+DT*(RRIDR(I)*(CFXL-CFXR)+RDZ*(CFXB-CFXT)             KACHYDR 492
     1           *1./RODPR(IJ) * (R*(TVC-TDC)+E(IJ) + RRI(I)*             KACHYDR 493
     2 KDODRSQ*(OMTHRB*RIP(I)*(TDR-TDC)-OMTHLB*RIP(I-1)*(TDC-TDL))        KACHYDR 494
     3+KDODZSQ*(OMTHTB*(TDT-TDC)-OMTHBB*(TDC-TDB))) + SIEDEL)             KACHYDR 495
      IF (I.GT.2) GO TO 1040                                              KACHYDR 496
      TH(IMJ) = THTE                                                      KACHYDR 497
      RODPR1(IMJ) = RODPR1T                                               KACHYDR 498
      P(IMJ) = P(IJ)                                                      KACHYDR 499
      KIJ(IMJ) = KIJ(IJ)                                                  KACHYDR 500
      SIEV(IMJ) = SIEVTE                                                  KACHYDR 501
      SIED(IMJ) = SIEDTE                                                  KACHYDR 502
1040  IF (J.GT.2) GO TO 1050                                              KACHYDR 503
      TH(IJM) = THTE + BINF*(THIN-THTE)                                   KACHYDR 504
      RODPR1(IJM) = RODPR1T                                               KACHYDR 505
      P(IJM) = P(IJ)                                                      KACHYDR 506
      KIJ(IJM) = KIJ(IJ)                                                  KACHYDR 507
      SIEV(IJM) = SIEVTE + BINF*(SIEVIN-SIEVTE)                           KACHYDR 508
      SIED(IJM) = SIEDTE                                                  KACHYDR 509
      GO TO 1060                                                          KACHYDR 510
1050  TH(IJM) = THTAB(I)                                                  KACHYDR 511
      RODPR1(IJM) = ROD1TAB(I)                                            KACHYDR 512
      SIEV(IJM) = SIEVTAB(I)                                              KACHYDR 513
      SIED(IJM) = SIEDTAB(I)                                              KACHYDR 514
1060  THTAB(I) = THTE                                                     KACHYDR 515
      ROD1TAB(I) = RODPR1T                                                KACHYDR 516
      SIEVTAB(I) = SIEVTE                                                 KACHYDR 517
      SIEDTAB(I) = SIEDTE                                                 KACHYDR 518
      IJ = IPJ                                                            KACHYDR 519
      IJP = IJP + NQ                                                      KACHYDR 520
1079  IJM = IJM + NQ                                                      KACHYDR 521
      TH(IJ) = THTE                                                       KACHYDR 522
      RODPR1(IJ) = RODPR1T                                                KACHYDR 523
      P(IJ) = P(IJ-NQ)                                                    KACHYDR 524
      KIJ(IJ) = KIJ(IJ-NQ)                                                KACHYDR 525
      SIEV(IJ) = SIEVTE                                                   KACHYDR 526
      SIED(IJ) = SIEDTE                                                   KACHYDR 527
1089  CALL LOOP                                                           KACHYDR 528
      DO 1099 I=2,IP1                                                     KACHYDR 529
      TH(IJM) = TH(IJ) = THTAB(I)                                         KACHYDR 530
      RODPR1(IJM) = RODPR1(IJ) = ROD1TAB(I)                               KACHYDR 531
      P(IJ) = P(IJM)                                                      KACHYDR 532
      KIJ(IJ) = KIJ(IJM)                                                  KACHYDR 533
```

```
      SIEV(IJM) = SIEV(IJ) = SIEVTAB(I)                                 KACHYDR 534
      SIED(IJM) = SIED(IJ) = SIEDTAB(I)                                 KACHYDR 535
      IJ = IJ + NQ                                                      KACHYDR 536
 1099 IJM = IJM + NQ                                                    KACHYDR 537
      CALL DONE                                                         KACHYDR 538
      KREQ = 0                                                          KACHYDR 539
 1500 IF (IBAR.EQ.1) GO TO 1550                                         KACHYDR 540
      CALL START                                                        KACHYDR 541
      DO 1549 J=2,JP1                                                   KACHYDR 542
      DO 1539 I=2,IBAR                                                  KACHYDR 543
      IJA = KREQ + IJ                                                   KACHYDR 544
      IJPA = KREQ + IJP                                                 KACHYDR 545
      IJMA = KREQ + IJM                                                 KACHYDR 546
      IPJA = IJA + NQ                                                   KACHYDR 547
      IMJA = IJA - NQ                                                   KACHYDR 548
      IPJPA = IJPA + NQ                                                 KACHYDR 549
      IPJMA = IJMA + NQ                                                 KACHYDR 550
      URB = UV(IJA)                                                     KACHYDR 551
      UC = .5*(URB+UV(IMJA))                                            KACHYDR 552
      UR = .5*(URB+UV(IPJA))                                            KACHYDR 553
      ROULB = .5*(ROVPR(IJA) +ROVPR(IMJA))*UV(IMJA)                     KACHYDR 554
      ROURB = .5*(ROVPR(IJA)+ROVPR(IPJA))*URB                           KACHYDR 555
      ROUR2B = .5*(ROVPR(IPJA)+ROVPR(IPJA+NQ))*UV(IPJA)                 KACHYDR 556
      VTRC = .5*(VV(IJA)+VV(IPJA))                                      KACHYDR 557
      VBRC = .5*(VV(IJMA)+VV(IPJMA))                                    KACHYDR 558
      VRB = .5*(VTRC+VBRC)                                              KACHYDR 559
      ROUJP = .5*(ROVPR(IJPA)+ROVPR(IPJPA))*UV(IJPA)                    KACHYDR 560
      ROUJM = .5*(ROVPR(IJMA)+ROVPR(IPJMA))*UV.IJMA)                    KACHYDR 561
      FR = - (SIGN(EPV(KREQ+1),VRB)*URB*VRB) / (URB**2+VRB**2+EM10)     KACHYDR 562
      XIC = BDTODR*UC + SIGN(A0,UC)                                     KACHYDR 563
      XIR = BDTODR*UR + SIGN(A0,UR)                                     KACHYDR 564
      XIBR = BDTODZ*VBRC + SIGN(A0,VBRC)                                KACHYDR 565
      XITR = BDTODZ*VTRC + SIGN(A0,VTRC)                                KACHYDR 566
      ROUTE(IJ) = ROURB + FR*DT +                                       KACHYDR 567
     1   ARIP(I)*DTODR*(UC*RI(I)*((.5+XIC) *ROULB+(.5-XIC) *ROURB)      KACHYDR 568
     2                -UR*RI(I+1)*((.5+XIR) *ROURB+(.5-XIR) *ROUR2B))   KACHYDR 569
     3               +DTODZ*(VBRC*((.5+XIBR)*ROUJM+(.5-XIBR)*ROURB)     KACHYDR 570
     4                      -VTRC*((.5+XITR)*ROURB+(.5-XITR)*ROUJP))    KACHYDR 571
      IJ = IJ + NQ                                                      KACHYDR 572
      IJP = IJP + NQ                                                    KACHYDR 573
 1539 IJM = IJM + NQ                                                    KACHYDR 574
      ROUTE(IJ) = ROUTE(IJ-NQ)*RCONT(J)                                 KACHYDR 575
      CALL LOOP                                                         KACHYDR 576
      IJ = IJ + NQL                                                     KACHYDR 577
      IJP = IJP + NQL                                                   KACHYDR 578
 1549 IJM = IJM + NQL                                                   KACHYDR 579
      CALL DONE                                                         KACHYDR 580
 1550 CALL START                                                        KACHYDR 581
      DO 1579 J=2,JBAR                                                  KACHYDR 582
      CALL HIJP2                                                        KACHYDR 583
      DO 1569 I=2,IP1                                                   KACHYDR 584
      IJA = KREQ + IJ                                                   KACHYDR 585
      IJPA = KREQ + IJP                                                 KACHYDR 586
      IJMA = KREQ + IJM                                                 KACHYDR 587
      IPJA = IJA + NQ                                                   KACHYDR 588
      IMJA = IJA - NQ                                                   KACHYDR 589
```

```
         IPJPA = IJPA + NO
         IMJPA = IJPA - NO                                                 KACHYDR   590
         VTB = VV(IJA)                                                     KACHYDR   591
         VC = .5*(VTB+VV(IJMA))                                            KACHYDR   592
         VT = .5*(VTB+VV(IJPA))                                            KACHYDR   593
         ROVBB = .5*(ROVPR(IJA)+ROVPR(IJMA))*VV(IJMA)                      KACHYDR   594
         ROTB = .5*(ROVPR(IJA)+ROVPR(IJPA))                                KACHYDR   595
         ROVTB = ROTB * VTB                                                KACHYDR   596
         ROVT2B = .5*(ROVPR(IJPA)+ROVSPL((I-1)*NO+1+KREQ)) * VV(IJPA)      KACHYDR   597
         UTLC = .5*(UV(IMJPA)+UV(IMJA))                                    KACHYDR   598
         UTRC = .5*(UV(IJPA)+UV(IJA))                                      KACHYDR   599
         UTB = .5*(UTLC+UTRC)                                              KACHYDR   600
         ROVIM = .5*(ROVPR(IMJA)+ROVPR(IMJPA))*VV(IMJA)                    KACHYDR   601
         ROVIP = .5*(ROVPR(IPJA)+ROVPR(IPJPA))*VV(IPJA)                    KACHYDR   602
         FZ = + (SIGN(EPV(KREQ+1),VTB)*UTB**2)  / (UTB**2+VTB**2+EM10)     KACHYDR   603
         XIC = BDTODZ*VC + SIGN(A0,VC)                                     KACHYDR   604
         XIT = BDTODZ*VT + SIGN(A0,VT)                                     KACHYDR   605
         XITL = BDTODR*UTLC + SIGN(A0,UTLC)                                KACHYDR   606
         XITR = BDTODR*UTRC + SIGN(A0,UTRC)                                KACHYDR   607
         ROVTE(IJ) = ROVTB + ROTB*GDT + FZ*DT +                            KACHYDR   608
        1   RRI(I)*DTODR*(UTLC*RIP(I-1)*((.5+XITL)*ROVIM+(.5-XITL)*ROVTB)  KACHYDR   609
        2              -UTRC*RIP(I)  *((.5+XITR)*ROVTB+(.5-XITR)*ROVIP))   KACHYDR   610
        3                 +DTODZ*(VC*((.5+XIC) *ROVBB+(.5-XIC) *ROVTB)     KACHYDR   611
        4                       -VT*((.5+XIT) *ROVTB+(.5-XIT) *ROVT2B))    KACHYDR   612
         IF (J.GT.2) GO TO 1560                                            KACHYDR   613
         ROVB = .5*(ROVPR(IJA)+ROVPIN) * VV(IJMA)                          KACHYDR   614
         ROVTE(IJM) = ROVB + ROUT*(ROVTE(IJ)-ROVA)                         KACHYDR   615
    1560 IF (J.EQ.JBAR) ROVTE(IJP) = ROVTE(IJ)*TOP                         KACHYDR   616
         IJ = IJ + NO                                                      KACHYDR   617
         IJP = IJP + NO                                                    KACHYDR   618
    1569 IJM = IJM + NO                                                    KACHYDR   619
    1579 CALL LOOP                                                         KACHYDR   620
         CALL DONE                                                         KACHYDR   621
         CALL START                                                        KACHYDR   622
         DO 1619 J=2,JP1                                                   KACHYDR   623
         DO 1609 I=2,IP1                                                   KACHYDR   624
         IF (IBAR.GT.1) ROVV(IJ,KREQ) = ROVTE(IJ)                          KACHYDR   625
         ROVV(IJM,KREQ) = ROVTE(IJM)                                       KACHYDR   626
         IJ = IJ + NO                                                      KACHYDR   627
    1609 IJM = IJM + NO                                                    KACHYDR   628
    1619 CALL LOOP                                                         KACHYDR   629
         DO 1629 I=2,IP1                                                   KACHYDR   630
         ROVV(IJM,KREQ) = ROVTE(IJM)                                       KACHYDR   631
    1629 IJM = IJM + NO                                                    KACHYDR   632
         CALL DONE                                                         KACHYDR   633
         KREQ = KREQ + 1                                                   KACHYDR   634
         GO TO (1500,1700) KREQ                                            KACHYDR   635
    1700 CALL START                                                        KACHYDR   636
         DO 1769 J=2,JP1                                                   KACHYDR   637
         DO 1759 I=2,IP1                                                   KACHYDR   638
         IPJ = IJ + NO                                                     KACHYDR   639
         IMJ = IJ - NO                                                     KACHYDR   640
         RODPR(IJ) = RODPRT(IJ)                                            KACHYDR   641
         DTKR(IJ) = DTO2*(KIJ(IJ)+KIJ(IPJ))                                KACHYDR   642
         DTKT(IJ) = DTO2*(KIJ(IJ)+KIJ(IJP))                                KACHYDR   643
         RA(IJ) = 1. / A(IJ)                                               KACHYDR   644
                                                                           KACHYDR   645
```

```
      ROVPRT(IJ) = F(IJ)*TH(IJ)*ROVO + (1.-F(IJ))*P(IJ)*RA(IJ)           KACHYDR 646
      IF (I.GT.2) GO TO 1710                                              KACHYDR 647
      RODPR(IMJ) = RODPR(IJ)                                              KACHYDR 648
      DTKR(IMJ) = DTOZ*(KIJ(IJ)+KIJ(IMJ))                                 KACHYDR 649
      ROVPRT(IMJ) = ROVPR(IMJ)                                            KACHYDR 650
 1710 IF (J.GT.2) GO TO 1720                                              KACHYDR 651
      RODPR(IJM) = RODPR(IJ)                                              KACHYDR 652
      DTKT(IJM) = DTOZ*(KIJ(IJ)+KIJ(IJM))                                 KACHYDR 653
      ROVPRT(IJM) = ROVPR(IJM)                                            KACHYDR 654
 1720 IF (I.LT.IP1) GO TO 1730                                            KACHYDR 655
      RODPR(IPJ) = RODPR(IJ)                                              KACHYDR 656
      ROVPRT(IPJ) = ROVPR(IPJ)                                            KACHYDR 657
 1730 IF (J.LT.JP1) GO TO 1740                                            KACHYDR 658
      RODPR(IJP) = RODPR(IJ)                                              KACHYDR 659
      ROVPRT(IJP) = ROVPR(IJP)                                            KACHYDR 660
 1740 IJ = IPJ                                                            KACHYDR 661
      IJP = IJP + NQ                                                      KACHYDR 662
 1759 IJM = IJM + NQ                                                      KACHYDR 663
 1769 CALL LOOP                                                           KACHYDR 664
      CALL DONE                                                           KACHYDR 665
      CALL START                                                          KACHYDR 666
      DO 2099 J=2,JP1                                                     KACHYDR 667
      DO 2089 I=2,IP1                                                     KACHYDR 668
      IPJ = IJ + NQ                                                       KACHYDR 669
      XX = DTODR*(P(IJ)-P(IPJ))                                           KACHYDR 670
      THTE = .5*(TH(IJ)+TH(IPJ))                                          KACHYDR 671
      BS = ROUD(IJ) + (1.-THTE)*XX                                        KACHYDR 672
      CS = ROUV(IJ) + THTE*XX                                             KACHYDR 673
      ROVTEM = .5*(ROVPR(IJ)+ROVPR(IPJ))                                  KACHYDR 674
      RODTEM = .5*(RODPR(IJ)+RODPR(IPJ))                                  KACHYDR 675
      DTK = DTKR(IJ)                                                      KACHYDR 676
      DTKBC = DTK*(BS+CS)                                                 KACHYDR 677
      RDENOM = 1. / (ROVTEM*(RODTEM+DTK)+DTK*RODTEM)                      KACHYDR 678
      UV(IJ) = RDENOM*(RODTEM*CS+DTKBC)                                   KACHYDR 679
      UD(IJ) = RDENOM*(ROVTEM*BS+DTKBC)                                   KACHYDR 680
 2089 IJ = IPJ                                                            KACHYDR 681
 2099 CALL LOOP                                                           KACHYDR 682
      CALL DONE                                                           KACHYDR 683
      CALL START                                                          KACHYDR 684
      KREQ = 1                                                            KACHYDR 685
      DO 2199 J=2,JTOP                                                    KACHYDR 686
      DO 2189 I=2,IP1                                                     KACHYDR 687
      XX = DTODZ*(P(IJM)-P(IJ))                                           KACHYDR 688
      THTE = .5*(TH(IJM)+TH(IJ))                                          KACHYDR 689
      DS = ROVD(IJM) + (1.-THTE)*XX                                       KACHYDR 690
      ES = ROVV(IJM) + THTE*XX                                            KACHYDR 691
      ROVTEM = .5*(ROVPR(IJM)+ROVPR(IJ))                                  KACHYDR 692
      RODTEM = .5*(RODPR(IJM)+RODPR(IJ))                                  KACHYDR 693
      DTK = DTKT(IJM)                                                     KACHYDR 694
      DTKDE = DTK*(DS+ES)                                                 KACHYDR 695
      RDENOM = 1. / (ROVTEM*(RODTEM+DTK)+DTK*RODTEM)                      KACHYDR 696
      VV(IJM) = RDENOM*(RODTEM*ES+DTKDE)                                  KACHYDR 697
      VD(IJM) = RDENOM*(ROVTEM*DS+DTKDE)                                  KACHYDR 698
      GO TO (2130,2140) KREQ                                              KACHYDR 699
 2130 VV(IJM) = BOUT*VV(IJM) + VVIN                                       KACHYDR 700
      VD(IJM) = BOUT*VD(IJM)                                              KACHYDR 701
```

```
2140  IJ = IJ + NQ                                                   KACHYDR  702
2189  IJM = IJM + NQ                                                 KACHYDR  703
      KREQ = 2                                                       KACHYDR  704
2199  CALL LOOP                                                      KACHYDR  705
      CALL DONE                                                      KACHYDR  706
      CONV = 0.                                                      KACHYDR  707
      CALL START                                                     KACHYDR  708
      DO 2299 J=2,JP1                                                KACHYDR  709
      DO 2289 I=2,IP1                                                KACHYDR  710
      IMJ = IJ - NQ                                                  KACHYDR  711
      IPJ = IJ + NQ                                                  KACHYDR  712
      R2A = .5*RA(IJ)                                                KACHYDR  713
      UVR = UV(IJ)                                                   KACHYDR  714
      UVL = UV(IMJ)                                                  KACHYDR  715
      VVT = VV(IJ)                                                   KACHYDR  716
      VVB = VV(IJM)                                                  KACHYDR  717
      IF (I.EQ.2) GO TO 2210                                         KACHYDR  718
      THTEL = THTER                                                  KACHYDR  719
      RODL = RODR                                                    KACHYDR  720
      PUDENL = PUDENR                                                KACHYDR  721
      TERM1L = TERM1R                                                KACHYDR  722
      TERM2L = TERM2R                                                KACHYDR  723
      GO TO 2220                                                     KACHYDR  724
2210  DTK = DTKR(IMJ)                                                KACHYDR  725
      THTEL = .5*(TH(IJ)+TH(IMJ))                                    KACHYDR  726
      ROVL = .5*(ROVPR(IJ) + ROVPR(IMJ))                             KACHYDR  727
      RODL = .5*(RODPR(IJ) + RODPR(IMJ))                             KACHYDR  728
      PUDENL = 1. / (ROVL*(RODL+DTK)+DTK*RODL)                       KACHYDR  729
      TERM1L = ROVL*(1.-THTEL)*DTODR                                 KACHYDR  730
      TERM2L = THTEL*DTODR                                           KACHYDR  731
2220  IF (J.EQ.2) GO TO 2230                                         KACHYDR  732
      THTEB = THTET(I)                                               KACHYDR  733
      RODB = RODT(I)                                                 KACHYDR  734
      PVDENB = PVDENT(I)                                             KACHYDR  735
      TERM1B = TERM1T(I)                                             KACHYDR  736
      TERM2B = TERM2T(I)                                             KACHYDR  737
      GO TO 2240                                                     KACHYDR  738
2230  DTK = DTKT(IJM)                                                KACHYDR  739
      THTEB = .5*(TH(IJ)+TH(IJM))                                    KACHYDR  740
      ROVB = .5*(ROVPR(IJ) + ROVPR(IJM))                             KACHYDR  741
      RODB = .5*(RODPR(IJ) + RODPR(IJM))                             KACHYDR  742
      PVDENB = 1. / (ROVB*(RODB+DTK)+DTK*RODB)                       KACHYDR  743
      TERM1B = ROVB*(1.-THTEB)*DTODZ                                 KACHYDR  744
      TERM2B = THTEB*DTODZ                                           KACHYDR  745
2240  DTK = DTKR(IJ)                                                 KACHYDR  746
      THTER = .5*(TH(IJ)+TH(IPJ))                                    KACHYDR  747
      ROVR = .5*(ROVPR(IJ)+ROVPR(IPJ))                               KACHYDR  748
      RODR = .5*(RODPR(IJ)+RODPR(IPJ))                               KACHYDR  749
      PUDENR = 1. / (ROVR*(RODR+DTK)+DTK*RODR)                       KACHYDR  750
      TERM1R = ROVR*(1.-THTER)*DTODR                                 KACHYDR  751
      TERM2R = THTER*DTODR                                           KACHYDR  752
      DTK = DTKT(IJ)                                                 KACHYDR  753
      THTET(I) = .5*(TH(IJ)+TH(IJP))                                 KACHYDR  754
      ROVT = .5*(ROVPR(IJ)+ROVPR(IJP))                               KACHYDR  755
      RODT(I) = .5*(RODPR(IJ)+RODPR(IJP))                            KACHYDR  756
      PVDENT(I) = 1. / (ROVT*(RODT(I)+DTK)+DTK*RODT(I))              KACHYDR  757
```

```
      TERM1T(I) = ROVT*(1.-THTET(I))*DTODZ                              KACHYDR  758
      TERM2T(I) = THTET(I)*DTODZ                                        KACHYDR  759
      IF (TH(IJ).LT.THO) GO TO 2250                                     KACHYDR  760
      PARUL = - (TERM1L   -RODL   *(TERM2L   +UVL*R2A))*PUDENL          KACHYDR  761
      PARVB = - (TERM1B   -RODB   *(TERM2B   +VVB*R2A))*PVDENB          KACHYDR  762
      PARUR = + (TERM1R   -RODR   *(TERM2R   -UVR*R2A))*PUDENR          KACHYDR  763
      PARVT = + (TERM1T(I)-RODT(I)*(TERM2T(I)-VVT*R2A))*PVDENT(I)       KACHYDR  764
      RBETA = RA(IJ)*RDT+RDR*RRI(I)*(DTODR*(RIP(I)*THTER+RIP(I-1)*THTEL) KACHYDR  765
     1      +RIP(I)*DTKR(IJ)*PARUR-RIP(I-1)*DTKR(IMJ)*PARUL)            KACHYDR  766
     2      +RDZ*(DTODZ*(THTET(I)+THTEB)+DTKT(IJ)*PARVT-DTKT(IJM)*PARVB) KACHYDR  767
      OMBETA(IJ) = OMP / RBETA                                          KACHYDR  768
      GO TO 2270                                                        KACHYDR  769
 2250 OMBETA(IJ) = OMBSPL * RODPR(IJ)                                   KACHYDR  770
 2270 CONV = AMAX1(CONV,(R2DR*(ABS(UVR)+ABS(UVL))                       KACHYDR  771
     1                         +R2DZ*(ABS(VVT)+ABS(VVB)))*ROVPR(IJ))    KACHYDR  772
      IJ = IPJ                                                          KACHYDR  773
      IJP = IJP + NQ                                                    KACHYDR  774
 2289 IJM = IJM + NQ                                                    KACHYDR  775
 2299 CALL LOOP                                                         KACHYDR  776
      CALL DONE                                                         KACHYDR  777
      CONV = EPS*(CONV*EM6)                                             KACHYDR  778
      NUMIT = 0                                                         KACHYDR  779
      MUSTIT = 1                                                        KACHYDR  780
 2400 CALL START                                                        KACHYDR  781
      DO 2499 J=2,JP1                                                   KACHYDR  782
      DO 2489 I=2,IP1                                                   KACHYDR  783
      IMJ = IJ - NQ                                                     KACHYDR  784
      IPJ = IJ + NQ                                                     KACHYDR  785
      IF (TH(IJ).LT.THO) GO TO 2450                                     KACHYDR  786
      UVRB = UV(IJ)                                                     KACHYDR  787
      UVLB = UV(IMJ)                                                    KACHYDR  788
      VVTB = VV(IJ)                                                     KACHYDR  789
      VVBB = VV(IJM)                                                    KACHYDR  790
      XIVR = BDTODR*UVRB + SIGN(A0,UVRB)                                KACHYDR  791
      XIVL = BDTODR*UVLB + SIGN(A0,UVLB)                                KACHYDR  792
      XIVT = BDTODZ*VVTB + SIGN(A0,VVTB)                                KACHYDR  793
      XIVB = BDTODZ*VVBB + SIGN(A0,VVBB)                                KACHYDR  794
      ROVPRTC = ROVPRT(IJ)                                              KACHYDR  795
      DTIL = RDT*(ROVPRTC-ROVPR(IJ)) + RRIDR(I)                         KACHYDR  796
     1   *(UVRB*RIP(I)  *((.5+XIVR)*ROVPRTC    +(.5-XIVR)*ROVPRT(IPJ))  KACHYDR  797
     2    -UVLB*RIP(I-1)*((.5+XIVL)*ROVPRT(IMJ)+(.5-XIVL)*ROVPRTC    )) KACHYDR  798
     3      + RDZ *(VVTB*((.5+XIVT)*ROVPRTC    +(.5-XIVT)*ROVPRT(IJP))  KACHYDR  799
     4             -VVBB*((.5+XIVB)*ROVPRT(IJM)+(.5-XIVB)*ROVPRTC    )) KACHYDR  800
      GO TO 2470                                                        KACHYDR  801
 2450 DTIL = RRIDR(I)*(UD(IJ)*RIP(I)-UD(IMJ)*RIP(I-1))                  KACHYDR  802
     1          +RDZ*(VD(IJ)        -VD(IJM)        )                   KACHYDR  803
 2470 IF (ABS(DTIL).GT.CONV) MUSTIT = MUSTIT + 1                        KACHYDR  804
      PP(IJ) = P(IJ) + DTIL*OMBETA(IJ)                                  KACHYDR  805
      IJ = IPJ                                                          KACHYDR  806
      IJM = IJM + NQ                                                    KACHYDR  807
 2489 IJP = IJP + NQ                                                    KACHYDR  808
 2499 CALL LOOP                                                         KACHYDR  809
      CALL DONE                                                         KACHYDR  810
      CALL START                                                        KACHYDR  811
      DO 2599 J=2,JP1                                                   KACHYDR  812
      DO 2589 I=2,IP1                                                   KACHYDR  813
```

```
      IPJ = IJ + NQ                                                     KACHYDR  814
      IMJ = IJ - NQ                                                     KACHYDR  815
      RODPRC = RODPR(IJ)                                                KACHYDR  816
      THTEC = TH(IJ)                                                    KACHYDR  817
      PC = P(IJ)                                                        KACHYDR  818
      PPC = P(IJ) = PP(IJ)                                              KACHYDR  819
      DPRT(IJ) = DPC = PPC - PC                                         KACHYDR  820
      ROVPRTO = ROVPRT(IJ)                                              KACHYDR  821
      ROVPRTN = ROVPRT(IJ) = F(IJ)*TH(IJ)*ROVO + (1.-F(IJ))*PPC*RA(IJ)  KACHYDR  822
      IF (I.GT.2) GO TO 2520                                            KACHYDR  823
      P(IMJ) = PPC                                                      KACHYDR  824
      ROVPRT(IMJ) = ROVPRTN                                             KACHYDR  825
 2520 IF (I.EQ.IP1) GO TO 2530                                          KACHYDR  826
      DPR = PP(IPJ) - P(IPJ)                                            KACHYDR  827
      DPDTODR = (DPC-DPR)*DTODR                                         KACHYDR  828
      THRB = .5*(THTEC+TH(IPJ))                                         KACHYDR  829
      UVTE = UV(IJ)                                                     KACHYDR  830
      UUTE = UD(IJ)                                                     KACHYDR  831
      DTK = DTKR(IJ)                                                    KACHYDR  832
      DTKDU = DTK*(UVTE-UDTE)                                           KACHYDR  833
      ROVRBO = .5*(ROVPRTO+ROVPRT(IPJ))                                 KACHYDR  834
      ROVRBN = .5*(ROVPRTN+ROVPRT(IPJ)+DPR*RA(IPJ))                     KACHYDR  835
      RODRB  = .5*(RODPRC +RODPR (IPJ))                                 KACHYDR  836
      RS = ROVRBO*UVTE + DTKDU + DPDTODR*THRB                           KACHYDR  837
      US = RODRB *UDTE - DTKDU + DPDTODR*(1.-THRB)                      KACHYDR  838
      RDENOM = 1. / (ROVRBN*(RODRB+DTK)+DTK*RODRB)                      KACHYDR  839
      UV(IJ) = RDENOM*(RS*(RODRB +DTK)+DTK*US)                          KACHYDR  840
      UD(IJ) = RDENOM*(US*(ROVRBN+DTK)+DTK*RS)                          KACHYDR  841
      GO TO 2540                                                        KACHYDR  842
 2530 P(IPJ) = PPC                                                      KACHYDR  843
      ROVPRT(IPJ) = ROVPRTN                                             KACHYDR  844
      UV(IJ) = UV(IMJ) * RCONT(J)                                       KACHYDR  845
      UD(IJ) = UD(IMJ) * RCONT(J)                                       KACHYDR  846
 2540 IF (J.EQ.JP1) GO TO 2560                                          KACHYDR  847
      DPT = PP(IJP) - P(IJP)                                            KACHYDR  848
      DPDTODZ = (DPC-DPT)*DTODZ                                         KACHYDR  849
      THTB = .5*(THTEC+TH(IJP))                                         KACHYDR  850
      VVTE = VV(IJ)                                                     KACHYDR  851
      VDTE = VD(IJ)                                                     KACHYDR  852
      DTK = DTKT(IJ)                                                    KACHYDR  853
      DTKDV = DTK*(VVTE-VDTE)                                           KACHYDR  854
      ROVTBO = .5*(ROVPRTO+ROVPRT(IJP))                                 KACHYDR  855
      ROVTBN = .5*(ROVPRTN+ROVPRT(IJP)+DPT*RA(IJP))                     KACHYDR  856
      RODTB  = .5*(RODPRC +RODPR (IJP))                                 KACHYDR  857
      SS = ROVTBO*VVTE + DTKDV + DPDTODZ*THTB                           KACHYDR  858
      VS = RODTB *VDTE - DTKDV + DPDTODZ*(1.-THTB)                      KACHYDR  859
      RDENOM = 1. / (ROVTBN*(RODTB+DTK)+DTK*RODTB)                      KACHYDR  860
      VV(IJ) = RDENOM*(SS*(RODTB +DTK)+DTK*VS)                          KACHYDR  861
      VD(IJ) = RDENOM*(VS*(ROVTBN+DTK)+DTK*SS)                          KACHYDR  862
      IF (J.GT.2) GO TO 2580                                            KACHYDR  863
      P(IJM) = PPC                                                      KACHYDR  864
      ROVPRT(IJM) = ROVPRTN + BINF*(ROVPIN-ROVPRTN)                     KACHYDR  865
      VV(IJM) = BOUT*VV(IJ) + VVIN                                      KACHYDR  866
      VD(IJM) = BOUT*VD(IJ)                                             KACHYDR  867
      GO TO 2580                                                        KACHYDR  868
 2560 P(IJP) = PPC                                                      KACHYDR  869
```

```
            ROVPRT(IJP) = ROVPRTN                                   KACHYDR  870
            VV(IJ) = VV(IJM)*TOP                                    KACHYDR  871
            VD(IJ) = VD(IJM)*TOP                                    KACHYDR  872
      2580  IJ = IPJ                                                KACHYDR  873
            IJM = IJM + NQ                                          KACHYDR  874
      2589  IJP = IJP + NQ                                          KACHYDR  875
      2599  CALL LOOP                                               KACHYDR  876
            CALL DONE                                               KACHYDR  877
            NUMIT = NUMIT + 1                                       KACHYDR  878
            MUSTPR = MUSTIT                                         KACHYDR  879
            IF (MUSTIT.EQ.0) GO TO 2600                             KACHYDR  880
            MUSTIT = 0                                              KACHYDR  881
            IF (NUMIT.LT.100) GO TO 2400                            KACHYDR  882
      2600  CALL START                                              KACHYDR  883
            DO 2699 J=2,JP1                                         KACHYDR  884
            DO 2689 I=2,IP1                                         KACHYDR  885
            IMJ = IJ - NQ                                           KACHYDR  886
            IPJ = IJ + NQ                                           KACHYDR  887
            DPRT(IJ) = DPRT(IJ) / OMBETA(IJ)                        KACHYDR  888
            ROVPR(IJ) = ROVPRT(IJ)                                  KACHYDR  889
            GO TO (2610,2620,2630) NVAP                             KACHYDR  890
      2610  ROVPRIT(IJ) = 0.                                        KACHYDR  891
            GO TO 2640                                              KACHYDR  892
      2620  ROVPRIT(IJ) = ROVPR(IJ)                                 KACHYDR  893
            GO TO 2640                                              KACHYDR  894
      2630  ROVPRIT(IJ) = ROVPRI(IJ)                                KACHYDR  895
            OMBETA(IJ) = OMRO /                                     KACHYDR  896
           1                    (RDI*R2DR*RRI(I)*(RIP(I)*UV(IJ)-RIP(I-1)*UV(IMJ))    KACHYDR  897
           2                        *R2DZ*(RIP(I)*VV(IJ)-RIP(I-1)*VV(IJM)))          KACHYDR  898
      2640  IF (I.GT.2)   GO TO 2650                                KACHYDR  899
            ROVPR(IMJ) = ROVPR(IJ)                                  KACHYDR  900
            ROVPRIT(IMJ) = ROVPRIT(IJ)                              KACHYDR  901
      2650  IF (J.GT.2)   GO TO 2660                                KACHYDR  902
            ROVPR(IJM) = ROVPR(IJ)+BINF*(ROVPIN-ROVPR(IJ))          KACHYDR  903
            ROVPRIT(IJM) = ROVPRIT(IJ) + BINF*(ROVPIN1-ROVPRIT(IJ)) KACHYDR  904
      2660  IF (I.LT.IP1) GO TO 2670                                KACHYDR  905
            ROVPR(IPJ) = ROVPR(IJ)                                  KACHYDR  906
            ROVPRIT(IPJ) = ROVPRIT(IJ)                              KACHYDR  907
      2670  IF (J.LT.JP1) GO TO 2680                                KACHYDR  908
            ROVPR(IJP) = ROVPR(IJ)                                  KACHYDR  909
            ROVPRIT(IJP) = ROVPRIT(IJ)                              KACHYDR  910
      2680  IJ = IPJ                                                KACHYDR  911
            IJP = IJP + NQ                                          KACHYDR  912
      2689  IJM = IJM + NQ                                          KACHYDR  913
      2699  CALL LOOP                                               KACHYDR  914
            CALL DONE                                               KACHYDR  915
            NUMRO = MUSTRO = 0                                      KACHYDR  916
            IF (NVAP.LT.3) GO TO 3000                               KACHYDR  917
            MUSTIT = 1                                              KACHYDR  918
      2700  CALL START                                              KACHYDR  919
            DO 2799 J=2,JP1                                         KACHYDR  920
            DO 2789 I=2,IP1                                         KACHYDR  921
            IMJ = IJ - NQ                                           KACHYDR  922
            IPJ = IJ + NQ                                           KACHYDR  923
            UVRB = UV(IJ)                                           KACHYDR  924
            UVLB = UV(IMJ)                                          KACHYDR  925
```

```
            VVIB = VV(IJ)                                                      KACHYDR   926
            VVBB = VV(IJM)                                                     KACHYDR   927
            XIVR = BDTODR*UVRB + SIGN(A0,UVRB)                                 KACHYDR   928
            XIVL = BDTODR*UVLB + SIGN(A0,UVLB)                                 KACHYDR   929
            XIVT = BDTODZ*VVTB + SIGN(A0,VVTB)                                 KACHYDR   930
            XIVB = BDTODZ*VVBB + SIGN(A0,VVBB)                                 KACHYDR   931
            ROVPRTO = ROVPRIT(IJ)                                              KACHYDR   932
            Q = RDT*(ROVPRTO-ROVPRI(IJ)) + RRIDR(I)                            KACHYDR   933
           1  *(UVRB*RIP(I)   *((.5+XIVR)*ROVPRTO      +(.5-XIVR)*ROVPRIT(IPJ))) KACHYDR   934
           2   -UVLB*RIP(I-1)*((.5+XIVL)*ROVPRIT(IMJ)+(.5-XIVL)*ROVPRTO      )) KACHYDR   935
           3     + RDZ *(VVTB*((.5+XIVT)*ROVPRTO      +(.5-XIVT)*ROVPRIT(IJP)) KACHYDR   936
           4             -VVBB*((.5+XIVB)*ROVPRIT(IJM)+(.5-XIVB)*ROVPRTO      )) KACHYDR   937
            IF (ABS(Q).GT.CONV) MUSTIT = MUSTIT + 1                            KACHYDR   938
            ROVPRTN = ROVPRIT(IJ) = ROVPRTO - OMBETA(IJ)*Q                     KACHYDR   939
            IF (I.EQ.2)   ROVPRIT(IMJ) = ROVPRTN                               KACHYDR   940
            IF (J.EQ.2)   ROVPRIT(IJM) = ROVPRTN + BINF*(ROVPIN1-ROVPRTN)      KACHYDR   941
            IF (I.EQ.IP1) ROVPRIT(IPJ) = ROVPRTN                               KACHYDR   942
            IF (J.EQ.JP1) ROVPRIT(IJP) = ROVPRTN                               KACHYDR   943
            IJ = IPJ                                                           KACHYDR   944
            IJM = IJM + NQ                                                     KACHYDR   945
     2789   IJP = IJP + NQ                                                     KACHYDR   946
     2799   CALL LOOP                                                          KACHYDR   947
            CALL DONE                                                          KACHYDR   948
            NUMRO = NUMRO + 1                                                  KACHYDR   949
            MUSTRO = MUSTIT                                                    KACHYDR   950
            IF (MUSTIT.EQ.0) GO TO 3000                                        KACHYDR   951
            MUSTIT = 0                                                         KACHYDR   952
            IF (NUMRO.LT.100) GO TO 2700                                       KACHYDR   953
     3000   CALL START                                                         KACHYDR   954
            DO 3099 J=2,JP1                                                    KACHYDR   955
            DTKRSV = DTKR(IJ-NQ)                                               KACHYDR   956
            E(IJ-NQ) = 0.                                                      KACHYDR   957
            DO 3089 I=2,IP1                                                    KACHYDR   958
            IPJ = IJ + NQ                                                      KACHYDR   959
            IMJ = IJ - NQ                                                      KACHYDR   960
            KIJ(IJ) = 2.*RDT*DTKRSV - KIJ(IMJ)                                 KACHYDR   961
            DTKRSV = DTKR(IJ)                                                  KACHYDR   962
            A(IJ) = 1. / RA(IJ)                                                KACHYDR   963
            E(IJ) = 0.                                                         KACHYDR   964
            ROVPRI(IJ) = ROVPRIT(IJ)                                           KACHYDR   965
            IF (I.EQ.2)   ROVPRI(IMJ) = ROVPRI(IJ)                             KACHYDR   966
            IF (J.EQ.2)   ROVPRI(IJM) = ROVPRI(IJ) + BINF*(ROVPIN1-ROVPRI(IJ)) KACHYDR   967
            IF (I.EQ.IP1) ROVPRI(IPJ) = ROVPRI(IJ)                             KACHYDR   968
            IF (J.EQ.JP1) ROVPRI(IJP) = ROVPRI(IJ)                             KACHYDR   969
            IJ = IPJ                                                           KACHYDR   970
            IJP = IJP + NQ                                                     KACHYDR   971
     3089   IJM = IJM + NQ                                                     KACHYDR   972
     3099   CALL LOOP                                                          KACHYDR   973
            CALL DONE                                                          KACHYDR   974
            IF (NPTOT.EQ.0) GO TO 100                                          KACHYDR   975
            CALL START                                                         KACHYDR   976
            DO 3199 J=2,JP2                                                    KACHYDR   977
            DO 3189 I=2,IP2                                                    KACHYDR   978
            IF (I.GT.2) GO TO 3110                                             KACHYDR   979
            VV(IJM-NQ) = VV(IJM)                                               KACHYDR   980
            VD(IJM-NQ) = VD(IJM)                                               KACHYDR   981
```

```
3110  IF (J.GT.2) GO TO 3120                                       KACHYDR   982
      UV(IJM-NQ) = UV(IJ-NQ)                                       KACHYDR   983
      UD(IJM-NQ) = UD(IJ-NQ)                                       KACHYDR   984
3120  IF (I.LT.IP2) GO TO 3130                                     KACHYDR   985
      VV(IJM) = VV(IJM-NQ)                                         KACHYDR   986
      VD(IJM) = VD(IJM-NQ)                                         KACHYDR   987
3130  IF (J.LT.JP2) GO TO 3140                                     KACHYDR   988
      UV(IJ-NQ) = UV(IJM-NQ)                                       KACHYDR   989
      UD(IJ-NQ) = UD(IJM-NQ)                                       KACHYDR   990
3140  IJM = IJM + NQ                                               KACHYDR   991
3189  IJ = IJ + NQ                                                 KACHYDR   992
      CALL LOOP                                                    KACHYDR   993
      IJM = IJM - NQL                                              KACHYDR   994
3199  IJ = IJ - NQL                                                KACHYDR   995
      CALL DONE                                                    KACHYDR   996
      KPN = 0                                                      KACHYDR   997
      DO 3299 KP=1,NPTOT                                           KACHYDR   998
      XTE = XP(KP)                                                 KACHYDR   999
      YTE = YP(KP)                                                 KACHYDR  1000
      KMAT = 0                                                     KACHYDR  1001
      KFLG = XP(KP).A.3B                                           KACHYDR  1002
      IF (KFLG.LE.1) KMAT = 1                                      KACHYDR  1003
      I = XTE + 2.                                                 KACHYDR  1004
      HPX = FLOAT(I) - 1. - XTE                                    KACHYDR  1005
      HMX = 1. - HPX                                               KACHYDR  1006
      J = YTE + 1.5                                                KACHYDR  1007
      HPY = FLOAT(J) - .5 - YTE                                    KACHYDR  1008
      HMY = 1. - HPY                                               KACHYDR  1009
      CALL RPARU                                                   KACHYDR  1010
      IJ = IJ + NQ*(I-1) + 1 + KMAT                                KACHYDR  1011
      IJP = IJ + NQI                                               KACHYDR  1012
      UK = HPX*HMY*UV(IJP-NQ) + HMX*HMY*UV(IJP)                    KACHYDR  1013
     1     +HPX*HPY*UV(IJ-NQ)  + HMX*HPY*UV(IJ)                    KACHYDR  1014
      I = XTE + 1.5                                                KACHYDR  1015
      HPX = FLOAT(I) - .5 - XTE                                    KACHYDR  1016
      HMX = 1. - HPX                                               KACHYDR  1017
      J = YTE + 2.                                                 KACHYDR  1018
      HPY = FLOAT(J) - 1. - YTE                                    KACHYDR  1019
      HMY = 1. - HPY                                               KACHYDR  1020
      CALL RPARV                                                   KACHYDR  1021
      IJ = IJ + NQ*(I-1) + 1 + KMAT                                KACHYDR  1022
      IJM = IJ - NQI                                               KACHYDR  1023
      VK = HPX*HMY*VV(IJ)  + HMX*HMY*VV(IJ+NQ)                     KACHYDR  1024
     1     +HPX*HPY*VV(IJM) + HMX*HPY*VV(IJM+NQ)                   KACHYDR  1025
      XTE = XTE + UK*DTODR                                         KACHYDR  1026
      YTE = YTE + VK*DTODZ                                         KACHYDR  1027
      IF (XTE.LE.0. .O. XTE.GE.FIBAR) GO TO 3299                   KACHYDR  1028
      IF (YTE.LE.0. .O. YTE.GE.FJBAR) GO TO 3299                   KACHYDR  1029
      KPN = KPN + 1                                                KACHYDR  1030
      XP(KPN) = (XTE.A..N.3B) .O. KFLG                             KACHYDR  1031
      YP(KPN) =  YTE                                               KACHYDR  1032
3299  CONTINUE                                                     KACHYDR  1033
      NPTOT = KPN                                                  KACHYDR  1034
      GO TO 100                                                    KACHYDR  1035
C                                                                  KACHYDR  1036
4000  FORMAT (*0  TAPE DUMP*I3* AT T=*1PE12.5* CYCLE*I5)           KACHYDR  1037
```

```
 4010 FORMAT (*0  RESTARTING FROM TD*I3)                                  KACHYDR 1038
 4020 FORMAT (3X,8A10*   T=*1PE12.5* CYCLE=*I5)                            KACHYDR 1039
 4030 FORMAT (*0 WRONG TAPF - WRONG DUMP.*)                                KACHYDR 1040
 4040 FORMAT (20X*PARTICLES---  VAPOR=., DROPS1=*, DROPS2=SO.*)            KACHYDR 1041
 4050 FORMAT (5XA10,2(5XA8),5X8A10/40X* T=*1PE12.5*  CYCLE=*I4)            KACHYDR 1042
 4060 FORMAT (1H+,16X*MAXIMUM VELOCITY=*1PE12.5)                           KACHYDR 1043
 4080 FORMAT (12X* MIN=*1PE12.5* MAX=*E12.5* L=*E12.5* H=*E12.5* DQ=*      KACHYDR 1044
     1 E12.5)                                                              KACHYDR 1045
 4090 FORMAT (2(1XI3),3X,8(2X,1PE12.5)/11X,8(2X,1PE12.5)/)                 KACHYDR 1046
 4100 FORMAT (*  I    J*10X*TH*12X*UV*12X*VV*11X*SIEV*10X*ROVPR1*8X*ROVPR KACHYDR 1047
     1*10X*D*13X*P*/   17X*E* 13X*UD*12X*VD*11X*SIED*10X*RODPR1*8X*RODPR KACHYDR 1048
     2*10X*A*13X*K*)                                                       KACHYDR 1049
 4110 FORMAT (1H1)                                                         KACHYDR 1050
 4120 FORMAT (* T=*1PE12.5* CYC=*I5* DT=*E12.5* CP=*E12.5* GRINDS=*       KACHYDR 1051
     1E12.5* ITP=*I3*, CELLS=*I4* ITRO=*I3*, CELLS=*I4)                    KACHYDR 1052
 4130 FORMAT (/* T=*1PE12.5*, CYC=*I4* HAS THE FOLLOWING SUMMATIONS...*/  KACHYDR 1053
     I          5X*TH=*1PE15.8*   MOMR=*E15.8*  MOMZV=*E15.8*  MOMZD=*E15. KACHYDR 1054
     18*   MOMZ=*E15.8/*     MV1=*E15.8*    MD1=*E15.6,4X*IEV=*E15.8,4X*I KACHYDR 1055
     2ED=*E15.8,5X*IE=*E15.8/*     MV2 *E15.8*    MD2=*E15.8,4X*KEV=*E15. KACHYDR 1056
     38,4X*KED=*E15.8,5X*KE=*E15.8/5X*MV=*E15.8,5X*MD=*E15.8,5X*EV=*E15. KACHYDR 1057
     48,5X*ED=*E15.8,6X*E=*E15.8)                                          KACHYDR 1058
      END                                                                  KACHYDR 1059
```

```
 INDEX  01/00/75                        SUBROUTINE HYDRO                                          PAGE   45

SINGLY REFEPENCED VARIABLES
 CDR      -R    3CO  EPD      -R   6EQ  GM11     -R   3CO  JX1      -I   3CO  KSC      -    3CN  REAL     -    7F   RO2      -R   3CO
 DATAREL -    75SU  FIXR     -R   3CO  GM12     -R   3CO  JX2      -I   3CO  NOI2     -I   3CO  REWIND   -   80F   RPAR     -R   3CO
 DRSQ     -R    3CO  FIYT     -R   3CO  HYDRO    -    1SU  JX3      -I   3CO  NO2      -I   3CO  RIBAR    -R   3CO  RPARU    -  927SU
 DZO2     -R    3CO  GAM1     -R   3CO  IABRT    -I   3CO  J2       -I   3CO  QSQRT    -  392SU  RIJP2    -  518SU  RPARV    -  937SU
 DZSQ     -R    3CO  GAM2     -R   3CO  IALL     -I   3CO  J3       -I   3CO  RDRSQ    -R   3CO  RJBAR    -R   3CO  XR       -R   3CO
 EM3      -R    3CO  GETQ     -   21SU  JRIGID   -I   3CO  KSB      -    2CN  RDZSQ    -R   3CO  RO1      -R   3CO  YT       -R   3CO
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
MULTIPLY-REFERENCED VARIABLES
    100 -     25*   892    950
    120 -     32AS   36*
    130 -     38*    68AS   91      94
    139 -     4400   47*
    149 -     4300   48*
    150 -     40     50*
    210 -     33     62*
    220 -     36     67*
    230 -     37     70*
    250 -     69     71*
    270 -     23     80*
    280 -     85     95*
    290 -     70AS   97SU
    300 -     34     63     66      98SU
    310 -    100*   131
    319 -    10700  111*
    320 -    103    112*
    330 -    116    122*
    339 -    12200  124    128*
    349 -    11200  129*
    400 -     98    133*
    409 -    13600  139*
    419 -    13500  140*
    429 -    15500  165*
    439 -    15200  166*
    449 -    14100  142    167*
    609 -    17500  179*
    619 -    17400  180*
    629 -    18500  187*
    639 -    18400  188*
    640 -    181    190*
    649 -    19300  194*
    710 -    219    224AS  226*
    720 -    226    231AS  233*    255
    730 -    233    238AS  240*
    740 -    225    232    239     246*
    769 -    21100  217    240     245AS   256*
    779 -    20600  258*
    789 -    20300  261*
    799 -    16900  191    262*
    800 -    168    263*
    820 -    266    267*
    830 -    266    269*
    840 -    263AS  272*
    859 -    278    278    279*
    860 -    271    282*
    870 -    268AS  278    287*
```

```
INDEX  01/00/75                     SUBROUTINE HYDRO                                PAGE   46

 880 -   266     267AS  290*
 889 -   265AS   273AS  276DO  295*
 899 -   275DO   297*
 900 -    35     298*
 909 -   298DO   299*
 919 -   302DO   337*
 929 -   301DO   338*
1000 -    61     343*
1010 -   381     396*
1020 -   395     398*
1030 -   424     427*
1040 -   435     442*
1050 -   442     450*
1060 -   449     454*
1079 -   345DO   460*
1089 -   344DO   467*
1099 -   468DO   476*
1500 -   479*    567
1539 -   482DO   509*
1549 -   481DO   514*
1550 -   479     516*
1560 -   545     548*
1569 -   519DO   551*
1579 -   517DO   552*
1609 -   556DO   560*
1619 -   555DO   561*
1629 -   562DO   564*
1700 -   567     568*
1710 -   578     582*
1720 -   582     586*
1730 -   586     539*
1740 -   589     592*
1759 -   570DO   594*
1769 -   569DO   595*
2089 -   599DO   612*
2099 -   598DO   613*
2130 -   630     631*
2140 -   630     633*
2189 -   618DO   634*
2199 -   617DO   636*
2210 -   649     656*
2220 -   655     663*
2230 -   663     670*
2240 -   669     677*
2250 -   691     699*
2270 -   698     700*
2289 -   641DO   703*
2299 -   640DO   704*
2400 -   709*    805
2450 -   714     726*
2470 -   725     727*
2489 -   711DO   731*
2499 -   710DO   732*
2520 -   746     749*
2530 -   749     765*
2540 -   765     776*
```

```
2560 -   770     792*
2580 -   786     791     796*
2589 -   736DO   793*
2599 -   735DO   794*
2600 -   803     806*
2610 -   813     814*
2620 -   813     816*
2630 -   813     818*
2640 -   815     817     820*
2650 -   820     823*
2660 -   823     826*
2670 -   826     829*
2680 -   829     832*
2689 -   808DO   834*
2699 -   807DO   835*
2700 -   840*    870
2789 -   842DO   863*
2799 -   841DO   864*
3000 -   838     854     871*
3089 -   875DO   889*
3099 -   872DO   890*
3110 -   896     899*
3120 -   899     902*
3130 -   902     905*
3140 -   905     908*
3160 -   895DO   909*
3199 -   894DO   912*
3299 -   915DO   943     944     948*
4000 -    71PR    72WR   951*
4010 -    69PR    92WR   952*
4020 -    90PR    93WR   953*
4030 -    95PR   954*
4040 -   105WR   955*
4050 -   106WR   121WR   150WR   201WR   284WR   292PR   956*
4060 -   149WR   957*
4089 -   200WR   958*
4099 -   279WR   287PR   959*
4100 -   285WR   293PR   960*
4110 -   290PR   961*
4120 -    29PR    30WR   962*
4130 -   339WR   340PR   963*
A        R      4EQ     801    279WR   287PR   383=    391    397=    428    428=    428    576    880=
AA       R      3EQ     73WR    82     83RD
AARG*    R      2EQ     5EQ
AASC     R      3EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ     4EQ
                4EQ     4EQ     4EQ
AB1      R      2EQ     74WR    84RD
ABS      -      45SU    45SU    46SU    46SU   137SU   137SU   138SU   138SU   700SU   700SU   700SU   700SU   727SU   855SU
ADV      -     100SU   117SU   145SU   196SU   270SU   282SU
AFSAZL   -      76SU    77SU    86SU    96SU
AKINTI   -R     3EQ    397     428
AMAX1    -      45SU    46SU   137SU   138SU   178SU   700SU
AMIN1    -      49SU   177SU
ASSIGN   -      32F     68F     70F    224F    231F    238F    245F    263F    265F    267F    268F    273F
AO       -R     3EQ    348     351     354     357     360     363     366     369     502     503     504     505     540     541     542     543
               719     720     721     722     849     850     851     852
```

```
#DT02A   -R     3C0   54*   348    351    360    363    502    503    542    543    719    720    849    850
#DT022   -R     3C0   55*   356    357    366    369    504    505    540    541    721    722    851    852
#01      -R     3C0   405   406    407    408    409
#02      -R     3C0   405   406    407    408    409
#IN1     -R     3C0   443   447    788    624    825    858    884
#OUT     -R     3C0   547   631    632    789    790
#5       -R    663*   648   611
#V1      -R     3C0   383   400    401    402    403    404
#V1OM11  -R     3C0   383
#V2      -R     3C0   383   400    401    402    403    404
#V2OM12  -R     3C0   383
#0       -R     3C0   54     58
CF1B     -R    420*   427   431*   434
CFAL     -R    418*   427   429*   434
CFAR     -R    419*   427   430*   434
CFE1     -R    421*   427   432*   434
COLUM    -    1015U  1155U
COLOUR   1IR    3C0  1194G
COMMON   -       2F    1F
CON      1IR   1C01  1160   194*   200*R  200*R  213    214    215    216    247
CONV     -R    618*  700*   700    706*   706    727    855
CO       1IR    4C0   801   176*   177    178    178    186    186*   213    214    215    216    247    247    247
C3       -R    604*   608   610
C1       -R     3C0  104*R  121*R  150*R  201*R  284*R  292PR
DATA     -      13F   14F   15F    16F    17F    18F
DIMENSI  -       8F    9F   10F
DONE     -    189SU 477SU  515SU  553SU  565SU  596SU  614SU  637SU  705SU  733SU  800SU  836SU  865SU  891SU  913SU
DPC      -R    743*   751   772
DPSTC0A  -R    751*   760   761
DPSTC02  -R    772*   781   782
DPA      -R    750*   751   758
DPAT     1IR    4C0   801   279*R  287PR  743*   811*   811
DP1      -R    771*   772   779
DQ       -R    142*   154   208*R
DR       -R     3C0   47    148    125    207    208
DRDU     -R    143*   157   158
DRDT     -R     3C0   143
DRV      -    142SU 251SU
DS       -R    621*   624   629
DT       -R     3C0  20PR   30PR   50*    51     51*    52     53     54     55     57     59     427    434    506    544
DTIL     -R    724*  726*   727    728
DTM      -R    607*   608   609    609    625*   626    627    627    656*   660    660    670*   674    674    677*   681    681
               684*   688   688    756*   756    762    762    763    763    764    764    776*   777    783    783    784    784
               785    785
DTMSC    -R    608*   610   611
DTMQZ    -R    626*   628   629
DTMQU    -R    750*   760   761
DTMQV    -R    777*   781   782
DTMM     1IR    4C0   801   574*   580*   607    656    677    696    696    755    873    879
DTMASV   -R    873*   874   879*
DTM1     1IR    4C0   801   575*   584*   625    670    684    696    696    776
DTO      1IR    3C0   62     64
DTSC     1IR    3C0   63     64
DTOD1    -R     3C0   55*    56    378    379    506    544    601    661    662    682    683    696    751    941
DTOD2    -R     3C0   57*    58    378    379    506    544    619    675    676    689    690    696    772    942
DTO2     -R     3C0   54*   574    575    580    584
```

```
     INDEX  01/00/75                        SUBROUTINE HYDRO                                    PAGE   49
DTPOS     -R     3CO   49=     50
DZ        -R     3CO   49      109     126     153     204     205
D1        -R     3CO   106WR   121WR   150WR   201WR   284WR   292PR
E        ()R     4EQ     8DI   279WR   287PR   434     874=    881=
EMPTY     -     31SU   341SU
EM10      -R     3CO   33       36      41      63     142     191     192     501     539
EM6       -R     3CO   706
EPS       -R     3CO   706
EPV      ()R     3CO     6EQ   501     539
EP10      -R     3CO   24       88
EP20      -R     3CO   171     172
EP9       -R     3CO   178     186
EQUIVAL  -       4F      5F      6F     11F     12F
ES        -R   622=    626     628
F        ()R     4EQ     8DI   390=    391     391     577     577     745     745
FIBAR     -R     3CO   943
FIXL      -R     3CO   108     125     159     160     248
FIYB      -R     3CO   109     126     154     161     249
FJBAR     -R     3CO   944
FLOAT     -    153SU   194SU   204SU   207SU   922SU   925SU   932SU   935SU
FORMAT    -    951F    952F    953F    954F    955F    956F    957F    958F    959F    960F    961F    962F    963F
FR        -R   501=    506
FRAME     -    102SU   118SU   146SU   197SU
FZ        -R   539=    544
G         -R     3CO   59
GAMTE     -R   385=    386
GDT       -R     3CO   59=     544
GGM11     -R     3CO   385
GGM12     -R     3CO   385
GIROM     -R   386=    389
HMX       -R   923=    930     930     933=    940     940
HMXIDB    -R   368=    378     379     431
HMXIDL    -R   362=    378     379     429
HMXIDR    -R   365=    378     379     430
HMXIDT    -R   371=    378     379     432
HMXIVB    -R   356=    420
HMXIVL    -R   350=    418
HMXIVR    -R   353=    419
HMXIVT    -R   359=    421
HMY       -R   926=    930     930     936=    940     940
HPX       -R   922=    923     930     930     932=    933     940     940
HPXIDB    -R   367=    378     379     431
HPXIDL    -R   361=    378     379     429
HPXIDR    -R   364=    378     379     430
HPXIDT    -R   370=    378     379     432
HPXIVB    -R   355=    420
HPXIVL    -R   349=    418
HPXIVR    -R   352=    419
HPXIVT    -R   358=    421
HPY       -R   925=    926     930     930     935=    936     940     940
```

```
I        -I    3CO   44DO  136DO 155DO 156   175DO 185DO 206DO 207   276DO 279WR 287PR 298DO 299   302DO 305   306
               307   309   310   312   313   314   315   316   345DO 372   373   374   375   378   379   422   422
               422   426   426   426   427   427   427   427   433   433   433   434   434   434   434   435   450
               451   452   453   454   455   456   457   468DO 469   470   473   474   482DO 506   506   506   519DO
               533   544   544   544   556DO 562DO 570DO 578   586   599DO 618DO 641DO 649   664   665   666   667
               668   685   687   688   688   688   689   689   690   690   695   695   695   695   696   696   696
               696   696   696   711DO 724   724   724   726   726   726   736DO 746   749   808DO 819   819   819
               820   826   842DO 854   854   854   857   859   875DO 883   885   895DO 896   902   921=  922   928
               931=  932   938
IBAR     -I    3CO   23    168   206DO 479   482DO 557
IC       -I    193DO 194   194   195=  200WR 211DO 253
ICHAR    -I    114=  127AG
ICHARS   ()I   10DI  15DA  111AG 114
IC1      -I    220=  227=  234=  241=  248   248   249   249
IC2      -I    221=  228=  235=  242=  248   249
IDCON    ()I   10DI  17DA  199WR
IDPP     ()I   10DI  14DA  120WR
IDTO     -I    3CO   62    63    64    64    65=   65
IDVV     ()I   10DI  16DA  148WR
II       -I    21AG  22
IJ       -I    3CO   137   137   138   138   139=  139   157   157   158   165=  165   176   176   177   178   178
               179=  179   186   186   187=  187   209   213   222   229   257=  260=  260   303   304   305   306
               307   309   310   310   312   313   313   314   314   314   314   315   315   316   316   320   321
               325   325   326   326   337=  346   347   351   351   357   357   363   363   369   369   372   374
               376   377   378   378   378   378   378   378   378   379   379   379   379   379   379   380   382
               382   383   383   383   383   384   384   385   385   386   386   388   389   389   390   391   391
               391   391   391   391   392   392   392   392   394   394   397   397   398   399   400   400   400
               400   405   405   405   405   421   422   422   425   425   425   425   425   426   426   426   426
               426   428   428   428   432   433   433   434   434   438   439   445   446   458=  461   462   463
               463   464   464   465   466   469   470   471   472   473   474   475=  475   483   506   507=  507
               510   510   512=  512   520   544   547   548   549=  549   557   557   559=  559   571   572   573
               573   574   574   575   575   576   576   577   577   577   577   577   577   579   580   583   584
               587   590   592=  600   601   602   603   604   605   606   607   610   611   612=  619   620   623
               624   633=  633   642   643   644   645   647   657   658   659   671   672   673   677   678   679
               680   684   685   686   687   691   696   696   696   697   699   699   700   701=  712   713   714
               715   717   723   724   726   726   728   728   728   729=  737   738   739   740   741   742   742
               743   744   745   745   745   745   745   753   754   755   763   764   768   769   774   775   776
               784   785   789   790   794   795   796=  809   810   811   811   811   812   812   814   816   816
               818   818   819   819   819   821   822   824   824   825   825   827   828   830   831   832=  843
               844   845   847   853   854   856   856   861=  873   874   876   877   878   879   880   880   881
               882   882   883   884   884   885   886   887=  900   901   906   907   909=  909   912=  912   928=
               928   929   930   930   938=  939   939   940   940
IJA      -I    483=  486   487   490   493   494   496   520=  523   524   527   530   531   535   546
IJM      -I    3CO   45    45    46    46    47=   47    158   164=  164   277   295=  295   297=  297   325   326
               335=  335   354   354   366   366   378   378   379   379   384   392   392   403   403   403   403
               403   408   408   408   408   408   414   420   420   422   425   425   426   426   431   431   433
               443   444   445   446   447   448   450   451   452   453   460=  460   469   470   471   472   473
               474   476=  476   485   509=  509   514=  514   522   547   551=  551   558   558   560=  560   563
               563   564=  564   583   584   584   585   585   594=  594   619   620   621   622   623   624   625
               628   629   631   631   632   632   634=  634   645   670   671   672   673   696   703=  703   718
               724   726   730=  730   787   788   789   790   794   795   797=  797   819   824   825   834=  834
               848   854   858   862=  862   864   869=  889   897   897   898   898   900   901   903   903   904
               904   906   907   908=  908   911=  911   939=  940   940
IJMA     -I    485=  489   497   500   500   522=  528   530   530   546
```

```
        INDEX  01/00/75                         SUBROUTINE HYDRO                                 PAGE   51
IJP      -I     3CO   210   215   223   243   258=  261=  261   315   316   336=  336   378   379   404   404   404
                404   404   409   409   409   409   409   416   421   432   459=  459   484   508=  508   513=  513
                521   548   550=  550   575   590   591   591   593=  593   685   686   687   702=  702   724   731=
                731   771   771   773   778   779   779   780   792   793   798=  798   830   831   833=  833   854
                860   863=  863   866   884=  888   929=  930   930
IJPA     -I     484=  484   499   499   521=  525   526   529   531   533   533   535
IJ1      -I     222=  229=  236=  243=  247   247
IJ2      -I     223=  230=  237=  244=  247   255
IK       -I     211DO 213   214   215   216   247   252   253
IMJ      -I     304=  325   326   347=  348   348   360   360   373   375   378   379   384   392   392   401   401
                401   401   401   406   406   406   406   406   410   418   422   425   425   426   426   429   433
                436   437   438   439   440   441   572=  579   580   580   581   581   642=  646   656   657   658
                659   696   712=  716   724   726   738=  747   748   768   769   809=  819   821   822   843=  846
                854   857   877=  878   883
IMJA     -I     487=  491   493   493   524=  534   537   537
IMJPA    -I     526=  534   537
IPJ      -I     209=  214   230   236   255   257   303=  314   314   337   346=  378   379   402   402   402   402
                402   407   407   407   407   407   412   419   430   458   571=  574   587   588   588   592   600=
                601   602   605   606   612   643=  678   679   680   701   713=  724   729   737=  750   756   752
                757   758   758   759   766   767   796   810=  827   828   832   844=  854   859   861   876=  885
                887
IPJA     -I     486=  492   494   495   495   495   496   523=  538   538
IPJMA    -I     489=  497   590
IPJP     -I     213=  215   237   244   258
IPJPA    -I     488=  494   525=  538
IP1      -I     3CO   43DO  136DO 155DO 175DO 185DO 302DO 345DO 468DO 519DO 556DO 562DO 570DO 586   599DO 618DO 641DO
                711DO 736DO 749   808DO 826   842DO 859   875DO 885
IP2      -I     3CO   276DO 895DO 902
ISPR     -I     3CO   35
IXL      -I     3CO   102AG 118AG 146AG 197AG
IXR      -I     3CO   102AG 118AG 146AG 197AG
IX1      ()I    10DI  11EQ  108=  111AG 125=  127AG 159=  162AG 163AG 248=  251AG 252AG 253AG
IX2      ()I    10DI  11EQ  160=  162AG 251AG
IYR      -I     3CO   102AG 118AG 146AG 197AG
IYT      -I     3CO   102AG 118AG 146AG 197AG
IY1      ()I    10DI  11EQ  109=  111AG 126=  127AG 154=  162AG 163AG 249=  251AG 252AG 253AG
IY2      ()I    10DI  11EQ  161=  162AG 251AG
J        -I     3CO   43DO  135DO 152DO 153   174DO 184DO 203DO 204   275DO 279WR 287PR 301DO 344DO 442   481DO 510
                517DO 545   548   555DO 569DO 582   589   598DO 617DO 640DO 663   710DO 735DO 768   769   770   786
                807DO 823   829   841DO 858   860   872DO 884   886   894DO 899   905   924=  925   934=  935
JBAR     -I     3CO   81    168   203DO 517DO 548
JNM      -I     3CO   106WR 121WR 150WR 201WR 284WR 292PR
JNSC     -I     82=   83RD
JP1      -I     3CO   43DO  135DO 152DO 174DO 184DO 301DO 344DO 481DO 555DO 569DO 589   598DO 640DO 710DO 735DO 770
                807DO 829   841DO 860   872DO 886
JP2      -I     3CO   275DO 894DO 905
JTD      -I     81=   85    89PR  92WR
JTOP     -I     3CO   617DO
KD       -R     3CO   7RL
KDDDRSQ  -R     3CO   7RL   434
KDDDZSQ  -R     3CO   7RL   434
KFLG     -I     919=  920   946
KIJ      ()R    4EQ   7RL   8DI   279WR 287PR 394=  397=  425   439=  439   446=  446   464=  464   472=  472   574
                574   575   575   580   580   584   584   878=  878
KMAT     -I     113=  124   918=  926=  928   938
KMAYS    ()I    10DI  13DA  113
```

```
KP       -I  915DO  916    917    919
KPN      -I  914=   945=   945    946    947    949
KREQ     -I  478=   483    484    485    501    520    521    522    533    539    557    558    563    566=   566    567    616=
             630    635=
KRET     -I   68AS   70AS   79
KRF      -I  263AS  264    267AS  272=   281    286
KRFP     -I  265AS  268AS  272
KROUT    -I   32AS  132    342
KRP      -I  264=   273AS  289    294
KR1      -I  224AS  231AS  238AS  245AS  250
KV       -R    3CO    7RL
KVODRSQ  -R    3CO    7RL  427
KVODZSQ  -R    3CO    7RL  427
K1       -I  212=   213=   217    217    219    226
K2       -I  212=   214=   217    217    226    233
K3       -I  212=   215=   217    217    219    240
K4       -I  212=   216=   217    217    233    240
L        -I  110=   111AG  123=   124    144=   157    157    158    158    169DO  170    199WR  277=   279WR  279WR  279WR  279WR
             279WR  279WR  279WR  279WR  279WR  279WR  279WR  279WR  279WR  279WR  279WR  279WR  287PR  287PR  287PR  287PR  287PR
             287PR  287PR  287PR  287PR  287PR  287PR  287PR  287PR  287PR  287PR  287PR  339WR  339WR  340PR  340PR
LCM      -I    3CO   74     84
LINCNT   -   104SU  119SU  147SU  198SU  269SU
LINESF   -I  280=   280    281    283=
LINESP   -I  288=   288    289    291=
LL       -I  218=   246=   246    248    249    250    254=
LM1      -I  170=   176    181
LOCF     -    82SU   82SU
LOOP     -    48SU  140SU  166SU  180SU  188SU  259SU  296SU  338SU  467SU  511SU  552SU  561SU  595SU  613SU  636SU  704SU  732SU
             799SU  835SU  864SU  890SU  910SU
LPR      -I    3CO   30     40     72     91    103    116    131    132    266    278    339    340
MUSTIT   -I  708=   727=   727    802    803    804=   839=   855=   855    867    868    869=
MUSTPR   -I    3CO   29PR   30WR  802=
MUSTRO   -I    3CO   29PR   30WR  837=   867=
N        -I   73WR   73WR   74WR   74WR   83RD   83RD   84RD   84RD
NAME     ()I   3CO   90PR   93WR  106WR  121WR  150WR  201WR  284WR  292PR
NCYC     -I    3CO   29PR   30WR   34     39=    39     40     71PR   72WR   90PR   93WR  106WR  121WR  131    150WR  201WR  284WR
             292PR  339WR  340PR  381
NFR      -I   99=   130=   130    131
NLC      -I    3CO   74WR   84RD
NP       -I  107DO  108    109    110    122DO  123    125    126
NPTOT    -I    3CO   98    107DO  122DO  892    915DO  949=
NQ       -I    3CO   47    139    157    164    165    179    187    209    210    277    295    303    304    335    336    346
             347    459    460    463    464    475    476    486    487    488    489    495    507    508    509    510    523
             524    525    526    533    549    550    551    559    560    564    571    572    593    594    600    633    634
             642    643    702    703    712    713    730    731    737    738    797    798    809    810    833    834    843
             844    862    863    873    874    876    877    888    889    897    898    900    900    901    901    903    904
             905    906    907    907    908    909    928    930    930    938    940    949
NOI      -I    3CO  929    939
NQL      -I    3CO  260    261    512    513    514    911    912
NO2L     -I    3CO  297
NSC      -I    3CO   73WR
NTYPE    -I  112DO  113    114    115AG  120WR  141DO  142    143    144    148WR  149WR
NUMIT    -I    3CO   29PR   30WR  707=   801=   801    805
NUMRO    -I    3CO   29PR   30WR  837=   866=   866    870
NUMTD    -I    3CO   71PR   72WR   78=    78     85     87=    87
NUV      -R    3CO    7RL
```

```
NUV3     -R     3C0    7RL    394
NVAP     -I     3C0   813    838
OMBAS    -R     3C0    60
OMBETA  ()R     4EQ    8DI   697=   699=   728    811    819=   856
OMBSPL   -R     3C0    60=   699
OMP      -R     3C0   697
OMRO     -R     3C0   819
OMTHBB   -R   415=    426    434
OMTHLB   -R   411=    426    434
OMTHP6   -R   413=    426    434
OMTHTB   -R   417=    426    434
P       ()R     4EQ    8DI   279WR  287PR  388    391=   391    426    438=   438    445=   445    463=   463    471=   471    577
              601    601    619    619    728    741    742=   747=   750    766=   771    787=   792=
PARUL    -R   692=    696
PARUR    -R   694=    696
PARVB    -R   693=    696
PARVT    -R   695=    696
PC       -R   741=    743
PLT      -    111SU  127SU  163SU  252SU  253SU
PP      ()R     4EQ    8DI   728=   742    750    771
PPC      -R   742=    743    745    747    766    787    792
PRINT    -     29F    71F    89F    96F    95F   287F   290F   292F   293F   340F
PTE      -R   423=   426=    427
PUDENL   -R   652=   660=    692
PUDENR   -R   652    681=    694
PVDENB   -R   666=   674=    693
PVDENT  ()R    10DI   11EQ   666    688=   695
Q        -R   854=   855    856
QMN      -R   171=   177=   177    182    190    194    200WR
QMX      -R   172=   178=   178    182    182    186=   190    200WR
R        -R     3C0   427    434
RA      ()R     4EQ    8DI   576=   577    644    696    745    750    779    880
RBETA    -R   696=   697
RCONT   ()R     3C0   510    768    769
RDENOM   -R   609=   610    611    627=   628    629    762=   763    764    783=   784    785
RDR      -R     3C0    55    696
RDT      -R     3C0    53=    69    696    724    819    854    878
RDZ      -R     3C0    57    422    426    427    433    434    696    724    726    854
READ     -     83F    84F
RETURN   -     38F    97SU
RI      ()R     3C0   156    506    506
RIBJB    -R     3C0    28
RIP     ()R     3C0   372    373    374    375    422    422    426    426    427    427    433    433    434    434    544    544
              696    696    696    696    724    724    726    726    819    819    854    854
RHSQ     -R   389=   390
RODB     -R   665=   673=   674    674    693
RODL     -R   651=   659=   660    660    692
RODPR   ()R     4EQ    8DI   279WR  287PR  306    313    314    314    316    316    378    378    378    378    378    378    378
              378    378    389    405    405    406    406    407    407    408    408    409    409    434    573=   579=   579
              583=   583    587=   587    590=   590    606    606    624    624    659    659    673    673    680    680    687
              687    699    739    759    780
RODPRC   -R   739=   759    780
RODPRT  ()R     4EQ    8DI   378=   380    573
RODPR1  ()R     4EQ    8DI   279WR  287PR  312    313    379    379    379    379    379    379    379    379    379    405    405
              406    406    407    407    408    408    409    409    437=   444=   451=   462=   470=   470=
RODPR1T  -R   379=   380    380    437    444    455    462
```

```
RODR     -R   651     680=    681     681     694
RODRB    -R   759=    761     762     762     763
RODT     ()R   10DI    11EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ    12EQ
               12EQ    12EQ    12EQ    12EQ    12EQ    12EQ   665     687=    688     688     695
RODTB    -R   780=    782     783     783     784
RODTEM   -R   316=    318     319     606=    609     609     610     624=    627     627     628
RODVOL   -R   306=    311     321     326
ROD1TAB()R     10DI    11EQ   451     455=    470
ROTB     -R   531=    532     544
ROUD     ()R    4EQ     8DI   603
ROUJM    -R   500=    506
ROUJP    -R   499=    506
ROULB    -R   493=    506
ROURB    -R   494=    506     506     506     506     506
ROUR2B   -R   495=    506
ROUTE    ()R    4EQ     8DI   506=    510=    510     557
ROUV     ()R    4EQ     8DI   557=    604
ROVB     -R   546=    547     547     672=    674     675
ROVBB    -R   530=    544
ROVD     ()R    4EQ     8DI   621
ROVIM    -R   537=    544
ROVIP    -R   538=    544
ROVL     -R   658=    660     661
ROVPIN   -R     3CO   546     788     824
ROVPIN1  -R     3CO   825     858     884
ROVPR    ()R    4EQ     8DI   279WR   287PR   305     310     314     314     315     315     377     382     386     389     391     394     400
              400     401     401     402     402     403     403     404     404     493     493     494     494     495     495     499     499
              500     500     530     530     531     531     533     537     537     538     538     546     581     585     588     591     605
              605     623     623     658     658     672     672     679     679     686     686     700     724     812=    816     821=    621
              824=    824     824     827=    827     830=    830
ROVPRT   ()R    4EQ     8DI   577=    581=    585=    588=    591=    723     724     724     724     724     744     745=    748=    757     758
              767=    778     779     788=    793=    812
ROVPRTC  -R   723=    724     724     724     724     724
ROVPRTN  -R   745=    748     758     767     779     788     788     793     856=    857     858     858     859     860
ROVPRTO  -R   744=    757     778     853=    854     854     854     854     854     856
ROVPR1   ()R    4EQ     8DI   279WR   287PR   309     310     382     383     383     385     385     400     400     401     401     402     402
              403     403     404     404     818     854     862=    883=    883     884=    884     884     885=    885     886=    886
ROVPR1T()R      4EQ     8DI   814=    816=    818=    822=    822     825=    825     825     828=    828     831=    831     853     854     854
              854     854     856=    857=    858=    859=    860=    882
ROVPR2   -R   382=    383     383
ROVR     -R   679=    681     682
ROVRBN   -R   758=    762     764
ROVRBO   -R   757=    760
ROVSPL   ()R    5EQ     9DI   533
ROVT     -R   686=    688     689
ROVTB    -R   532=    544     544     544     544     544
ROVTBN   -R   779=    783     785
ROVTBO   -R   778=    781
ROVTE    ()R    4EQ     8DI   544=    547=    547     548=    548     558     563
ROVTEM   -R   315=    317     319     605=    609     611     623=    627     629
ROVT2B   -R   533=    544
ROVV     ()R    4EQ     8DI   558=    563=    622
ROVVOL   -R   305=    308     320     325
ROV0     -R     3CO   577     745
RPCOR    -R     3CO   394
RPCOF    -R     3CO   394
```

```
       INDEX  01/00/75                                   SUBROUTINE HYDRO                                      PAGE   55
RR       -R   377=   385    385    388    426    427
RRI     ()R    3CO   378    379    427    434    544    696    819
RRIDR   ()R    3CO   422    426    427    433    434    724    726    854
RRIP    ()R    3CO   506
RRO1     -R    3CO   380
RRO2     -R    3CO   380
RS       -R   760=   763    764
R2A      -R   644=   692    693    694    695
R2DR     -R    3CO   700    819
R2DZ     -R    3CO   700    819
SE       -R   12EQ   334=   334
SECOND   -    19SU    27SU
SED      -R   12EQ   333=   333
SEV      -R   12EQ   332=   332
SIED    ()R    4EQ    8DI   279WR  287PR  321    399    406    407    408    409    429    430    431    432    441=   448=   453=
              466=   474=   474=
SIEDC    -R   399=   405    429    430    431    432    433    434
SIEDEL   -R   422=   427    433=   434
SIEDTAB()R    10DI   11EQ   453    457=   474
SIEDTE   -R   434=   441    448    457    466
SIEV    ()R    4EQ    8DI   279WR  287PR  320    383    386    398    401    402    403    404    418    419    420    421    440=
              447=   452=   465=   473=   473=
SIEVC    -R   398=   400    418    419    420    421    422    427    428
SIEVIN   -R    3CO   447
SIEVTAB()R    10DI   11EQ   452    456=   473
SIEVTE   -R   427=   428    440    447    447    456    465
SIGN     -    348SU  351SU  354SU  357SU  360SU  363SU  366SU  369SU  501SU  502SU  503SU  504SU  505SU  539SU  540SU  541SU  542SU
              543SU  719SU  720SU  721SU  722SU  849SU  850SU  851SU  852SU
SINTE    -R   12EQ   324=   324
SINTED   -R   12EQ   323=   323
SINTEV   -R   12EQ   322=   322
SKE      -R   12EQ   329=   329
SKED     -R   12EQ   328=   328
SKEV     -R   12EQ   327=   327
SMO      -R   12EQ   311=   311
SMO1     -R   12EQ   312=   312
SMO2     -R   12EQ   313=   313
SMOMR    -R   12EQ   314=   314
SMOMZ    -R   12EQ   319=   319
SMOMZD   -R   12EQ   318=   318
SMOMZV   -R   12EQ   317=   317
SMV      -R   12EQ   308=   308
SMV1     -R   12EQ   309=   309
SMV2     -R   12EQ   310=   310
SPSUMS  ()R   10DI   12EQ   299=   339WR  340PR
SQMO     -R    3CO   306
SS       -R   781=   784    785
START    -     42SU  134SU  151SU  173SU  183SU  202SU  274SU  300SU  343SU  480SU  516SU  554SU  568SU  597SU  615SU  639SU  709SU
              734SU  806SU  840SU  871SU  893SU
STED     -R   331=   333    334
STEV     -R   330=   332    334
STH      -R   12EQ   307=   307
STIED    -R   321=   323    324    331
STIEV    -R   320=   322    324    330
STKED    -R   326=   328    329    331
STKEV    -R   325=   327    329    330
```

```
T         -R    3CO    29PR   30WR   33     38     51     51     52=    52     63     71PR   72WR   90PR   93WR   106WR  121WR  150WR
               201WR  284WR  292PR  339WR  340PR
TBASE     -R   19AG   20     37
TDB       -R   408=   434
TDC       -R   405=   427    434    434    434    434    434
TDL       -R   406=   434
TDR       -R   407=   434
TDT       -R   409=   434
TERM1B    -R   667=   675=   693
TERM1L    -R   653=   661=   692
TERM1R    -R   653    682=   694
TERM1T    ()R   10DI   11EQ  667    689=   695
TERM2R    -R   668=   676=   693
TERM2L    -R   654=   662=   692
TERM2R    -R   654    683=   694
TERM2T    ()R   10DI  668    690=   695
TH        ()R    4EQ    8DI  176    279WR  287PR  307    376    410    412    414    416    436=   443=   450=   461=   469=   469=
               577    602    602    620    620    657    657    671    671    678    678    685    685    691    714    740    745
               752    773
THBB      -R   414=   415    426    427
THIJ      -R   376=   410    412    414    416
THIN      -R    3CO   443
THLB      -R   410=   411    426    427
THRB      -R   412=   413    426    427    752=   760    761
THTAB     ()R   10DI   11EQ   11EQ   11EQ   11EQ   11EQ   11EQ   11EQ   11EQ  450    454=   469
THTB      -R   416=   417    426    427    773=   781    782
THTE      -R   380=   381    383    387    387    388    393    396=   424    436    443    443    454    461    602=   603    604
               620=   621    622
THTEB     -R   664=   671=   675    676    696
THTEC     -R   740=   752    773
THTEL     -R   650=   657=   661    662    696
THTER     -R   650    678=   682    683    696
THTERM    -R   393=   394
THTESQ    -R   387=   389    393
THTET     ()R   10DI   11EQ  664    685=   689    690    696
THO       -R    3CO   381    424    691    714
TLIM      -R    22=    24=    24     37     88=    88
TLIMD     -R    3CO    24     88
TOLD      -R    18DA   26=    28
TOP       -R    3CO   548    794    795
TOUT      -R    3CO    33     51     51     62=    62     64=
TVB       -R   403=   427
TVC       -R   400=   427    427    427    427    427    434
TVL       -R   401=   427
TVR       -R   402=   427
TVT       -R   404=   427
TWFIN     -R    3CO    38
T1        -R    20=    36     67=
T2        -R    18DA   26     27AG   28     29PR   30WR   36     37     67
T20MD     -R    3CO    36
UC        -R   491=   502    502    506
UD        ()R    4EQ    8DI   45    138    279WR  287PR  314    326    326    340    360    363    363    374    375    392    392
               425    425    426    426    433    433    611=   726    726    754    764=   769=   769    901=   901    907=   907
UDRL      -R   375=   378    379    429
UDRR      -R   374=   378    379    430
UDTE      -R   754=   756    761
```

```
UK       -R   930=    941
UR       -R   492=    503     503     506
URB      -R   490=    491     492     494     501     501
US       -R   761=    763     764
UTB      -R   536=    539     539
UTLC     -R   534=    536     542     542     544
UTRC     -R   535=    536     543     543     544
UV      ()R     4EQ     8DI     45    137     157     157     279WR   287PR   314     325     325     348     348     351     351     372     373
              384     384     392     392     422     422     425     425     426     426     490     491     492     493     495     499     500
              534     534     535     535     610=    645     646     715     716     753     763=    768=    768     819     819     845     846
              900=    900     906=    906     930     930     930     930
UVL      -R   646=    692     700
UVLB     -P   716=    720     720     724     846=    850     850     854
UVR      -R   645=    694     700
UVRB     -R   715=    719     719     724     845=    849     849     854
UVRL     -R   373=    418
UVRR     -R   372=    419
UVTE     -R   753=    756     760
VBRC     -R   497=    498     504     504     506
VC       -R   528=    540     540     544
VD      ()R     4EQ     8DI     46    138     279WR   287PR   316     326     326     366     366     369     369     378     378     379     379
              392     392     425     425     426     426     431     432     433     433     629=    632=    632     726     726     775     785=
              790=    790     795=    795     898=    898     904=    904
VDTE     -R   775=    777     782
VECVEL   -R   392=    394
VEL      -R   384=    388=    388     389     423=    425=    427
VELMX   ()R    10DI     41=     41=    45=     45      46=     46      49      49      133=    133=    137=    137     138=    138     142     143
              149WR
VK       -R   940=    942
VOL     ()R     3CO   305     306     307     309     310     312     313     315     316
VOLR    ()R     3CO   314
VRB      -R   498=    501     501     501
VS       -R   782=    784     785
VT       -R   529=    541     541     544
VTB      -P   527=    528     529     532     539     539
VTRC     -R   496=    498     505     505     506
VV      ()R     4EQ     8DI     46    137     158     158     279WR   287PR   315     325     325     354     354     357     357     384     384
              392     392     420     421     422     422     425     425     426     426     496     496     497     497     527     528     529
              530     533     537     538     546     628=    631=    631     647     648     717     718     774     784=    789=    789     794=
              794     819     819     847     848     897=    897     903=    903     940     940     940     940
VVB      -R   648=    693     700
VVRB     -R   718=    722     722     724     848=    852     852     854
VVIN     -R     3CO   631     789
VVT      -R   647=    695     700
VVTB     -R   717=    721     721     724     847=    851     851     854
VVTE     -R   774=    777     781
WRITE    -      30F     72F     73F   74F     92F     93F     105F    106F    120F    121F    148F    149F    150F    199F    200F    201F    279F
              284F    285F    339F
XCCHV    -R     3CO   108     125     159     160     248
XIBR     -R   504=    506     506
XIC      -R   502=    506     506     540=    544     544
XIDR     -R   366=    367     368
XIDL     -R   360=    361     362
XIDR     -R   363=    364     365
XIDT     -R   369=    370     371
XIR      -R   503=    506     506
```

```
XIT      -R  541=    544    544
XITL     -R  542=    544    544
XITR     -R  505=    506    506    543=   544    544
XIVB     -R  354=    355    356    722=   724    724    852=   854    854
XIVL     -R  348=    349    350    720=   724    724    850=   854    854
XIVR     -R  351=    352    353    719=   724    724    849=   854    854
XIVT     -R  357=    358    359    721=   724    724    851=   854    854
XL       -R    3C0   108    125    159    160    248
XP      ()R    3C0   108    110    123    125    916    919    946=
XTE      -R  916=    921    922    931    932    941=   941    943    943    946
XX       -R   28=     29PR   30WR  182=   186    190=   191    192    247=   248    249    601=   603    604    619=   621    622
X1      ()R   10DI    11EQ  156=   157    159    207=   207=   208=   208=   208    248    248    248
X2       -R  157=    160
YB       -R    3C0   109    126    154    161    249
YCONV    -R    3C0   109    126    154    161    249
YP      ()R    3C0   109    126    917    947=
YTE      -R  917=    924    925    934    935    942=   942    944    944    947
Y1      ()R   10DI    11EQ  153=   154    158    204=   204=   205=   205=   205    249    249    249
Y2       -R  158=    161
ZZ       -R    3C0    82
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

```
INDEX  01/00/75                        MASTER INDEX                                          PAGE   59

LIST OF ALL VARIABLES DEFINED IN INPUT
 C MEANS VARIABLE WAS DEFINED IN COMMON IN THAT ROUTINE
 D MEANS VARIABLE WAS DEFINED IN OTHER DECLARATIONS
 NON-BLANK NUMERIC IS NUMBER OF NON-DECLARATORY REFERENCES
 S PRECEDING MEANS SUBROUTINE(PROGRAM,FUNCTION) NAME
 L PRECEDING MEANS COMMON(LCM) NAME
 F PRECEDING MEANS FORTRAN KEYWORD
 * PRECEDING MEANS VARIABLE IS DECLARED, NOT USED ANYWHERE

VARIABLE  ROUTINE TYPE  ROUTINE TYPE  ROUTINE TYPE  ROUTINE TYPE
  A        KACHINA  D    SETUP     D   HYDRO   10D
  AA       KACHINA  C    LOOP      C   SETUP    1C   HYDRO    3C
  AAROW    KACHINA  CD   LOOP     1C   SETUP    CD   HYDRO    CD
  AASC     KACHINA  CD   LOOP     1C   SETUP   1CD   HYDRO    CD
  AA1      KACHINA  C    LOOP      C   SETUP    2C   HYDRO    2C
S ABS      HYDRO   14
S ADV      HYDRO    6
S AFSREL   HYDRO    4
  AKINFI   KACHINA  C    LOOP      C   SETUP    1C   HYDRO    2C
S AMAX1    SETUP    1    HYDRO     6
S AMIN1    HYDRO    2
F ASSIGN   SETUP    2    HYDRO    12
  A0       KACHINA 1C    LOOP      C   SETUP    1C   HYDRO   24C
  BDTODR   KACHINA  C    LOOP      C   SETUP    1C   HYDRO   13C
  BDTODZ   KACHINA  C    LOOP      C   SETUP    1C   HYDRO   13C
  BD1      KACHINA  C    LOOP      C   SETUP    2C   HYDRO    5C
  BD2      KACHINA  C    LOOP      C   SETUP    2C   HYDRO    5C
  BINF     KACHINA  C    LOOP      C   SETUP    7C   HYDRO    7C
  BOUT     KACHINA  C    LOOP      C   SETUP    2C   HYDRO    5C
  BS       HYDRO    3
  BV1      KACHINA  C    LOOP      C   SETUP    4C   HYDRO    6C
  BV1GM11  KACHINA  C    LOOP      C   SETUP    2C   HYDRO    1C
  BV2      KACHINA  C    LOOP      C   SETUP    4C   HYDRO    6C
  BV2GM12  KACHINA  C    LOOP      C   SETUP    2C   HYDRO    1C
  B0       KACHINA 1C    LOOP      C   SETUP    3C   HYDRO    2C
  CDR      KACHINA 1C    LOOP      C   SETUP    2C   HYDRO     C
  CFXB     HYDRO    4
  CFXL     HYDRO    4
  CFXR     HYDRO    4
  CFXT     HYDRO    4
S CLOCK1   KACHINA  1
S COLOR    HYDRO    2
  COLOUR   KACHINA  C    LOOP      C   SETUP    2C   HYDRO    1C
F COMMON   KACHINA  2    LOOP      2   SETUP    2    HYDRO    2
  CON      HYDRO   8D
  CONV     HYDRO    7
  CQ       KACHINA  D    SETUP     D   HYDRO   13D
  CS       HYDRO    3
  C1       KACHINA 1C    LOOP      C   SETUP     C   HYDRO    6C
S DATAREL  HYDRO    1
F DATA     SETUP    2    HYDRO     6
S DATE1    KACHINA  1
F DIMENSI  SETUP    3    HYDRO     3
  DONE     LOOP     1    SETUP     2   HYDRO   15
  DPC      HYDRO    3
```

```
  DPDTODR  HYDRO    3
  DPDTODZ  HYDRO    3
  DPR      HYDRO    3
  DPRT     KACHINA  D   SETUP   D   HYDRO   5D
  DPT      HYDRO    3
  DQ       HYDRO    3
  DR       KACHINA  1C  LOOP    C   SETUP   10C   HYDRO   5C
  DROU     HYDRO    3
  DRO2     KACHINA  C   LOOP    C   SETUP   1C    HYDRO   1C
  DRSQ     KACHINA  C   LOOP    C   SETUP   3C    HYDRO    C
S DRV      HYDRO    2
  DS       HYDRO    3
  DT       KACHINA  C   LOOP    C   SETUP   8C    HYDRO  15C
  DTIL     HYDRO    4
  DTK      HYDRO   36
  DTKBC    HYDRO    3
  DTKDE    HYDRO    3
  DTKDU    HYDRO    3
  DTKDV    HYDRO    3
  DTKR     KACHINA  D   SETUP   D   HYDRO   10D
  DTKRSV   HYDRO    3
  DTKT     KACHINA  D   SETUP   D   HYDRO   8D
  DTO      KACHINA  C   LOOP    C   SETUP   3C    HYDRO   2C
  DTOC     KACHINA  C   LOOP    C   SETUP   2C    HYDRO   2C
  DTODR    KACHINA  C   LOOP    C   SETUP   2C    HYDRO  14C
  DTODZ    KACHINA  C   LOOP    C   SETUP   2C    HYDRO  14C
  DTO2     KACHINA  C   LOOP    C   SETUP   1C    HYDRO   5C
  DTPOS    KACHINA  C   LOOP    C   SETUP   1C    HYDRO   2C
  DZ       KACHINA  1C  LOOP    C   SETUP   7C    HYDRO   6C
  DZO2     KACHINA  C   LOOP    C   SETUP   1C    HYDRO    C
  DZSQ     KACHINA  C   LOOP    C   SETUP   3C    HYDRO    C
  D1       KACHINA  1C  LOOP    C   SETUP    C    HYDRO   6C
  E        KACHINA  D   SETUP   D   HYDRO   5D
S EMPTY    KACHINA  1   HYDRO   2
  EM10     KACHINA  C   LOOP    C   SETUP   3C    HYDRO   8C
  EM3      KACHINA  C   LOOP    C   SETUP   1C    HYDRO    C
  EM6      KACHINA  C   LOOP    C   SETUP   1C    HYDRO   1C
F ENTRY    LOOP     9
  EPD      KACHINA  D   SETUP   2D  HYDRO    D
  EPS      KACHINA  C   LOOP    C   SETUP   2C    HYDRO   1C
  EPV      KACHINA  CD  LOOP    C   SETUP   2CD   HYDRO   2CD
  EP10     KACHINA  C   LOOP    C   SETUP   2C    HYDRO   2C
  EP20     KACHINA  C   LOOP    C   SETUP   1C    HYDRO   2C
  EP9      KACHINA  C   LOOP    C   SETUP   1C    HYDRO   2C
F EQUIVAL  KACHINA  3   SETUP   3   HYDRO   5
  ES       HYDRO    3
  F        KACHINA  D   SETUP   D   HYDRO   7D
  FIBAR    KACHINA  C   LOOP    C   SETUP   3C    HYDRO   1C
  FILM     KACHINA  2
  FIXL     KACHINA  C   LOOP    C   SETUP   3C    HYDRO   5C
  FIXR     KACHINA  C   LOOP    C   SETUP   3C    HYDRO    C
  FIYB     KACHINA  C   LOOP    C   SETUP   3C    HYDRO   5C
  FIYT     KACHINA  C   LOOP    C   SETUP   3C    HYDRO    C
  FJBAR    KACHINA  C   LOOP    C   SETUP   3C    HYDRO   1C
S FLOAT    SETUP    9   HYDRO   8
  FNB      SETUP    3
```

```
   INDEX  01/00/75                    MASTER INDEX                                          PAGE  61
  FNL      SETUP    3
  FNPUA    SETUP    4
  FNR      SETUP    2
  FNT      SETUP    2
F FORMAT   KACHINA  1    KASET    1    SETUP   17   KACHYDR       HYDRO   13
  FR       HYDRO    2
S FRAME    HYDRO    4
  FSET12   KACHINA  1
  FSET7    KACHINA  1
  FSET8    KACHINA  1
  FSET9    KACHINA  1
  FZ       HYDRO    2
  G        KACHINA   C   LOOP      C   SETUP   3C   HYDRO    1C
  GAMTE    HYDRO    2
  GAM1     KACHINA   C   LOOP      C   SETUP   4C   HYDRO     C
  GAM2     KACHINA   C   LOOP      C   SETUP   4C   HYDRO     C
  GDT      KACHINA   C   LOOP      C   SETUP   1C   HYDRO    2C
S GETO     KACHINA  1    HYDRO    1
  GGM11    KACHINA   C   LOOP      C   SETUP   1C   HYDRO    1C
  GGM12    KACHINA   C   LOOP      C   SETUP   1C   HYDRO    1C
  GIROM    HYDRO    2
  GM11     KACHINA   C   LOOP      C   SETUP   3C   HYDRO     C
  GM12     KACHINA   C   LOOP      C   SETUP   3C   HYDRO     C
  HEIGHT   SETUP    3
  HMX      HYDRO    6
  HMXIDB   HYDRO    4
  HMXIDL   HYDRO    4
  HMXIDR   HYDRO    4
  HMXIDT   HYDRO    4
  HMXIVB   HYDRO    2
  HMXIVL   HYDRO    2
  HMXIVR   HYDRO    2
  HMXIVT   HYDRO    2
  HMY      HYDRO    6
  HPX      HYDRO    8
  HPXIDB   HYDRO    4
  HPXIDL   HYDRO    4
  HPXIDR   HYDRO    4
  HPXIDT   HYDRO    4
  HPXIVB   HYDRO    2
  HPXIVL   HYDRO    2
  HPXIVR   HYDRO    2
  HPXIVT   HYDRO    2
  HPY      HYDRO    8
S HYDRO    KACHYDR  1    HYDRO    1
  I        KACHINA   C   LOOP     1C   SETUP  21C   HYDRO    99C
  IARRT    KACHINA  2C   LOOP      C   SETUP   1C   HYDRO     C
  IALL     KACHINA   C   LOOP      C   SETUP   1C   HYDRO     C
  IBAR     KACHINA  2C   LOOP      C   SETUP   4C   HYDRO    4C
  IBOT     SETUP    4
  IC       HYDRO    7
  ICHAR    HYDRO    2
  ICHARS   HYDRO    2D
  IC1      HYDRO    8
  IC2      HYDRO    6
  IDCOMP   SETUP    2D
```

```
IOCON    HYDRO     10
IOPP     HYDRO     10
IOTO     KACHINA    C   LOOP       C   SETUP    1C   HYDRO    6C
IOVV     HYDRO     10
II       HYDRO      2
IJ       KACHINA    C   LOOP      6C   SETUP   29C   HYDRO   49C
IJA      HYDRO     15
IJM      KACHINA    C   LOOP      3C   SETUP    8C   HYDRO   39C
IJMA     HYDRO     10
IJP      KACHINA    C   LOOP      3C   SETUP    3C   HYDRO   74C
IJPA     HYDRO     12
IJ1      HYDRO      5
IJ2      HYDRO      5
IK       HYDRO      8
IMJ      HYDRO     73
IMJA     HYDRO      8
IMJPA    HYDRO      3
IN       SETUP     10
INP      KACHINA    1
IPJ      HYDRO     69
IPJA     HYDRO     10
IPJMA    HYDRO      3
IPJP     HYDRO      5
IPJPA    HYDRO      4
IP1      KACHINA    C   LOOP       C   SETUP    3C   HYDRO   25C
IP2      KACHINA    C   LOOP       C   SETUP    4C   HYDRO    3C
ISPD     KACHINA    C   LOOP       C   SETUP    2C   HYDRO    1C
ITOP     SETUP      4
IXL      KACHINA    C   LOOP       C   SETUP    1C   HYDRO    4C
IXR      KACHINA    C   LOOP       C   SETUP    1C   HYDRO    4C
IX1      HYDRO    110
IX2      HYDRO     30
IYB      KACHINA    C   LOOP       C   SETUP    1C   HYDRO    6C
IYT      KACHINA    C   LOOP       C   SETUP    1C   HYDRO    4C
IY1      HYDRO    110
IY2      HYDRO     30
J        KACHINA    C   LOOP      4C   SETUP   10C   HYDRO   49C
JBAR     KACHINA   1C   LOOP       C   SETUP    4C   HYDRO    5C
JJ       LOOP       2
JMM      KACHINA   1C   LOOP       C   SETUP     C   HYDRO    6C
JMSC     HYDRO      2
JP1      KACHINA    C   LOOP       C   SETUP    4C   HYDRO   22C
JP2      KACHINA    C   LOOP       C   SETUP    5C   HYDRO    3C
JRIGID   KACHINA    C   LOOP       C   SETUP    3C   HYDRO     C
JTD      HYDRO      4
JTOP     KACHINA    C   LOOP       C   SETUP    1C   HYDRO    1C
JX1      KACHINA    C   LOOP      1C   SETUP    1C   HYDRO     C
JX2      KACHINA    C   LOOP      1C   SETUP    1C   HYDRO     C
JX3      KACHINA    C   LOOP      1C   SETUP    1C   HYDRO     C
J2       KACHINA    C   LOOP       C   SETUP    3C   HYDRO     C
J3       KACHINA    C   LOOP       C   SETUP    2C   HYDRO     C
K        LOOP       3   SETUP      2
$ KACHINA KACHINA   1
$ KACHYDR KACHYDR   1
$ KASET  KASET      1
KO       KACHINA   C0   LOOP       C   SETUP   4C0   HYDRO    C0
```

```
   INDEX  01/00/75                            MASTER INDEX                                     PAGE   63
 KDODRSQ  KACHINA   CD  LOOP       C    SETUP     1CD  HYDRO     1CD
 KDODZSQ  KACHINA   CD  LOOP       C    SETUP     1CD  HYDRO     1CD
 KF       SETUP    3
 KFLG     HYDRO    3
 KTJ      KACHINA   D   SETUP      D    HYDRO    23D
 KK       LOOP     2
 KL       SETUP    2
 KMAT     SETUP    2    HYDRO     6
 KMATS    SETUP    1D   HYDRO     1D
 KP       HYDRO    4
 KPN      HYDRO    6
 KREQ     HYDRO   14
 KRET     SETUP    8    HYDRO     3
 KRF      HYDRO    6
 KRFP     HYDRO    3
 KROUT    HYDRO    3
 KRP      HYDRO    4
 KR1      HYDRO    5
L KSB     KACHINA  1    LOOP      1     SETUP     1    HYDRO      1
L KSC     KACHINA  1    LOOP      1     SETUP     1    HYDRO      1
 KT       SETUP   11
 KV       KACHINA   CD  LOOP       C    SETUP     4CD  HYDRO      CD
 KVODRSQ  KACHINA   CD  LOOP       C    SETUP     1CD  HYDRO     1CD
 KVODZSQ  KACHINA   CD  LOOP       C    SETUP     1CD  HYDRO     1CD
 K1       HYDRO    6
 K2       HYDRO    6
 K3       HYDRO    6
 K4       HYDRO    6
 L        HYDRO   49
 LCM      KACHINA   C   LOOP      1C    SETUP     1C   HYDRO     2C
 LCMFLG   LOOP     1    SETUP     1
S LINCNT  HYDRO    5
 LINESF   HYDRO    4
 LINESP   HYDRO    4
 LL       HYDRO    7
 LM1      HYDRO    3
S LOCF    SETUP    4    HYDRO     2
S LOOP    LOOP     1    SETUP     2     HYDRO    22
 LPR      KACHINA   C   LOOP       C    SETUP     5C   HYDRO    12C
 MUSTIT   HYDRO   12
 MUSTPR   KACHINA   C   LOOP       C    SETUP      C   HYDRO     3C
 MUSTPO   KACHINA   C   LOOP       C    SETUP      C   HYDRO     4C
 N        SETUP   12    HYDRO     8
 NAME     KACHINA   C   LOOP       C    SETUP     2C   HYDRO     8C
 NB       SETUP    5
 NB2      SETUP    2
 NCYC     KACHINA   C   LOOP       C    SETUP     1C   HYDRO    20C
 NFR      HYDRO    4
 NL       SETUP    5
 NLC      KACHINA   C   LOOP       C    SETUP     1C   HYDRO     2C
 NL2      SETUP    2
 NP       HYDRO    8
 NPTOT    KACHINA   C   LOOP       C    SETUP     7C   HYDRO     6C
 NPUA     SETUP    5
 NPX      SETUP    4
 NPY      SETUP    4
```

```
   NQ       KACHINA  1C   LOOP     1C   SETUP    16C   HYDRO    96C
   NQI      KACHINA   C   LOOP     5C   SETUP     7C   HYDRO     2C
   NQI2     KACHINA   C   LOOP      C   SETUP     1C   HYDRO      C
   NQL      KACHINA   C   LOOP      C   SETUP     3C   HYDRO     7C
   NQM1     SETUP    2
   NQ2      KACHINA   C   LOOP     3C   SETUP     1C   HYDRO      C
   NQ2L     KACHINA   C   LOOP      C   SETUP     2C   HYDRO     1C
   NR       SETUP    6
   NR1      SETUP    2
   NSC      KACHINA   C   LOOP      C   SETUP     1C   HYDRO     1C
   NT       SETUP    5
   NTYPE    SETUP    3    HYDRO    11
   NT1      SETUP    2
   NUMIT    KACHINA   C   LOOP      C   SETUP      C   HYDRO     6C
   NUMRO    KACHINA   C   LOOP      C   SETUP      C   HYDRO     6C
   NUMTO    KACHINA   C   LOOP      C   SETUP     1C   HYDRO     7C
   NUV      KACHINA  1CD  LOOP      C   SETUP    2CD   HYDRO     CD
   NUV3     KACHINA   CD  LOOP      C   SETUP    1CD   HYDRO    1CD
   NVAP     KACHINA   C   LOOP      C   SETUP     3C   HYDRO     2C
   OFFSET   SETUP    3
   OMBAS    KACHINA   C   LOOP      C   SETUP     2C   HYDRO     1C
   OMBETA   KACHINA   D   SETUP     D   HYDRO     6D
   OMBSPL   KACHINA   C   LOOP      C   SETUP     1C   HYDRO     2C
   OMP      KACHINA   C   LOOP      C   SETUP     5C   HYDRO     1C
   OMRO     KACHINA   C   LOOP      C   SETUP     2C   HYDRO     1C
   OMTHBB   HYDRO    3
   OMTHLB   HYDRO    3
   OMTHRB   HYDRO    3
   OMTHTB   HYDRO    3
   OUT      KACHINA  2
S  OVERLAY  KACHINA  2
   P        KACHINA   D   SETUP    1D   HYDRO    28D
   PARUL    HYDRO    2
   PARUR    HYDRO    2
   PARVB    HYDRO    2
   PARVT    HYDRO    2
   PC       HYDRO    2
   PI       SETUP    1D
S  PLT      HYDRO    5
   PNEFF    SETUP    4
   PNI      SETUP    2
   PP       KACHINA   D   SETUP     D   HYDRO     4D
   PPC      HYDRO    7
F  PRINT    KASET    1    SETUP    3    KACHYDR  1     HYDRO    10
   PTE      HYDRO    3
   PUDENL   HYDRO    3
   PUDENR   HYDRO    3
   PVDENB   HYDRO    3
   PVDENT   HYDRO    3D
   Q        HYDRO    3
   QMN      HYDRO    7
   QMX      HYDRO    8
S  QSQRT    SETUP    1    HYDRO    1
   R        KACHINA   C   LOOP      C   SETUP     2C   HYDRO     2C
   RA       KACHINA   D   SETUP     D   HYDRO     8D
   RBETA    HYDRO    2
```

```
  RCONT     KACHINA   C   LOOP      C   SETUP    2C   HYDRO     3C
  RDENOM    HYDRO    12
  RDR       KACHINA   C   LOOP      C   SETUP    8C   HYDRO     2C
  RDRSQ     KACHINA   C   LOOP      C   SETUP    3C   HYDRO      C
  RDT       KACHINA   C   LOOP      C   SETUP    2C   HYDRO     7C
  RDZ       KACHINA   C   LOOP      C   SETUP    5C   HYDRO    10C
  RDZSQ     KACHINA   C   LOOP      C   SETUP    3C   HYDRO      C
F READ      KACHINA   1   SETUP     9   HYDRO    2
F REAL      KACHINA   1   SETUP     1   HYDRO    1
F RETURN    LOOP     10   SETUP     2   HYDRO    2
F REWIND    HYDRO     1
  RI        KACHINA   C   LOOP      C   SETUP    5C   HYDRO     3C
  RIBAR     KACHINA   C   LOOP      C   SETUP    2C   HYDRU      C
  RIBJB     KACHINA   C   LOOP      C   SETUP    1C   HYDRO     1C
  RIJP2     LOOP      1   HYDRO     1
  RIP       KACHINA   C   LOOP      C   SETUP    5C   HYDRO    28C
  RJBAR     KACHINA   C   LOOP      C   SETUP    2C   HYDRO      C
  RMSQ      HYDRO     2
  RODB      HYDRO     5
  RODL      HYDRO     5
  RODPP     KACHINA   D   SETUP    3D   HYDRO   54D
  RODPRC    HYDRO     3
  RODPRI    SETUP     3
  RODPRI1   SETUP     7
  RODPRI2   SETUP     5
  ROOPRT    KACHINA   D   SETUP     D   HYDRO    3D
  ROOPRI    KACHINA   D   SETUP    3D   HYDRO   29D
  RODPRIT   HYDRO     7
  RODR      HYDRO     5
  RODRAT    SETUP     3
  RODRB     HYDRO     5
  RODT      HYDRO    5D
  RODTB     HYDRO     5
  RODTEM    HYDRO    11
  RODVOL    HYDRO     4
  RODITAB   HYDRO    3D
  ROTB      HYDRO     3
  ROUO      KACHINA   D   SETUP     D   HYDRO    1D
  ROUJM     HYDRO     2
  ROUJP     HYDRO     2
  ROULB     HYDRO     2
  ROURB     HYDRO     6
  ROURZB    HYDRO     2
  ROUTE     KACHINA   D   SETUP     D   HYDRO    4D
  ROUV      KACHINA   D   SETUP     D   HYDRO    2D
  ROVB      HYDRO     6
  ROVBB     HYDRO     2
  ROVD      KACHINA   D   SETUP     D   HYDRO    1D
  ROVIM     HYDRO     2
  ROVIN1    SETUP     4
  ROVIN2    SETUP     3
  ROVIP     HYDRO     2
  ROVL      HYORD     3
  ROVPIN    KACHINA   C   LOOP      C   SETUP    2C   HYDRO     3C
  ROVPIN1   KACHINA   C   LOOP      C   SETUP    2C   HYDRO     3C
  ROVPR     KACHINA   D   SETUP    6D   HYDRO   73D
```

```
  ROVPRI    SETUP     4
  ROVPRI1   SETUP     5
  ROVPRI2   SETUP     3
  ROVPRT    KACHINA    D    SETUP      D    HYDRO    21D
  ROVPRTC   HYDRO     6
  ROVPRTN   HYDRO    14
  ROVPRTO   HYDRO    10
  ROVPR1    KACHINA    D    SETUP     6D    HYDRO    31D
  ROVPR1T   KACHINA    D    SETUP      D    HYDRO    23D
  ROVPR2    HYDRO     3
  ROVR      HYDRO     3
  ROVRBN    HYDRO     3
  ROVRBO    HYDRO     2
  ROVSPL    KACHINA    D    SETUP      D    HYDRO     1D
  ROVT      HYDRO     3
  ROVTB     HYDRO     6
  ROVTBN    HYDRO     3
  ROVTBO    HYDRO     2
  ROVTE     KACHINA    D    SETUP      D    HYDRO     7D
  ROVTEM    HYDRO     9
  ROVT28    HYDRO     2
  ROVV      KACHINA    D    SETUP      D    HYDRO     3D
  ROVVOL    HYDRO     4
  ROV0      KACHINA   1C    LOOP       C    SETUP     1C    HYDRO     2C
  ROV1      SETUP     6
  ROV2      SETUP     5
  RO1       KACHINA    C    LOOP       C    SETUP     3C    HYDRO      C
  RO2       KACHINA    C    LOOP       C    SETUP     3C    HYDRO      C
  RPAR      KACHINA   1C    LOOP       C    SETUP     3C    HYDRO      C
  RPARU     LOOP      1     HYDRO     1
  RPARV     LOOP      1     HYDRO     1
  RPCDR     KACHINA    C    LOOP       C    SETUP     1C    HYDRO     1C
  RPCOF     KACHINA    C    LOOP       C    SETUP     1C    HYDRO     1C
  RR        HYDRO     6
  RRI       KACHINA    C    LOOP       C    SETUP     4C    HYDRO     7C
  RRIDR     KACHINA    C    LOOP       C    SETUP     2C    HYDRO     8C
  RRIP      KACHINA    C    LOOP       C    SETUP     2C    HYDRO     1C
  RRO1      KACHINA    C    LOOP       C    SETUP     2C    HYDRO     1C
  RRO2      KACHINA    C    LOOP       C    SETUP     2C    HYDRO     1C
  RS        HYDRO     3
  R1ROW     LOOP      1     SETUP     1
  R2A       HYDRO     5
  R2DR      KACHINA    C    LOOP       C    SETUP     1C    HYDRO     2C
  R2DZ      KACHINA    C    LOOP       C    SETUP     1C    HYDRO     2C
  SE        HYDRO    2D
S SECOND    HYDRO     2
  SED       HYDRO    2D
  SETIJ     LOOP      1     SETUP     1
S SETUP     KASET     1     SETUP     1
  SEV       HYDRO    2D
  SIED      KACHINA    D    SETUP     3D    HYDRO    18D
  SIEDC     HYDRO     8
  SIEDEL    HYDRO     4
  SIEDT     SETUP     4
  SIEDTAB   HYDRO    3D
  SIEDTE    HYDRO     5
```

```
   INDEX  01/00/75                     MASTER INDEX                                    PAGE   67
 SIEV      KACHINA   D    SETUP     6D    HYDRO    20D
 SIEVC     HYDRO    9
 SIEVT     SETUP    5
 SIEVTN    KACHINA   C    LOOP          C    SETUP     3C    HYDRO      1C
 SIEVTAB   HYDRO    3D
 SIEVTE    HYDRO    7
S SIGN     HYDRO   26
 SINTE     HYDRO    2D
 SINTED    HYDRO    2D
 SINTEV    HYDRO    2D
 SKE       HYDRO    2D
 SKED      HYDRO    2D
 SKEV      HYDRO    2D
 SMD       HYDRO    2D
 SMD1      HYDRO    2D
 SMD2      HYDRO    2D
 SMOMR     HYDRO    2D
 SMOMZ     HYDRO    2D
 SMOMZD    HYDRO    2D
 SMOMZV    HYDRO    2D
 SMV       HYDRO    2D
 SMV1      HYDRO    2D
 SMV2      HYDRO    2D
 SPSUMS    HYDRO    3D
 SGMO      KACHINA   C    LOOP          C    SETUP     1C    HYDRO      1C
 SROV1     SETUP    5
 SROV2     SETUP    5
 SS        HYDRO    3
 START     LOOP     1     SETUP     2     HYDRO    22
 STED      HYDRO    3
 STEV      HYDRO    3
 STH       HYDRO    2D
 STIED     HYDRO    4
 STIEV     HYDRO    4
 STKED     HYDRO    4
 STKEV     HYDRO    4
 T         KACHINA   C    LOOP          C    SETUP     3C    HYDRO     21C
 TBASE     HYDRO    3
 TDB       HYDRO    2
 TDC       HYDRO    7
 TDL       HYDRO    2
 TDR       HYDRO    2
 TDT       HYDRO    2
 TERM1B    HYDRO    3
 TERM1L    HYDRO    3
 TERM1R    HYDRO    3
 TERM1T    HYDRO    3D
 TERM2B    HYDRO    3
 TERM2L    HYDRO    3
 TERM2R    HYDRO    3
 TERM2T    HYDRO    3D
 TH        KACHINA   D    SETUP     6D    HYDRO    34D
 THBB      HYDRO    4
 THI       SETUP   10
 THIJ      HYDRO    5
 THIN      KACHINA   C    LOOP          C    SETUP     5C    HYDRO      1C
```

```
THLB      HYDRO     4
THRB      HYDRO     7
THTAB     HYDRO    3D
THTB      HYDRO     7
THTE      HYDRO    20
THTEB     HYDRO     5
THTEC     HYDRO     3
THTEL     HYDRO     5
THTER     HYDRO     5
THTERM    HYDRO     2
THTESQ    HYDRO     3
THTET     HYDRO    5D
THO       KACHINA   C   LOOP      C   SETUP    1C   HYDRO    4C
TLIM      HYDRO     6
TLIMD     KACHINA   C   LOOP      C   SETUP    2C   HYDRO    2C
TOLD      HYDRO    2D
TOP       KACHINA   C   LOOP      C   SETUP    1C   HYDRO    3C
TOUT      KACHINA   C   LOOP      C   SETUP    1C   HYDRO    6C
TVB       HYDRO     2
TVC       HYDRO     7
TVL       HYDRO     2
TVR       HYDRO     2
TVT       HYDRO     2
TWFIN     KACHINA   C   LOOP      C   SETUP    2C   HYDRO    1C
T1        HYDRO     3
T2        HYDRO    8D
T20MD     KACHINA   C   LOOP      C   SETUP    2C   HYDRO    1C
UC        HYDRO     4
UD        KACHINA   D   SETUP     D   HYDRO   32D
UDRL      HYDRO     4
UDRR      HYDRO     4
UDTE      HYDRO     3
UK        HYDRO     2
UR        HYDRO     4
URB       HYDRO     6
US        HYDRO     3
UTB       HYDRO     3
UTLC      HYDRO     5
UTRC      HYDRO     5
UV        KACHINA   D   SETUP     D   HYDRO   57D
UVL       HYDRO     3
UVLB      HYDRO     8
UVR       HYDRO     3
UVRR      HYDRO     8
UVRL      HYDRO     2
UVRR      HYDRO     2
UVTE      HYDRO     3
VBRC      HYDRO     5
VC        HYDRO     4
VCON      SETUP     3
VD        KACHINA   D   SETUP     D   HYDRO   40D
VDTE      HYDRO     3
VECVEL    HYDRO     2
VEL       HYDRO     7
VELMX     HYDRO   17D
VK        HYDRO     2
```

```
    INDEX   01/00/75                              MASTER INDEX                                   PAGE   69
   VOL      KACHINA   C    LOOP       C    SETUP     1C    HYDRO     6C
   VOLR     KACHINA   C    LOOP       C    SETUP     1C    HYDRO     1C
   VRB      HYDRO     4
   VS       HYDRO     3
   VT       HYDRO     4
   VTB      HYDRO     6
   VTRC     HYDRO     5
   VV       KACHINA   D    SETUP     1D    HYDRO    62D
   VVB      HYDRO     3
   VVBB     HYDRO     8
   VVIN     KACHINA   C    LOOP       C    SETUP     3C    HYDRO     2C
   VVT      HYDRO     3
   VVTB     HYDRO     8
   VVTE     HYDRO     3
   WIDTH    SETUP     4
F  WRITE    SETUP    11    HYDRO     20
   WIROW    LOOP      1    SETUP      1
   XCONV    KACHINA   C    LOOP       C    SETUP     1C    HYDRO     5C
   XD       SETUP     5
   XIBR     HYDRO     3
   XIC      HYDRO     6
   XIDB     HYDRO     3
   XIDL     HYDRO     3
   XIDR     HYDRO     3
   XIDT     HYDRO     3
   XIR      HYDRO     3
   XIT      HYDRO     3
   XITL     HYDRO     3
   XITR     HYDRO     6
   XIVB     HYDRO     9
   XIVL     HYDRO     9
   XIVR     HYDRO     9
   XIVT     HYDRO     9
   XL       KACHINA   C    LOOP       C    SETUP     2C    HYDRO     5C
   XNP      SETUP     3
   XP       KACHINA   C    LOOP       C    SETUP     1C    HYDRO     7C
   XR       KACHINA   C    LOOP       C    SETUP     2C    HYDRO      C
   XSPAC    SETUP     3
   XTE      SETUP     4    HYDRO     10
   XX       HYDRO    17
   X1       SETUP     2    HYDRO    11D
   X2       HYDRO     2
   YB       KACHINA   C    LOOP       C    SETUP     3C    HYDRO     5C
   YCONV    KACHINA   C    LOOP       C    SETUP     1C    HYDRO     5C
   YNP      SETUP     2
   YP       KACHINA   C    LOOP       C    SETUP     1C    HYDRO     4C
   YSPAC    SETUP     3
   YT       KACHINA   C    LOOP       C    SETUP     3C    HYDRO      C
   YTE      HYDRO    10
   YY       SETUP     7
   Y1       SETUP     4    HYDRO    11D
   Y2       HYDRO     2
   ZZ       KACHINA   C    LOOP       C    SETUP     1C    HYDRO     1C
```

```
    INDEX  01/00/75                     MASTER INDEX                                   PAGE   70

ROUTINES INDEXED
ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE   ROUTINE PAGE

DONE       5   KACHINA    2   KASET      8   LOOP       5   RPARU      5   RIROW      5   SETUP     10   WIROW      5
HYDRO     25   KACHYDR   23   LCMFLG     5   RIJP2      5   RPARV      5   SETIJ      5   START      5
MASTER INDEX   59


    INDEX  01/00/75                     MASTER INDEX                                   PAGE   71

****************************************END OF COMPUTATION************************************************************************
          1535 CARDS PROCESSED
          3084 MAXIMUM BUFFER USED BY ANY ROUTINE
          2048 TOTAL ECS REQUIRED BY INDEX
         5.868 SECONDS OF CP TIME USED
**********************************************************************************************************************************
```

APPENDIX C

MODIFICATIONS FOR LCM STORAGE OF CELL DATA

The basic SCM version of KACHINA shown in Appendixes A and B has been purposely designed so that both the (1,0) and (2,0) overlays are completely compatible with LCM usage and require no modifications in conversion to LCM storage of cell data. This limits the necessary changes to the following two places in the (0,0) overlay.

(1) Storage Definition. An LCM cell-storage block is defined, and the SCM buffer AASC is shortened and moved out of the main common KSC into the scratchpad common KSB. Thus, the beginning of the storage definition card set reads as follows.

```
LCM/KLC1/AA1(131000)
CØMMØN/KSB/AASC(5814),AARØW(1938)
CØMMØN/KSC/AA(1),AKINFI,AO,BDTØDR,BDTØDZ,
(etc.)
```

The dimension of 5814 words for AASC allows any $\bar{I} \leq 100$ with NQ = 19 words per cell. $\bar{J}$ is limited only by available LCM.

(2) Cell Data Handling. The entire instruction part of SUBRØUTINE LØØP is replaced by the following version.

```
      CALL ECWR (AASC(JW),IECW,NQI,NE)
      IECW = IECW + NQI
      GØ TØ (10, 20, 30) IBUF
   10 JR = 1
      JW = J2
      IJP = JX1
      IJ = JX3
      IJM = JX2
      IBUF = 2
      GØ TØ 40
   20 JR = J2
      JW = J3
      IJP = JX2
      IJ = JX1
      IJM = JX3
      IBUF = 3
      GØ TØ 40
      ENTRY START
      CALL ECRD (AASC,0,NQI2,NE)
      IECR = NQI2
      IECW = 0
   30 JR = J3
      JW = IBUF = 1
      IJP = JX3
      IJ = JX2
      IJM = JX1
   40 CALL ECRD (AASC(JR),IECR,NQI,NE)
      IECR = IECR + NQI
      RETURN
      ENTRY DØNE
      CALL ECWR (AASC(JW),IECW,NQI,NE)
      IECW = IECW + NQI
      GØ TØ (50, 60, 70) IBUF
   50 JW = J2
      GØ TØ 80
   60 JW = J3
      GØ TØ 80
   70 JW = 1
   80 CALL ECWR (AASC(JW),IECW,NQI,NE)
      RETURN
      ENTRY R1RØW
      IEC = (J - 1) * NQI
      CALL ECRD (AASC,IEC,NQI,NE)
      RETURN
      ENTRY SETIJ
      IJ = (I - 1) * NQ + 1
      RETURN
      ENTRY W1RØW
```

```
     CALL ECWR (AASC,IEC,NQI,NE)
     RETURN
     ENTRY LCMFLG
     LCM = 1
     RETURN
     ENTRY RIJP2
     CALL ECRD (AARØW,IECR,NQI,NE)
     RETURN
     ENTRY RPARU
     IEC = (J - 1) * NQI
     IJ = 0
     GØ TØ 300
     ENTRY RPARV
     IEC = (J - 2) * NQI
     IJ = NQI
 300 CALL ECRD (AASC,IEC,NQI2,NE)
     RETURN
     END
```

The SETUP and HYDRØ subroutines contain several peculiarities that are specially treated to make them compatible with either SCM or LCM usage. The required modifications deserve some explanation.

(1) The routine that initially sets all required cell-storage to zeros (DØ loops 189 and 199 in SETUP) is designed to handle either SCM or LCM cell storage automatically without testing.

(2) Note that the SCM version of ENTRY LØØP advances the three row indices IJ, IJP, and IJM over two columns of cells (i = IP2 and i = 1), the assumption being that the I DØ loops normally encompass all interior cells in the row (i = 2 through i = IP1). Several DØ loops in the code, however, have I DØ loops with a lesser or greater range of columns, requiring some increment or decrement of these indices upon RETURN from LØØP, to keep the indexing properly phased.

In the LCM version, though, such adjustments are unnecessary, as here LØØP invariably sets IJ, IJP, and IJM to point to column i = 2 cells. This distinction between the versions is handled automatically because the values of the increments or decrements used (NQL and NQ2L) are set to zero in SETUP in the LCM version. Again, this allows affected DØ loops to be fully general with no required testing of whether SCM or LCM is being used.

In the present version, I DØ loops so treated are those with terminal statement numbers of 189 in SETUP, and 779, 889, 1539, and 3189 in HYDRØ. Remember that a similar treatment may be required if new code is constructed that has I DØ loops with ranges other than the usual i = 2 through i = IP1.

(3) In the calculation of $(\overline{\rho' v})_i^{j+\frac{1}{2}}$ in region 1550 in HYDRØ, reference is made to $(\overline{\rho' v})_i^{j+3/2}$ to obtain a donor cell term for the equation. Because this reference, in turn, involves the use of ρ_i^{j+2}, which is not available in the three rows that have been read from LCM, cell data from row j + 2 must be read in separately. This is the responsibility of ENTRY RIJP2 in LØØP. In the LCM version, it simply reads in row j + 2, whose address is specified by the current setting of the LCM read index, to the one-row buffer AARØW. The statement in region 1550 that calculates $(\rho' v)_i^{j+3/2}$ (named RØVT2B) then references the density $(\rho')_i^{j+2}$ (RØVSPL) from this one-row buffer.

The problem of referencing quantities in rows beyond j + 1 and j - 1 obviously doesn't exist in the pure SCM version, but for compatibility between the two versions, we use ENTRY RIJP2 to place an image of row j + 2 in AARØW. Remember that the same technique and buffer are available to any code addition requiring data lying beyond rows j + 1 or j - 1.

(4) In the particle movement, the area-weighting scheme requires u or v velocities from two adjacent rows of cells. In the LCM version, entries RPARU and RPARV in LØØP obtain the two appropriate rows from LCM, and initialize that part of index IJ that is a function of J.

In the SCM version, these two ENTRY points perform only this latter task, and, again, the procedure is such that the particle mover is not concerned with whether the velocities were obtained from LCM or SCM.

(5) In the Tape Dump and Tape Restart (Regions 250 and 270, respectively, in the HYDRØ Control Region), tests that determine whether LCM data are involved in the dump information are included, so the user need not be concerned with this aspect.

From the above, it can be seen that the philosophy of the LCM package has been to achieve a user-oriented conversion at the expense of some computer efficiency. The additional CP time required for LCM storage of cell data is not an unreasonable increase over the SCM version. For a sample calculation with $\overline{I} = 20$ and $\overline{J} = 45$, the LCM version grind time was about 16% greater than that of the SCM version when running at three iterations per cycle. This percentage decreases as the number of iterations increases, becoming less than 13% at 11 iterations per cycle, for example.

In the present version of KACHINA, there is provision in SCM for storing the coordinates of up to 4000 particles. If necessary, particle storage could be effectively doubled by storing coordinates at a half word each, and/or particle storage could also be moved to LCM.

With storage block AASC dimensioned at 26 676 words, as discussed in Sec. II.C, approximately 3500 words of SCM remain available for code expansion. In the LCM conversion described here, this number increases to nearly 20 000 words, and at the same time the available number of cells increases from 1250 to nearly 10 000.

The SCM KACHINA user who can be sure that his memory requirements are so small that he will never require the LCM compatibility feature can definitely reduce grind time by eliminating SUBRØUTINE LØØP entirely and replacing all CALLS to its ENTRY points by copies of the instructions that the SCM version of LØØP performs. In addition, the special considerations of the five items discussed above can be completely eliminated with very little code modification.

One final note concerning LCM usage: be sure to request LCM on both the \$JØB card (we add the field "LC = 400000") and on the \$LDGØ card (we add the field "LC = 1000000B"). Failure to request LCM will cause an immediate LCM Block Range Error and task abortion.

Note in proof: A "CALL DØNE" should be added permanently, immediately following statement No. 619 in HYDRØ. (Without this CALL, the LCM version will reference erroneous values in row $j = JP1$ when constructing contour plots.)

☆US GOVERNMENT PRINTING OFFICE 1974—677-176/28