AD815263

DASA-1800-IV

DEPARTMENT OF DEFENSE LAND FALLOUT PREDICTION SYSTEM

Volume IV
ATMOSPHERIC TRANSPORT

DASA • NDL • NRDL • ECOM • TECH OPS

316

NOTICE

DASA-1800-IV

DEPARTMENT OF DEFENSE
LAND FALLOUT PREDICTION SYSTEM

Volume IV - Atmospheric Transport

TO-B-66-46
2 Feb[redacted] [redacted]67

[redacted]

T. W. Schwenke, I. Kohlberg, H. G. Norment
[redacted] W. Y. G. Ing

Technical Operations Research
Burlington, Massachusetts

This research has been sponsored by the
Defense Atomic Support Agency
under NWER Subtask A7a/10.058

Submitted to

U.S. Army Nuclear Defense Laboratory
Edgewood Arsenal, Maryland

I/II

ACKNOWLEDGMENTS

The work reported in this volume of DASA-1800 was funded by the Defense Atomic Support Agency under subtask A7a/10.058 through contract DA 28-043-AMC-01309(E) with the U. S. Army Atmospheric Sciences Laboratory. The authors gratefully acknowledge the contributions and support of W. C. Conover and W. Barr of the U. S. Army Atmospheric Sciences Laboratory, R. C. Tompkins and L. M. Hardin of the U. S. Army Nuclear Defense Laboratory, and LCDR J. W. Cane of the Defense Atomic Support Agency.

ABSTRACT

A collection of models developed to simulate atmospheric transport of local fallout from nuclear detonations is described. These models comprise the Transport Module of the Department of Defense Land Fallout Prediction System (acronym DELFIC). Details of the physical bases of the models as well as the Transport Module computer programs are presented. The programs provide for temporal and three-dimensional spatial variation of the wind field. Wind-field construction from input data can be accomplished by one of several preprogrammed methods that may be selected on the basis of the type and quantity of available data. Submodels for special local circulation systems can be superposed on the macrowind system. A capability to simulate highly variable topography is included. The computer programs are essentially open ended with regard to capacity for particle, wind field, and topography data.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

TABLE OF CONTENTS (Cont'd.)

LIST OF TABLES

LIST OF ILLUSTRATIONS

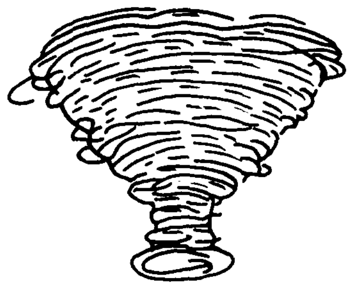
LIST OF ILLUSTRATIONS (Cont'd.)

LIST OF FLOW CHARTS

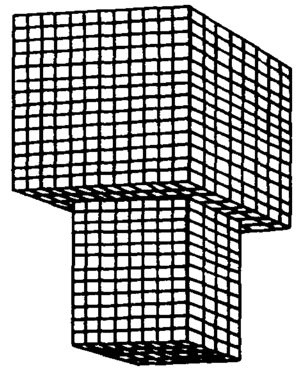
INTRODUCTION

The purpose of the Transport Module is to accept a list of fallout particle properties and positions at the end of the cloud rise and mathematically transport these particles through a temporally and spatially varying wind velocity field until they land on the ground or until the researcher's interests are otherwise satisfied. This module can be characterized by the terms atomistic, deterministic, and discrete. It is atomistic because the basic element of the module calculations is the fallout particle and, at least in concept, the end results of the model are based on the summation of the effects of individual particles. It is deterministic because the trajectories of individual particles falling through the atmosphere are uniquely determined by particle and atmospheric properties. It is discrete since the distributions of particles in space, particle size, and radioactivity are divided into discrete parts, the effects of which are associated with representative central particles. The macroscale atmospheric description used within the Transport Module is also discrete in that the atmospheric volume of interest during a given time period is divided into subvolumes (cells). Everywhere within a cell the atmospheric properties are considered to be uniform. Thus, the Transport Module is discrete in space, time, and particle size.

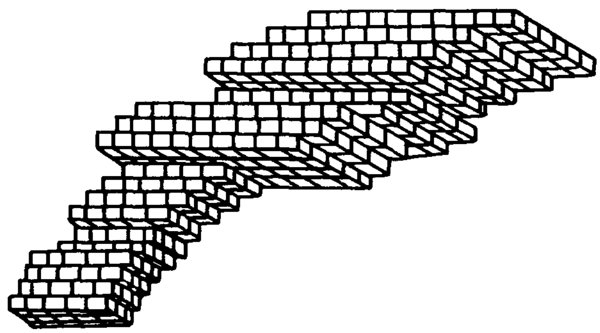
A set of fallout particles chosen as representative of the contents of cloud subdivisions is prepared by the Cloud Rise-Transport Interface program of the Cloud Rise Module. The generation of this input is described in detail in Volume III of this documentation; here we review only its essential highlights. Figure 1(a) depicts the particle cloud resulting from the rise and growth of the nuclear cloud before accounting for wind drift during cloud rise. A region of space that includes the cloud is subdivided, as shown in Figure 1(b), and a particle content is defined for each subdivision. In general, the contents of each cloud subdivision are unique. Each subdivision depicted in Figure 1(b) may be further subdivided into a large number of spatial subdivisions. Furthermore, each of these spatial subdivisions will be represented by a number of different central particles — one for each size class that is actually represented within the original cloud subdivision. Figure 1(c) depicts the location of the subdivisions representing a particular size range after the effect of wind drift during cloud rise has been accounted for.



(a) Accept Particle Sample Resulting from Cloud Rise Module



(b) Load Sample into Array, Smooth the Array to Define all Transportable Cloud Wafers



(c) Adjust Positions of Wafer Centers to Account for Winds During Cloud Rise

Figure 1. Operations of the Cloud Rise - Transport Interface Module

The Transport Module takes as input the coordinates of the center of each subdivision, at which position it assumes residence of a representative central particle of given mass and size. The time of input of the central particle to the Transport Module also is given. A diagrammatic representation of a cloud subdivision and its defining parameters as accepted by the Transport Module are shown in Figure 2. Within the Transport Module the trajectory of each cloud subdivision

PARAMETERS

1. Central Particle Size
2. Mass Per Unit Area
3. X Coordinate (E-W)
4. Y Coordinate (N-S)
5. Z Coordinate (ALT)
6. Time Coordinate

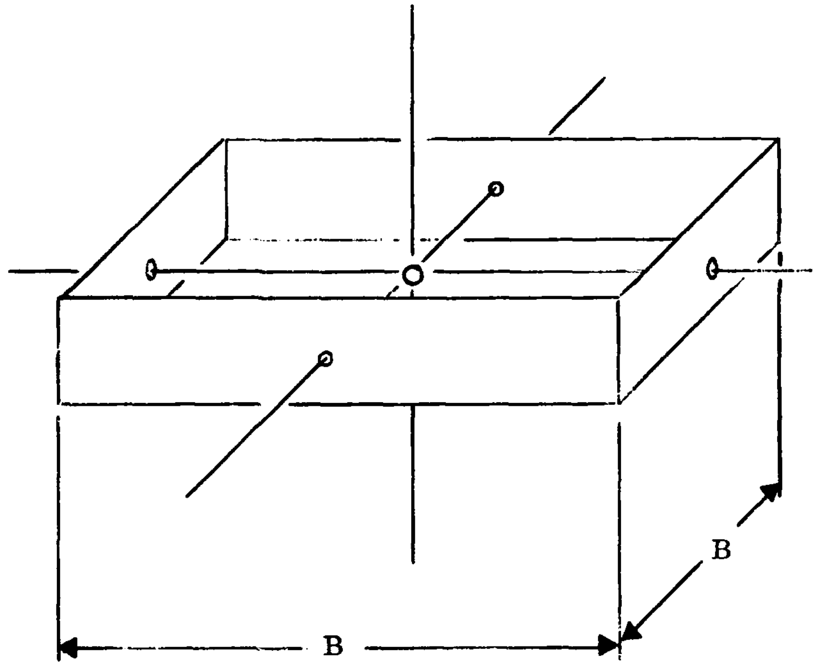


Figure 2. The Elementary Cloud Subdivision and Its Characterization
(B is the dimension of all cloud subdivisions at the time of their definition.)

(represented by its central particle) is determined independently of all others and transport ceases when the central particle lands on the topography.

Within the Transport Module there are two systems for the description of atmospheric flow: the primary, or "macro," system; and the secondary, or "local," system. The use of these systems of description, however, is merely suggestive of but not restricted to the macrometeorological and local meteorological scales. In the macroscale description relatively large cells may be employed, and the totality of cells may include a vast volume of atmosphere perhaps on a macrometeorological scale. In the local atmospheric system cells more freedom is allowed in the mode of circulation description. Within each local circulation system unique particle transport procedures can apply. For practical reasons the DELFIC system restricts the researcher to use, at any one time, only a small number of local circulation systems that are defined within specified boundaries. Where "local" and "macro" description systems overlap, the former take precedence since they are capable of greater precision.

PHYSICAL AND MATHEMATICAL MODELS

Fallout Particle Kinematics

Relationship Between Wind Field and Particle Velocity

The fundamental equations that describe the motion of fallout particles (which are typically greater than 10μ in diameter) in the wind field are the momentum equation

$$\frac{d\underset{\sim}{V}_p}{dt} = -\left\{ \underset{\sim}{V}_p(t) - \underset{\sim}{V}_w[\underset{\sim}{r}(t), t]\right\} \phi\left(|\underset{\sim}{V}_p - \underset{\sim}{V}_w|\right) + \underset{\sim}{G} \tag{1}$$

and the displacement equation

$$\frac{d\underset{\sim}{r}}{dt} = \underset{\sim}{V}_p \quad , \tag{2}$$

where $\underset{\sim}{V}_p$ and $\underset{\sim}{V}_w$ are the particle and wind velocity respectively, $\underset{\sim}{G} = -G\underset{\sim}{k}$ where G is the gravitational constant and $\underset{\sim}{k}$ is a unit vector which points in the positive z direction, $\underset{\sim}{r}$ is the particle's position, and $\phi(|\underset{\sim}{V}_p - \underset{\sim}{V}_w|)$ is a friction function defined so that the frictional force per unit mass between the particle and the wind is given by*

$$F = -\left(\underset{\sim}{V}_p - \underset{\sim}{V}_w\right) \phi\left(|\underset{\sim}{V}_p - \underset{\sim}{V}_w|\right) \quad . \tag{3}$$

---

*A commonly used expression for ϕ in the pressure flow regime is

$$\phi = \frac{1}{2}\,\frac{C_D}{m}\,\rho A\,|\underset{\sim}{V}_p - \underset{\sim}{V}_w| = K|\underset{\sim}{V}_p - \underset{\sim}{V}_w| \quad ,$$

while in the Stokes law regime ϕ is a constant.

We have shown in Appendix A of Ref. 1 that for all but the most extreme conditions of airflow, for example, tornadoes, the components of particle velocity are given by

$$V_{px} = U \ , \tag{4}$$

$$V_{py} = V \ , \tag{5}$$

and

$$V_{pz} = -V_F + W \ , \tag{6}$$

where U, V, and W are the x, y, and z components of the wind velocity, respectively, and V_F is the still-air particle settling rate. In effect we have been able to solve the momentum equation for the fallout particle, thus reducing the dynamics of the transport problem to the solution of the position equation.

Particle Settling Rates

We have performed a comprehensive survey of the methods used for computing particle settling rates as given both in the open literature and in the literature on fallout prediction methods.[1] On the basis of this survey, we have concluded that the equations of Davies[2] for spheres are most appropriate for use in the DOD Land Fallout Prediction System. The following procedure is used in computing particle settling rates:

1. The dimensionless quantity $C_D R^2$, where C_D is the drag coefficient and R is the Reynolds number, is evaluated by the equation

$$C_D R^2 = \frac{4G\rho\rho_p d^3}{3\eta^2} \ , \tag{7}$$

where G is the acceleration of gravity, ρ and ρ_p are the densities of air and particle, d is the particle diameter, and η is the dynamic viscosity of the air.

2. The Reynolds number is evaluated from the Davies polynomials:

$$R = \frac{C_D R^2}{24} - 2.3363 \times 10^{-4}\left(C_D R^2\right)^2 + 2.0154 \times 10^{-6}\left(C_D R^2\right)^3 - 6.9105 \times 10^{-9}\left(C_D R^2\right)^4, \quad C_D R^2 < 140 \tag{8}$$

or

$$\log_{10} R = -1.29536 + 0.986\left(\log_{10} C_D R^2\right) - 0.046677\left(\log_{10} C_D R^2\right)^2 + 0.0011235\left(\log_{10} C_D R^2\right)^3, \quad 100 < C_D R^2 < 4.5 \times 10^7 . \tag{9}$$

3. The settling velocity V_F is computed from

$$V_F = \frac{R\eta}{\rho d} . \tag{10}$$

4. For small particles at high altitudes, the settling velocity must be multiplied by a drag slip correction, f, where

$$f = 1 + \frac{2.33 \times 10^{-4}}{d\rho} , \tag{11}$$

and d and ρ are in microns and grams per cubic centimeter, respectively.

We have concluded[1] that methods commonly used in the past to correct particle fall rates for shape effects in fallout prediction calculations are incorrect. Apparently it is true that irregularity of shape can have a significant influence on settling rate; however, the only precise information of a general nature that seems to be available is that a particle of spherical shape falls at a rate that is a maximum for particles of equivalent volume of all shapes. In addition, irregularity of shape can cause deviation of particle trajectories from the vertical in still air. It is known that both of these effects become more pronounced with increase in Reynolds number. Unfortunately, so little experimental work has been done for particles in the pressure flow range (i.e., for large Reynolds numbers) that the importance of these effects to fallout prediction calculations cannot be precisely determined. Additional studies of these effects should be performed to resolve the issue.

Appendix B of Ref. 1 presents the details of our study and a comparison of particle settling rate computation methods.

Effect of Atmospheric Diffusion on Particle Transport

In our model of cloud subdivision transport a segment of cloud volume of height ΔZ and lateral dimensions $2X_o$, $2Y_o$ (see Figure 3) is assumed to move

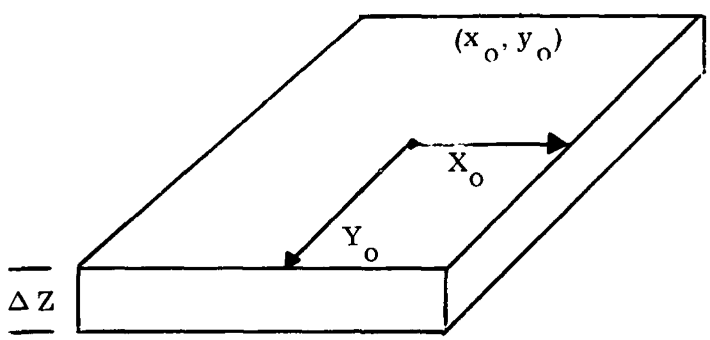


Figure 3. Segment of Cloud Volume

through the atmosphere as a rigid body if turbulent diffusion is absent. To be sure, it is assumed (still neglecting diffusion) that the initial extent of the cloud subdivision is small enough so that the equation of motion of a hypothetical particle located at the periphery will not differ from that at the center. The motion of the center is determined from the conventional transport equations as previously developed (i.e., Eqs. (4)-(6)).

In reality, the cloud subdivision represents a group of particles (of a particular size range) whose total number is N and whose initial uniform lateral density* is

$$\sigma_o = \frac{N}{4Y_oX_o} \left(\text{particles - m}^{-2}\right) \quad . \tag{12}$$

During transport, turbulent diffusion tends to disperse the particles of the cloud subdivision so that by the time the subdivision reaches the ground, its shape will have changed and its particle density, σ, will have decreased and become nonuniform.

The increase in lateral area is due to the cumulative effect of diffusion of all the particles contained in the slice. If the origin is established at the center of the slice, the lateral density of particles, $P(x, y, t)$, at a time t is given by

$$P(x, y, t) = \sigma_o \int_{-X_o}^{+X_o} \int_{-Y_o}^{+Y_o} G(x - x', y - y', t)\, dx'\, dy' \quad , \tag{13}$$

where the diffusion kernel $G(x - x', y - y', t)$ is given by

$$G = (2\pi Dt)^{-1} \exp\left\{-\left[(x - x')^2 + (y - y')^2\right]/2Dt\right\} \quad , \tag{14}$$

with D being the diffusion constant. Consideration of Eqs. (13) and (14) show that $P(x, y, t)$ is defined over the entire x, y plane, but as an approximation to the theoretical result for computational purposes we have chosen to construct an equivalent rectangular segment of uniform surface density σ with dimensions defined as X, Y. These equivalent dimensions are determined by requiring that the mean-square

*The term lateral density is used to refer to the surface density (particle/unit area) that would result if the particles represented by a cloud subdivision were deposited vertically onto a horizontal plane.

displacements $\overline{x^2}$, $\overline{y^2}$ of the rectangular segment be the same as those computed from the exact probability distribution P(x, y, t).

It is easy to show that for a uniform distribution, $\overline{x^2}$ and $\overline{y^2}$ are related to the limiting dimensions via the formulas:

$$\overline{x^2} = 1/3\, X^2 ; \; \overline{y^2} = 1/3\, Y^2 \quad . \tag{15}$$

On the other hand, we have

$$\overline{x^2} = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^2 P(x,y,t)\, dx\, dy}{4\sigma_o X_o Y_o} = Dt + 1/3\, X_o^2 \; ,$$

and (16)

$$\overline{y^2} = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} y^2 P(x,y,t)\, dx\, dy}{4\sigma_o X_o Y_o} = Dt + 1/3\, Y_o^2 \; .$$

The equivalent dimensions of the slice at time t are thus given by

$$X^2 = X_o^2 + 3Dt \; ,$$

and (17)

$$Y^2 = Y_o^2 + 3Dt \; ,$$

with corresponding lateral density

$$\sigma = \frac{N}{4XY} \quad . \tag{18}$$

The user is referred to Pasquill's "Atmospheric Diffusion"[3] for a discussion on reasonable estimates of D.

Wind-Field Description

As previously mentioned, there are two complementary and simultaneously compatible modes for describing the wind field: (1) the macrowind description system which makes use of a numerical approximation to a complete three dimensional wind field derived from observed data and is of greatest general utility; and (2) the local circulation description system which makes use of analytical representations of special atmospheric situations (e.g., sea breezes or mountain winds). These local systems also are three dimensional and can coexist with a macrowind field, in which case they override the macrowind field within the volume of space common to both. These modes of wind-field description are described in detail in the subsequent sections.

Macrowind Fields

The macrowind-field descriptions are accomplished as follows. A Cartesian coordinate system that encompasses the region of close-in fallout is established with arbitrary origin. With reference to this coordinate system, grid square arrays are specified on horizontal planes at arbitrarily spaced intervals in the vertical direction. Figure 4 illustrates how such a set of strata is used to fill the volume of atmosphere of interest. Each stratum is further subdivided into a number of wind cells in a regular manner as is shown in Figure 5.

To assign vectors to wind cells, the user must first specify as input a data set of wind vectors and vector positions. This data set can be arbitrary in number and distributed in an arbitrary manner throughout the atmospheric volume of interest. The program then determines and associates a wind vector with each wind cell in the volume of interest. These wind cell vectors are based on the input data, and there are three interpolation-extrapolation computational methods available for use in determining them.

In the first option the program assigns to each wind cell the data vector nearest the cell's center. The second option uses a weighted average of nearest data vectors, where the user is free to specify both the number and the distances of the vectors to be considered. The third option uses a statistically derived three dimensional linear model of the atmosphere based on the N nearest data vectors to perform the required interpolation or extrapolation for each cell. The method to be used in any particular

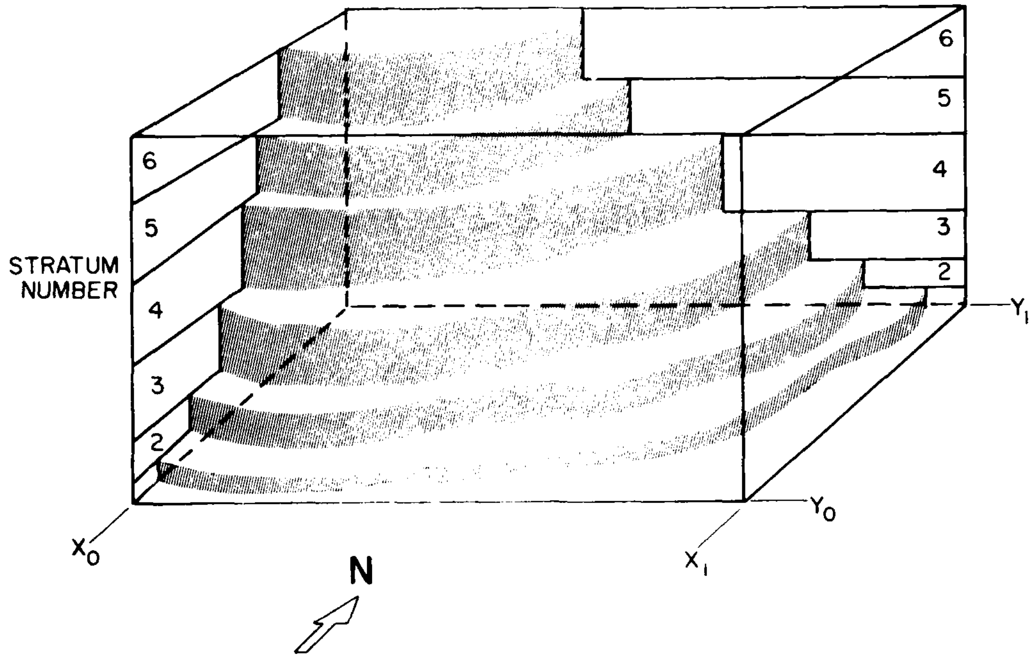


Figure 4. Strata within the Specified Wind Field Volume (illustrated for six strata)

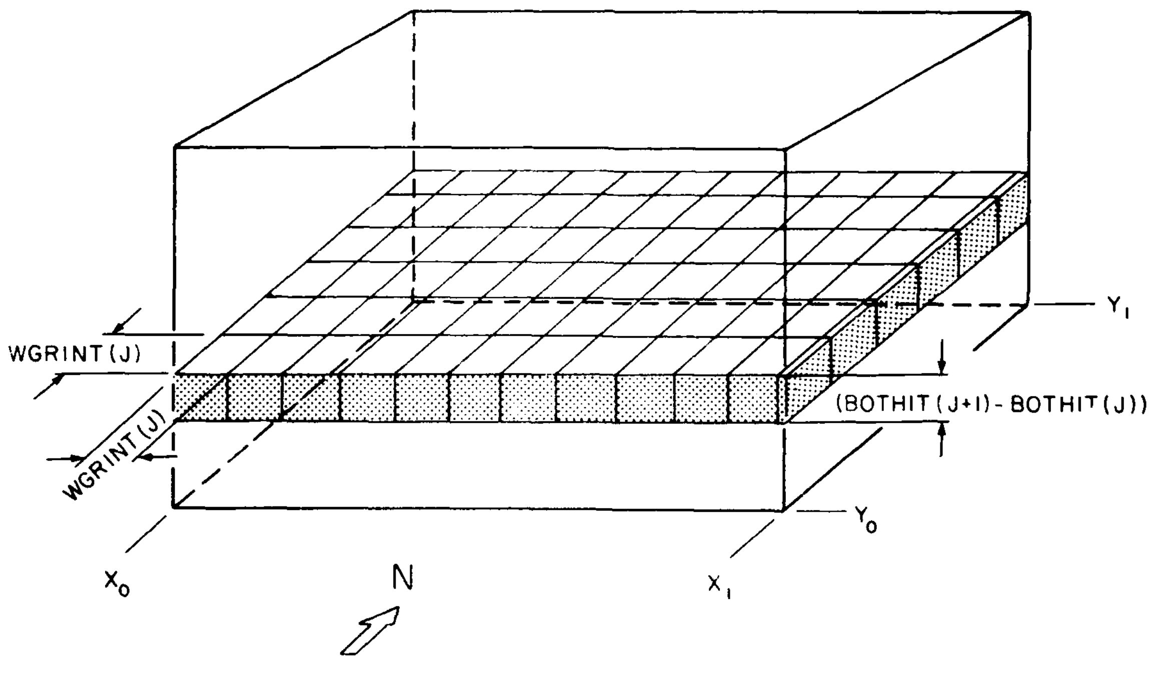


Figure 5. Wind Cells — Subdivisions of a Stratum (illustrated for the Jth stratum from the bottom)

case must be determined on the basis of the quantity and quality of the data available. The notation used in the explanation of the three methods is as follows:

$\underset{\sim}{R}_i$ = position of ith observed wind velocity vector relative to the wind-field-array grid point $\underset{\sim}{R}_o$

$\underset{\sim}{V}_i$ = measured wind velocity at position $\underset{\sim}{R}_i$

$\underset{\sim}{V}_o$ = wind velocity at a wind-field-array grid point $\underset{\sim}{R}_o$. $\underset{\sim}{V}_o$ is to be determined from $\underset{\sim}{R}_i$ and $\underset{\sim}{V}_i$.

<u>The Closest Datum Method</u>. In this method the velocity at the grid point is assumed to be the same as that of the closest datum point. This will probably be a good approximation if the location of a measurement is sufficiently close to the arbitrary point.

<u>The Preferential-Weighting Method.</u> In the preferential weighting method $\underset{\sim}{V}_o$ is computed as a weighted average of the velocities from observations that lie within distance β from the grid point in the horizontal plane and distance α from the grid point in the vertical direction. Specifically, the relationship between $\underset{\sim}{V}_o$ and $\underset{\sim}{V}_i$ is given by

$$\underset{\sim}{V}_o = \sum_{i=1}^{N} f_i \, \underset{\sim}{V}_i \qquad (19)$$

where

$$\sum_{i=1}^{N} f_i = 1 \quad . \qquad (20)$$

A weighting method described by Cressman[4] has been used in deriving an expression for f_i in the form

$$f_i = \frac{\left(\dfrac{\alpha^2 - z_i^2}{\alpha^2 + z_i^2}\right)\left(\dfrac{\beta^2 - x_i^2 - y_i^2}{\beta^2 + x_i^2 + y_i^2}\right)}{\left(\displaystyle\sum_{k=1}^{N} \dfrac{\alpha^2 - z_k^2}{\alpha^2 + z_k^2}\right)\left(\dfrac{\beta^2 - x_k^2 - y_k^2}{\beta^2 + z_k^2 + y_k^2}\right)} \quad . \qquad (21)$$

The parameters α, β, and N are specified by the user, α and β have the physical significances described previously. The calculations of the f_i are performed so that whenever a factor in Eq. (21) is found to be negative, its value is replaced with zero. If N is specified to be less than the total number of observations, only the N observations closest to the grid point are considered in the calculations.

The Least-Squares Method. Here, we assume that each velocity component is an analytic function of position. Since the wind velocity in the macrowind field will not undergo very great spatial variations in a short distance, it becomes possible to approximate each component of the wind velocity by the first few terms of the Taylor expansion taken about the grid point as origin. We can then write

$$u = u_o + (\nabla u)_o \cdot \underset{\sim}{R} ,$$

$$v = v_o + (\nabla v)_o \cdot \underset{\sim}{R} , \tag{22}$$

and

$$w = w_o + (\nabla w)_o \cdot \underset{\sim}{R} ,$$

where u_o, v_o, and w_o are the x, y, and z components of the wind velocity at the origin. By least-squares fitting of Eq. (22) to the data points, we can determine the twelve unknown constants u_o, v_o, w_o, $(\nabla u)_o \equiv \underset{\sim}{A}$, $(\nabla v)_o \equiv \underset{\sim}{B}$, and $(\nabla w)_o \equiv \underset{\sim}{C}$. Actually, the computation breaks down into three separate parts involving $(u_o, \underset{\sim}{A})$, $(v_o, \underset{\sim}{B})$, and $(w_o, \underset{\sim}{C})$. To illustrate the procedure, we shall outline the method for computing u_o. If U_i denotes the x component of wind velocity at the ith sounding station, the ith residual is given by

$$\xi_i = U_i - u_i = U_i - \left(u_o + A_x x_i + A_y y_i + A_z z_i \right) \quad . \tag{23}$$

The constants u_o, A_x, A_y, and A_z are determined by the least-squares method by minimizing the functional

$$F\left(u_o, \underset{\sim}{A}\right) = \sum_{i=1}^{N} \xi_i^2 \tag{24}$$

with respect to these four parameters. The four linear equations so deduced are

$$\frac{\partial F}{\partial u_o} = 0 = -\sum U_i + \sum \left(u_o + \underset{\sim}{A} \cdot \underset{\sim}{R}_i\right) , \tag{25}$$

$$\frac{\partial F}{\partial A_x} = 0 = -\sum U_i x_i + \sum \left(u_o + \underset{\sim}{A} \cdot \underset{\sim}{R}_i\right) x_i , \tag{26}$$

$$\frac{\partial F}{\partial A_y} = 0 = -\sum U_i y_i + \sum \left(u_o + \underset{\sim}{A} \cdot \underset{\sim}{R}_i\right) y_i , \tag{27}$$

and

$$\frac{\partial F}{\partial A_z} = 0 = -\sum U_i z_i + \sum \left(u_o + \underset{\sim}{A} \cdot \underset{\sim}{R}_i\right) z_i . \tag{28}$$

Introducing the averaged quantities,

$$\bar{u} = \left(\frac{1}{N}\right) \sum U_i, \ \bar{x} = \left(\frac{1}{N}\right) \sum x_i, \ \bar{y} = \left(\frac{1}{N}\right) \sum y_i ,$$

$$\bar{z} = \left(\frac{1}{N}\right) \sum z_i, \ \overline{ux} = \left(\frac{1}{N}\right) \sum U_i x_i, \ \overline{uy} = \left(\frac{1}{N}\right) \sum U_i y_i ,$$

$$\overline{uz} = \left(\frac{1}{N}\right) \sum U_i z_i, \ \overline{x^2} = \left(\frac{1}{N}\right) \sum x_i x_i, \ \overline{xy} = \left(\frac{1}{N}\right) \sum x_i y_i , \tag{29}$$

$$\overline{xz} = \left(\frac{1}{N}\right) \sum x_i z_i, \ \overline{yz} = \left(\frac{1}{N}\right) \sum y_i z_i, \ \overline{y^2} = \left(\frac{1}{N}\right) \sum y_i y_i ,$$

and

$$\overline{z^2} = \left(\frac{1}{N}\right) \sum z_i z_i ,$$

gives the following matrix equation for u_o and $\underset{\sim}{A}$:

$$\begin{pmatrix} 1 & \bar{x} & \bar{y} & \bar{z} \\ \bar{x} & \overline{x^2} & \overline{xy} & \overline{xz} \\ \bar{y} & \overline{xy} & \overline{y^2} & \overline{yz} \\ \bar{z} & \overline{xz} & \overline{yz} & \overline{z^2} \end{pmatrix} \begin{pmatrix} u_o \\ A_x \\ A_y \\ A_z \end{pmatrix} = \begin{pmatrix} \bar{u} \\ \overline{ux} \\ \overline{uy} \\ \overline{uz} \end{pmatrix} \quad . \tag{30}$$

By use of conventional matrix inversion techniques, Eq. (30) can be solved for u_o. We have

$$u_o = \gamma_i \bar{u} + \gamma_2 \overline{ux} + \gamma_3 \overline{uy} + \gamma_4 \overline{uz} \quad , \tag{31}$$

where

$$\gamma_1 = \frac{B^{11}}{|B|} \quad ,$$

$$\gamma_2 = \frac{B^{21}}{|B|} \quad ,$$

$$\gamma_3 = \frac{B^{31}}{|B|} \quad , \tag{32}$$

and

$$\gamma_4 = \frac{B^{41}}{|B|} \quad ,$$

in which $|B|$ denotes the determinant of the matrix Eq. (30). The quantities B^{ki} are the cofactors which equal $(-1)^{i+k}$ times the complementary minor of the matrix element B_{ki}. It is easy to show that the y and z components of velocity are given by

$$v_o = \gamma_1 \bar{v} + \gamma_2 \overline{vx} + \gamma_3 \overline{vy} + \gamma_4 \overline{vz} \quad , \tag{33}$$

and

$$w_o = \gamma_1 \bar{w} + \gamma_2 \overline{wx} + \gamma_3 \overline{wy} + \gamma_4 \overline{wz} \quad , \tag{34}$$

where the averaged quantities in Eqs. (33) and (34) are of the same nature as those shown in Eq. (29) with the replacement of U_i with V_i and W_i.

Some reflection shows that the determinant of the matrix can equal zero when the measured points lie on the same plane or on a line. (For example: if $z = z^*$ is the same for all stations, then the fourth column of B is z^* times the first and $|B|$ vanishes.) This is a manifestation of the impossibility of passing a different plane through the N points. We have provided for these degenerate cases in the computer program. When the determinant of B is very small, we revert back to the preferential-weighting method.

<u>Local Circulation Systems</u>

Provision has been made to incorporate local circulation systems in the computer program to afford prediction of the wind velocity in regions where (1) direct measurements of the wind velocity are not readily available and (2) the density of measuring stations is not adequate to account for rapid spatial changes in the wind field. At present, two such local circulation systems are available: the orographic and sea-breeze systems.

The regions controlled by these models are bounded by planes perpendicular to the coordinate axes. Inside these regions, wind vectors are computed for specific circulation model parameters. Figure 6 represents three of these local circulation cells as they may be superimposed upon the macrostratum and wind cell structure. The important physical features of these local circulation systems, as they pertain to user application, are now discussed, although the details of the theory in each case are presented in Appendixes A and B, respectively.

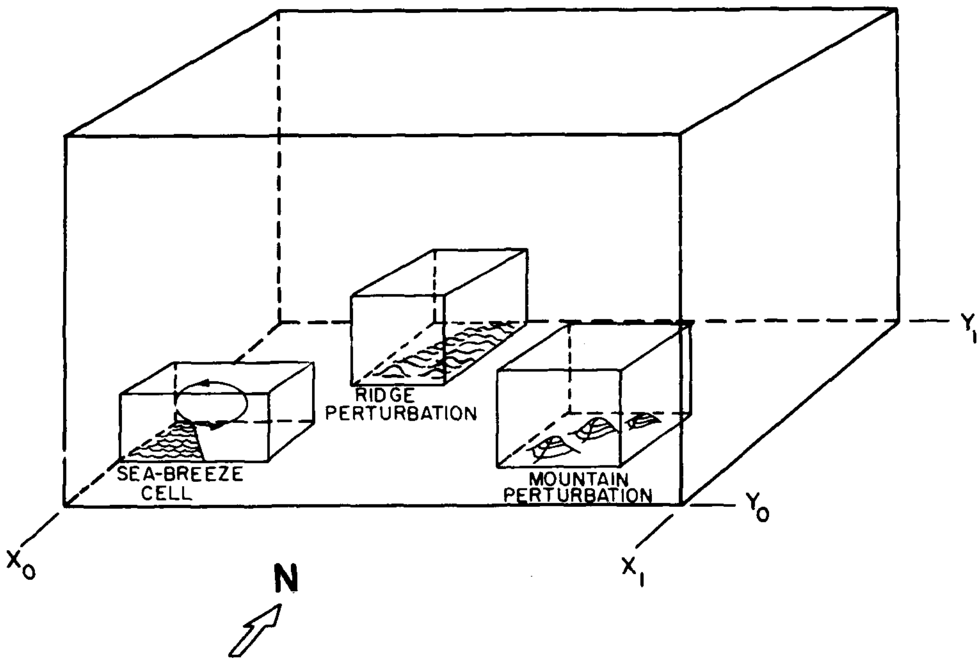


Figure 6. Wind-Field Volume with Superimposed Local Circulation System Cells

Orographic Effects. The theoretical model of orographic flow is intended for use in regions where suitable meteorological data are not readily available. Specifically, the model assumes that in the absence of the variable terrain region under consideration, a certain uniform steady velocity field would exist. The mountains and valleys then cause the assumed flow to change, and it is the resulting wind field which is computed by the model. It is possible to compute the wind field in a region which contains several orographic features by first computing the wind field due to a single one and then summing up the effects. This procedure works as follows:

Let u_o be the velocity of the unperturbed flow (i.e. the flow that would exist in the absence of the mountains and valleys). Now orient the coordinate system so that the x direction points along u_o, and let the y axis be perpendicular to u_o and the z axis point in the direction of the zenith. The functions $u(x, y, z)$, $v(x, y, z)$, and $w(x, y, z)$ denote the x, y, and z components of the wind velocity respectively.

We have found that a suitable mathematical representation for a single mountain is

$$z = f(x, y) = \frac{h a^3}{\left(a^2 + r^2\right)^{3/2}} \quad , \tag{35}$$

where z is the elevation of the mountain, expressed as a function of

$$r = \left(x^2 + y^2\right)^{1/2} \quad , \tag{36}$$

the horizontal distance from the center of the mountain; h is the maximum elevation of the mountain as can be seen by setting $r = 0$ in Eq. (35); and a is a characteristic width of the mountain (when $r = a$ the elevation $z = 0.35h$). The components of wind velocity resulting from the mountain whose vertical position with distance is given by Eq. (35) is given by:

$$u(x, y, z) = u_o \left[1 + \left(a^2 h\right) \frac{\left(y^2 + \lambda^2 - 2x^2\right)}{\left(r^2 + \lambda^2\right)^{5/2}} \right] \quad , \tag{37}$$

$$v(x, y, z) = -3u_o \left(a^2 h\right) \frac{xy}{\left(r^2 + \lambda^2\right)^{5/2}} \quad , \tag{38}$$

and

$$w(x, y, z) = -3u_o \left(a^2 h\right) \frac{\lambda x}{\left(r^2 + \lambda^2\right)^{5/2}} \quad , \tag{39}$$

where

$$\lambda = (z + a) \quad . \tag{40}$$

Obviously, the foregoing expressions for the components of wind velocity are applicable for

$$z \geq f(x, y) \tag{41}$$

(i.e. for those points which lie above the ground). Equations (37)-(39)can be used to describe the flow of wind over a valley whose mathematical description is like that of an inverted mountain. For this situation we merely replace h by -h, the maximum depression of the mountain.

Another important obstacle to be considered is a mountain ridge whose crestline makes an arbitrary angle γ with respect to the direction of the unperturbed flow u_o. The pertinent geometric details are shown in Figure 7.

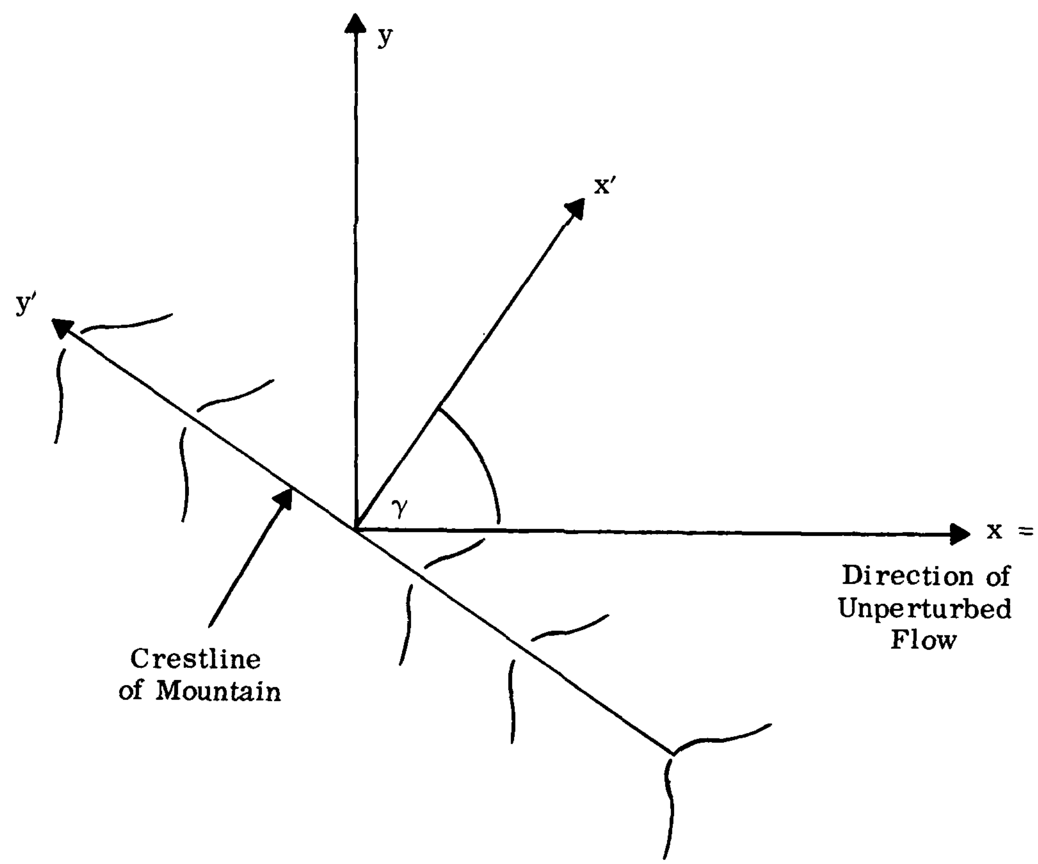


Figure 7. Mountain Ridge Not Perpendicular to Flow

The mathematical description of the elevation of the mountain ridge when viewed along the y' axis is given by the expression

$$z^* = \frac{h}{1 + (x'/a)^2} \quad , \tag{42}$$

where h is the maximum elevation of the ridge; a, in this case, is the half width ($z = 0.5h$ when $x' = a$); and the x and y coordinates are related to x' and y' by the equations

$$\begin{aligned} x &= x' \cos\gamma - y' \sin\gamma \,, \qquad & x' &= x\cos\gamma + y\sin\gamma \;; \\ y &= x' \sin\gamma + y' \cos\gamma \,, \qquad & y' &= -x\sin\gamma + y\cos\gamma \;. \end{aligned} \tag{43}$$

The wind velocity components referred to the x, y, z coordinates are given by

$$u = u_o - u_o(ah)\cos^2\gamma \, \frac{(x\cos\gamma + y\sin\gamma)^2 - \lambda^2}{\left[(x\cos\gamma + y\sin\gamma)^2 + \lambda^2\right]^2} \quad , \tag{44}$$

$$v = -u_o(ah)\cos\gamma\sin\gamma \, \frac{(x\cos\gamma + y\sin\gamma)^2 - \lambda^2}{\left[(x\cos\gamma + y\sin\gamma)^2 + \lambda^2\right]^2} \quad , \tag{45}$$

and

$$w = -2u_o(ah)\,\lambda\cos\gamma \, \frac{(x\cos\gamma + y\sin\gamma)}{\left[(x\cos\gamma + y\sin\gamma)^2 + \lambda^2\right]^2} \quad , \tag{46}$$

where

$$\lambda = z + a \quad .$$

It should be carefully noted that u, v, and w do <u>not</u> depend on y', as can be seen from the substitution $x' = x \cos \gamma + y \sin \gamma$ in Eqs. (44)-(46), so that the origin of the mountain ridge can be located anywhere along the crestline. Equations (44)-(46) can also be applied to a valley ridge whose shape is that of an inverted mountain ridge, with the replacement of h by -h.

In summary then, we can compute the wind field due a mountain, inverted mountain (valley), mountain ridge, and inverted mountain ridge (valley ridge). For the single mountain (valley) the expressions for the velocity are referred to the center of the mountain whose coordinates can be denoted by

$$\left(x_i, y_i\right) .$$

That is, if x, y and z denote the point in question, then the components of the wind field due to the mountain in question that are computed at this point are given by

$$u_i(x, y, z) = u\left(x - x_i, y - y_i, z\right) ,$$

$$v_i(x, y, z) = v\left(x - x_i, y - y_i, z\right) , \tag{47}$$

and

$$w_i(x, y, z) = w\left(x - x_i, y - y_i, z\right) ,$$

where $u(x - x_i, y - y_i, z)$, $v(x - x_i, y - y_i, z)$, and $w(x - x_i, y - y_i, z)$ are obtained from Eqs. (37)-(39) with the replacement of x by $x - x_i$, and y by $y - y_i$. As in Eq. (41), the inequality

$$z \geq z_i^* = f_i\left(x - x_i, y - y_i\right) \tag{48}$$

must also be satisfied.

Precisely the same considerations concerning the calculation of the wind field apply for the mountain (valley) ridge. That is, Eqs. (44)-(46) give the velocity of the wind field when x and y are replaced by $x - x_i$ and $y - y_i$, where x_i and y_i are the coordinates of the center of the ridge and z lies above the ground.

As demonstrated in Appendix A, the theory shows that the principle of superposition of ground disturbances is applicable under most conditions, the exceptions to which are subsequently discussed. What this means is that in a region where the topography can be described by the equation

$$z_T^* = \sum_i f_i\left(x - x_i, y - y_i\right) , \tag{49}$$

where $f_i(x - x_i, y - y_i)$ is the mathematical description of a particular orographic feature (referred to a suitable origin whose coordinates are x_i, y_i), the resulting velocity field can be written as

$$u(x, y, z) = \sum_i u_i\left(x - x_i, y - y_i, z\right) ,$$

$$v(x, y, z) = \sum_i v_i\left(x - x_i, y - y_i, z\right) , \tag{50}$$

and

$$w(x, y, z) = \sum_i w_i\left(x - x_i, y - y_i, z\right) ,$$

where $u_i(x - x_i, y - y_i, z)$, $v_i(x - x_i, y - y_i, z)$, and $w_i(x - x_i, y - y_i, z)$ are the contributions to the velocity field resulting from the orographic feature whose mathematical description is given by $f_i(x - x_i, y - y_i)$. To be sure, we have assumed in this model that the topographical description can be resolved into combinations of mountains, valleys, and mountain and valley ridges whose individual mathematical description is given by Eqs. (35) or (42) with h either positive or negative. In the event that this is not feasible, or satisfactory, the user can use the general technique as outlined in Appendix A to compute the wind field for an arbitrary topographical description.

Thus at this time the user is obliged to represent the topography through combinations of the four features just discussed. The point to be carefully noted is that the resulting analytical expression for the topography, which will be of form given by Eq. (49), should as closely as possible resemble the terrain. Suppose there are two mountain ridges each of half width a separated by a distance ℓ_1, as shown in Figure 8(a).

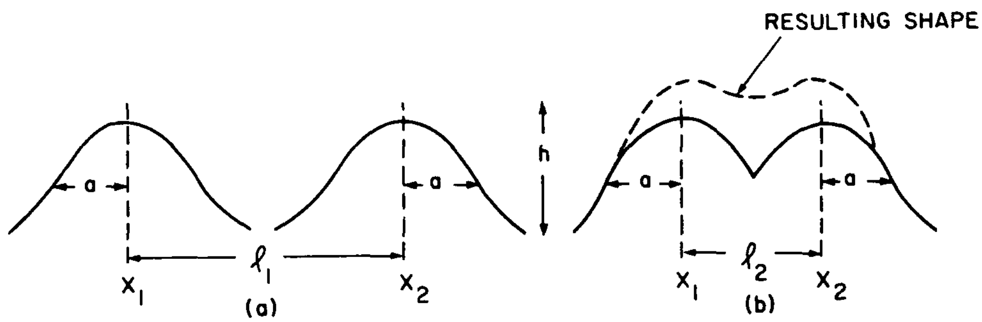


Figure 8. Mountain Ridge Separations

If ℓ_1 is large compared to a, then with good approximation the topography can be represented by the equation

$$z_T^* = \frac{h}{1 + \left(x - x_1\right)^2 / a^2} + \frac{h}{1 + \left(x - x_2\right)^2 / a^2} , \tag{51}$$

because when z_T^* is evaluated in the vicinity of the second mountain ridge (i.e., $x \approx x_2$) the contribution from the first term is negligible. Evaluating z_T^* at $x = x_2$ gives

$$z_T^*\left(x = x_2\right) = \frac{h}{1 + \left(\ell_1^2 / a^2\right)} + h ,$$

which if $\ell_1 >> a$ approximately equals h, the contribution from the second ridge only. Now consider the same ridges, but this time separated by a smaller distance ℓ_2 (Figure 8(b)). Equation (51) will no longer be adequate because $z_T^*(x = x_2)$ becomes

$$z_T^*\left(x = x_2\right) = \frac{h}{1 + \left(\ell_2^2 / a^2\right)} + h \ ,$$

which can be significantly greater than h if ℓ_2 is not very much larger than a. Thus the dashed line shown in Figure 8(b) might be the resulting topographical shape if Eq. (51) were used. A possible method for circumventing problems of this type is to use an expression of the form

$$z_T^* = \frac{h'}{1 + \left(x - x_1\right)^2 / a'^2} + \frac{h'}{1 + \left(x - x_2\right)^2 / a'^2} \ ,$$

where h' and a' are "adjusted" parameters, deduced by developing a best fit approximation to the actual terrain.

In brief, the resulting analytic expression for the topography should be deduced by a "best fit" procedure.

As mentioned earlier, there are certain limitations of the model which the user should be aware of. These restrictions are basically of two types and are concerned with the extent or actual size of the orographic flow of the local circulation system, and the shape of the terrain. These aspects of the problem are discussed in detail in Appendix A; however, a summary of the major conclusions is as follows:

1. <u>Size Limitations</u>

The theoretical model is based upon a perturbation treatment of the usual hydrodynamic-thermodynamic equations under the assumption that an adiabatic atmosphere prevails. The relationship between the change in the wind field $\Delta \underset{\sim}{v}(x, y, z)$ and the curvature of the terrain is deduced by first expressing the three components of $\Delta \underset{\sim}{v}$ (namely $\Delta v_i(x, y, z)$) in a spatial Fourier transform representation,

$$\Delta v_i(x, y, z) = \int A_i(\underset{\sim}{k})\, e^{i\underset{\sim}{k} \cdot \underset{\sim}{r}}\, d^3k \quad ,$$

and then solving for the $A_i(\underset{\sim}{k})$. The solution for the $A_i(\underset{\sim}{k})$ involves the derivation of the dispersion relationship for the system, which basically connects the vertical attenuation constant of the velocity field to the periodicity of the terrain. This relationship is of the form

$$k_z = k_z\left(k_x, k_y\right) \quad ,$$

and becomes greatly simplified for (1) short wavelengths and (2) when the Coriolis effect is neglected. It is in fact these simplifications of the dispersion relationship which yield the relatively simple forms of the wind fields. The short wavelength restriction requires that the area designated as a local circulation region be no greater than 50 mi in one direction. On the other hand, the neglect of the Coriolis effect requires that the extent of the local circulation system, L, be no greater than

$$d = 24\, u_{om} \quad , \tag{52}$$

where u_{om} is the unperturbed wind velocity expressed in miles per hour. The condition for which

$$L < d = 24\, u_{om}$$

is not really a limitation on the applicability of the theory for fallout prediction. If u_{om} is small, the perturbed wind velocity will also be small (as shown in the analysis) and terrain effects will not be important since the motion of the fallout particle will be essentially vertical. Thus, the expressions derived for the wind field by applying the calculation for short horizontal wavelengths and neglecting the Coriolis effect are entirely justified from the local circulation viewpoint. For all practical purposes the requirement

$$L \leq 50 \text{ mi}$$

is sufficient.

2. Shape Limitations

The first-order perturbation theory solution is only approximate and gives increasingly better results as the change in velocity, $\Delta \underset{\sim}{v}$, as compared to u_o diminishes. As shown in the analysis, $\Delta \underset{\sim}{v}$ increases with a corresponding increase of curvature or slope of the terrain; consequently, we can expect uncertainties between the unknown exact solution and the results computed from the first-order perturbation theory to also increase with an increase in slope. Roughly speaking, these uncertainties are of the order $|S|^2$, where S is the slope of the terrain. Clearly then, the model should not be used when S is very large, although the question of "how large" is not yet resolved. We have been able to partly compensate for the inadequacies of the calculation for the case of a mountain ridge whose crestline is perpendicular to the airflow, and we suggest that the conclusions drawn from this investigation be extended to all cases.

Fundamentally, we have found that the first-order perturbation theory underestimates the vertical lift in the case of the aforementioned mountain ridge (see Appendix A). This was demonstrated by showing that the calculated surface wind trajectory (which for the exact solution should be identical with the contour of the mountain ridge) actually intersected the ridge. The discrepancies between the exact and calculated surface trajectories increase with a corresponding increase in ridge slope, as given by the ratio of the maximum elevation, h, to the half width, a.

$$S = (h/a) \quad .$$

However, by performing the calculations with a larger slope,

$$S' = h'/a \quad ,$$

where h' is larger than h, it becomes possible to make the calculated surface trajectory follow the mountain ridge contour. Figure 9 shows the relationship between the actual slope S and the required slope S' whose use will partially compensate for the limitations of the first-order perturbation theory. Thus, if $|h|$ is the actual height of the mountain (valley) ridge, the calculations should be performed with an h' given by

$$|h'| = |h|(S'/S) \quad , \tag{53}$$

where the ratio S'/S is evaluated by first determining S (e.g. point A) and then finding the corresponding value of S' (point B). We suggest that the modification in mountain ridge height, as given by Eq. (53), be extended to single mountains (valleys), although calculations supporting this

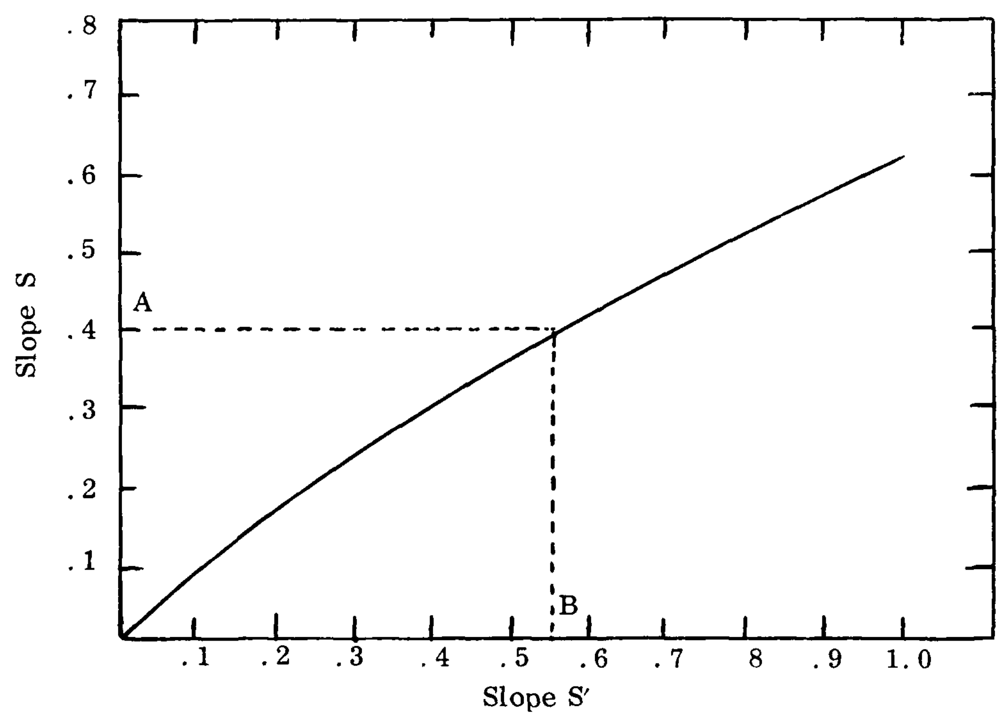


Figure 9. Slope Compensation

conjecture have not been rendered. The modification of elevation does not alleviate the shortcomings of first-order perturbation theory; consequently, we further suggest and, moreover, stipulate in the program itself that $S' \leq 0.6$.

It should also be noted in passing that the orographic effects extend indefinitely in altitude as can be seen by examining the mathematical expressions for the components of wind velocity. However, we have decided (based on a few sample calculations) to limit vertical consideration of an orographic region to three times the height of the highest obstacle in the region.

The Sea Breeze. The linearized model of the sea breeze as developed by Defant has been selected as the most suitable model for the sea breeze for two reasons: (1) it gives good agreement with experimental observation, and (2) the resulting analytical expressions for the components of the sea breeze are relatively simple from a computational standpoint. Defant[5] approaches the sea-breeze circulation problem in the sense of Lord Rayleigh's convection theory, the dynamics of which are governed by the continuity equation, the three momentum equations, the equation of state, and the heat-diffusion equation. By neglecting density variations in the continuity equation, and including them in the momentum equations since they modify the action of gravity, it becomes possible to construct a vorticity function from which the components of velocity in a plane perpendicular to the coast can be determined. Included in Defant's model is the assumption of an infinitely long coastline which points in the y direction; variations of the meteorological variables in this direction are neglected. The x axis is perpendicular to the coast and positive inland, while the z axis denotes the vertical.

Figure 10 shows the typical circulation pattern after sunrise when viewed along the direction of the coastline (positive y axis). In addition to the x-z circulation there is an accompanying y component of velocity which is related to the other components in a determined way, but is not shown in the figure. The driving force is

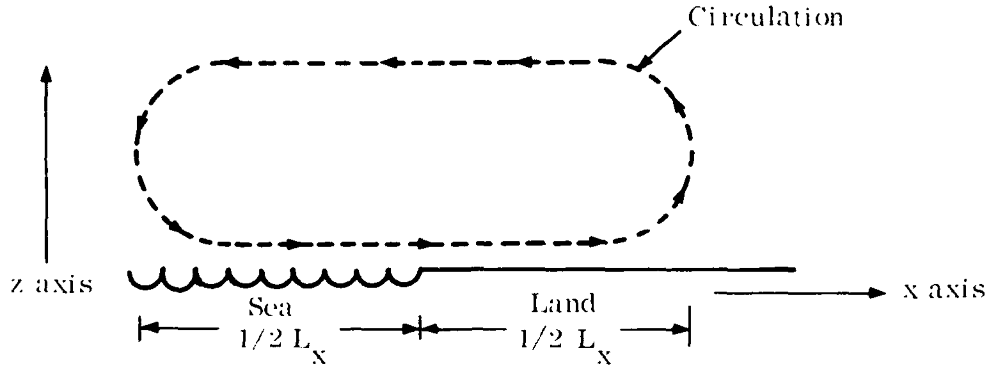


Figure 10. Sea-Breeze Circulation

of course the potential temperature differential at the surface, whose behavior with x and t is assumed to be given by

$$\theta(x, z = 0, t) = \sin \lambda x T(t) \quad , \tag{54}$$

where $\lambda = (\pi/2L_x)$ and T(t) is a function of time alone. The circulation pattern shown in Figure 10 occurs when the land temperature is higher than the water temperature (discounting the 1 hr or so lag time due to the inertia of the system). A positive value of T(t) corresponds to the surface temperature differential profile shown in Figure 11.

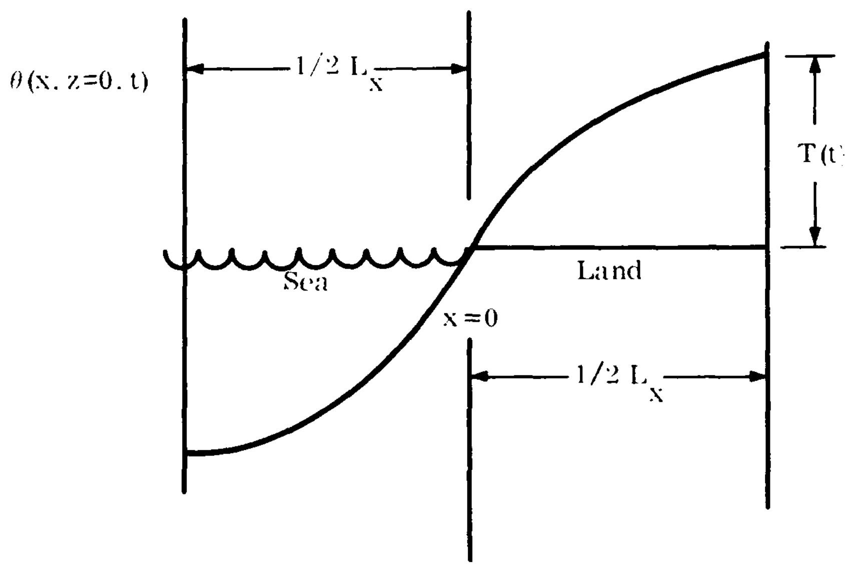


Figure 11. Temperature Differential Profile

According to the theory T(t) is expressible as a Fourier series in multiples of the sidereal day frequency, Ω. That is

$$T(t) = \sum_{n=1}^{\infty} T_n e^{in\Omega t} , \tag{55}$$

where

$$T_n = \Omega(2\pi)^{-1} \int_0^{2\pi/\Omega} T(t)\, e^{-in\Omega t}\, dt = T_n^* e^{i\tau_n} \tag{56}$$

is in general a complex quantity with amplitude T_n^* and phase τ_n. In addition to specifying the extent of the sea breeze, L_x, and T(t), it is necessary to specify the other characteristic physical parameters of the sea breeze which include: σ, the Guldberg-Mohn friction parameter; K, the thermal eddy diffusivity; θ_o, the average ground temperature; $\Gamma = (d\theta_o/dz)$, the initial unperturbed temperature gradient; and sin ϕ, where ϕ is the latitude at which the sea breeze is occurring. (A more comprehensive discussion of these physical parameters and their relationship to the overall structure of the sea-breeze circulation is available in Appendix B.)

The expansion of T(t) in a Fourier series results in the following expansion of the components of the wind field:

$$u(x, y, z, t) = \sum u_n(x, y, z, t) , \tag{57}$$

$$v(x, y, z, t) = \sum v_n(x, y, z, t) , \tag{58}$$

and

$$w(x, y, z, t) = \sum w_n(x, y, z, t) , \tag{59}$$

where u_n, v_n, and w_n are the partial contributions to the x, y, and z components of the wind field respectively from the nth harmonic. These quantities are essentially given by Eqs. (B.59), (B.60), and (B.61) of Appendix B, but can be simplified to the following form:

$$w_n = \sin \lambda x J_{nz} \Big[e^{k_{n1} z} \cos \left(n\Omega t + \ell_{n1} z + \phi_n \right) - e^{k_{n2} z} \cos \left(n\Omega t + \ell_{n2} + \phi_n \right) \Big] , \tag{60}$$

$$u_n = \cos \lambda x J_{nx} \Big[\bar{K}_{n1} e^{k_{n1} z} \cos \left(n\Omega t + \ell_{n1} z + \phi_n + \eta_{n1} \right) - \bar{K}_{n2} e^{k_{n2} z} \cos \left(n\Omega t + \ell_{n2} z + \phi_n + \eta_{n2} \right) \Big] , \tag{61}$$

and

$$v_n = \cos \lambda x J_{ny} \Big[\bar{K}_{n1} e^{k_{n1} z} \cos \left(n\Omega t + \ell_{n1} z + \phi_n + \eta_{n1} + \nu_n \right) - \bar{K}_{n2} e^{k_{n2} z} \cos \left(n\Omega t + \ell_{n2} z + \phi_n + \eta_{n2} + \nu_n \right) \Big] . \tag{62}$$

The constants J_{nz}, J_{nx}, and J_{ny} are each proportional to T_n^*, the magnitude of the nth temperature harmonic, and like all the mode-dependent constants appearing in Eqs. (60)-(62) are dependent on the physical parameters of the sea breeze. The constants k_{n1}, ℓ_{n1}, k_{n2}, ℓ_{n2}, $\bar{K}_{n1}$, $\bar{K}_{n2}$, η_{n1}, η_{n2}, and ν_n are completely independent of T_n^* or τ_n, while $\phi_n = \alpha_n + \tau_n$ where α_n is mode-dependent but otherwise independent of T_n^* or τ_n.

Since k_{n1} and k_{n2} are negative, all the components of the sea breeze will decay with altitude. The sea breeze does not have a precisely defined height but an effective height can clearly be related to the exponential decay constant. Because the first harmonic will always be the predominating term, we have decided to define the height of the sea breeze as twice the reciprocal of the minimum of $|k_{11}|$ or $|k_{12}|$. Thus H_s, the height of the sea breeze, is calculated internally and the user need not concern himself with its specification.

It is appreciated that situations can arise where information regarding the internal structure of the predicted sea breeze may be required. For this reason provision has been made to have the program print out the important mode-dependent constants and H_s.

We shall now briefly discuss the availability of the physical parameters of the sea breeze. A summary of suggested parameter values is given in Table 3 (p. 110).

L_x, the total extent of the sea breeze, is twice the inland or seaward extent of the sea breeze (in our sea-breeze model it is assumed that the inland and seaward extent of the sea breeze, as measured from the coastline, are equal). The dimensions of L_x are assumed to be available.

K, the thermal eddy diffusivity, is taken to be a space-independent quantity and as such its precise numerical value is not well defined. Measurements of K can, however, be made, and from them a suitable average value deduced, characteristic of a particular situation.

θ_o, the average ground temperature, can be determined by standard techniques.

Although σ, the Guldberg-Mohn parameter, does describe the effect of viscosity on damping the sea breeze, it is in some respects a device for incorporating friction in a simplified way — the reason being that it leads to relatively simple mathematical descriptions of circulation systems which appear to be in agreement with experiment. By increasing the values of σ we shorten the time lag between the maximum temperature and the maximum wind intensity of the sea breeze and also decrease the intensity per unit of temperature differential. For instance, in calculations performed by Defant,[6] it was shown that holding all other parameters fixed and increasing σ from 0 to 2.5×10^{-4} sec^{-1} shortened the time lag between maximum temperature differential and the maximum wind velocity from 6.7 to 1.4 hr. Concurrently, for the same temperature differential, a factor of 3 decrease in wind velocity occurred. The value of σ to be used in a given situation must be based upon past observations; that is, the sea-breeze circulation must be matched with the mathematical model by adjustment of σ. There are to our knowledge no known experimental methods which yield σ; however, suggested values are given in Table 3 (p. 110).

Γ, the vertical temperature gradient of the unperturbed atmosphere, is assumed as is done in all models of the sea breeze, to be positive.

T_n^* and τ_n, the amplitude and phases of the temperature harmonics, are input quantities calculated from the following formulas. Defining certain quantities δ_n and Δ_n by the equations

$$\delta_n = (2\pi)^{-1} \Omega \int_0^{2\pi/\Omega} T(t) \cos (n\Omega t) \, dt \tag{63}$$

and

$$\Delta_n = (2\pi)^{-1} \Omega \int_0^{2\pi/\Omega} T(t) \sin (n\Omega) \, dt \ , \tag{64}$$

where the time integration extends over 24 hr beginning at 1200 (noon) local time, gives

$$T_n^* = \left(\delta_n^2 + \Delta_n^2\right)^{1/2} \tag{65}$$

and

$$\tau_n = \tan^{-1} \left(\Delta_n / \delta_n\right) \ . \tag{66}$$

It is assumed that the meteorologist who is using the sea-breeze program can identify those measurements which can lead to the designation of the temporal behavior of the temporal differential T(t). It should be understood that the time t, used in the sea-breeze calculations is always relative to local noon time.

Besides the inherent physical parameters just described, there is one other parameter, related to the compatibility of the geometric description of the sea-breeze coastline to the computer program grid structure requirements, which must be discussed. It is anticipated that in any real situation a well-defined coastline length L_y will exist for the sea breeze. Thus, L_x, L_y, and ψ, the angle describing the orientation of the sea-breeze coastline with respect to the y-grid axis, Y_g, establish the horizontal configuration of the sea breeze.

For computational purposes it is necessary to render the sea-breeze geometry compatible with the (X_g, Y_g) grid structure. This necessitates redefining the extent of the sea breeze over the area bounded by the dashed lines (in Figure 12) with maximum and minimum values given by Y_{max}, Y_{min}, X_{max}, and X_{min}, which are determined by establishing the geometric center of the sea breeze, L_x, L_y, and ψ. However, the calculated values of the wind field are strictly defined over the domain of sea breeze as determined by L_x and L_y and x-y coordinate system. Thus, we must extrapolate the calculations into the stipled and hatched areas. Since the shoreline is assumed infinite in extent, it is theoretically permissible to use the calculated results, as they are, to determine the wind field in the stipled area. On the other hand, the extrapolation of the results for values of $|x| > (L_x/2)$ is not immediately obvious, but nevertheless can be achieved by interpreting the sea breeze as a circulation cell located in a continuous chain of circulation cells. However, this is only an approximation, arising from lack of a better method for attacking the problem. The degree to which this approximation may be meaningful is unresolved, although there is evidence to suggest that compensating air currents flow in regions adjacent to the sea breeze. If the sea breeze were really a single cell in a chain of circulating cells, then the sea-breeze equations as already derived would suffice to determine the wind field beyond $|x| > (L_x/2)$ because of the x periodicity of the system. To incorporate the idea of the circulation cells, and at the same time provide enough flexibility to account for departures from the idealiztion, we have decided to define the wind field in the hatched region by the relationship

$$\underset{\sim}{V}(x, y, z, t) = \underset{\sim}{V}_c(x, y, z, t) \exp\left\{-k_a\left[|x| - \left(L_x/2\right)\right]\right\},$$

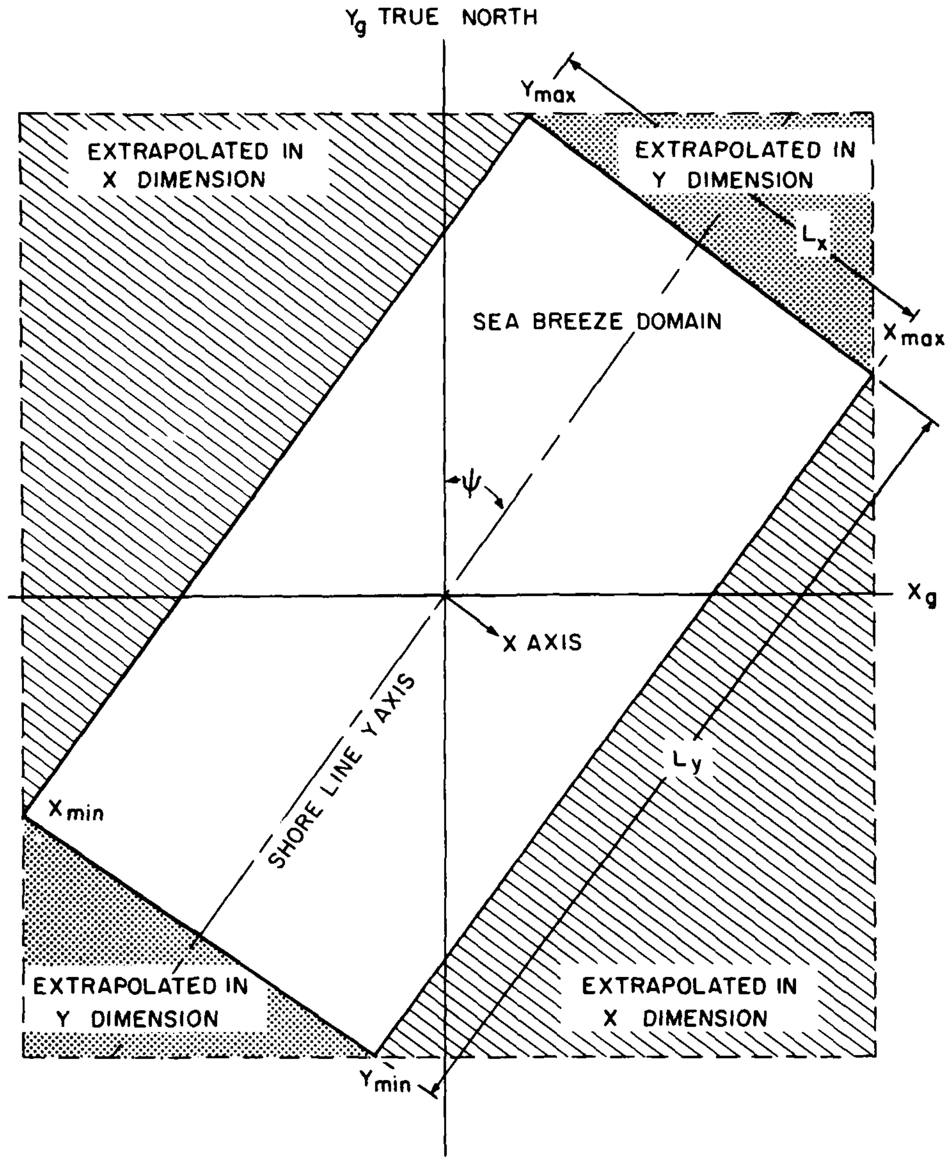


Figure 12. Geometric Considerations Related to the Sea Breeze

where $\underset{\sim}{V}_c$ is the calculated wind field in vector form whose x, y, and z components are given by Eqs. (60)-(62), and k_a is an attenuation factor. The case $k_a = 0$ corresponds to the idealized circulation cell system, whereas large values of k_a correspond to attenuated adjacent circulation cells. The computer program is constructed so that the present method of extrapolation can be changed at a later date. k_a is an input parameter which must be specified by the user.

Transport in a Macrowind Cell

Particle velocity for all particle transport is assumed to be given by the wind velocity (three dimensional) at the particle position minus the still-air particle settling rate. Within macrocells, particle trajectories are taken as straight lines; therefore, particles can be moved from one boundary to the next in one computational step. Such boundary-to-boundary transport is illustrated in two dimensions in Figure 13, which also shows the boundaries of one local cell superimposed on the macrostructure. In more detail, when a particle intercepts the boundary of a macrocell, the computations proceed as follows. We obtain the particle velocity components normal to the boundary planes of the wind cell. We then compute the time at which a boundary intercept would occur in each of the (three) component directions. The earliest of these (three) intercepts indicates the time of exit and the coordinates of the exit point are computed. Transport of a single particle

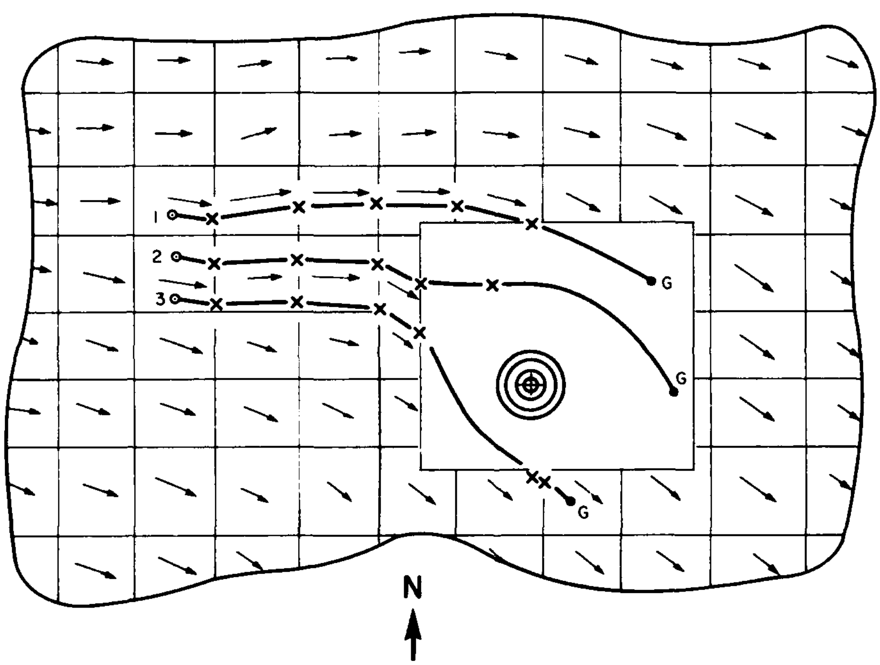


Figure 13. Boundary-to-Boundary Transport and a Mountain Wind Cell

through the compartmented macrowind field is merely an iteration on this single particle — single cell logic. During a calculation, complete trajectories are computed serially for individual particles between major time, or topography boundaries, or both. The exact natures of these boundaries are discussed later.

Transport in a Local Circulation System

When a particle passes into a local circulation system cell the mode of trajectory calculation changes from that used in the macrowind-field cells. Within local circulation system cells it is possible to calculate unique wind field velocities at all points. For this reason particle trajectories are computed from the particle velocity equations using point-slope numerical integration with a constant time step. The method is as follows. Suppose after n time steps the particle is at location (x_n, y_n, z_n) and has velocity $(v_{x,n}, v_{y,n}, v_{z,n})$. Then to determine the position of the particles at the n + 1th time step, for example, in the x direction, we perform the computation $x_{n+1} = x_n + v_{x,n} \Delta t$ (it is repeated for the other directions). The magnitude of Δt is determined by the user. The point-slope method of integration, including restriction on values of Δt, is discussed by Milne.[7]

Temporal Variation of the Wind Field

Temporal variation of the wind field is achieved by periodically replacing the entire wind field description data set. The period of data replacement is variable and each replacement interval is specified by the user.

Topography Description

Three different methods of specification are available. First, the user can specify a planar deposition surface at any altitude for use in areas not covered by local circulation cells. Alternatively, a system has been provided to allow the user to specify the topography in a piecewise-planar manner such as that illustrated in Figure 14. A grid system that can be subdivided indefinitely to yield any desired resolution of detail is used to achieve the desired resolution without the excessive redundancy of a strictly regular grid. Within local circulation cells other topographic descriptions must be used. For instance, the topography of mountains

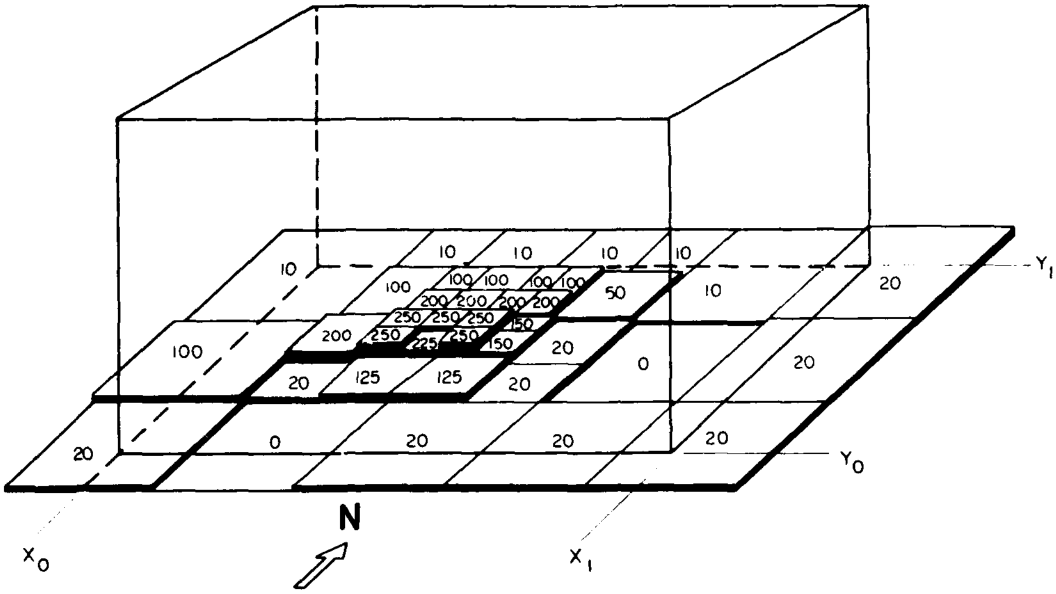


Figure 14. Piecewise-Planar Topography Specification Below the Macrowind-Field Volume (numbers are surface heights; vertical scale is exaggerated)

covered by a mountain wind model cell is described by an analytical mountain shape function. There is no provision in the model to account for shielding effects of highly variable terrain. Additional details are given in the User Information section (p. 133 ff) and in Appendix C.

COMPUTER PROGRAM OUTLINE

Description

In its initial form the DELFIC system is designed for execution on the IBM 7094 computer via the IBSYS-IBJOB processor, and the "overlay" feature is used to control the input sequence of major sections of the system. To facilitate discussions of the programs, we have assigned the executive programs of each major section the names LINK1, LINK2, ... , which are more-or-less indicative of their positions in the computation flow sequence. The Transport Module essentially consists of three such major program sections:

LINK5 Initialization and control

LINK6 Wind-field description

LINK7 Particle transport.

Figure 15 shows the arrangement in which the computa ions required during the transport period are grouped for execution. Note that final exit from LINK5, the transport executive, is made to a program called LINK8 — the output processor. Figure 16(a) is a flow chart of the general program logic of the Transport Module. This simplified representation shows in some detail the hierarchy of computation loops that make up the transport logic. A simpler representation of this hierarchy is given in (b) of Figure 16, which shows a nested set of five loops. In the outermost loop, there is a test to determine if the specified temporal extent of the transport has been achieved; if not, an updated version of the wind-field description is computed. In the next lower hierarchy level a part of a multipart wind field description is brought into the computer (if a multipart description is in use) in order to transport particles which have gone beyond the in-core part of the description. In the third level of the hierarchy the topographic description is treated like the multipart wind description (if required). In the particles aloft list loop individual particle descriptions are given sequential attention, and in the actual transport code the individual fallout particle is transported until it reaches either the ground or some boundary at which in-core data are insufficient to move it further.

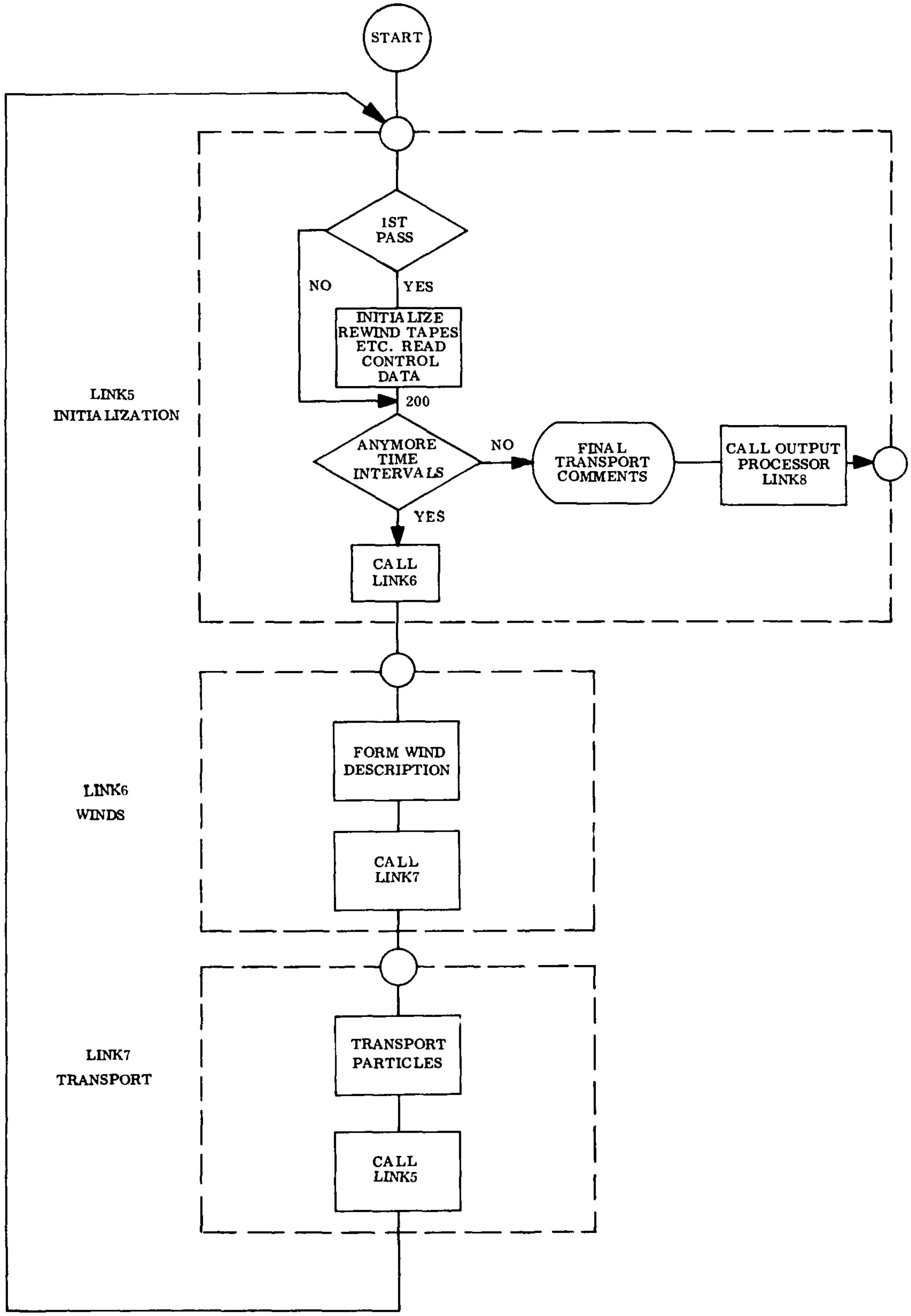
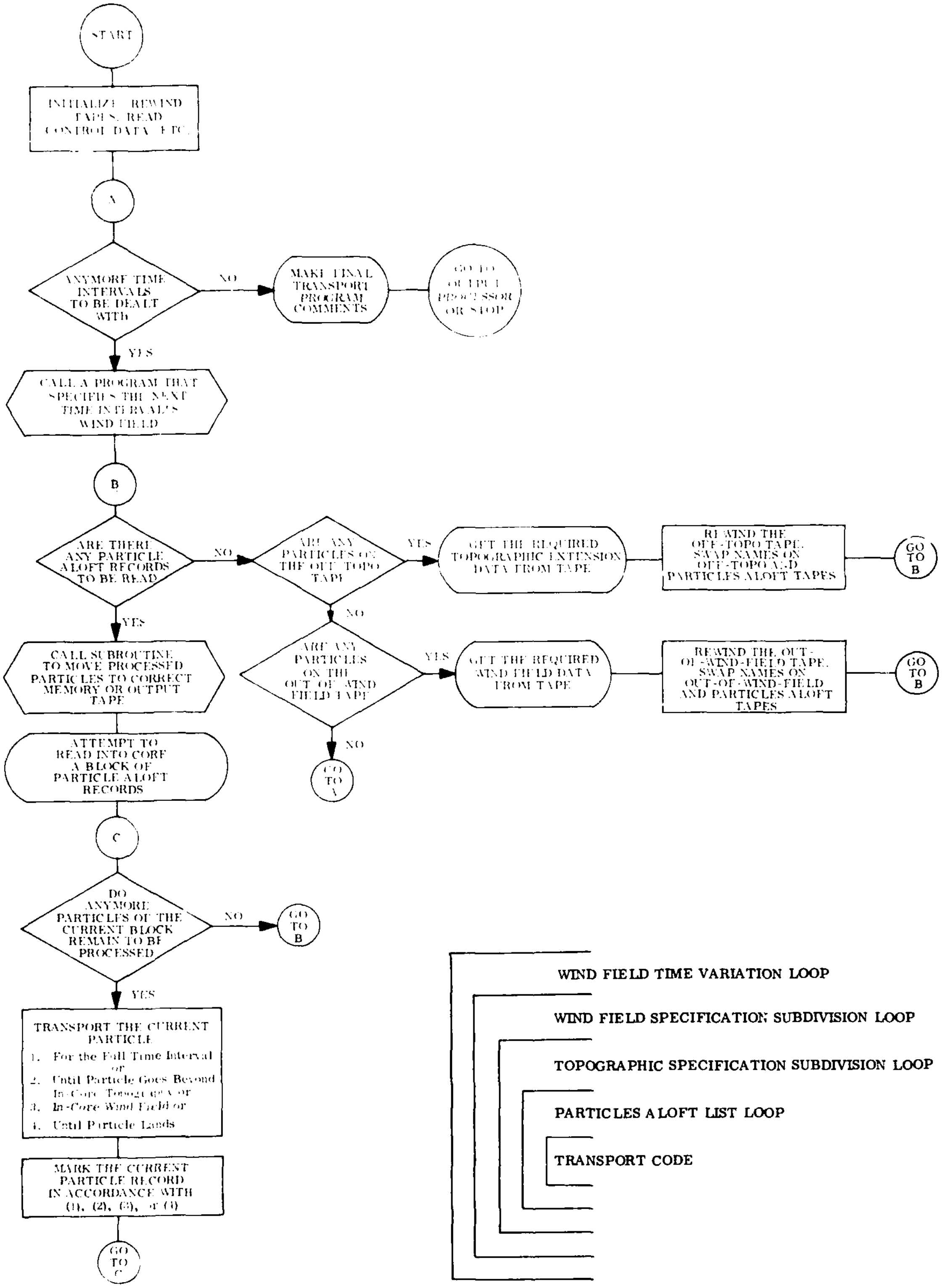


Figure 15. Program Arrangement for the Transport Module



(a) (b)

Figure 16. General Flow Chart of the Transport Module (a), and Transport Module Loops (b)

Figure 17 represents schematically the flow of information from secondary (tape) memory to primary (core) memory and back during an extensive run of the transport program. Using Figure 17 as a guide, let us consider the sequence of data flows.

Initially, only the particles (input) and topography tapes contain any information, and only the transport codes themselves are in primary memory. The initialization and control program (LINK5) reads identification information from the particles (input) tape, writes comments on the system output tape, and then, if required, loads the topography arrays from a previously prepared topography tape.* At this point the wind-field description program (LINK6) is called and a wind-field description is generated. This description is generated directly (and completely) into the wind arrays in primary memory by the current versions of LINK6. However, if future requirements warrant, a modified version of LINK6 can produce a more extensive description of the wind field and be forced to store part of it on tape. In either case, when LINK6 is completed, the wind arrays are loaded and a "map" of the wind tape (if any) has been produced and stored in primary memory.

Next, we enter LINK7, the actual transport program, and read a part of the particles (input) tape into primary memory. The particle descriptions are then transported one at a time until one of five possible conditions arises. These conditions, which may be thought of as boundaries, are.

1. The particle drifts beyond the area for which a topographic height has been specified in core. In this case the particle's description is marked so that it will be eventually written onto the off-topo tape

2. The particle drifts beyond the region for which the wind velocity field has been specified in core. In this case the description is marked to go on the out-of-wind-field tape.

---

*A special program has been written to aid the researcher in preparing topography tapes from topographic maps or other sources (see Appendix C). The user may, however, specify a planar topography and bypass the use of a detailed topographic tape.

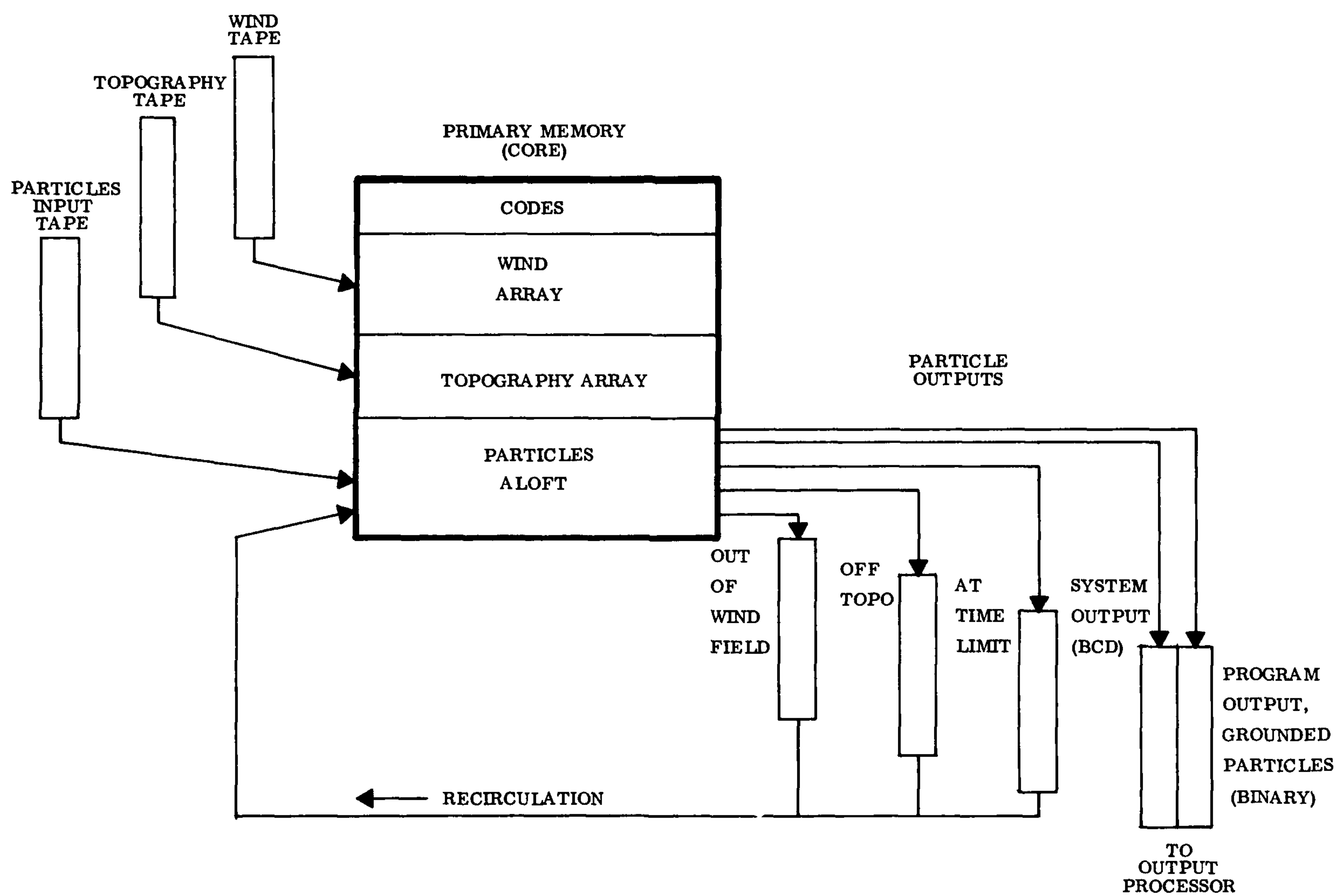


Figure 17. Transport Module Data Flow

3. The particle encounters neither of the previously mentioned boundaries and is still aloft at the time when the wind-field description must be updated to achieve discrete temporal variability of the wind field. In this case the description is marked to go on the time boundary tape.

4. The particle becomes grounded on the topography. In this case the particle description is marked so that it eventually is written on the program output tape which is used as an input to the output processor.

5. The particle drifts beyond the entire secondary as well as primary memory region of specification for either topography or winds In this case the particle is labeled as a "lost particle" and it is removed from the transport process.

When the entire block of descriptions has been read into memory and processed the next block of particle descriptions is read into memory and processed. After all particle descriptions on the original input tape have been processed treatment of the data (if any) on the three recirculation tapes begins. First, if any descriptions were written on the off-topo tape, a new block of topographic data is read in and the off-topo tape is put into the position (symbolically) of the original particles input tape. Processing continues as before, and eventually the condition will obtain that at the end of a pass no descriptions will be found on the off-topo tape Under this condition we next consider the out-of-wind-field tape in a manner analogous to "off-topo." The treatment given to the time boundary tape is similar, but when all particles that are still aloft are on the time boundary tape, a new description of the wind field must be computed. Before each call of the wind-field program (LINK6) a check is made to see if the transport time limit has been exceeded, and if it has been, a termination procedure is executed to record the final status of memory.

Table 1 is a summary of the 14 programs of the Transport Module. Detailed discussions of these programs are given in the next section.

TABLE 1

A SYNOPSIS OF THE PROGRAMS OF THE TRANSPORT MODULE

Name	Called By	Purpose
LINK5	Executive Program M3*	Transport initialization and control.
RDTOPO	LINK5 and LINK7	Reads a block of topographic data into core memory.
LINK6	Executive Program M3*	Calls subroutine MKWIND
DUMPP	LINK5 and LINK7	Makes room in the particle array for a block of N new particle descriptions by writing a set of particle descriptions onto some memory or output tape.
MKWIND	LINK5	Updates entire wind field description directly into the common wind field arrays of the Transport Module. It accepts many wind vector data and computes a spatially variant wind field description by a number of different methods such as: 1. Assign to the wind grid point the vector at the nearest data points 2. Assign to the wind grid point a distance weighted average of the vectors at the N nearest data points 3. Fit a linear model to the N nearest data points by least squares and use that model to assign the vector to the grid point Provision has been made throughout the programming for the eventual inclusion of a system for the use of a voluminous wind field description recorded on and retrieved from a secondary memory system such as magnetic tape or disk.
RDCIRS	MKWIND	Reads data which describe any local circulation system which may exist. These data state the size and location of each local circulation cell and identify the computation program which is to be used within each cell.
LINK7	Executive Program M3	Transports all input particle descriptions through the specified wind field.
FALRAT	LINK7	Computes settling rate for a particle as a function of particle size and altitude.
HEIGHT	LINK7	Retrieves the height of the topography for the position of the current particle from the topographic data arrays.
LOTRAN	LINK7	Transports a particle within or above a local circulation system cell.
MTWND1	LINK7 and LOTRAN	A dual purpose subroutine which (1) reads the data that is needed by the MTWND1 (mountain wind) program and carries out those computations that are invariant with position, or (2) computes wind vectors at specified positions within the MTWND1 cell.
RGWND1	LINK7 and LOTRAN	Like MTWND1 but for the analytical ridge wind model.
CBREZ1	LINK7 and LOTRAN	Like MTWND1 but for the analytical sea breeze wind model.
GETWND	LINK7 and LOTRAN	Retrieves the appropriate wind vectors from the macro-wind-field description arrays.

*See DASA-1800-VII (Operator's Manual).

Program Discussion

In this section we present a detailed description of each of the executive programs and subroutines* of the Transport Module Each program description is headed by the program name, its call list (if any), and flow chart (FC) number.

Subroutine FALRAT (ALT, PSIZE, FV, ATEMP, RHO. FROG, ISOUT) (FC-1)

This subroutine computes the settling rate of a particle at height ALT in an atmosphere for which the density and dynamic viscosity are tabulated in arrays RHO and ATEMP respectively. These tabulations must be for 200 m intervals starting from 1000 m below MSL.† Fall rate equations derived by Davies[2] are used All units are in the meter-kilogram-second (mks) system except for PSIZE, the diameter of the particle, which is in microns, and FROG, which is the pre-computed product $4/3*g*ROPART*10^{-8}$ where ROPART is the density of fallout particles (mks) and g is the acceleration of gravity (mks).

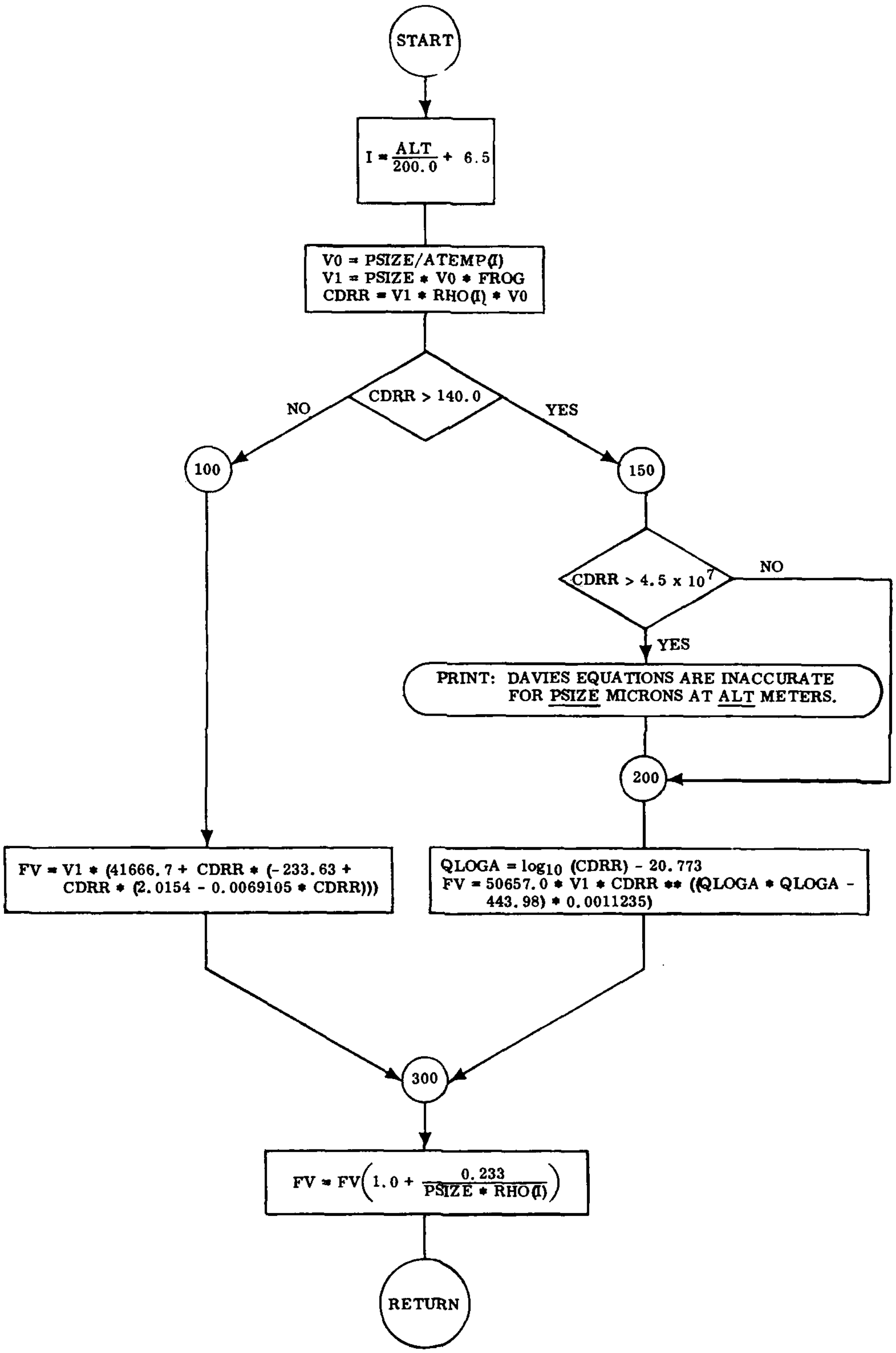
The Davies equations which are functions of the quantity C_DR^2 are valid over separate ranges of C_DR^2. The separation occurs at $C_DR^2 = 140$. An overall upper limit of $C_DR^2 = 4.7 \times 10^7$ is imposed by Davies for the validity of his equations. However, for lack of an appropriate substitute for use in computing the settling rate for particles which exceed this limit, we have chosen to use Davies equation for cases where $C_DR^2 > 4.7 \times 10^7$. The program will record an indication that the limit was exceeded for each case encountered

The computation proceeds in the following manner. After locating the particle in one of the atmospheric layers,† the program computes CDRR(C_DR^2) and several intermediate parameters Next CDRR is tested to determine which expression is to be used for the terminal velocity. If the upper range is used, a check is made to determine if CDRR > 4.7×10^7. If this is so, the printout "DAVIES EQUATIONS ARE INACCURATE FOR PSIZE MICRONS AT ALT METERS" is made. PSIZE refers to particle diameter in microns and ALT refers to particle altitude in meters Then, the settling rate of the particle, FV, is computed Finally, a drag slip correction in the form of Cunningham's factor (see Appendix B of Ref. 1) is applied to FV and control is returned to the calling program.

---

*There are numerous error checks throughout the programs that result in calls to subroutine ERROR when termination is required A full description of subroutine ERROR is included in DASA-1800-VII (Operator's Manual)

† The atmosphere structure defined for the cloud-rise computations is used



FC-1. Flow Chart for Subroutine FALRAT

Subroutine DUMPP (FC-2 and FC-3)

This subroutine along with parts of the main programs of LINK5 and LINK7 manages the system of primary (core) and secondary (tape) memory that is used to record descriptions of particles (central particles of cloud subdivisions) during transport. DUMPP serves to select and write one or more of the subsets of the particle descriptions (defined in Table 2) in primary memory onto some secondary memory or output tape and thus to make room available in primary memory. As one of its inputs DUMPP accepts the number (N) of particle descriptions for which room must be prepared in primary memory. It does not return until at least N blank lines have been made available in the top (low-numbered end) of the particle description arrays. DUMPP begins by selecting for dumping onto tape that set of particles which is considered best from the point of view of machine efficiency. In general, the largest set is considered to be best to dump because of the time required to put a tape drive into motion. However, an exception is made for the class of grounded particles, since they will be written on the transport-output tape (IPOUT) and will never be recirculated into the primary memory; therefore, whenever dumping the set of grounded particles would make sufficient room available (counting those lines that are already blank) for N incoming particle descriptions, the set of grounded particles is dumped. Before the actual dumping occurs, the particle description in core storage is reordered so that all descriptions to be dumped are located in a solid block beginning at the top of the particle descriptions array, and all particle descriptions that are to remain in core are moved below this block. The dumping operation then is executed, and finally a block of blanks (empty spaces) large enough to receive the incoming particles is prepared at the top of the particle descriptions array.

The main transport loop (in LINK7) passes sequentially across the list of particle descriptions which consist, for the Jth particle, of three spatial coordinates XP(J), YP(J), and ZP(J); a time coordinate TP(J); a particle size PS(J); and a mass per unit area FMAS(J). At the end of its pass the main transport will have marked each of the descriptions to indicate its membership in one of the five classes listed in Table 2. To avoid the use of another array of data, the sign bit of FMAS(J) and the sign and magnitude of the time coordinate TP(J) are used to record the class of the description as indicated in Table 2.

TABLE 2

PARTICLE CLASSIFICATION IDENTIFIERS USED BY DUMPP

Class	FMAS(J)	TP(J)	JTEST1
Blank	0	Not Used	
Grounded particles	-FMAS(J)	-TP(J)	1
Lost particles. These are particles that have gone beyond the complete wind field or topographic description	-FMAS(J)	TLIMIT	2
Topography boundary particles. These are particles at the limit of the in-core topography	+FMAS(J)	-TP(J)	3
Time boundary particles. These are particles at the time limit for the in-core wind field	+FMAS(J)	ENDTIM	4
Wind-field boundary particles. These are particles at the spatial limit of the in-core wind field	-FMAS(J)	+TP(J)	5

Referring to the general and the deailed flow charts of subroutine DUMPP (FC-2, and FC-3, respectively), we shall next consider its operation. First by comparing N, the number of incoming particle descriptions, with NFREE, the current number of blank lines in the arrays, we can immediately determine whether any descriptions must be dumped. If none need be dumped, we set JTEST = 0 to indicate that no blanks are known to already be at the top of the particle arrays and then transfer to 152 where the needed number of blank lines are brought to the top of the arrays from wherever they may be within them. If some particles must be dumped, we transfer to 151 to determine which set to dump.

At 151 we determine if the number of particles in the grounded set plus the number of blank lines in total provide enough space for the block of N particles which are to come in. If they do, we set the parameters JTEST = 1 and JTEST = NG to indicate respectively the class of particles to be dumped and the size of that class. Then a transfer is made to 18 where other preparations are made to carry out the dump. If a larger dump is required to yield N empty spaces, the set with the largest membership is selected and JTEST1 (see Table 2) and JTEST are set appropriately.

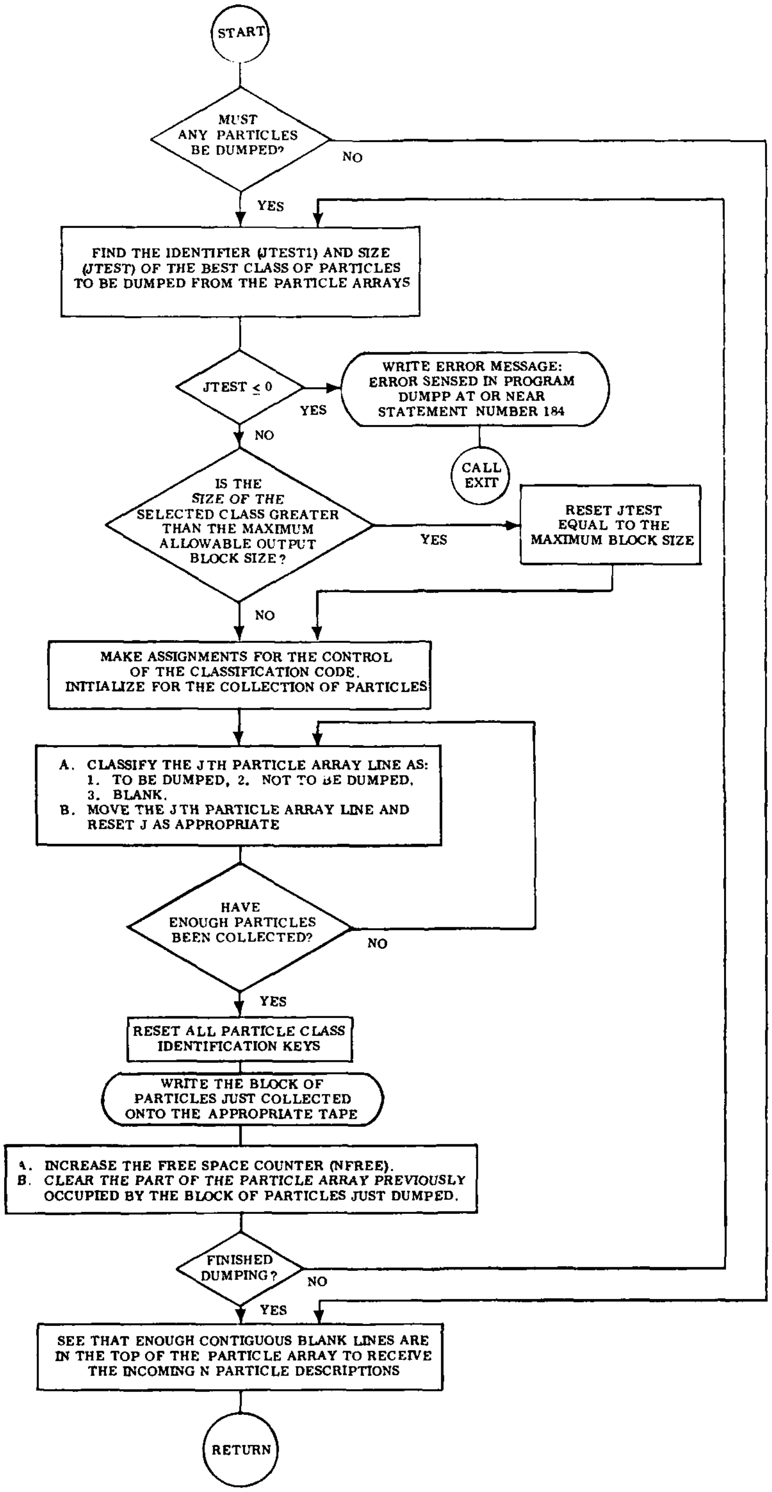
At 18 a safety test leading to an error stop is carried out followed by a threshold test on the size of the set to be dumped. Because a limit exists on the size of any particle block read by the output processor (see DASA-1800-VI), and also because we must impose block size control to allow for recirculation of data during transport itself, a maximum block size is defined within the LINK5 program. No block larger than NBMAX will be written by DUMPP.

At 181 the program branches, on the basis of the class of particles to be dumped (JTEST1), to a code that appropriately sets a group of assigned go-to statements and tape name parameters for use within the code that actually selects particle descriptions. Also at these points, the appropriate class count (NG, NLOST, NTO, NTI, or NW) is decreased in accordance with the number of descriptions about to be dumped.

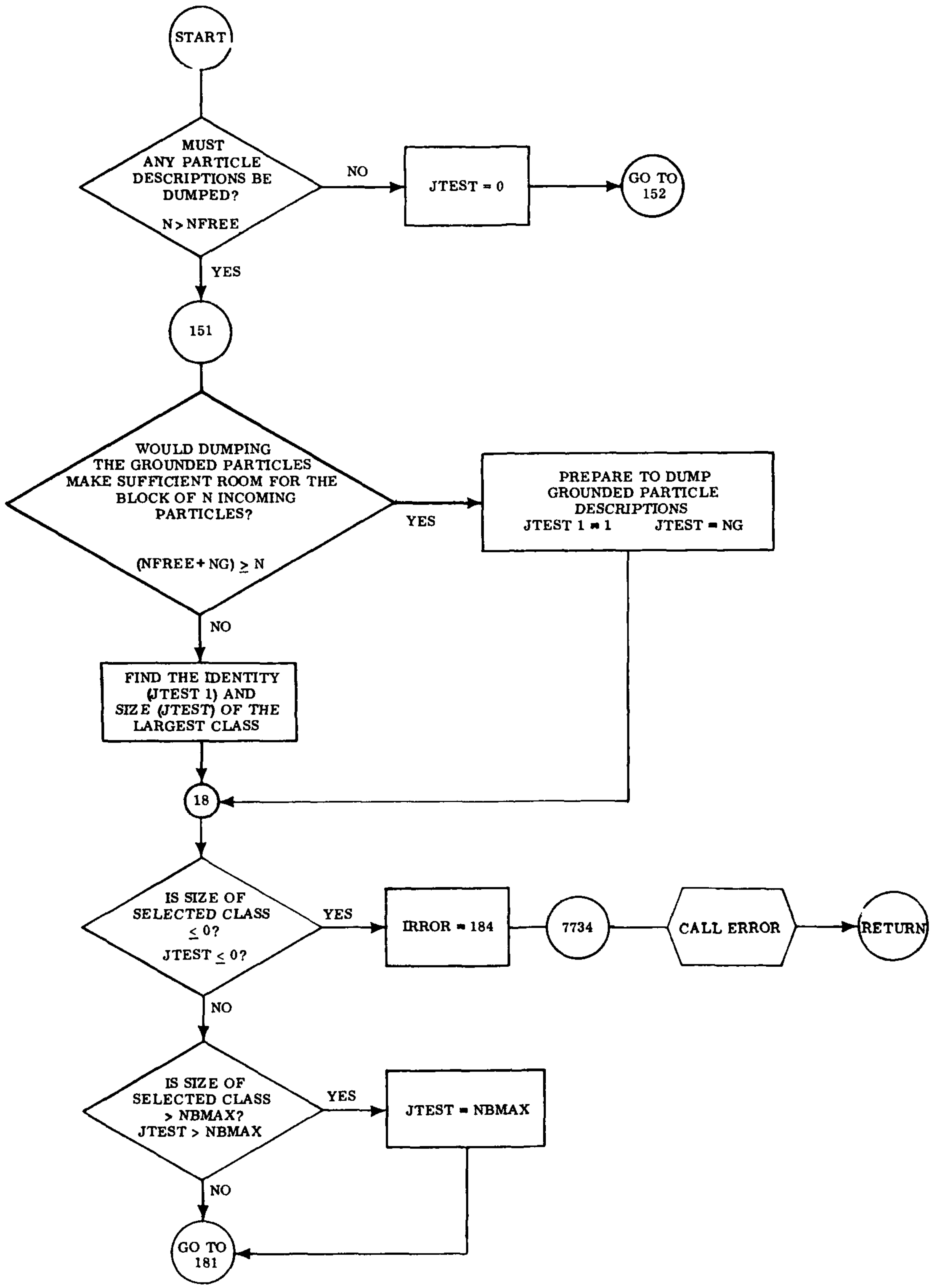
At 99 a one-line summary printout of information on the particle block to be dumped and of the particle counts is executed. Specifically, this output consists of the following data in order of printing from left to right: JTEST, JTEST1, and the current (predump) values of the in-core counts for blanks, grounded particles, lost particles, topography boundary particles, time boundary particles, and windfield boundary particles. Then we set certain parameters that are used within the loop that actually sorts the particles to be dumped into the top of the particles array. That loop, beginning at 98, first classifies a line in the particle array into one of three classes: blank, to be dumped, or not to be dumped. Classification is done by a set of assigned go-to statements. After this three-way classification, various actions occur in such a way to provide the needed sort into a contiguous block with something close to the theoretically minimum number of word movements. The particle classification and sorting code is logically complex and should be modified only with great caution.

At 1102 the sort is completed and all class indicator signs are set positive in preparation for actual dumping. In the case that lost particles are to be dumped, the control parameter IC(8) is tested to determine if printed listings of lost particles are requested. If IC(8) = 0, the lost particle count and particle descriptions (XP, YP, ZP, TP, PS, and FMAS) for the complete block are written on the IBSYS output tape for printing. If IC(8) $\neq$ 0, this printing is deleted. In any case, no further dumping action is required for lost particles. For all other classifications of particles, the block of particle descriptions is written on the appropriate binary auxiliary tape following its block count.

At 154 additional sorting is done, if necessary, to prepare a solid block of blanks at the top of the particles description array that is large enough to receive the incoming block of particle descriptions. This is done by interchanging locations of particles that lie above the block boundary with blanks that lie below it.

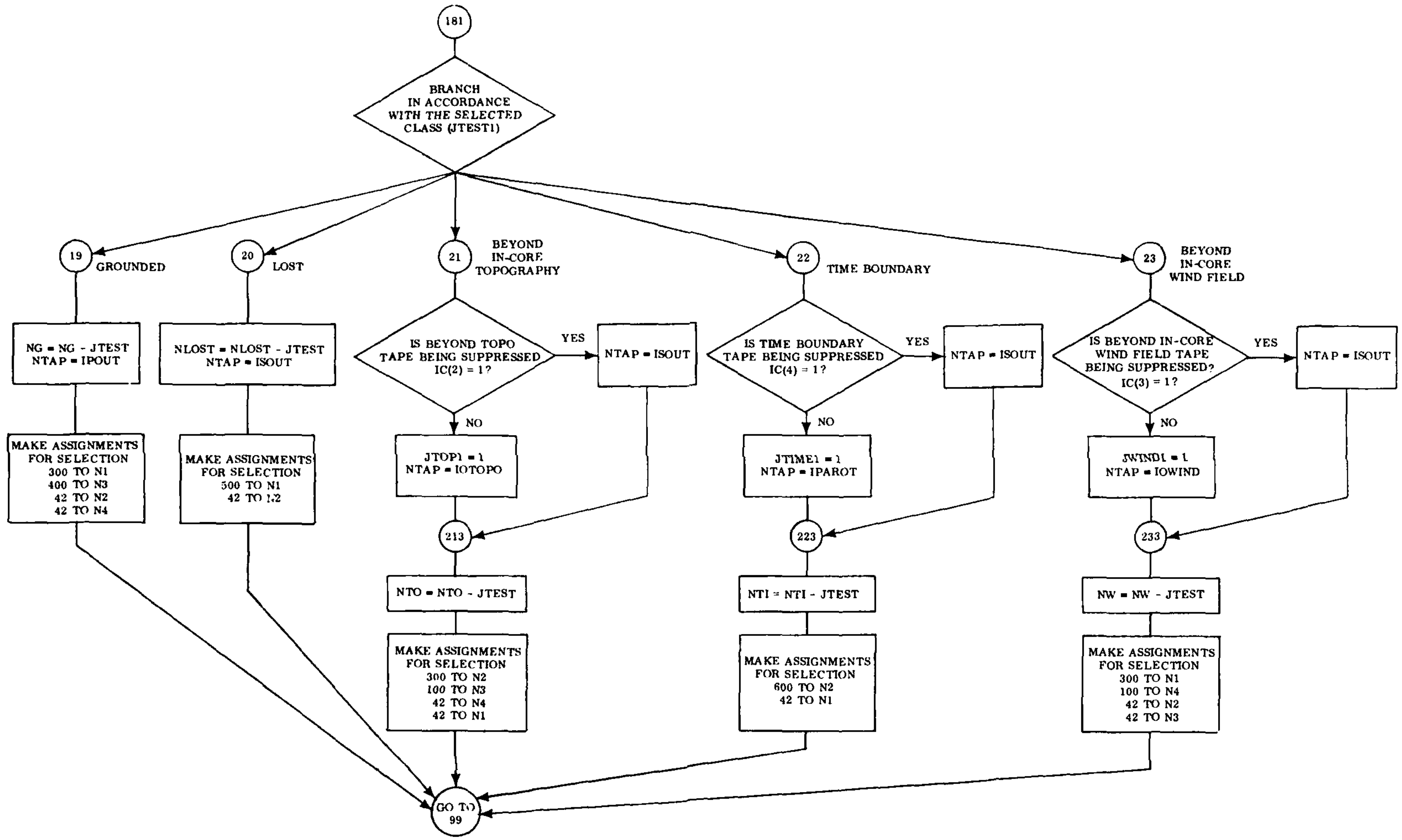


FC-2. Organizational Flow Chart for Subroutine DUMPP



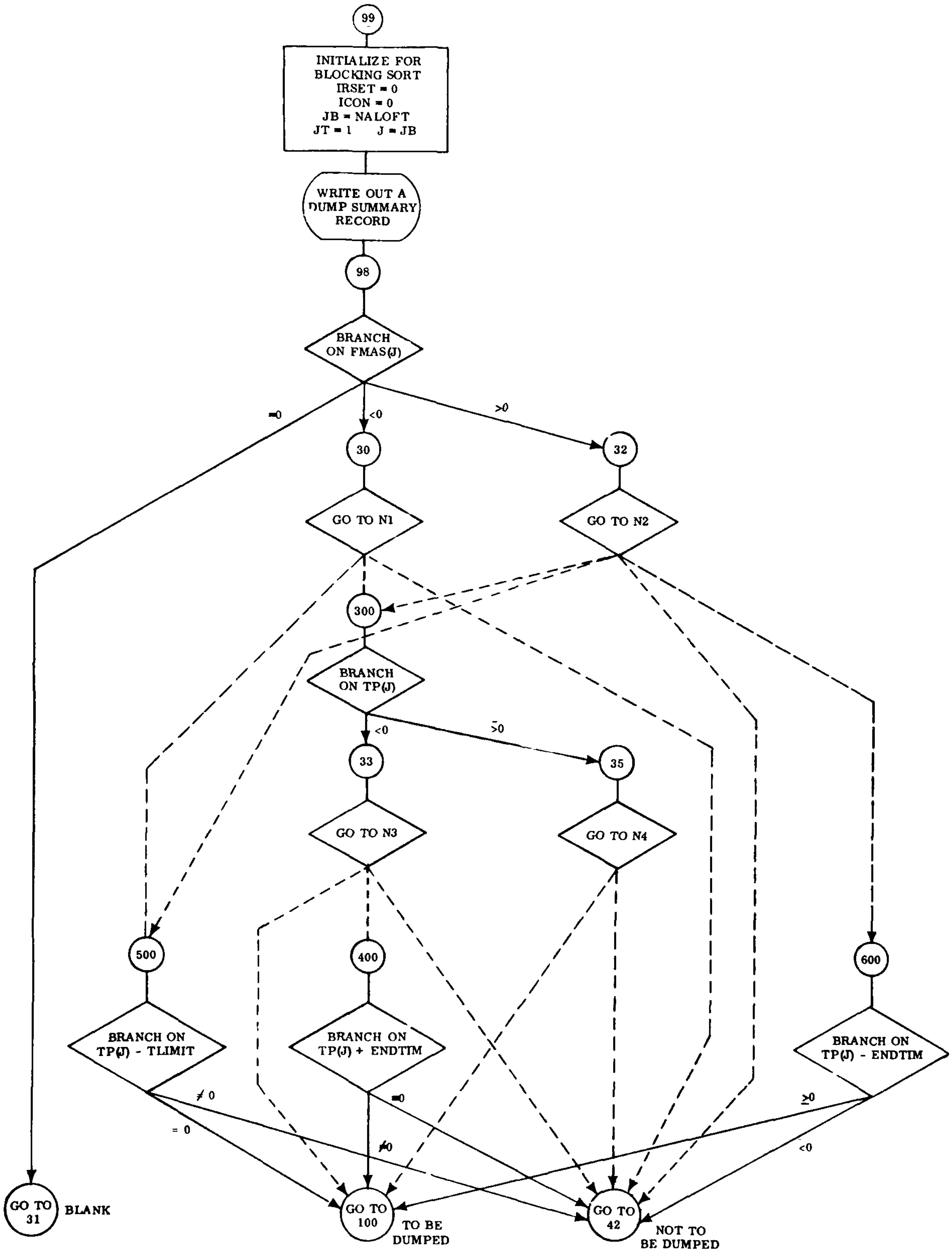
(a)

FC-3. Detailed Flow Charts for Subroutine DUMPP



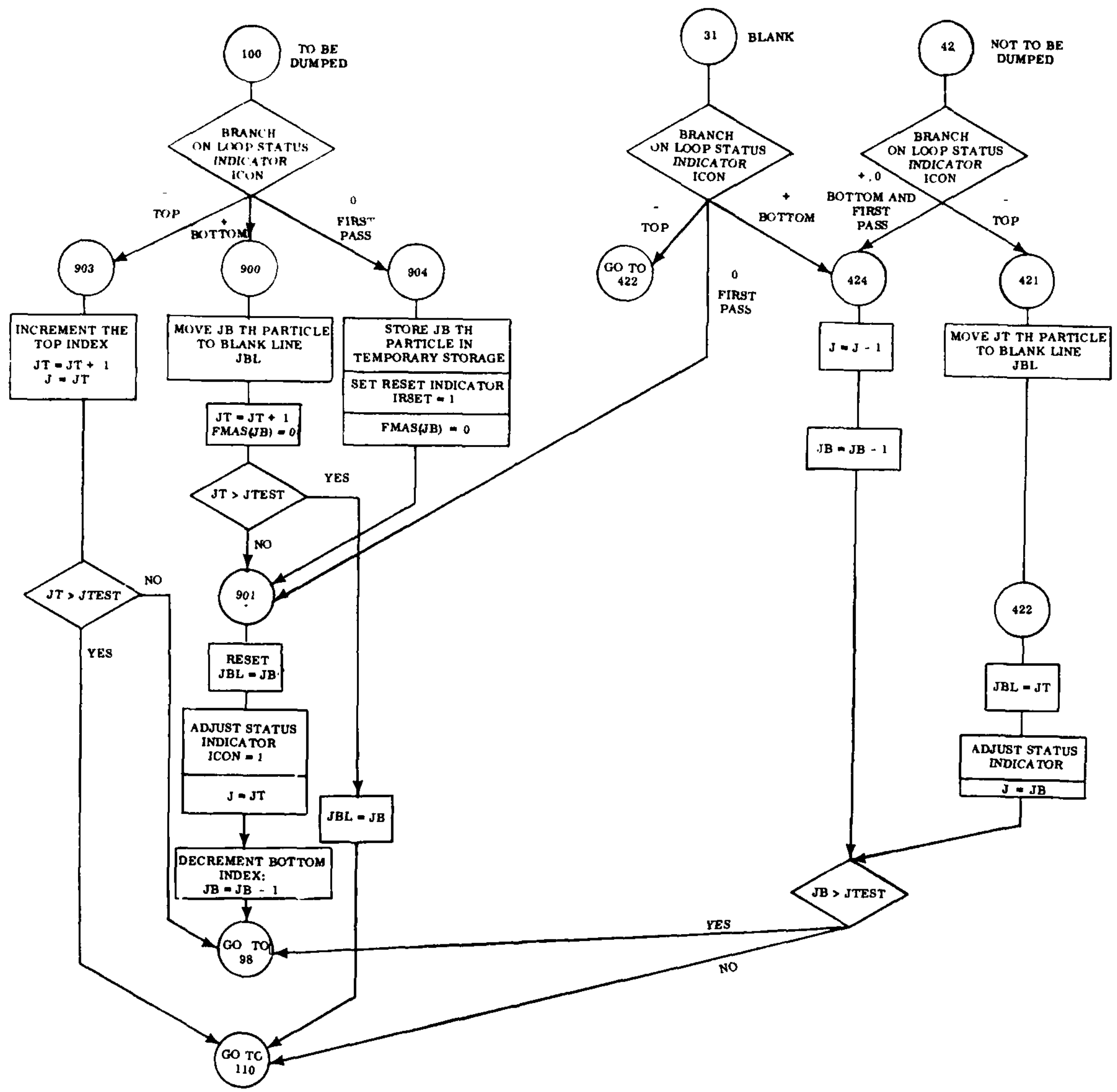
(b)

FC-3. (Continued) Detailed Flow Charts for Subroutine DUMPP



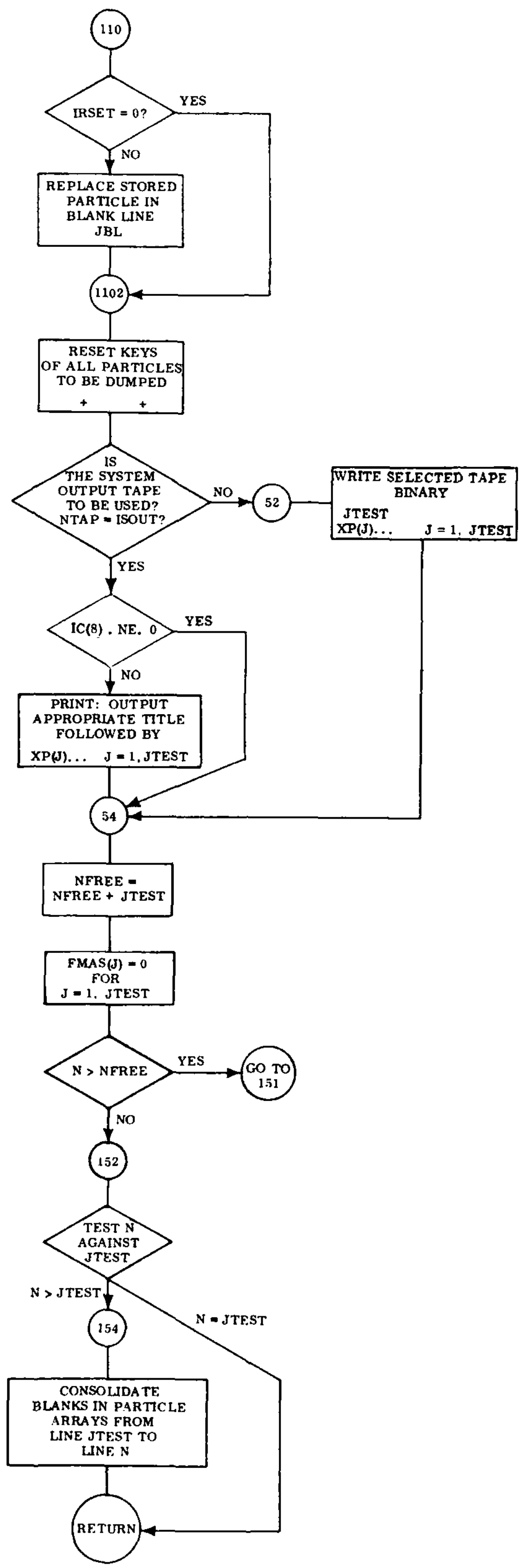
(c)

FC-3. (Continued) Detailed Flow Charts for Subroutine DUMPP



(d)

FC-3. (Continued) Detailed Flow Chart for Subroutine DUMPP



(e)

FC-3. (Continued) Detailed Flow Chart for Subroutine DUMPP

Subroutine RDTOPO (no flow chart)

This subroutine is used by both LINK5 and LINK7 to read topographic data from the topographic data tape IHTOPO. The contents of tape IHTOPO are described in detail in the User Information section, and the FORTRAN variables referred to below are defined there.

For each block of topography data to be read, subroutine RDTOPO checks the values of II, JJ, and KK to determine whether they are within the prescribed range of values to avoid the possibility of an overflow in core storage beyond the space reserved for the arrays. If an error is found, the comment — INCORRECT TOPO TABLE OF CONTENTS — is made and execution of the run is terminated. If satisfactory values of II, JJ, and KK are found, the arrays S and SUBSID are read into core memory from tape IHTOPO, and control is returned to the calling program.

Subroutine LINK5 (FC-4 and FC-5)

This program acts as an initializer and controller for the Transport Module. Upon the first entrance to LINK5 it initializes parameters and reads the following information from the IBSYS input tape: a transport identifier, transport control data (array IC(J)), and the transport time limit (TLIMIT). Based on the control data LINK5 next rewinds only those tapes that may be used during transport. If a piecewise-planar topography tape is to be used, its identifier is next read and checked. If the wrong tape has been mounted, a comment is written and the program awaits operator action before trying again. Next, the tape of particles, IPARIN,* ready for transport is checked in a manner similar to that used on the topography tape. When found to be correct the program next reads from this tape (IPARIN) a number of data sets that are needed by either transport or the output processor, or both. Included in these data sets are: detonation parameters; the Cloud Rise-Transport Interface Module run identifier; the cloud-rise identifier; the detonation identifier; the fallout particle density; tabulated distributions of particle mass, activity (optional), and surface-to-volume ratio as functions of particle diameter; and a tabulated atmospheric description that consists of viscosity and density versus altitude.

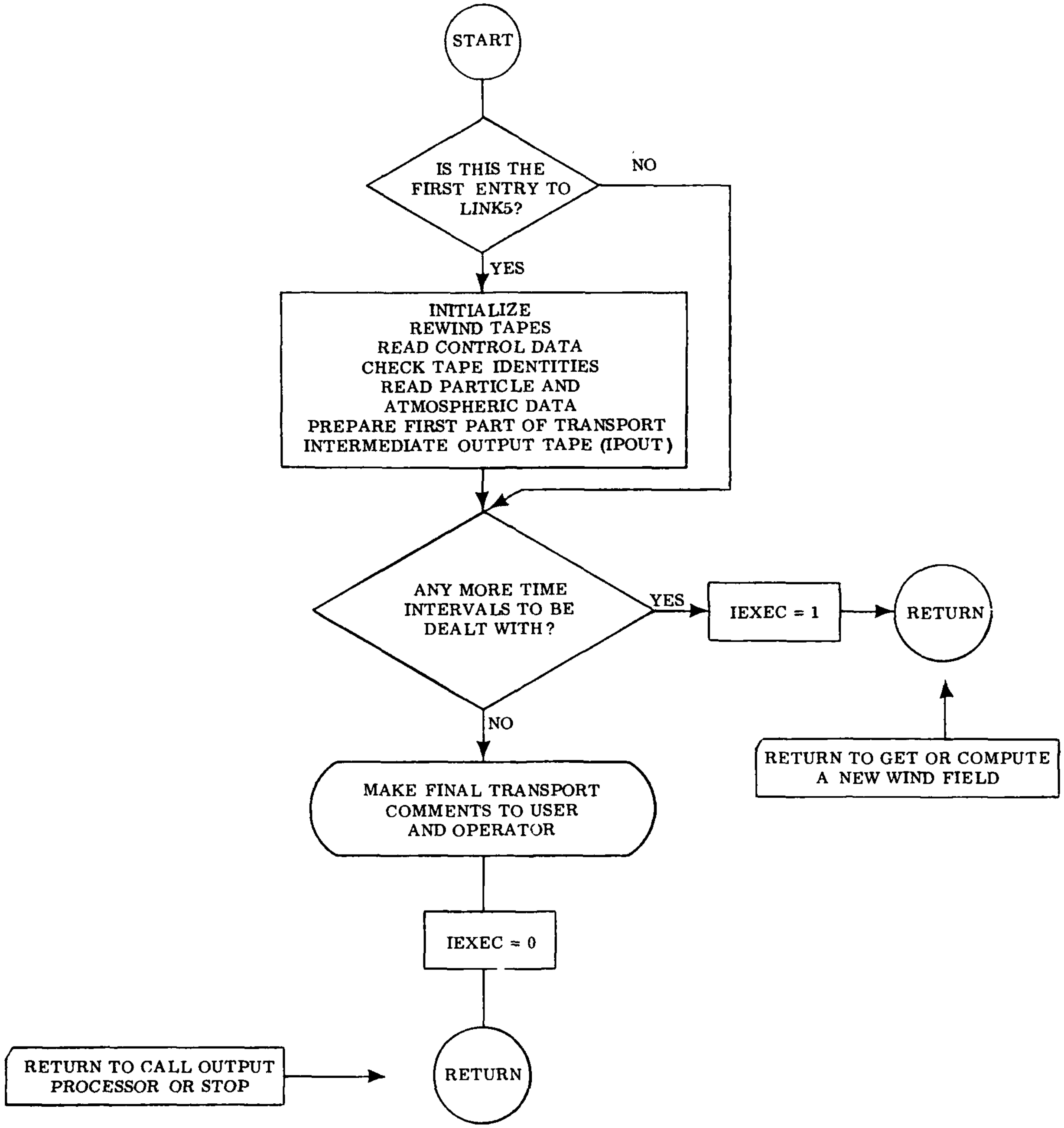
*Tape IPARIN has been prepared by the Cloud Rise-Transport Interface Module. See DASA-1800-III.

Next, a parameter (FROG) is computed that is required by the particle setting rate computations (subroutine FALRAT). Then if a piecewise-planar topography tape is not to be used, a height is read and stored for use as the height of a fully planar topography and a transfer is made to statement 205 where wind data are read. On the other hand, if a piecewise-planar topography is to be used, its identifier and table of contents are read from the tape IHTOPO. The parameter HTOPO is set at the highest topographic height on the whole tape and the first topo data block is read by calling subroutine RDTOPO.

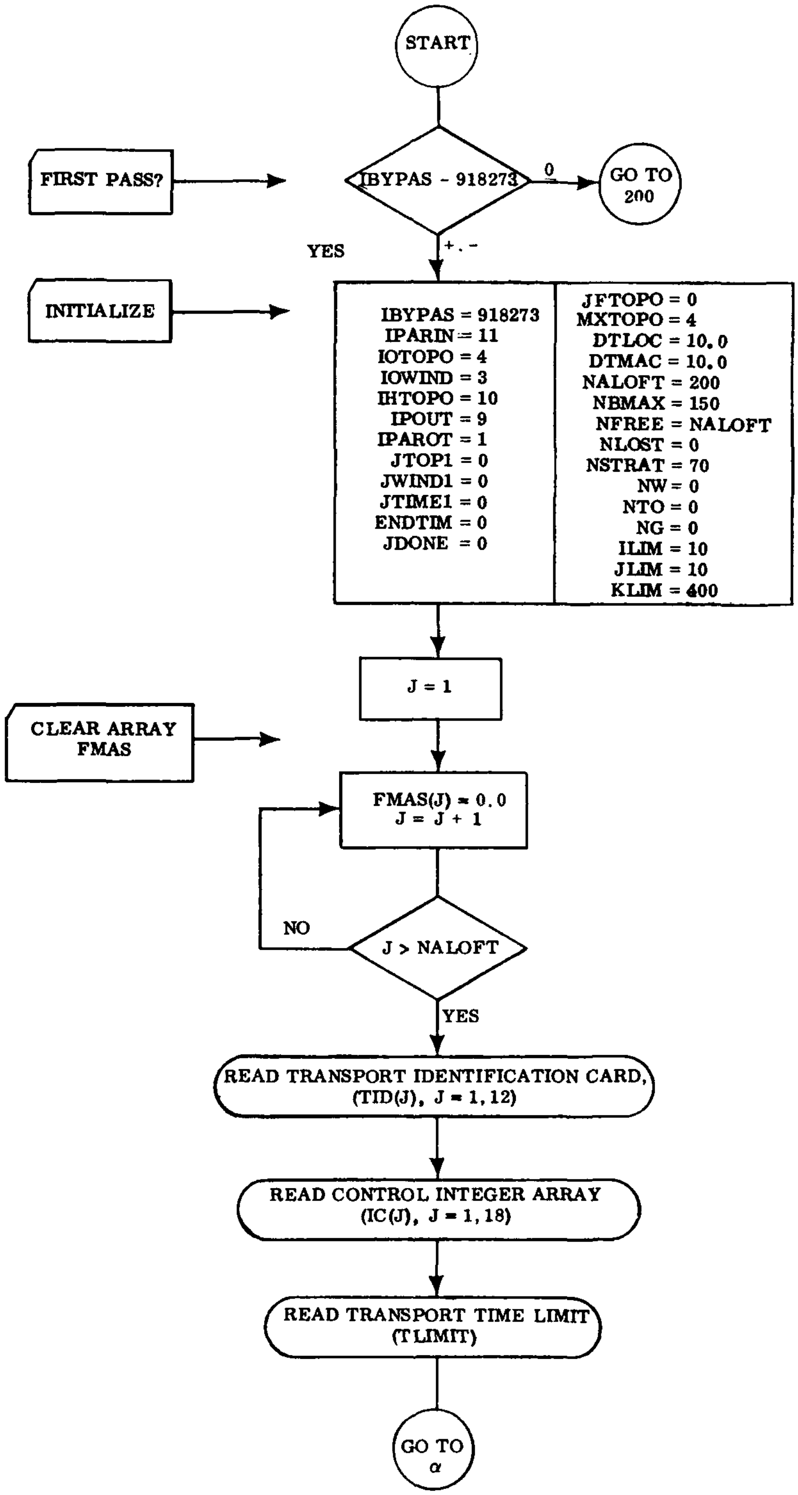
At 205 the program reads a wind-field data set identifier from the system input tape and transfers to the transport output tape (IPOUT) all identifiers and descriptive tables required by the output processor. Next LINK5 prints a title page for the transport run including identifiers and atmospheric data and then transfers to 200 where the transport executive begins.

Statement 200 is the place to which control is immediately transferred upon any entrance to LINK5 except for the first. At 200 TLIMIT and ENDTIM are compared to determine if the processing of the Transport Module has been completed. Note that transport is considered to be unfinished so long as ENDTIM, the time at which the current wind field must be updated is not greater (later) than the user-specified time of transport cutoff (TLIMIT). (ENDTIM is initialized to 0.0 on the first pass through LINK5. It is assigned its true value by the wind description program MKWIND which is called by LINK6, subsequent to LINK5, when LINK5 has set IEXEC = 1 at statement number 400.)

When transport has been completed, LINK5 sets N = NALOFT and calls DUMPP to dispose of any particle descriptions that may remain within core memory. Next, if any particles remain on the time boundary tape they are read in and printed as lost particles for the benefit of the user. Finally at 501 the terminating zero is written on the transport output tape (IPOUT), the comment — TRANSPORT IS COMPLETED, etc. — is written and the executive control word IEXEC is set to zero to cause a transfer to LINK8 of the Output Processor Module (see DASA-1800-VI.)

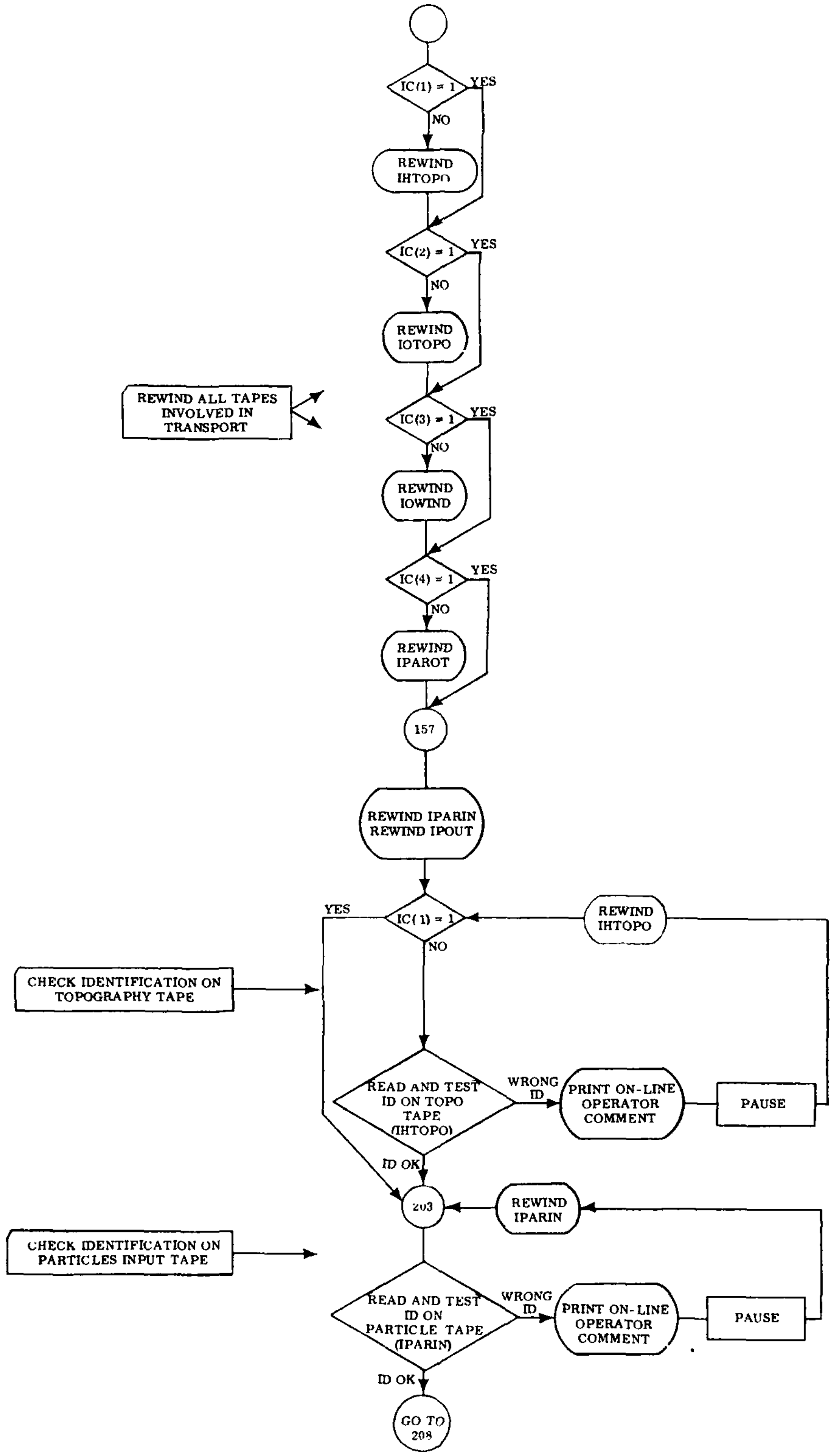


FC-4. General Flow Chart for Subroutine LINK5



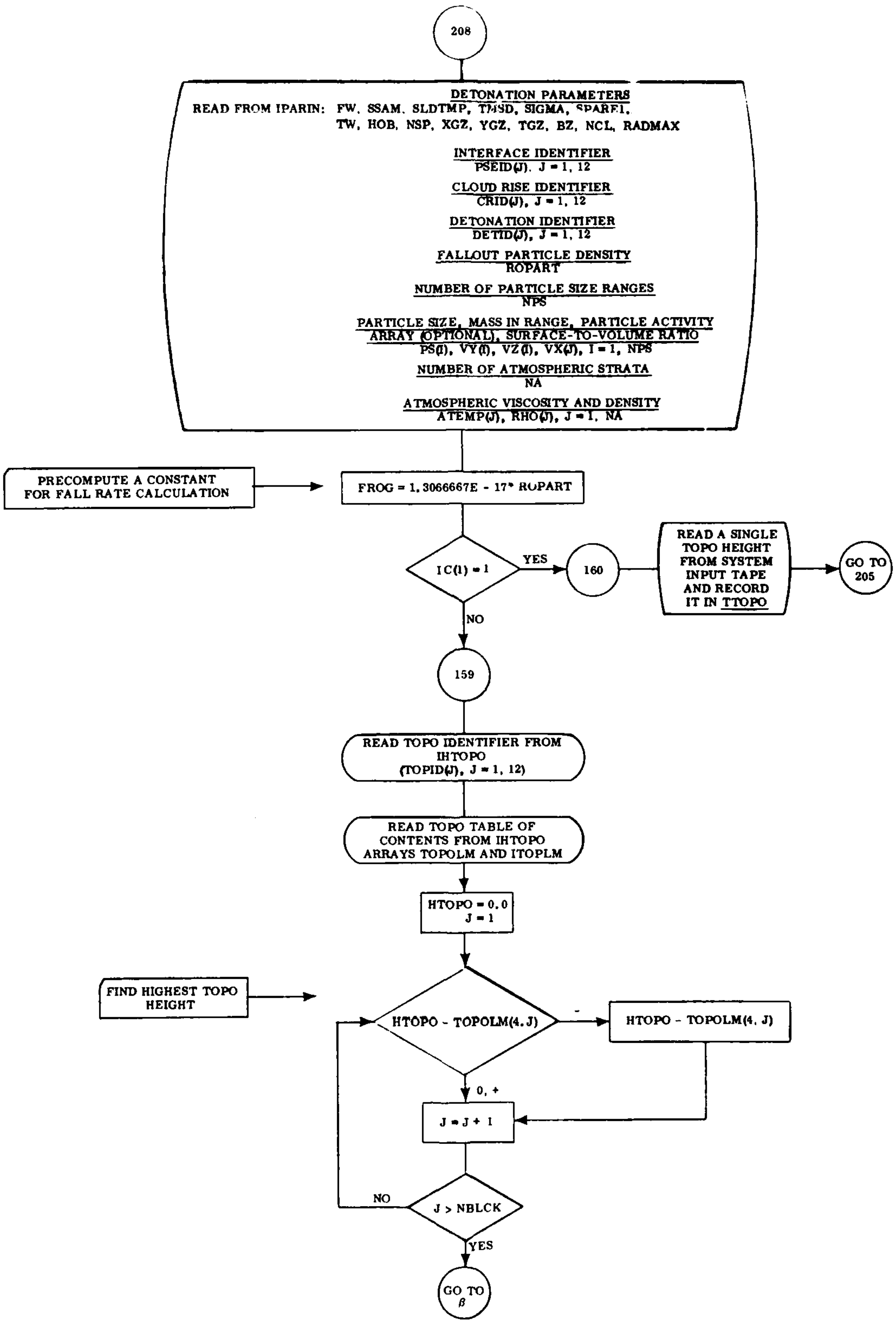
(a)

FC-5. Detailed Flow Charts for Subroutine LINK5



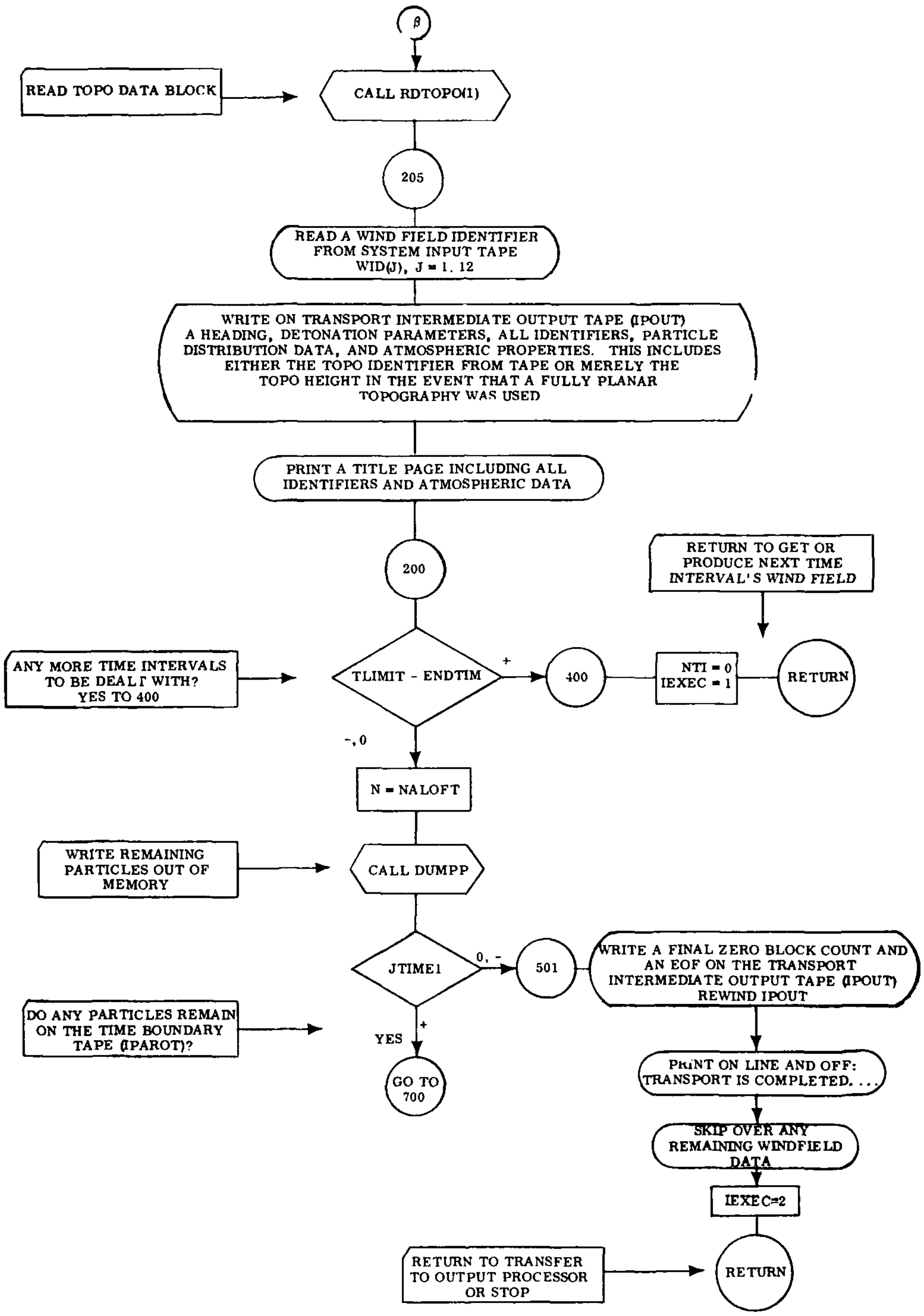
(b)

FC-5. (Continued) Detailed Flow Charts for Subroutine LINK5



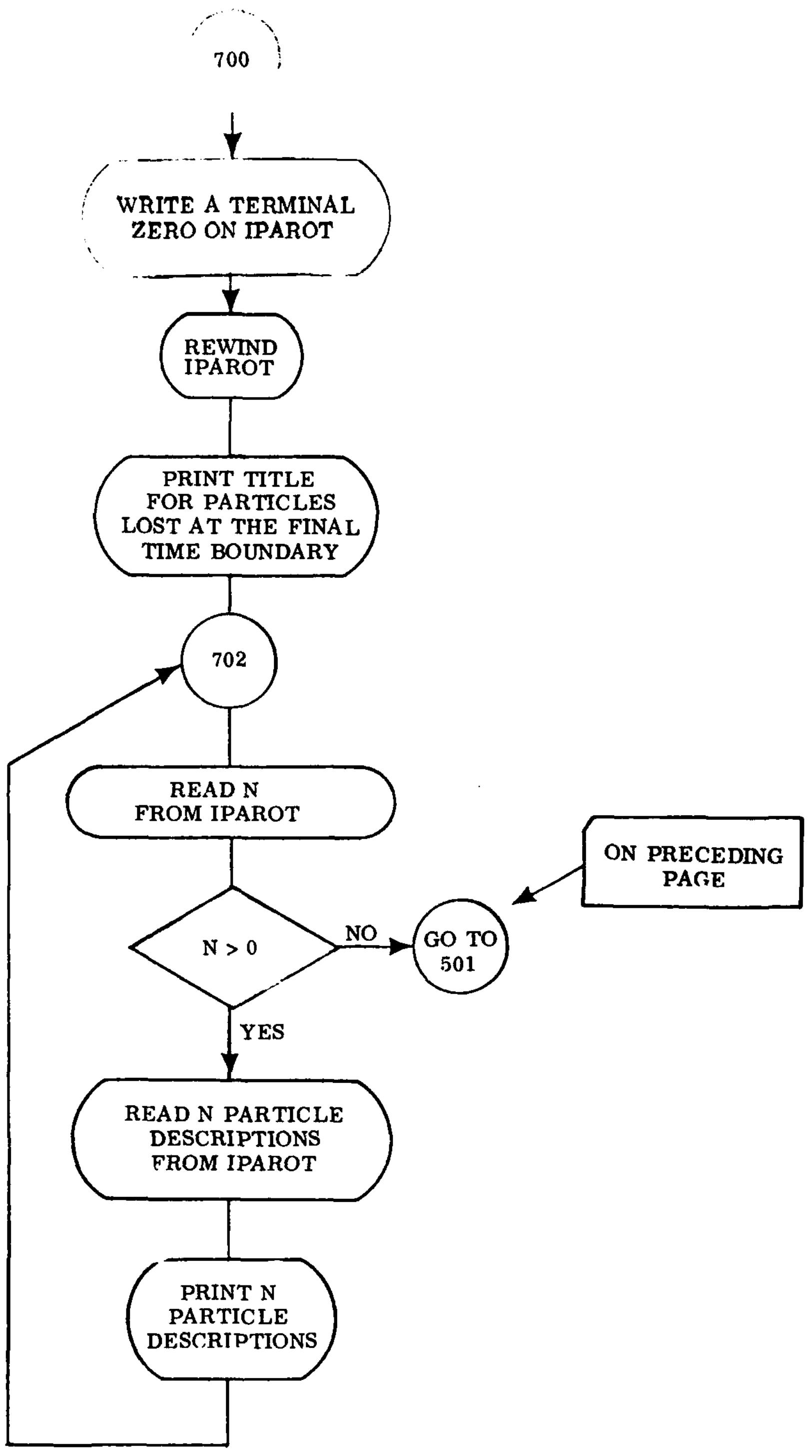
(c)

FC-5. (Continued) Detailed Flow Charts for Subroutine LINK5



(d)

FC-5. (Continued) Detailed Flow Charts for Subroutine LINK5



(e)

FC-5. (Continued) Detailed Flow Charts for Subroutine LINK5

Subroutine LINK6 (no flow chart)

This program merely calls subroutine MKWIND, the wind-field description subroutine. It has been left as a separate subroutine in anticipation of its use as a branch point to select the desired program to be used for the wind-field description.

Subroutine RDCIRS (FC-6)

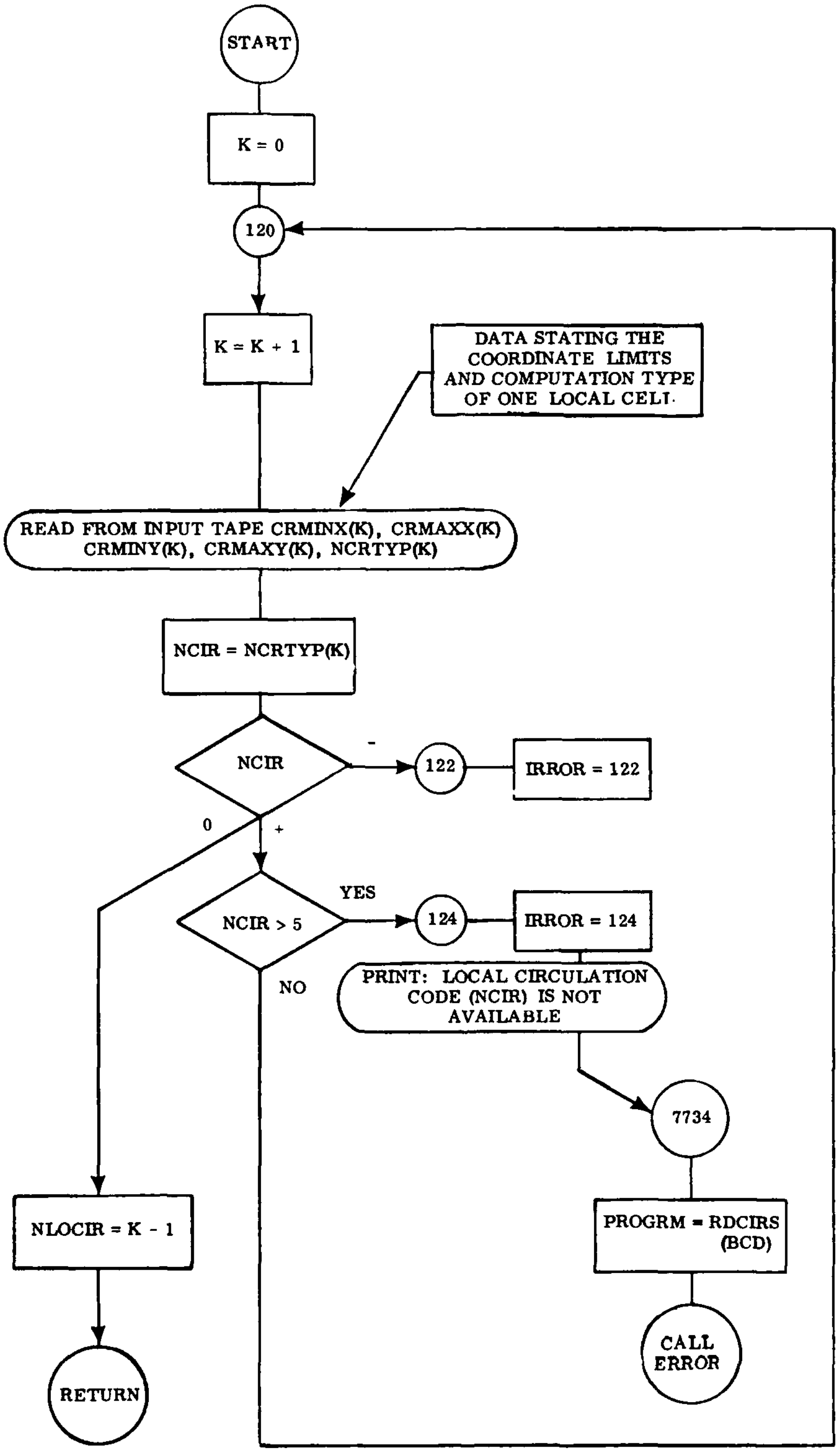
The purpose of this subroutine is to read a set of data which describes the geographical limits of the area covered by each of the local circulation systems that are to be used within the transport. Also, the identification number (NCRTYP(K)) for the computation code (local circulation model) applicable within each of the local circulation cells is read. At the time of this writing only three types of local circulation systems are used:

Identification Number	Program to be Used	Model
1	MTWND1	Mountain wind
2	RGWND1	Ridge wind
3	CBREZ1	Sea breeze

The data are read from the IBSYS input tape, one card image at a time, with all data pertaining to the Kth local circulation area appearing on the same card. A count of card images read is accumulated in variable K and reading is terminated whenever a blank card (NCRTYP(K) = 0) is encountered. At this time the number of local cells for which data have been read is stored in NLOCIR and a return is made to the calling program. An error stop occurs whenever a circulation code identifier (NCRTYP(K)) which is either negative or greater than 5 is encountered.

Subroutine MKWIND (FC-7 and FC-8)

This subroutine forms and stores in core a horizontally and vertically variant wind description on the basis of inputs from the IBSYS input tape. Inputs are as follows:



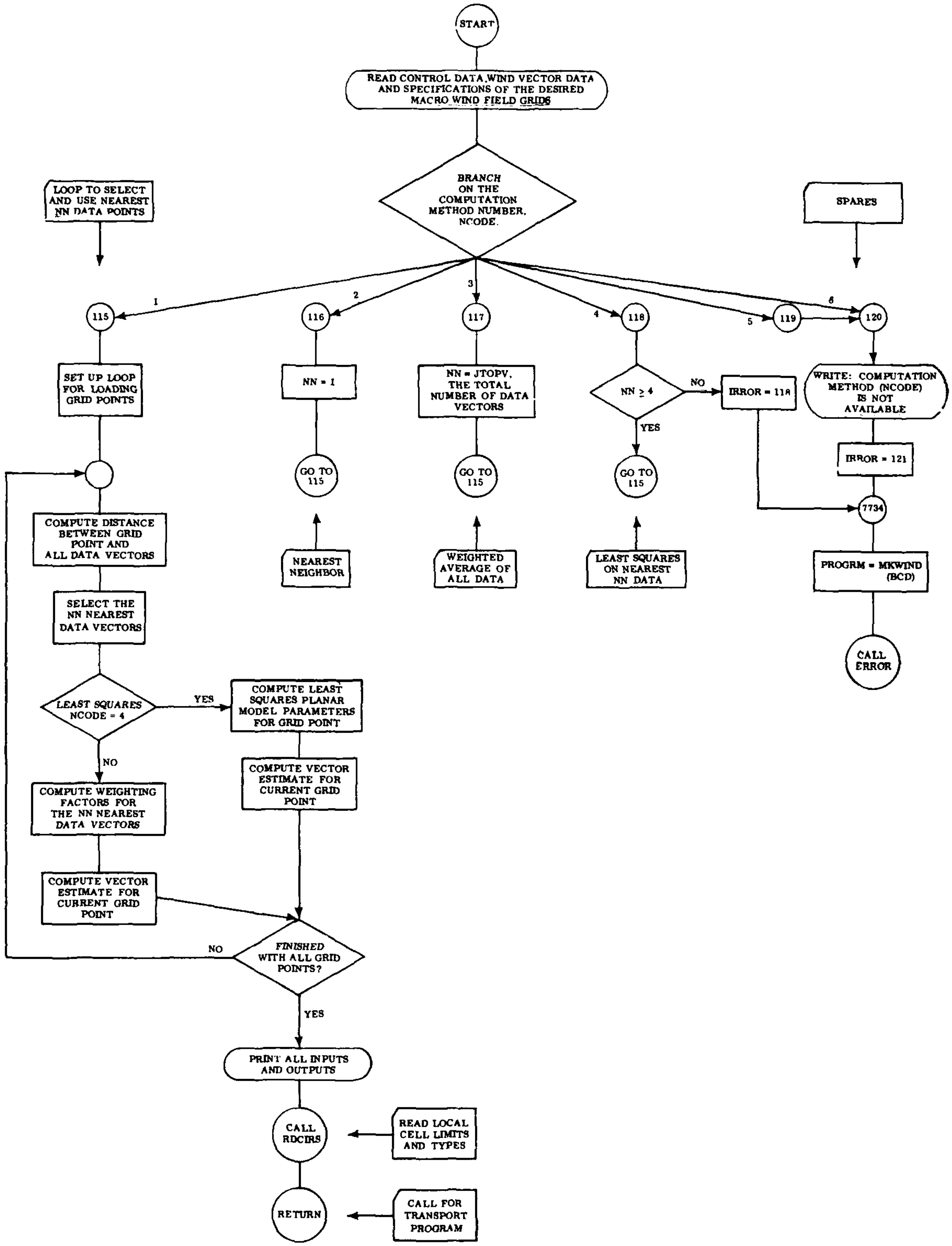
FC-6. Flow Chart for Subroutine RDCIRS

1. Control variables ENDTIM, which gives the time at which wind field to be constructed from the following data ceases to be valid; ALPHA and BETA, which are weighting parameters to be applied to vertical and horizontal distances (see Eq. (21) ff.); NN, which specifies the number of nearest vectors to be used in estimating the wind vector at a grid point; and NCODE, which identifies the desired computational option.

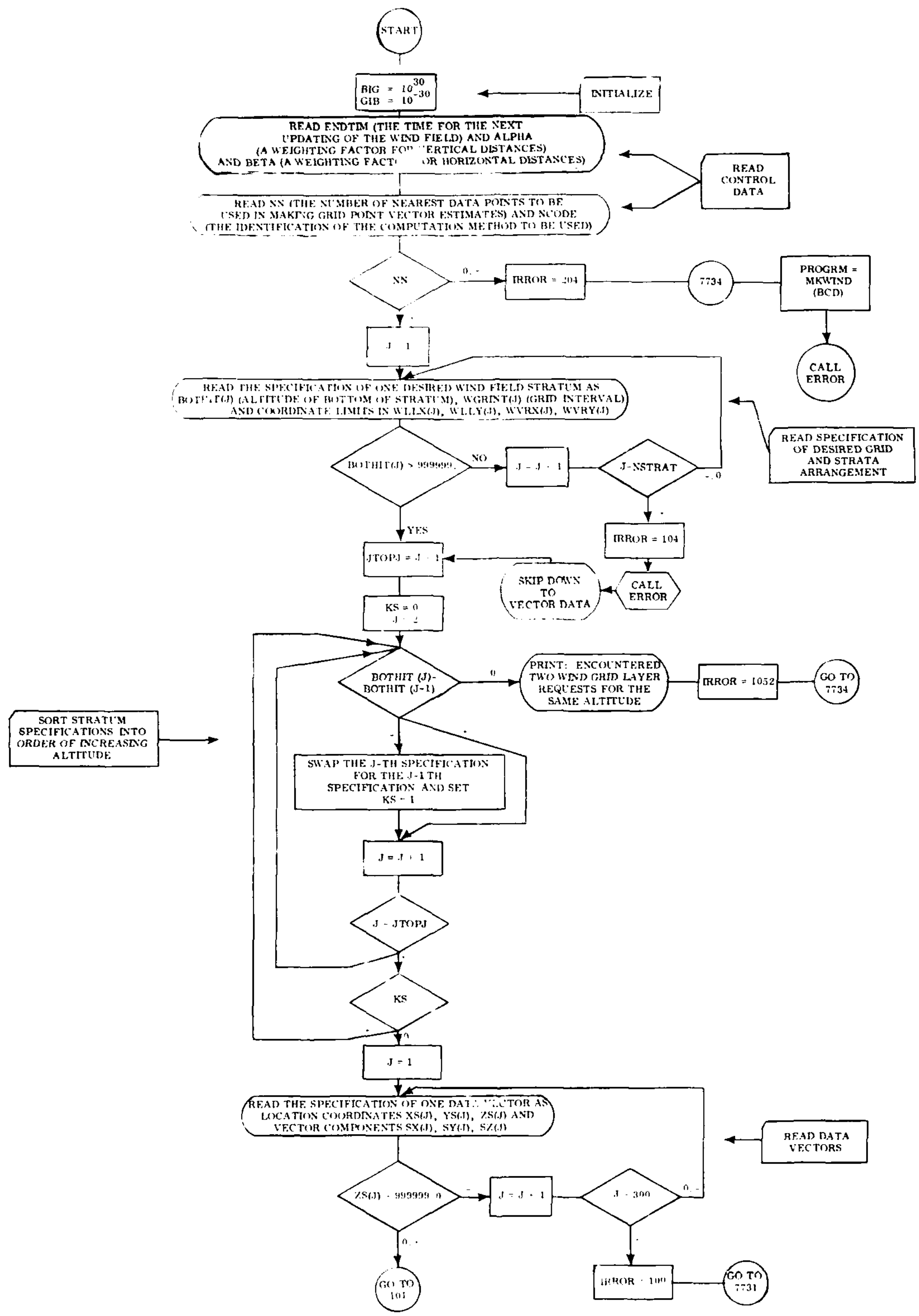
2. Specifications for constructing the wind-field grid for the Jth vertical stratum in the form BOTHIT(J), WGRINT(J), WLLX(J), WLLY(J), WURX(J), and WURY(J); BOTHIT(J) is the height of the bottom of the Jth stratum, WGRINT(J) is the grid interval to be used in the Jth stratum, and WLLX(J), WLLY(J), WURX(J), WURY(J) are lower left corner and upper right corner limit coordinates. Note that each stratum specification is independent of all others. The specification input is terminated when a value BOTHIT(J) $\geq$ 999999.0 is encountered.

3. Wind vector data from which the wind field is to be constructed: ZS(K), XS(K), YS(K), SX(K), SY(K), and SZ(K); ZS(K) is the height of the Kth vector, XS is the east-west coordinate of the Kth vector, YS is the north-south coordinate of the Kth vector, SX(K) is the eastward component of the Kth vector, SY(K) is the northward component of the Kth vector, and SZ(K) is the upward component of the Kth vector. The vector reading operation is terminated when a value ZS(K) $\geq$ 999999.0 is encountered.

A wind-field tape IS NOT WRITTEN by this program. Flow chart FC-7 is a functional flow chart of this program that shows how the four available computation options are arranged to use much of the same code. Flow chart FC-8 presents the details of the subroutine and may be used to follow the ensuing discussion.

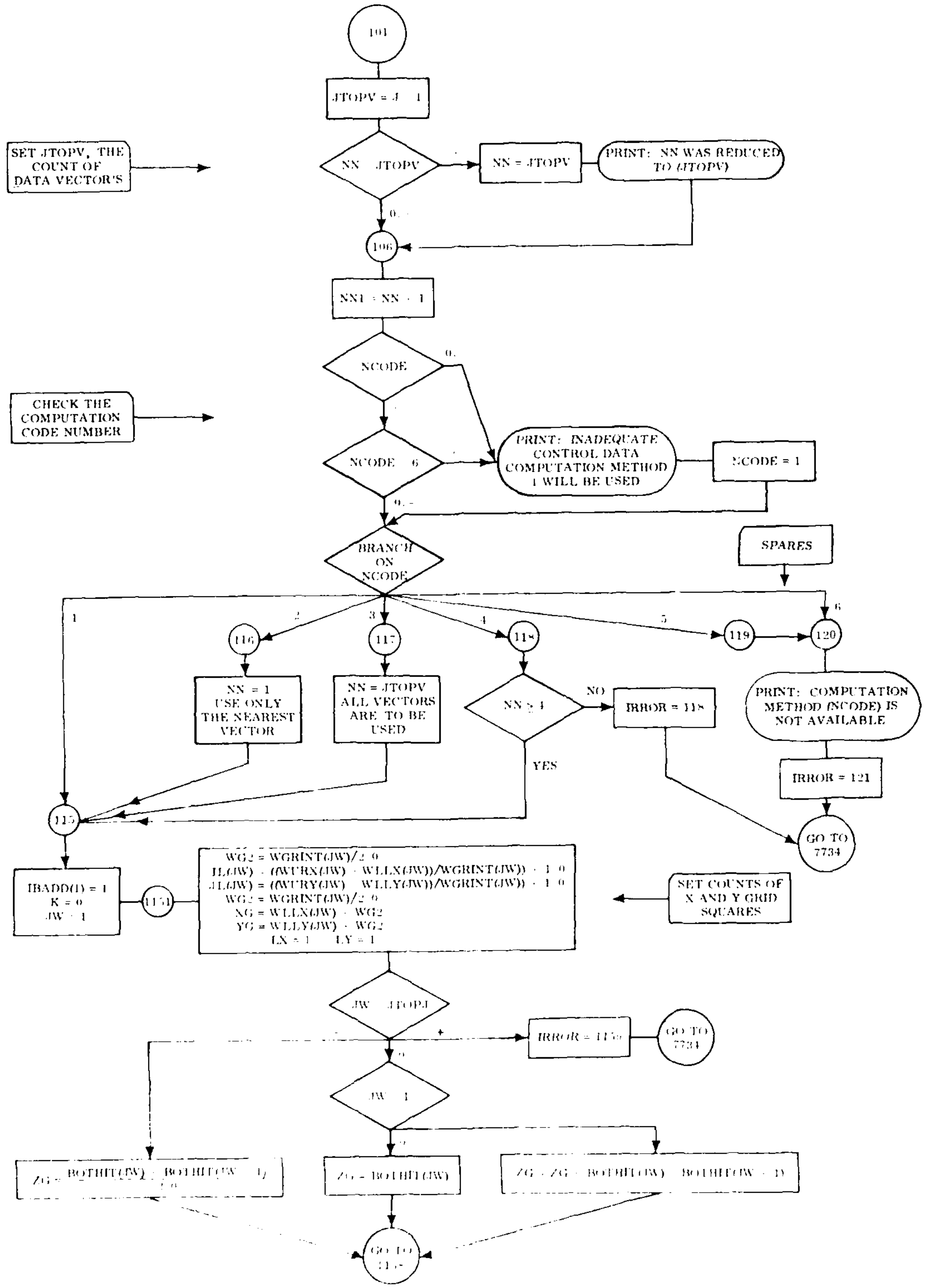


FC-7. Organizational Flow Chart for Subroutine MKWIND



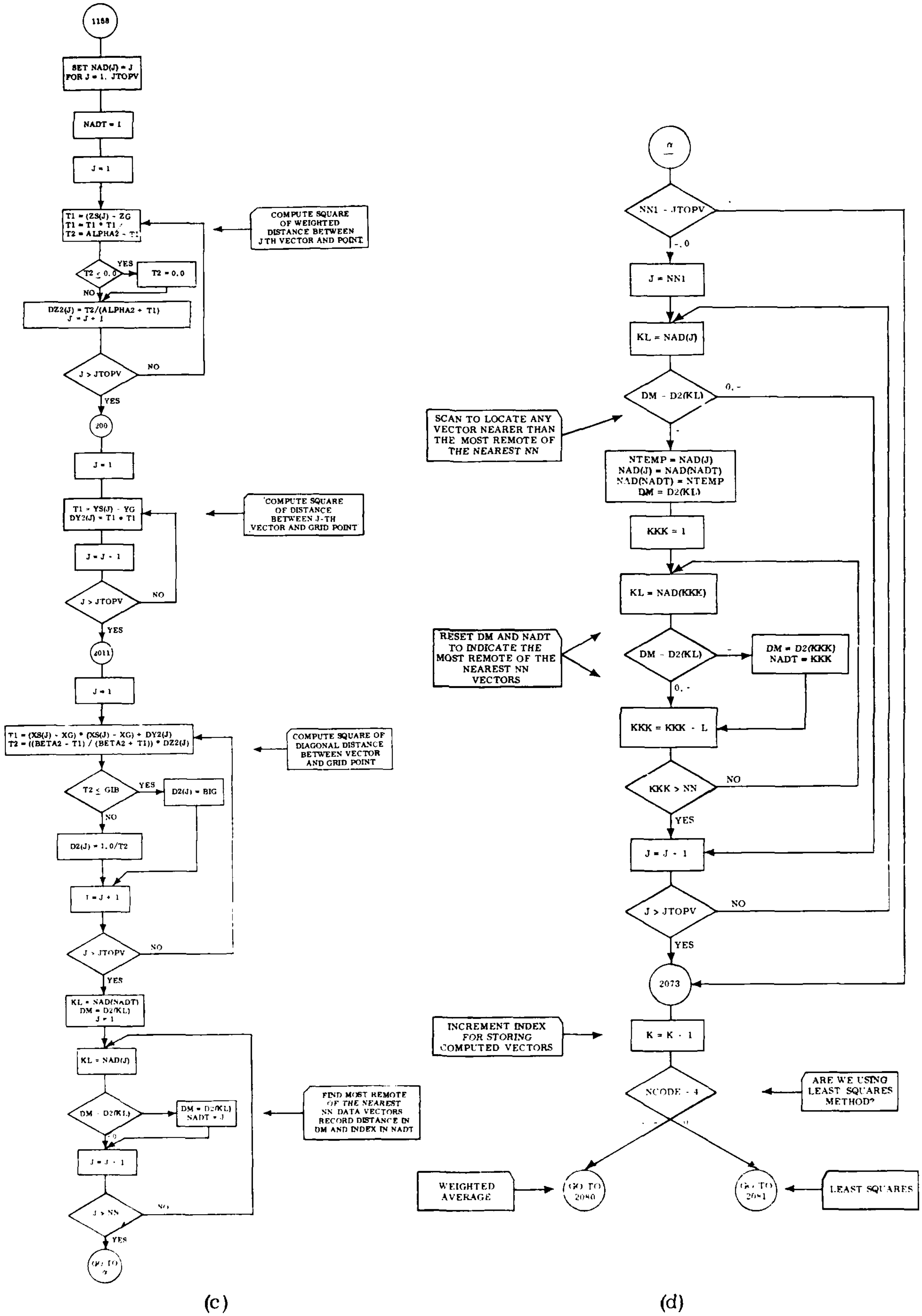
(a)

FC-8. Detailed Flow Charts for Subroutine MKWIND



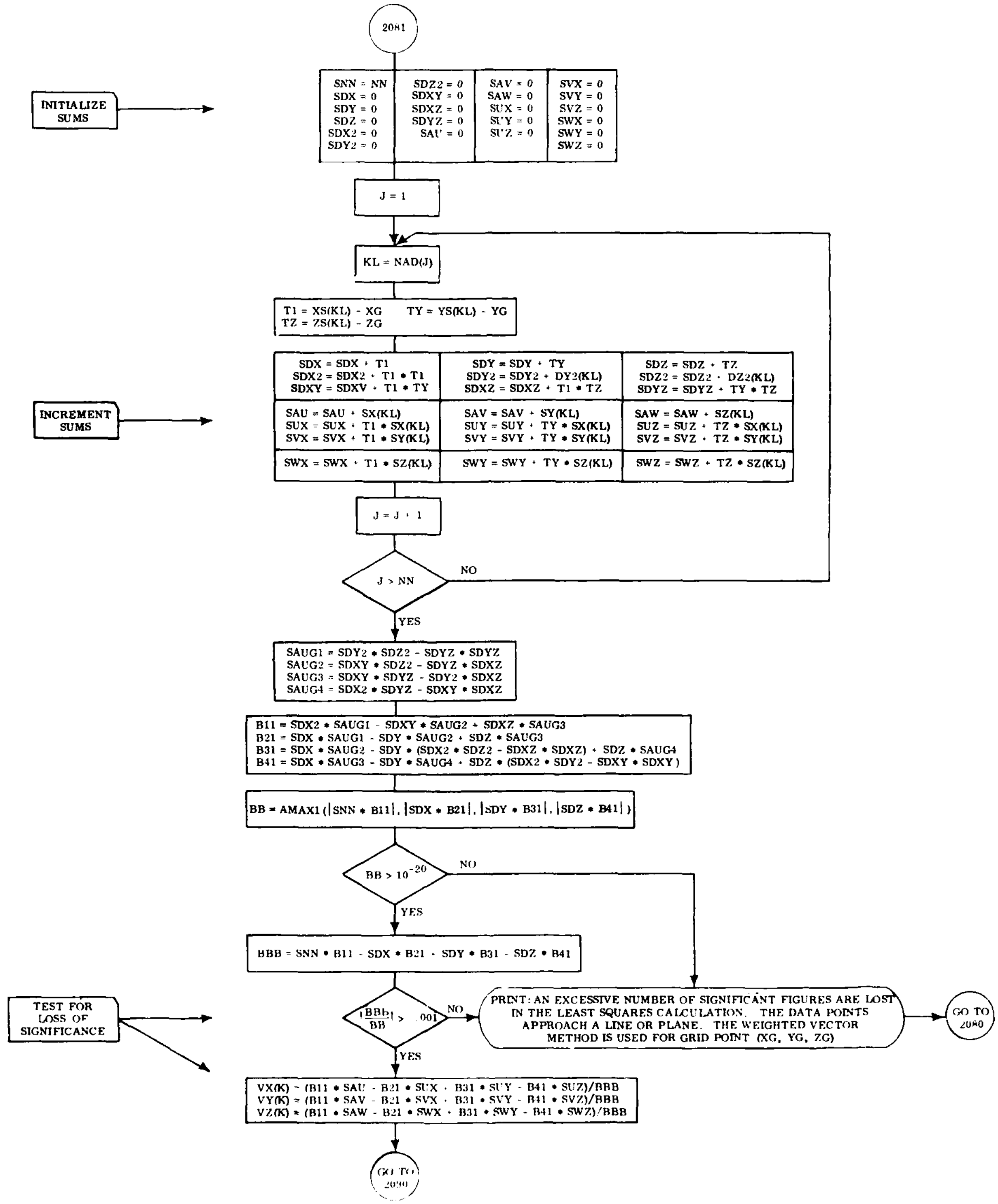
(b)

FC-8. (Continued) Detailed Flow Charts for Subroutine MKWIND



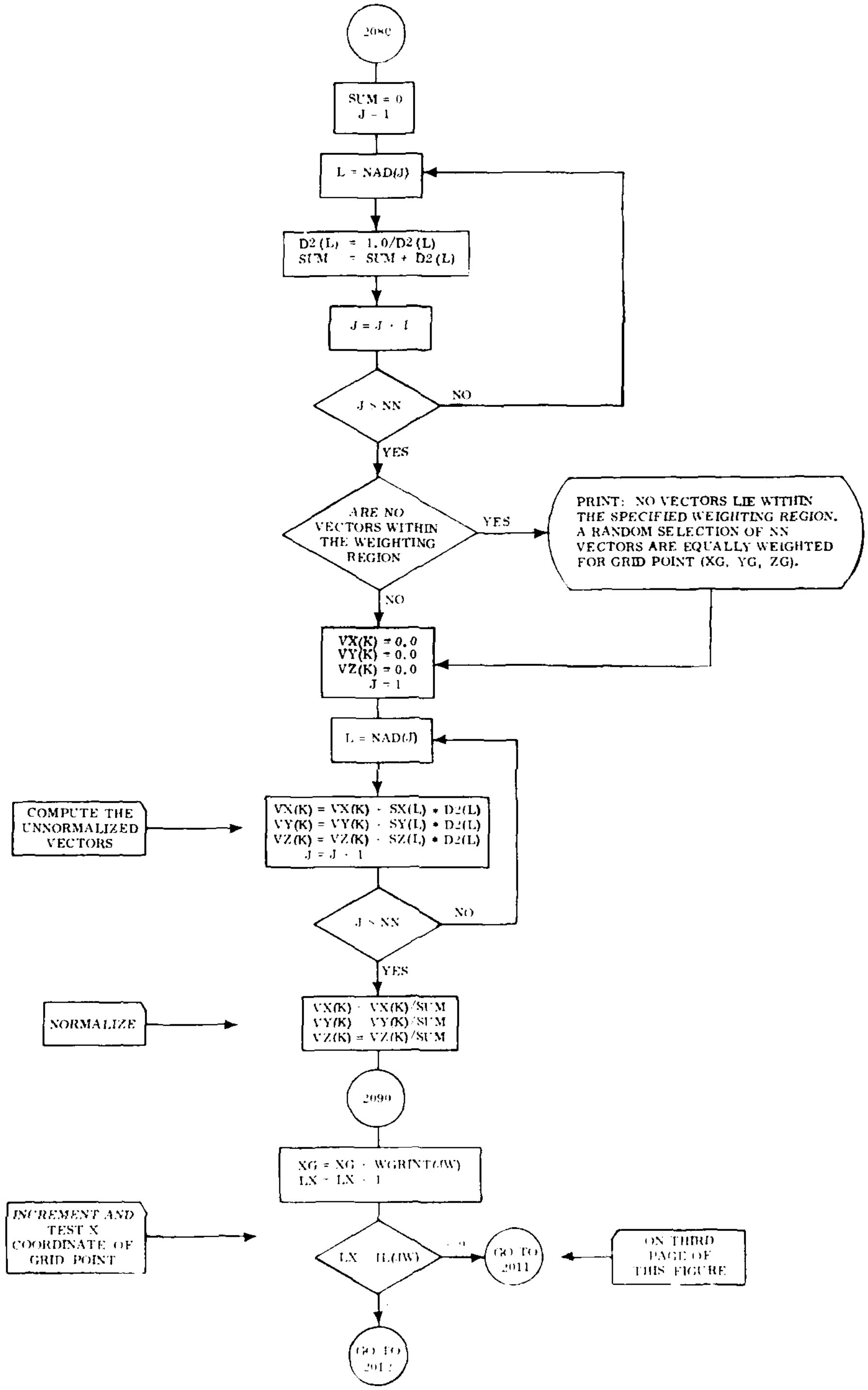
(c) (d)

FC-8. (Continued) Detailed Flow Charts for Subroutine MKWIND



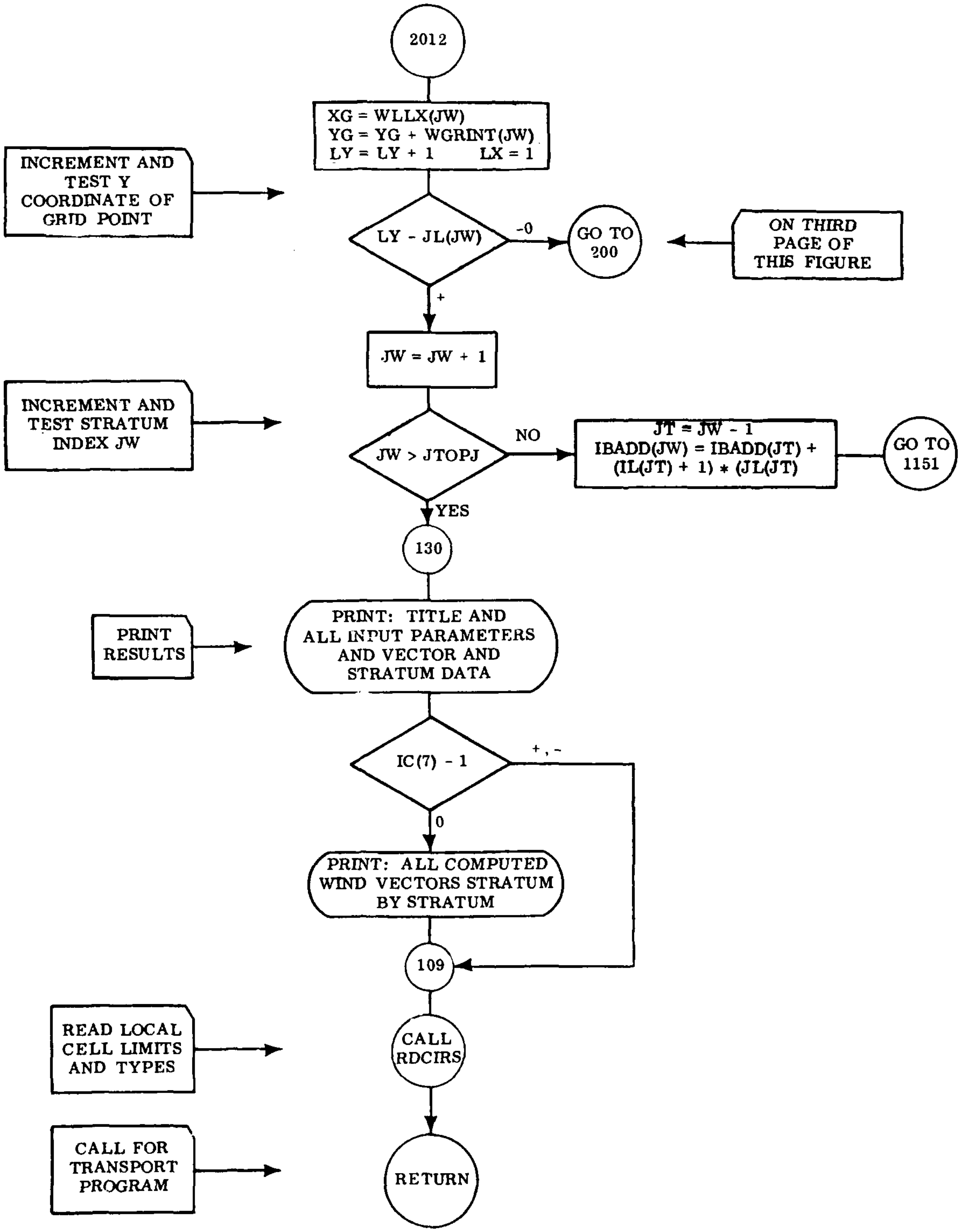
(e)

FC-8. (Continued) Detailed Flow Charts of Subroutine MKWIND



(f)

FC-8. (Continued) Detailed Flow Charts of Subroutine MKWIND



(g)

FC-8. (Continued) Detailed Flow Charts of Subroutine MKWIND

In the beginning the parameters ENDTIM, ALPHA, BETA, NN, and NCODE are read from the IBSYS input tape. If NN is zero or negative, an error stop is printed and the program terminates;* with NN positive, the program transfers control to 2041 where it begins reading the deck of data in which the user specifies the wind-field subdivision structure that he wishes the program to use. This reading operation continues until a card having the value 999999.0 in the field BOTHIT(J) is encountered. If such a card is not encountered before more than NSTRAT (specified in LINK6) cards have been read, an error comment will be written and processing will be continued. When the deck ending card (BOTHIT(J) = 999999.0) is encountered, the variable JTOPJ is set to the number of stratum specifications that have been read. At statement 1054 the data just read are arranged into ascending order of stratum base altitude (BOTHIT(J)) by a pair comparison replacement sort. If during the sort two specifications are found for the same altitude, a comment is printed and an error stop occurs.

When the program reaches statement number 1055 the sort of stratum specifications is complete and the program begins to read a deck of wind vector data. This read operation is of the same form as the read of stratum specifications, but the count of data vectors is recorded in the variable JTOPV. If at the end of the vector-read operation the number of vectors read does not exceed NN, the specified number of nearest data vectors to be used in the computation of each wind cell vector, NN is reset to JTOPV and the computations will continue after a comment is written.

---

*The use of NN with the preferential weighting method is somewhat redundant in that the weighting procedure automatically limits consideration to only those observations that lie within specified distances, horizontal and vertical distances being specified independently of the wind-field grid points. Normally one should specify NN to equal the total number of input wind vector observations when the preferential weighting method is used. In any case, only the NN wind vectors closest to each grid point will be used in determining each wind field vector.

Continuing from statement 106, the program determines if 0 < NCODE $\leq$ 6 and, if so, branches on NCODE to make preparations for further processing by the chosen computation method (see flow chart FC-9 and Table 8). After all transfers are made to an available computation method via NCODE, control eventually returns to statement 115.

At 115 initializations are made for a loop that will fill in sequentially all of the wind cells of the specified strata and will record the vector values in the arrays VX(J), VY(J), and VZ(J). The storage index of the first entry in the wind-field description arrays for the first stratum is set at 1 (i.e., IBADD(1) = 1), the stratum index JW is set at 1 to designate the first and lowest stratum, and the vector storage index, K, is initialized at 0.

At 1151 IL(JW) and JL(JW), the number of wind cells in stratum JW in the X and Y directions, respectively, are computed. The constant 0.9999999 is added before truncation of the floating point value to an integer to insure that the cells will always cover the complete area specified by the user. Next, further initialization occurs and the grid point coordinates XG, YG, and ZG are set at the center of the first cell of the stratum. Note that special treatment must be given to the Z coordinate of both the top and bottom strata.

At 1158 the program begins to set up the array NAD, which is used to store address indices of wind data vectors that are nearest neighbors to a particular wind field grid point. It first sets all NAD(J) = J, J = 1, JTOPV, to provide indices for the full set of data points and to provide an initial set of nearest data points. Note that in the beginning the NAD do not reference data vectors in order of increasing distance from the grid point (XG, YG, ZG), but merely provide an initial input to a sort procedure that will provide such an ordering. Initially, we set NADT, the index of the NAD representing the data vector which is the most remote (from the grid point) of the nearest NN vectors, at 1, since prior to the first pass through the distance sorter all NN data vectors are equally likely to be the most remote of the set.

Next, in three DO loops ending at 199, 201, and 202 we compute weighting factors related to the vertical and horizontal distances between the current grid point and each data vector point, and store the result as a measure of remoteness in the array D2(J) which is parallel to the data vector arrays. We attempt to minimize computation by keeping weighting factor components in parallel arrays DY2 and DZ2 during the evaluation of a wind field.

After 202 we find the address of and distance to the most remote point (from the grid point) of the currently specified NN "nearest" data points. (These are the points whose addresses (indices) are given by NAD(1) through NAD(NN). This maximum distance is stored in the word DM and NADT is set such that DM = D2(NAD(NADT)).

At 2072 we may scan the data vectors that are not within the set of nearest NN to ascertain that there is no vector nearer than the most remote of the nearest NN. If one is found, its address must be inserted in the place of the most remote and adjustments made to NADT and DM. (This somewhat obscure procedure is intended to achieve efficiency by making extensive use of the strong correlation that will exist between the interpoint distances in the array D2 as the calculation progresses from one grid point evaluation to the next.) At the end of this procedure (after 2073) the nearest NN data vectors have been located and their addresses are recorded in NAD(J), J = 1, NN.

The grid data storage index K is next incremented and a second branch is made on the basis of NCODE.

If NCODE = 4, we branch to the least-squares method which uses the NN nearest data points under the restraint that NN $\geq$ 4. Rectilinear coordinates of the points are determined with respect to the grid point at which we wish to calculate the wind field. Next, the elements of the normal equations matrix are computed and the complementary minors B11, B21, B31, and B41 are determined. If BB, the absolute value of the largest of the four products of the cofactors times their corresponding matrix elements, is not less than 10^{-20}, the determinant BBB is computed and the ratio $\left|\frac{\text{BBB}}{\text{BB}}\right|$ is found. If BB is less than 10^{-20} or the ratio $\left|\frac{\text{BBB}}{\text{BB}}\right|$ is less than 10^{-3}, an excessive number of significant figures are lost in the least-squares calculation for this particular grid point (i.e., the normal equations matrix is essentially singular), and the code prints this information and then branches to the preferential weighting method (as though NCODE = 1). If neither of these cases occurs, the wind velocity vectors are computed and stored using index K.

If the preferential weighting method is to be used (NCODE = 1), a transfer is made to 2080 where weighting factors are computed and summed for the NN nearest data vectors. Next (after 214), the three vector components are computed as a weighted average of the vectors at the NN nearest data points and the results are stored in the arrays VX, VY, and VZ under the index K.

The least squares and preferential weighting methods converge again at statement 2090 where the indexing and control scheme begins. First, the X coordinate of the current grid point is incremented, and if the new grid point is still within the desired wind field, the program returns to 2011 to begin the evaluation of its vector. If the new X coordinate is beyond the wind-field range, X is reset and Y is incremented and tested. If both X and Y end up beyond the range of interest, the program moves on to the next higher stratum. When all strata have been evaluated in full the program branches to 130 where all input data are printed, and if desired (IC(7) = 1), *all computed wind cell vectors* are also printed. Finally at 109 a call is made to subroutine RDCIRS which reads a set of data describing the limits of all local circulation cells and the types of circulation systems within them. Upon return from RDCIRS, MKWIND returns to the monitor so that transport may be continued using the newly updated wind field description.

Subroutine LINK7 (FC-9 and FC-10)

This subroutine is the primary transport program. It accepts a tape of transportable particles and transports them, stopping only when it has no more particles to transport or when a new version of the wind-field description must be prepared. The first action of LINK7 is to interrogate the input parameter IC(6) (see Table 6) to ascertain whether the transport traces have been requested. If IC(6) < 1, no traces are printed. If IC(6) = 1, the complete in-core particle arrays are printed after each block of new particles is read in from tape IPARIN. Each line of this output consists of XP, YP, ZP, TP, PS, and FMAS. If IC(6) > 1, at the beginning of the main transport loop this same information is printed for each particle in turn, and in addition after each transport increment the quantities XP, YP, ZP, TP, TSM, NTI, NG, NTO, NW, NLOST, and IR (see the LINK5 glossary for definition of these quantities) are printed for each particle. In the execution of the Transport Module LINK7 is always preceded by a call to LINK6, the wind-field description generator program. Since the data peculiar to each existing local circulation system (as defined by RDCIRS which is called by LINK6) must also be updated before transport begins, LINK7 first transfers to each of the required local circulation codes to cause them to read their data. If there are no local circulation systems in use, or after reading the data for the required local circulation codes, LINK7 continues at statement number 510. There, assignments are made for parameters IT and ITT according to the value of IC(1) to control the transport of particles as they approach the topography (see Table 6).

Next, at 1000 the program makes preparations to enter the main transport loop. IS and IF are set for use as particle index limits of the main transport loop. If JTIME1 is zero a regular entrance is being made, but if JTIME1 is negative, there may be transportable particles in the particle arrays left over from the preceding pass (prior to the most recent updating of the wind field). In the latter case the main transport loop is entered with index limits set to cover the full particle array so that all left-over transportable particles will be dealt with.

If JTIME1 is zero or positive (no particles remain at the time boundary), the program at statement 1112 begins processing transportable particles from tape IPARIN. Note that the logical tape number recorded in parameter IPARIN is not always the number of the unit on which the data was originally received from LINK4. IPARIN always identifies a tape containing transportable particles, but these may be either the original input from LINK4 or a recirculation of particles that were written onto some one of the secondary memory units IPAROT, IOWIND, or IOTOPO. At 1112 LINK7 reads a block count, N, from IPARIN; if N is positive and N particle descriptions can fit into the particle arrays, subroutine DUMPP is called to prepare a place for the N particles. The loop index limits are reset to cause processing of the incoming N particles and the N particles are read from IPARIN. Finally, NFREE, the count of empty spaces in the particle arrays, is decreased by N and control is transferred to 1001 where the main transport loop begins. In the event that the block count was zero, the end of the set of transportable particles on IPARIN has been reached and a transfer is made to 100 where preparations are made to either recirculate data from secondary memory tapes or transfer to the transport executive (LINK5). At this point LINK5 will either call for updating of the wind field or for the Output Processor Module.

Continuing this explanation at statement 100 we see that if off-topo particles exist (JTOP1≠0), the program selects the next needed topo file, fetches it from IHTOPO, and subsequently returns to the main transport loop (1001) to make use of the newly acquired topo data.

At 104 a similar treatment is given to particles that may have gone beyond the in-core wind field. However, since currently existing wind field programs do not make use of a tape wind field file, the code beginning at statement 130 will not be executed.

At 200 preparations are made to return to the transport executive where a call is provided for either the output processor or the wind-field program.

The main transport loop (between statement numbers 1001 and 160) uses the index J to identify the current particle description. It begins by determining if the current (Jth) particle is to be transported. To be transportable it must be identified by a positive FMAS(J) and O < TP(J) < TLIMIT; the program avoids all untransportable particles by transferring immediately to the loop control point at 160 whenever one is encountered.

At 195 NLOCIR, the number of local circulation systems in use, is tested. If any are in use, the Jth particle is tested to see if it is within or above any local cell, but if there are none in use, this test is avoided. If a particle is found to be in or above any local cell, LOTRAN is called to transport the particle until it passes beyond the cell's vertical boundary planes. Since a particle may pass out of one local cell and immediately into another, control cannot be returned to the main body of the transport loop (at 1950) until it has been ascertained that the particle is no longer within or above any of the local cells.

At 1950 arguments are set for a call to subroutine GETWND at 1961. GETWND gets the macrowind-field vector that applies at the point whose coordinates are in arguments XX, YY, ZZ. If upon return the index JWAD is set negative, the needed macrowind data is not available and the particle must be considered lost to the computation. However, if JWAD is positive, a correct retrieval has been accomplished and the program continues to 196.

At 196 the particle settling rate is computed for the current particle by the call to FALRAT and VPZ is set as the net vertical particle velocity component. Next, distances are computed from the particle position to each of the vertical planes that bound the macrowind cell containing the particle. Time of flight is then computed to the north-south and east-west boundary planes and also to the horizontal plane which would be first encountered.

At 1711 the time of flight to the first intersection with a local circulation cell is computed, but note that if NLOCIR (the number of local cells in use) is zero much code is avoided and a transfer is made directly to 172. In the event that intersections with local cells must be sought, a DO loop sequentially computes the time of intersection to each of the defined cells keeping track of the time of flight to the first intersection (if there is one) in variable CIRMIN.

At 172 the program selects the time of flight to the first of all intersections with boundary planes; if that time of flight is excessively small, special steps must be taken (at 1811) to assure that program efficiency is not lost. Asymptotic approaches to boundaries are avoided by never using a time step smaller than EPSIL. Oscillations at boundaries are avoided by treating the occurrence of two sequential, excessively small time steps as a sign of oscillation and by subsequently avoiding movements to or from the plane of oscillation.

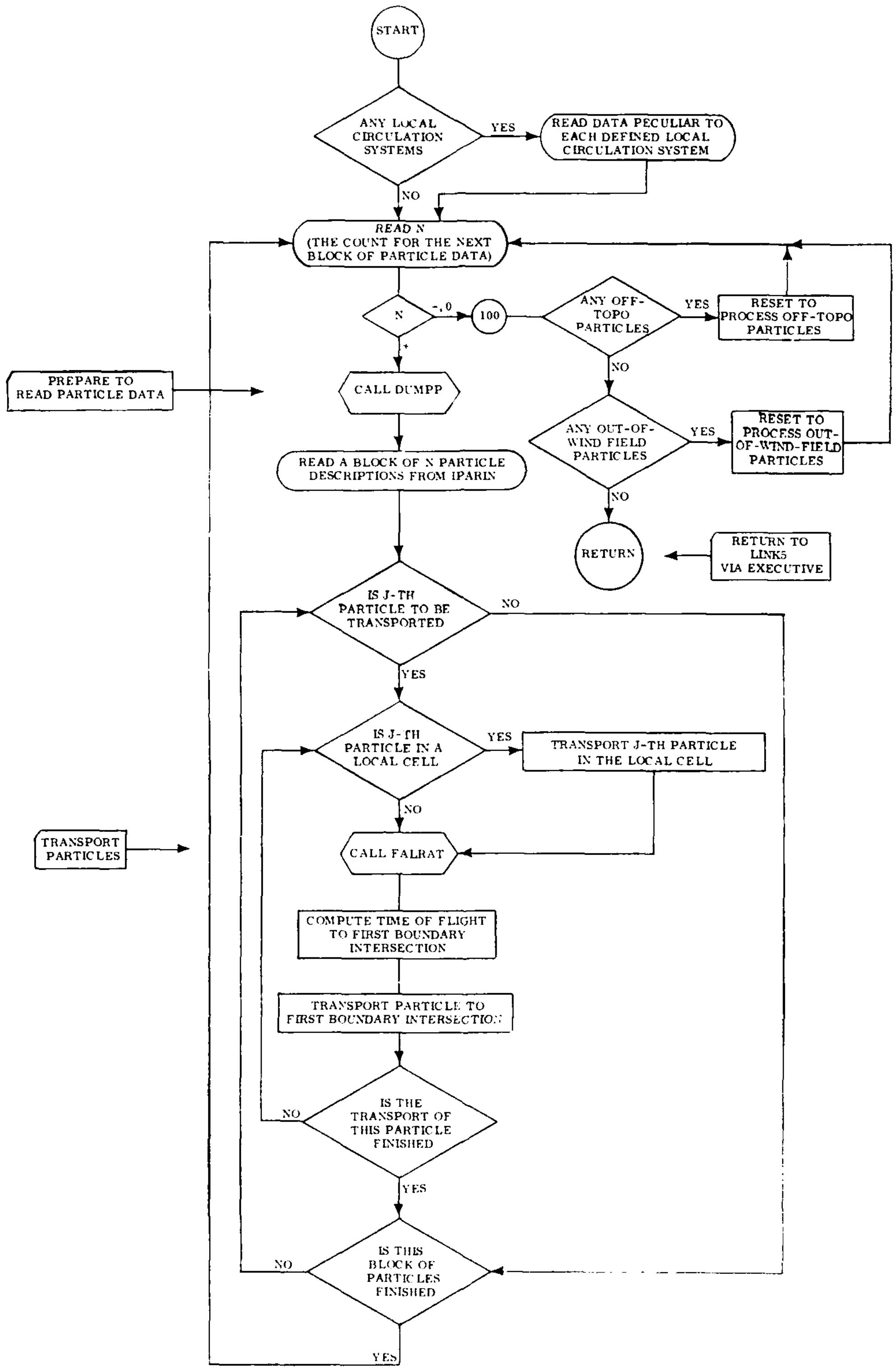
Continuing at 3067 a comparison of particle altitude and maximum topo height is made and if the particle is above TTOPO, simple linear transport occurs. However, if particle altitude is below TTOPO, a special loop beginning at 1814 is used to transport the particle by constant time steps (DTMAC) for the interval TSM or until impact on topography occurs. It should be noted that the main transport loop never moves any particle descriptions within the particle arrays. It does, however, mark the status of particles within the arrays using the sign of parameter FMAS(J) and the sign and value of TP(J) in accordance with the conventions described in Table 2.

Subroutine GETWND(XX, YY, ZZ, JWAD, JW) (FC-11)

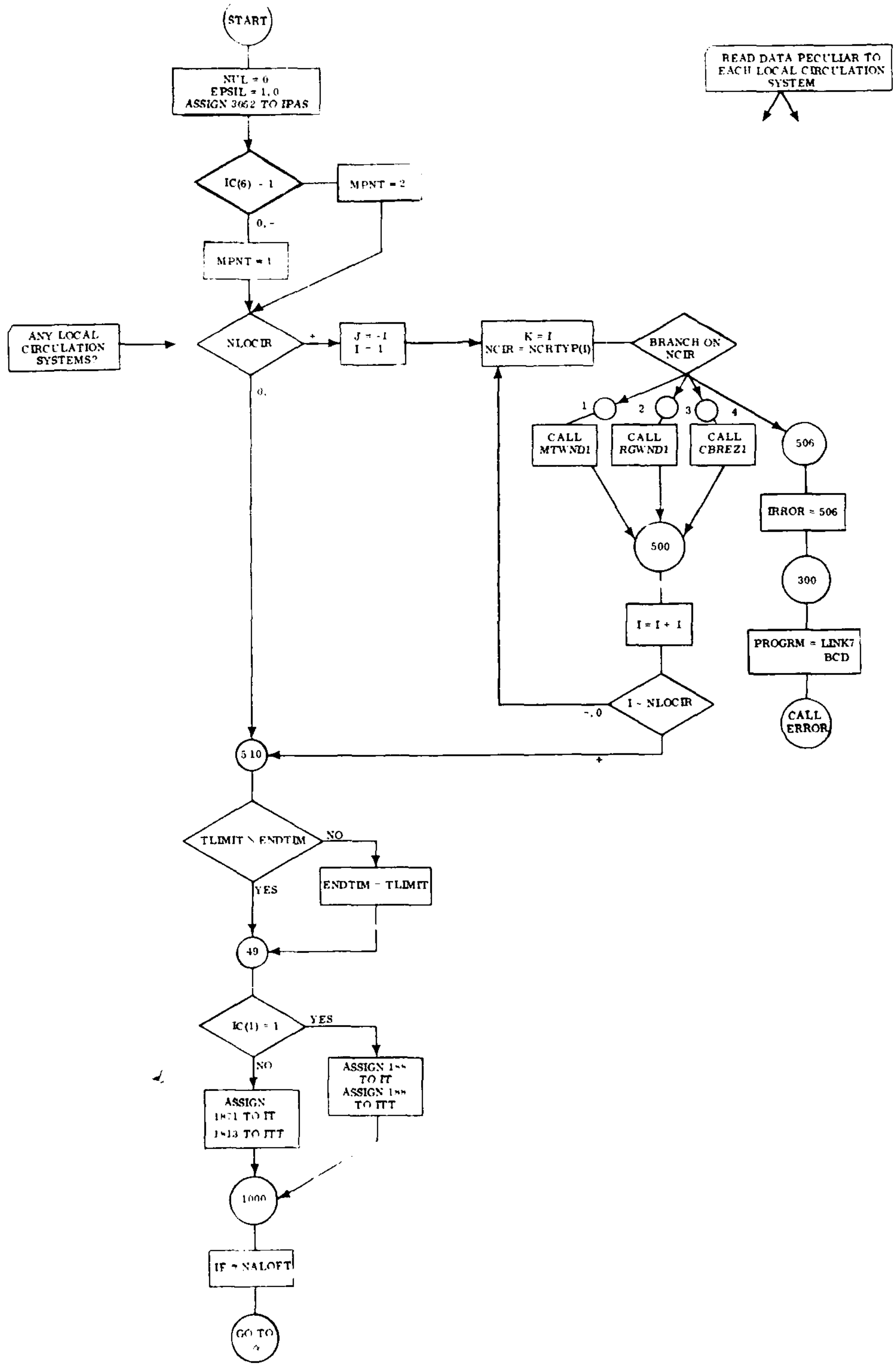
The purpose of this program is to determine the index to be used for retrieving the macrowind vector that applies at a particle position point XX, YY, ZZ. The desired index is stored in the argument JWAD upon return. JWAD is set negative in the event that the point XX, YY, ZZ is outside the volume for which the macrowind field has been specified.

The computation of index JWAD consists of two parts: first, the computation of JW, the index of the wind stratum containing the point; and second, the actual computation of the retrieval index JWAD using information describing the data structure of the JWth stratum. In the event that it is known that the value of the JW last computed is still valid, the computation of JW can be avoided. The calling program must only set the sign of the valid JW negative to cause GETWND to avoid recomputing it.

The execution of GETWND begins by testing the sign of argument JW. If the sign is negative, it is set positive and a transfer is made to statement 270 where JW is used to compute JWAD. If JW is nonnegative, a two-boundaried binary search is used to set JW. In that search JT is initialized as the index of the top wind layer and JW is initialized as the index of the bottom wind layer of the whole macrowind field. A test index (JTEST) is computed as the (truncated) mean between JT and JW and the program determines whether the point is above or below the bottom height (BOTHIT(JTEST)) of the test index's wind layer. If the particle is above the bottom of layer JTEST, the bottom index JW is reset equal to JTEST to indicate that the particle has been found to lie in some layer from JTEST(JW) through JT. Had the particle been below the test layer, the top index would have been reset to equal the test index. The algorithm proceeds by converging iteratively on the layer containing the particle and exits when JT and JW are separated by unity at which point the particle must be within the JWth layer.

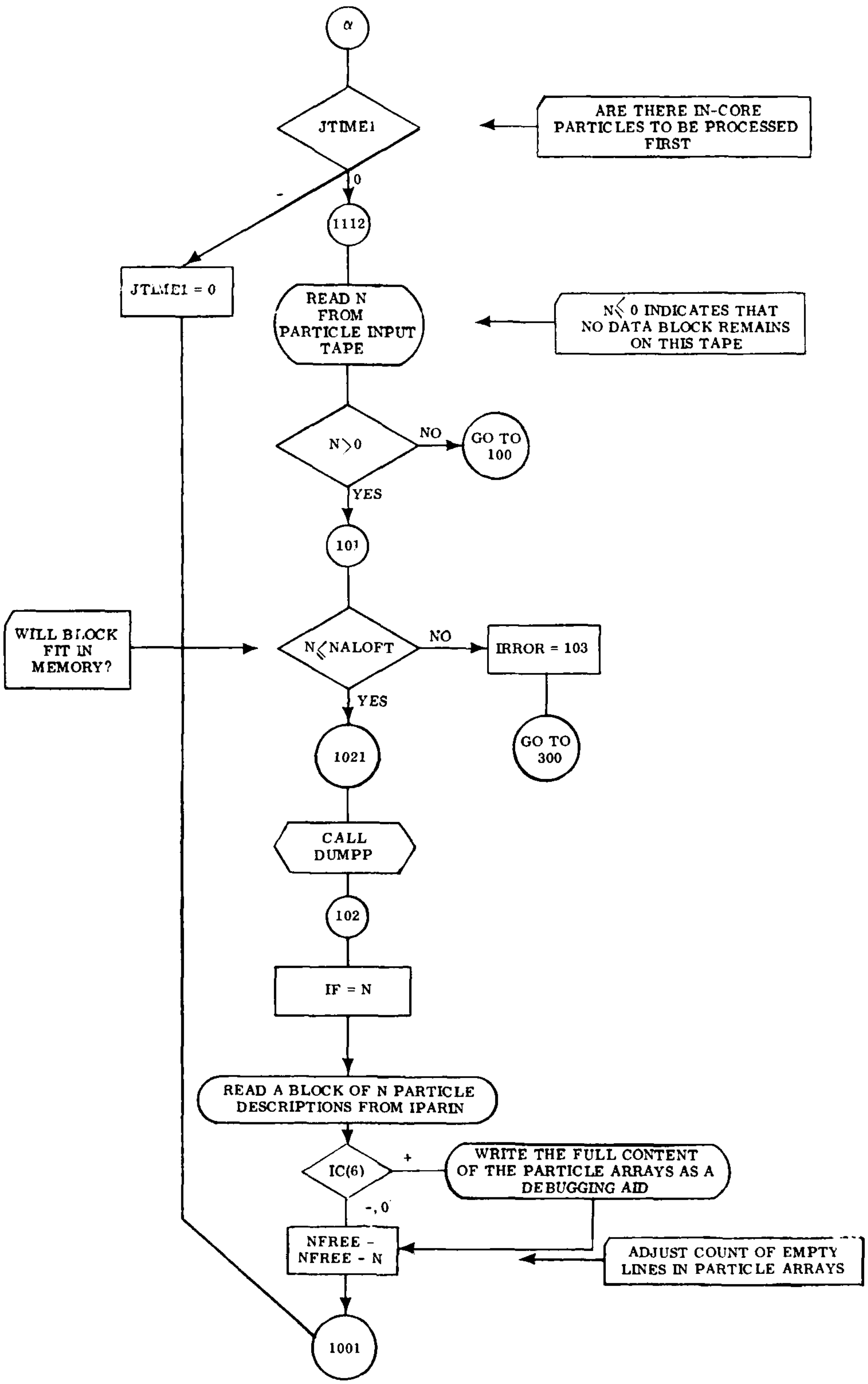


FC-9. General Flow Chart for Subroutine LINK7



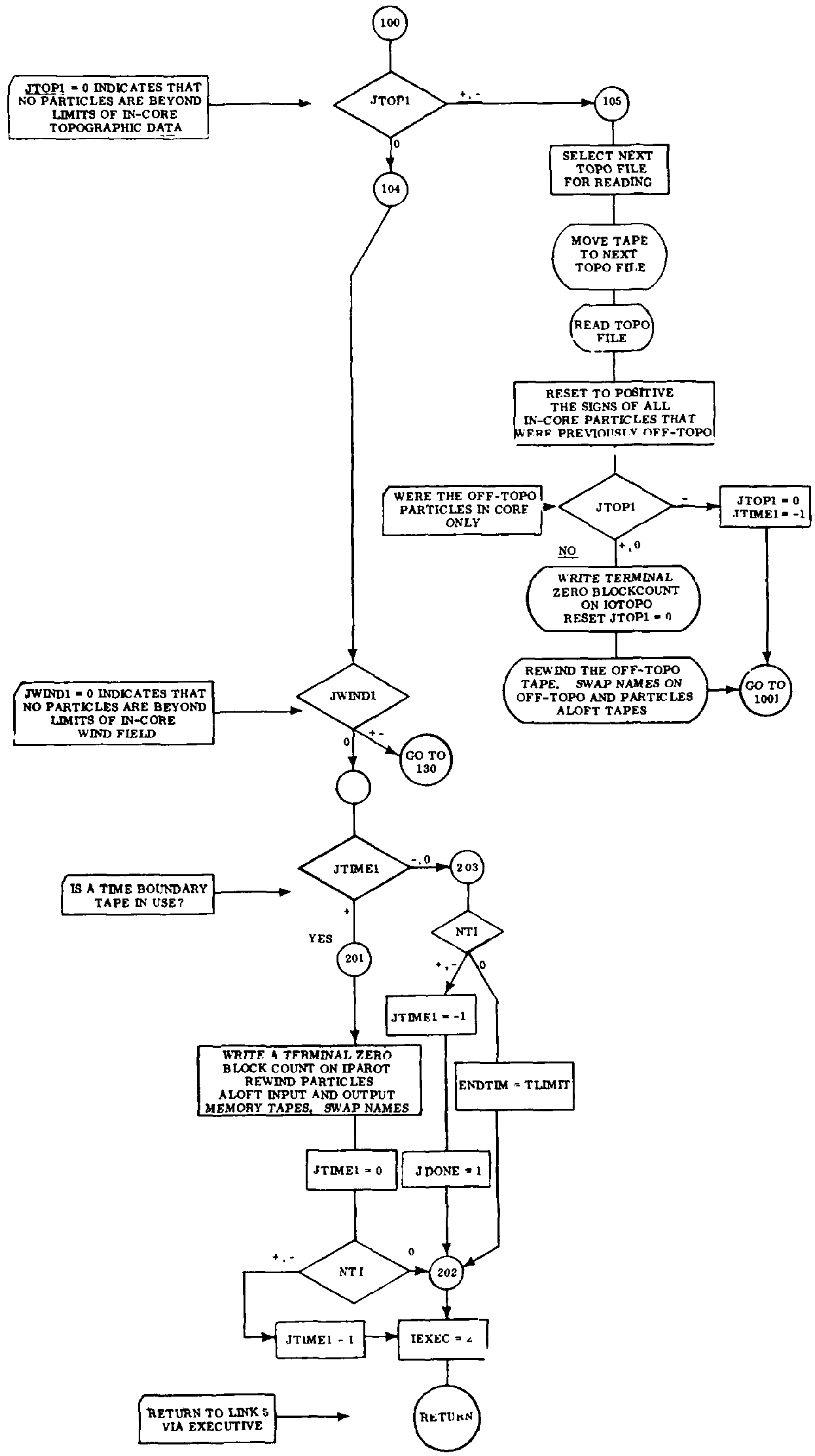
(a)

FC-10. Detailed Flow Charts for Subroutine LINK7



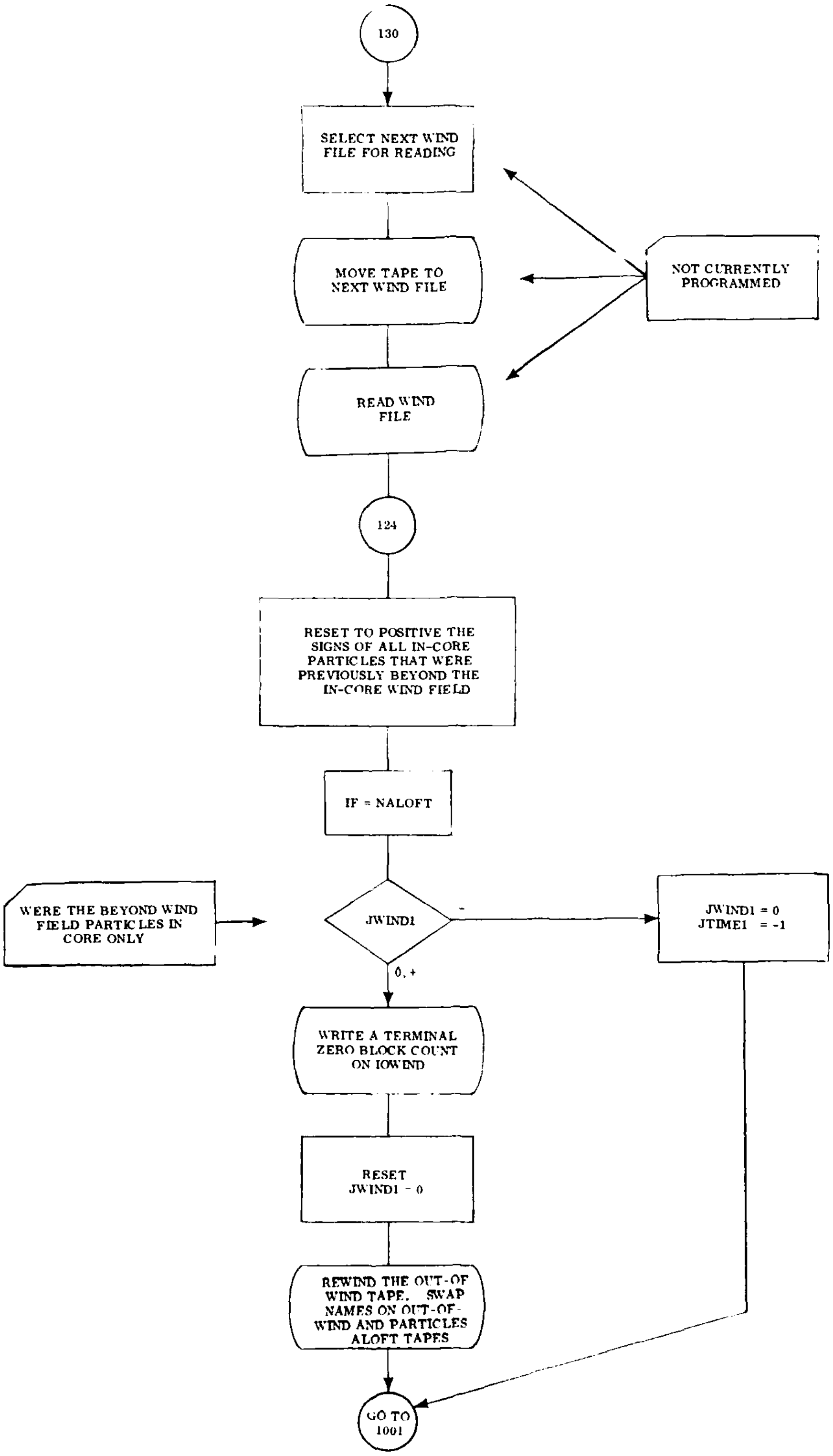
(b)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



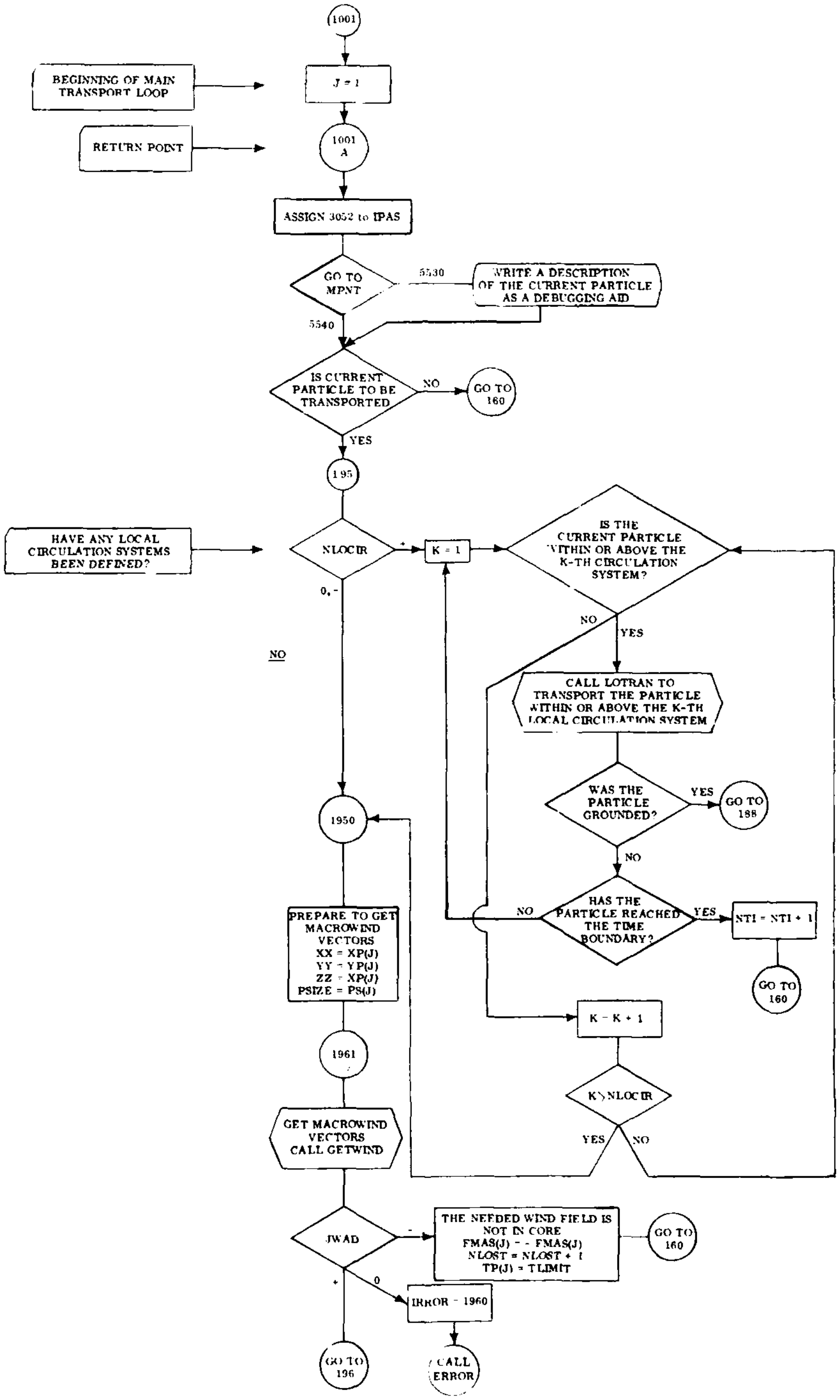
(c)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



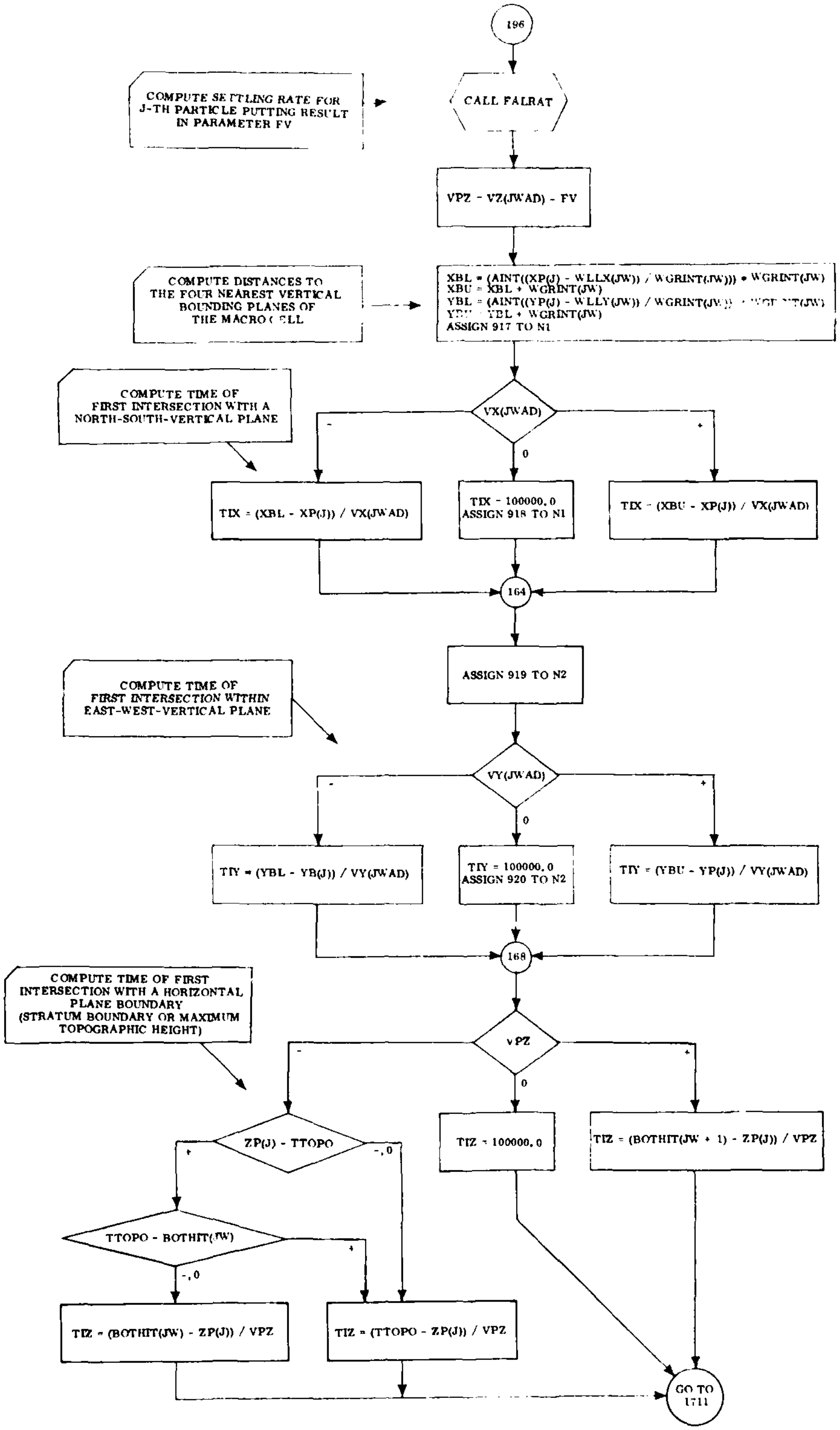
(d)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



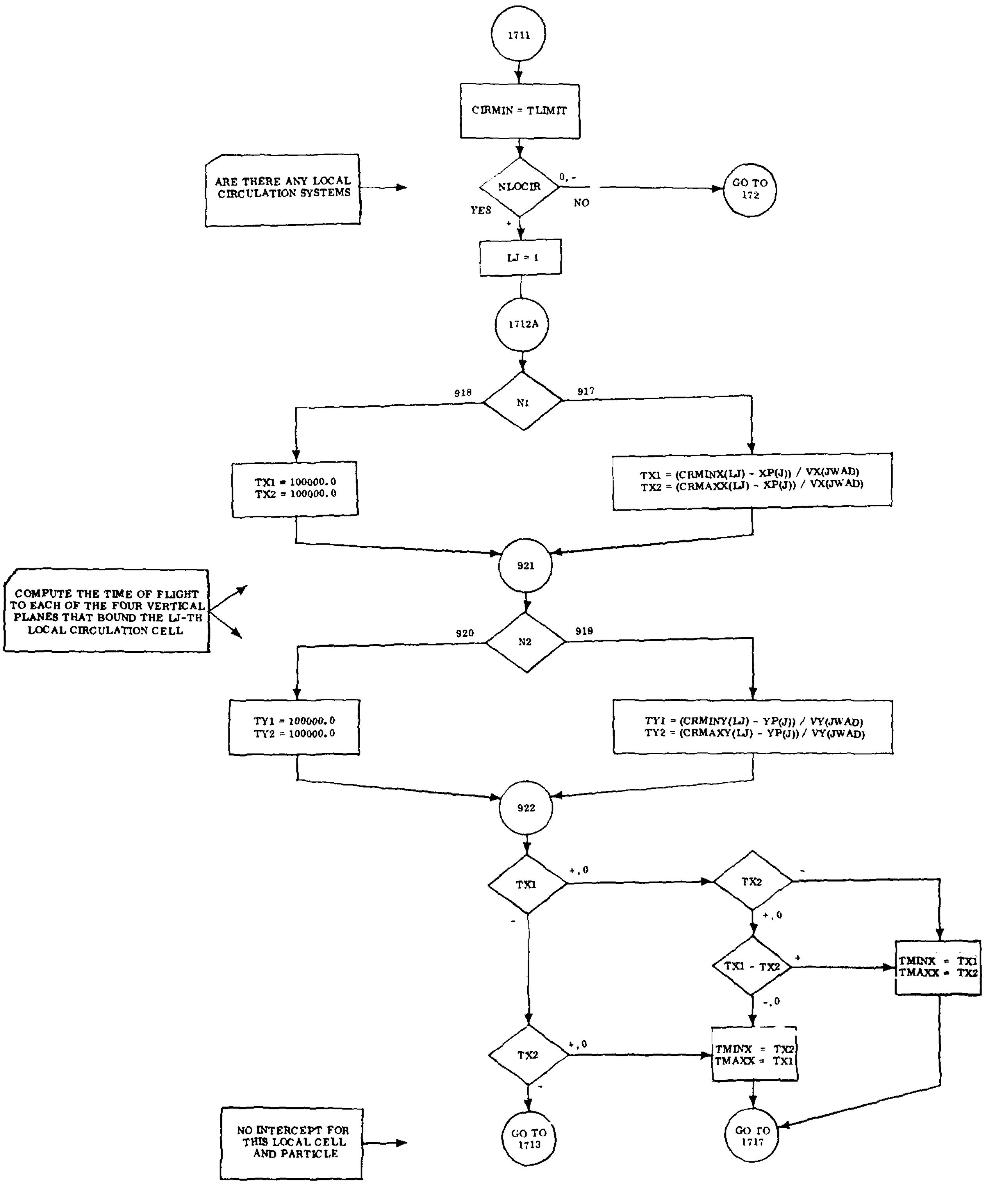
(e)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



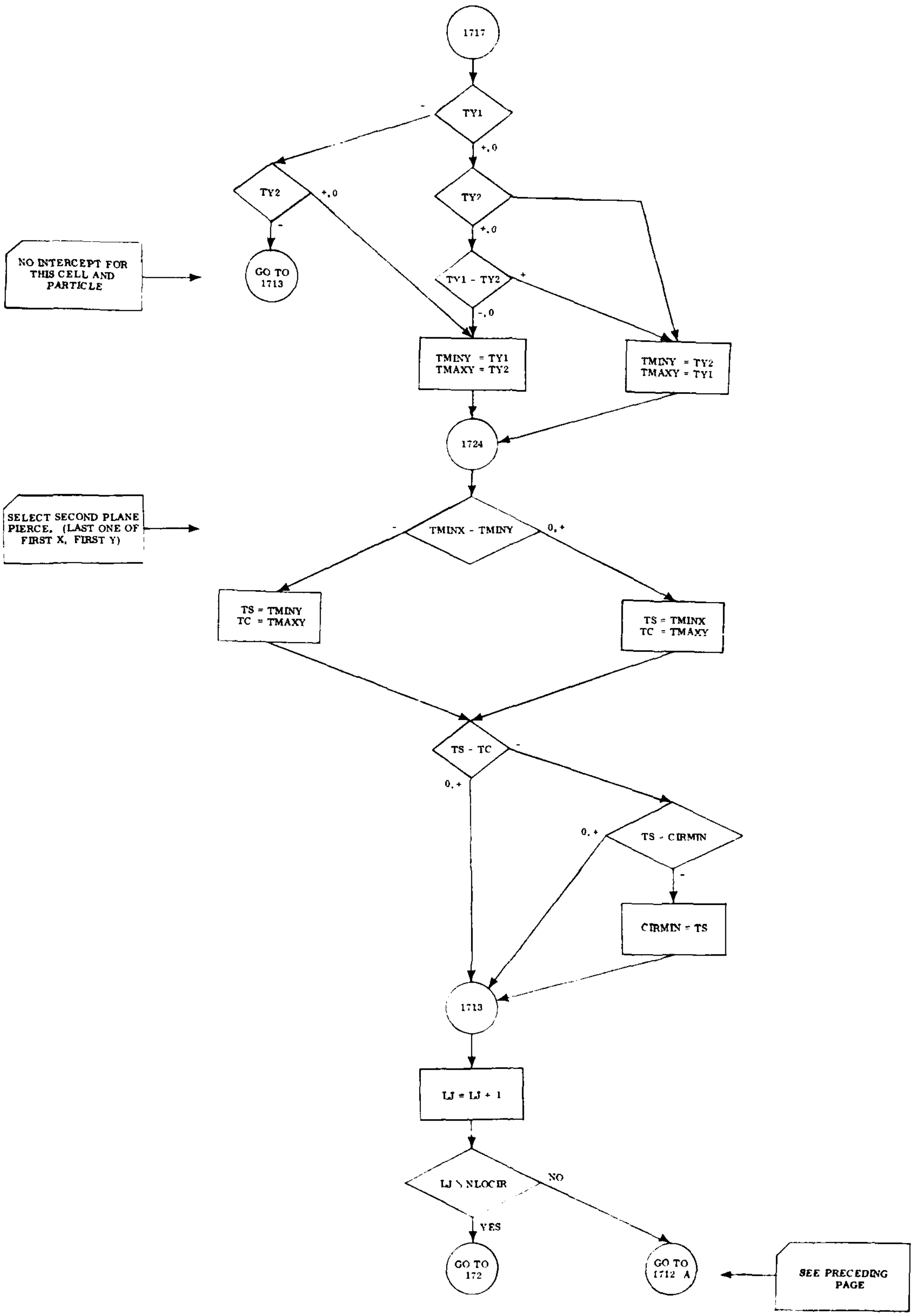
(f)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



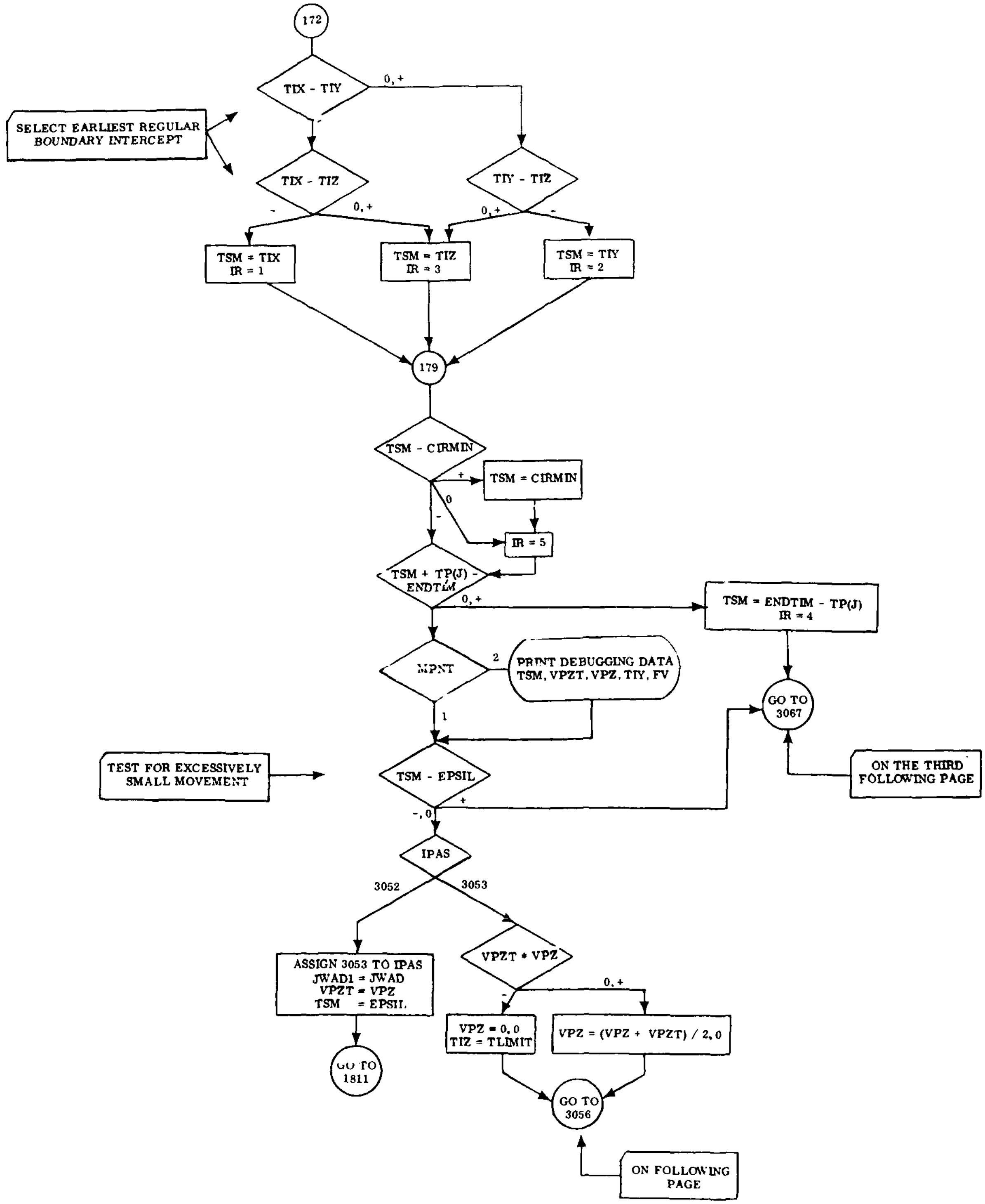
(g)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



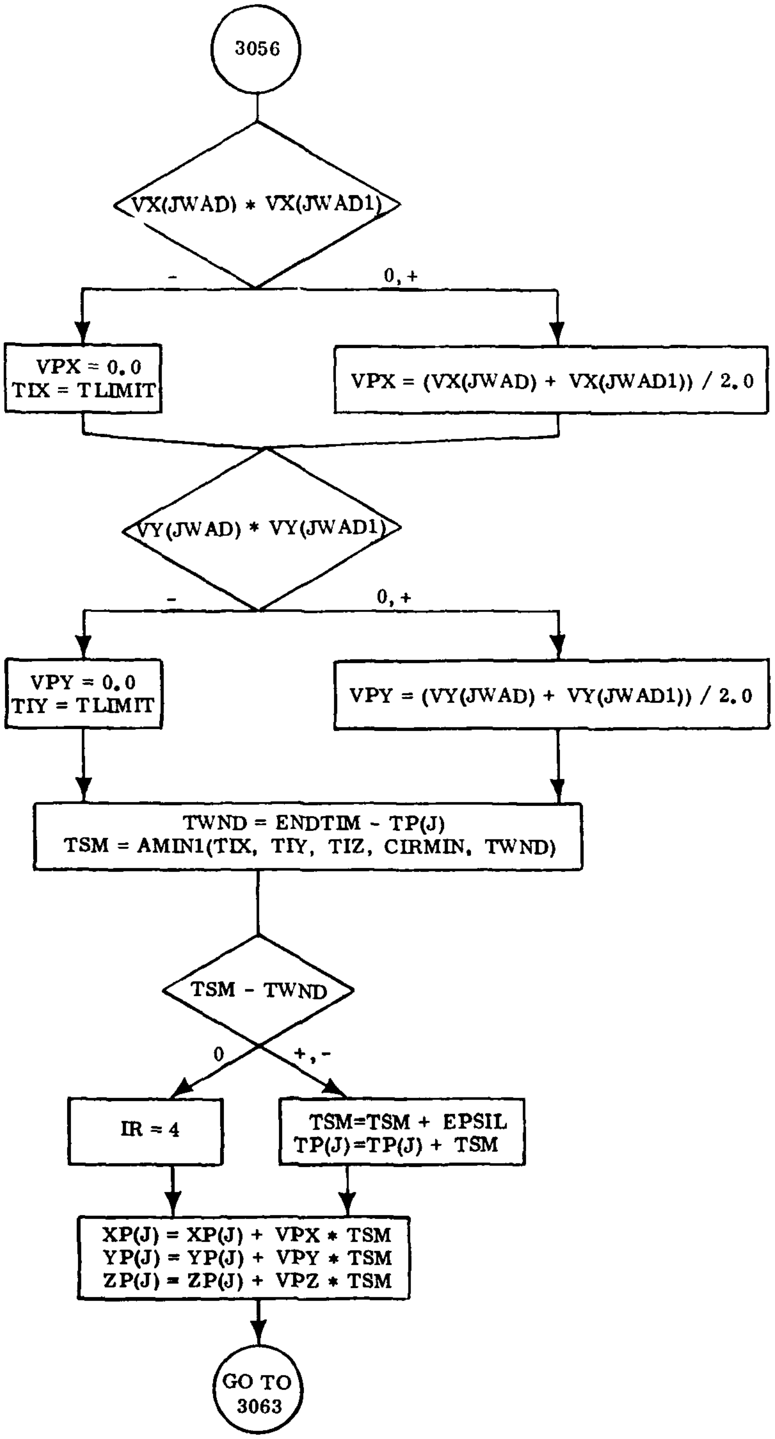
(h)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



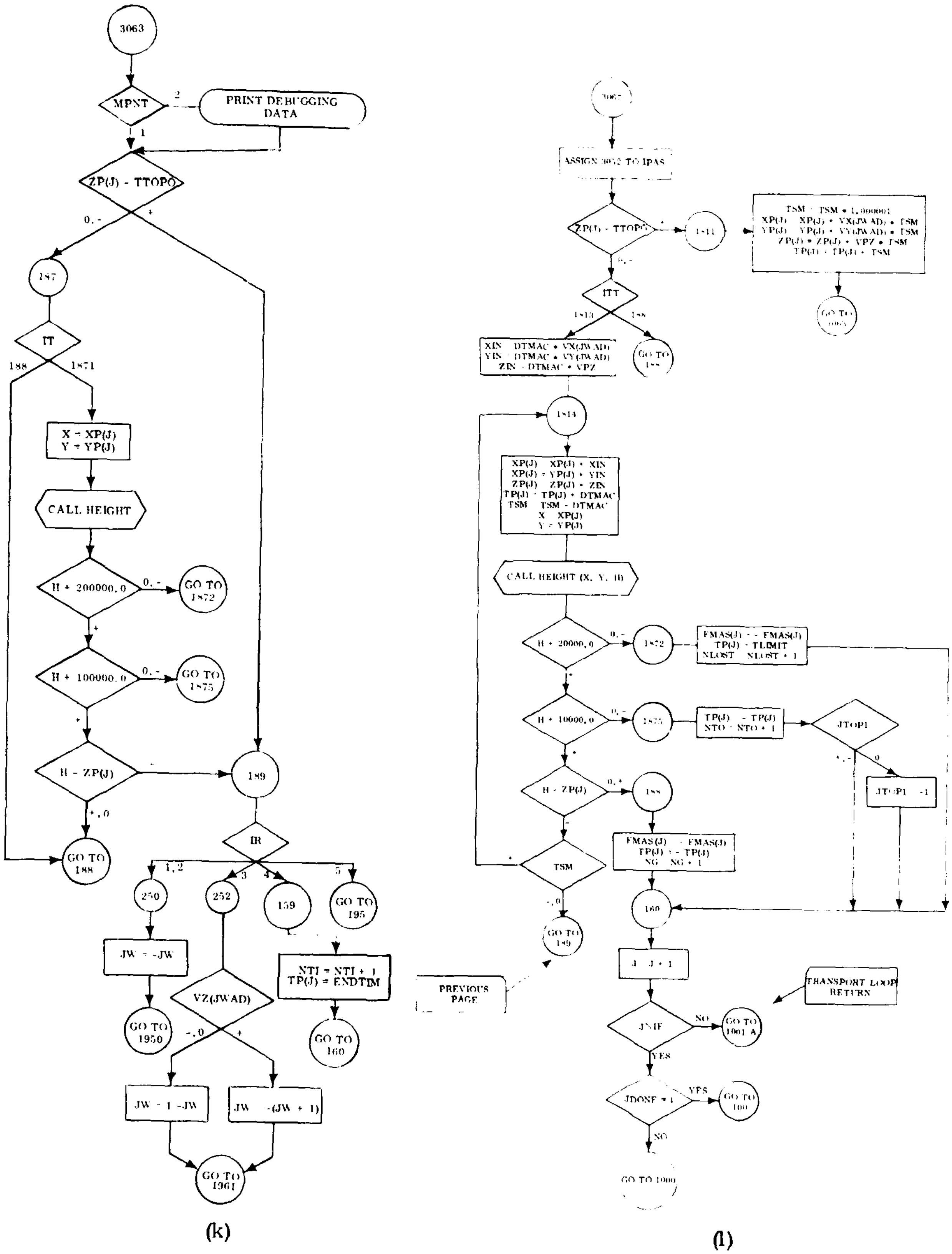
(i)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7



(j)

FC-10. (Continued) Detailed Flow Charts of Subroutine LINK7



FC-10. (Continued) Detailed Flow Charts of Subroutine LINK7

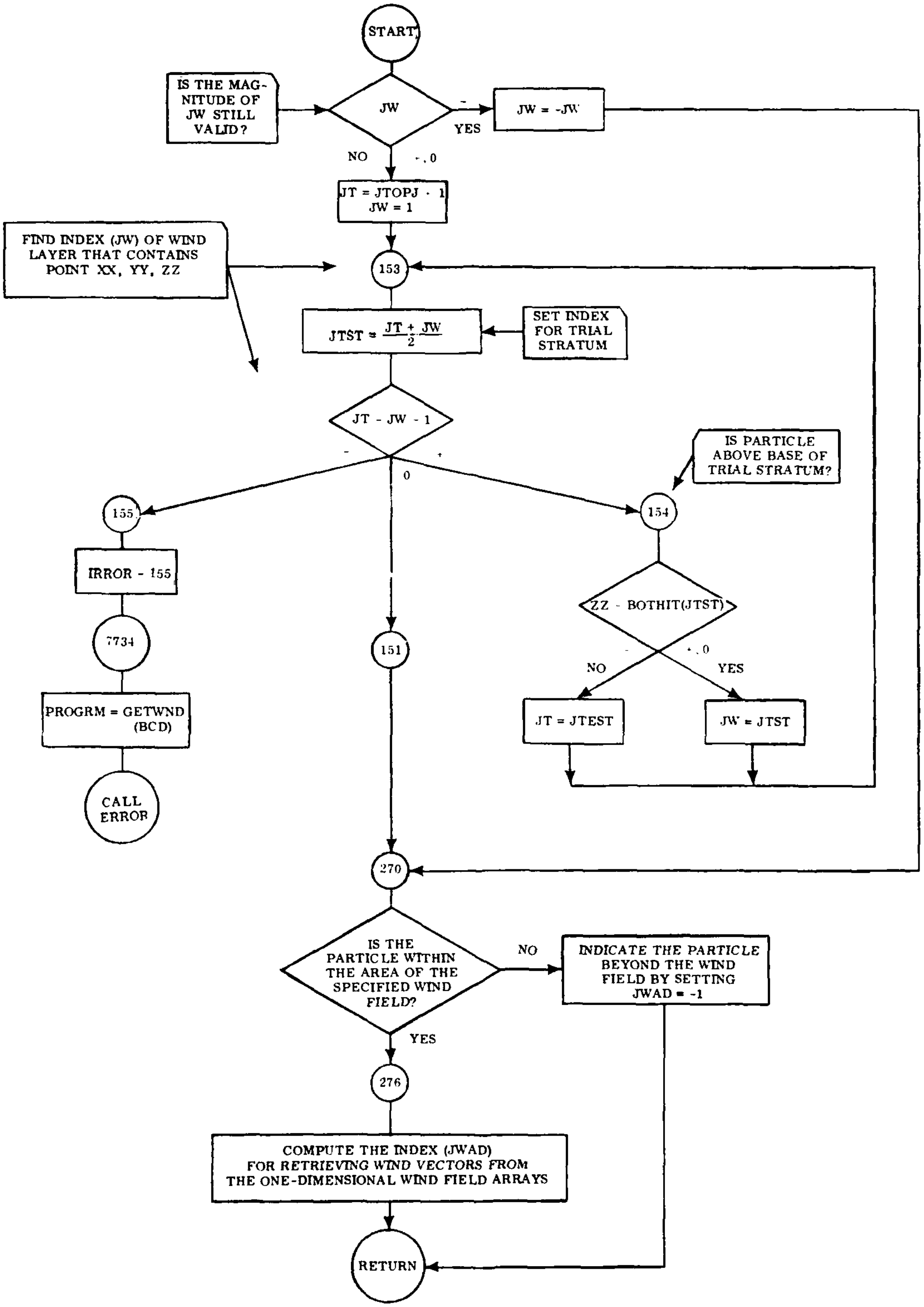
Next, at statement number 270 a check is made to see that the particle is within the bounds of the macrowind field. If it is not, JWAD is set -1 to indicate the problem to the calling program and GETWND returns. If the particle is in a satisfactory position, JWAD is computed to locate the desired vector and then GETWND returns control to the calling program.

Subroutine LOTRAN(J, K) (FC-12)

The purpose of this subroutine is to transport a particle when it is either within or above a local circulation system cell. This program is called from only one place in the main transport program (LINK7). The call is made from within the main transport loop but only when it is known that the particle being transported is either within or above the Kth local circulation cell. In the actual execution of LOTRAN, first an assignment is made on the basis of the type (CIRTYP(K)) of circulation program that is applicable within the Kth local cell. The purpose of this assignment is to allow efficient branching to the desired program within the actual local transport loop. After making the assignment the program branches to statement number 120.

At statement 120 the particle settling rate for the current particle is computed and stored in variable FV. Then by comparing the particle Z coordinate (ZP(J)) and the height of the top of the Kth local cell we determine whether the particle is above or within the local cell. If the particle is above the cell, we wish to transport the particle making use of the macrowind-field specification. Thus, we call subroutine GETWND to retrieve the macrowind vector for the particle position. Then the vertical particle velocity is computed as the sum of the settling rate, FV, and the vertical wind component. In order to be able to move the particle as far as possible in the next step, we must next compute the time of flight to all applicable boundaries and select the first intercept. These boundaries are an X-boundary plane, a Y-boundary plane, a plane forming a horizontal boundary between layers of the macrowind-field description, and the plane forming the top of the local cell.

Having selected the earliest intercept time, the particle is simply transported for that increment in one step making use of the macrowind vectors. At this point the particle will either be going into the local cell through its top, going out of the



FC-11. Flow Chart for Subroutine GETWND

local transport volume, or resting at a macrowind layer boundary or the time boundary. In any case, a transfer is made to statement number 131.

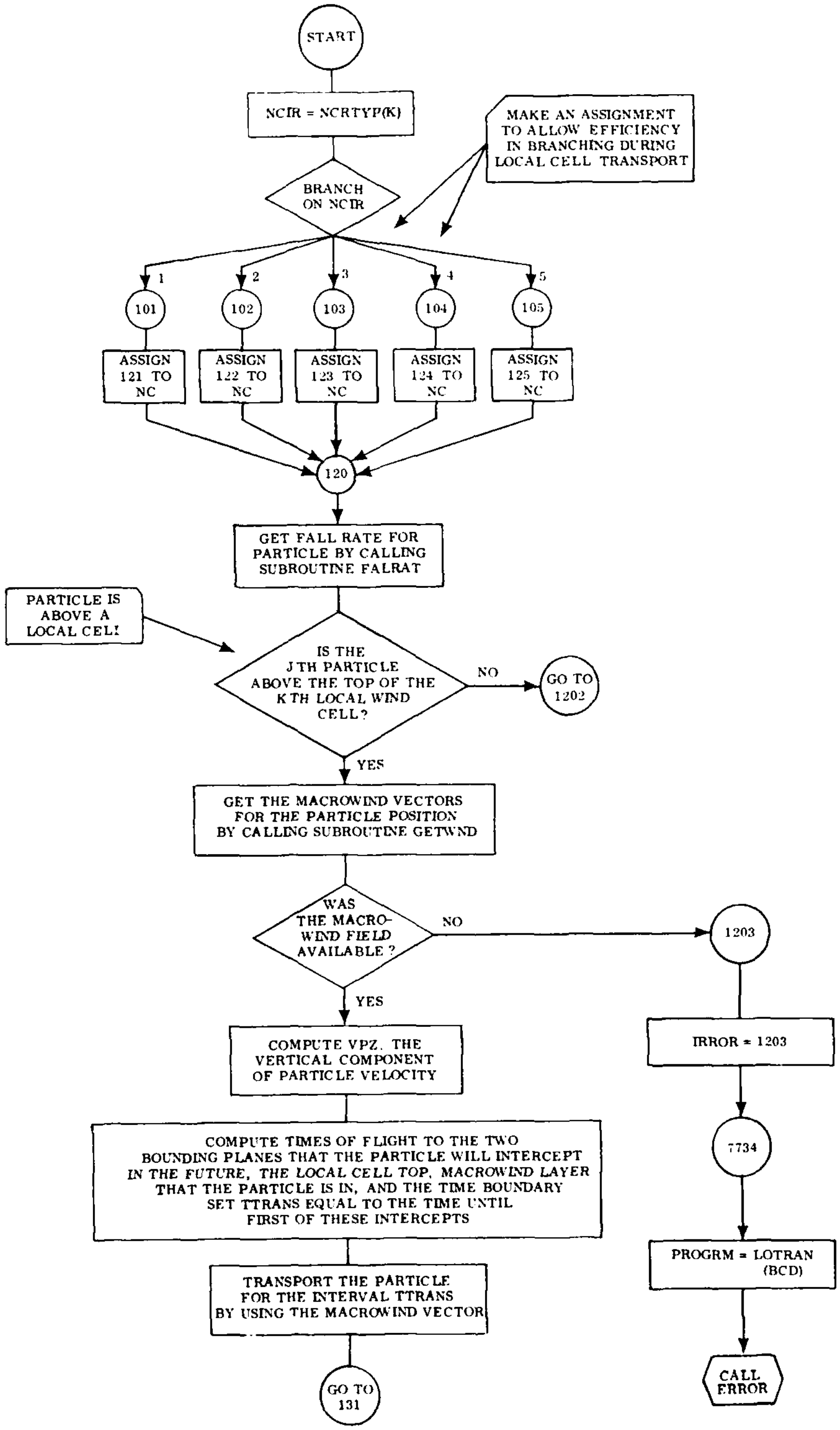
At 131 the program determines whether the particle is still in or above the local cell. If it is not, the program is finished and returns. If the particle is still in or above the local cell, the time boundary is checked and if not violated a return is made to the top of the loop at statement 120.

If the particle was originally or is now within the local cell, an ASSIGNED GO TO is used to transfer to a subroutine CALL statement which transfers control to a local circulation system subroutine (either MTWNDI, RGWNDI, or CBREZ1). Within the local circulation system subroutine the three wind velocity vector components are computed at the position of the particle and control then is returned to LOTRAN. The wind vector component is used to transport the particle over one (small) time step by point-slope integration (see p. 37). Particles within the local cell iterate through the transport loop procedure until they either leave the cell or become grounded.

Subroutine MTWND1 (J, K, AX, AY, AZ) (FC-13)

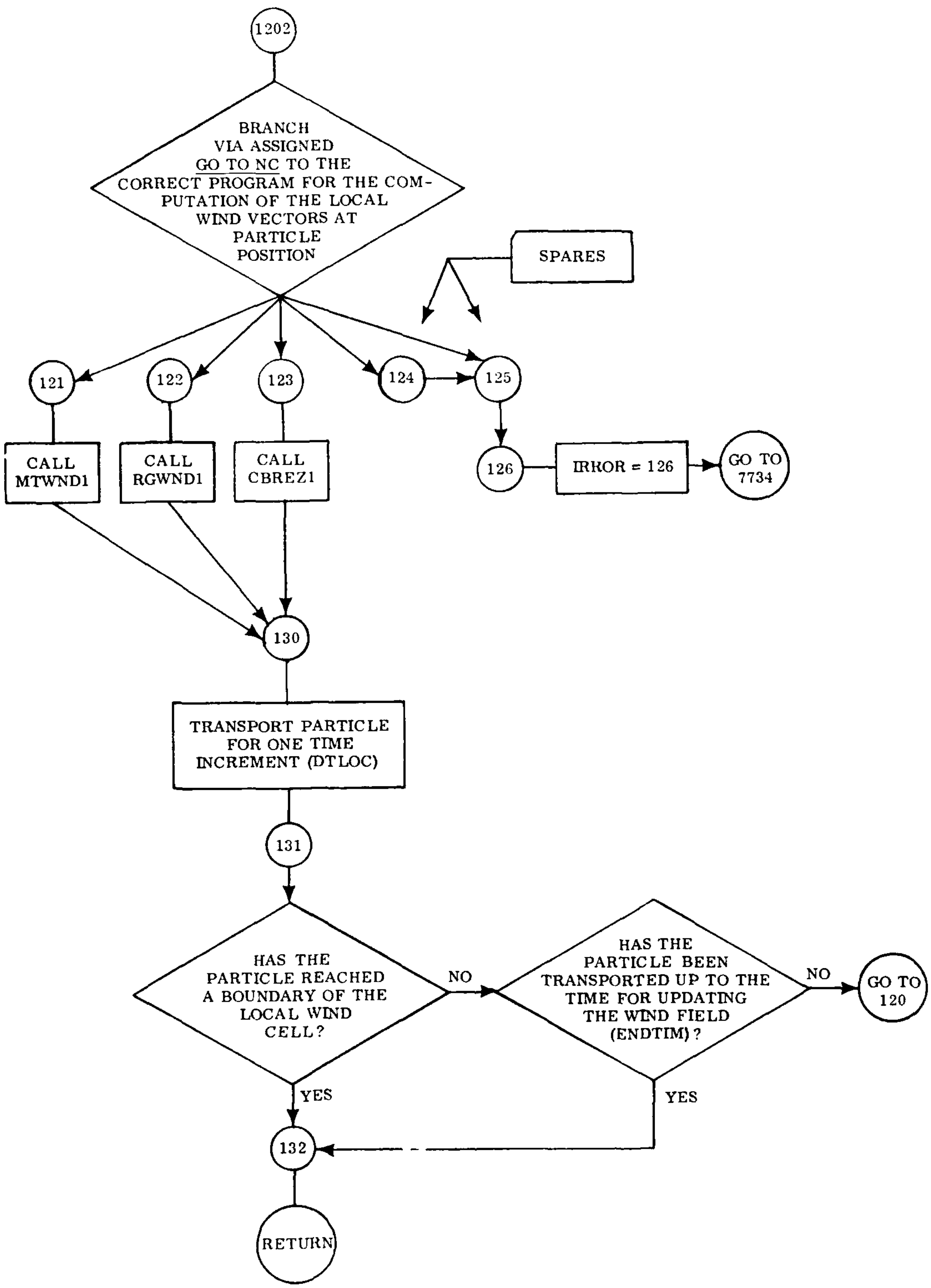
This subroutine, used for the Kth local cell, consists of two logical routes that serve the purpose of (1) reading mountain wind data and (2) computing the mountain wind components for the Jth particle at location XP(J), YP(J), ZP(J) after first checking for impact on the ground. If impact is sensed, the wind velocity is assigned a large downward velocity component, $AZ = -10^8$. The read route, entered when the sign of the argument J is negative, also serves to precompute constant geometrical relationships between the unperturbed wind, mountains, and macrosystem so as to facilitate computation on the compute route. The compute route is entered during actual particle transport when J, now positive, is the argument of the particular particle being moved.

In the read route, first the coordinates XM(I), YM(I), height H(I), and half-width A(I) of the Ith mountain are read for each of the I mountains with I ranging from 1 to NMT, the maximum number of mountains. Each mountain is checked for the ratio, H(I)/A(I), and location within the Kth local cell boundaries of north (CRMAXY(K)), south (CRMINY(K)), east (CRMAXX(K)), and west (CRMINX(K)).



(a)

FC-12. Flow Charts for Subroutine LOTRAN



(b)

FC-12. (Continued) Flow Charts for Subroutine LOTRAN

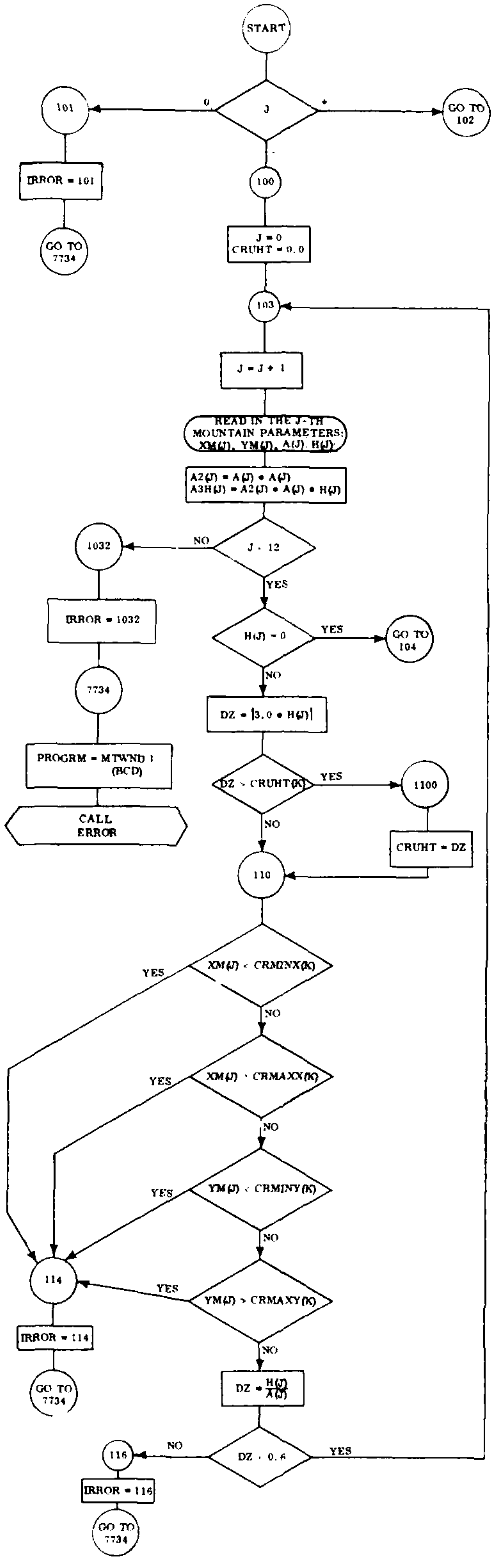
The height CRLHT(K), of the Kth cell is defined as three times the height of the tallest mountain. The error subroutine is called if a mountain ratio exceeds 0.6, if a mountain does not lie within the boundaries of the Kth local cell, or if the number of mountains exceeds the maximum of twelve mountains allowed in the Kth cell.

Next the read route computes the geometric center (XX, YY, ZZ) of the Kth local cell and calls the subroutine GETWND to retrieve the index JWAD of the unperturbed wind vector at that (XX, YY, ZZ) location within the macrowind field. A nonpositive JWAD index will call the ERROR subroutine. Using the stored wind components indexed on JWAD, the magnitude of the unperturbed wind vector, UO(K), and its direction in the macrowind field are computed. Constant parameters used in the compute route are also computed and stored. The local cell identification, the mountains, the unperturbed wind vector, and the boundaries of the local cell are printed out. This ends the read route.

The computing route first determines the distance of the Jth particle from the Ith mountain in terms of a component parallel and a component perpendicular to the direction of the unperturbed wind. The analytical height of the Ith mountain at this particle location is also determined. Then, the perturbed wind components parallel, horizontally perpendicular, and vertically perpendicular to the unperturbed wind vector are calculated. The height and perturbed wind components due to each mountain are summed, and DZ, the total analytic height of the mountain, is checked against ZP(J), the height of the particle, to determine impact. If DZ is greater than ZP(J), the velocity $(0, 0, -10^{8})$ is assigned to the wind. However, if the particle is still aloft, the unperturbed wind vector is added to the summed perturbed components, the resulting influence of all the mountains, and the wind-field vector is rotated back into the macrowind-field coordinate system. This ends the compute route with the desired wind components stored in variables AX, AY, and AZ.

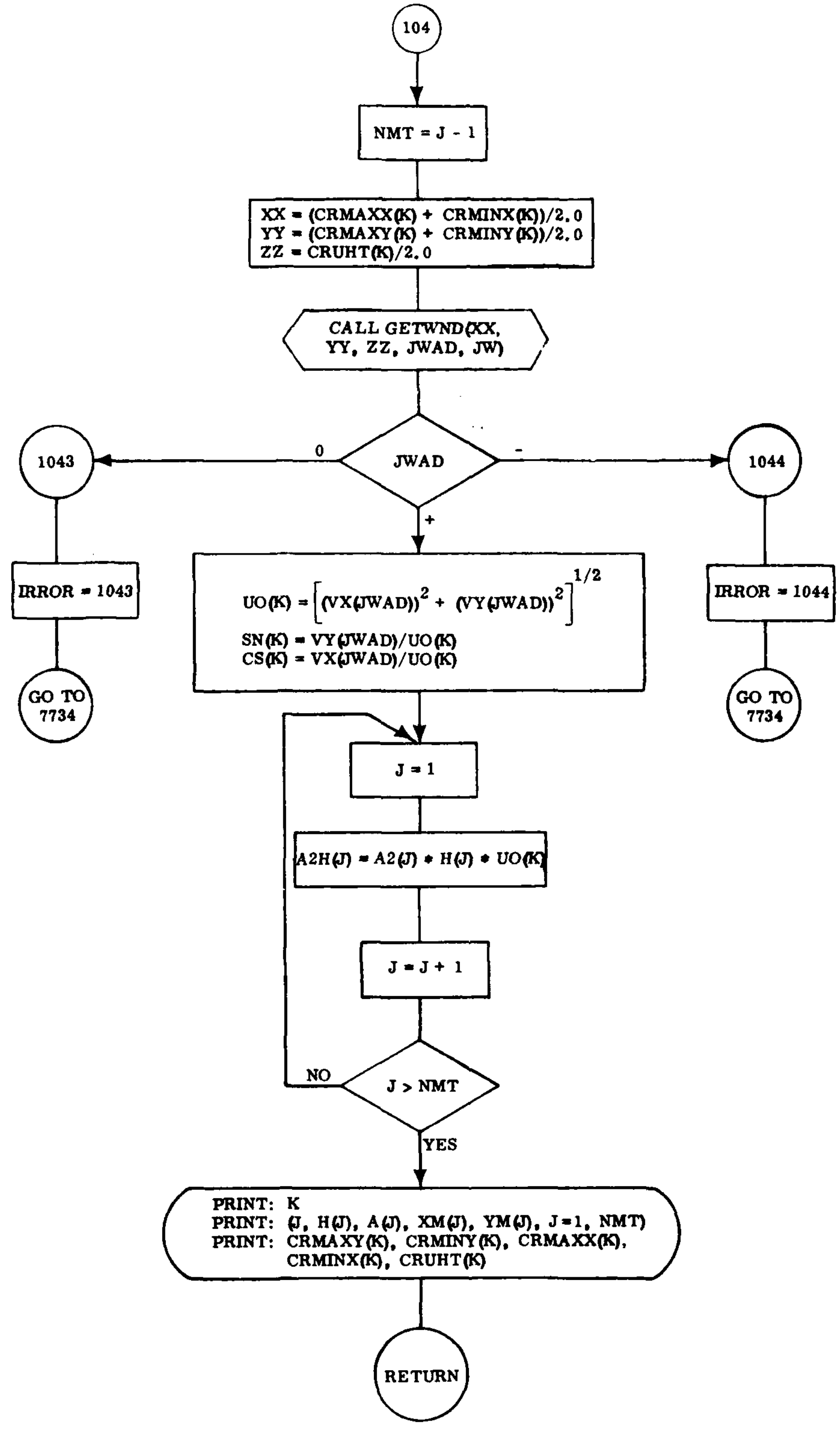
Subroutine RGWND1(J, K, AX, AY, AZ) (FC-14)

This subroutine for the Kth local cell consists of two logical routes that serve the purpose of reading ridge wind data and computing the ridge wind components for the Jth particle at location XP(J), YP(J), ZP(J) after first checking for impact on



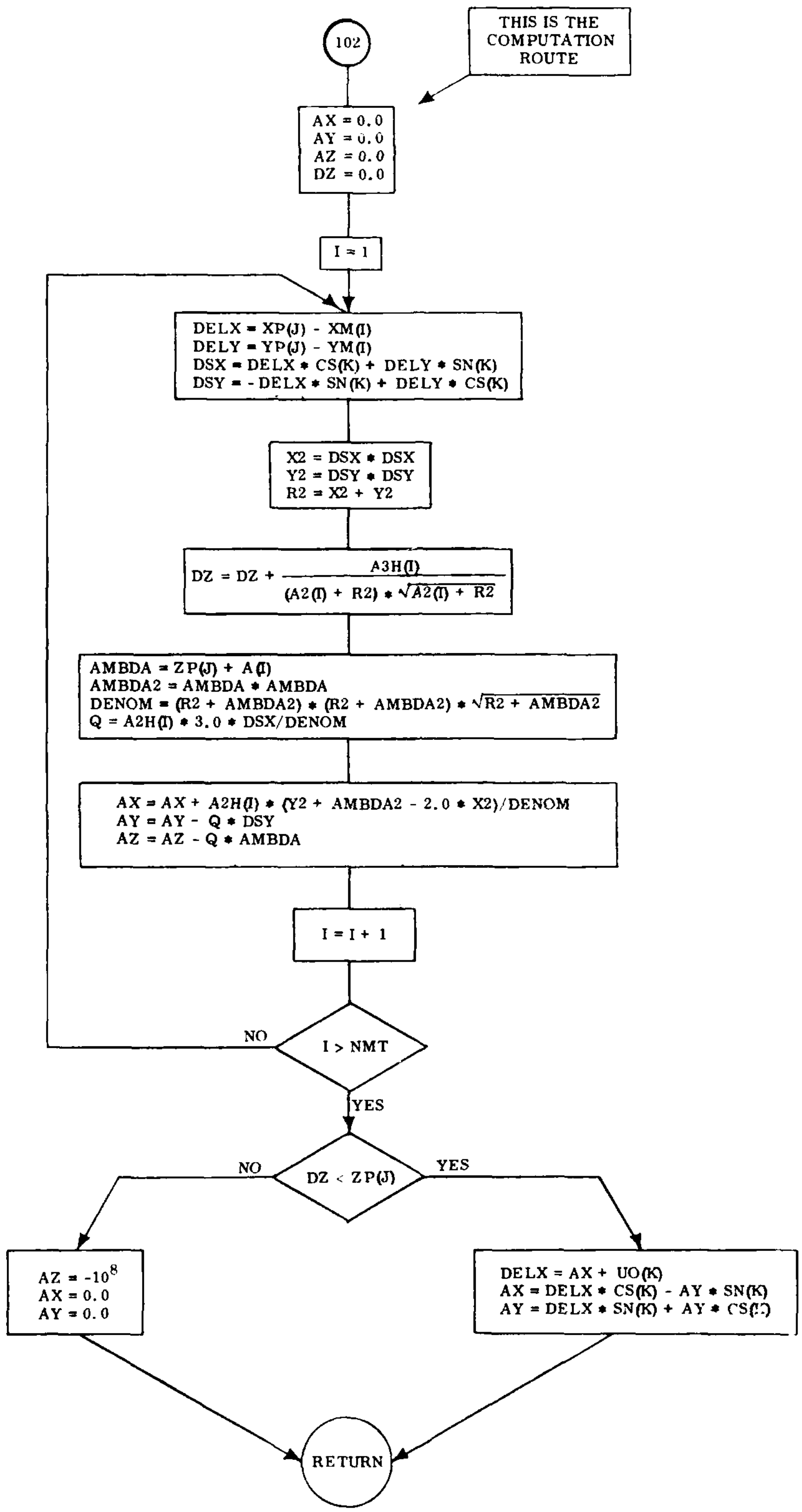
(a)

FC-13. Flow Charts for Subroutine MTWND1



(b)

FC-13. (Continued) Flow Charts for Subroutine MTWND1



(c)

FC-13. (Continued) Flow Charts for Subroutine MTWND1 the ground. If impact is sensed, the wind velocity is assigned a large downward velocity component. The read route, entered when the sign of argument J is minus, serves to pre-compute constant geometrical relationships between the ridges, unperturbed wind, and macrosystem to facilitate computation during the compute route. The compute route is entered during actual particle transport when J, now positive, is the argument of the particular particle being moved.

In the read route, first the coordinates XM(I), YM(I), height H(I), halfwidth A(I), and orientation B(I) of the Ith ridge are read for I ranging from 1 to NRG, the maximum number of ridges. The orientation, B(I), of a ridge is defined as the clockwise rotation of the ridge in radians, where zero radians indicates a ridge oriented north-south. Each ridge is checked for the ridge ratio, H(I)/A(I), and for location within the Kth local cell boundaries of north (CRMAXY(K)), south (CRMINY(K)), east (CRMAXX(K)) and west (CRMINX(K)). The height, CRUHT(K), of the Kth cell is defined as three times the height of the tallest ridge. The error subroutine is called if: a ridge ratio exceeds 0.6, a ridge does not lie within the boundaries of the Kth cell, or the number of ridges exceeds the maximum of twelve ridges allowed in the Kth cell.

Next, the read route computes the geometric center (XX, YY, ZZ) of the Kth local cell and calls the subroutine GETWND to retrieve JWAD, the index of the unperturbed wind vector at that (XX, YY, ZZ) location within the macrowind field. A nonpositive JWAD index will lead to the ERROR subroutine. Using the stored wind components indexed on JWAD, the magnitude of the unperturbed wind vector, UO(K), and its direction in the macrowind field are computed. Constant geometrical relationships between wind vector components (UO(K)), the macrowind field, and the orientation and location of the ridges are computed and stored for use in the compute route. The local cell identification, the ridges, the retrieved unperturbed wind vector, and the boundaries of the local cell are printed out. This ends the read route.

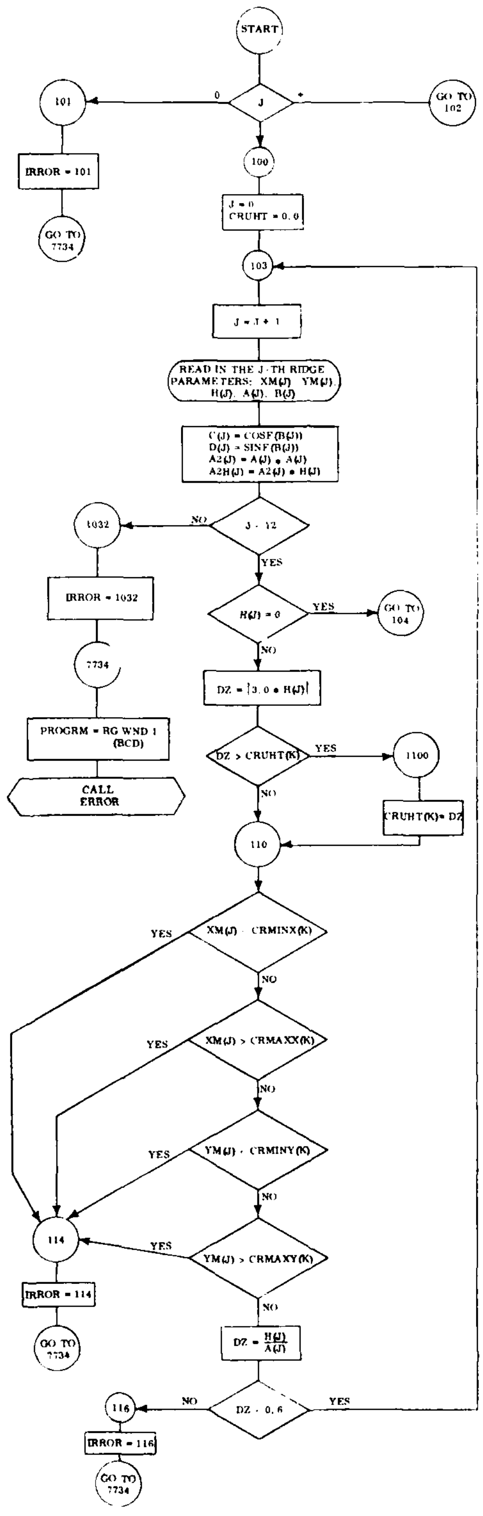
The computing route first determines the perpendicular distance of the Jth particle from the Ith ridge. The analytical height of the ground and the parallel, horizontally perpendicular, and vertically perpendicular perturbed wind components with respect to the unperturbed wind, UO(K), are now computed at this particle position. The results for each ridge are summed with the succeeding ridges to yield total height and perturbed wind vectors due to all the ridges at this point. Next, the particle height, ZP(J), is checked against the total analytical ground height, DZ, to determine impact; upon which, the velocity $(0, 0, -10^8)$ msec^{-1} is assigned to the wind. However, if the particle is still aloft, the unperturbed wind vector is added to the summed perturbed components and the result is rotated back into the macrowind field coordinate system. This ends the compute route with the desired wind components stored in variables AX, AY, and AZ.

Subroutine CBREZ1(J, K, AX, AY, AZ) (FC-15)

Before attempting to use the sea-breeze local circulation system, the reader is advised to obtain a thorough understanding of the model by studying the presentations in the Physical and Mathematical Models section and in Appendix B.

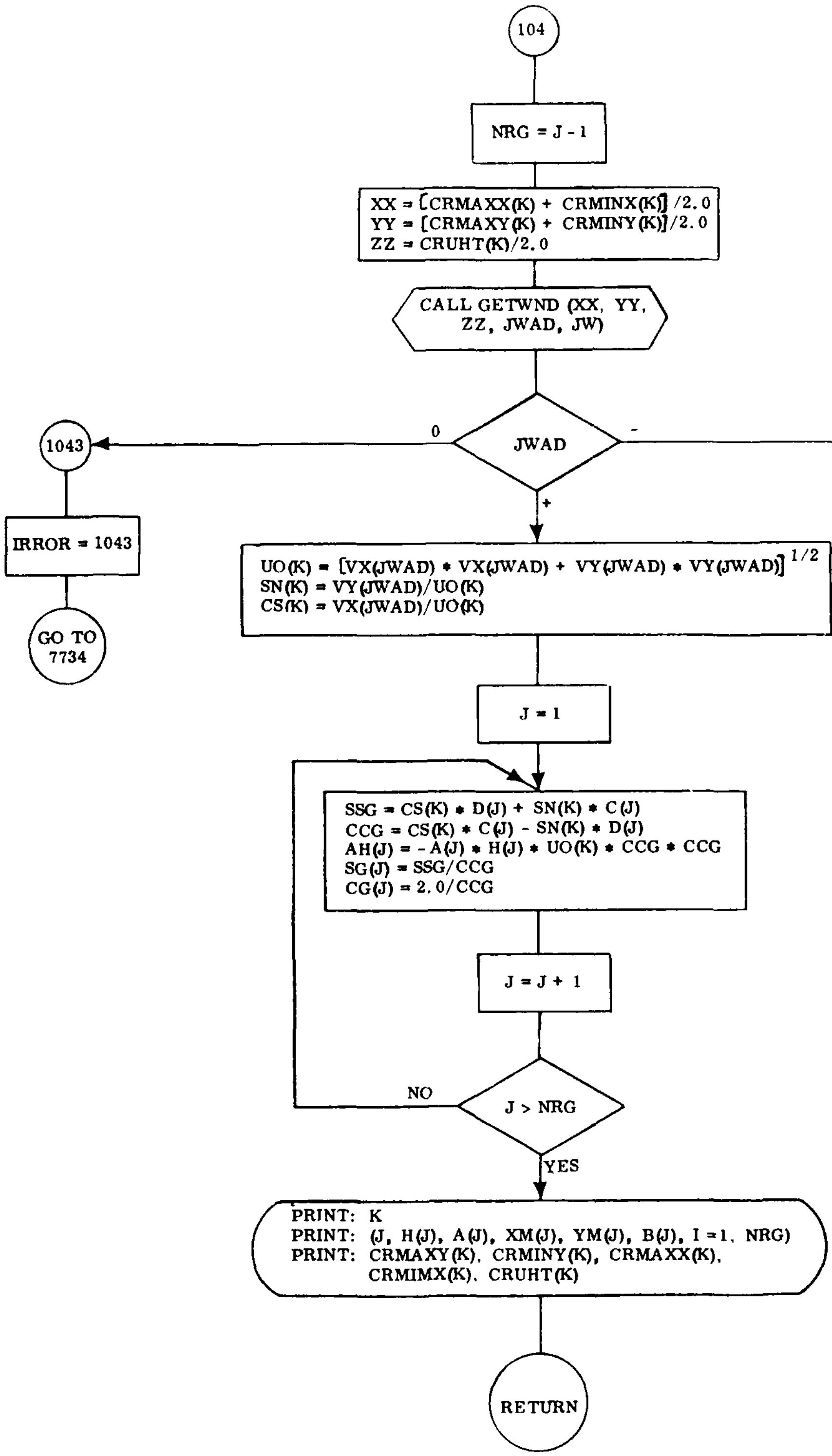
The CBREZ1 subroutine serves the dual purpose of reading the sea-breeze data and computing the sea-breeze velocity components for the Jth particle at location XP(J), YP(J), ZP(J) and time TP(J), after first checking for particle impact at sea level. The programming of CBREZ1 is divided into two mutually exclusive chains of logic that will be referred to as the read route and the compute route. The read route, entered at statement number 100 when the sign of the argument J is negative, also serves to pre-compute constant sea breeze parameters to facilitate computation during the compute route. The compute route is entered during actual particle transport when J, now positive, is the index of the particular particle being moved.

The read route starts by reading from the system input tape values for parameters B, GRAD, NN and the pairs DELTX(N) and TAUX(N) for N ranging from 1 to NN. The maximum value of NN is nine (see Table 3). N represents the order of the harmonic described by DELTX(N) and TAUX(N). These parameters serve to calculate OMGX(N), AJZX(N), AJXX(N), AJY(N), R1RX(N), R1IX(N), R2RX(N), R2IX(N), ESQ1, ESQ2, AN1, AN2, FAX(N), T1, DELTX(N), and TAUX(N), for each Nth harmonic, and are printed out as the respective symbols: OMGN(N), AJZ(N), AJX(N), AJY(N), AKN1(N), ALN1(N), AKN2(N), ALN2(N), BLOW1(N), BLOW2(N),



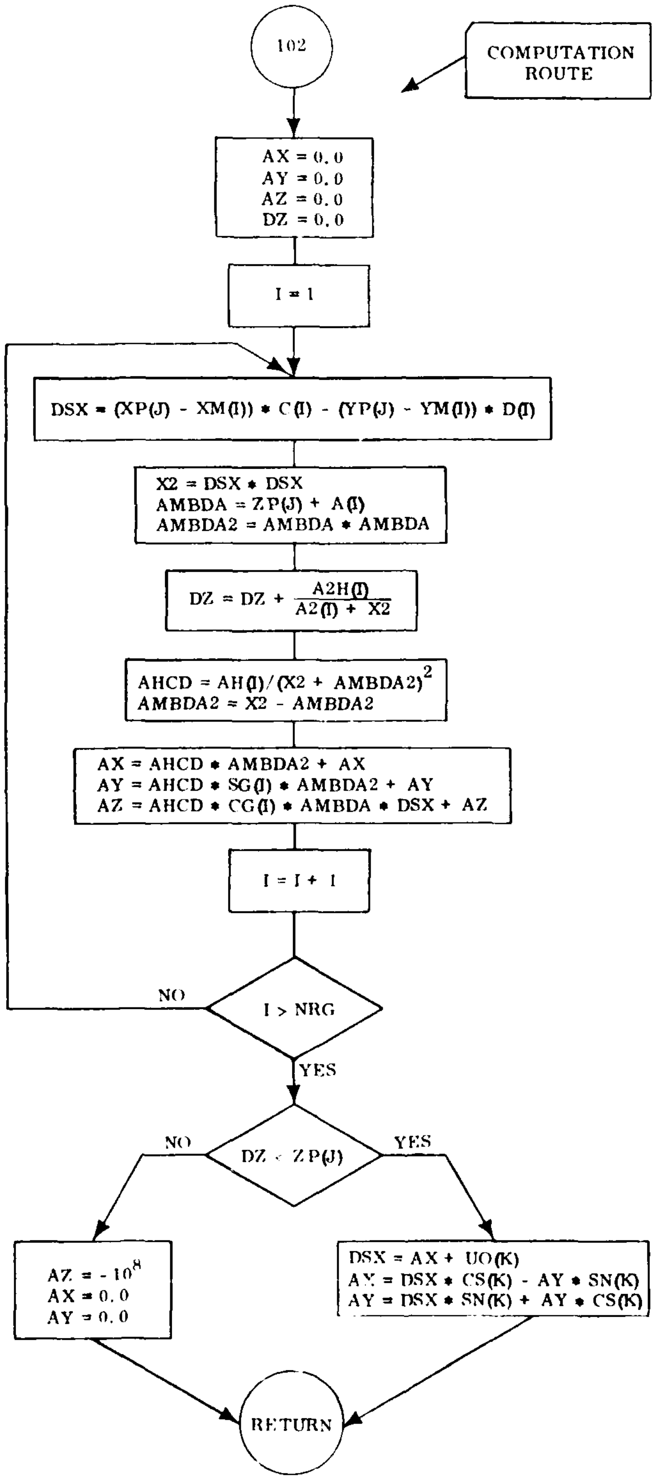
(a)

FC-14. Flow Charts for Subroutine RGWND1



(b)

FC-14. (Continued) Flow Charts for Subroutine RGWND1



(c)

FC-14. (Contined) Flow Charts for Subroutine RGWND1

ANN1(N), ANN2(N), PHIN(N), ENU(N), DELTX(N), and TAUX(N). Correspondences of these mnemonics with the symbols defined in the Physical and Mathematical Models section are given in Tables 3 and 4.

During the computation of these constants checks are made to ensure that SGMA, the Guldberg-Mohn friction parameter, and AKY, the thermal eddy diffusivity, are not zero. If either one of these constants is zero, subroutine ERROR is called.

Also calculated in the read route are the height, CRUHT(K), of the sea-breeze cell and geometrical constants to rotate the particle coordinates and wind vectors in and out of the sea-breeze system. The angle of rotation, B, is zero when the sea lies on the west side of a shore line parallel to the north-south axis. The shore line is defined to run through the center, (XCB, YCB), of the area bounded by the north (CRMAXY(K)), south (CRMINY(K)), east (CRMAXX(K)), and west (CRMINX(K)) vertical sides of the sea-breeze cell. The printout from the read route sequentially consists of the local cell identification, the cell boundaries, the input parameters, and the harmonic parameters.

The compute route first checks particle altitude, ZP(J), against sea level. It assigns the wind vector $(0, 0, -10^8)$ for negative altitudes and returns to the calling program. The coordinates of the particles nonnegative altitude are rotated into the sea-breeze coordinate system. If the horizontal distance between the particle and the shore line (measured perpendicular to the shore line) is greater than the half width of the sea-breeze cell, an exponential attenuation based on the perpendicular distance from the edge of this primary cell is used. No attenuation is used within the primary cell.

The wind-field constants for each harmonic mode are now calculated. The vertical (AZ) and the horizontally parallel (AY) and perpendicular (AZ)(with respect to the shore line) wind vectors are computed from Eqs. (60), (61), and (62) and summed over all the harmonic modes. The resultant wind vectors are next rotated back into macrowind-field coordinates and subroutine CBREZ1 returns control to the calling program.

To aid the user in evaluating the properties of the sea-breeze circulation system generated from the input data, an "interpretative output" of key model parameters is provided. This output is described in Table 4. These parameters are

TABLE 3

INPUT QUANTITIES FOR THE SEA BREEZE

Physical Quantity	Text Designation	Program Designation	Dimension Units	Typical Values and Comments
Number of harmonics	n	NN		Approximately two or three
Total extent of sea breeze	L_x	ELX	m	Less than 10^5m
Sine of latitude	$\sin \phi$	SNPHI		$-1 \leq \sin \phi \leq +1$
Angle of coastline relative to y axis of grid	ψ	B		
Wind field extrapolation attenuation constant	k_a	WW	m^{-1}	$0 \leq k_a \leq \infty$
Guldberg-Mohn friction parameter	σ	SGMA	sec^{-1}	A value of zero is not allowed; typical values $0.5 \times 10^{-4} \leq \sigma \leq 2.5 \times 10^{-4}$
Average ground temperature	θ_o	THET	^oK	Expressed in degrees Kelvin: $\theta_o = 300^oK$
Unperturbed temperature gradient	$\Gamma = (d\theta_o/dz)$	GRAD	$^oKm^{-1}$	A constant z-independent positive value must be used; typical values $2 \times 10^{-3} \leq \Gamma \leq 7.5 \times 10^{-3}$
Thermal eddy diffusivity	K	AKY	m^2sec^{-1}	A constant z-independent value must be used; typical values $25 \leq K \leq 75$
Magnitude of nth temperature differential	T_n^*	DELTX(N)	^oK	First harmonic will generally be less than 10^oK, with subsequent harmonics decreasing in magnitude
Phase of nth temperature differential	τ_n	TAUX(N)		Phase of first harmonic should correspond to about 1 hr, or $\tau_1 \sim \Omega\,(3.6) \times 10^3 = 0.26$
Lag time between sea-breeze local time and Greenwich time	Δt_s	ELAG	sec	

TABLE 4

INTERPRETATIVE OUTPUT DESCRIPTION

Output Designation	Text Designation	Interpretative Output Description Designation
OMGN(N)	$n\Omega$	$n\Omega$
AJZ(N)	J_{nz}	$-T_n^* L/B$
AJX(N)	J_{nx}	$\lambda^{-1} J_{nz}$
AJY(N)	J_{ny}	$G J_{nx}$
AKN1(N)	k_{n1}	k_1
ALN1(N)	ℓ_{n1}	ℓ_1
AKN2(N)	k_{n2}	k_2
ALN2(N)	ℓ_{n2}	ℓ_2
BLOW1(N)	$\overline{K}_{n1}$	$\epsilon_1 U_1$
BLOW2(N)	$\overline{K}_{n2}$	$\epsilon_2 U_2$
ANN1(N)	η_{n1}	η_1
ANN2(N)	η_{n2}	η_2
PHIN(N)	ϕ_n	$h - m + \tau_n$
ENU(N)	ν_n	$-\theta_1$

are sufficient to calculate the wind velocity components w_n, u_n, and v_n as given by Eqs. (60), (61), and (62). Column 1 of Table 4 gives the parameter designation in the computer output and column 2 gives the parameter designations in the text. In order that the user may be able to understand in detail the computations required for evaluation of these parameters, a path through the calculation is presented in the paragraphs to follow. Column 3 of Table 4 gives the expressions, in terms of the fundamental quantities used in the following calculation description, used to calculate the parameters in column 2.

After reading the input data, the machine computes the constants:

$$f = 2\Omega \sin \phi \quad ,$$

$$\lambda = (2\pi/L_x) \quad ,$$

$$\alpha = g/\theta_o \quad .$$

At this point the selection of the harmonic mode takes place. Since all the physical quantities with the exception of the input parameters T_n^* and τ_n depend on the mode only through their dependence on $n\Omega$, we now set

$$\Omega = n\Omega \quad .$$

When the foregoing substitution is made it becomes possible to drop the subscript n, it being understood that we are dealing with the nth mode. This permits many of the dummy analytical variables subsequently defined to bear a one-to-one correspondence with the mode-dependent variables. For instance, defined quantities such as q, a, b, and ϵ_1 correspond to q_n, a_n, b_n, ϵ_{1n}.

The next group of calculations is:

$$q = \sigma + i\Omega = A_1 e^{i\theta_1}, \; A_1 = \left(\sigma^2 + \Omega^2\right)^{1/2}, \; \theta_1 = \tan^{-1} (\Omega/\sigma) \; ;$$

$$q^2 = A_2 e^{i\theta_2}, \; A_2 = A_1^2 , \; \theta_2 = 2\theta_1 \; ;$$

$$q^2 + f^2 = A_3 e^{i\theta_3}, \; A_3 = \left(\left(\sigma^2 + f^2 - \Omega^2\right)^2 + 4\Omega^2\sigma^2\right)^{1/2}, \; \theta_3 = \tan^{-1}\left(\frac{2\Omega\sigma}{\sigma^2 + f^2 - \Omega^2}\right) \; .$$

We next turn our attention to the coefficients a, b, c and d:

$$a = q^2\lambda^2/\left(q^2+f^2\right) = A_4 e^{i\theta_4},\ A_4 = \left(A_2\lambda^2/A_3\right),\ \theta_4 = \theta_2 - \theta_3 \quad ;$$

$$b = q\alpha\lambda^2/\left(q^2+f^2\right) = Le^{ih},\ L = \left(A_1\alpha\lambda^2/A_3\right),\ h = \theta_1 - \theta_3 \quad ;$$

$$c = (\Gamma/K) = A_7 \ ;$$

$$d = i\,(\Omega/K) = iA_6,\ A_6 = (\Omega/K)\ .$$

At this point the roots of the dispersion relationship are calculated. First we have:

$$\mu_1 = \frac{a+d}{2} + \frac{R}{2} = C_1 + iD_1 = E_1 e^{i\gamma_1} \quad ,$$

$$\mu_2 = \frac{a+d}{2} - \frac{R}{2} = C_2 + iD_2 = E_2 e^{i\gamma_2} \quad ,$$

$$R = \left((a+d)^2 - 4\,(ad+bc)\right)^{1/2} = Be^{im} \quad ,$$

where

$$(a+d) = A_4\cos\theta_4 + i\left(A_4\sin\theta_4 + A_6\right)$$

$$= \xi_1 + i\,\xi_2 \quad ,$$

$$(a+d)^2 - 4(ad+bc) = B_1 + i\,B_2 = \left(B_1^2 + B_2\right)^{1/2} e^{i\beta} \ ,$$

and

$$B_1 = \xi_1^2 - \xi_2^2 - 4\left(LA_7\cos h - A_4A_6\sin\theta_4\right) \quad ,$$

$$B_2 = 2\,\xi_1\,\xi_2 - 4\left(A_4A_6\cos\theta_4 + LA_7\sin h\right) \quad ,$$

$$\beta = \tan^{-1}\left(B_2/B_1\right).$$

We define

$$B = \left(B_1^2 + B_2^2\right)^{1/4} \quad ,$$

$$m = \beta/2 \quad .$$

Using the foregoing expressions, C_1, D_1, C_2, D_2, E_1, E_2, γ_1, and γ_2 are then calculated.

$$C_1 = \frac{1}{2}\left[A_4 \cos\theta_4 + B\cos m\right] \; ; \; D_1 = \frac{1}{2}\left[A_4 \sin\theta_4 + A_6 + B\sin m\right] \quad ;$$

$$C_2 = \frac{1}{2}\left[A_4 \cos\theta_4 - B\cos m\right] \; ; \; D_2 = \frac{1}{2}\left[A_4 \sin\theta_4 + A_6 - B\sin m\right] \quad ;$$

$$\gamma_1 = \tan^{-1}\left(D_1/C_1\right) \; ; \; \gamma_2 = \tan^{-1}\left(D_2/C_2\right) \quad ;$$

$$E_1 = \left(C_1^2 + D_1^2\right)^{1/2} \; ; \; E_2 = \left(C_2^2 + D_2^2\right)^{1/2} \quad .$$

The attenuation constants for the nth mode (symbolized here by α_1 and α_2) are given by

$$\alpha_1 = \pm\left(\mu_1\right)^{1/2} = \epsilon_1 U_1 e^{i\eta_1} = k_1 + i\ell_1 \quad ,$$

$$\alpha_2 = \pm\left(\mu_2\right)^{1/2} = \epsilon_2 U_2 e^{i\eta_2} = k_2 + i\ell_2 \quad ,$$

where

$$U_1 = E_1^{1/2}, \; U_2 = E_2^{1/2}, \; \eta_1 = \gamma_1/2, \; \eta_2 = \gamma_2/2 \quad ,$$

$$k_1 = \epsilon_1 U_1 \cos\eta_1, \; \ell_1 = \epsilon_1 U_1 \sin\eta_1 \quad ,$$

$$k_2 = \epsilon_2 U_2 \cos\eta_2, \; \ell_2 = \epsilon_2 U_2 \sin\eta_2 \quad ,$$

and ϵ_1 and ϵ_2 are chosen so that $\epsilon_1 \cos\eta_1$ and $\epsilon_2 \cos\eta_2$ are both negative ($\epsilon_1 = \pm 1$, $\epsilon_2 = \pm 1$).

Next, the following quantities are calculated:

$$\phi' = h - m \quad ,$$

$$g = -(f/q) = Ge^{i\nu} \quad ,$$

where

$$G = -f/\left(\sigma^2 + \Omega^2\right)^{1/2} = -\left(f/A_1\right) \quad ,$$

$$\nu = -\theta_1 \quad .$$

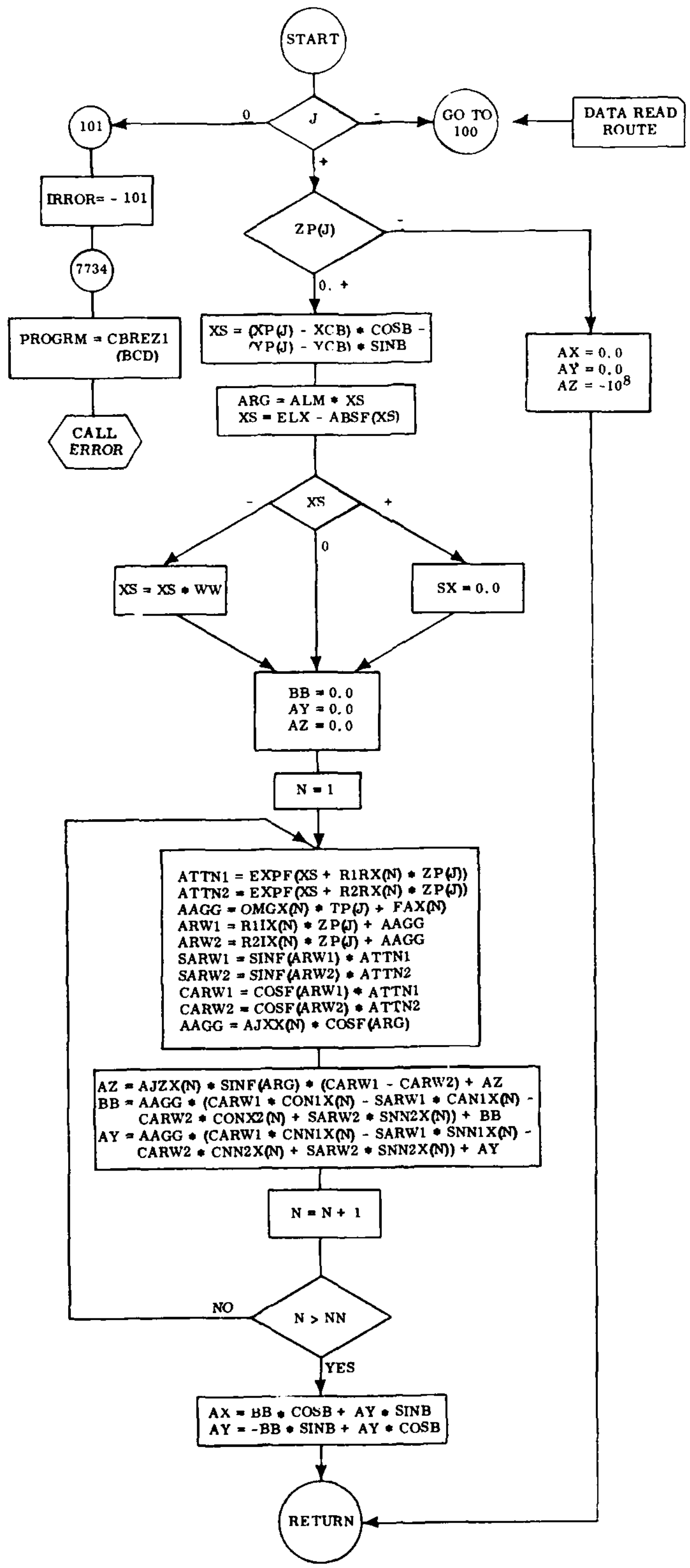
At this point all the mode-dependent constants necessary to describe the nth mode wind field have been computed.

Subroutine HEIGHT (X, Y, H) (FC-16)

Subroutine HEIGHT puts into argument H the topographic height at horizontal position X, Y. It makes use of the in-core topographic data block in arrays S(I, J) and SUBSID(K), and data from the topographic table of contents as transferred to block limit words BXLL, BYLL, BXLU, and BYLU, as well as the block grid interval as found in GRINT and the overall topography coordinate limits TXLL, TYLL, TXLU, and TYLU. (See the discussion of the topography data input in the User Information section.)

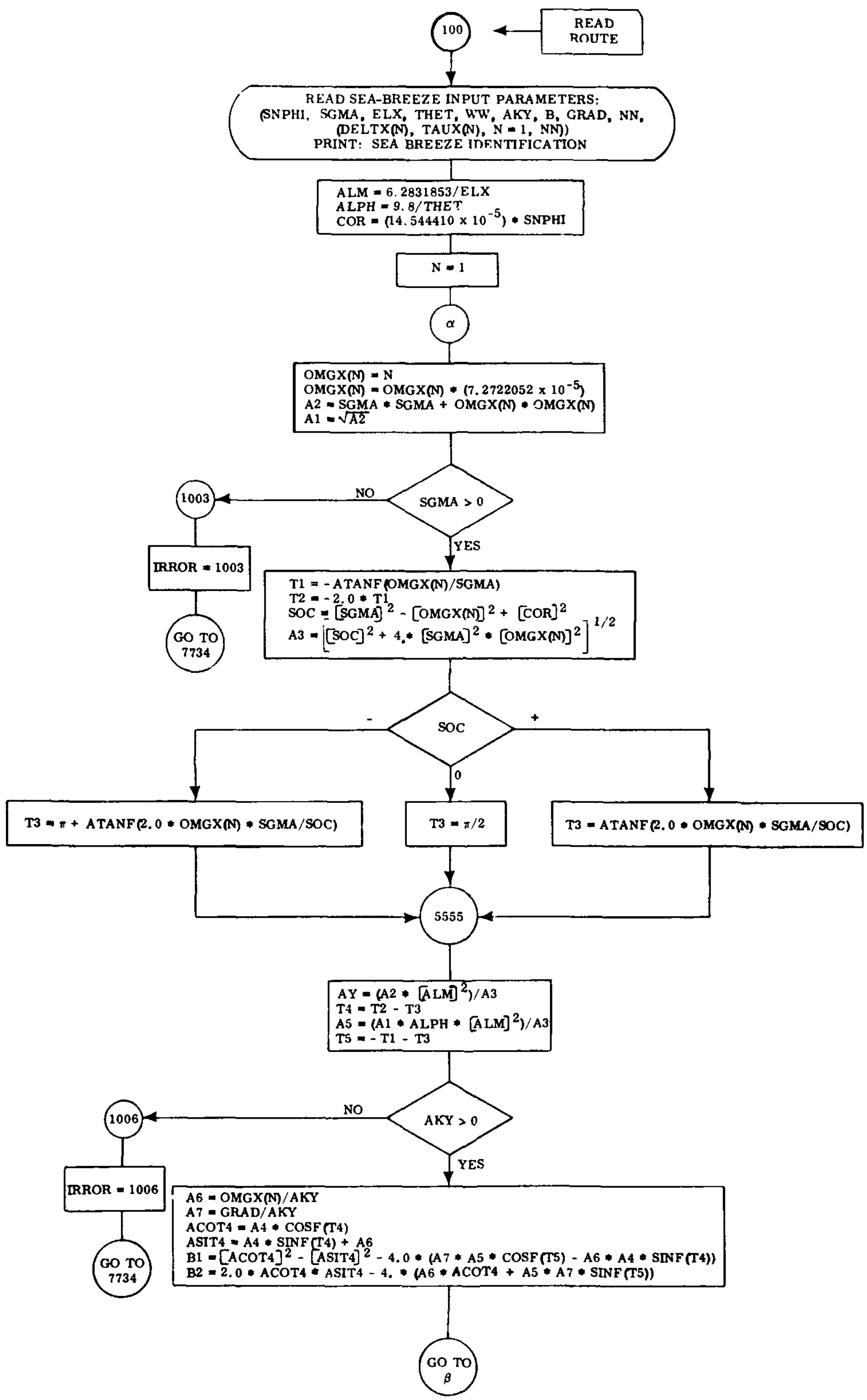
Upon entrance, the particle coordinates X and Y are checked first to determine if the particle is over the in-core topography block. If the particle is over in-core topo, a transfer is made to statement number 11 where retrieval begins. If it is not, a second check is made to determine if the particle is over any specified topography block. If it is over a topography block not currently in core, this is indicated by setting H = - 10000. If it is over undefined topography then it sets H = - 20000. In either case, control then returns to the calling program.

Actual height retrieval begins at statement number 11 with the computation of basic retrieval indices I and J. I and J are respectively the indices of the regular grid square of side GRINT in which the point XX, YY is located. The point BXLL, BYLL is the southwesternmost point with the in-core topo data block and it is



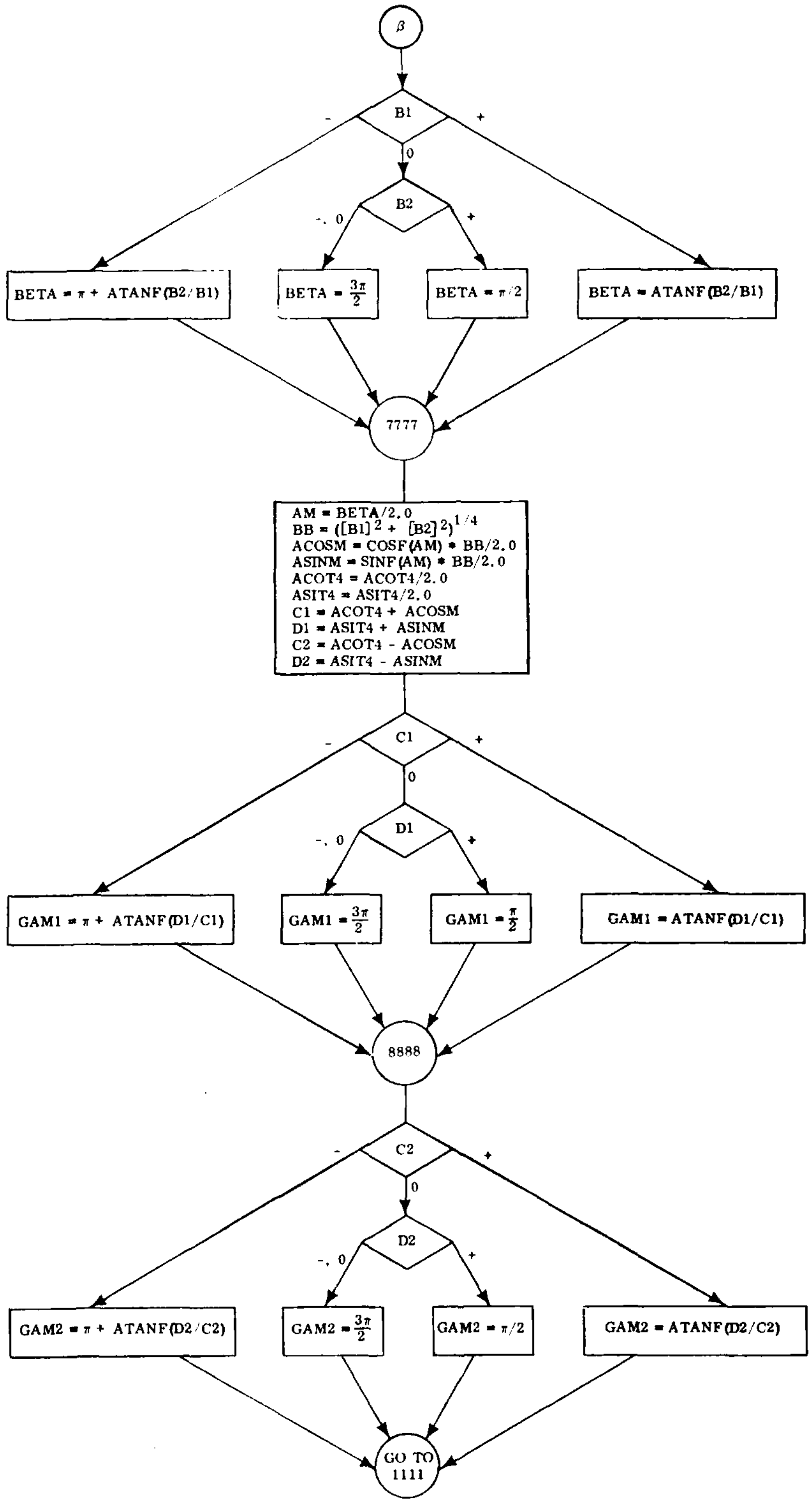
(a)

FC-15. Flow Charts for Subroutine CBREZ1



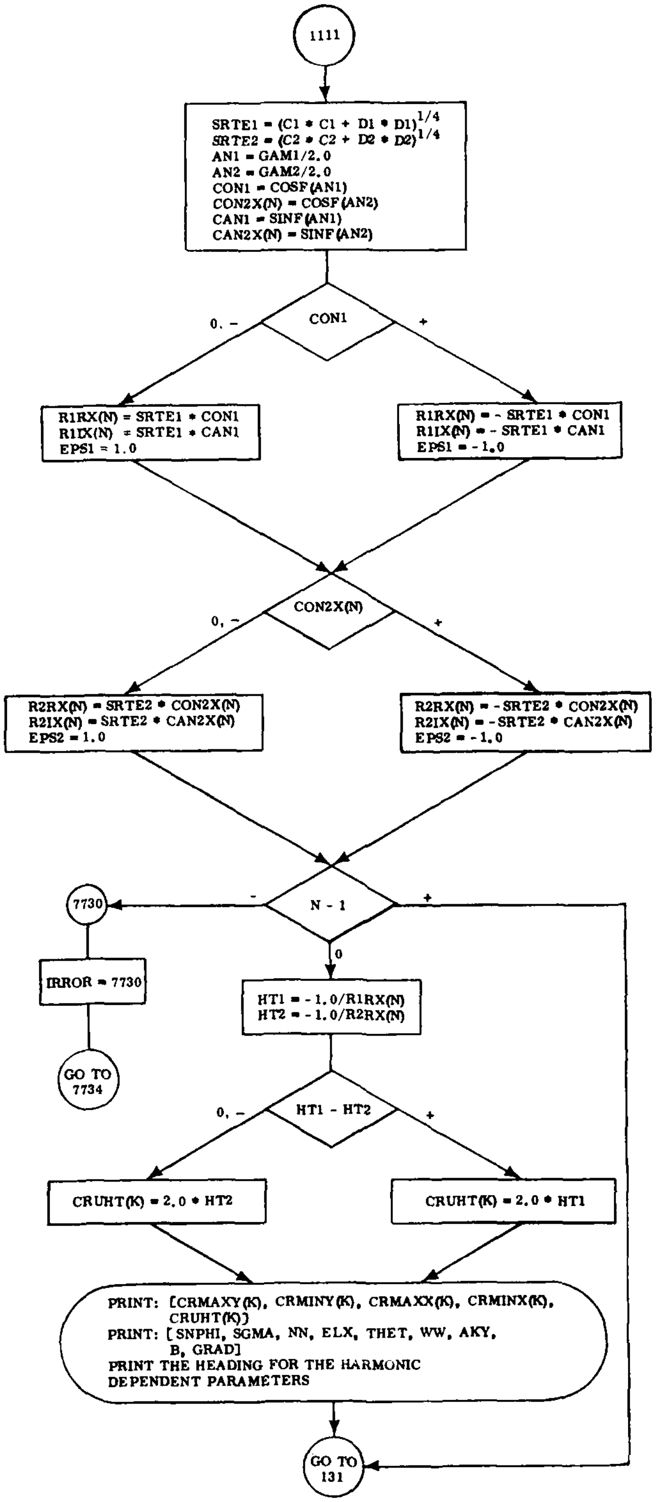
(b)

FC-15. (Continued) Flow Charts for Subroutine CBREZ1



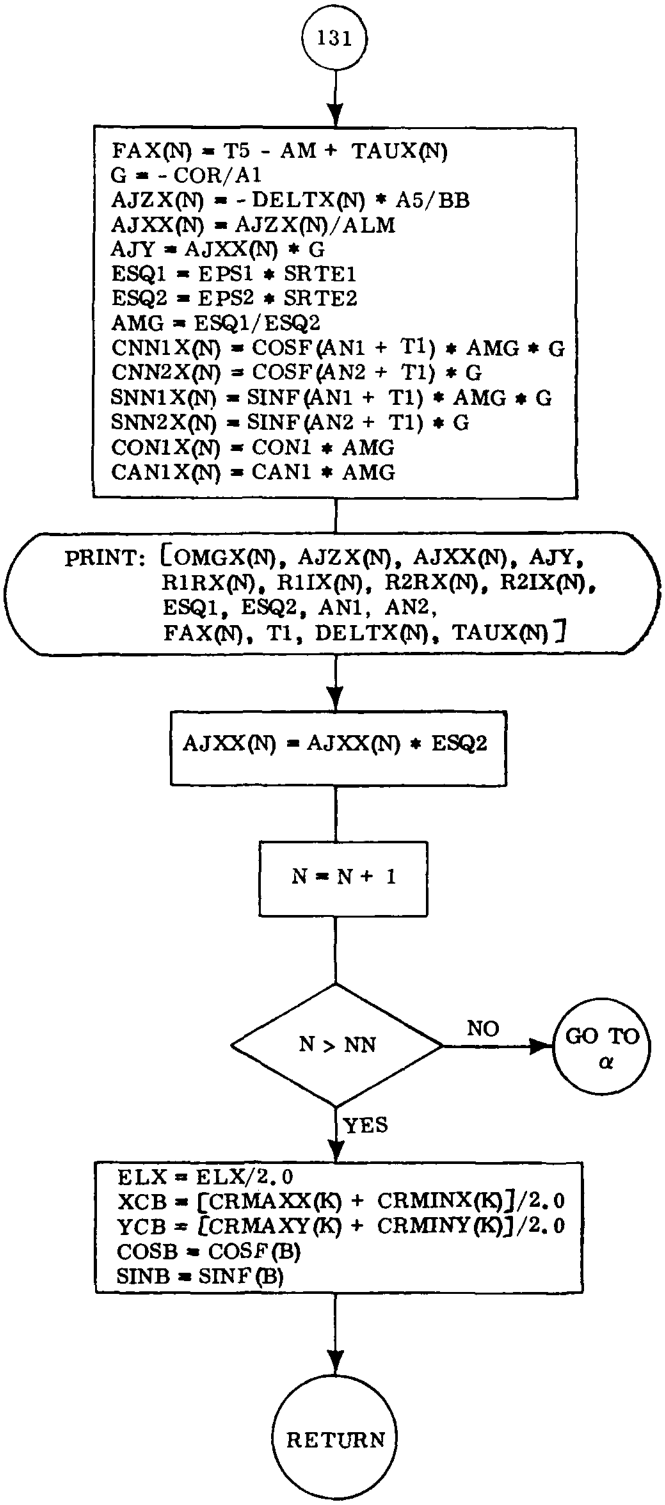
(c)

FC-15. (Continued) Flow Charts for Subroutine CBREZ1



(d)

FC-15. (Continued) Flow Charts for Subroutine CBREZ1

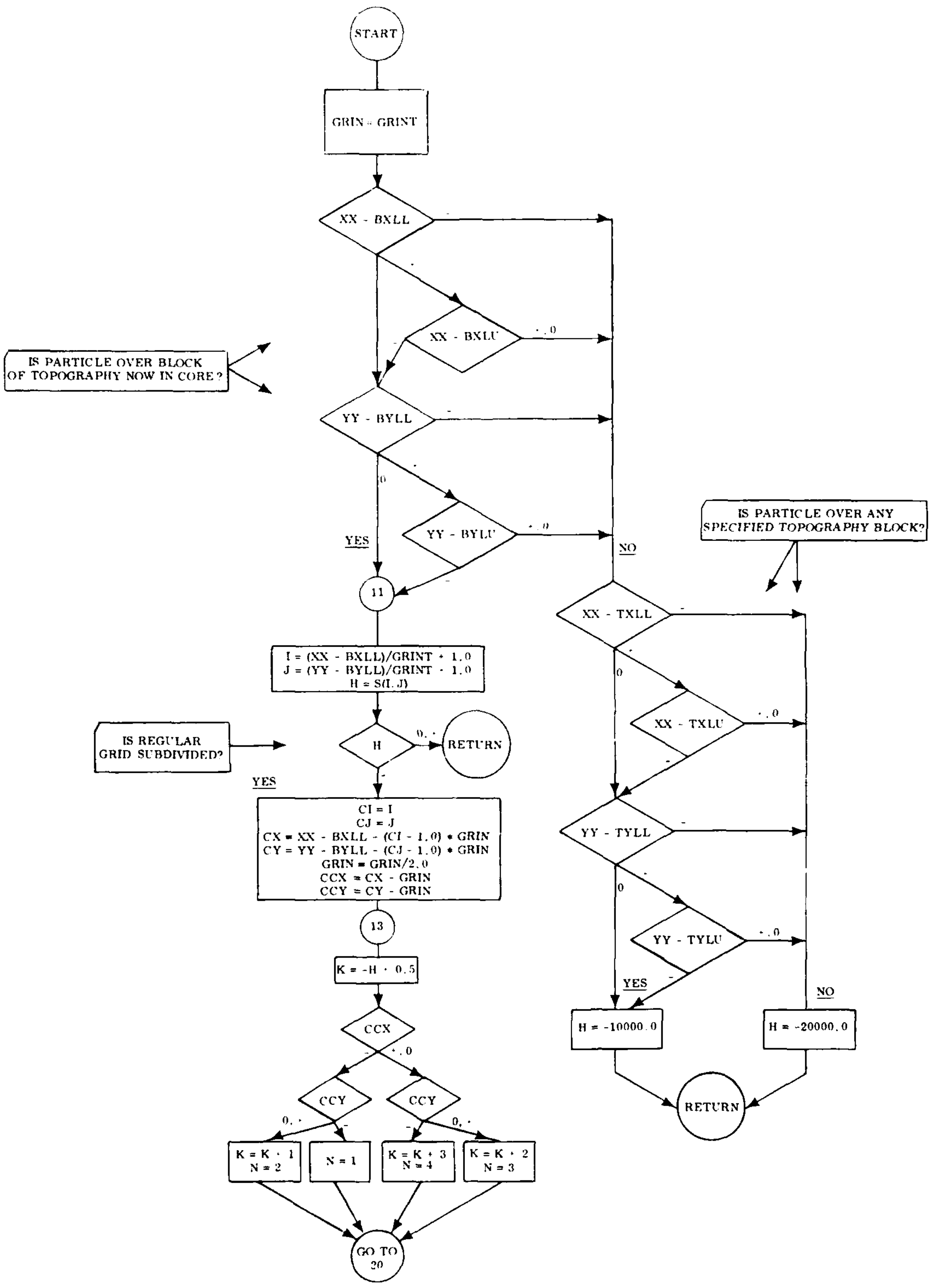


(e)

FC-15. (Continued) Flow Charts for Subroutine CBREZ1

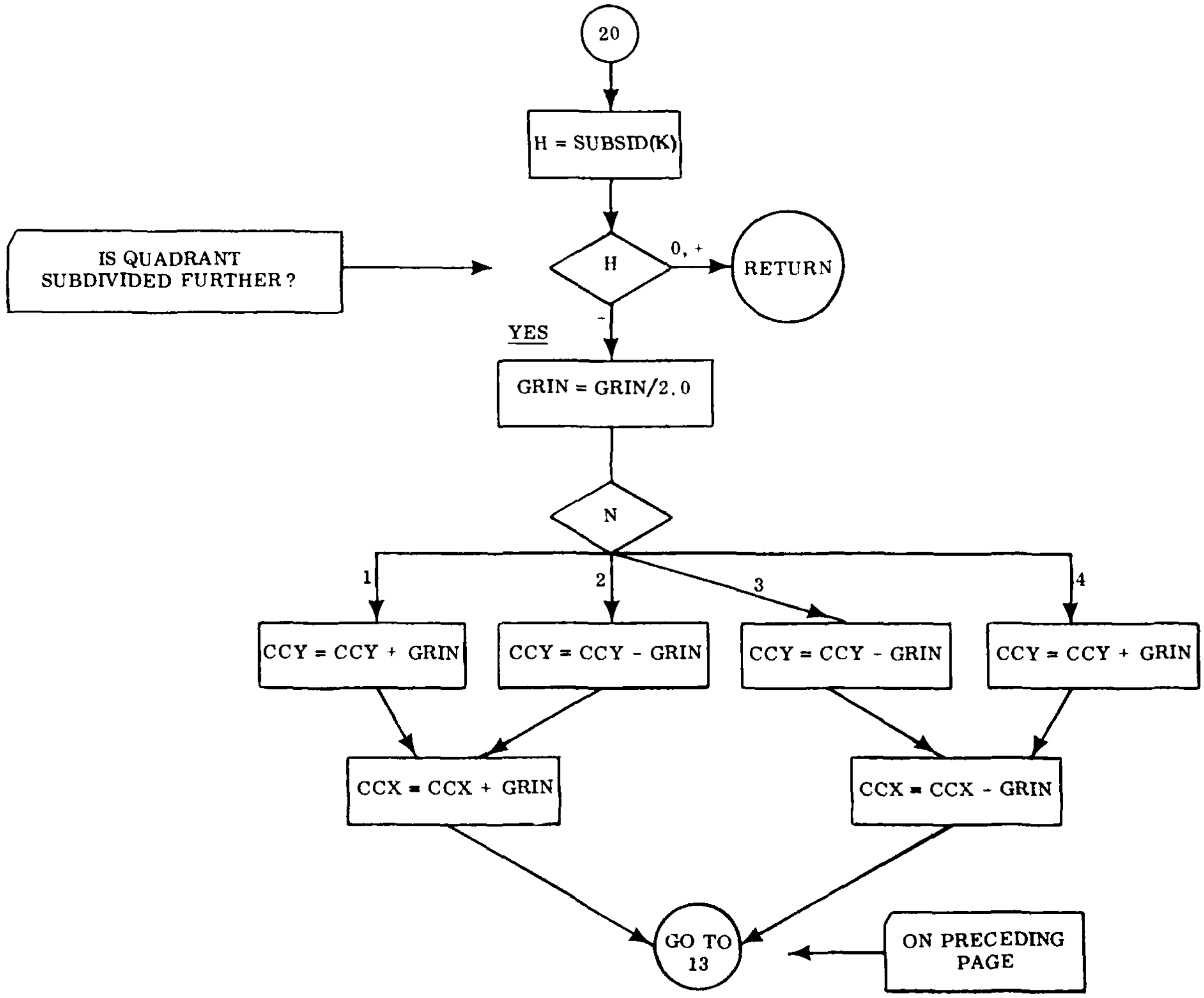
located in grid square (1, 1), the height data of which is located in memory word S(1, 1). According to the storage convention of the piecewise-planar topography, if H = S(I, J) is positive, it is the height of the topography in the cell (I, J) and thus the height which is being sought. If, on the other hand, H is negative, the cell (I, J) is subdivided further and $|S(I, J)|$ is the address (index) of the place in array SUBSID(K) where the data for the subdivided cell begins. If H is negative, the program makes preparations to retrieve topographic data from array SUBSID(K), and then between statements number 13 and 20 computes retrieval index K and control integer N. Whenever a cell (topographic unit) is subdivided it is always divided into four equal-sized squares (quadrants). The integer N identifies the quadrant that contains the point XX, YY.

The sign convention of S(I, J) also applies within array SUBSID, and if a negative entry is encountered, a further subdivision of the cell containing the point XX, YY is indicated. In that case coordinate adjustments are made again, and the program returns to statement 13 where the new retrieval index K is computed and used. Eventually HEIGHT will find a positive height H and return to the calling program.



(a)

FC-16. Flow Charts for Subroutine HEIGHT



(b)

FC-16. (Continued) Flow Charts for Subroutine HEIGHT

USER INFORMATION

Input Description

General

The Transport Module requires two kinds of input: (1) a binary tape output from the Cloud Rise-Transport Interface Module, and (2) a set of card inputs to be read from the system (IBSYS) input tape. The binary input tape carries the identifier IPARIN so that the program can ascertain that the correct tape has been mounted. This tape provides a detailed description of a large number of cloud subdivisions that are ready to be processed by the Transport Module. The Cloud Rise-Transport Interface Module produces two structurally identical binary output tapes, both labeled IPARIN, that (1) describe an axially symmetric cloud defined at some time of stabilization and (2) describe an asymmetric cloud resulting from the adjustment of the stabilized cloud in accordance with the winds that existed during the period of cloud rise. Either one of these tapes can be used as input to the Transport Module. It is important to note that in neither case are all cloud subdivisions defined at the same time. The content and structure of tape IPARIN is described in detail in Table 5.

Card inputs to the Transport Module consist of two classes: first, identification and control information; and second, wind-field information. Since the wind-field data required depends on what options are to be used, we cannot describe the deck of card inputs to the Transport Module in an invariant form; therefore, we shall describe first only the invariant portion of the deck and later provide individual descriptions of the data required by the various options.

The first card input required for the Transport Module is an identification card on which the user may punch any alphanumeric characters to identify his run of the Transport Module. The second card contains the values of the array of parameters for use in controlling the execution of the Transport Module. Only 8 of the 18 elements of the array IC have been given functions at this time and their uses are summarized in Table 6. The remaining parameters are for use in future improvements or simplifications of the Transport Module.

TABLE 5

BINARY INPUT
Tape IPARIN

Logical Binary Record	Record Contents	Variable Names
1	Tape identification word (IPARIN), spare, x and y coordinates of ground zero, shot time, cloud subdivision edge length, spare	DENTI, NSP, XGZ, YGZ, TGZ, BZ, NCL
2	Detonation parameters: yield, cloud soil burden, soil solidification temperature, soil solidification time, ln(SD),† spares	FW, SSAM, SLDTMP, TMSD, SIGMA, SPARE1, SPARE2, SPARE3
3	LINK4 run identification	PSEID(J), J = 1, 12
4	Cloud-rise identification	CRID(J), J = 1, 12
5	Initial-conditions run identification	DETID(J), J = 1, 12
6	Fallout particle density	ROPART
7	Number of particle size ranges	NPS
8	Central particle size, associated mass, associated activity,* and surface-to-volume ratio for each size range	PS(I), A(I), PACT(I), SV(I), I = 1, NPS
9	Number of atmospheric strata	NAT
10	Atmospheric viscosity and density for each stratum	ATEMP (I), RHO(I), I = 1, NAT
11	Number of particles described in the first data block	NP
12	Particle data for first data block: x, y, z, and time coordinates, particle size, mass per unit area of cloud subdivision bottom	XPAR(I), YPAR(I), ZPAR(I), TP(I), PSIZ (I), SMAS(I), I = 1, NP
13	Same as record 11 for the second data block	
14	Same as record 12 for the second data block	
.		
.		
.		
M	Tape termination indicator	NP = O
M + 1	End of file	

*Not yet calculated unless the user has provided a LINK3 particle activity calculation.

†See LINK1 glossary in DASA-1800-II.

TABLE 6

DESCRIPTION OF THE TRANSPORT CONTROL ARRAY —
IC(I), I = 1, 8

I	Function
1	IC(1) > 0, suppresses usage of topography tape, IHTOPO, and a planar topography is assumed.
2	IC(2) > 0, suppresses usage of off-topo secondary tape, IOTOPO
3	IC(3) > 0, suppresses usage of out-of-wind-field secondary tape, IOWIND
4	IC(4) > 0, suppresses usage of time-boundary secondary tape, IPAROT
5	IC(5) > 0, suppresses usage of all secondary tapes
6	IC(6) < 1, no transport traces are printed IC(6) = 1, in-core particle arrays are printed following read-in of each block of particles from IPARIN (see p. 78) IC(6) > 1, in addition, a print-out is executed following each transport increment (see p. 78)
7	IC(7) = 1, causes the computed wind field to be printed each time it is updated (see Table 13)
8	IC(8) = 0, causes a listing of lost particles (see Table 2) whenever a group of lost particles are discarded by subroutine DUMPP.

The third card indicates the latest simulated time at which the user wishes the transport process to terminate. The fourth card indicates the altitude of the deposition surface (topography) in the event that the planar topography option of the Transport Module is to be used (IC(1) = 0). The fifth card is an identification card on which the user may punch any alphanumeric characters to identify the forthcoming wind data set. Table 7 summarizes the card inputs for identification and control of the Transport Module.

TABLE 7

CARD INPUTS FOR IDENTIFICATION AND CONTROL OF THE TRANSPORT MODULE

Card Number	Content	Variable Names and Format
1	Transport model run identification	TID(J), J = 1, 12 (12A6)
2	Control integer array	IC(J), J = 1, 18(18I4)
3	Transport time limit (sec)	TLIMIT (F10.5)
4	Altitude of planar topography. This card is to be omitted if a topography input tape is used (m).	TTOPO (F10.5)
5	Wind-field data set identification card	WID(J); J = 1, 12 (12A6)

The remaining card inputs describe the wind field through which particle transport is to be carried out. As mentioned previously, temporal variation of the atmosphere is achieved by periodically updating the entire wind field description. Input data is required for each updating of the wind field, but since the form of the required data deck is the same in each case we shall describe it only once.

MKWIND Data

The first card contains the values of parameters ENDTIM, the time (seconds) at which the forthcoming data set should be updated, and ALPHA and BETA empirical parameters which the program uses for distance weighting (see Eq. (21)).

The second card contains NN, the number of data vectors that are to be used in computing the vector estimate for each wind cell of the wind field. The NN data vectors that are closest to the grid point are used. Also on the second card is the parameter NCODE which identifies the computation method to be used in accordance with Table 8.

TABLE 8

WIND-FIELD COMPUTATION METHODS
SPECIFIED BY NCODE

NCODE Value	Method
1	Use preferential weighting method with the nearest NN data vectors
2	Set NN = 1 and use code number 1 (this is the nearest station method)
3	Set NN equal to the total number of data vectors available and use code number 1
4	Use the least-squares method to fit to a linear model of the atmosphere. In this case NN must be greater than 3.

In the next series of cards the program reads the user's specifications for the subdivisions of the stratum and cell atmospheric structure. Each card of this set contains the altitude of a stratum bottom (meters), the width of the wind cell bottom edges (assumed square) within this stratum (meters), and four coordinates that indicate the horizontal limits of this stratum (meters). Here also, the data cards need not be in ascending order of altitude since they are sorted into that order by the program after being read, but the end of the data set must be marked by a card having the value 999999.0 in the stratum base altitude position.

In the next series of cards the program reads all wind data vectors, one to a card. The position of each vector is specified by three coordinates; its magnitude and direction are specified by three vector components. The order of these cards is completely immaterial, but the end of the deck of data vectors must be marked by a card having the value 999999.0 in the vector altitude position. A maximum of 299 data vectors may be provided. Table 9 summarizes the card input to MKWIND.

TABLE 9

SUMMARY OF CARD INPUTS TO SUBROUTINE MKWIND

Card Number	Content	Variable Names and Format
1	Time (seconds) at which the forthcoming wind data set is to be updated, α, β (Eq. 21)	ENDTIM, ALPHA, BETA (3F10.3)
2	The number of nearest data vectors that the user wishes the program to use in making a vector estimate for each grid point, the identification number of the computation method that the user wishes to be used in making grid point vector estimates (see Table 8).	NN, NCODE (2I4)
3	Altitude of first stratum base (meters above MSL), width of wind cells in the stratum, coordinate of planes limiting this stratum on the west, south, east, and north respectively. (A right-handed coordinate system is used.)	BOTHIT(J) WGRINT(J), WLLX(J), WLLY(J), WURX(J), WURY(J), J = 1 (6F10.3)
4	Same as card 3 but for second stratum	Same as card 3 but for J = 2
. . .	. . .	. . .
Last of Subdivision Specifications	The end of the subdivision specifications is marked by the number 999999.0 in the stratum base altitude place	
First Data Vector	Vector altitude, X coordinate, Y coordinate, X-velocity component, Y-velocity component, Z-velocity component (A west wind (from the west) has a positive X component; a south wind has a positive Y component; the Z direction is positive upward) (m and m/sec)	ZS(J), XS(J), YS(J), SX(J), SY(J), SZ(J), J = 1 (6F12.3)
Second Data Vector	Same as preceding card but for second data vector	Same as preceding card but for J = 2
. . .	. . .	. . .
Last Vector Card	The end of the deck of data vectors is marked by the number 999999.0 in the vector altitude position	

Local Circulation System Data

Two types of data are required for the description of local circulation systems to be included within a transport atmosphere. First are data that specify the sizes and locations of all local circulation cells, and second are the data that describe the wind fields within each of the local circulation cells. Data of the first type are read by subroutine RDCIRS, while the data actually describing the local systems must be read by the corresponding local circulation system programs. To achieve this the local system programs have dual purposes — dependent upon an argument value, these programs will either (1) read the required input data from the system (IBSYS) input tape (and precompute certain parameters) or (2) compute the wind vector at a position specified in its argument list.

TABLE 10

CARD INPUTS TO SUBROUTINE RDCIRS

Card Number	Content	Variable Names and Format
1	Coordinates of planes that bound the Jth local circulation system cell on the west, east, south, and north, respectively, and the circulation type identifier.	CRMINX(J), CRMAXX(J) CRMINY(J), CRMAXY(J) NCRTYP(J) (4E12.5, I3)
Last Card	The end of the deck of cell descriptions is marked by a card having a circulation type identifier of zero (a blank card will do). Note that if no local circulation cells are to be used in a transport run, a blank card must still be provided to RDCIRS.	Blank

Table 10 summarizes the input cards to subroutine RDCIRS. The first card read by RDCIRS contains the coordinates of the four planes (perpendicular to the coordinate axes) that bound a local circulation cell and also a number that identifies the type of associated local circulation system according to the following designations:

Identification Number	Local Circulation Type
0	Marks the end of the set of local circulation cell descriptions
1	Mountain wind (MTWND1)
2	Ridge wind (RGWND1)
3	See breeze (CBREZ1)
4	Not assigned
5	Not assigned

The reading of all descriptions is terminated when a blank card is encountered; therefore, if no local circulation systems are in use, a blank card is still required by RDCIRS. The maximum allowable number of local circulation systems is currently set at 5.

RDCIRS establishes the order of the entries in a table of local cell descriptions by storing the cell data sequentially as it is read. Later calls are made to the associated local circulation system programs in the established sequence so that these programs may read the data that they require. Table 11 presents a summary of the data decks required by each of the three available local circulation programs. More detailed descriptions of these data may be found in the individual discussions of the local circulation system programs. (Mks units are used.)

Topography Data

Two basically different forms of topography may be specified for use by the Transport Module in regions not covered by local circulation systems. They are referred to here as fully planar topography (a single plane) and piecewise-planar topography (many segments of planes). The choice of method of topographic description is communicated to the Transport Module by the user in the control parameter IC(1) (see Table 6) which must be given the value 1 if the fully planar option is desired and 0 if not. In the fully planar option, the program merely reads from a

TABLE 11

SUMMARY OF CARD INPUTS TO SUBROUTINES MTWND1, RGWND1, AND CBREZ1

Card Number	Content	Variable Names and Format
	Subroutine MTWND1	
1	X and Y coordinates. Maximum height, and half width of the Jth mountain	XM(J), YM(J), H(J), A(J), (4F10.3), J = 1
2	Same as card 1 but for 2nd mountain	Same as card 1 but for J = 2
.	.	.
.	.	.
.	.	.
Last Card	The end of the deck of mountain descriptions is indicated by a card having a zero in the mountain height position	
	Subroutine RGWND1	
1	X and Y coordinates of a point on the 1st ridge line, height of 1st ridge, half width of 1st ridge, orientation angle of 1st ridge (radians clockwise from time north)	XM(J), YM(J), H(J), A(J), B(J), J = 1 (5F10.3)
2	Same as card 1 but for 2nd ridge	Same as card 1 but for J = 2
.	.	.
.	.	.
.	.	.
Last Card	The end of the deck of ridge description is marked by a card having a zero in its ridge height position	
	Subroutine CBREZ1	
1	Sine of the latitude of the sea-breeze cell, Guldberg-Mohn friction parameter, the total extent of the sea breeze, average ground temperature	SNPHI, SGMA, ELX, THET, (4F10.3)
2	Wind-field extrapolation attenuation constant, thermal eddy diffusivity, coastline orientation angle, unperturbed temperature gradient, number of harmonics used in temperature-time description	WW, AKY, B, GRAD, NN, (4F10.3, I10)
3	Magnitude of 1st temperature differential, phase of 1st temperature differential	DELTX(N), TAUX(N), N = 1
4	Same as card 3 but for 2nd harmonic	Same as card 3 but for N = 2
.	.	.
.	.	.
.	.	.

card the height of the planar topographic surface and uses it throughout transport. If the piecewise-planar option is specified, the program expects that a topographic data tape has been prepared and is available for use. This tape carries the identification word IHTOPO and its data structure is indicated in Table 12. Complete details are given in Appendix C. (Mks units are used.)

TABLE 12

THE BINARY TOPOGRAPHY TAPE DATA

Record Number	Content	Variable Names
1	Tape identification symbol (IHTOPO), overall topography area limits, and the number of data blocks	DENTI, TXLL, TXLU, TYLL, TYLU, NBLCK
2	Arbitrary topographic identification card image	TOPID(J), J = 1, 12
3	Topography table of contents (first part) for all data blocks	TOPOLM(I, J), I = 1, 4, J = 1, NBLCK
4	Topography table of contents (second part) for all data blocks	ITOPLM(I, J), I = 1, 3, J = 1, NBLCK
5	2-D table of data for first data block	S(I, J), I = 1, II, J = 1, JJ
6	1-D table of data for first data block	SUBSID(K), K = 1, KK
7 8	Same as records 5 and 6 but for second data block	
. . .		
N	End of file	

The Transport Module uses subroutine RDTOPO to read blocks of topographic data into memory from the tape IHTOPO. Subroutine HEIGHT is used to determine the elevation of the topographic surface at the horizontal position of a particle. Two other programs, TOPIN and DATERR, which are not strictly part of the Transport Module, have been written to prepare and check the topographic data tape and to write the piecewise-planar topography tape IHTOPO. Since these programs are out of the main stream of the Transport Module, their inputs, operations, and outputs will not be described here, but rather are dealt with in Appendix C. However, the contents of tape IHTOPO are as follows.

The topographic data must be divided into blocks and only one block at a time can be accommodated in core storage during transport. With reference to Table 12, we see that the first record consists of the tape identifier, the coordinate limits of the area covered by all the data blocks on the complete topography tape, and the number of topography data blocks on the tape. The second record consists of a Hollerith card image that contains a descriptive comment that identifies the particular topographic data on the tape. To describe the contents of the remaining records, we must briefly review, as follows, the nature of the topography description.

1. Consider the topographic unit to be a surface segment that projects a square area onto the z = 0 plane such that the sides of the projected square are parallel to the coordinate axes (north-south and east-west).

2. Location coordinates of all topography units are specified in the z = 0 (horizontal) plane of the macrowind-field coordinate system.

3. Topography descriptions are arranged on tape IHTOPO in data blocks, each of which consists of arrays ((S(I, J), I = 1, II) J = 1, JJ) and (SUBSID(K), K = 1, KK).

4. Array S represents a rectangular area in the z = 0 plane (with sides parallel to the x and y axes) that otherwise is arbitrarily placed within the limits of the overall topo area. Its minimum x and y coordinates are BXLL and BYLL (in meters). It is subdivided by a square grid with interval GRINT (meters). Each element S(I, J) of array S has the following significance:

   a. If S(I, J) is positive, then S(I, J) is the altitude of the (I, J)th topography unit it represents in the array area (meters above mean sea level).

   b. If S(I, J) is negative, then the fixed-point equivalent of |S(I, J)| is the index of an element in array SUBSID that is the first element of a quartet (see item 5 below).

The indices I and J of the S(I, J) array represent increments of distance GRINT along the x and y axes respectively. S(1, 1) represents the grid element in the lower left corner of the area. S(2, 1) is the next element to the right of the corner, S(1, 2) is the element just above the corner, etc.

5. Array SUBSID consists of a sequence of groups of four elements (quartets) each of which represents the four square areas (topography units) resulting from an equal subdivision of a topography unit. Each element SUBSID(K) of array SUBSID has the following significance:

   a. If SUBSID(K) is positive, it is the altitude of the topography unit it represents (meters above mean sea level).

   b. If SUBSID(K) is negative, then the fixed-point equivalent of $|\text{SUBSID(K)}|$ is the index of an element in array SUBSID that is the first element of a quartet.

We see that array SUBSID allows (in principle) an unlimited capability for successive subdivision of the original topography units defined in array S. Furthermore, a unique altitude is specified for each topography unit that results finally from the successive subdivision process. The sequence numbering of quartet members is as follows: lower left SUBSID(K), upper left SUBSID(K+1), upper right SUBSID(K+2), lower right SUBSID(K+3).

The correspondence between arrays S and SUBSID is as follows. Picture the array S to be set up in the fashion of a conventional matrix

$$
\begin{array}{ccccc}
S(1,1) & S(1,2) & S(1,3) & \ldots & S(I,JJ) \\
S(2,1) & S(2,2) & S(2,3) & \ldots & S(2,JJ) \\
\vdots & & \vdots & & \vdots \\
S(II,1) & S(II,2) & S(II,3) & \ldots & S(II,JJ)
\end{array}
$$

The sequence of quartets in the array SUBSID is determined by scanning through each row of the S matrix, in its numerical sequence, from left to right. Each negative element so encountered in the matrix starts the next quartet in SUBSID.

With reference to the discussion above we now can define records 3 and 4 on tape IHTOPO. Together these records provide a complete table of contents for the remainder of the tape by defining all of the data blocks on the tape. For each of the arrays TOPOLM(I, J) and ITOPLM(I, J), the index J identifies the data block sequence number (J = 1, 2, 3, 4). The index I specifies the parameters:

I	TOPOLM	ITOPLM
1	BXLL	II
2	BYLL	JJ
3	GRINT	KK
4	TTOPO	

The variable TTOPO gives the maximum topography altitude specified in the data block. All distances are specified in meters, and altitudes in meters above mean sea level.

Then on the tape IHTOPO the arrays S and SUBSID follow for each data block.

Output Description

Printed Output

The printed output of the Transport Module is largely self-explanatory since extensive labeling is done. Table 13 presents a summary of this output. Not included are the (optional) transport trace printouts which are described in the discussion of subroutine LINK7.

Binary Output

The primary output of the Transport Module is a magnetic tape containing a binary mode complete description of all cloud subdivisions that landed during the transport run. In addition, the Transport Module prepares printed output designed to identify and describe the transport run in sufficient detail so that the resulting

TABLE 13

PRINTED OUTPUT OF THE TRANSPORT MODULE

Output Sequence	Content	Variable Names
1	Run identifiers for LINK1, LINK2, LINK4, and transport	DETID(J), J = 1, 12 CRID(J), J = 1, 12 PSEID(J), J = 1, 12 TID(J), J = 1, 12
2	Transport control array (Table 6)	IC(J), J = 1, 18
3	Transport time limit (sec)	TLIMIT
4	Fallout particle density (kg/m^3)	ROPART
5	Topographic data: a. If continuous planar topography is specified, the topography altitude (meters) is printed. b. If a piecewise planar topography is specified, the topography tape (IHTOPO) identifier is printed.	 TTOPO TOPOID(J), J = 1, 12
6	Wind-field identifier	WID(J), J = 1, 12
7	Atmospheric properties used for particle fall rate calculations: height of stratum bottom, viscosity, and density (mks units)	height not stored, ATEMP, RHO
8	Replacement time of the wind field whose description follows (items 9 and 10) (sec)	ENDTIM
9	Wind vector input data array: z, x, y coordinates, and x, y, z wind vector components (meters and meters sec^{-1})	ZS(J), XS(J), YS(J), VX(J), VY(J), VZ(J)
10	Macrowind-field definition input data array: bottom height of stratum, grid interval, minimum x and y coordinates, maximum x and y coordinates (all in meters)	BOTHIT(J), GRINT(J), WLLX(J), WLLY(J), WURX(J), WURY(J)
11	If IC(7) = 1, the wind vectors at each grid point of the macrowind field are printed in the following arrangement: Level (stratum) number, altitude of the bottom of the stratum, x components of all wind vectors in the southernmost east-west row, y components of all wind vectors in the same row, z components for all vectors in the same row, repeat for the next row, etc.; repeat for the next level, etc.	J, BOTHIT(J), VS, VY, VZ
12	A one line in-core particle array summary printout is executed on each pass through subroutine DUMPP:JTEST, JTEST1 (Table 2), number of blanks, number of grounded particles, number of lost particles, number of particles on the topography boundary, number of particles on the time boundary, and number of particles on the wind-field boundary.	JTEST, JTEST1, NFREE, NG, NLOST, NTO, NTI, NW
13	If IC(8) = 0, properties of all "lost particles" are printed: z, y, z coordinates, time, diameter, and mass per unit area.	XP, YP, ZP, TP, PS, FMAS

tape of grounded cloud subdivisions can be used repetitively. This is achieved by printing the identifiers of all the sets of input data that were used by the Transport Module as well as recording some of the data directly. The content of the intermediate output tape produced by the Transport Module and subsequently used by the output processor as an input is set forth in Table 14. Mks units are used except for particle diameters which are in microns.

Data Structures for Secondary Memory Tapes

Three secondary memory tapes may be used by the Transport Module to temporarily record descriptions of particles that have been transported as far as possible using the data currently available in primary memory but which are still to be transported further. In the event that room must be made in the Transport Module's particle arrays for incoming particle descriptions, the transported (but not yet grounded) particles may be collected and written out onto a tape for: (1) particles beyond the in-core memory topography, (2) particles beyond the in-core memory wind field, or (3) particles awaiting the next updating of the wind field. Since all of these tapes are subsequently put symbolically into the place of the regular particle input tape IPARIN, they must all have the same data structure as the particle data portion of IPARIN. That structure consists of pairs of data records arranged in sequence on the tape -- the first record of a pair is a count of the number of particle descriptions to be found in the second record of the pair. The end of the data set is always marked by a zero particle count record.

TABLE 14

THE GROUNDED PARTICLES TAPE, IPOUT
(Binary output of the Transport Module)

Logical Record Number	Record Content	Variable Names
1	Identification word (IPOUT), spare, time at which transport was terminated, width of cloud subdivisions at time of definition, and density of fall-out particles	POUT, NCL, TLIMIT, BZ, ROPART
2	Fission yield, mass of soil lifted, solidification temperature, time of solidification, ln(SD)† and 3 spares	FW, SSAM, SLDTMP, TMSD, SIGMA, SPARE1, SPARE2, SPARE3
3	Run identifiers for Initial Conditions, Cloud Rise, Cloud Rise-Transport Interface, Transport, and Wind Field	(DETID(J), J=1, 12), (CRID(J), J=1, 12), (PSEID(J), J=1, 12), (TID(J), J=1, 12), (WID(J), J=1, 12)
4	Number of particle size ranges	NPS
5	Central particle size, associated mass, associated activity,* and surface-to-volume ratio for each size range	PS(J), A(J), PACT(J), SV(J), J=1, NPS
6	Number of atmospheric strata	NA
7	Atmospheric viscosity and density for each stratum	ATEMP(J), RHO(J), J=1, NA
8	Topography identifier	TOPID(J), J=1, 12
9	Number of particle (cloud subdivision) descriptions in the following data block	N
10	X coordinate, Y coordinate, time, particle size, and mass per unit area associated with each of N particles	NP(J), YP(J), TP(J), PS(J), FMAS(J), J=1, N
11	Same as record 9	
12	Same as record 10	
	Pairs of records like 9 and 10 are repeated until all grounded particles are recorded	
Last record	The end of the ground particles data set is indicated by a particle count of zero	N=0

*Not yet calculated unless the user has provided a LINK3 particle activity calculation

†See LINK1 glossary in DASA-1800-II.

REFERENCES

1. H. G. Norment, T. W. Schwenke, and I. Kohlberg, "Development of an Improved Land-Surface Fallout Model. Interim Report," Technical Operations Research, Report No. TO-B 65-99 (ECOM-01309-1, Contract DA 28-043-AMC-01309(E), AD 629-002), January 1966.

2. C. N. Davies, "Definitive Equations for the Fluid Resistance of Spheres," Proc. Phys. Soc. (London) 57, 259 (1945).

3. F. Pasquill, "Atmospheric Diffusion A Study of the Dispersion of Windborne Material from Industrial and Other Sources," (New York, N. Y.: D. Van Nostrand Co., 1962).

4. G. P. Cressman, "An Operational Objective Analysis System," Monthly Weather Review 87, 367 (1959).

5. F. Defant, "Theorie der Land-und Seewinde," Arch. Meteorol. Geophys. Bioklimatol. 2, 404 (1950).

6. F. Defant, "Local Winds," in Compendium of Meteorology (Boston, Mass.: Am. Meteorol. Soc., 1951) pp. 655-671.

7. W. E. Milne, "Numerical Solution of Differential Equations," (New York, N. Y.: Chapman and Hall Limited, 1953).

FORTRAN LISTINGS

FORTRAN listings for the subroutines are included on the following pages.

```
$IBFTC FALRA    LIST,DECK,M94/2
C     SUBROUTINE FALRAT
C     W.Y.G.ING            TECHNICAL OPERATIONS RESEARCH
C     DEC  14 1965
      SUBROUTINE  FALRAT(ALT,PSIZE,FV,ATEMP,RHO,FROG,ISOUT)
C
C*****************************************************************************
C
C     SUBROUTINE FALRAT,USING DAVIE'S EQUATIONS, COMPUTES THE SETTLING
C     RATE OF PARTICLES BY USING PARTICLE DIAMETER (MICRONS), ALTITUDE
C     (METERS ABOVE MSL) AND A STANDARD ATMOSPHERE FOR WHICH THE DENSITY
C     AND VISCOSITY HAVE BEEN TABULATED IN ARRAYS RHO(I) AND ATEMP(I)
C     RESPECTIVELY.
C
C*****************************************************************************
C
      DIMENSION ATEMP(260),RHO(260)
      REAL LOG10
C
C**********************     FALRAT GLOSSARY     ******************************
C
C     ALT            HEIGHT OF THE PARTICLE ABOVE MSL    (METERS)
C     ATEMP(I)       DYNAMIC VISCOSITY OF AIR AT ((I-1)*200) METERS ABOVE
C                    MSL.               (KILOGRAM/METER-SECOND)
C     CDRR           THE DRAG COEFFICIENT * SQUARE OF THE REYNOLD-S
C                    NUMBER.
C     FROG           (4/3)*PARTICLE DENSITY*GRAVITY*(CUBIC METERS/ CUBIC
C                    MICRON).  KILOGRAM-METER/((SQR. SEC.)*(CUBIC MICRON))
C     FV             SETTLING RATE    (METERS/SEC)
C     PSIZE          PARTICLE DIAMETER     (MICRONS)
C     RHO(J)         ATM DENSITY AT (I-1)*200 METERS  ABOVE MSL.  (KILO-
C                    GRAMS/ CUBIC METER)
C
C*****************************************************************************
C
    2 FORMAT(//38H DAVIES EQUATIONS ARE INACCURATE FOR  ,F12.3,12H MICRO
     1NS AT ,F12.3,7H METERS)
C
C ****************************************************************************
C ****************************************************************************
C
C     I IS THE INDEX (IN THE ARRAYS RHO(I) AND ATEMP(I)) THAT IDENTIFIES
C     THE 200 METER THICK LAYER CONTAINING THE PARTICLE.  THE ADDITION
C     OF 6.5 TO THE INDEX BEFORE TRUNCATION INSURES THAT PARTICLES
C     BETWEEN -100 AND +100 METERS ABOVE MSL WILL LIE IN THE SIXTH LAYER
C     WHICH HAS ITS CENTER LOCATED AT MSL, PARTICLES BETWEEN 100 AND 300
C     METERS ABOVE MSL WILL LIE IN THE  7-TH  LAYER WHICH HAS ITS CENTER
C     LOCATED AT 200 METERS ABOVE MSL, AND SO FORTH.
      I=(ALT/200.0)+6.5
      V0=PSIZE/ATEMP(I)
      V1=PSIZE*V0*FROG
      CDRR=V1*RHO(I)*V0
      IF(CDRR-140.0)100,100,150
  150 IF (CDRR-4.5E+7)200,151,151
C151 CDRR EXCEEDS THE UPPER RANGE OF DAVIE'S EQUATIONS.  HOWEVER, THE
C     NUMERICAL RESULT IS STILL USED.
  151 WRITE (ISOUT,2)PSIZE,ALT
      GO TO 200
C100 CDRR IS LESS THAN OR EQUAL TO 140.
  100 FV=V1*(41666.7 +CDRR*(-2.3363E+2+CDRR*(2.0154    -6.9105E-3*CDRR)))
      GO TO 300
```

```
C200  CDRR IS GREATER THAN 140.
 200  GLOGA=ALOG10(CDRR)-20.775
      FV=50657.0 *V1*CDRR**((GLOGA*GLOGA-445.98)*0.0011255)
C300  DRAG SLIP CORRECTED FALL RATE
 300  FV=FV*(1.0+2.55E-17/(PSIZE*RHO(I)))
 301  RETURN
      END
```

```
$IBFTC DUMP     LIST,DECK,M94/2                                         DUMP
      SUBROUTINE DUMPP                                                  DUMP
C          T.W.SCHWENKE  TECHNICAL OPERATIONS RESEARCH        SR  DUMPP DUMP
C     28 NOVEMBER 1966                                                  DUMP
C                                                                       DUMP
C***********************************************************************DUMP
C                                                                       DUMP
C     THIS SUBROUTINE SELECTS THE BEST SET OF PARTICLES TO DUMP,        DUMP
C     SORTS IT INTO THE LOW NUMBERED END OF THE PARTICLE ARRAYS, WRITES DUMP
C     IT OUT ONTO THE APPROPRIATE TAPE AND ADJUSTS PARTICLE SET COUNTERSDUMP
C     THE SET SELECTED FOR DUMPING IS THE GROUNDED PARTICLES SET WHEN-  DUMP
C     EVER DUMPING IT WOULD MAKE SUFFICIENT ROOM FOR THE INCOMING BLOCK DUMP
C     OF N PARTICLES.  IF THIS IS NOT THE CASE, THE LARGEST PARTICLE SETDUMP
C     IS SELECTED.                                                      DUMP
C                                                                       DUMP
C**********************  GLOSSARY  **************************************DUMP
C                                                                       DUMP
C     FOR ADDITIONAL GLOSSARY ENTRIES SEE SUBROUTINE LINK5              DUMP
C                                                                       DUMP
C     FMAST   SEE XPT                                                   DUMP
C     IC( )   THE CONTROL INTEGER ARRAY.    SEE LINK 5 GLOSSARY         DUMP
C     ICON    BLOCKING SORT MODE INDICATOR.  0=FIRST PASS, +=BOTTOM LOOPDUMP
C             ,-=TOP LOOP                                               DUMP
C     IOTOPO  THE OFF-TOPO MEMORY TAPE NUMBER                           DUMP
C     IOWIND  THE OUT-OF-WIND DATA MEMORY TAPE NUMBER                   DUMP
C     IPAROT  THE TIME LIMIT BOUNDARY MEMORY TAPE NUMBER                DUMP
C     IPOUT   THE TRANSPORT MODULE INTERMEDIATE OUTPUT TAPE NUMBER      DUMP
C     IRSET   A MARKER FOR THE BLOCKING SORT WHICH INDICATES BY THE     DUMP
C             VALUE  1  THAT THE TEMPORARY STORAGE LINE FOR A PARTICLE  DUMP
C             IS LOADED AND MUST BE EVENTUALLY UNLOADED                 DUMP
C     ISOUT   THE FORTRAN SYSTEM OUTPUT TAPE NUMBER                     DUMP
C     J       A GENERAL INDEX.    IN THE BLOCKING SORT, IT IS USED TO   DUMP
C             IDENTIFY THE PARTICLE THAT WAS JUST CLASSIFIED            DUMP
C     JB      BOTTOM LOOP INDEX FOR THE BLOCKING SORT                   DUMP
C     JBL     BLANK LINE INDEX FOR THE BLOCKING SORT                    DUMP
C     JFR     USED TO RECORD THE INDEX OF A FREE (BLANK) LINE IN THE    DUMP
C             BOTTOM PART OF THE PARTICLE ARRAY DURING THE CONSOLIDATIONDUMP
C             OF N BLANKS INTO THE TOP OF THE ARRAY                     DUMP
C     JT      TOP LOOP INDEX FOR THE BLOCKING SORT                      DUMP
C     JTEST   A TEMPORARY STORAGE THAT EVENTUALLY CONTAINS THE NUMBER   DUMP
C             OF PARTICLE DESCRIPTIONS IN THE CLASS TO BE DUMPED        DUMP
C     JTEST1  A TEMPORARY STORAGE WHICH EVENTUALLY CONTAINS THE NUMBER  DUMP
C             INDICATING THE KIND (CLASS) OF PARTICLE DESCRIPTION TO BE DUMP
C             DUMPED                                                    DUMP
C     N       THE NUMBER OF PARTICLES IN THE DATA BLOCK THAT IS WAITING DUMP
C             TO BE READ NEXT AT THE TIME WHEN DUMPP IS CALLED          DUMP
C     NALOFT  THE DIMENSIONED (MAXIMUM) SIZE OF THE PARTICLE ARRAY      DUMP
C     NBMAX   THE MAXIMUM NUMBER OF PARTICLE DESCRIPTIONS THAT CAN BE   DUMP
C             INCLUDED IN A SINGLE BLOCK AS WRITTEN ON ANY MEMORY OR    DUMP
C             INTERMEDIATE OUTPUT TAPE                                  DUMP
C     NFREE   THE NUMBER OF BLANK LINES (DENOTED BY FMAS( )=0) IN THE   DUMP
C             PARTICLE ARRAYS                                           DUMP
C     NG      A COUNT OF IN-CORE GROUNDED PARTICLES                     DUMP
C     NLOST   A COUNT OF THE PARTICLES IN THE PARTICLE ARRAY BUT LOCATEDDUMP
C             BEYOND THE COORDINATE LIMITS OF THE WIND OR TOPO DATA SETSDUMP
C     NTAP    A TEMPORARY STORAGE FOR THE NUMBER OF THE TAPE ONTO WHICH DUMP
C             THE DUMP IS TO BE MADE                                    DUMP
C     NTI     A COUNT OF IN-CORE PARTICLES THAT HAVE REACHED THE TIME   DUMP
C             BOUNDARY (ENDTIM)                                         DUMP
C     NTO     A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE TOPO DATA DUMP
C     NW      A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE WIND DATA DUMP
```

```
C       N1        AN ASSIGNED GO TO BRANCH POINT FOR THE CLASSIFYING CODE
C       N2        SEE N1
C       N3        SEE N1
C       N4        SEE N1
C       PST       SEE XPT
C       TPT       SEE XPT
C       XPT       TEMPORARY STORAGE FOR XP( ) SOMETIMES USED TO START A SORT
C       YPT       SEE XPT
C       ZPT       SEE XPT
C
C*****************************************************************************
C
      COMMON /SET1/
     1          DIAM    ,DETID(12),IRISE  , IEXEC  , ISIN   , ISOUT   ,
     2          SD      , SPAR    , SSAM  , TME    , TMP1   , TMP2    ,
     3          T2M     , U       , VPR   , W      , X      , Z       ,
     4          WHY(4 ), RMIN     , IDISTR , SPAR1 , SPAR2  , SPAR3   ,
     5          SPAR4   , SPAR5   , SPAR6 , SPAR7  , SPAR8  , SPAR9
C
C ****************************************************************************
C
      COMMON /SET2/
     1          S       , SUBSID  , GRINT  , BXLL   , BXLU   , BYLL
     2,         BYLU    , TXLL    , TXLU   , TYLL   , TYLU   , XGZ
     3,         YGZ     , NBLCK   , HTOPO  , TTOPO  , ILIM   , JLIM
     4,         KLIM    , II      , JJ     , KK     , XP     , YP
     5,         ZP      , FMAS    , TP     , PS     , VX     , VY
     6,         VZ      , IL      , JL     , IBADD  , WGRINT , NSTRAT
     7,         WLLX    , WLLY    , WURX   , WURY   , BOTHIT , IPARIN
     8,         IOTOPO  , IOWIND  , IHTOPO , IPOUT  , IPAROT , JTOP1
     9,         JWIND1  , IRROR   , TLIMIT , ENDTIM , IC     , IBYPAS
     1,         JTOPJ   , NLOST   , N3     , NTO    , NTI    , NW
     2,         NALOFT  , JTIME1  , NBMAX  , NFREE  , N      , NCL
     3,         CRMAXY  , CRUHT   , NCRTYP , BZ     , CRMINX , CRMINY
     4,         UO      , SN      , CS     , NLOCIR , DTLOC  , ATEMP
     5,         RHO     , NA      , TGZ    , DTMAC  , FROG   , CRMAXX
     6,         ROPART
      DIMENSION TOPOLM(4,4)      ,NINTAR(4)        ,ITOPLM(3,4)
      DIMENSION S(1 ,10)         ,SUBSID(400)      ,IC(18)
      DIMENSION XP(200)          ,YP(200)          ,ZP(200)         ,FMAS(200)
      DIMENSION TP(200)          ,PS(200)          ,ATEMP(260)      ,RHO(260)
      DIMENSION VX(1500)         ,VY(1500)         ,VZ(1500)        ,IL(70)
      DIMENSION JL(70)           ,IBADD(70)        ,WURX(70)
      DIMENSION WGRINT(70)                         ,WLLX(70)        ,WLLY(70)
      DIMENSION WURY(70)         ,BOTHIT(70)       ,SN(6)           ,CS(6)
      DIMENSION CRMINX(6)        ,CRMAXX(6)        ,CRMINY(6)       ,CRMAXY(6)
      DIMENSION CRUHT(6)         ,NCRTYP(6)        ,UO(6)
C
C*****************************************************************************
C
 1    FORMAT(1H1,4XI7,15H LOST PARTICLES)
 2    FORMAT(/6X,2HXP,10X2HYP,10X,2HZP,10X,2HTP,10X,2HPS,8X,4HFMAS)
 3    FORMAT(1X,6E12.5)
 4    FORMAT(10I5)
 6    FORMAT(5X11HBEYOND TOPO)
 7    FORMAT(5X13HTIME BOUNDARY)
 8    FORMAT(5X11HBEYOND WIND)
 9    FORMAT(5X8HGROUNDED)
```

```
C
C *********************************************************************
C
      DATA PROGRM/6H DUMPP/
C
C**********************************************************************
C**********************************************************************
C
C     MUST ANY PARTICLES BE DUMPED TO MAKE ROOM FOR THE INCOMING BLOCK
C     OF N PARTICLES OR TO CLEAR THE PARTICLE ARRAYS.  YES TO 151
      IF(N-NFREE)150,150,151
 150  JTEST=0
      GO TO 152
C
C 151 WOULD DUMPING THE GROUNDED PARTICLES PROVIDE SUFFICIENT ROOM FOR
C     THE BLOCK OF N INCOMING PARTICLE DESCRIPTIONS.  YES TO 1512
 151  IF(NFREE+NG-N)1511,1512,1512
C
C 1512 PREPARE TO DUMP THE GROUNDED PARTICLE DESCRIPTIONS
 1512 JTEST1=1
      JTEST=NG
      GO TO 18
C1511 DETERMINE WHICH SET OF PARTICLES TO DUMP
C     FIND THE IDENTIFIER (JTEST1) AND SIZE (JTEST) OF THE MOST
C     NUMEROUS CLASS OF PARTICLES IN THE PARTICLE ARRAYS
 1511 IF(NLOST-NG)10,11,11
 10   JTEST=NG
      JTEST1=1
      GO TO 12
 11   JTEST=NLOST
      JTEST1=2
 12   IF(JTEST-NTO)13,14,14
 13   JTEST=NTO
      JTEST1=3
 14   IF(JTEST-NTI)15,16,16
 15   JTEST1=4
      JTEST=NTI
 16   IF(JTEST-NW)17,18,18
 17   JTEST1=5
      JTEST=NW
C 18  AT THIS POINT JTEST HAS MAX(NG,NLOST,NTO,NTI,NW).
C     JTEST1 INDICATES THE KIND OF PARTICLE DESCRIPTION TO BE DUMPED
C     SEE THE FOLLOWING CODE EXPLANATION FOR JTEST1=1 THRU 5
C
C     JTEST1    NAME OF            KIND OF PARTICLE DESCRIPTIONS
C     VALUE     CLASS COUNTER      TO BE DUMPED ONTO TAPE
C      1            NG                GROUNDED PARTICLES
C      2            NLOST             PARTICLES BEYOND THE AREAS FOR WHICH
C                                     BOTH TOPO AND WINDS HAVE BEEN SPEC-
C                                     IFIED
C      3            NTO               PARTICLES BEYOND THE LIMITS OF THE
C                                     TOPO DATA CURRENTLY AVAILABLE IN CORE
C      4            NTI               PARTICLES THAT CANNOT BE VALIDLY
C                                     TRANSPORTED FURTHER UNTIL THE WIND
C                                     FIELD DESCRIPTION IS UPDATED
C      5            NW                PARTICLES BEYOND THE LIMITS OF THE
C                                     WIND DATA CURRENTLY AVAILABLE IN CORE
C
```

```
C      TEST TO SEE THAT JTEST HAS AN ACCEPTABLE VALUE.   UNACCEPTABLE TO
C      ERROR STOP AT 184
C
 18    IF(JTEST)184,184,182
 184   IRROR= 184
 7734  CALL ERROR(PROGRM,IRROR,ISOUT)
       GO TO 60
C
C 182  IS THE SIZE (JTEST) OF THE SELECTED CLASS GREATER THAN THE MAXIMUM
C      ALLOWABLE OUTPUT BLOCK SIZE (NBMAX)).  YES TO 183.
 182   IF(JTEST-NBMAX)181,181,183
C
C 183  RESET JTEST EQUAL TO THE MAXIMUM ALLOWABLE BLOCK SIZE SO THAT AN
C      ACCEPTABLE BLOCK SIZE WILL BE DUMPED.
 183   JTEST=NBMAX
C
C 181  MAKE ASSIGNMENTS FOR THE EFFICIENT CONTROL OF THE CODE THAT WILL
C      LATER BE USED TO CLASSIFY PARTICLE DESCRIPTIONS AS TO WHETHER
C      THEY ARE TO BE DUMPED, NOT TO BE DUMPED, OR MERELY BLANK.  ALSO
C      DECREASE THE APPROPRIATE CLASS COUNTER BY THE NUMBER OF DESCRIP-
C      TIONS ABOUT TO BE DUMPED (JTEST) AND MAKE AN APPROPRIATE SETTING
C      OF THE OUTPUT TAPE NAME NTAP.
C 181  GO TO THE APPROPRIATE SELECTION CODE
 181   GO TO (19,20,21,22,23),JTEST1
C
C 19   CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF GROUNDED PARTICLES.
C      GROUNDED PAR ICLES ARE IDENTIFIED BY THE PATTERN  -,-  IN THE
C      SIGNS OF FMAS( ) AND IP( ) UNDER THE CONDITION THAT IP( )+TLIMIT
C      DOES NOT EQUAL ZERO.
C      DESCRIPTIONS OF GROUNDED PARTICLES ARE ALWAYS WRITTEN ONTO THE
C      TRANSPORT INTERMEDIATE OUTPUT TAPE IPOUT
 19    NG=NG-JTEST
       NTAP=IPOUT
       ASSIGN 300 TO N1
       ASSIGN 400 TO N3
       ASSIGN 42 TO N2
       ASSIGN 42 TO N4
       GO TO 99
C
C 20   CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT ARE
C      LOST TO THE INVESTIGATION.   THESE PARTICLES ARE IDENTIFIED BY A
C      NEGATIVE FMAS( ) AND A IP( ) WHICH EQUALS TLIMIT, THE TIME WHEN
C      THE TRANSPORT OF PARTICLES IS TO CEASE.  LOST PARTICLES ARE MERELY
C      WRITTEN ONTO THE SYSTEM OUTPUT TAPE TO INFORM THE RESEARCHER OF
C      THEIR REMOVAL FROM THE TRANSPORT.
 20    NLOST=NLOST-JTEST
       NTAP=ISOUT
       ASSIGN 500 TO N1
       ASSIGN 42 TO N2
       GO TO 99
C
C 21   CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT HAVE
C      GONE BEYOND THE IN-CORE TOPOGRAPHY.  THESE PARTICLES ARE IDENTI-
C      FIED BY A POSITIVE FMAS( ) AND A NEGATIVE TP( ).  THEY ARE
C      WRITTEN ONTO THE OFF-TOPO TAPE (IOTOPO) BUT IF USE OF IOTOPO HAS
C      BEEN SUPPRESSED (BY SETTING IC(2)=1),THEY ARE WRITTEN ONTO THE
C      SYSTEM OUTPUT TAPE INSTEAD.  THIS IS TO LET THE RESEARCHER KNOW
C      THAT HIS SUPPRESSION OF IOTOPO HAS LED TO A LOSS OF PARTICLES FROM
```

```
C        THE TRANSPORT PROCESS.
 21      IF(IC(2)-1)211,212,211
 212     NTAP=ISOUT
         GO TO 213
C 211    JTOP1=1 INDICATES THAT THE ONLY OFF-TOPO PARTICLES THAT REMAIN IN
C        THE TRANSPORT ARE THOSE THAT ARE IN CORE IN THE PARTICLE ARRAYS.
C
 211     JTOP1=1
         NTAP=IOTOPO
 213     NTO=NTO-JTEST
         ASSIGN 300 TO N2
         ASSIGN 100 TO N3
         ASSIGN 42 TO N4
         ASSIGN 42 TO N1
         GO TO 99
C
C 22     CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT CAN
C        NOT BE TRANSPORTED FURTHER UNTIL THE WIND FIELD IS UPDATED.  THESE
C        PARTICLES ARE IDENTIFIED BY A POSITIVE FMAS( ) AND A TP( ) EQUAL
C        TO ENDTIM.  NORMALLY THEY ARE WRITTEN ON TAPE IPAROT,BUT WHEN THE
C        USER HAS SET IC(4)=1 TO SUPPRESS IPAROT, THEY ARE WRITTEN ON THE
C        SYSTEM OUTPUT TAPE TO NOTIFY THE USER.
 22      IF(IC(4)-1)221,222,221
 222     NTAP=ISOUT
         GO TO 223
C
C 221    JTIME1=1 INDICATES THAT THE ONLY OUT-OF-WIND PARTICLES THAT REMAIN
C        IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS.
 221     JTIME1=1
         NTAP=IPAROT
 223     NTI=NTI-JTEST
         ASSIGN 600 TO N2
         ASSIGN 42 TO N1
         GO TO 99
C
C 23     CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT ARE
C        BEYOND THE LIMITS OF THE WIND DATA CURRENTLY AVAILABLE IN CORE.
C        THESE PARTICLES ARE IDENTIFIED BY A NEGATIVE FMAS( ) AND  POSITIVE
C        TP( ).  NORMALLY THEY ARE WRITTEN ONTO TAPE IOWIND, BUT WHEN THE
C        USER HAS SET IC(3)=1 TO SUPPRESS IOWIND, THEY ARE WRITTEN ON THE
C        SYSTEM OUTPUT TAPE TO NOTIFY THE USER.
 23      IF(IC(3)-1)231,232,231
 232     NTAP=ISOUT
         GO TO 233
C
C 231    JWIND1=1 INDICATES THAT THE ONLY OUT-OF-WIND-FIELD PARTICLES THAT
C        REMAIN IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS.
 231     JWIND1=1
         NTAP=IOWIND
 233     NW=NW-JTEST
         ASSIGN 300 TO N1
         ASSIGN 100 TO N4
         ASSIGN 42 TO N2
         ASSIGN 42 TO N3
C
C 99     INITIALIZE FOR BLOCKING SORT
 99      IRSET=0
         ICON=0
```

```
      JB=NALOFT
      JT=1
      J=JB
C
C     WRITE OUT A DUMP SUMMARY
      WRITE (ISOUT,4)JTEST,JTEST1,NFREE,NG,NLOST,NTO,NTI,NW
C     NOW BEGIN THE BLOCKING SORT
C 98  CLASSIFY THE JTH PARTICLE AS BLANK, TO BE DUMPED, OR NOT TO BE
C     DUMPED
 98   IF(FMAS(J))30,31,32
 30   GO TO N1,(300,500,42)
 32   GO TO N2,(42,300,500,600)
 300  IF(TP(J))33,35,35
 33   GO TO N3,(400,100,42)
 35   GO TO N4,(42,100)
 400  IF(TP(J)+ENDTIM)100,42,100
 500  IF(TP(J)-TLIMIT) 42,100,42
 600  IF(TP(J)-ENDTIM)42,100,100
C
C 31   BLANK     NOT TO BE DUMPED
 31   IF(ICON)422,901,424
C 42   NON-BLANK    NOT TO BE DUMPED
 42   IF(ICON)421,424,424
C 100       TO BE DUMPED
 100  IF(ICON)903,9-4,900
C
C     ICON=0     FIRST PASS
C     ICON=+1     BOTTOM LOOP
C     ICON=-1    TOP LOOP
C
C 900 MOVE THE JB-TH LINE TO THE BLANK LINE (JBL)
 900  XP(JBL)=XP(JB)
      YP(JBL)=YP(JB)
      ZP(JBL)=ZP(JB)
      TP(JBL)=TP(JB)
      PS(JBL)=PS(JB)
      FMAS(JBL)=FMAS(JB)
      JT=JT+1
      FMAS(JB)=0.0
      IF(JT-JTEST)901,901,1103
 901  JBL=JB
      ICON=-1
      J=JT
 902  JB=JB-1
      GO TO 98
C 904 STORE THE JB-TH PARTICLE IN TEMPORARY STORAGE AND SET IRSET=1 TO
C     INDICATE THAT IT MUST BE PUT BACK INTO THE PARTICLE ARRAYS AT THE
C     END OF THIS DUMP OPERATION.
 904  XPT=XP(JB)
      YPT=YP(JB)
      ZPT=ZP(JB)
      TPT=TP(JB)
      PST=PS(JB)
      FMAST=FMAS(JB)
      IRSET= 1
      FMAS(JB)=0.0
      GO TO 901
 903  JT=JT+1
```

```
      J=JT
      IF(JT-JTEST)98,98,110
 424  J=J-1
      JB=JB-1
      GO TO 1104
C
C 421 MOVE THE JT-TH LINE TO THE BLANK LINE (JBL)
 421  XP(JBL)=XP(JT)
      YP(JBL)=YP(JT)
      ZP(JBL)=ZP(JT)
      TP(JBL)=TP(JT)
      PS(JBL)=PS(JT)
      FMAS(JBL)=FMAS(JT)
 422  JBL=JT
 423  ICON=1
      J=JB
 1104 IF( JB-JTEST)110,110,98
 1103 JBL = JB
C 110 IS THE TEMPORARY STORAGE LOADED.    YES TO 1101
 110  IF(IRSET)1101,1102,1101
C
C 1101 REPLACE THE TEMPORARILY STORED PARTICLE IN THE BLANK LINE (JBL)
 1101 XP(JBL) = XPT
      YP(JBL) = YPT
      ZP(JBL) = ZPT
      TP(JBL) = TPT
      PS(JBL) = PST
      FMAS(JBL)=FMAST
 1102 CONTINUE
C
C     RESET KEYS OF PARTICLES BEING DUMPED JUST BEFORE PRINTING OR
C     DUMPING THEM
      DO 131 J=1,JTEST
      IF(FMAS(J))101,111,111
 101  FMAS(J)=-FMAS(J)
 111  IF(TP(J))121,131,131
 121  TP(J)=-TP(J)
 131  CONTINUE
C
C
C
C     NOW DUMP THE SELECTED DESCRIPTIONS
C 50  IF THE SYSTEM OUTPUT TAPE IS TO BE WRITTEN, FIRST SELECT AND
C     WRITE AN APPROPRIATE TITLE.
 50   IF(NTAP-ISOUT)52,51,52
C
C     IF THE PRINTING OF LOST PARTICLE DESCRIPTIONS IS TO BE SUPRESSED,
C     GO TO 54
 51   IF(IC(8).NE.0) GO TO 54
      WRITE (ISOUT,1)JTEST
      WRITE (ISOUT,2)
      GO TO (511,516,513,514,515),JTESTI
 511  WRITE (ISOUT,9)
      GO TO 516
 513  WRITE (ISOUT,6)
      GO TO 516
 514  WRITE (ISOUT,7)
      GO TO 516
```

```
515   WRITE (ISOUT,8)
516   WRITE (ISOUT,3)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
      GO TO 54
52    WRITE (NTAP)JTEST
      IF(NTAP-IPOUT)252,155,252
155   WRITE (NTAP)(XP(J),YP(J), TP(J),PS(J),FMAS(J),J=1,JTEST)
      GO TO 54
252   WRITE (NTAP)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
      IF(IC(6)-1)54,2521,2521
2521  WRITE (ISOUT,3)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
C
C 54  ADD THE NUMBER OF LINES JUST DUMPED TO THE NUMBER OF LINES EMPTY
C     PREVIOUSLY AND THEN ZERO OUT THE IDP OF THE LINES JUST DUMPED TO
C     AVOID DOUBLE COUNTING
54    NFREE=NFREE+JTEST
      DO 541 J=1,JTEST
541   FMAS(J)=0.0
C
      IF(NFREE-N)151,152,152
C
C 152 ARE THERE NOW ENOUGH CONTIGUOUS BLANK LINES IN THE TOP OF THE
C     PARTICLE ARRAY TO RECIEVE THE N PARTICLES THAT ARE WAITING TO BE
C     READ IN.  YES TO 60
152   IF(N-JTEST)60,60,154
C
C 154 CONSOLIDATE N BLANK LINES INTO THE TOP OF THE PARTICLE ARRAY
154   JFR=NALOFT+1
      K=JTEST+1
      DO 56 J=K,N
      IF(FMAS(J))57,56,57
C 57  A PARTICLE MUST BE MOVED DOWN
57    JFR=JFR-1
      IF(FMAS(JFR))58,59,58
58    IF(JFR-JTEST)60,60,57
C 59  MOVE THE PARTICLE
59    XP(JFR)=XP(J)
      YP(JFR)=YP(J)
      ZP(JFR)=ZP(J)
      TP(JFR)=TP(J)
      PS(JFR)=PS(J)
      FMAS(JFR)=FMAS(J)
      FMAS(J)=0.0
56    CONTINUE
60    RETURN
      END
                                                                  454*

                                                                  454 *
```

```
$IBFTC RDTOP    LIST,DECK,M94/2                                          RDTO
      SUBROUTINE RDTOPO (LB)                                             RDTO
C     11 OCT 66                                                          RDTO
C     T. W. SCHWENKE   TECHNICAL OPERATIONS RESEARCH    SR  OUTPRO CHAINRDTO
C     THIS SUBROUTINE MERELY READS ONE TOPO BLOCK INTO ARRAYS S AND      RDTO
C     SUBSID. IT EXPECTS READ LIMITS TO BE IN COMMON WORDS II,JJ,KK.     RDTO
C     ERROR EXIT IF BAD LIMITS                                           RDTO
C                                                                        RDTO
C ***********************************************************************RDTO
C                                                                        RDTO
      COMMON /SET1/                                                      RDTO
     1           DIAM    , DETID  , IRISE  , IEXEC  , ISIN   , ISOUT   , RDTO
     2           SD      , SPAR   , SSAM   , TME    , TMP1   , TMP2    , RDTO
     3           T2M     , C      , VPR    , W      , X      , Z       , RDTO
     4           WHY     , RMIN   , IDISTR , SPAR1  , SPAR2  , SPAR3   , RDTO
     5           SPAR4   , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9     RDTO
      DIMENSION  DETID(12),WHY(40)                                       RDTO
C                                                                        RDTO
C ***********************************************************************RDTO
C                                                                        RDTO
      COMMON /SET2/                                                      RDTO
     1           S       , SUBSID , GRINT  , BXLL   , BXLU   , BYLL      RDTO
     2,          BYLU    , TXLL   , TXLU   , TYLL   , TYLU   , XGZ       RDTO
     3,          YGZ     , NBLCK  , HTOPO  , TTOPO  , ILIM   , JLIM      RDTO
     4,          KLIM    , II     , JJ     , KK     , XP     , YP        RDTO
     5,          ZP      , FMAS   , TP     , PS     , VX     , VY        RDTO
     6,          VZ      , IL     , JL     , IBADD  , WGRINT , NSTRAT    RDTO
     7,          WLLX    , WLLY   , WORX   , WORY   , BOTHIT , IPARIN    RDTO
     8,          IOTOPO  , IOWIND , IHTOPO , IPOUT  , IPAROT , JTOP1     RDTO
     9,          JWIND1  , IRROR  , TLIMIT , ENDTIM , IC     , IBYPAS    RDTO
     1,          JTOPO   , NLOST  , NG     , NTO    , NTI    , NW        RDTO
     2,          NALOFT  , JTIME1 , NBMAX  , NFREE  , N      , NCL       RDTO
     3,          CRMAXY  , CROHT  , NCRTYP , BZ     , CRMINX , CRMINY    RDTO
     4,          UU      , SN     , CS     , NLOCIR , DTLOC  , ATEMP     RDTO
     5,          RHO     , NA     , TGZ    , DTMAC  , FROG   , CRMAXX    RDTO
     6,          ROPART                                                  RDTO
      DIMENSION TOPOLM(4,4)      ,NINTAR(4)        ,ITOPLM(3,4)          RDTO
      DIMENSION S(1 ,10)         ,SUBSID(400)      ,IC(10)               RDTO
      DIMENSION XP(200)          ,YP(200)          ,ZP(200)      ,FMAS(200)  RDTO
      DIMENSION TP(200)          ,PS(200)          ,ATEMP(260)   ,RHO(260)   RDTO
      DIMENSION VX(1500)         ,VY(1500)         ,VZ(1500)     ,IL(70)     RDTO
      DIMENSION JL(70)           ,IBADD(70)        ,WORX(70)                 RDTO
      DIMENSION WGRINT(70)                         ,WLLX(70)     ,WLLY(70)   RDTO
      DIMENSION WORY(70)         ,BOTHIT(70)       ,SN(6)        ,CS(6)      RDTO
      DIMENSION CRMINX(6)        ,CRMAXX(6)        ,CRMINY(6)    ,CRMAXY(6)  RDTO
      DIMENSION CROHT(6)         ,NCRTYP(6)        ,UU(6)                    RDTO
C                                                                        RDTO
C ***********************************************************************RDTO
C                                                                        RDTO
 9    FORMAT(33HOTOPO DATA TOO LARGE FOR PROGRAM.)                       RDTO
 11   FORMAT(35HO INCORRECT TOPO TABLE OF CONTENTS.)                     RDTO
 100  FORMAT(10F10.3)                                                    RDTO
C                                                                        RDTO
C                                                                        RDTO
C ***********************************************************************RDTO
C ***********************************************************************RDTO
C                                                                        RDTO
      II=ITOPLM(1,LB)                                                    RDTO
      JJ=ITOPLM(2,LB)                                                    RDTO
      KK=ITOPLM(3,LB)                                                    RDTO
      TTOPO=TOPOLM(4,LB)                                                 RDTO
```

```
      BXLL=TOPOLM(1,LB)                                                 RDTO  61
      BXLU=TOPOLM(3,LB)*FLOAT(II)+BXLL                                  RDTO  62
      BYLL=TOPOLM(2,LB)                                                 RDTO  63
      BYLU=TOPOLM(3,LB)*FLOAT(JJ)+BYLL                                  RDTO  64
      JFTOPO=LB+1                                                       RDTO  65
      IF(II)1,1,2                                                       RDTO  66
2     IF(JJ)1,1,3                                                       RDTO  67
3     IF(KK)1,4,4                                                       RDTO  68
4     IF(II-ILIM)5,5,6                                                  RDTO  69
5     IF(JJ-JLIM)7,7,6                                                  RDTO  70
7     IF(KK-KLIM)8,8,6                                                  RDTO  71
6     WRITE (ISOUT,9)                                                   RDTO  72
10    STOP                                                              RDTO  73
1     WRITE (ISOUT,11)                                                  RDTO  74
      GO TO 10                                                          RDTO  75
8     READ (IHTOPO)((S(I,J),I=1,II),J=1,JJ)                             RDTO  76
      READ (IHTOPO)(SUBSID(K),K=1,KK)                                   RDTO  77
      WRITE (ISOUT,100)(SUBSID(K),K=1,KK)                               RDTO  78
      RETURN                                                            RDTO  79
      END                                                               RDTO  80
                                                                              81*

                                                                              81 *
```

```
$IBFTC LNK5    LIST,DECK,M94/2                                          LNK5
      SUBROUTINE LINK5                                                  LNK5
C          T.W.SCHWENKE  TECHNICAL OPERATIONS RESEARCH          LINK 5  LNK5
C    15 OCTOBER 1966                                                    LNK5
C     FIRST OVERLAY LINK OF TRANSPORT PROGRAM. INITIALIZATION AND CALL  LNK5
C       OF WIND FIELD PREPARING SUBROUTINE  MKWIND                      LNK5
C                                                                       LNK5
C **********************************************************************LNK5
C                                                                       LNK5
C************       TAPE IDENTIFICATIONS AND ASSIGNMENTS      ********** LNK5
C                                                                       LNK5
C         NAME                CONTENT                                   LNK5
C                                                                       LNK5
C       IHTOPO              TOPOGRAPHIC HEIGHT DATA TAPE                LNK5
C       IOTOPO              PARTICLES ALOFT BUT BEYOND IN CORE TOPOGRAPHY LNK5
C       IOWIND              PARTICLES ALOFT BUT BEYOND IN CORE WIND FIELD LNK5
C       IPARIN              PARTICLES ALOFT AND TO BE PROCESSED         LNK5
C       IPAROT              PARTICLES ALOFT AWAITING NEXT TIME PERIOD WINDS LNK5
C       IPOUT               PROGRAM SPECIFIC OUTPUT                     LNK5
C       ISOUT               SYSTEM (BCD) OUTPUT TAPE                    LNK5
C       ISIN                SYSTEM INPUT TAPE                           LNK5
C                                                                       LNK5
C************      PROGRAM GLOSSARY     ************                    LNK5
C                                                                       LNK5
C       ATEMP(J)            ATMOSPHERIC VISCOSITY IN THE J-TH STRATUM   LNK5
C       AX,AY,AZ            OUTPUT ARGUMENTS OF LOCAL CIRCULATION SYSTEM LNK5
C                           CODES.  X,Y,AND Z COMPONENTS OF WIND AT THE LNK5
C                           POSITION OF THE J-TH PARTICLE.              LNK5
C       BLANK               BLANK LITERAL  (        )                   LNK5
C       BOTHIT(K)           ALTITIDE OF BOTTOM OF KTH WIND DATA BLOCK   LNK5
C       BZ                  LENGTH OF THE SIDE OF THE SQUARE CLOUD SUBDIVIS-LNK5
C                           IONS AT THE TIME OF THEIR DEFINITION        LNK5
C       CBREZ1              SUBROUTINE     SEA BREEZE CIRCULATION MODEL LNK5
C       CIRMIN              TIME UNTIL INTERSECTION WITH THE FIRST LOCAL LNK5
C                           WIND SYSTEM                                 LNK5
C       CIRTYP(J)           THE LOCAL CIRCULATION TYPE OF THE JTH SYSTEM LNK5
C                1       MOUNTAIN WIND 1                                LNK5
C                2       RIDGE WIND 1                                   LNK5
C                3       SEA BREEZE 1                                   LNK5
C                4       SEA BREEZE 2                                   LNK5
C                5          NOT ASSIGNED                                LNK5
C       CRID( )             CLOUD RISE IDENTIFICATION                   LNK5
C       CRMINX(J)           SMALLEST X COORDINATE OF THE JTH LOCAL SYSTEM LNK5
C       CRMAXX(J)           LARGEST  X COORDINATE OF THE JTH LOCAL SYSTEM LNK5
C       CRMINY(J)           SMALLEST Y COORDINATE OF THE JTH LOCAL SYSTEM LNK5
C       CRMAXY(J)           LARGEST  Y COORDINATE OF THE JTH LOCAL SYSTEM LNK5
C       CROHT(K)         HEIGHT OF TOP SURFACE OF THE KTH LOCAL CIRCULATION LNK5
C                        SYSTEM CELL.                                   LNK5
C       DETID(J)            INITIAL CONDITIONS (FIREBALL) IDENTIFICATION LNK5
C       DENTI               TOPOGRAPHY TAPE IDENTIFICATION LITERAL (IHTOPO) LNK5
C                           AS READ FROM TAPE                           LNK5
C       DENTT               PARTICLES ALOFT INPUT TAPE IDENTIFICATION   LNK5
C                           LITERAL AS READ FROM TAPE                   LNK5
C       DTLOC               TIME INCREMENT FOR USE IN TRANSPORT WITHIN LOCALLNK5
C                           CIRCULATION SYSTEM CELLS                    LNK5
C       DTMAC               TIME INCREMENT FOR USE IN TRANSPORT WITHIN THE  LNK5
C                           MACRO FIELD BUT BELOW MAXIMUM TOPOGRAPHIC HEIGHTLNK5
C       DTST                CANNED COPY OF PARTICLES ALOFT INPUT TAPE   LNK5
C                           IDENTIFICATION LITERAL (IPARIN)             LNK5
C       DUMPP               SUBROUTINE     SELECTS AND DUMPS ONE OR MORE LNK5
C                           SETS OF PARTICLE DESCRIPTIONS ONTO TEMPORARY,OR LNK5
```

```
C                               INTERMEDIATE OUTPUT TAPES.
C       ENDTIM                  TIME UP TO WHICH THE CURRENT WIND FIELD DESCRIPT
C                               ION IS ASSUMED TO BE VALID.
C       EPSIL                   A SMALL NUMBER
C       ERROR                   SUBROUTINE     ERROR COMMENT AND/OR STOP
C       FALRAT                  SUBROUTINE   PARTICLE SETTLING RATE COMPUTATION
C       FMAS(J)                 MASS PER UNIT AREA (MKS) ASSOCIATED WITH THE
C                               J-TH CLOUD SUBDIVISION (PARTICLE) AT THE TIME OF
C                               ITS DEFINITION
C       FROG                    A PRECOMPUTED CONSTANT INPUT FOR SUBROUTINE
C                               FALRAT
C       FV                      OUTPUT ARGUMENT OF FALRAT (MKS)  A POSITIVE
C                               QUANTITY
C       FW                      FISSION YIELD
C       GETWND                  SUBROUTINE  GETS MACRO WIND VECTORS AT PARTICLE
C                               POSITION
C       H                       OUTPUT ARGUMENT OF SUBROUTINE HEIGHT
C       HEIGHT                  SUBROUTINE   RETRIEVES TOPO HEIGHT AT X,Y
C       HOB                     HEIGHT OF BURST RELATIVE TO SURFACE HEIGHT
C       HS                      TEMPORARY STORAGE FOR STANDARD ATMOSPHERIC
C                               STRATUM BOTTOM HEIGHT
C       HTOPO                   HEIGHEST TOPO HEIGHT ANYWHERE ON THE TOPO TAPE
C       HTST                    CANNED COPY OF TOPOGRAPHY TAPE IDENTIFICATION
C                               LITERAL (IHTOPO)
C       IBADD(K)                BASE ADDRESS OF KTH WIND DATA BLOCK IN ARRAYS
C                         VX, VY, AND VZ
C       IBYPAS                  INITIALIZE BYPASS FLAG
C
C       IC(J)                   CONTROL VARIABLES   INTERPRETATIONS AS FOLLOWS
C          1                    SUPRESS TOPO TAPE USEAGE IF= 1
C          2                    SUPRESS OFF TOPO TAPE USEAGE IF = 1
C          3                    SUPRESS OUT OF WIND FIELD TAPE USEAGE IF = 1
C          4                    SUPRESS PARTICLES ALOFT SECONDARY MEMORY IF = 1
C          5                    SUPRESS ALL TAPE SECONDARY MEMORY
C          6                    TRANSPORT TRACE CONTROL.  0 = NO TRACES,
C                               1 = A PARTIAL TRACE,  2 =  A MORE COMPLETE TRACE
C          7                    PRINT WIND FIELD IF  = 1
C          8                    SUPRESS PRINT AND WRITE OF LOST PARTICLES IF = 1
C
C
C       IEXEC                   EXECUTIVE CONTROL WORD TO CONTROL BRANCHING
C                               BETWEEN CHAIN LINK SUBROUTINES
C       IF                      INDEX OF THE LAST LINE IN THE PARTICLE ARRAYS
C       II                      ITOPLM(1,J) FOR THE TOPO DATA BLOCK CURRENTLY IN
C                               CORE
C       IL(K)                   LIMITS ON INDICES OF WIND BLOCK DATA SET K
C       ILIM
C       IPAS                    ASSIGNED GO TO VARIABLE FOR USE IN DEALING WITH
C                               EXCESSIVELY SMALL PARTICLE MOVEMENTS ARISING
C                               DURING TRANSPORT
C       IR                      INDICATOR OF TYPE OF BOUNDING PLANE ENCOUNTERED
C                               DURING TRANSPORT.  1=X-BOUNDARY, 2=Y-BOUNDARY,
C                               3=Z-BOUNDARY, 4= TIME BOUNDARY, 5= LOCAL CIRC-
C                               ULATION CELL BOUNDARY.
C       IRROR                   NUMBER OF THE SOURCE STATEMENT NEAREST TO WHERE
C                               AN ERROR CONDITION WAS SENSED.
C                               THAT IS TO BE PROCESSED BY THE TRANSPORT LOOP
C       IT                      ASSIGNED GO TO VARIABLE FOR USE IN CODE SELECT-
```

```
C                                 ION REGARDING USE OF PLANAR OR PIECEWISE PLANAR LNK5 119
C                                 TOPOGRAPHIC DESCRIPTION.                        LNK5 120
C      ITOPLM(1,J)                NUMBER OF CELLS IN THE X DIRECTION OF THE       LNK5 121
C                                 REGULAR GRID SECTION OF THE J-TH TOPO DATA BLOCKLNK5 122
C      ITOPLM(2,J)                SEE ITOPLM(1,J)  BUT FOR Y-DIRECTION            LNK5 123
C      ITOPLM(3,J)                NUMBER OF ENTRIES IN THE ONE-DIMENSIONAL SUB-   LNK5 124
C                                 SIDIARY TOPOGRAPHIC DATA ARRAY FOR THE J-TH     LNK5 125
C                                 DATA BLOCK                                      LNK5 126
C      ITT                        SEE IT.                                         LNK5 127
C      JDONE                      IF 1, INDICATES THAT THE ONLY PARTICLES THAT RE-LNK5 128
C                                 MAIN TO HAVE THEIR TRANSPORT COMPLETED ARE      LNK5 129
C                                 CURRENTLY IN CORE MEMORY.  NONE ARE ON THE TIME LNK5 130
C                                 BOUNDARY TAPE.                                  LNK5 131
C      JFTOPO                     INDICATOR OF TOPO TAPE FILE POSITION            LNK5 132
C                                 THIS WORD RECORDS THE NUMBER OF THE FILE        LNK5 133
C                                 WHICH A READ COMMAND WOULD BRING IN NEXT.       LNK5 134
C      JJ                         ITOPLM(2,J) FOR THE TOPO DATA BLOCK CURRENTLY INLNK5 135
C                                 CORE                                            LNK5 136
C      JL(K)                                                                      LNK5 137
C      JLIM                                                                       LNK5 138
C      JTEST                      TEMPORARY STORAGE                               LNK5 139
C      JTEST1                                                                     LNK5 140
C                +1               THE TIME BOUNDARY TAPE IS IN USE                LNK5 141
C                 0               NO TIME BOUNDARY PARTICLES ARE ON TAPE          LNK5 142
C                -1               USED ONLY TO CAUSE AN ENTRANCE TO MAIN TRANSPORTLNK5 143
C                                 LOOP WITHOUT FIRST  READING MORE PARTICLES FROM LNK5 144
C                                 TAPE.  THUS  ALL TRANS PORTABLE PARTICLES       LNK5 145
C                                 REMAINING IN CORE ARE TRANSPORTED JUST AFTER A  LNK5 146
C                                 NEW WIND FIELD HAS BEEN COMPUTED.               LNK5 147
C      JTEST2                     TEMPORARY STORAGE                               LNK5 148
C      JTIME1                     INDICATES THE PRESENCE OF PARTICLES ON THE TIME LNK5 149
C                                 BOUNDARY TAPE                                   LNK5 150
C                                 +1 INDICATES THAT THE TIME BOUNDARY TAPE IS IN  LNK5 151
C                                 USE.   0 INDICATES THAT IT IS NOT IN USE.       LNK5 152
C                                 -1 INDICATES THAT THERE ARE TIME BOUNDARY       LNK5 153
C                                 PARTICLES IN CORE.  -1 IS USED TO CAUSE AN      LNK5 154
C                                 ENTRANCE TO THE MAIN LOOP WITHOUT FIRST         LNK5 155
C                                 READING MORE PARTICLES IN SO AS TO TRANSPORT    LNK5 156
C                                 ALL TRANSPORTABLE PARTICLES THAT REMAIN IN      LNK5 157
C                                 MEMORY JUST AFTER A NEW WIND FIELD IS COMPUTED. LNK5 158
C      JTOPJ                      INDEX OF HIGHEST WIND LAYER.  SET BY MKWIND OR  LNK5 159
C                                 LINK 6                                          LNK5 160
C      JTOP1                      FLAG FOR PARTICLES OFF THE IN CORE TOPO GRID    LNK5 161
C      JW                         USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THELNK5 162
C                                 WIND STRATUM CONTAINING A PARTICLE              LNK5 163
C      JWAD                       INDEX OF MACRO WIND CELL CONTAINING PARTICLE    LNK5 164
C      JWAD1                      TEMPORARY STORAGE FOR PRECEEDING VALUE OF JWAD  LNK5 165
C      JWIND1                     FLAG FOR PARTICLES OUT OF IN CORE WIND SPEC     LNK5 166
C      KLIM                       LIMITS ON INDICES OF TOPO ARRAYS                LNK5 167
C      KK                         ITOPLM(3,J) FOR THE TOPO DATA BLOCK CURRENTLY INLNK5 168
C                                 CORE                                            LNK5 169
C      LOTRAN                     SUBROUTINE    LOCAL SYSTEM PARTICLE TRANSPORT   LNK5 170
C                                 CODE                                            LNK5 171
C      MXTOPO                     MAX. NO. TOPO BLOCK CAPABILITY OF PROGRAM       LNK5 172
C      MPNT                       USED TO CONTROL THE AMOUNT OF TRANSPORT TRACE   LNK5 173
C                                 PRINTING IN LINK 7.  1=NO PRINTING, 2=PRINT ALL LNK5 174
C                                 TRACES.                                         LNK5 175
C      MTWND1                     SUBROUTINE    MOUNTAIN WIND CIRCULATION MODEL   LNK5 176
```

```
C     N                  NUMBER OF PARTICLE DESCRIPTIONS ABOUT TO BE READLNK5 177
C                        FROM THE CURRENT BLOCK OF THE PARTICLES INPUT   LNK5 178
C                        TAPE                                            LNK5 179
C     NA                 NUMBER OF ATMOSPHERIC LEVELS DESCRIBED IN THE   LNK5 180
C                        TABLES OF STANDARD ATMOSPHERIC DENSITY AND      LNK5 181
C                        VISCOSITY                                       LNK5 182
C     NALOFT             SIZE OF ARRAY FOR PARTICLES ALOFT               LNK5 183
C     NBLCK              NUMBER OF BLOCKS OF TOPO DATA ON TOPO TAPE      LNK5 184
C     NBMAX              SIZE LIMIT ON DUMP BLOCK                        LNK5 185
C     NCIR               TEMPORARY STORAGE FOR NCRTYP(J)                 LNK5 186
C     NCL                SPARE                                           LNK5 187
C     NCRTYP(J)          LOCAL CIRCULATION TYPE IDENTIFIER OF THE J-TH   LNK5 188
C                        LOCAL SYSTEM.  1 IS MTWND1, 2 IS RGWND1, 3 IS   LNK5 189
C                        CBREZ1, OTHERS ARE UNASSIGNED.                  LNK5 190
C     NFREE              NUMBER OF EMPTY LINES IN PARTICLES ALOFT LIST   LNK5 191
C     NG                 NUMBER OF PARTICLES GROUNDED         (IN CORE)  LNK5 192
C     NINTAR(J)          NUMBER OF PARTICLES OVER TOPO AREA J AND IN     LNK5 193
C                        THE OUT-OF-TOPO BUFFER                          LNK5 194
C     NLOCIR             THE NUMBER OF LOCAL CIRCULATION SYSTEMS IN USE  LNK5 195
C     NLOST              NUMBER OF PARTICLES THAT DRIFTED BEYOND LIMITS  LNK5 196
C     NPS                NUMBER OF PARTICLE SIZE RANGES DESCRIBED IN THE LNK5 197
C                        PARTICLE SIZE DISTRIBUTION ARRAYS               LNK5 198
C     NSP                SPARE                                           LNK5 199
C     NSTRAT             MAXIMUM NUMBER OF ATMOSPHERIC STRATUM DESCRIPT- LNK5 200
C                        IONS THAT THE PROGRAM CAN ACCOMODATE            LNK5 201
C     NTI                NUMBER OF PARTICLES AT TIME BARRIER. (IN CORE)  LNK5 202
C     NTO                NUMBER OF PARTICLES AT TOPO BARRIER  (IN CORE)  LNK5 203
C     NUL                INTEGER ZERO                                    LNK5 204
C     NW                 NUMBER OF PARTICLES AT WIND BARRIER  (IN CORE)  LNK5 205
C     N1,N2              ASSIGNED GO TO VARIABLES FOR USE IN SPEEDING UP LNK5 206
C                        THE COMPUTATION OF TIME OF FLIGHT TO LOCAL CIRC-LNK5 207
C                        ULATION SYSTEM CELL BOUNDARIES.                 LNK5 208
C     POUT               CANNED COPY OF GROUNDED PARTICLES TAPE IDENT-   LNK5 209
C                        IFIER LITERAL (IPOUT )                          LNK5 210
C     PROGRM             CONTAINS PROGRAM NAME IN BCD                    LNK5 211
C     PS(J)              MID-RANGE PARTICLE SIZE (MICRONS) OF THE J-TH   LNK5 212
C                        PARTICLE SIZE RANGE DESCRIBED IN THE PARTICLE   LNK5 213
C                        SIZE DISTRIBUTION ARRAYS                        LNK5 214
C     PSEID              PARTICLE SET EXPANSION  PROGRAM OF INTERFACE    LNK5 215
C                        PROGRAM RUN IDENTIFICATION                      LNK5 216
C     PSIZE              TEMPORARY STORAGE FOR PARTICLE SIZE OF CURRENT  LNK5 217
C                        PARTICLE                                        LNK5 218
C     RADMAX             MAXIMUM CLOUD RADIUS DURING CLOUD RISE          LNK5 219
C     RDTOPO             SUBROUTINE  READS IN A TOPO DATA BLOCK          LNK5 220
C     RGWND1             SUBROUTINE    RIDGE WIND CIRCULATION MODEL      LNK5 221
C     RHO(J)             ATMOSPHERIC DENSITY IN THE J-TH TABULATED       LNK5 222
C                        STRATUM                                         LNK5 223
C     ROPART             FALLOUT PARTICLE DENSITY (MKS)                  LNK5 224
C     RTST               TEMPORARY STORAGE                               LNK5 225
C     SIGMA              STANDARD DEVIATION OF PARTICLE SIZE DISTRIBUTIONLNK5 226
C     SLDTMP             PARTICLE SOLIDIFACATION TEMPERATURE (K)         LNK5 227
C                                                                        LNK5 228
C     SSAM               MASS OF CONDENSED PHASE MATERIAL AT SPECIFICAT- LNK5 229
C                        ION TIME                                        LNK5 230
C     TC                 TEMPORARY STORAGE                               LNK5 231
C     TGZ                TIME OF DETONATION                              LNK5 232
C     TID( )             TRANSPORT IDENTIFICATION                        LNK5 233
C     TIX,TIY,TIZ        TIMES OF FLIGHT TO THE FIRST X,Y,AND Z MACRO    LNK5 234
```

```
C                          CELL BOUNDING PLANES                                 LNK5 235
C      TLIMIT              TIME SPECIFIED FOR THE TERMINATION OF TRANSPORT      LNK5 236
C      TMAXX,TMAXY         TEMPORARY STORAGE FOR LARGEST OF (TX1,TX2) AND       LNK5 237
C                          (TY1,TY2)                                            LNK5 238
C      TMINX,TMINY         TEMPORARY STORAGE FOR SMALLEST OF (TX1,TX2) AND      LNK5 239
C                          (TY1,TY2) RESPECTIVELY                               LNK5 240
C      TMSD                TIME OF SOLIDIFICATION                               LNK5 241
C      TOPID(J)            TOPOGRAPHY IDENTIFIER                                LNK5 242
C      TOPOLM              FLOATING POINT TOPOGRAPHIC DATA ARRAY                LNK5 243
C      TOPOLM(1,J)         MIN. X-COORDINATE FOR J-TH TOPO DATA BLOCK           LNK5 244
C      TOPOLM(2,J)         MIN. Y-COORDINATE FOR J-TH TOPO DATA BLOCK           LNK5 245
C      TOPOLM(3,J)         INITIAL (REGULAR) GRID INTERVAL FOR J-TH TOPO        LNK5 246
C                          DATA BLOCK                                           LNK5 247
C      TOPOLM(4,J)         HEIGHEST TOPO HEIGHT IN THE J-TH TOPO DATA BLOCKLNK5 248
C      TP                  PARTICLE TIME STATUS                                 LNK5 249
C      TS                  TIME OF FLIGHT TO POINT OF ENTRANCE TO CURRENT       LNK5 250
C                          LOCAL WIND CELL                                      LNK5 251
C      TTOPO               HEIGHEST TOPO ELEVATION IN THE IN CORE DATA          LNK5 252
C      TW                  TOTAL YIELD                                          LNK5 253
C      TWND                TIME OF FLIGHT TO TIME BOUNDARY                      LNK5 254
C      TXLL,TXLU,TYLL,     LOWER AND UPPER X AND Y COORDINATE LIMITS FOR        LNK5 255
C      TX1,TX2             TIMES OF FLIGHT TO THE TWO X-PLANES BOUNDING THELNK5 256
C                          CURRENT LOCAL CIRCULATION CELL.                      LNK5 257
C      TYLU                THE AREA ACCOUNTED FOR ON THE TOPOGRAPHY TAPE        LNK5 258
C      TY1,TY2             SEE TX1 BUT FOR Y-PLANES                             LNK5 259
C      VPX,VPY             TEMPORARY STORAGE FOR X AND Y PARTICLE VELOCITY LNK5 260
C      VPZ                 NET VERTICAL VELOCITY OF PARTICLE                    LNK5 261
C      VPZT                TEMPORARY STORAGE FOR PRECEEDING VALUE OF VPZ        LNK5 262
C      VX, VY, VZ          WIND VELOCITY COMPONENTS                             LNK5 263
C      WGRINT(K)           GRID INTERVAL OF KTH WIND DATA BLOCK                 LNK5 264
C      WID(J)              WIND FIELD DESCRIPTION IDENTIFICATION                LNK5 265
C      WLLX(K)             HORIZONTAL LIMITS OF KTH WIND DATA BLOCK             LNK5 266
C      WURY(K)             HORIZONTAL LIMITS OF KTH WIND DATA BLOCK             LNK5 267
C      X,Y                 TEMPORARY STORAGE FOR PARTICLE COORDINATES           LNK5 268
C      XBL,XBU             LOWER AND UPPER X COORDINATES OF MACRO CELL CON-LNK5 269
C                          TAINING PARTICLE                                     LNK5 270
C      XGZ,YGZ             COORDINATES OF GROUND ZERO                           LNK5 271
C      XIN,YIN,ZIN         X,Y,AND Z INCREMENTIAL DISTANCES FOR USE IN          LNK5 272
C                          CONSTANT TIME INTERVAL STEPPING BELOW MAXIMUM        LNK5 273
C                          TOPO HEIGHT.                                         LNK5 274
C      XP, YP, ZP          PARTICLE POSITION COORDINATES                        LNK5 275
C      XX,YY,ZZ            TEMPORARY STORAGE FOR X,Y,AND Z PARTICLE COORDINLNK5 276
C                          ATES                                                 LNK5 277
C      YBL,YBU             SEE XBL.   FOR Y COORDINATE                          LNK5 278
C                                                                               LNK5 279
C ***********************************************************************LNK5 280
      COMMON /SET1/                                                             LNK5 281
     1          DIAM    ,DETID(12),IRISE   , IEXEC   , ISIN    , ISOUT     ,    LNK5 282
     2          SD      , SPAR    , SSAM    , TME     , TMP1    , TMP2      ,    LNK5 283
     3          TZM     , U       , VPR     , W       , X       , Z         ,    LNK5 284
     4          WHY(40), RMIN    , IDISTR  , SPAR1   , SPAR2   , JDONE     ,    LNK5 285
     5          SPAR4   , SPAR5   , SPAR6   , SPAR7   , SPAR8   , SPAR9         LNK5 286
C                                                                               LNK5 287
C ***********************************************************************LNK5 288
C                                                                               LNK5 289
      COMMON /SET2/                                                             LNK5 290
     1          S       , SUBSID , GRINT   , BXLL    , BXLU    , BYLL           LNK5 291
     2,         BYLU    , TXLL    , TXLU    , TYLL    , TYLU    , XGZ            LNK5 292
```

```
     3,         YGZ     , NBLCK   , HTOPO   , TTOPO   , ILIM    , JLIM
     4,         KLIM    , II      , JJ      , KK      , XP      , YP
     5,         ZP      , FMAS    , TP      , PS      , VX      , VY
     6,         VZ      , IL      , JL      , IBADD   , WGRINT  , NSTRAT
     7,         WLLX    , WLLY    , WJRX    , WJRY    , BOTHIT  , IPARIN
     8,         IOTOPO  , IOWIND  , IHTOPO  , IPOUT   , IPAROT  , JTOP1
     9,         JWIND1  , IRROR   , TLIMIT  , ENDTIM  , IC      , IBYPAS
     1,         JTOPJ   , NLOST   , NS      , NIO     , NTI     , NW
     2,         NALOFT  , JTIME1  , NBMAX   , NFREE   , N       , NCL
     3,         CRMAXY  , CROHT   , NCRTYP  , BZ      , CRMINX  , CRMINY
     4,         UU      , SN      , CS      , NLOCIR  , DTLOC   , ATEMP
     5,         RHO     , NA      , TGZ     , DTMAC   , FROG    , CRMAXX
     6,         ROPART
      DIMENSION TOPOLM(4,4)     ,NINTAR(4)       ,ITOPLM(3,4)
      DIMENSION S(10,10)        ,SUBSID(400)     ,IC(18)
      DIMENSION XP(200)         ,YP(200)         ,ZP(200)         ,FMAS(200)
      DIMENSION TP(200)         ,PS(200)         ,ATEMP(260)      ,RHO(260)
      DIMENSION VX(1500)        ,VY(1500)        ,VZ(1500)        ,IL(70)
      DIMENSION JL(70)          ,IBADD(70)       ,WJRX(70)
      DIMENSION WGRINT(70)                       ,WLLX(70)        ,WLLY(70)
      DIMENSION WJRY(70)        ,BOTHIT(70)      ,SN(6)           ,CS(6)
      DIMENSION CRMINX(6)       ,CRMAXX(6)       ,CRMINY(6)       ,CRMAXY(6)
      DIMENSION CROHT(6)        ,NCRTYP(6)       ,UU(6)
C
C ***************************** ***************************************
C
      DIMENSION CRID(12)        ,PSEID(12)       ,TID(12)
      DIMENSION WID(12)         ,TOPID(12)
    1 FORMAT(12A6)
    2 FORMAT(///25X,56H****  INITIAL CONDITIONS (FIREBALL) IDENTIFICATIO
     1N  ****/25X,12A6,//25X,37H****  CLOUD RISE IDENTIFICATION  ****/25
     2X,12A6,//25X,49H****  PARTICLE SET EXPANSION IDENTIFICATION  ****
     3      /25X,12A6//25X,83H****  THIS RUN OF THE TRANSPORT MODULE WAS
     4GIVEN THE FOLLOWING IDENTIFICATION  ****/25X,12A6///25X,28H****  O
     5THER INPUT DATA  ****)
    3 FORMAT(18X,12A6)
    4 FORMAT(I5)
    5 FORMAT(2E12.5)
    6 FORMAT(1H1,24X,48HATMOSPHERIC PROPERTIES FOR FALL RATE CALCULATION
     1//25X16HHEIGHT OF BOTTOM4X,9HVISCOSITY12X,7HDENSITY/26X10HOF STRAT
     2UM/25X16HMETERS ABOVE MSL6X5H(MKS)15X5H(MKS)//)
    7 FORMAT(/15X71HTHE CONTROL VARIABLE ARRAY, IC(J), HAS BEEN GIVEN TH
     1E FOLLOWING VALUES.)
    8 FORMAT(15X,18I4)
    9 FORMAT(/15X28HTHE TRANSPORT TIME LIMIT IS F12.3)
   10 FORMAT(18X63H IN THIS RUN WE ASSUME A PLANAR DEPOSITION SURFACE AT
     1 ELEVATIONF10.3)
   11 FORMAT(42H0PARTICLES REMAINING ON TIME BOUNDARY TAPE)
   12 FORMAT(6(1X,E13.6))
   13 FORMAT(A6,4F10.3,I3)
   14 FORMAT(29H0 WRONG TAPE REEL ON DRIVE    I2)
   15 FORMAT(42H0PLEASE MOUNT CORRECT TAPE AND PRESS START)
   16 FORMAT(/18X,35HIDENTIFICATION FROM TOPOGRAPHY TAPE18X12A6)
   17 FORMAT(25X,E13.5,5X,E13.5,6X,E13.5)
   18 FORMAT(58H0TRANSPORT IS COMPLETED. INTERMEDIATE RESULTS ARE ON TAP
     1E I2)
   19 FORMAT(44H0PLEASE FILE PROTECT THE REEL ON TAPE DRIVE I2,25H  AT T
     1HE END OF THIS RUN.)
```

```
 20   FØRMAT(6F10.5)
 21   FØRMAT(18I4)
 22   FØRMAT(A6,I3,4E12.5,I5)
 23   FØRMAT(1H1///51X19H* * * * * * * * * *//12X101HT H E   D E P A R T
     1 M E N T   Ø F   D E F E N S E   F A L L Ø U T   P R E D I C T I Ø
     2 N   S Y S T E M,//51X,19H* * * * * * * * * *////52X,16HTRANSPØRT
     3MØDULE///55X,11HPREPARED BY/43X,34HTECHNICAL ØPERATIØNS RESEARCH,I
     4NC./52X,17HBURLINGTØN, MASS.////29X,63H****  SUMMARY ØF INPUT IDEN
     5TIFIERS AND INITIAL CØNDITIØNS  ****)

 24   FØRMAT(//15X16HTØPØGRAPHIC DATA)
 27   FØRMAT(//15X13HPARTICLE DATA/18X28HDENSITY ØF FALLØUT PARTICLESF20
     1.3,2X,7HKG/M**3)
 28   FØRMAT(//15X9HWIND DATA/)
 29   FØRMAT(18X,A6,1X,I6,4(1X,E13.5),1X,I10)
 30   FØRMAT(18X,A6,4(1X,F13.5),I6)
 31   FØRMAT(6F12.3)
C
C ***********************************************************************
C
      DATA HTST,DTST,BLANK,PØUT,ENDWFD/6HIHTØPØ,6HIPARIN,6H      ,6HIPØU
     1T ,6HEND ØF/
C
C ***********************************************************************
C ***********************************************************************
C
C     THIS BYPASSES INITIALIZATIØN CØDING AFTER THE FIRST PASS
      NUL=0
      IF (IBYPAS-918273)201,200,201
 201  IBYPAS=918273
C     INITIALIZE
      JDØNE=0
      IPARIN=11
      IØTØPØ=4
      IØWIND=3
      IHTØPØ=10
      IPØUT= 9
      IPARØT=1
      JTØP1=0
      JWIND1=0
      JTIME1=0
      ENDTIM=0.0
      JFTØPØ=1
      MXTØPØ=4
      DTMAC=10.
      DTLØC=10.
      NALØFT=200
      NBMAX=150
      NFREE=NALØFT
      NLØST=0
      NSTRAT =70
      NW=0
      NTØ=0
      NG=0
C
C     ILIM,JLIM,KLIM,ARE LIMITS ØN TØPØ ARRAYS. SEE DIMENSIØN.
      ILIM=10
      JLIM=10
      KLIM=400
```

```
      DO 2011 J=1,NALOFT
 2011 FMAS(J)=0.0
C
C     READ IDENTIFICATION FOR TRANSPORT
      READ (ISIN,1)(TID(J),J=1,12)
C
C     READ CONTROL DATA FOR TRANSPORT
C     THESE CONTROL PARAMETERS ARE FOR USE AS SIMPLIFYING SWITCHES
      READ (ISIN,21)(IC(J),J=1,18)
      READ (ISIN,31)TLIMIT
C
C     REWIND ALL TAPES INVOLVED IN TRANSPORT
      IF(IC(1)-1)150,151,151
 150  REWIND IHTOPO
 151  IF(IC(2)-1)152,153,153
 152  REWIND IOTOPO
 153  IF(IC(3)-1)154,155,155
 154  REWIND IOWIND
 155  IF(IC(4)-1)156,157,157
 156  REWIND IPAROT
 157  CONTINUE
      REWIND IPARIN
      REWIND IPOUT
C
C     CHECK IDENTIFICATIONS ON TOPO AND PARTICLE INPUT TAPES
 206  IF(IC(1)-1)158,203,203
 158  READ (IHTOPO)DENTI
      RTST=AND(DENTI,COMPL(HTST))
      IF(RTST)202,2031,202
C
C 202  WRONG TAPE AS IHTOPO
 202  PRINT 14,IHTOPO
      WRITE (ISOUT,14)IHTOPO
      PRINT 15
      REWIND IHTOPO
      PAUSE
      REWIND IHTOPO
      GO TO 206
C
C 204  WRONG TAPE AS IPARIN
 204  PRINT 14,IPARIN
      WRITE (ISOUT,14)IPARIN
      PRINT 15
      REWIND IPARIN
      PAUSE
      REWIND IPARIN
      GO TO 207
 2031 READ(IHTOPO)TXLL,TXLU,TYLL,TYLU,NBLCK
C
 203  CONTINUE
 207  READ (IPARIN)DENTT
      RTST=AND(DENTT,COMPL(DTST))
      IF(RTST)204,208,204
C
C 208 READ ARBITRARY 72 CHARACTER FIREBALL,CLOUD-RISE,AND PARTICLE
C     ACTIVITY IDENTIFICATIONS FROM IPARIN
  208 READ (IPARIN) FW,SSAM,SLDTMP,TMSD,SIGMA,TW,HOB,NSP,XGZ,YGZ,TGZ,BZ,
     1 NCL,RADMAX
```

```
      READ (IPARIN)(PSEID(J),J=1,12)
      READ (IPARIN)(CRID(J),J=1,12)
      READ (IPARIN)(DETID(J),J=1,12)
C
C     READ DENSITY OF FALLOUT PARTICLES
C     ROPART IS PARTICLE DENSITY IN KILOGRAMS PER CUBIC METER
      READ (IPARIN)ROPART
C
C     READ PARTICLE SIZE MASS AND ACTIVITY DISTRIBUTIONS
      READ (IPARIN)NPS
C
C     VX(   ) IS USED TO TEMPORARILY STORE THE SURFACE TO VOLUME RATIO
C     ARRAY SV
C     VY(   ) IS USED TO TEMPORARILY STORE THE A ARRAY FROM PSE (LINK4)
C     VZ(   ) IS USED TO TEMPORARILY STORE THE PACT ARRAY FROM PSE(LINK4)
      READ (IPARIN)(PS(I),VY(I),VZ(I),VX(I),I=1,NPS)
C
C     READ ATMOSPHERIC DENSITY AND VISCOSITY
C     A TABLE OF ATMOSPHERIC VISCOSITY (ATEMP(J)) AND DENSITY (RHO(J))
C     STATED IN THE MKS SYSTEM FOR 200 METER STRATA STARTING FROM 1100
C     METERS BELOW MSL
      READ (IPARIN)NA
      READ (IPARIN)(ATEMP(J),RHO(J),J=1,NA)
C
C     COMPUTE CONSTANT FOR FALL RATE CALCULATIONS
      FROG=1.3066667E-17*ROPART
C
C     READ ARBITRARY TOPO IDENTIFICATION
      IF(IC(1)-1)159,160,160
  160 READ (ISIN,20)TTOPO
      GO TO 205
  159 READ (IHTOPO)(TOPID(J),J=1,12)
C
C     READ TOPO TABLE OF CONTENTS
      READ (IHTOPO)TOPOLM
      READ (IHTOPO)ITOPLM
C
C     FIND HIGHEST TOPO HEIGHT
      HTOPO=0.0
      DO 170 J=1,NBLCK
      IF(HTOPO-TOPOLM(4,J))171,170,170
  171 HTOPO=TOPOLM(4,J)
  170 CONTINUE
C
C     READ FIRST TOPO DATA BLOCK
      CALL RDTOPO (1)
C
C 205  PUT AN IDENTIFICATION ON THE TRANSPORT INTERMEDIATE OUTPUT TAPE
  205 READ (ISIN,1)(WID(J),J=1,12)
      WRITE (IPOUT)POUT
      WRITE(IPOUT ) FW,SSAM,SLDIMP,IMOD,SIGMA,TW,HOB,NCL,TLIMIT,BZ,
     1 ROPART,XGZ,YGZ,ICZ,RADMAX
      WRITE (IPOUT) (DETID(J),J=1,12),(CRID(J),J=1,12),(PSEID(J),J=1,12)
     1,(TID(J),J=1,12),(WID(J),J=1,12)
      WRITE (IPOUT)NPS
      WRITE (IPOUT)(PS(J),VY(J),VZ(J),VX(J),J=1,NPS)
      IF(IC(1)-1)2054,2055,2054
 2055 CONTINUE
```

```
      WRITE (IPOUT) (BLANK,J=1,12)
      GO TO 2056
 2054 WRITE (IPOUT)(TOPID(J),J=1,12)
C
C     PRINT TRANSPORT OUTPUT HEADING
 2056 WRITE (ISOUT,23)
      WRITE (ISOUT,2) (DETID(J),J=1,12),(CRID(J),J=1,12),(PSEID(J),J=1,1
     12),(TID(J),J=1,12)
      WRITE (ISOUT,7)
      WRITE (ISOUT,8)(IC(J),J=1,18)
      WRITE (ISOUT,9)TLIMIT
      WRITE (ISOUT,27)KOPART
      WRITE (ISOUT,29)DENTI,NSP,XGZ,YGZ,TGZ,BZ,NCL
      WRITE (ISOUT,24)
      IF(IC(1)-1)2051,2052,2051
 2052 WRITE (ISOUT,10)TTOPO
      GO TO 2053
 2051 WRITE (ISOUT,16)(TOPID(J),J=1,12)
      WRITE (ISOUT,30)DENTI,TXLL,TXLU,TYLL,TYLU,NBLCK
 2053 WRITE (ISOUT,28)
      WRITE (ISOUT,3)(WID(J),J=1,12)
      WRITE(ISOUT,6)
      HS=-1100.0
      DO 2057 J=1,NA
      WRITE(ISOUT,17)HS,ATEMP(J),RHO(J)
 2057 HS=HS+200.0
C
C ***********************************************************************
C
C 200 ANY MORE TIME INTERVALS TO BE DEALT WITH.     NO TO 500
  200 IF(TLIMIT-ENDTIM)500,500,400
C
C 500 MAKE FINAL TRANSPORT PROGRAM OUTPUT AND COMMENTS
C     SET N=NALOFT TO CAUSE  DUMPP  TO CLEAR OUT THE ENTIRE PARTICLE ARR
C
  500 N=NALOFT
      CALL DUMPP
C
C     ARE ANY PARTICLES ON THE TIME BOUNDARY TAPE.    YES TO 700
C********************     TEMP     ***********************************
      JTIME1=0
      IF(JTIME1)501,501,700
C
C 700  PRINT ANY PARTICLE DESCRIPTIONS THAT REMAIN ON THE TIME BOUNDARY
C     OVERFLOW TAPE, IPAROT
  700 WRITE(IPAROT)NUL
      REWIND IPAROT
      WRITE (ISOUT,11)
  702 READ (IPAROT)N
      IF(N)501,501,701
  701 READ (IPAROT)(XP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,N)
      WRITE (ISOUT,12)(XP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,N)
      GO TO 702
C
C 501  WRITE A FINAL ZERO BLOCK COUNT AND EOF ON IPOUT
  501 WRITE (IPOUT)NUL
      END FILE IPOUT
      REWIND IPOUT
```

```
      WRITE (ISOUT,18)IPOUT
      PRINT 18,IPOUT
      PRINT 19,IPOUT
C
C 5010  SKIP OVER ANY UNUSED WIND DATA
C       A CARD CONTAINING 'END OF WIND FIELD DATA' MUST MARK THE END OF
C       THE WIND FIELD DATA DECK
 5010 READ(ISIN,1)RTST
      RTST=AND(ENDWFD,COMPL(RTST))
      IF(RTST)5010,800,5010
C
C 800 PREPARE TO CALL OUTPUT PROCESSOR PROGRAM
 800  IEXEC=2
      RETURN
C
C 400 GET OR OTHERWISE PRODUCE THE NEXT TIME INTERVAL'S WIND FIELD.
 400  NTI=0
      IEXEC = 1
      RETURN
      END
```

```
$IBFTC LNK6     LIST,DECK,M94/2
      SUBROUTINE LINK6
      CALL MKWIND
      RETURN
      END
```

```
$IBFTC RDCIR    LIST,DECK,M94/2                                          RDCI
      SUBROUTINE RDCIRS                                                  RDCI
C     12 OCT 66                                                          RDCI
C     T.W.SCHWENKE       TECHNICAL OPERATIONS RESEARCH, INC.             RDCI
C                                                                        RDCI
C ***********************************************************************RDCI
C                                                                        RDCI
C     THIS PROGRAM READS LOCAL CIRCULATION SYSTEM INPUTS.  IT READS      RDCI
C     SYSTEM COORDINATE LIMITS, CIRMINX( ),CRMAXX( ),CRMINY( ),CRMAXY(   RDCI
C     THE INDEX OF THE APPLICABLE COMPUTATION CODE FOR EACH LOCAL        RDCI
C     SYSTEM IS STORED IN NCRTYP( )                                      RDCI
C     A COUNT OF THE NUMBER OF LOCAL SYSTEMS IS RECORDED IN NLOCIR       RDCI
C                                                                        RDCI
C ***********************************************************************RDCI
C                                                                        RDCI
      COMMON /SET1/                                                      RDCI
     1          DIAM    , DETID  , IRISE  , IEXEC  , ISIN   , ISOUT   ,  RDCI
     2          SD      , SPAR   , SSAM   , TME    , TMP1   , TMP2    ,  RDCI
     3          T2M     , U      , VPR    , W      , X      , Z       ,  RDCI
     4          WHY     , RMIN   , IDISTR , SPAR1  , SPAR2  , SPAR3   ,  RDCI
     5          SPAR4   , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9      RDCI
      DIMENSION  DETID(12),WHY(40)                                       RDCI
C                                                                        RDCI
C ***********************************************************************RDCI
C                                                                        RDCI
      COMMON /SET2/                                                      RDCI
     1          S       , SUBSID , GRINT  , BXLL   , BXLU   , BYLL       RDCI
     2,         BYLU    , TXLL   , TXLU   , TYLL   , TYLU   , XGZ        RDCI
     3,         YGZ     , NBLCK  , HTOPO  , TTOPO  , ILIM   , JLIM       RDCI
     4,         KLIM    , II     , JJ     , KK     , XP     , YP         RDCI
     5,         ZP      , FMAS   , TP     , PS     , VX     , VY         RDCI
     6,         VZ      , IL     , JL     , IBADD  , WGRINT , NSTRAT     RDCI
     7,         WLLX    , WLLY   , WURX   , WURY   , BOTHIT , IPARIN     RDCI
     8,         IOTOPO  , IOWIND , IHTOPO , IPOUT  , IPAROT , JTOP1      RDCI
     9,         JWIND1  , IRROR  , TLIMIT , ENDTIM , IC     , IBYPAS     RDCI
     1,         JTOPJ   , NLOST  , N3     , NTO    , NTI    , NW         RDCI
     2,         NALOFT  , JTIME1 , N3MAX  , NFREE  , N      , NCL        RDCI
     3,         CRMAXY  , CROHT  , NCRTYP , BZ     , CRMINX , CRMINY     RDCI
     4,         UU      , SN     , CS     , NLOCIR , DTLOC  , ATEMP      RDCI
     5,         RHO     , NA     , TGZ    , DTMAC  , FROG   , CRMAXX     RDCI
     6,         ROPART                                                   RDCI
      DIMENSION TOPOLM(4,4)      ,NINTAR(4)       ,ITOPLM(3,4)           RDCI
      DIMENSION S(10,10)         ,SUBSID(400)     ,IC(18)                RDCI
      DIMENSION XP(200)          ,YP(200)         ,ZP(200)      ,FMAS(200) RDCI
      DIMENSION TP(200)          ,PS(200)         ,ATEMP(260)   ,RHO(260) RDCI
      DIMENSION VX(1500)         ,VY(1500)        ,VZ(1500)     ,IL(70)  RDCI
      DIMENSION JL(70)           ,IBADD(70)       ,WURX(70)              RDCI
      DIMENSION WGRINT(70)                        ,WLLX(70)     ,WLLY(70) RDCI
      DIMENSION WURY(70)         ,BOTHIT(70)      ,SN(6)        ,CS(6)   RDCI
      DIMENSION CRMINX(6)        ,CRMAXX(6)       ,CRMINY(6)    ,CRMAXY(6) RDCI
      DIMENSION CROHT(6)         ,NCRTYP(6)       ,UU(6)                 RDCI
C                                                                        RDCI
C ***********************************************************************RDCI
C                                                                        RDCI
 1    FORMAT(4E12.5,I3)                                                  RDCI
 2    FORMAT(///15X,22HLOCAL CIRCULATION CODEI4,18H IS NOT AVAILABLE.)   RDCI
C                                                                        RDCI
C ***********************************************************************RDCI
C                                                                        RDCI
      DATA PROGRM /6HRDCIRS/                                             RDCI
C                                                                        RDCI
```

```
C ***********************************************************************RDCI   61
C ***********************************************************************RDCI   62
C                                                                         RDCI   63
C       READ DEFINING DATA FOR LOCAL CIRCULATION SYSTEMS                  RDCI   64
C                                                                         RDCI   65
        K=0                                                               RDCI   66
  120   K=K+1                                                             RDCI   67
        READ (ISIN,1) CRMINX(K),CRMAXX(K),CRMINY(K),CRMAXY(K), NCRTYP(K)  RDCI   68
        NCIR=NCRTYP(K)                                                    RDCI   69
        IF(NCIR)122,100,125                                               RDCI   70
  122   IRROR=122                                                         RDCI   71
        GO TO 7734                                                        RDCI   72
  125   IF(NCIR-5)120,120,124                                             RDCI   73
  124   IRROR=124                                                         RDCI   74
        WRITE (ISOUT,2)NCIR                                               RDCI   75
 7734   CALL ERROR(PROGRM,IRROR,ISOUT)                                    RDCI   76
C 100   THIS IS THE NORMAL EXIT                                           RDCI   77
C       NLOCIR      THE NUMBER OF LOCAL CIRCULATION SYSTEMS DEFINED FOR USERDCI   78
C                   IN TRANSPORTING PARTICLES                             RDCI   79
        RETURN                                                            RDCI   81
  100   NLOCIR = K-1                                                      RDCI   80
        END                                                               RDCI   82
                                                                                      83*

                                                                                      83 *
```

```
$IBFTC MKWIN2  LIST,DECK,M94/2
      SUBRØUTINE MKWIND
C     28 NØVEMBER 1966
C     T. W. SCHWENKE   TECHNICAL ØPERATIØNS RESEARCH    SR  MKWIND  MK2
C
C ***********************************************************************
C
C     THIS SUBRØUTINE FØRMS A HØRIZØNTALLY AND VERTICALLY VARIANT WIND
C     DESCRIPTIØN IN CØRE ØN THE BASIS ØF INPUTS FRØM THE SYSTEM INPUT
C     TAPE.  INPUTS ARE AS FØLLØWS..
C       1  CØNTRØL VARIABLES ENDTIM ,    WHICH GIVES THE TIME AT WHICH
C        THE FØLLØWING DATA CEASE TØ BE VALID , ALPHA ,     WHICH IS A
C        WEIGHTING FACTØR TØ BE APPLIED TØ VERTICAL DISTANCES, BETA,
C        WHICH IS A WEIGHTING FACTØR TØ BE APPLIED TØ HØRIZØNTAL
C        DISTANCES, AND NN ,   WHICH SPECIFIES THE NUMBER ØF NEAREST
C        VECTØRS TØ BE USED IN ESTIMATING THE WIND VECTØR AT A GRID
C        PØINT.
C       2  WIND GRID SPECIFICATIØNS IN THE FØRM BØTHIT(J),WGRINT(J),
C        WLLX(J),WLLY(J),WURX(J),WURY(J)        (6F10.3)
C        WHERE BØTHIT(J) IS THE HEIGHT ØF THE BØTTØM ØF THE J-TH ARRAY,
C        WGRINT(J) IS THE GRID INTERVAL TØ BE USED IN THE J-TH LAYER,
C        AND WLLX(J), WLLY(J),WURX(J),WURY(J) ARE LØWER LEFT CØRNER AND
C        UPPER RIGHT CØRNER LIMIT CØØRDINATES.
C       3  WIND VECTØR DATA IN THE FØRM (BØTHIT(J),ZS(J),XS(J),YS(J),
C         SX(J),SY(J),SZ(J)
C        (6F12.3) WHERE  ZS(J) IS THE HEIGHT ØF THE J-TH VECTØR,
C        +SX(J) IS THE EASTWARD PØINTING CØMPØNENT ØF THE J-TH VECTØR,
C        +SY(J) IS THE NØRTHWARD PØINTING CØMPØNENT ØF THE J-TH VECTØR,
C        +SZ(J) IS THE UPWARD CØMPØNENT ØF THE J-TH VECTØR, XS(J) IS THE
C        EAST-WEST CØØRDINATE ØF THE J-TH VECTØR, AND YS(J) IS THE
C        NØRTH-SØUTH CØØRDINATE ØF THE J-TH VECTØR.
C        THE  LAYER READING ØPERATIØN IS TERMINATED WHEN BØTHIT(J) =
C        999999. ØR MØRE IS ENCØUNTERED.  THE VECTØR READING ØPERATIØN
C        IS TERMINATED WHEN ZS(J)=999999.0 ØR MØRE IS ENCØUNTERED.
C     A WIND FIELD TAPE IS- NØT- WRITTEN BY THIS PRØGRAM.
C
C************************** GLØSSARY*************************************
C
C       ALPHA        A WEIGHTING FACTØR FØR THE VERTICAL DISTANCES
C       BETA         A WEIGHTING FACTØR FØR THE HØRIZØNTAL DISTANCES
C       BIG          AN ARBITRARILY LARGE NUMBER
C       DN           DISTANCE BETWEEN THE CURRENT GRID PØINT AND THE MØST
C                    REMØTE ØF THE NEAREST NN DATA PØINTS
C       D12(J)       SEE D22(J).     FØR Y-DIRECTIØN BUT UNWEIGHTED
C       D22(J)       SQUARE ØF WEIGHTED Z DISTANCE BETWEEN GRID PØINT AND
C                    THE J-TH DATA VECTØR
C       SML          AN ARBITRARILY SMALL NUMBER
C       ISIN         SYSTEM INPUT TAPE NUMBER
C       ISØUT        SYSTEM ØUTPUT TAPE NUMBER
C       JH           FINAL (HIGHER) X INDEX .     SEE JLL
C       JJM          NUMBER ØF X RØWS IN ØUTPUT GRID
C       JLL          INITIAL (LØWER) X-INDEX FØR PRINTING PLANE ØF THE
C                    WIND FIELD ARRAY
C       JSTRS        THE NUMBER ØF WIND STRATA IN THE DESIRED WIND FIELD
C                    DESCRIPTIØN
C       JSTRV        THE TØTAL NUMBER ØF WIND DATA PØINTS BEING USED
C       K            USED BY MKWIND AS A STRATUM INDEX  AT PRINTING TIME
C       KK           Y-DIRECTIØN INDEX AT PRINTING TIME
C       NAD(J)       INDICES ØF DISTANCES BETWEEN THE CURRENT GRID PØINT
C                    AND THE JTH DATA PØINT
```

```
C        NADT        INDEX ØF THE NAD THAT CØNTAINS THE ADDRESS ØF THE D2
C                    WHICH IS THE LARGEST ØF NEAREST NN DATA PØINTS
C        NCØDE       IDENTIFICATIØN NUMBER FØR THE METHØD ØF CØMPUTATIØN TØ
C                    BE USED IN TRANSLATING THE WIND DATA INTØ THE WIND
C                    ARRAYS
C        NN          THE NUMBER ØF NEAREST DATA VECTØRS THAT THE USER WISHES
C                    TØ BE USED IN CØMPUTATIØNS
C        TI          TEMPØRARY STØRAGE
C        WG2         HALF ØF GRID INTERVAL WØRINT(JW)
C        XG          X CØØRDINATE ØF GRID PØINT
C        XLIM        AN X-DIRECTIØN LIMIT FØR TESTING FØR THE CØMPLETIØN ØF
C                    A RØW IN THE WIND FIELD ARRAY
C        YG          Y CØØRDINATE ØF GRID PØINT
C        YLIM        SEE XLIM.   FØR THE Y-DIRECTIØN
C        ZG          Z CØØRDINATE ØF GRID PØINT
C
C*****************************************************************************
C
      CØMMØN /SET1/
     1         DIAM    , DETID   , IRIØE   , IEXEC   , ISIN    , ISØUT    ,
     2         SD      , SPAR    , SSAM    , TME     , TMP1    , TMP2     ,
     3         T2M     , U       , VPR     , W       , X       , Z        ,
     4         WHY     , RMIN    , IDISTR  , SPAR1   , SPAR2   , SPAR3    ,
     5         SPAR4   , SPAR5   , SPAR6   , SPAR7   , SPAR8   , SPAR9
      DIMENSIØN  DETID(12),WHY(40)
      CØMMØN /SET2/
     1         S       , SUBSID  , GRINT   , BXLL    , BXLØ    , BYLL
     2,        BYLØ    , TALL    , TXLØ    , TYLL    , TYLØ    , AØL
     3,        YGZ     , NØLCK   , HTØP2   , TTØPØ   , ILIM    , JLIM
     4,        XLIM    , II      , JJ      , KK      , XP      , YP
     5,        ZP      , FMAS    , TP      , PS      , VX      , VY
     6,        VZ      , IL      , JL      , IBADD   , WØRINT  , NØTRAT
     7,        WLLX    , WLLY    , WØRX    , WØRY    , BØTHIT  , IPARIN
     8,        IØTØP2  , IØWIND  , IHTØPØ  , IPØUT   , IPARØT  , JTØP1
     9,        JWIND1  , IRRØR   , TLIMIT  , ENDTIM  , IC      , ISYPAS
     1,        JTØPJ   , NLØST   , NG      , NT2     , NT1     , NS
     2,        NALØFT  , JTIME1  , NBMAX   , NFREE   , N       , NCL
     3,        CRMAXY  , CRØHT   , NCRTYP  , SZ      , CRMINX  , CRMINY
     4,        ØZ      , SN      , CS      , NLØCIR  , DTLØC   , ATEMP
     5,        RHØ     , NA      , TGZ     , DTMAC   , FRØG    , CRMAXX
     6,        RØPART
      DIMENSIØN TØPØLM(4,4)   ,NINTAR(4)      ,ITØPLM(3,4)
      DIMENSIØN S(10,10)      ,SUBSID(400)    ,IC(18)
      DIMENSIØN XP(200)       ,YP(200)        ,ZP(200)        ,FMAS(200)
      DIMENSIØN TP(200)       ,PS(200)        ,ATEMP(260)     ,RHØ(260)
      DIMENSIØN VX(1500)      ,VY(1500)       ,VZ(1500)       ,IL(70)
      DIMENSIØN JL(70)        ,IBADD(70)      ,WØRX(70)
      DIMENSIØN WGRINT(70)                    ,WLLX(70)       ,WLLY(70)
      DIMENSIØN WØRY(70)      ,BØTHIT(70)     ,SN(6)          ,CS(6)
      DIMENSIØN CRMINX(6)     ,CRMAXX(6)      ,CRMINY(6)      ,CRMAXY(6)
      DIMENSIØN CRUHT(6)      ,NCRTYP(6)      ,UØ(6)
C     PARAMETERS PECULIAR TØ MKWIND MK2
      DIMENSIØN XS(300),YS(300),SX(300),SY(300),SZ(300),ZS(300),
     1DZ2(300),DY2(300),D2(300),NAD(300)
C
C ****************************************************************************
C
 1    FØRMAT(6F10.3)
 2    FØRMAT(///78HØTHE FØLLØWING WIND VECTØRS HAVE BEEN USED TØ DEFINE A
     1N ATMØSPHERE UP TØ TIME F10.0)
```

```
3     FØRMAT(/7X6HVECTØR7X22HHØRIZØNTAL CØØRDINATES8X17HVECTØR CØMPØNENT
     1S/7X6HHEIGHT10X2HXS10X2HYS10X2HVX10X2HVY10X2HVZ)
4     FØRMAT(4X,6F12.3)
5     FØRMAT(/62H  INADEQUATE CØNTRØL DATA.  CØMPUTATIØN METHØD 1 WILL B
     1E USED.)
6     FØRMAT(64H ENCØUNTERED TWØ WIND GRID LAYER REQUESTS FØR THE SAME A
     1LTITUDE.)
8     FØRMAT(5I4)
9     FØRMAT(19HCCØMPUTATIØN METHØD,I4,17H IS NØT AVAILABLE)
10    FØRMAT(/10X26HREQUESTED GRID ARRANGEMENT/7X6HHEIGHT9X8HINTERVAL22X
     16HLIMITS/33X4HWLLX8X4HWLLY8X4HWURX8X4HWURY/(4X6F12.3))
11    FØRMAT(16H1WIND CØMPØNENTS)
12    FØRMAT(1X,13HEAST-WEST RØW,I6)
13    FØRMAT(1X,10F12.5)
14    FØRMAT(19HCCØMPUTATIØN METHØDI6,17H WAS USED ØN THE I6,21H NEAREST
     1 DATA PØINTS.)
15    FØRMAT(//6H LEVELI3,6X,8HBASE AT ,F12.3,7H METERS)
16    FØRMAT(/19H NN WAS REDUCED TØ 15)
17    FØRMAT(6F12.3)
22    FØRMAT(6E20.8)
23    FØRMAT(  / 119H AN EXCESSIVE NUMBER ØF SIGNIFICANT FIGURES ARE LØS
     1T IN THE LEAST SQUARES CALCULATIØN.  THE DATA PØINTS APPRØACH A LI
     2NE/63H ØR A PLANE.  THE WEIGHTED VECTØR METHØD IS USED FØR GRID PØ
     3INT, 5X, 9H(X,Y,Z)=(, F12.3,1H,,F12.3,1H,,F12.3,1H))
24    FØRMAT(// 76H NØ VECTØRS LIE WITHIN THE SPECIFIED WEIGHTING REGIØN
     1.  A RANDØM SELECTIØN ØF ,I4, 30H VECTØRS ARE EQUALLY WEIGHTED ,
     2/ 5X, 15H FØR GRID PØINT,
     3       5X, 9H(X,Y,Z)=(, F12.3,1H,,F12.3,1H,,F12.3,1H))
25    FØRMAT ( // 10X,8HALPHA = F14.3, 7HMETERS,10X 7HDELTA = F14.3,
     1  7HMETERS.  )
C
C ***************************************************************************
C
      DATA PRØGRM,BIG,NWIND,NWTST,GIB /6HMKWIND,1.0E+30,1500,0,1.0E-30/
C
C ***************************************************************************
C ***************************************************************************
C
      READ (ISIN,1)ENDTIM,ALPHA,BETA
      ALPHA2=ALPHA*ALPHA
      BETA2=BETA*BETA
C     READ SPECIFICATIØN ØF DESIRED WIND ARRAY PRØPERTIES
      READ (ISIN,8)NN,NCØDE
      IF(NN)204,204,2041
204   IRRØR=204
7734  CALL ERRØR(PRØGRM,IRRØR,ISØUT)
2041  DØ 104 J=1,NSTRAT
      READ (ISIN,1) BØTHIT(J), AGRINT(J),WLLX(J),WLLY(J),WURX(J),WURY(J)
      IF(BØTHIT(J)-999999.0)104,105,105
104   CØNTINUE
      IRRØR=104
      CALL ERRØR(PRØGRM,IRRØR,ISØUT)
1041  READ(ISIN,1)XST
      IF(XST-999999.0)1041,105,105
105   JTØPJ=J-1
C
C     NØW SØRT
1054  KS=0
      DØ 1051 J=2,JTØPJ
      IF(BØTHIT(J)-BØTHIT(J-1))1153,1052,1051
```

```
 1153 KS=1
      HTST=BØTHIT(J-1)
      VXT=WGRINT(J-1)
      VYT=WLLX(J-1)
      VZT=WLLY(J-1)
      XST=WURX(J-1)
      YST=WURY(J-1)
      BØTHIT(J-1)=BØTHIT(J)
      WGRINT(J-1)=WGRINT(J)
      WLLX(J-1)=WLLX(J)
      WLLY(J-1)=WLLY(J)
      WURX(J-1)=WURX(J)
      WURY(J-1)=WURY(J)
      BØTHIT(J)=HTST
      WGRINT(J)=VXT
      WLLX(J)=VYT
      WLLY(J)=VZT
      WURX(J)=XST
      WURY(J)=YST
 1051 CØNTINUE
      IF(KS)1054,1055,1054
 1052 WRITE (ISØUT,6)
      IRRØR=1052
      GØ TØ 7734
C
C 1055   SØRT ØF THE REQUESTED LAYERS IS CØMPLETE
C     NØW MAKE SURE THAT THERE IS SUFFICIENT SPACE FØR THE WIND FIELD
 1055 DØ 1056 J=1,JTØPJ
      K1=(WURX(J)-WLLX(J))/WGRINT(J)+1.0
      K2=(WURY(J)-WLLY(J))/WGRINT(J)+1.0
 1056 NWTST=NWTST+K1*K2
C
C     IS AVAILABLE WIND MEMØRY EXCEEDED
      IF(NWIND .GT. NWTST) GØ TØ 1057
 1058 IRRØR=-1058
      GØ TØ 7734
 1057 DØ 100 J=1,500
      READ (ISIN,17)ZS(J),XS(J),YS(J),SX(J),SY(J),SZ(J)
      IF(ZS(J)-999999.0)100,101,101
 101  JTØPV=J-1
      IF(NN-JTØPV)106,106,2051
 2051 NN=JTØPV
      WRITE (ISØUT,16)JTØPV
      GØ TØ 106
 100  CØNTINUE
      IRRØR=100
      GØ TØ 7734
C
C                    VECTØR DATA ARE IN WIND ARRAYS ØN INDICES J=1,JTØPV
C     FIRST USE NCØDE AS A METHØD CØNTRØL VARIABLE.  BRANCH ØN NCØDE VIA
C     A CØMPUTED GØ TØ TØ THE DESIRED CØMPUTATIØN METHØD CØDE.
 106  NN1=NN+1
      IF(NCØDE)110,110,112
 112  IF(NCØDE-6)113,113,110
C
C 110  NCØDE IS INCØRRECT
 110  WRITE (ISØUT,5)
      NCØDE=1
 113  GØ TØ (115,116,117,118,119,120),NCØDE
C
```

```
C 115 METHØD 1 USES THE NN NEAREST DATA PØINTS.  METHØDS 2,3 AND 4 ALSØ
C     USE THIS CØDE BUT FØR METHØD 2, NN=1 AND FØR METHØD 3, NN=JTØPV.
C     FØR METHØD 4, THE NN SPECIFIED BY THE USER IS USED IN THE
C     LEAST SQUARES METHØD(NN MUST BE GREATER THAN THREE).
 115  IBADD(1)=1
      K=0
      JW=1
C
C     NØW FILL IN THE WIND GRID SIZE WØRDS
 1151 IL(JW)=((WØRX(JW)-WLLX(JW)))/WGRINT(JW)   +.9999999
      JL(JW)=((WØRY(JW)-WLLY(JW)))/WGRINT(JW) +.9999999
C
C     NØW INITIALIZE FØR FILLING IN THE WIND GRID
      WG2=WGRINT(JW)/2.0
      LX=1
      LY=1
      XG=WLLX(JW)+WG2
      YG=WLLY(JW)+WG2
      IF(JW-JTØPJ)1154,1155,1159
 1159 IRRØR=1159
      GØ TØ 7734
 1155 IF(JW-1)1156,1156,1157
 1156 ZG=BØTHIT(JW)
      GØ TØ 1158
 1157 ZG=ZG+BØTHIT(JW)-BØTHIT(JW-1)
      GØ TØ 1158
 1154 ZG=(BØTHIT(JW)+BØTHIT(JW+1))/2.0
C
C     SET ALL NAD(J) EQUAL TØ J TØ PRØVIDE INDICES FØR THE FULL SET ØF
C     DATA PØINTS AND TØ PRØVIDE AN INITIAL SET ØF -NEAREST- DATA PØINTS
C     SET NADT=1 TØ BEGIN THE SØRT PRØCEDURE THAT SELECTS THE MØST
C     REMØTE ØF THE SET ØF -NEAREST- DATA PØINTS.  NØTE THAT FØR THE 1ST
C     PASS ALL THE NN -NEAREST- PØINTS ARE EQUALLY LIKELY TØ BE THE MØST
C     REMØTE ØF THE SET.
 1158 DØ 203 J=1,JTØPV
 203  NAD(J)=J
      NADT=1
C
C     CØMPUTE DISTANCES BETWEEN THE CURRENT GRID PØINT (XG,YG,ZG) AND
C     EACH ØF THE DATA VECTØR LØCATIØNS
C
C     CØMPUTE SQUARED Z DELTAS
      DØ 199 J=1,JTØPV
      T1=(ZS(J)-ZG)
      T1=T1*T1
      T2=(ALPHA2-T1)
      IF(T2.LT.0.0)  T2=0.0
 199  DZ2(J)=T2/(ALPHA2+T1)
C
C     CØMPUTE SQUARED Y DELTAS
 200  DØ 201 J=1,JTØPV
      T1=YS(J)-YG
 201  DY2(J)=T1*T1
C
C     CØMPUTE SQUARED DISTANCES
 2011 DØ 202 J=1,JTØPV
      T1=(XS(J)-XG)*(XS(J)-XG)+DY2(J)
      T2=((BETA2-T1)/(BETA2+T1))*DZ2(J)
      IF(T2-GID) 2021,2021,2022
 2021 D2(J)=GID
```

```
      GØ TØ 202
 2022 D2(J)=1.0/T2
 202  CØNTINUE
C
C     FIND THE ADDRESS ØF AND DISTANCE TØ THE MØST REMØTE PØINT ØF THE
C     NN -NEAREST- PØINTS (THE PØINTS WHØSE ADDRESSES ARE GIVEN BY
C     NAD(1),NAD(NN).) STØRE THAT MAXIMUM DISTANCE IN THE WØRD DM AND
C     SET NADT SUCH THAT DM=D2(NAD(NADT)).
      KL=NAD(NADT)
      DM=D2(KL)
      DØ 207 J=1,NN
      KL=NAD(J)
      IF(DM-D2(KL))208,207,207
 208  DM=D2(KL)
      NADT=J
 207  CØNTINUE
C     AT THIS PØINT, DM IS THE LARGEST D2(J) FØR J=NAD(1),NAD(NN)
C
      IF (NN1-JTØPV)2072,2072,2073
C
C2072 NØW SELECT BEST NN PØINTS
C     SCAN THE SET D2(J),J=NAD(NN+1,JTØPV) UNTIL A D2(J) LESS THAN DM
C     IS FØUND.  IF ØNE IS FØUND, SWITCH NAD(NADT) WITH THE SELECTED NAD
C     THEN RESET DM AND NADT TØ INDICATE THE MØST REMØTE ØF THE NEAREST
C     NN PØINTS.  WHEN THE FULL SET D2(J),J=NAD(NN+1,JTØPV) HAS BEEN
C     SCANNED, THE SET ØF NEAREST DATA PØINTS HAS BEEN SELECTED.  ØNLY
C     ØNE SCAN IS REQUIRED.
 2072 DØ 210 J=NN1,JTØPV
      KL=NAD(J)
      IF(DM-D2(KL))210,210,211
 211  NTEMP=NAD(J)
      NAD(J)=NAD(NADT)
      NAD(NADT)=NTEMP
C
C     NØW RESET DM AND NADT TØ THE NEW MØST REMØTE PØINT
      DM=D2(KL)
      DØ 212 KKK=1,NN
      KL=NAD(KKK)
      IF(DM-D2(KL))213,212,212
 213  DM=D2(KL)
      NADT=KKK
C
C     DM AND NADT ARE SET WITH THE PARAMETERS ØF THE MØST REMØTE ØF
C     THE NEAREST NN PØINTS
 212  CØNTINUE
 210  CØNTINUE
 2073 CØNTINUE
C
C     THE NEAREST NN HAVE BEEN FØUND
C     *********SØME DAY INSERT HERE A BRANCH TØ A WEIGHTING METHØD HERE**
C
C     INCREMENT INDEX FØR STØRING VECTØR CØMPUTED FØR PØINT (XG,YG,ZG)
      K=K+1
C
C     IS THE LEAST SQUARES METHØD TØ BE USED.  YES TØ 2081
      IF(NCØDE-4)2080,2081,2080
C
C2081 THIS IS THE LEAST SQUARES METHØD
C     INITIALIZE FØR LEAST SQUARES METHØD
 2081 SNN=NN
```

```
      SDX=0.0
      SDY=0.0
      SDZ=0.0
      SDX2=0.0
      SDY2=0.0
      SDZ2=0.0
      SDXY=0.0
      SDXZ=0.0
      SDYZ=0.0
      SAU=0.0
      SAV=0.0
      SAW=0.0
      SUX=0.0
      SUY=0.0
      SUZ=0.0
      SVX=0.0
      SVY=0.0
      SVZ=0.0
      SWX=0.0
      SWY=0.0
      SWZ=0.0
C
C     BEGIN LOOP TO EVALUATE INTERMEDIATE STEP FOR LEAST SQUARES CALC.
      DO 3100 J=1,NN
      KL=NAD(J)
C
C      COMPUTE DISTANCE BETWEEN KL-TH DATA POINT AND CURRENT GRID POINT
      T1=XS(KL)-XG
      TY=YS(KL)-YG
      TZ=ZS(KL)-ZG
C
C     COMPUTE ELEMENTS OF LEAST SQUARES MATRIX, B
      SDX=SDX+T1
      SDY=SDY+TY
      SDZ=SDZ+TZ
      SDX2=SDX2+T1*T1
      SDY2=SDY2+DY2(KL)
      SDZ2=SDZ2+DZ2(KL)
      SDXY=SDXY+T1*TY
      SDXZ=SDXZ+T1*TZ
      SDYZ=SDYZ+TY*TZ
      SAU=SAU+SX(KL)
      SAV=SAV+SY(KL)
      SAW=SAW+SZ(KL)
      SUX=SUX+T1*SX(KL)
      SUY=SUY+TY*SX(KL)
      SUZ=SUZ+TZ*SX(KL)
      SVX=SVX+T1*SY(KL)
      SVY=SVY+TY*SY(KL)
      SVZ=SVZ+TZ*SY(KL)
      SWX=SWX+T1*SZ(KL)
      SWY=SWY+TY*SZ(KL)
 3100 SWZ=SWZ+TZ*SZ(KL)
      SAUG1=SDY2*SDZ2-SDYZ*SDYZ
      SAUG2=SDXY*SDZ2-SDYZ*SDXZ
      SAUG3=SDXY*SDYZ-SDY2*SDXZ
      SAUG4=SDX2*SDYZ-SDXY*SDXZ
C
C     COMPUTE COMPLEMENTARY MINORS OF MATRIX B
      B11=SDX2*SAUG1-SDXY*SAUG2+SDXZ*SAUG3
```

```
      B21=SDX*SAUG1-SDY*SAUG2+SDZ*SAUG3
      B31=SDX*SAUG2-SDY*(SDX2*SDZ2-SDXZ*SDXZ)+SDZ*SAUG4
      B41=SDX*SAUG3-SDY*SAUG4+SDZ*(SDX2*SDY2-SDXY*SDXY)
C
C       TEST TØ SEE IF A RØW ØR CØLUMN IS APPRØXIMATELY ZERØ
      BB= AMAX1(ABS(SNN*B11),ABS(SDX*B21),ABS(SDY*B31), ABS(SDZ*B41))
      IF(BB-1.0E-20)3800,3800,3700
C
C     CØMPUTE DETERMINANT ØF B
 3700 BBB=SNN*B11-SDX*B21+SDY*B31-SDZ*B41
C
C     TEST FØR LØSS ØF PRECISIØN
      IF(ABS(BBB/BB)-0.001)3800,3800,3900
C
C3800 TØØ MANY SIGNIFICANT FIGURES ARE LØST IN THE LEAST SQUARES
C     CALCULATIØN.  THE DATA PØINTS APPRØACH A PØINT, A LINE, ØR A
C     PLANE.  USE THE WEIGHTED VECTØR METHØD
 3800 WRITE (ISØUT,23)XG,YG,ZG
      GØ TØ 2080
C
C     CØMPUTE WIND VECTØRS
 3900 VX(K)=(B11*SAU-B21*SUX+B31*SUY-B41*SUZ)/BBB
      VY(K)=(B11*SAV-B21*SVX+B31*SVY-B41*SVZ)/BBB
      VZ(K)=(B11*SAW-B21*SWX+B31*SWY-B41*SWZ)/BBB
      GØ TØ 2090
C
C2080 CØMPUTE AND SUM  THE WEIGHTING FACTØRS
 2080 SUM=0.0
      DØ 214 J=1,NN
      L=NAD(J)
 2142 D2(L)=1.0/D2(L)
 214  SUM=SUM+D2(L)
      IF(SUM/FLØAT(NN) .LE. GIB) WRITE(ISØUT,24) NN,XG,YG,ZG
C
C     NØW CØMPUTE VECTØR ESTIMATE AT GRID PØINT
C     CØMPUTE STØRAGE INDEX
C     CØMPUTE AND STØRE WIND ESTIMATE AT GRID PØINT
      VX(K)=0.0
      VY(K)=0.0
      VZ(K)=0.0
      DØ 216 J=1,NN
      L=NAD(J)
      VX(K)=VX(K)+SX(L)*D2(L)
      VY(K)=VY(K)+SY(L)*D2(L)
 216  VZ(K)=VZ(K)+SZ(L)*D2(L)
      VX(K)=VX(K)/SUM
      VY(K)=VY(K)/SUM
      VZ(K)=VZ(K)/SUM
 2090 XG=XG+WGRINT(JW)
      LX=LX+1
      IF(LX-IL(JW))2011,2011,2012
 2012 XG=WLLX(JW)+WG2
      LY=LY+1
      LX=1
      YG=YG+WGRINT(JW)
      IF(LY-JL(JW))200, 200,1152
 1152 JW=JW+1
      IF(JW-JTØPJ)1160,1160,150
 1160 JT=JW-1
      IBADD(JW)=IBADD(JT)+(IL(JT))*(JL(JT))
```

```
      GO TO 1151
C
C 116  METHOD 2   NEAREST VECTOR
 116  NN=1
      GO TO 115
C
C 117  METHOD 3   ALL VECTORS WEIGHTED
 117  NN=JTOPV
      GO TO 115
 118  CONTINUE
C
C 118  METHOD 4   LEAST SQUARES
C     USE BRANCH ON NCODE=4 TO BRANCH TO LEAST SQUARES IN CODE
C     NN MUST BE GREATER THAN 3 FOR THE LEAST SQUARES METHOD
 1180 IF(NN-4)1181,115,115
 1181 IERROR=1181
      GO TO 7734
 119  CONTINUE
 120  CONTINUE
 121  WRITE (ISOUT,9)NCODE
      IERROR=121
      GO TO 7734
 130  WRITE (ISOUT,2)ENOTIN
      WRITE (ISOUT,3)
      WRITE (ISOUT,4)(ZS(J),XS(J),YS(J),SX(J),SY(J),SZ(J),J=1,JTOPV)
      WRITE (ISOUT,10)(WEIGHT(J),OWEIGHT(J),ACCX(J), ACCY(J),WORX(J),WORY
     1(J),J=1,JTOPV)
      WRITE (ISOUT,14)NCODE,NN
      WRITE (ISOUT,20) ALPHA, BETA
      IF(IO(7)-1)109,1091,109
 1091 WRITE (ISOUT,11)
      DO 107 K=1,JTOPO
      WRITE (ISOUT,15)K,SPHIT(K)
      JJM=JL(K)
      JLL=IJADD(K)
      DO 108 KK=1,JJM
      JH=JLL+IL(K)-1
      WRITE (ISOUT,12)KK
      WRITE (ISOUT,13)(VX(J),J=JLL,JH)
      WRITE (ISOUT,13)(VY(J),J=JLL,JH)
      WRITE (ISOUT,13)(VZ(J),J=JLL,JH)
      JLL=JH+1
 108  CONTINUE
 107  CONTINUE
 109  CALL RDCIRC
      RETURN
      END
```

```
$IBFTC LIN7     LIST,DECK,M94/2
      SUBROUTINE LINK7
C        T.W.SCHWENKE   TECHNICAL OPERATIONS RESEARCH          LINK 7
C     19 OCT 66
C       THIRD PART OF TRANSPORT MODULE.  PARTICLE TRANSPORT.
C
C *************************************************************************
C
C     SEE SUBROUTINE LINK5 FOR A TRANSPORT GLOSSARY
C
C *************************************************************************
C
      COMMON /SET1/
     1          DIAM    ,DETID(12),IRISE  , IEXEC  , ISIN   , ISOUT    ,
     2          SD      , SPAR   , SSAM   , TME    , TMP1   , TMP2     ,
     3          T2M     , U      , VPR    , W      , X      , Z        ,
     4          WHY(4 ), RMIN    , IDIST  , SPAR1  , SPAR2  , JDONE    ,
     5          SPAR4   , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9
C
C *************************************************************************
C
      COMMON /SET2/
     1          S       , SUBSID , GRINT  , BALL   , BALU   , BYLL
     2,         BYLU    , TXLL   , TXLU   , TYLL   , TYLU   , XGZ
     3,         YGZ     , NBLCK  , HTOPO  , ITOPO  , ILIM   , JLIM
     4,         KLIM    , II     , JJ     , KK     , XP     , YP
     5,         ZP      , FMAS   , TP     , PS     , VX     , VY
     6,         VZ      , IL     , JL     , IBADD  , WGRINT , NSTRAT
     7,         WLLX    , WLLY   , WORX   , WORY   , BOTHIT , IPARIN
     8,         IOTOPO  , IOWIND , INTOPO , IPOUT  , IPAROT , JTOP1
     9,         JWIND1  , IRROR  , TLIMIT , ENDTIM , IC     , ISYPAS
     1,         JTOPJ   , NLOST  , NG     , NTO    , NTI    , NW
     2,         NALOFT  , JTIME1 , NBMAX  , NFREE  , N      , NCL
     3,         CRMAXY  , CROHT  , NCRTYP , BZ     , CRMINX , CRMINY
     4,         OO      , SN     , CS     , NLOCIR , DILOC  , ATEMP
     5,         RHO     , NA     , TGZ    , DIMAC  , FROG   , CRMAXX
     6,         ROPART
      DIMENSION TOPOLM(4,4)     ,NINTAR(4)      ,ITOPLM(3,4)
      DIMENSION S(10,10)        ,SUBSID(400)    ,IC(18)
      DIMENSION XP(200)         ,YP(200)        ,ZP(200)        ,FMAS(200)
      DIMENSION TP(200)         ,PS(200)        ,ATEMP(260)     ,RHO(260)
      DIMENSION VX(1500)        ,VY(1500)       ,VZ(1500)       ,IL(70)
      DIMENSION JL(70)          ,IBADD(70)      ,WORX(70)
      DIMENSION WGRINT(70)                      ,WLLX(70)       ,WLLY(70)
      DIMENSION WORY(70)        ,BOTHIT(70)     ,SN(6)          ,CS(6)
      DIMENSION CRMINX(6)       ,CRMAXX(6)      ,CRMINY(6)      ,CRMAXY(6)
      DIMENSION CROHT(6)        ,NCRTYP(6)      ,OO(6)
C
C *************************************************************************
C
    1 FORMAT(3I5,6E12.5)
   10 FORMAT (I5)
   11 FORMAT(6E15.5)
   12 FORMAT(5E15.5,6I5)
C
C *************************************************************************
C
      DATA PROGRM /6HLINK7 /
C
C *************************************************************************
C *************************************************************************
```

```
C
      JW=0
      NUL=0
      EPSIL=1.0
C
C     ARE TRANSPORT TRACES TO BE WRITTEN..  YES TO 5500
      IF(IC(6)-1)5510,5510,5500
 5500 MPNT=2
      GO TO 5520
 5510 MPNT =1
C
C     BRANCH TO READ ADDITIONAL DATA
C     READ DATA PECULIAR TO EACH LOCAL WIND SYSTEM
 5520 IF(NLOCIR) 510,510,511
C
C 511 SETTING J NEGATIVE WILL CAUSE THE LOCAL CIRCULATION SYSTEM CODES
C     TO READ THE DATA THAT THEY NEED WHEN THEY ARE FIRST ENTERED
 511  J=-1
      DO 500 I=1,NLOCIR
      K=I
      NCIR=NCRTYP(I)
      GO TO (501,502,503,504,505),NCIR
 501  CALL MTWND1(J,K,AX,AY,AZ)
      GO TO 500
 502  CALL RGWND1(J,K,AX,AY,AZ)
      GO TO 500
 503  CALL CBREZ1(J,K,AX,AY,AZ)
      GO TO 500
C ****************  CODE INSERTION POINTS  ****************************
 504  CONTINUE
 505  CONTINUE
C ****************  CODE INSERTION POINTS  ****************************
 506  IRROR=-506
      GO TO 300
 500  CONTINUE
C
 510  IF(TLIMIT-ENDTIM)48,49,49
 48   ENDTIM=TLIMIT
 49   IF(IC(1)-1)51,50,51
 50   ASSIGN 188 TO IT
      ASSIGN 188 TO ITI
      GO TO 1000
 51   ASSIGN 1871 TO IT
      ASSIGN 1813 TO ITI
 1000 IF=NALOFT
      IF(JTIME1)1113,1112,1112
 1113 JTIME1=0
      GO TO 1001
C
C     ATTEMPT TO READ IN A BLOCK OF PARTICLES ALOFT RECORDS
C     FIRST READ BLOCK SIZE
 1112 READ (IPARTM)N
C
C     IF BLOCK SIZE IS NEGATIVE OR ZERO, NO BLOCK EXISTS
      IF(N)100,100,101
C
C     CHECK TO SEE IF BLOCK WILL FIT IN ARRAY
 101  IF(N-NALOFT)1 21,1021,103
```

```
C
C103  ERROR - ALOFT LIST TOO LARGE. SHOULD NEVER HAPPEN. GO TO EXIT.
 103  IRROR=103
C
C300  GENERALIZED ERROR STOP
 300  CALL ERROR(PROGRM,IRROR,ISOUT)
C
 1021 CALL DUMPP
C
C 102 NOW READ A BLOCK OF PARTICLE ALOFT DESCRIPTIONS
 102  IF=N
      READ (IPARIN)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1 ,IF)
C
C     ARE TRANSPORT TRACES TO BE WRITTEN..  YES TO 5522
      IF(IC(6))5521,5521,5522
 5522 WRITE (ISOUT,11)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,NALOFT)
 5521 NFREE=NFREE-N
C
C ***********************************************************************
C
C     BEGINNING OF PARTICLES ALOFT LIST LOOP
 1001 DO 160 J=1,IF
      ASSIGN 3052 TO IPAS
      GO TO (5540,5530),MPNT
 5530 WRITE  (ISOUT,11) XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J)
C
C     SELECT PARTICLE TO BE TRANSPORTED  --  +TP     +FMAS
 5540 IF(FMAS(J))160,160,192
 192  IF(TP(J))160,193,194
 194  IF(TP(J)-ENDTIM)195,160,1971
 1971 IF(TP(J)-TLIMIT)160,160,197
 193  IRROR=-193
      GO TO 300
 197  IRROR=-197
      GO TO 300
C
C 195  A PARTICLE HAS BEEN SELECTED
C     IS THE CURRENT PARTICLE WITHIN A LOCAL CIRCULATION SYSTEM
 195  IF(NLOCIR)1950,1950,1951
 1951 DO 1952 K=1,NLOCIR
C***********************   TEMP   ********************************
      WRITE (ISOUT,1)J,K,NLOCIR,FMAS(J),XP(J),YP(J),ZP(J), TP(J)
      IF(XP(J)-CRMINX(K))1952,1953,1953
 1953 IF(CRMAXX(K)-XP(J))1952,1954,1954
 1954 IF(YP(J)-CRMINY(K))1952,1955,1955
 1955 IF(CRMAXY(K)-YP(J))1952,1956,1956
C
C 1956 THE JTH PARTICLE IS WITHIN THE KTH LOCAL CIRCULATION SYSTEM
C     SUBROUTINE LOTRAN(J,K) TRANSPORTS THE JTH PARTICLE IN THE KTH
C     LOCAL WIND SYSTEM.
 1956 CALL LOTRAN(J,K)
C
C     188 IS WHERE GROUNDED PARTICLES ARE DEALT WITH.
      IF(ZP(J)+9000.0)188,188,1957
 1957 IF(TP(J)-ENDTIM)1951,1958,1958
 1958 NTI=NTI+1
      GO TO 160
 1952 CONTINUE
```

```
C
 1950 XX=XP(J)
      YY=YP(J)
      ZZ=ZP(J)
      PSIZE=PS(J)
 1961 CALL GETWND(XX,YY,ZZ,JWAD,JW)
      IF(JWAD)1959,1960,196
C
C 1959 THE NEEDED WIND FIELD IS NOT IN CORE
 1959 FMAS(J)=-FMAS(J)
      TP(J)=TLIMIT
      NLOST=NLOST+1
      GO TO 160
 1960 IRROR=-1960
      GO TO 300
C
C 196 GET PARTICLE FALL RATE.  PUT IT IN FV WITH SIGN POSITIVE.
 196  CALL FALRAT(ZZ,PSIZE,FV,ATEMP,RHO,FROG,ISOOT)
C
C     COMPUTE VERTICAL PARTICLE MOTION COMPONENT
C     A POSITIVE VPZ DENOTES AN UPWARD POINTING VECTOR
      VPZ=VZ(JWAD)-FV
C
C     COMPUTE TIMES TO NEXT MACRO WIND FIELD BORDERS
C     COMPUTE TIX -- TRANSIT TIME TO X BOUNDARY
      XBL=(AINT((XP(J)-WLLX(JW))/WGRINT(JW)))*WGRINT(JW) +WLLX(JW)
      XBU=XBL+WGRINT(JW)
      YBL=(AINT((YP(J)-WLLY(JW))/WGRINT(JW)))*WGRINT(JW) +WLLY(JW)
      YBU=YBL+WGRINT(JW)
      ASSIGN 917 TO N1
      IF(VX(JWAD))161,162,163
 161  TIX=(XBL-XP(J))/(VX(JWAD))
      GOTO 164
 162  TIX=100000.0
      ASSIGN 918 TO N1
      GO TO 164
 163  TIX=(XBU-XP(J))/(VX(JWAD))
C
C     COMPUTE TIY -- TRANSIT TIME TO Y BOUNDARY
 164  ASSIGN 919 TO N2
      IF(VY(JWAD))165,166,167
 165  TIY=(YBL-YP(J))/(VY(JWAD))
      GO TO 168
 166  TIY=100000.0
      ASSIGN 920 TO N2
      GO TO 168
 167  TIY=(YBU-YP(J))/(VY(JWAD))
C
C     COMPUTE TIZ -- TRANSIT TIME TO Z BOUNDARY
C 168 IS PARTICLE MOVING UP OR DOWN.   UP TO 171
 168  IF(VPZ)169,170,171
C
C 169 IS PARTICLE BELOW MAX TOPO HEIGHT.    YES TO 1691
 169  IF (ZP(J)-TTOPO)1691,1691,1693
C
C     1693 IS MAX TOPO HEIGHT ABOVE SLICE BOTTOM.  YES TO 1692
 1693 IF (TTOPO-BOTHIT(JW))1691,1691,1692
 1692 TIZ = (TTOPO-ZP(J))/VPZ
```

```
      GO TO 1711
 1691 TIZ=(BOTHIT(JW)-ZP(J))/VPZ
      GO TO 1711
 170  TIZ=1000000.0
      GO TO 1711
 171  TIZ=(BOTHIT(JW+1)-ZP(J))/VPZ
C
C1711 FIND THE EARLIEST INTERSECTION WITH A LOCAL CIRCULATION SYSTEM
 1711 CIRMIN=TLIMIT
C     ARE THERE ANY LOCAL CIRCULATION SYSTEMS.   YES TO 1712
      IF(NLOCIR)172,172,1712
C
C 1712 COMPUTE TIME OF FLIGHT TO EACH OF THE FOUR VERTICAL PLANES THAT
C     BOUND THE LJ-TH LOCAL CIRCULATION CELL.
 1712 DO 1713 LJ=1,NLOCIR
      GO TO (917,918),N1
 918  TX1=1000000.0
      TX2=1000000.0
      GO TO 921
 917  TX1=(CRMINX(LJ)-XP(J))/VX(JWAD)
      TX2=(CRMAXX(LJ)-XP(J))/VX(JWAD)
 921  GO TO (919,920),N2
 920  TY1=1000000.0
      TY2=1000000.0
      GO TO 922
 919  TY1=(CRMINY(LJ)-YP(J))/VY(JWAD)
      TY2=(CRMAXY(LJ)-YP(J))/VY(JWAD)
C
C     TEST  X INTERCEPTS
C     IS THE FIRST X DIRECTION INTERCEPT IN THE PAST.  YES TO 1714
 922  IF(TX1)1714,1716,1716
C
C 1714 IS THE SECOND X DIRECTION INTERCEPT ALSO IN THE PAST.  IF YES,
C     BOTH X DIRECTION INTERCEPTS ARE IN THE PAST AND THE PARTICLE WILL
C     NOT INTERSECT THIS CELL.  GO TO 1713 TO CONSIDER THE NEXT CELL.
 1714 IF(TX2)1713,1715,1715
 1715 TMINX=TX1
      TMAXX=TX2
      GO TO 1717
 1716 IF(TX2)1718,1719,1719
 1718 TMINX=TX2
      TMAXX=TX1
      GO TO 1717
C
C 1719 BOTH X INTERCEPTS ARE IN THE FUTURE
 1719 IF(TX1-TX2)1715,1715,1718
C
C 1717 NOW TEST FOR Y INTERCEPTS
 1717 IF(TY1)1720,1721,1721
 1720 IF(TY2)1713,1723,1723
 1723 TMINY=TY1
      TMAXY=TY2
      GO TO 1724
 1721 IF(TY2)1725,1726,1726
 1725 TMINY=TY2
      TMAXY=TY1
      GO TO 1724
C
```

```
C 1726 BOTH Y INTERCEPTS ARE IN THE FUTURE
 1726 IF(TY1-TY2)1723,1723,1725
C 1724 NOW SELECT FIRST INTERCEPT
C
C      SELECT THE SECOND PLANE PIERCE  (LAST OF FIRSTX,FIRSTY)
 1724 IF(TMINX-TMINY)1727,1728,1728
 1728 TS=TMINX
      TC=TMAXY
      GO TO 1729
 1727 TS=TMINY
      TC=TMAXX
 1729 IF(TS-TC)1730,1713,1713
C
C 1730 KEEP TIME OF EARLIEST INTERCEPT
 1730 IF(TS-CIRMIN)1731,1713,1713
 1731 CIRMIN=TS
 1713 CONTINUE
C******** * * * * * *     TEMP     * * * * * * *********
      WRITE (ISOUT,11)CIRMIN
C
C      AT THIS POINT CIRMIN CONTAINS THE TIME OF THE FIRST INTERCEPT
C      BETWEEN PARTICLE AND A LOCAL CIRCULATION SYSTEM
C 172 NOW SELECT EARLIEST BOUNDARY
 172  IF(TIX-TIY)173,175,175
 173  IF(TIX-TIZ)176,178,178
 176  TSM=TIX
      IR=1
      GO TO 179
C
C 178 SET SMALLEST TIME OF FLIGHT STORAGE AT THE TIME OF FLIGHT TO THE
C      Z BOUNDARY
 178  TSM=TIZ
      IR=3
      GO TO 179
 175  IF(TIY-TIZ)180,178,178
 180  TSM=TIY
      IR=2
C
C 179 IS INTERSECTION WITH LOCAL SYSTEM PRIOR TO EARLIEST BOUNDARY
C      INTERCEPT.     YES TO 1792
 179  IF(TSM-CIRMIN)1791,1793,1792
 1792 TSM=CIRMIN
 1793 IR=5
C
C 1791  DOES TIME LIMIT COME BEFORE EARLIEST OTHER BOUNDARY. YES TO 182
 1791 IF(TSM+TP(J)-ENDTIM)181,182,182
C
C 182  TRANSPORT PARTICLE UNTIL ENDTIM
 182  TSM=ENDTIM-TP(J)
      IR=4
      GO TO 306'
C
C 181 TEST FOR EXCESSIVELY SMALL MOVEMENT
  181 GO TO(5580,5570),MPNT
 5570  WRITE (ISOUT,11) TSM,VPZT,VPZ,TIY,FV
 5580 IF(TSM-EPSIL)3050,3050,3067
C
C      SPECIAL TRANSPORT IN THE EVENT OF EXCESSIVELY SMALL TSM
```

```
 3050 GO TO IPAS,(3052,3053)
C
 3052 ASSIGN 3053 TO IPAS
      JWAD1=JWAD
      VPZT=VPZ
      TSM=EPSIL
      GO TO 1811
 3053 IF(VPZT*VPZ)3054,3055,3055
 3054 VPZ=0.0
      TIZ=TLIMIT
      GO TO 3056
 3055 VPZ=(VPZ+VPZT)/2.0
C
C     IF VECTORS ARE OF OPPOSITE SIGNS, USE A ZERO
 3056 IF(VX(JWAD)*VX(JWAD1))3057,3058,3058
 3057 VPX=0.0
      TIX=TLIMIT
      GO TO 3059
 3058 VPX=(VX(JWAD)+VX(JWAD1))/2.0
C
 3059 IF(VY(JWAD)*VY(JWAD1))3060,3061,3061
 3060 VPY=0.0
      TIY=TLIMIT
      GO TO 3062
C
 3061 VPY=(VY(JWAD)+VY(JWAD1))/2.0
 3062 TWND=ENDTIM-TP(J)
      TSM=AMIN1(TIX,TIY,TIZ,CIRMIN,TWND)
      IF(TSM-TWND)3065,3064,3065
 3064 IR=4
      GO TO 3066
 3065 IR=5
      TSM=TSM+EPSIL
      TP(J)=TP(J)+TSM
C
 3066 XP(J)=XP(J)+VPX*TSM
      YP(J)=YP(J)+VPY*TSM
      ZP(J)=ZP(J)+VPZ*TSM
      GO TO 3063
 3067 ASSIGN 3052 TO IPAS
C
C 3051IS PARTICLE BELOW MAXIMUM TOPO HEIGHT     NO TO 1811
 3051 IF(ZP(J)-TTOPO)1812,1812,1811
 1812 GO TO ITT,(188,1813)
C
C 1813 TRANSPORT PARTICLE FOR TSM BY STEPS OF DTMAC CHECKING TOPO AS WE
C        GO
C     COMPUTE MOVEMENT INCREMENTS FOR X,Y, AND Z DIRECTIONS
 1813 XIN=DTMAC*VX(JWAD)
      YIN=DTMAC*VY(JWAD)
      ZIN=DTMAC*VPZ
 1814 XP(J)=XP(J)+XIN
      YP(J)=YP(J)+YIN
      ZP(J)=ZP(J)+ZIN
      TP(J)=TP(J)+DTMAC
      TSM=TSM-DTMAC
C
C     TEST FOR PARTICLE IMPACT ON TOPOGRAPHY
```

```
      X=XP(J)
      Y=YP(J)
      CALL HEIGHT(X,Y,H)
      IF(H+20000.0)1872,1872,1815
 1815 IF(H+10000.0)1875,1875,1816
 1816 IF(H-ZP(J))1817,188,188
 1817 IF(TSM)189,189,1814
C
C 1811 TRANSPORT PARTICLE FOR TSM
C      FIRST INCREASE TSM TO ASSURE THAT THE PARTICLE WILL ACHIEVE ITS
C      BOUNDARY
 1811 TSM=TSM*1.000 01
      XP(J)=XP(J)+VX(JWAD)*TSM
      YP(J)=YP(J)+VY(JWAD)*TSM
      ZP(J)=ZP(J)+VPZ*TSM
      TP(J)=TP(J)+TSM
 3063 CONTINUE
C
C      ARE TRANSPORT TRACES TO BE WRITTEN..  YES TO 5550
      GO TO (5560,5550),MPNT
 5550 WRITE(ISOUT,12)XP(J),YP(J),ZP(J),TP(J),TSM,NTI,NG,NTO,NW,NLOST,
     1IR
C
C      TEST FOR PARTICLE IMPACT ON TOPOGRAPHY
 5560 IF(ZP(J)-ITOPO)187,187,189
C
C 189 PARTICLE IS NOT GROUNDED.  ADJUST INDICES JW AND JWAD AND THEN
C      RECYCLE.
 189  GO TO (250,250,252,159,195),IR
 159  NTI=NTI+1
      TP(J)=ENDTIM
      GO TO 160
 250  JW=-JW
      GO TO 1950
 252  IF(VZ(JWAD))258,258,259
 258  JW=1-JW
      GO TO 1961
 259  JW=-(JW+1)
 270  GO TO 1961
 187  GO TO IT,(188,1871)
 1871 X=XP(J)
      Y=YP(J)
      CALL HEIGHT(X,Y,H)
      IF(H+20000.0)1872,1872,1874
 1874 IF(H+10000.0)1875,1875,1876
C
C 1872 H=-20000.0     PARTICLE BEYOND SPECIFIED TOPO
 1872 FMAS(J)=-FMAS(J)
      TP(J)=TLIMIT
      NLOST=NLOST+1
      GO TO 160
C
C 1875  H=-10000.0     PARTICLE BEYOND IN-CORE TOPO
 1875 TP(J)=-TP(J)
      NTO=NTO+1
      IF(JTOP1)160,1877,160
 1877 JTOP1=-1
      GO TO 160
```

```
1876 IF(H-ZP(J))189,188,188
C
C 188  TAKES CARE OF GROUNDED PARTICLES
 188  FMAS(J)=-FMAS(J)
      TP(J)=-TP(J)
      NG=NG+1
 160  CONTINUE
C
C ****************************************************************************
C
C     END OF MAIN TRANSPORT LOOP
C
C ****************************************************************************
C
      IF(JDONE-1)1000,100,1000
C
C     ARE ANY PARTICLES ON THE OFF THE TOPO TAPE
 100  IF(JTOP1)105,104,105
C     JTOP1 GR    ZERO INDICATES SOME PARTICLES ARE ON OFF TOPO TAPE
C     A NEGATIVE JTOP1 INDICATES PARTICLES IN OFF TOPO BUFFER BUT NOT
C     ON THE OFF TOPO TAPE
C
C     ARE ANY PARTICLES ON THE OUT OF WIND FIELD TAPE
 104  IF(JWIND1)130,200,130
C 200  IS PARTICLES ALOFT TIME BOUNDARY TAPE IN USE $$
 200  IF(JTIME1)203,203,201
 203  IF(NTI)2021,2031,2021
 2021 JTIME1=-1
      JDONE = 1
      GO TO 202
 2031 ENDTIM=TLIMIT
      GO TO 202
 201  WRITE(IPAROT)NUL
      REWIND IPAROT
      REWIND IPARIN
      ITEMP=IPARIN
      IPARIN =IPAROT
      IPAROT=ITEMP
      JTIME1=0
      IF(NTI)2022,202,2022
 2022 JTIME1=-1
 202  IEXEC=2
      RETURN
C
C     A NEGATIVE JWIND1 INDICATES SOME PARTICLES ARE IN THE OUT OF THE
C     WIND FIELD BUFFER BUT NOT ON TAPE
C     JWIND1 GREATER THAN ZERO INDICATES PARTICLES ARE ON THE OUT OF
C     THE WIND FIELD TAPE
C
C
C  105 GET THE REQUIRED TOPO DATA FROM TAPE
C     105 TO 107 SCANS BUFFER PARTICLE COUNTERS TO DETERMINE NEXT NEEDED
C     TOPO DATA BLOCKS AND CHOOSES THE NEAREST ONE FOR READING.
C
C     NBLCK  IS SET BY THE INITIALIZING PROGRAM WHICH READS THE TOPO
C     TAPES IDENTIFICATION RECORDS
C     NINTAR(J) IS SET BY THE TRANSPORT LOOP WHEN PARTICLES LEAVE TOPO
C     AND RESET WHEN OFF-TOPO BUFFER IS EMPTIED.
```

```
C
 105  JTEST=1000
      DO 107 J=1,NBLCK
      JTEST1=NINTAR(J)
      IF(JTEST1)107,107,108
 108  JTEST2=JFTOPO-J+1
      IF(JTEST2)109,110,111
 110  IRROR=-110
C
C     SEEKS THE FILE IN CORE. SHOULD NEVER HAPPEN. GO TO A STOP EXIT.
      GO TO 300
 111  IF(JTEST2-JTEST)112,107,107
 112  JTEST=JTEST2
      GO TO 140
 109  IF(JTEST+JTEST2)107,107,113
 113  JTEST=-JTEST2
 140  JF=J
 107  CONTINUE
C
C     AT THIS POINT JF HAS THE NUMBER OF THE DESIRED FILE
C     NOW MOVE TAPE TO SELECTED FILE
      IF(JF-JFTOPO)1072,1071,1071
C
C 1071 PREPARE TO MOVE FORWARD ON IHTOPO.  COMPUTE NUMBER OF BLOCKS TO
C     READ IN
 1071 JR=JF-JFTOPO+1
      GO TO 1074
C
C 1072 DESIRED FILE IS BEHIND READ HEAD.  BACK UP TO GET IT.
 1072 REWIND IHTOPO
C
C     NOW SKIP OVER INITIAL RECORDS
      READ (IHTOPO)IST
      READ (IHTOPO)3ST,IST,IST,IST,IST
      READ (IHTOPO)TOPOLM
      READ (IHTOPO)ITOPLM
      JR=JF
C
C 1074 NOW READ UP THROUGH THE DESIRED BLOCK
 1074 DO 1073 J=1,JR
 1073 CALL RDTOPO(J)
C
C 116 RESET ALL OFF-TOPO PARTICLES
 116  DO 118 J=1,NALOFT
      IF(FMAS(J))117,118,118
 117  IF(TP(J))119,118,118
 119  IF(TP(J)-TLIMIT)120,118,120
 120  FMAS(J)=-FMAS(J)
      TP(J)=-TP(J)
 118  CONTINUE
      IF=NALOFT
      IF(JTOP1)1151,115,115
 1151 JTOP1=0
      JTIME1=-1
      GO TO 1001
C
C 115 REWIND OFF TOPO AND PARTICLES ALOFT TAPES AND SWAP NAMES
 115  WRITE(IOTOPO)NUL
```

```
      REWIND IOTOPO
      REWIND IPARIN
      ITEMP=IOTOPO
      IOTOPO=IPARIN
      IPARIN=ITEMP
      JTOP1=0
      GO TO 1001
C
C 130  GET THE REQUIRED WIND FIELD DATA FROM TAPE
 130  GO TO 124
C    INSERT CODE HERE
C
C     RESET ALL IN-CORE OUT-OF-WIND-FIELD PARTICLE KEYS
 124  DO 122 J=1,NALOFT
      IF(FMAS(J)) 1241,122,122
 1241 IF(TP(J))122,122,1242
 1242 IF(TP(J)-TLIMIT)1243,122,1243
 1243 FMAS(J)=-FMAS(J)
 122  CONTINUE
      IF=NALOFT
      IF(JWIND1)123,125,125
 123  JWIND1=0
      JTIME1=-1
      GO TO 1001
C 125 REWIND OUT OF WIND AND PARTICLES ALOFT TAPES AND SWAP NAMES
 125  WRITE (IOWIND,10)NUL
      JWIND1=0
      REWIND IOWIND
      REWIND IPARIN
      ITEMP=IOWIND
      IOWIND=IPARIN
      IPARIN=ITEMP
      GO TO 1001
C
      END
```

```
$IBFTC GETWN    LIST,DECK,M94/2
      SUBROUTINE GETWND(XX,YY,ZZ,JWAD,JW)
C     28 NOVEMBER 1966
C     T.W. SCHWENKE     TECHNICAL OPERATIONS RESEARCH
C
C ***********************************************************************
C
C     THIS SUBROUTINE RETRIEVES THE MACRO WIND FIELD VECTOR WHICH
C     APPLIES AT THE POINT WHOSE COORDINATES ARE IN THE ARGUMENT WORDS
C     XX,YY, ANDZZ AT ENTRANCE.  THE X,Y, AND Z WIND VECTOR COMPONENTS
C     ARE IN THE COMMON VARIABLES VX(JWAD),VY(JWAD), AND VZ(JWAD) WHEN
C     THE SUBROUTINE RETURNS.  IF AN ENTRANCE IS MADE WITH ARGUMENT JW
C     SET NEGATIVE, JW IS SET POSITIVE AND ITS VALUE IS USED RATHER THAN
C     RECOMPUTED.  UPON EXIT JWAD IS SET NEGATIVE IN THE EVENT THAT THE
C     MACRO WIND FIELD PERTAINING TO THE DESIRED POINT IS NOT AVAILABLE
C     IN CORE.
C
C ***************************  GLOSSARY  ********************************
C
C     JW              INDEX OF THE WIND ARRAY STRATA IN THE MACRO WIND FIELD.
C                     THIS INDEX INCREASES FROM BOTTOM TO TOP OF THE FIELD.
C                     IF NEGATIVE AT ENTRANCE, IT IS SET POSITIVE AND USED BY
C                     GETWND.  IF ZERO OR POSITIVE AT ENTRANCE, IT IS RE-
C                     COMPUTED BY GETWND.
C     JWAD            AN INDEX FOR THE STORAGE ARRAYS (ONE DIMENSIONAL) FOR
C                     MACRO WIND FIELD VECTORS.  JWAD IS SET NEGATIVE BY
C                     GETWND IF THE NEEDED MACRO WIND FIELD VECTORS ARE NOT
C                     IN CORE.
C     JT              INDEX OF WIND LAYER USED IN A SEARCH FOR THE LAYER CON-
C                     TAINING THE J-TH PARTICLE.   TOP INDEX
C     JTOPJ           INDEX OF THE TOP LAYER IN THE MACRO WIND FIELD DESCRIP-
C                     TION
C     JTST            TEMPORARY STORAGE
C     ZZ              Z COORDINATE OF THE POINT FOR WHICH A MACRO WIND FIELD
C                     VECTOR IS SOUGHT
C     XX              X COORDINATE.   SEE ZZ
C     YY              Y COORDINATE.   SEE ZZ
C     IRROR           NUMBER OF THE STATEMENT NEAR THE POINT WHERE AN ERROR
C                     WAS DISCOVERED
C     WLLX(JW)        LOWER LIMIT FOR X COORDINATES IN THE MACRO WIND FIELD
C     WLLY(JW)        LOWER LIMIT FOR Y COORDINATES IN THE MACRO WIND FIELD
C     WURX(JW)        UPPER LIMIT FOR X COORDINATES IN THE MACRO WIND FIELD
C     WURY(JW)        UPPER LIMIT FOR Y COORDINATES IN THE MACRO WIND FIELD
C     IIW             X DIRECTION WIND FIELD RETRIEVAL INDEX
C     JJW             Y DIRECTION WIND FIELD RETRIEVAL INDEX
C     WGRINT(JW) GRID INTERVAL FOR THE WIND FIELD IN JW-TH STRATUM IN
C                     METERS
C     IL(JW)          THE NUMBER OF GRID DIVISIONS IN THE X DIRECTION OF THE
C                     WIND FIELD IN STRATUM JW
C     IBADD(JW)       BASE ADDRESS FOR STORING DATA FROM THE JW-TH STRATUM OF
C                     THE WIND FIELD.  THIS IS AN INDEX IN THE 1-D WIND ARRAY
C
C ***********************************************************************
C
      COMMON /SET1/
     1         DIAM    , DETID  , IRISE  , IEXEC  , ISIN   , ISOUT     ,
     2         SD      , SPAR   , SSAM   , TME    , TMP1   , TMP2      ,
     3         T2M     , J      , VPR    , W      , X      , Z         ,
     4         WHY     , RMIN   , IDISTR , SPAR1  , SPAR2  , SPAR3     ,
     5         SPAR4   , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9
      DIMENSION  DETID(12),WHY(40)
```

```
C
C ****************************************************************************
C
      COMMON /SET2/
     1          S       , SUBSID , GRINT  , BXLL   , BXLU   , BYLL
     2,         BYLU    , TXLL   , TXLU   , TYLL   , TYLU   , XGZ
     3,         YGZ     , NBLCK  , HTOPO  , TTOPO  , ILIM   , JLIM
     4,         KLIM    , II     , JJ     , KK     , XP     , YP
     5,         ZP      , FMAS   , TP     , PS     , VX     , VY
     6,         VZ      , IL     , JL     , IBADD  , WORINT , NOTTAP
     7,         WLLX    , WLLY   , WORX   , WORY   , BOTHIT , IPARIN
     8,         IOTOPO  , IOWIND , IHTOPO , IPOUT  , IPAROT , JTOPI
     9,         JWINDI  , IRROR  , TLIMIT , ENDTIM , IC     , ISTPRO
     1,         JTOPJ   , NLOST  , NG     , NTO    , NTI    , NN
     2,         NALOFT  , STIMEI , NOMAX  , NFREE  , N      , NCL
     3,         CRMAXY  , CROHT  , NCRTYP , BZ     , CRMINX , CRMINY
     4,         JJ      , SN     , CS     , NLOCIR , STLOC  , ATEMP
     5,         RHO     , NA     , TGZ    , DTMAC  , PROG   , CRMAXX
     6,         NOPART
      DIMENSION TOPOLM(4,4)      ,NINTAR(4)         ,ITOPLM(3,4)
      DIMENSION S(10,10)         ,SUBSID(400)       ,IC(18)
      DIMENSION XP(200)          ,YP(200)           ,ZP(200)        ,FMAS(200)
      DIMENSION TP(200)          ,PS(200)           ,ATEMP(200)     ,RHO(200)
      DIMENSION VX(1500)         ,VY(1500)          ,VZ(1500)       ,IL(10)
      DIMENSION JL(70)           ,IBADD(70)         ,WORX(70)
      DIMENSION WORINT(70)                          ,WLLX(70)       ,WLLY(70)
      DIMENSION WORY(70)         ,BOTHIT(70)        ,SN(6)          ,CS(6)
      DIMENSION CRMINX(6)        ,CRMAXX(6)         ,CRMINY(6)      ,CRMAXY(6)
      DIMENSION CROHT(6)         ,NCRTYP(6)         ,JJ(6)
C
C ****************************************************************************
C
      DATA PROG/6HGETWND/
C
C ****************************************************************************
C ****************************************************************************
C
C     TEST FOR SPECIAL ENTRANCE FROM MAIN TRANSPORT LOOP WITH JW SET -.
      IF(JW)100,102,102
C 100 SPECIAL ENTRANCE.    MAGNITUDE OF JW IS STILL VALID.
  100 JW=-JW
      GO TO 270
C     REGULAR ENTRANCE              COMPUTE JW
C     FIND INDEX OF WIND FIELD LAYER THAT CONTAINS THE POINT XX,YY,ZZ
C     AND STORE IT IN JW.   USE A TWO-BOUNDED BINARY SEARCH.
  102 JT=JTOPJ+1
      JW=1
C
C 153  COMPUTE TRIAL INDEX NUMBER
  153 JTST=(JT+JW)/2
C     HAVE TRIAL TOP AND BOTTOM INDICES CONVERGED TO INDICATE THE
C     DESIRED LAYER.    NO TO 154
      IF(JT-JW-1)155,270,154
C
C 154  IS PARTICLE ABOVE THE BOTTOM OF THE TRIAL LAYER.     NO TO 152
  154 IF(ZZ-BOTHIT(JTST))152,150,150
C 150  PARTICLE IS IN OR ABOVE SLICE JTST.
  150 JW=JTST
```

```
      GO TO 153
C 152  PARTICLE IS BELOW SLICE JTST
  152 JT=JTST
      GO TO 153
 155  IRROR=-155
 7734 CALL ERROR(PROGRM,IRROR,IOUT)
C
C
C 270  IS THE PARTICLE WITHIN THE SPECIFIED WIND FIELD.     NO TO 271
 270  IF(XX-WLLX(JW))271,272,272
C 271 MARK PARTICLE BEYOND SPECIFIED WIND FIELD
 271  JWAD=-1
      GO TO 160
 272  IF(XX-WURX(JW))273,273,271
 273  IF(YY-WLLY(JW))271,275,275
 275  IF(YY-WURY(JW))274,274,271
C
C 274 IS SPECIFIED WIND FIELD IN CORE      ASSUMED YES IN THIS PROGRAM
 274  CONTINUE
C 276      NOW COMPUTE JWAD - THE INDEX OF THE APPLICABLE WIND CELL
 276  IIW=  (XX-WLLX(JW))/WDXIN(JW)
      JJW=  (YY-WLLY(JW))/WDYIN(JW)
      JWAD=IBADD(JW)+(JJW  )*NIC(JW)+IIW
 160  RETURN
      END
```

```
$IBFTC LUTRA    LIST,DECK,M94/2
      SUBROUTINE LUTRAN(J,K)
C     11 OCT 66
C     T.H.SCHWERNE          TECHNICAL OPERATIONS RESEARCH
C
C ******************************************************************
C
C     THIS IS THE EXECUTIVE ROUTINE FOR TRANSPORTING PARTICLES THROUGH
C     ALL LOCAL WIND SYSTEMS.
C     IF THE J-TH PARTICLE IS ABOVE THE TOP OF THE LOCAL CIRCULATION
C     CELL, LUTRAN TRANSPORTS IT USING WIND VECTORS RETRIEVED FROM THE
C     MACRO WIND FIELD ARRAYS.  FOR PARTICLES LOCATED IN THE LOCAL
C     CIRCULATION CELL IT CALLS THE APPROPRIATE LOCAL CIRCULATION CODE
C     TO OBTAIN AN ESTIMATE OF THE WIND VECTOR AT PARTICLE POSITION.
C     IN EITHER CASE THE PARTICLE IS MOVED FOR ONE TIME STEP OF
C     DURATION DT SECONDS.  THIS PROCEDURE IS REPEATED UNTIL THE
C     PARTICLE LEAVES THE LOCAL CELL OR LANDS ON THE PARAMETRICALLY
C     DESCRIBED TOPOGRAPHY OR UNTIL THE TIME FOR UPDATING THE
C     ENTIRE WIND FIELD IS REACHED.  PARTICLES MAY FALL INTO THE LOCAL
C     CELL FROM ABOVE DURING TRANSPORT.
C
C ************************  GLOSSARY  ******************************
C
C     WX          X COMPONENT OF A WIND VECTOR RETRIEVED FROM A LOCAL
C                 WIND FIELD DESCRIPTION.  METERS/SEC.
C     WY          Y COMPONENT  -  SEE WX
C     WZ          Z COMPONENT  -  SEE WX
C     CRWSTX(K)   COORDINATE OF THE WESTERN BOUNDING PLANE FOR THE K-TH
C                 LOCAL CIRCULATION SYSTEM CELL
C     CRSOTH(K)   SEE CRWSTX(K) - SOUTHERN BOUNDARY
C     CRNRTH(K)   SEE CRWSTX(K) - NORTHERN BOUNDARY
C     CREAST(K)   SEE CRWSTX(K) - EASTERN BOUNDARY
C     CRTOP(K)    ALTITUDE (METERS ABOVE SEA) OF THE TOP BOUNDARY OF THE
C                 K-TH LOCAL CIRCULATION CELL
C     DTLOC       TIME STEP DURATION (SECONDS) FOR TRANSPORT WITHIN EACH
C                 LOCAL CIRCULATION CELLS
C     ENDTIM      TIME FOR THE NEXT UPDATING OF THE WIND FIELD. SECONDS
C     FV          SETTLING RATE (ABSOLUTE VALUE IN METERS/SEC.) OF THE
C                 J-TH PARTICLE AS COMPUTED BY SUBROUTINE FALRAT
C     IERROR      STATEMENT NEAR THE POINT WHERE AN ERROR WAS DISCOVERED
C     J           INDEX FOR PARTICLE DESCRIPTIONS
C     JW          INDEX OF THE WIND ARRAY STRATA IN THE MACRO WIND FIELD.
C                 THIS INDEX INCREASES FROM BOTTOM TO TOP OF THE FIELD.
C                 IF NEGATIVE AT ENTRANCE, IT IS SET POSITIVE AND USED BY
C                 GETWND.  IF ZERO OR POSITIVE AT ENTRANCE, IT IS RE-
C                 COMPUTED BY GETWND.
C     JWAD        AN INDEX FOR THE STORAGE ARRAYS (ONE DIMENSIONAL) FOR
C                 MACRO WIND FIELD VECTORS.  JWAD IS SET NEGATIVE BY
C                 GETWND IF THE NEEDED MACRO WIND FIELD VECTORS ARE NOT
C                 IN CORE.
C     JWT         TEMPORARY INDEX OF THE BOUNDING LEVEL
C     K           ARGUMENT OF LUTRAN THAT COMMUNICATES THE INDEX OF THE
C                 LOCAL SYSTEM CELL IN OR ABOVE WHICH THE J-TH PARTICLE
C                 IS LOCATED WHEN LUTRAN IS CALLED
C     NC          A BRANCH POINT VARIABLE FOR USE WITH AN ASSIGNED GO TO
C                 NC THAT EFFICIENTLY TRANSFERS TO THE APPROPRIATE LOCAL
C                 CIRCULATION SYSTEM PROGRAM DURING EXECUTION OF THE
C                 LOCAL TRANSPORT LOOP
C     NCIR        TEMPORARY NON-SUBSCRIPTED STORAGE FOR LOCAL CIRCULATION
C                 SYSTEM TYPE NUMBER
C     NCRTYP(K)   TYPE DESIGNATION FOR THE K-TH LOCAL CIRCULATION SYSTEM
```

```
C                     SEE THE EXPLANATION OF TYPE NUMBERS BELOW.             LOTR  61
C                          1              MTWND1                             LOTR  62
C                          2              RGWND1                             LOTR  63
C                          3              CBREZ1                             LOTR  64
C                          4           NOT ASSIGNED                          LOTR  65
C                          5           NOT ASSIGNED                          LOTR  66
C      TCELLT     SEE TXX.   FOR THE PLANE DEFINING LOCAL CELL TOP           LOTR  67
C      TLAYER     SEE TXX.   FOR THE HORIZONTAL PLANES BOUNDING THE MACROLOTR  68
C                 WIND FIELD LAYER THAT THE PARTICLE IS IN                   LOTR  69
C      TP(J)      TIME COORDINATE FOR THE J-TH PARTICLE                      LOTR  70
C                 SINCE DETONATION                                           LOTR  71
C      TXX        TIME OF FLIGHT TO NEXT (IN TIME) INTERCEPT BETWEEN         LOTR  72
C                 PARTICLE TRAJECTORY AND AN X=CONSTANT PLANE BOUNDING       LOTR  73
C                 THE LOCAL WIND CELL                                        LOTR  74
C      TYY        SEE TXX.   FOR A Y-BOUNDARY PLANE                          LOTR  75
C      VPZ        VERTICAL PARTICLE VELOCITY, M/SEC.                         LOTR  76
C      VX( )      X COMPONENT ARRAY OF THE MACRO WIND FIELD DESCRIPTION      LOTR  77
C      VY( )      Y COMPONENT - SEE VX( )                                    LOTR  78
C      VZ( )      Z COMPONENT - SEE VX( )                                    LOTR  79
C      XP(J)      X COORDINATE OF THE J-TH PARTICLE.  A COMMON ARRAY         LOTR  80
C      XX         X COORDINATE OF THE POINT FOR WHICH GETWND IS TO GET       LOTR  81
C                 THE APPLICABLE MACRO WIND VECTOR                           LOTR  82
C      YP(J)      Y COORDINATE OF THE J-TH PARTICLE.  A COMMON ARRAY         LOTR  83
C      YY         Y COORDINATE - SEE XX                                      LOTR  84
C      ZP(J)      Z COORDINATE OF THE J-TH PARTICLE.  A COMMON ARRAY         LOTR  85
C                 ALTITUDE OF J-TH PARTICLE IN METERS ABOVE MSL              LOTR  86
C      ZZ         Z COORDINATE - SEE XX                                      LOTR  87
C                                                                            LOTR  88
C*************************************************************************LOTR  89
C                                                                            LOTR  90
      COMMON /SET1/                                                          LOTR  91
     1        DIAM   , SETID  , TRIGD  , ICALC  , IOIN   , ISOUT   ,         LOTR  92
     2        GD     , SPAR   , SPAN   , THE    , TPPL   , TPPZ    ,         LOTR  93
     3        T24    , U      , VPR    , W      , X      , Z       ,         LOTR  94
     4        AMY    , RMIN   , IDIGIR , SPAR1  , SPAR2  , SPAR3   ,         LOTR  95
     5        SPAR4  , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9             LOTR  96
      DIMENSION  SETID(12),AMY(40)                                           LOTR  97
C                                                                            LOTR  98
C*************************************************************************LOTR  99
C                                                                            LOTR 100
      COMMON /SET2/                                                          LOTR 101
     1        W      , SUBSID , GRINT  , EXLL   , EXLO   , EYLL              LOTR 102
     2,       EYLO   , TXLL   , TXLO   , TYLL   , TYLO   , XGZ               LOTR 103
     3,       TGZ    , HMAXIN , HTOPO  , TTOPO  , ILIM   , JLIM              LOTR 104
     4,       KLIM   , II     , JJ     , KK     , XP     , YP                LOTR 105
     5,       ZP     , PMAS   , TP     , PS     , VX     , VY                LOTR 106
     6,       VZ     , IL     , JL     , IBADD  , WSNINT , NSTRAT            LOTR 107
     7,       ACCX   , ACCY   , ACCZ   , ACRT   , SUTHIT , IPARIN            LOTR 108
     8,       IOTOPO , ISWIND , IMTOPO , IPOUT  , IPAROT , JTOPI             LOTR 109
     9,       SWIND1 , IXNOR  , ILIMIT , ENDTIM , IC     , ISTPAS            LOTR 110
     1,       STOPO  , RESULT , NO     , NTC    , NTI    , NW                LOTR 111
     2,       NALOFT , STIMEI , NBMAX  , NPREL  , N      , NCL               LOTR 112
     3,       CXMAXT , CROHT  , VIRTFP , DZ     , CRMINX , CRMINY            LOTR 113
     4,       CO     , SN     , CS     , NLOCIN , DTLOC  , ATEMP             LOTR 114
     5,       RHO    , NA     , TSZ    , STMAC  , PROB   , CRMAXX            LOTR 115
     6,       RUPART                                                         LOTR 116
      DIMENSION TOPOLM(4,4)    ,WINTAR(4)     ,ITOPLM(3,4)                   LOTR 117
      DIMENSION STI(6,10)      ,SUBSID(400)   ,IC(18)                        LOTR 118
```

```
      DIMENSION XP(200)         ,YP(200)         ,ZP(200)         ,FMAS(200)
      DIMENSION TP(200)         ,PS(200)         ,ATEMP(200)      ,RHO(200)
      DIMENSION VX(1500)        ,VY(1500)        ,VZ(1500)        ,IL(70)
      DIMENSION JL(70)          ,IBADD(70)       ,WORX(70)
      DIMENSION WORINT(70)                       ,WLEX(70)        ,WLEY(70)
      DIMENSION WORY(70)        ,OUTHIT(70)      ,SN(6)           ,CS(6)
      DIMENSION CRMINX(6)       ,CRMAXX(6)       ,CRMINY(6)       ,CRMAXY(6)
      DIMENSION CROHT(6)        ,NCRTYP(6)       ,UU(6)
C
C ***********************************************************************
C
      DATA PROGRM/6HLOTRAN/
C
C ***********************************************************************
C ***********************************************************************
C
C     PUT LOCAL CIRCULATION TYPE NUMBER IN TEMPORARY STORAGE
      NCIR=NCRTYP(K)
C     MAKE AN ASSIGNMENT TO ALLOW AN EFFICIENT
C     BRANCH TO THE APPROPRIATE LOCAL CIRCULATION TRANSPORT CODE
C     IN THE ACTUAL LOCAL TRANSPORT LOOP
      GO TO (101,102,103,104,105),NCIR
 101  ASSIGN 121 TO NC
      GO TO 120
 102  ASSIGN 122 TO NC
      GO TO 120
 103  ASSIGN 123 TO NC
      GO TO 120
 104  ASSIGN 124 TO NC
      GO TO 120
 105  ASSIGN 125 TO NC
C
C 120 BEGIN THE LOCAL TRANSPORT LOOP.
C     FIRST DETERMINE IF THE PARTICLE IS BELOW THE LEVEL OF THE TOP
C     OF THE K-TH LOCAL CIRCULATION SYSTEM CELL.  IF IT IS NOT, CALL
C     GETWIND TO GET THE MACRO WIND VECTOR AT PARTICLE POSITION AND THEN
C     MOVE THE PARTICLE BY TRANSFERING TO 130. IF PARTICLE IS WITHIN THE
C     LOCAL CELL, A BRANCH MUST BE MADE TO THE APPROPRIATE SUBROUTINE.
 120  ZZ=ZP(J)
      PSIZE=PS(J)
      CALL FALRAT(ZZ,PSIZE,FV,ATEMP,RHO,PROG,ISOUT)
      IF(ZP(J)-CROHT(K))1202,1202,1201
C 1201 PARTICLE IS ABOVE LOCAL CELL
 1201 XX = XP(J)
      YY = YP(J)
      ZZ = ZP(J)
      CALL GETWND(XX,YY,ZZ,JWAD,JW)
C     WAS THE MACRO WIND FIELD SPECIFICATION AVAILABLE FOR THE PARTICLE
C     POSITION.  NO TO 1203
      IF(JWAD)1203,1203,1204
C 1203 MACRO WIND FOR THIS PARTICLE IS NOT AVAILABLE.
 1203 IRROR = 1203
      GO TO 7734
C
C 1204 COMPUTE VERTICAL COMPONENT OF PARTICLE VELOCITY
 1204 VPZ=VZ(JWAD)-FV
C
C     COMPUTE TIMES OF FLIGHT TO ALL BOUNDARY INTERCEPTS THAT ARE IN THE
```

```
C      FUTURE.  FIRST COMPUTE TIME TO COMING X BOUNDARY OF LOCAL CELL
       IF(VX(JWAD))1205,1206,1206
 1205  TXX=(XP(J)-CRMINX(K))/VX(JWAD)
       GO TO 1207
 1206  TXX=(CRMAXX(K)-XP(J))/VX(JWAD)
C
C 1207 COMPUTE TIME  TO COMING Y BOUNDARY INTERCEPT
 1207  IF(VY(JWAD))1208,1209,1209
 1208  TYY=(YP(J)-CRMINY(K))/VY(JWAD)
       GO TO 1210
 1209  TYY=(CRMAXY(K)-YP(J))/VY(JWAD)
C
C 1210 COMPUTE TIMES TO COMING Z BOUNDARY (HORIZONTAL) OF MACRO WIND
C      FIELD AND TO THE TOP OF THE LOCAL WIND CELL
 1210  IF(VPZ)1211,1212,1212
 1211  TCELLT=(CROHT(K)-ZP(J))/VPZ
       JWT=JW
       GO TO 1213
 1212  TCELLT=1.0E+08
       JWT=JW+1
 1213  TLAYER=(BOTHT(JWT)-ZP(J))/VPZ
C
C       COMPUTE TIME UNTIL TIME FOR UPDATING THE WIND FIELD
       TTIM=ENDTIM-TP(J)
C
C      NOW SELECT THE TIME  UNTIL THE FIRST VALID BOUNDARY INTERCEPT
C      ADD A SMALL INCREMENT TO PUSH THE PARTICLE PAST THE BOUNDARY OF
C      THE LOCAL CELL.
       TTRANS=AMIN1(TXX,TYY,TCELLT,TLAYER,TTIM) +.01
C
C      NOW TRANSPORT THE PARTICLE FOR THAT PERIOD OF TIME
       XP(J)=XP(J)+TTRANS*VX(JWAD)
       YP(J)=YP(J)+TTRANS*VY(JWAD)
       ZP(J)=ZP(J)+TTRANS*VPZ
       TP(J)=TP(J)+TTRANS
       GO TO 131
C
C 1202 PARTICLE IS WITHIN LOCAL CELL.  BRANCH TO CALL APPROPRIATE WIND
C      PROGRAM
C      BRANCH TO CALL A LOCAL WIND PROGRAM TO TEST PARTICLE FOR IMPACT ONTO
C      TOPOGRAPHY.  IF IMPACTED,IT ASSIGNS A LARGE DOWNWARD WIND COM-
C      PONENT.  IF NOT IMPACTED, IT COMPUTES CORRECT WIND COMPONENTS AT
C      THE PARTICLE POSITION
 1202  GO TO NC,(121,122,123,124,125)
 121   CALL MTWND1(J,K,AX,AY,AZ)
       GO TO 130
 122   CALL RGWND1(J,K,AX,AY,AZ)
       GO TO 130
 123   CALL CBREZ1(J,K,AX,AY,AZ)
       GO TO 130
C *********  CODE INSERTION  POINTS  **********************************
 124   CONTINUE
 125   CONTINUE
C *********  CODE INSERTION  POINTS  **********************************
 126   IRROR=-126
 7734  CALL ERROR(PROGRM,IRROR,ISOUT)
C
C 130  TRANSPORT THE PARTICLE FOR ONE TIME INCREMENT  (DTLOC).
```

```
 130  ZP(J)=ZP(J)+ DTLOC*(AZ-FV)                                    LOTR 235
      XP(J)=XP(J)+ DTLOC*AX                                         LOTR 236
      YP(J)=YP(J)+ DTLOC*AY                                         LOTR 237
      TP(J)=TP(J)+ DTLOC                                            LOTR 238
C                                                                   LOTR 239
C 131 TEST FOR BOUNDARY CROSSINGS                                   LOTR 240
C     IS PARTICLE AT OR BEYOND THE LOCAL CIRCULATION BOUNDARIES     LOTR 241
C     YES TO 132                                                    LOTR 242
 131  IF(XP(J)-CRMINX(K))132,132,133                                LOTR 243
 133  IF(CRMAXX(K)-XP(J))132,132,134                                LOTR 244
 134  IF(YP(J)-CRMINY(K))132,132,135                                LOTR 245
 135  IF(CRMAXY(K)-YP(J))132,132,137                                LOTR 246
C 137 TEST TO REMOVE IMPACTED PARTICLES                             LOTR 247
C     WAS THE PARTICLE BELOW THE ANALYTICAL TOPOGRAPHY WHEN THE LOCAL LOTR 248
C     WIND WAS COMPUTED.   YES TO 132                               LOTR 249
 137  IF(AZ-1.0E+06)132,132,136                                     LOTR 250
C                                                                   LOTR 251
C 136 PARTICLE IS STILL WITHIN KTH LOCAL SYSTEM. NOW CHECK TIME BOUNDARYLOTR 252
C     HAS THE PARTICLE BEEN TRANSPORTED UP TO OR BEYOND THE TIME FOR LOTR 253
C     UPDATING THE WIND FIELD                                       LOTR 254
 136  IF(TP(J)-ENDTIM)120,132,132                                   LOTR 255
C                                                                   LOTR 256
C 132 PARTICLE CANNOT BE MOVED FURTHER BY LOCAL SYSTEM TRANSPORT CODE LOTR 257
C                                                                   LOTR 258
  132 RETURN                                                        LOTR 259
      END                                                           LOTR 260
                                                                        261*

                                                                        261 *
```

```
$IBFTC MTWN1    LIST,DECK,M94/2
      SUBROUTINE MTWND1(J,K,AX,AY,AZ)
C     11 OCT 66
C     I. KOHLBERG, T.W.SCHWENKE   TECHNICAL OPERATIONS RESEARCH, INC.
C
C     THIS SUBROUTINE SERVES THE DUAL PURPOSE OF READING MOUNTAIN WIND
C     DATA WHEN THE SIGN OF ARGUMENT J IS MINUS) AND COMPUTING THE
C     MOUNTAIN WIND FOR THE JTH PARTICLE AFTER FIRST CHECKING FOR IMPACT
C     ON THE ANALYTICAL GROUND.  IF IMPACT IS SENSED THE PARTICLE IS
C     ASSIGNED A LARGE DOWNWARD VELOCITY COMPONENT.
C
C ***********************************************************************
C
      COMMON /SET1/
     1           DIAM   , DETID  , IRISE  , IEXEC  , ISTN   , ISOUT  ,
     2           SD     , SPAR   , SSAM   , TME    , TMP1   , TMP2   ,
     3           T2M    , U      , VPR    , W      , X      , Z      ,
     4           WHY    , RMIN   , IDISTR , SPAR1  , SPAR2  , SPAR3  ,
     5           SPAR4  , SPAR5  , SPAR6  , SPAR7  , SPAR8  , SPAR9
C
C ***********************************************************************
C
      DIMENSION  DETID(12),WHY(40)
      COMMON /SET2/
     1           G      , SUBSID , GRINT  , BXLL   , BXLU   , BYLL
     2,          BYLU   , TXLL   , TXLU   , TYLL   , TYLU   , XGZ
     3,          YGZ    , NBLCK  , HTOPO  , TTOPO  , ILIM   , JLIM
     4,          KLIM   , II     , JJ     , KK     , XP     , YP
     5,          ZP     , FMAS   , TP     , PS     , VX     , VY
     6,          VZ     , IL     , JL     , IBADD  , WGRINT , NOTRAT
     7,          WLLX   , WLLY   , WORX   , WORY   , BOTHIT , IPARTN
     8,          IOTOPO , IOWIND , IHTOPO , IPOUT  , IPAROT , JTOP1
     9,          OWIND1 , IRROR  , TLIMIT , ENDTIM , IC     , IBYPAS
     1,          JTOPO  , NLOST  , NG     , NTO    , NTI    , NW
     2,          NALOFT , JTIME1 , NOMAX  , NFREE  , N      , NCL
     3,          CRMAXY , CROHT  , NCRTYP , SZ     , CRMINX , CRMINY
     4,          OO     , SN     , CS     , NLOCIR , DILOC  , ATEMP
     5,          RHO    , MX     , TGZ    , DTMAC  , FROG   , CRMAXX
     6,          ROPART
      DIMENSION TOPOLM(4,4)    ,NINTAR(4)      ,ITOPLM(3,4)
      DIMENSION S(1 ,10)       ,SUBSID(400)    ,IC(18)
      DIMENSION XP(200)        ,YP(200)        ,ZP(200)        ,FMAS(200)
      DIMENSION TP(200)        ,PS(200)        ,ATEMP(200)     ,RHO(200)
      DIMENSION VX(1500)       ,VY(1500)       ,VZ(1500)       ,IL(70)
      DIMENSION JL(70)         ,IBADD(70)      ,WORX(70)
      DIMENSION WGRINT(70)                     ,WLLX(70)       ,WLLY(70)
      DIMENSION WORY(70)       ,BOTHIT(70)     ,SN(6)          ,CS(6)
      DIMENSION CRMINX(6)      ,CRMAXX(6)      ,CRMINY(6)      ,CRMAXY(6)
      DIMENSION CROHT(6)       ,NCRTYP(6)      ,OO(6)
C ***********************************************************************
C
C     PARAMETERS PECULIAR TO SUBROUTINE MTWND1
C
      DIMENSION XM(12),YM(12),H(12),A(12),A2(12),A2H(12),A3H(12)
C
C******************MOUNTAIN WIND1 GLOSSARY*******************************
C
C     A(J)           HALF-WIDTH OF THE J-TH MOUNTAIN
C     AX             WIND VECTOR EAST
C     AY             WIND VECTOR NORTH
C     AZ             VERTICAL WIND VECTOR
```

```
C       CN(K)             COSINE OF COUNTER CLOCKWISE ANGLE FROM EAST OF
C                         UNPERTURBED WIND
C       CRMAXY(K)         NORTH      BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMINY(K)         SOUTH      BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMAXX(K)         EAST       BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMINX(K)         WEST       BOUNDARY OF THE K-TH LOCAL CELL.
C       CRGHT(K)          THE LOCAL CELL TOP HEIGHT  (METERS)
C       DELX
C       DELX              X COORDINATE OF MACROSYSTEM TRANSLATED INTO LOCAL
C                         CELL COORDINATES
C       DELY              Y COORDINATE         SEE DELX
C       DSX               THE X PRIME RESULT OF ROTATION OF (DELX,DELY) INTO
C                         THE K-TH LOCAL COORDINATE SYSTEM.
C       DSY               THE Y PRIME RESULT      SEE DSX
C       DZ                SUM OF TOPO HEIGHT INCREMENTS
C       DZ                MOUNTAIN RATIO OF H(J)/A(J)
C       H(J)              HEIGHT OF THE J-TH MOUNTAIN
CC      NMT               THE NUMBER OF MOUNTAINS REPRESENTED IN THIS MOUNTAIN
C                         WIND SYSTEM
C       SN(K)             SINE OF ANGLE COUNTER-CLOCKWISE FROM EAST OF
C                         UNPERTURBED WIND
C       UU(K)             MAGNITUDE OF UNPERTURBED WIND VECTOR
C       XM(J)             X LOCATION COORDINATE OF THE J-TH MOUNTAIN
C       YM(J)             Y LOCATION COORDINATE OF THE J-TH MOUNTAIN
C       XX                X COORDINATE OF CENTER OF LOCAL CELL
C       YY                Y COORDINATE OF CENTER OF LOCAL CELL
C       ZZ                Z COORDINATE OF CENTER OF LOCAL CELL
C       JW                INDEX OF THE WIND STRATUM CONTAINING THE PARTICLE
C       JWMD              INDEX OF MACRO WIND CELL CONTAINING PARTICLE
C       XP(J)             PARTICLE POSITION COORDINATE
C       YP(J)             PARTICLE POSITION COORDINATE
C       ZP(J)             PARTICLE POSITION COORDINATE
C
C ***********************************************************************
C
 1      FORMAT(4F10.3)
 2      FORMAT(//25X24HLOCAL CIRCULATION NUMBER16/30X15HMOUNTAIN WIND 1//1
       15X6HMOUNTAIN4X6HMOUNTAIN4X8HMOUNTAIN4X20HLOCATION COORDINATES/15X6
       2HNUMBER6X6HHEIGHT6X5HWIDTH/X1HX11X1HY)
 3      FORMAT(/12X,I5,6X,4F11.5//)
 4      FORMAT(//25X,36HCOORDINATES OF LOCAL CELL BOUNDARIES/,15X,5HNORTH,
       110X,5HSOUTH,11X,4HEAST,11X,4HWEST,10X,6HHEIGHT//,10X,5F15.5//)
 5      FORMAT(/25X,46HCHARACTERISTICS OF THE UNPERTURBED WIND VECTOR/20X,
       116HUNPERTURBED WIND,4X,17HCOSINE OF ANGULAR,4X,15HSINE OF ANGULAR,
       2/20X,16HVECTOR MAGNITUDE,5X,15HDEV. FROM NORTH,5X,15HDEV. FROM NOR
       3TH//,18X,3F20.5//)
C
C ***********************************************************************
C
      DATA PROGRM/6HMTWND1/
C
C ***********************************************************************
C ***********************************************************************
C
      IF(J)100,101,102
 101  IRROR=-101
7734  CALL ERROR(PROGRM,IRROR,ISOUT)
C
```

```
C 100 THIS IS THE DATA READING ROUTE
 100  J=0
      CRUHT(K)=0.0
 103  J=J+1
      READ (ISIN,1) XM(J),YM(J),H(J),A(J)
      A2(J)=A(J)*A(J)
      A2H(J)=A2(J)*A(J)*H(J)
      IF(J-12)1031,1032,1032
 1032 IRROR=1032
      GO TO 7734
 1031 IF(H(J))1099,104,1099
C1099 COMPUTE THE KTH LOCAL CELL HEIGHT.
 1099 DZ=ABS(3.0*H(J))
      IF (DZ-CRUHT(K))110,110,1100
 1100 CRUHT(K)=DZ
C 110 CHECK TO SEE THAT THE MOUNTAIN JUST READ IS WITHIN THE LIMITS OF
C     THE KTH LOCAL WIND SYSTEM.
  110 IF(XM(J)-CRMINX(K)) 114,111,111
  111 IF(XM(J)-CRMAXX(K)) 112,112,114
  112 IF  (YM(J)-CRMINY(K))  114,113,113
 113  IF  (YM(J)-CRMAXY(K))  115,115,114
C
C 114 THE MOUNTAIN IS NOT WITHIN THE LIMITS OF THE KTH LOCAL WIND SYSTEM
  114 IRROR= 114
      GO TO 7734
C
C 115 CHECK TO SEE THAT THE MOUNTAIN RATIO H(J)/A(J) IS LESS THAN 0.6
 115  DZ = H(J)/A(J)
      IF (DZ-0.6)103,116,116
C
C 116 THE MOUNTAIN RATIO H(J)/A(J) IS NOT LESS THAN  0.6
  116 IRROR = 116
      GO TO 7734
C
 104  NMT=J-1
C     NMT               THE NUMBER OF MOUNTAINS REPRESENTED IN THIS
C                       MOUNTAIN WIND SYSTEM
C 1042  COMPUTE UNPERTURBED VECTOR HERE
C     THE FOLLOWING THREE CARDS CONSTITUTE THE LOCATION COORDINATES OF
C     THE UNPERTURBED WIND VECTOR
      YY=(CRMAXY(K)+CRMINY(K))/2.0
      XX=(CRMAXX(K)+CRMINX(K))/2.0
      ZZ=CRUHT(K)/2.0
      CALL GETWND(XX,YY,ZZ,JWAD,JW)
      IF(JWAD)1043,1044,1045
 1043 IRROR=1043
      GO TO 7734
 1044 IRROR=1044
      GO TO 7734
C     THE FOLLOWING THREE CARDS CONSTITUTE THE MAGNITUDE AND DIRECTION
C     OF THE UNPERTURBED WIND VECTOR
 1045 UO(K)=SQRT(VX(JWAD)*VX(JWAD)+VY(JWAD)*VY(JWAD))
      SN(K)=VY(JWAD)/UO(K)
      CS(K)=VX(JWAD)/UO(K)
      DO 1049 J=1,NMT
 1049 A2H(J)=A2(J)*H(J)*UO(K)
      WRITE (ISOUT,2)K
      WRITE (ISOUT,3)(J,H(J),A(J),XM(J),YM(J),J=1,NMT)
```

```
      WRITE (ISOUT,4)CRMAXY(K),CRMINY(K),CRMAXX(K),CRMINX(K),CRGHT(K)
      WRITE (ISOUT,5)UO(K),SN(K),CS(K)
 105  RETURN
C
C 102 THIS IS THE TESTING AND COMPUTING ROUTE
C      COMPUTE THE TOPOGRAPHIC INCREMENT AT POSITION OF THE JTH PARTICLE
C     COMPUTE THE PERTURBED WIND COMPONENTS, SUM THEM, AND ADD THEM
C     TO THE UNPERTURBED WIND VECTOR.
 102  AX=0.0
      AY=0.0
      AZ=0.0
      DZ=0.0
      DO 106 I=1,NMT
C     THE FOLLOWING TWO CARDS TRANSLATE THE PARTICLE INTO THE MOUNTAIN
C     COORDINATE SYSTEM.
      DELX=XP(J)-XM(I)
      DELY=YP(J)-YM(I)
C     THE FOLLOWING TWO CARDS ROTATE THE PARTICLE INTO THE MOUNTAIN
C     COORDINATE SYSTEM.
      DSX = DELX * CS(K) + DELY * SN(K)
      DSY =-DELX * SN(K) + DELY * CS(K)
      Y2 = DSY * DSY
      X2 = DSX * DSX
      R2=X2+Y2
C     NOW COMPUTE TOPO HEIGHT INCREMENT RESULTING FROM THE ITH MOUNTAIN
C     AND ADD IT TO SUM.
      DZ = DZ + A3H(I)/((A2(I)+R2)*SQRT(A2(I)+R2))
C
C     COMPUTE PERTURBATION WIND INCREMENTS
      AMBDA = ZP(J)+A(I)
      AMBDA2=AMBDA*AMBDA
      DENOM= (R2+AMBDA2)*(R2+AMBDA2)*SQRT(R2+AMBDA2)
      Q= A2H(I)*3.0*DSX/DENOM
C
C1061   AX, THE PERTURBED WIND COMPONENT IN THE DIRECTION OF THE
C     UNPERTURBED WIND.
 1061 AX=AX+A2H(I)*(Y2+AMBDA2-2.0*X2)/DENOM
C     AY, AZ, THE PERTURBED WIND COMPONENTS PERPENDICULAR TO THE
C     DIRECTION OF THE UNPERTURBED WIND.
      AY= AY - Q*DSY
 106  AZ= AZ - Q*AMBDA
C
C     NOW TEST FOR IMPACTED PARTICLE
      IF(DZ-ZP(J))109,108,108
C 108  PARTICLE HAS IMPACTED
 108  AZ=-1.0E+08
      AX=0.0
      AY=0.0
      GO TO 105
C 109 THE PARTICLE IS ALOFT.  NOW ADD THE UNPERTURBED  WIND VECTOR TO
C     THE PERTURBED COMPONENT IN THE SAME DIRECTION.
 109  DELX = AX+UO(K)
C     THE FOLLOWING TWO CARDS DEROTATE THE WIND VECTOR INTO THE
C     MACRO SYSTEM.
      AX=DELX*CS(K)-AY*SN(K)
      AY=DELX*SN(K)+AY*CS(K)
      GO TO 105
      END
```

```
$IBFTC RGWN1    LIST,DECK,M94/2                                          RGWN
      SUBROUTINE RGWND1(J,K,AX,AY,AZ)                                    RGWN
C     11 OCT 66                                                          RGWN
C     I. KOHLBERG, T.W.SCHWENKE  TECHNICAL OPERATIONS RESEARCH, INC.     RGWN
C                                                                        RGWN
C     THIS SUBROUTINE SERVES THE DUAL PURPOSE OF READING RIDGE WIND      RGWN
C     DATA WHEN THE SIGN OF ARGUMENT J IS MINUS) AND COMPUTING THE       RGWN
C     RIDGE WIND FOR THE JTH PARTICLE AFTER FIRST CHECKING FOR IMPACT    RGWN
C     ON THE ANALYTICAL GROUND.  IF IMPACT IS SENSED THE PARTICLE IS     RGWN
C     ASSIGNED A LARGE DOWNWARD VELOCITY COMPONENT.                      RGWN
C                                                                        RGWN
C **********************************************************************RGWN
C                                                                        RGWN
      COMMON /SET1/                                                      RGWN
     1              DIAM    , DETID   , IRIDE   , IEXEC   , IDIN    , ISOUT   ,  RGWN
     2              SD      , SPAN    , SSAM    , TIME    , TMP1    , TMP2    ,  RGWN
     3              T2M     , U       , VPR     , W       , X       , Z       ,  RGWN
     4              WHY     , RMIN    , IDISTR  , SPAR1   , SPAR2   , SPAR3   ,  RGWN
     5              SPAR4   , SPAR5   , SPAR6   , SPAR7   , SPAR8   , SPAR9      RGWN
C                                                                        RGWN
C **********************************************************************RGWN
C                                                                        RGWN
      DIMENSION   DETID(12),WHY(40)                                      RGWN
      COMMON /SET2/                                                      RGWN
     1              S       , SUBSID  , GRINT   , BXLL    , BXLU    , BYLL     RGWN
     2,             BYLU    , IXLL    , IXLU    , IYLL    , IYLU    , XGZ      RGWN
     3,             YGZ     , NBLCK   , HTOPO   , ITOPO   , ILIM    , JLIM     RGWN
     4,             KLIM    , II      , JJ      , KK      , XP      , YP       RGWN
     5,             ZP      , FMAS    , TP      , PS      , VA      , VY       RGWN
     6,             VZ      , IL      , JL      , ISADD   , WORINT  , NSIRAT   RGWN
     7,             WLLX    , WLLY    , WORX    , WORY    , BOTHIT  , IPARIN   RGWN
     8,             IOTOPO  , IOWIND  , IHTOPO  , IPOUT   , IPAROT  , JTOP1    RGWN
     9,             JWIND1  , IRROR   , TLIMIT  , ENDTIM  , IC      , IBYPAS   RGWN
     1,             JTOPO   , NLOST   , N3      , NTO     , NTI     , NW       RGWN
     2,             NALOFT  , DTIME1  , N3MAX   , NFREE   , N       , NCL      RGWN
     3,             CRMAXY  , CROHT   , NCRTYP  , BZ      , CRMINX  , CRMINY   RGWN
     4,             UU      , SN      , CS      , NLOCIR  , DTLOC   , ATEMP    RGWN
     5,             RHO     , NA      , TGZ     , DTMAC   , FROG    , CRMAXX   RGWN
     6,             ROPART                                               RGWN
      DIMENSION TOPOLM(4,4)    ,WINTAR(4)      ,ITOPLM(3,4)              RGWN
      DIMENSION S(10,10)       ,SUBSID(400)    ,IC(18)                   RGWN
      DIMENSION XP(200)        ,YP(200)        ,ZP(200)       ,FMAS(200) RGWN
      DIMENSION TP(200)        ,PS(200)        ,ATEMP(200)    ,RHO(200)  RGWN
      DIMENSION VA(1500)       ,VY(1500)       ,VZ(1500)      ,IL(70)    RGWN
      DIMENSION JL(70)         ,ISADD(70)      ,WORX(70)                 RGWN
      DIMENSION WORINT(70)                     ,WLLX(70)      ,WLLY(70)  RGWN
      DIMENSION WORY(70)       ,BOTHIT(70)     ,SN(6)         ,CS(6)     RGWN
      DIMENSION CRMINX(6)      ,CRMAXX(6)      ,CRMINY(6)     ,CRMAXY(6) RGWN
      DIMENSION CROHT(6)       ,NCRTYP(6)      ,UU(6)                    RGWN
C                                                                        RGWN
C **********************************************************************RGWN
C                                                                        RGWN
C     PARAMETERS PECULIAR TO SUBROUTINE RGWND1                           RGWN
C                                                                        RGWN
      DIMENSION XM(12),YM(12),H(12),A(12),B(12),C(12),SG(12),CG(12),     RGWN
     1A2(12),AH(12),D(12),A2H(12)                                        RGWN
C                                                                        RGWN
C*************************RIDGE WIND 1 GLOSSARY*********************** RGWN
C                                                                        RGWN
C     A(K)        THE HALF WIDTH OF THE KTH RIDGE                        RGWN
C     AX          WIND VECTOR EAST                                       RGWN
```

```
C       AY              WIND VECTOR NORTH
C       AZ              VERTICAL WIND VECTOR
C       AX              PERTURBED WIND COMPONENT IN THE DIRECTION OF THE
C                       UNPERTURBED WIND
C       B(K)            CLOCKWISE ANGULAR DEVIATION OF KTH RIDGE FROM
C                       TRUE NORTH
C       CCG             COSINE OF ANGLE UNPERTURBED WIND MAKES WITH
C                       PERPENDICULAR FROM KTH RIDGE
C       CN(K)           COSINE OF COUNTER-CLOCKWISE ANGLE FROM EAST OF
C                       UNPERTURBED WIND
C       CRMAXY(K)       NORTH       BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMINY(K)       SOUTH       BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMAXX(K)       EAST        BOUNDARY OF THE K-TH LOCAL CELL.
C       CRMINX(K)       WEST        BOUNDARY OF THE K-TH LOCAL CELL.
C       CRQHT(K)        THE LOCAL CELL TOP HEIGHT  (METERS)
C       DDX             THE PERPENDICULAR DISTANCE OF THE JTH PARTICLE FROM
C                       THE ITH RIDGE  (METERS)
C       DZ              SUM OF TOPO HEIGHT INCREMENTS
C       DZ              RIDGE RATIO OF H(J)/A(J)
C       H(K)            THE HEIGHT OF THE KTH RIDGE
C       NRG             NUMBER OF RIDGES IN THIS RIDGE WIND SYSTEM
C       SN(K)           SINE OF ANGLE COUNTER-CLOCKWISE FROM EAST OF
C                       UNPERTURBED WIND
C       SSG             SINE OF ANGLE UNPERTURBED WIND MAKES WITH
C                       PERPENDICULAR FROM KTH RIDGE
C       UU(K)           MAGNITUDE OF UNPERTURBED WIND VECTOR
C       XM(J)           X LOCATION COORDINATE OF THE J-TH RIDGE
C       XX              X COORDINATE OF CENTER OF LOCAL CELL
C       YM(J)           Y LOCATION COORDINATE OF THE J-TH RIDGE
C       YY              Y COORDINATE OF CENTER OF LOCAL CELL
C       ZZ              Z COORDINATE OF CENTER OF LOCAL CELL
C       JW              INDEX OF THE WIND STRATUM CONTAINING THE PARTICLE
C       JWAD            INDEX OF MACRO WIND CELL CONTAINING PARTICLE
C       XP(J)           PARTICLE POSITION COORDINATE
C       YP(J)           PARTICLE POSITION COORDINATE
C       ZP(J)           PARTICLE POSITION COORDINATE
C
C ********************************************************************
C
 1      FORMAT(5F10.3)
 2      FORMAT(//25X24HLOCAL CIRCULATION NUMBER16/30X15H  RIDGE  WIND I//I
       15X5HRIDGE/X5HRIDGE/X5HRIDGE/X20HLOCATION COORDINATES5X15HRIDGE AZI
       2MUTH/15X6HNUMBER5X6HHEIGHT6X5HWIDTH5X1HX11X1HY)
 3      FORMAT(/12X,I6,5X,4F12.3,8X,F11.3//)
 4      FORMAT(//25X,36HCOORDINATES OF LOCAL CELL BOUNDARIES/,15X,5HNORTH,
       110X,5HSOUTH,11X,4HEAST,11X,4HWEST,10X,6HHEIGHT//,7X,5F15.3///)
 5      FORMAT(/25X,46HCHARACTERISTICS OF THE UNPERTURBED WIND VECTOR/20X,
       116HUNPERTURBED WIND,4X,17HCOSINE OF ANGULAR,4X,15HSINE OF ANGULAR,
       2/20X,16HVECTOR MAGNITUDE,5X,15HDEV. FROM NORTH,5X,15HDEV. FROM NOR
       3TH//,13X,3F20.5//)
C
C ********************************************************************
C
        DATA PROGRM/6HRGWND1/
C
C ********************************************************************
C ********************************************************************
C
```

```
      IF(J)100,101,102
 101  IRROR=-101
 7734 CALL ERROR(PROGRM,IRROR,IOUUT)
C 100 THIS IS THE DATA READING ROUTE
 100  J=0
      CRUHT(K)=0.0
 103  J=J+1
      READ (ISIN,1) XM(J),YM(J),H(J),A(J),B(J)
      C(J) = COS(B(J))
      S(J) = SIN(B(J))
      A2(J) = A(J)*A(J)
      A2H(J)=A2(J)*H(J)
      IF(J-12)1031,1032,1032
 1032 IRROR=1032
      GO TO 7734
 1031 IF(H(J))1099,104,1099
C1099 COMPUTE THE KTH LOCAL CELL HEIGHT.
 1099 DZ=ABS(3.0*H(J))
      IF (DZ-CRUHT(K))110,110,1100
 1100 CRUHT(K)=DZ
C 110 CHECK TO SEE THAT THE RIDGE JUST READ IS WITHIN THE LIMITS OF
C     THE KTH LOCAL WIND SYSTEM.
  110 IF(XM(J)-CRMINX(K)) 114,111,111
  111 IF(XM(J)-CRMAXX(K)) 112,112,114
  112 IF (YM(J)-CRMINY(K)) 114,113,113
 113  IF (YM(J)-CRMAXY(K)) 115,115,114
C
C 114 THE RIDGE IS NOT WITHIN THE LIMITS OF THE KTH LOCAL WIND SYSTEM
  114 IRROR= 114
      GO TO 7734
C
C 115 CHECK TO SEE THAT THE RIDGE RATIO H(J)/A(J) IS LESS THAN 0.6
 115  DZ = H(J)/A(J)
      IF (DZ-0.6) 103,116,116
C
C 116 THE RIDGE RATIO H(J)/A(J) IS NOT LESS THAN  0.6
  116 IRROR = 116
      GO TO 7734
C
 104  NRG=J-1
C     NRG                    THE NUMBER OF RIDGES REPRESENTED IN THIS
C                            RIDGE WIND SYSTEM
C 1042  COMPUTE UNPERTURBED VECTOR HERE
C     THE FOLLOWING THREE CARDS CONSTITUTE THE LOCATION COORDINATES OF
C     THE UNPERTURBED WIND VECTOR
      YY=(CRMAXY(K)+CRMINY(K))/2.0
      XX=(CRMAXX(K)+CRMINX(K))/2.0
      ZZ=CRUHT(K)/2.0
      CALL GETWND(XX,YY,ZZ,JWAD,JW)
      IF(JWAD)1043,1044,1045
 1043 IRROR=1043
      GO TO 7734
 1044 IRROR=1044
      GO TO 7734
C     THE FOLLOWING THREE CARDS CONSTITUTE THE MAGNITUDE AND DIRECTION
C     OF THE UNPERTURBED WIND VECTOR
 1045 UU(K)=SQRT(VX(JWAD)*VX(JWAD)+VY(JWAD)*VY(JWAD))
      SN(K)=VY(JWAD)/UU(K)
```

```
      CS(K)=VX(JWAD)/UU(K)
      DO 1049 J=1,NRG
      SSG   = CS(K)*D(J)+SN(K)*C(J)
      CCG   = CS(K)*C(J)-SN(K)*D(J)
      AH(J)= -A(J)*H(J)*UU(K)*CCG*CCG
      SG(J)=SSG/CCG
 1049 CG(J)= 2.0/CCG
C
      WRITE (ISOUT,2)K
      WRITE (ISOUT,3)(J,H(J),A(J),AH(J),ZP(J),B(J),J=1,NRG)
      WRITE (ISOUT,4)CRMAX(K),CRMIN(K),CMMAX(K),CMMINA(K),CSUM(K)
      WRITE (ISOUT,5)UU(K),SN(K),CS(K)
 105  RETURN
C
C 102 THIS IS THE TESTING AND COMPUTING ROUTE
C      COMPUTE THE TOPOGRAPHIC INCREMENT AT POSITION OF THE JTH PARTICLE
C     COMPUTE THE PERTURBED WIND COMPONENTS, SUM THEM, AND ADD THEM
C     TO THE UNPERTURBED WIND VECTOR.
 102  AX=0.0
      AY=0.0
      AZ=0.0
      DZ=0.0
      DO 106 I=1,NRG
C107  THIS CARD TRANSLATES AND ROTATES THE PARTICLE INTO THE RIDGE
C     WIND SYSTEM.  DSX IS THE PERPENDICULAR DISTANCE OF THE J-TH
C     PARTICLE FROM THE I-TH RIDGE.
 107  DSX = (XP(J)-XM(I))*C(I)-(YP(J)-YM(I))*D(I)
      X2 = DSX*DSX
      AMBDA = ZP(J) +A(I)
      AMBDA2 = AMBDA*AMBDA
C     NOW COMPUTE TOPO HEIGHT INCREMENT RESULTING FROM THE ITH RIDGE
C     AND ADD IT TO DZ.
      DZ= DZ + AZH(I)/(A2(I)+X2)
C
C     THE FOLLOWING CARDS COMPUTE THE PERTURBATION WIND INCREMENTS
      AHCD= AH(I)/((X2+AMBDA2)*(X2+AMBDA2))
      XAMBDA2= X2-AMBDA2
C     AX  IS THE COMPONENT IN THE DIRECTION OF THE UNPERTURBED WIND
      AX=AHCD*XAMBDA2 + AX
      AY= AHCD*SG(I)*AMBDA2 + AY
 106  AZ= AHCD*CG(I)*AMBDA*DSX + AZ
C
C     NOW TEST FOR IMPACTED PARTICLE
      IF(DZ-ZP(J))109,108,108
C 108  PARTICLE HAS IMPACTED
 108  AZ=-1. E+08
      AX=0.0
      AY=0.0
      GO TO 105
C 109 THE PARTICLE IS ALOFT.  NOW ADD THE UNPERTURBED  WIND VECTOR TO
C     THE PERTURBED COMPONENT IN THE SAME DIRECTION.
 109  DSX=AX+UU(K)
C     THE FOLLOWING TWO CARDS DEROTATE THE WIND SYSTEM INTO MACRO FIELD
C     COORDINATES
      AX= DSX*CS(K)-AY*SN(K)
      AY= DSX*SN(K)+AY*CS(K)
      GO TO 105
      END
```

235*

235 *

```
$IBFTC CBRE11  LIST,DECK,M94/2
      SUBROUTINE CBREZ1(J,K,AX,AY,AZ)
C     11 OCT 66
C     I. KOHLBERG, T.W.SCHWENKE  TECHNICAL OPERATIONS RESEARCH, INC.
C     THIS SUBROUTINE SERVES THE DUAL PURPOSE OF READING SEA BREEZE
C     DATA(WHEN THE SIGN OF ARGUMENT J IS MINUS) AND COMPUTING THE
C     SEA BREEZE FOR THE J-TH PARTICLE.
C
C ***********************************************************************
C
      COMMON /SET1/
     1          DIAM    , DETIO  , IRISE  , IEACC  , IOIN   , IOUUT   ,
     2          DD      , DPAR   , DDAR   , TME    , IMP1   , IMP2    ,
     3          T2      , D      , VPR    , W      , X      , Z       ,
     4          AMT     , RAIN   , IDIOTR , DPAR1  , DPAR2  , DPAR3   ,
     5          DPAR4   , DPAR5  , DPAR6  , DPAR7  , DPAR8  , DPAR9
C
C ***********************************************************************
C
      DIMENSION  DETIO(12),WHY(40)
      COMMON /SET2/
     1,         D       , DDBDIO , GRINT  , DALL   , DALD   , ETLL    ,
     2,         BYLD    , TALL   , TALD   , TTLL   , TTLD   , ASE     ,
     3,         YGZ     , NBLCK  , MICRO  , TIORC  , ILIM   , DLIM    ,
     4,         NLIM    , II     , JJ     , KK     , XP     , YP      ,
     5,         ZP      , FRAD   , TP     , PO     , VA     , VT      ,
     6,         VZ      , IL     , JL     , IDADD  , WORKIN , NOTRAI  ,
     7,         WLLA    , WLLT   , WORA   , WORT   , DOTIMI , IPARIN  ,
     8,         ICTOPC  , IDWIND , IMTOPC , IPOUT  , IDROUT , DTOP1   ,
     9,         DWINDI  , IRNOR  , TLIMIT , [illegible] , IC     , ICTPAC  ,
     1,         DTOPD   , NLOST  , NG     , NTC    , NTI    , NA      ,
     2,         NALOFT  , DTTMLI , NDMAX  , [illegible] , N      , NCL     ,
     3,         CRMAXT  , CROMI  , NCRITT , DZ     , CRMINA , CRMINY  ,
     4,         DD      , DN     , CS     , NLOCIR , DTLOC  , ATLM    ,
     5,         AMU     , NA     , TDZ    , [illegible] , FROD   , CRMAXA  ,
     6,         NOPART
      DIMENSION  ICPOLM(4,4)     ,NINTAR(4)        ,ITOPLM(5,4)
      DIMENSION  D(10,10)        ,DDBDIO(400)      ,IC(15)
      DIMENSION  XP(200)         ,YP(200)          ,ZP(200)          ,FRAD(200)
      DIMENSION  TP(200)         ,PO(200)          ,ATLMP(200)       ,[illegible](200)
      DIMENSION  VA(1500)        ,VT(1500)         ,VZ(1500)         ,IL(70)
      DIMENSION  JL(70)          ,IDADD(70)        ,WORA(70)
      DIMENSION  WORIN(70)                         ,WLLA(70)         ,WLLT(70)
      DIMENSION  WORT(70)        ,DOTIMI(70)       ,DN(5)            ,CS(5)
      DIMENSION  CRMINA(5)       ,CRMAXA(5)        ,CRMINY(5)        ,CRMAXT(5)
      DIMENSION  CROMI(6)        ,NCRITT(5)        ,DD(5)
C
C ***********************************************************************
C     PARAMETERS  PECULIAR TO CBREZ
      DIMENSION AOZA(9),ACAA(9),FAA(9),R1IA(9),R2IA(9),RINA(9),
     1RZRA(9),CON1A(9),CON2A(9),CANIA(9),CANZA(9),CNNIA(9),
     2CNNZA(9),CNNIA(9),CNNZA(9),CNGA(9),DELTA(9),TROA(9)
C
C*****************SEA BREEZE GLOSSARY*************************************
C
C     ACCG            GRAVITATIONAL CONSTANT (METERS/SQR. SECOND)
C     AKT             THERMAL EDDY DIFFUSIVITY (SQR.METERS/ SECOND)
C     AX              WIND VECTOR EAST     (METERS/SECOND)
C     AY              WIND VECTOR NORTH    (METERS/SECOND)
C     AZ              WIND VECTOR VERTICAL  (METERS/SECOND)
C     B               THE ANGLE (MEASURED CLOCKWISE FROM TRUE NORTH) THAT
```

```
C                            DESCRIBES THE ORIENTATION OF THE SHORE LINE.  FOR B   CORE  61
C                            EQUAL TO ZERO,THE SEA IS ON THE LEFT HAND SIDE OF THECORE  62
C                            SHORELINE. (RADIANS)                                  CORE  63
C        BB                  A GENERAL PURPOSE VARIABLE USED FOR                   CORE  64
C                               CONSTANT STORAGE IN READ ROUTE,                    CORE  65
C                               WIND VECTOR PERPENDICULAR TO SHORE                 CORE  66
C        CKDHT               HEIGHT OF THE SEA BREEZE CELL   (METERS)              CORE  67
C        DELTX(N)            MAXIMUM TEMPERATURE DIFFERENTIAL FOR THE NTH          CORE  68
C                            HARMONIC  (DEGREES KELVIN)                            CORE  69
C        ELX                 TOTAL EXTENT OF THE SEA BREEZE CELL (METERS)          CORE  70
C        GRAD                INITIAL TEMPERATURE GRADIENT (DEGREES KELVIN /METER)  CORE  71
C        NH                  NUMBER OF HARMONICS                                   CORE  72
C        [illegible]         [illegible] PARAMETER  (1/ SECONDS)                   CORE  73
C        SPHI                SINE OF THE LATITUDE OF THE SEA BREEZE CELL           CORE  74
C        TAUX(N)             PHASE LAG OF NTH HARMONIC    (RADIANS)                CORE  75
C        TM                  AVERAGE GROUND TEMPERATURE (DEGREES KELVIN)           CORE  76
C        XS                  A GENERAL PURPOSE VARIABLE USED FOR                   CORE  77
C                               PERPENDICULAR DISTANCE FROM SHORE LINE,            CORE  78
C                               PERPENDICULAR DISTANCE FORM EDGE OF PRIMARY CELL,  CORE  79
C                               ATTENUATION CONSTANT                               CORE  80
C                                                                                  CORE  81
C*****************************************************************************CORE  82
C                                                                                  CORE  83
 1    FORMAT(4F10.5,I10)                                                           CORE  84
 2    FORMAT(//45X24HLOCAL CIRCULATION NUMBER16/50X12HSEA BREEZE 1/ )              CORE  85
 3    FORMAT([illegible] NUMBER OF [illegible] TOTAL                               CORE  86
     1EXT[illegible] AVG. GROUND/[illegible] PARAMETER[illegible] HARMONICCORE  87
     2[illegible] SEA BREEZE CELL[illegible] TEMPERATURE/10X[illegible] 10X11H(1/SECORE  88
     3[illegible] (METERS)[illegible] (DEGREES KELVIN)/ 10XE14.5,F10.5,I1X,       CORE  89
     4[illegible] ORIENTATION[illegible] SHORELINE ANGLE[illegible]CORE  90
     5[illegible] INITIAL TEMP.[illegible] PHASE  [illegible] TEMPERATURE  /6X8HCONSTANTCORE  91
     6[illegible] NORTH[illegible] GRADIENT[illegible]                             CORE  92
     7[illegible] DIFFERENTIAL  [illegible] (METERS/SEC.))[illegible](RADIANS)CORE  93
     8[illegible] (DEGREES KELVIN/METER),[illegible](RADIANS),[illegible](DEGREES KELVIN)/, CORE  94
     9[illegible] TAUX(N) ,[illegible] DELTA(N),////)                               CORE  95
 4    FORMAT(2F10.5)                                                               CORE  96
 20   FORMAT(//45X,35HCOORDINATES OF LOCAL CELL BOUNDARIES/,50X,5HNORTH,CORE  97
     1[illegible]///)                                                               CORE  98
 21   FORMAT(//25X74HTHE NTH HARMONIC CONSTANT COEFFICIENTS USED IN CALCORE  99
     1CULATING THE WIND FIELD//)                                                   CORE 100
 22   FORMAT(5X,6H SHON(,I1,2H)=,E10.5,  5X,6H  AO2(,I1,2H)=,E10.5,                CORE 101
     1          5X,6H  AOX(,I1,2H)=,E10.5,  5X,6H  AOT(,I1,2H)=,E10.5/             CORE 102
     2          5X,6H ANN1(,I1,2H)=,E10.5,  5X,6H ALN1(,I1,2H)=,E10.5,             CORE 103
     3          5X,6H ANN2(,I1,2H)=,E10.5,  5X,6H ALN2(,I1,2H)=,E10.5/             CORE 104
     4          5X,6HSLON1(,I1,2H)=,E10.5,  5X,6HSLON2(,I1,2H)=,E10.5,             CORE 105
     5          5X,6H ANN1(,I1,2H)=,E10.5,  5X,6H ANN2(,I1,2H)=,E10.5/             CORE 106
     6          5X,6H PHI (,I1,2H)=,E10.5,  5X,6H  ENO(,I1,2H)=,E10.5,             CORE 107
     7          5X,6HDELTA(,I1,2H)=,E10.5,  5X,6H TAUX(,I1,2H)=,E10.5///)          CORE 108
C                                                                                  CORE 109
C*****************************************************************************CORE 110
C                                                                                  CORE 111
      DATA PROGR/6H[illegible]/                                                    CORE 112
C                                                                                  CORE 113
C*****************************************************************************CORE 114
C*****************************************************************************CORE 115
C                                                                                  CORE 116
      IF(J)100,101,102                                                             CORE 117
 101  IERROR=-101                                                                  CORE 118
```

```
 7734 CALL ERROR(PROGRM,IRROR,ISOUT)
C 100 THIS IS THE DATA READING ROUTE
C
 100  READ (ISIN,1)SNPHI,SGMA,ELX,THET
      READ (ISIN,1)WW,AKY,B,GRAD,NN
      READ (ISIN,4)(DELTA(N),TAUX(N),N=1,NN)
      WRITE (ISOUT,2)K
      ALM=6.2831853/ELX
      ALPH=9.8/THET
      COR=(14.5444100E-05)*SNPHI
      DO 2000 N=1,NN
      OMGX(N)=N
      OMGX(N)=OMGX(N)*7.2722052E-05
      A2=SGMA*SGMA+OMGX(N)*OMGX(N)
      A1=SQRT(A2)
      IF(SGMA)1003,1003,1004
C1003 SGMA IS ZERO OR NEGATIVE.  THIS IS NOT ALLOWED
 1003 IRROR=1003
      GO TO 7734
 1004 T1=-ATAN(OMGX(N)/SGMA)
      T2=-2.0*T1
      SCC=SGMA**2-OMGX(N)**2+COR**2
      A3=SQRT(SCC**2+4.0*SGMA**2*OMGX(N)**2)
      IF(SCC)1505,1515,1555
 1505 T3=ATAN(2.0*OMGX(N)*SGMA/SCC)+ 3.1415927
      GO TO 5555
 1515 T3=1.5707963
      GO TO 5555
 1555 T3=ATAN(2.0*OMGX(N)*SGMA/SCC)
 5555 A4=(A2*ALM**2)/A3
      T4=T2-T3
      A5=(A1*ALPH*ALM**2 )/A3
      T5=-T1-T3
      IF(AKY)1006,1006,1606
C1006 AKY  IS ZERO OR NEGATIVE.  THIS IS NOT ALLOWED
 1006 IRROR=1006
      GO TO 7734
 1606 A6=OMGX(N)/AKY
      A7=SGMA/AKY
      ACOT4 = A4*COS(T4)
      ASIT4 = A4*SIN(T4)+A6
      B1=ACOT4*ACOT4-ASIT4*ASIT4-4.0*(A7*A5*COS(T5)-A6*A4*SIN(T4))
      B2=2.0*ACOT4*ASIT4-4.0*(A6*ACOT4+A5*A7*SIN(T5))
      IF(B1)1007,1077,1777
 1007 BETA=ATAN(B2/B1) + 3.1415927
      GO TO 7777
 1077 IF(B2)4077,9077,9077
 4077 BETA=4.712389
      GO TO 7777
 9077 BETA=1.5707963
      GO TO 7777
 1777 BETA=ATAN(B2/B1)
 7777 AM=BETA/2.0
      BB=(B1*B1+B2*B2)**0.25
      ACOSM=COS(AM)*BB/2.0
      ASINM=SIN(AM)*BB/2.0
      ACOT4=ACOT4/2.0
      ASIT4=ASIT4/2.0
```

```
      C1= ACOT4+ACOSM
      D1= ASIT4+ASINM
      C2=ACOT4-ACOSM
      D2= ASIT4-ASINM
      IF(C1)1008,1088,1888
 1008 GAM1=ATAN(D1/C1) + 3.1415927
       GO TO 5888
 1088 IF(D1)9008,9008,9088
 9008 GAM1=4.7123890
       GO TO 8888
 9088 GAM1=1.5707963
       GO TO 8888
 1888 GAM1=ATAN(D1/C1)
 5888 IF(C2)1009,1099,1999
 1009 GAM2=ATAN(D2/C2) + 3.1415927
      GO TO 1111
 1099 IF(D2)9009,9009,9099
 9009 GAM2=4.7123890
      GO TO 1111
 9099 GAM2=1.5707963
      GO TO 1111
 1999 GAM2=ATAN(D2/C2)
 1111 SRTE1=(C1*C1+D1*D1)**0.25
      SRTE2=(C2*C2+D2*D2)**0.25
      AN1=GAM1/2.0
      AN2=GAM2/2.0
      CON1=COS(AN1)
      CON2X(N)=COS(AN2)
      CAN1=SIN(AN1)
      CAN2X(N)=SIN(AN2)
      IF(CON1)5,5,66
 5    R1RX(N)=SRTE1*CON1
      R1IX(N)=SRTE1*CAN1
      EPS1=1.0
      GO TO 7
 66   R1RX(N)=-SRTE1*CON1
      R1IX(N)=-SRTE1*CAN1
      EPS1=-1.0
 7    IF(CON2X(N))8,8,9
 8    R2RX(N)= SRTE2*CON2X(N)
      R2IX(N)=SRTE2*CAN2X(N)
      EPS2=1.0
      GO TO 10
 9    R2RX(N)=-SRTE2*CON2X(N)
      R2IX(N)=-SRTE2*CAN2X(N)
      EPS2=-1.0
 10   IF(N-1)7730,110,151
 7730 IERROR=7730
      GO TO 7734
C DETERMINATION OF HEIGHT OF SEA BREEZE
 110  HT1=-1.0/R1RX(N)
      HT2=-1.0/R2RX(N)
C     CROHT(K) IS THE HEIGHT OF THE SEA BREEZE CELL.
      IF(HT1-HT2)11,11,12
 11   CROHT(K)=2.0*HT2
      GO TO 13
 12   CROHT(K)=2.0*HT1
 13   WRITE (I500I,20)CRMAXY(K),CRMINY(K),CRMAXX(K),CRMINX(K),CROHT(K)
```

```
      WRITE (IOUUT,3)SNPHI,SGMA, NN,ELX,THET,WW,ANT,B,GRAD
      WRITE (IOUUT,21)
  131 FAX(N)=T5-AM+TAUX(N)
      G=-COR/AI
C
C     COMPUTATIONS FOR CONSTANT COEFFICIENTS OF THE WIND FIELD
      AJZA(N)=-DELTA(N)*AD/BB
      AJXX(N)=AJZA(N)/ALM
      AJY=AJXX(N)*G
      ESQ1=EPS1*SRTE1
      ESQ2=EPS2*SRTE2
      AMG=ESQ1/ESQ2
      CNN1X(N)=COS(AN1+T1)*AMG*G
      CNN2X(N)=COS(AN2+T1)*G
      SNN1X(N)=SIN(AN1+T1)*AMG*G
      SNN2X(N)=SIN(AN2+T1)*G
      CON1X(N)=CON1*AMG
      CAN1X(N)=CAN1*AMG
      WRITE (IOUUT,22) N,OMGX(N),N,AJZA(N),N,AJXX(N),N,AJY,N,CNN1X(N),N,
     1IIA(N),N,R2RA(N),N,R2IA(N),N,ESQ1,N,ESQ2,N,AN1,N,AN2,N,FAX(N),N,T1
     2,N,DELTX(N),N,TAUX(N)
 2000 AJXX(N)=AJXX(N)*ESQ2
      ELX=ELX/2.0
C1015 COMPUTE CENTER OF SEA BREEZE CELL.
 1015 XCB=(CRMAXX(K)+CRMINX(K))/2.0
      YCB=(CRMAXY(K)+CRMINY(K))/2.0
      COSB=COS(B)
      SINB=SIN(B)
  105 RETURN
C
C     THIS IS THE COMPUTING ROUTE
C
C102  TEST FOR IMPACTED PARTICLE
  102 IF(ZP(J)) 1019,1020,1020
 1019 AX=0.0
      AY=0.0
      AZ=-1.0E+8
      GO TO 105
C
C1020 PARTICLE IS ALOFT
C     THIS CARD ROTATES AND TRANSLATES THE MACRO SYSTEM COORDINATES
C     INTO THE SEA BREEZE CELL COORDINATES.
 1020 XS=(XP(J)-XCB)*COSB-(YP(J)-YCB)*SINB
      ARG=ALM*XS
C     THE FOLLOWING CARDS ATTENUATE THE WIND FIELD IN THE ADJACENT
C     REGION.
      XS=ELX-ABS(XS)
      IF(XS)130,103 ,103
  130 XS=XS*WW
      GO TO 1030
  103 XS=0.0
 1030 SB=0.0
      AY=0.0
      AZ=0.0
      DO 3000 N=1,NN
C     THE FOLLOWING TEN CARDS COMPUTE WIND FIELD COEFFICIENTS.
      ATTN1=EXP(XS+R1RX(N)*ZP(J))
      ATTN2=EXP(XS+R2RX(N)*ZP(J))
```

```
      AAGG=OMGX(N)*TP(J)+FAX(N)
      ARW1=R1IX(N)*ZP(J)+AAGG
      ARW2=R2IX(N)*ZP(J)+AAGG
      SARW1=SIN(ARW1)*ATTN1
      SARW2=SIN(ARW2)*ATTN2
      CARW1=COS(ARW1)*ATTN1
      CARW2=COS(ARW2)*ATTN2
      AAGG=AJXX(N)*COS(ARG)
C     THE FOLLOWING FIVE CARDS ARE THE WIND FIELD
      AZ=AJZX(N)*SIN(ARG)*(CARW1-CARW2) +AZ
      BB=AAGG*(CARW1*CON1X(N)-SARW1*CAN1X(N)-CARW2*CON2X(N)+SARW2*
     1CAN2X(N)) +BB
 3000 AY=AAGG*   (CARW1*CNN1X(N)-SARW1*SNN1X(N)-CARW2*CNN2X(N)+SARW2*
     1SNN2X(N)) +AY
C15   DEROTATION OF THE WIND VECTORS INTO THE MACROSYSTEM
 15   AX=BB*COSB+AY*SINB
      AY=-BB*SINB+AY*COSB
      GO TO 105
      END
```

312*

312 *

```
$IBFTC HEIGH    LIST,DECK,M94/2
      SUBROUTINE HEIGHT(XX,YY,H)
C     11 OCT 66
C     THIS SUBROUTINE RETRIEVES TOPO HEIGHT AT HORIZONTAL COORDINATES
C     XX,YY AND PUTS IT IN H
C
C ***************************************************************************
C
      COMMON /SET1/
     1          DIAM    , DETID   , IRISE   , IEXEC   , ISIN    , IOUST   ,
     2          SD      , SPAR    , SSAM    , TML     , TMP1    , TMP2    ,
     3          T2M     , U       , VPR     , W       , X       , Z       ,
     4          WHY     , RMIN    , IDISTR  , SPAR1   , SPAR2   , SPAR3   ,
     5          SPAR4   , SPAR5   , SPAR6   , SPAR7   , SPAR8   , SPAR9
      DIMENSION  DETID(12),WHY(40)
C
C ***************************************************************************
C
      COMMON /SET2/
     1          G       , SUBSID  , GRINT   , BXLL    , BXLU    , BYLL    ,
     2,         BYLU    , TXLL    , TXLU    , TYLL    , TYLU    , XGZ     ,
     3,         YGZ     , NBLCK   , HTOPO   , ITOPO   , ILIM    , JLIM    ,
     4,         KLIM    , II      , JJ      , KK      , XP      , YP      ,
     5,         ZP      , FMAS    , TP      , PS      , VX      , VY      ,
     6,         VZ      , IL      , JL      , IGADD   , WGRINT  , NSTRAT  ,
     7,         WLLX    , WLLY    , WORX    , WORY    , BOTHIT  , IPARIN  ,
     8,         IOTOPO  , IOWIND  , ITTOPO  , IPOUT   , IPAROT  , JTOP1   ,
     9,         JWIND1  , IRROR   , ILIMIT  , ENDTIM  , IC      , ISYPAS  ,
     1,         JTOPJ   , NLOST   , NG      , NTU     , NTI     , NW      ,
     2,         NALOFT  , JTIME1  , NBMAX   , NFREE   , N       , NCL     ,
     3,         CRMAXY  , CRUHT   , NCRTYP  , SL      , CRMINX  , CRMINY  ,
     4,         OU      , SN      , CS      , NLOCIR  , DTLOC   , ATEMP   ,
     5,         RHO     , NA      , TGZ     , DTMAC   , FROG    , CRMAXX  ,
     6,         ROPART
      DIMENSION TOPOLM(4,4)   ,WINTAR(4)       ,ITOPLM(3,4)
      DIMENSION S(10,10)      ,SUBSID(400)     ,IC(18)
      DIMENSION XP(200)       ,YP(200)         ,ZP(200)        ,FMAS(200)
      DIMENSION TP(200)       ,PS(200)         ,ATEMP(200)     ,RHO(200)
      DIMENSION VX(1500)      ,VY(1500)        ,VZ(1500)       ,IL(70)
      DIMENSION JL(70)        ,IGADD(70)       ,WORX(70)
      DIMENSION WGRINT(70)                     ,WLLX(70)       ,WLLY(70)
      DIMENSION WORY(70)      ,BOTHIT(70)      ,SN(6)          ,CS(6)
      DIMENSION CRMINX(6)     ,CRMAXX(6)       ,CRMINY(6)      ,CRMAXY(6)
      DIMENSION CRUHT(6)      ,NCRTYP(6)       ,OU(6)
C
C ***************************************************************************
C
C     PRESERVE STD GRID INTERVAL
      GRIN=GRINT
C     IS PARTICLE OVER BLOCK OF TOPOGRAPHY NOW IN CORE
      IF(XX-BXLL)4,2,1
    1 IF(XX-BXLU)2,4,4
    2 IF(YY-BYLL)4,11,3
    3 IF(YY-BYLU)11,4,4
C     IS PARTICLE OVER RANGE OF TOPOGRAPHY UNDER STUDY
    4 IF(XX-TXLL)10,6,5
    5 IF(XX-TXLU)6,10,10
    6 IF(YY-TYLL)10,9,7
    7 IF(YY-TYLU)9,10,10
    8 FORMAT (2F12.4)
C     PARTICLE IS BEYOND IN-CORE TOPO SPECIFICATION BUT WITHIN THE TOPO
```

```
C     SPECIFICATION AREA. H IS SET   10000.0
    9 H=-10000.0
      GO TO 101
C     PARTICLE IS BEYOND TOPO SPECIFICATION AREA. H IS SET -20000.0
   10 H=-20000.0
C******** [illegible] TEMP [illegible] ************************************
  101 WRITE(IOUT,8)XX,YY
      GO TO 28
C     FOR PARTICLE OVER TOPO BLOCK NOW IN CORE,FIND COORD OF NEAREST
C     TOPO POINT IN 2 DIM ARRAY (IN NO OF GRINTS FROM LOWER LEFT COR OF
C     BLOCK IN CORE)
   11 I=(XX-BXLL)/GRINT+1.0
      J=(YY-BYLL)/GRINT+1.0
C     H IS TOPO HEIGHT IF SQUARE OF SIDE GRINT TO UPPER RT OF PT(I,J)
C     NOT SUBDIVIDED,A NEG NO IF SUBDIV
      H=S(I,J)
      IF(H)12,28,28
C     SHIFT ORIGIN OF PARTICLE COORD TO LOWER LT OF SQUARE(SIDE,GRINT)
C     UNDER PARTICLE
   12 CI=I
      CJ=J
      CX=XX-BXLL-(CI-1.0)*GRIN
      CY=YY-BYLL-(CJ-1.0)*GRIN
C     DIV SQ GRINT INTO 4 QUADRANTS
      GRIN=GRIN/2.0
C     SHIFT ORIGIN OF PARTICLE COORD TO CENTER OF DIVIDED SQ
      CCX=CX-GRIN
      CCY=CY-GRIN
C     (LOOP STARTS HERE)
C     CONVERT NEG H TO BASE INDEX FOR SUBSID(K)
   13 K=-H+0.5
C     WHICH QUADRANT IS PARTICLE OVER
      IF(CCX)14,15,15
   14 IF(CCY)19,16,16
   15 IF(CCY)17,18,18
C     MODIFY BASE INDEX,REMEMBER QUADRANT
   16 K=K+1
      N=2
      GO TO 20
   17 K=K+3
      N=4
      GO TO 20
   18 K=K+2
      N=3
      GO TO 20
   19 N=1
C     H IS TOPO HT IF QUADRANT NOT SUBDIV IN SUBSID(K),A NEG NO IF DIV
   20 H=SUBSID(K)
      IF(H)21,28,28
C     SUBDIVIDE QUADRANT
   21 GRIN=GRIN/2.0
C     SHIFT ORIGIN OF PARTICLE COORD TO CENTER OF SUBDIVIDED QUADRANT
C     PARTICLE IS OVER
      GO TO(22,24,25,27),N
   22 CCY=CCY+GRIN
   23 CCX=CCX+GRIN
      GO TO 13
   24 CCY=CCY-GRIN
```

```
      GO TO 28                                                          METG 119
   25 CCY=CCY-GRIN                                                      METG 120
   26 CCX=CCX-GRIN                                                      METG 121
      GO TO 13                                                          METG 122
   27 CCY=CCY+GRIN                                                      METG 123
      GO TO 26                                                          METG 124
   28 RETURN                                                            METG 125
      END                                                               METG 126
                                                                            127*

                                                                            127 *
```

SAMPLE TEST PROBLEM AND PRINTOUT

The sample printout that follows contains essentially all of the information necessary to reconstruct the inputs that define the atmosphere and wind-field structure. The output has already been described in detail in Table 13.

* * * * * * * * * * *

THE DEPARTMENT OF DEFENSE FALLOUT PREDICTION SYSTEM

* * * * * * * * * * *

TRANSPORT MODULE

PREPARED BY
TECHNICAL OPERATIONS RESEARCH,INC.
BURLINGTON, MASS.

**** SUMMARY OF INPUT IDENTIFIERS AND INITIAL CONDITIONS ****

**** INITIAL CONDITIONS (FIREBALL) IDENTIFICATION ****
FOURTH LARGE SCALE TEST OF THE DELFIC SYSTEM, 15 NOV. 1966, INIT. COND.

**** CLOUD RISE IDENTIFICATION ****
FOURTH LARGE SCALE TEST OF THE DELFIC SYSTEM, 15 NOV. 1966, CLOUD RISE

**** PARTICLE SET EXPANSION IDENTIFICATION ****
FOURTH LARGE SCALE TEST OF THE DELFIC SYSTEM, 15 NOV. 1966, PSE

**** THIS RUN OF THE TRANSPORT MODULE WAS GIVEN THE FOLLOWING IDENTIFICATION ****
FOURTH LARGE SCALE TEST OF THE DELFIC SYSTEM, 15 NOV. 1966, TRANSPORT

**** OTHER INPUT DATA ****

THE CONTROL VARIABLE ARRAY, IC(J), HAS BEEN GIVEN THE FOLLOWING VALUES.
1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0

THE TRANSPORT TIME LIMIT IS 86400.000

PARTICLE DATA
DENSITY OF FALLOUT PARTICLES 2600.000 KG/M**3
IPARIN 1 0.10000E+07 0.10000E+07 0.00000E-38 0.33729E+03 0

TOPOGRAPHIC DATA
IN THIS RUN WE ASSUME A PLANAR DEPOSITION SURFACE AT ELEVATION 938.174

WIND DATA
THIS WIND FIELD USES THE FRENCHMAN FLATS AND ROAD 8 STATIONS. 12/20/66

Preceding page blank

ATMOSPHERIC PROPERTIES FOR FALL RATE CALCULATION

HEIGHT OF BOTTOM OF STRATUM METERS ABOVE MSL	VISCOSITY (MKS)	DENSITY (MKS)
-0.11000E+04	0.18206E-04	0.13470E+01
-0.90000E+03	0.18144E-04	0.13219E+01
-0.70000E+03	0.18082E-04	0.12972E+01
-0.50000E+03	0.18019E-04	0.12728E+01
-0.30000E+03	0.17957E-04	0.12487E+01
-0.10000E+03	0.18514E-04	0.11592E+01
0.10000E+03	0.18443E-04	0.11466E+01
0.30000E+03	0.18372E-04	0.11220E+01
0.50000E+03	0.18301E-04	0.11034E+01
0.70000E+03	0.18230E-04	0.10847E+01
0.90000E+03	0.18624E-04	0.10279E+01
0.11000E+04	0.18452E-04	0.10216E+01
0.13000E+04	0.18373E-04	0.10064E+01
0.15000E+04	0.18283E-04	0.99005E+00
0.17000E+04	0.18180E-04	0.97250E+00
0.19000E+04	0.18090E-04	0.95639E+00
0.21000E+04	0.18000E-04	0.94029E+00
0.23000E+04	0.17905E-04	0.92373E+00
0.25000E+04	0.17837E-04	0.91509E+00
0.27000E+04	0.17714E-04	0.89403E+00
0.29000E+04	0.17616E-04	0.87841E+00
0.31000E+04	0.17527E-04	0.86372E+00
0.33000E+04	0.17426E-04	0.84615E+00
0.35000E+04	0.17329E-04	0.83352E+00
0.37000E+04	0.17256E-04	0.82046E+00
0.39000E+04	0.17169E-04	0.80210E+00
0.41000E+04	0.17094E-04	0.78804E+00
0.43000E+04	0.17008E-04	0.77388E+00
0.45000E+04	0.16925E-04	0.75955E+00
0.47000E+04	0.16891E-04	0.74462E+00
0.49000E+04	0.16820E-04	0.72667E+00
0.51000E+04	0.16766E-04	0.71151E+00
0.53000E+04	0.16723E-04	0.69477E+00
0.55000E+04	0.16682E-04	0.67882E+00
0.57000E+04	0.16641E-04	0.66392E+00
0.59000E+04	0.16558E-04	0.65062E+00
0.61000E+04	0.16471E-04	0.63732E+00
0.63000E+04	0.16380E-04	0.62492E+00
0.65000E+04	0.16290E-04	0.61253E+00
0.67000E+04	0.16200E-04	0.60013E+00
0.69000E+04	0.16109E-04	0.58773E+00
0.71000E+04	0.16039E-04	0.57613E+00
0.73000E+04	0.15969E-04	0.56457E+00
0.75000E+04	0.15900E-04	0.55301E+00
0.77000E+04	0.15820E-04	0.54188E+00
0.79000E+04	0.15739E-04	0.53079E+00
0.81000E+04	0.15658E-04	0.51970E+00
0.83000E+04	0.15577E-04	0.50861E+00
0.85000E+04	0.15496E-04	0.49752E+00
0.87000E+04	0.15415E-04	0.48643E+00
0.89000E+04	0.15334E-04	0.47535E+00
0.91000E+04	0.15255E-04	0.46455E+00
0.93000E+04	0.15180E-04	0.45448E+00
0.95000E+04	0.15105E-04	0.44442E+00

0.97000E+04	0.15029E-04	0.43436E+00
0.99000E+04	0.14954E-04	0.42430E+00
0.10100E+05	0.14879E-04	0.41424E+00
0.10300E+05	0.14804E-04	0.40418E+00
0.10500E+05	0.14729E-04	0.39412E+00
0.10700E+05	0.14656E-04	0.38443E+00
0.10900E+05	0.14587E-04	0.37444E+00
0.11100E+05	0.14520E-04	0.36545E+00
0.11300E+05	0.14461E-04	0.35635E+00
0.11500E+05	0.14414E-04	0.34773E+00
0.11700E+05	0.14368E-04	0.33911E+00
0.11900E+05	0.14322E-04	0.33046E+00
0.12100E+05	0.14275E-04	0.32186E+00
0.12300E+05	0.14229E-04	0.31324E+00
0.12500E+05	0.14183E-04	0.30462E+00
0.12700E+05	0.14136E-04	0.29600E+00
0.12900E+05	0.14090E-04	0.28738E+00
0.13100E+05	0.14044E-04	0.27875E+00
0.13300E+05	0.14060E-04	0.26956E+00
0.13500E+05	0.14087E-04	0.26027E+00
0.13700E+05	0.14091E-04	0.25204E+00
0.13900E+05	0.14053E-04	0.24528E+00
0.14100E+05	0.14035E-04	0.23852E+00
0.14300E+05	0.14006E-04	0.23176E+00
0.14500E+05	0.13976E-04	0.22500E+00
0.14700E+05	0.13950E-04	0.21825E+00
0.14900E+05	0.13921E-04	0.21149E+00
0.15100E+05	0.13893E-04	0.20473E+00
0.15300E+05	0.13905E-04	0.19846E+00
0.15500E+05	0.13926E-04	0.19230E+00
0.15700E+05	0.13948E-04	0.18615E+00
0.15900E+05	0.13970E-04	0.18000E+00
0.16100E+05	0.13992E-04	0.17385E+00
0.16300E+05	0.14014E-04	0.16770E+00
0.16500E+05	0.14036E-04	0.16155E+00
0.16700E+05	0.14053E-04	0.15574E+00
0.16900E+05	0.14055E-04	0.15146E+00
0.17100E+05	0.14052E-04	0.14719E+00
0.17300E+05	0.14051E-04	0.14292E+00
0.17500E+05	0.14055E-04	0.13865E+00
0.17700E+05	0.14050E-04	0.13438E+00
0.17900E+05	0.14049E-04	0.13011E+00
0.18100E+05	0.14048E-04	0.12584E+00
0.18300E+05	0.14057E-04	0.12177E+00
0.18500E+05	0.14100E-04	0.11787E+00
0.18700E+05	0.14133E-04	0.11396E+00
0.18900E+05	0.14156E-04	0.11006E+00
0.19100E+05	0.14200E-04	0.10615E+00
0.19300E+05	0.14233E-04	0.10225E+00
0.19500E+05	0.14266E-04	0.98341E-01
0.19700E+05	0.14299E-04	0.94436E-01
0.19900E+05	0.14314E-04	0.91675E-01
0.20100E+05	0.13970E-04	0.93182E-01
0.20300E+05	0.13994E-04	0.90184E-01
0.20500E+05	0.14018E-04	0.87185E-01
0.20700E+05	0.14042E-04	0.84186E-01
0.20900E+05	0.14067E-04	0.81187E-01
0.21100E+05	0.14089E-04	0.78690E-01
0.21300E+05	0.14111E-04	0.76193E-01
0.21500E+05	0.14133E-04	0.73695E-01

0.21700E+05	0.14155E-04	0.71198E-01
0.21900E+05	0.14177E-04	0.68701E-01
0.22100E+05	0.14199E-04	0.66607E-01
0.22300E+05	0.14221E-04	0.64512E-01
0.22500E+05	0.14242E-04	0.62418E-01
0.22700E+05	0.14264E-04	0.60324E-01
0.22900E+05	0.14286E-04	0.58229E-01
0.23100E+05	0.14308E-04	0.56471E-01
0.23300E+05	0.14329E-04	0.54712E-01
0.23500E+05	0.14351E-04	0.52954E-01
0.23700E+05	0.14373E-04	0.51195E-01
0.23900E+05	0.14394E-04	0.49436E-01
0.24100E+05	0.14416E-04	0.47955E-01
0.24300E+05	0.14438E-04	0.46475E-01
0.24500E+05	0.14459E-04	0.44994E-01
0.24700E+05	0.14481E-04	0.43513E-01
0.24900E+05	0.14502E-04	0.42032E-01
0.25100E+05	0.14524E-04	0.40785E-01
0.25300E+05	0.14545E-04	0.39537E-01
0.25500E+05	0.14567E-04	0.38290E-01
0.25700E+05	0.14588E-04	0.37042E-01
0.25900E+05	0.14610E-04	0.35795E-01
0.26100E+05	0.14631E-04	0.34740E-01
0.26300E+05	0.14653E-04	0.33686E-01
0.26500E+05	0.14674E-04	0.32631E-01
0.26700E+05	0.14696E-04	0.31576E-01
0.26900E+05	0.14717E-04	0.30521E-01
0.27100E+05	0.14738E-04	0.29630E-01
0.27300E+05	0.14760E-04	0.28739E-01
0.27500E+05	0.14781E-04	0.27848E-01
0.27700E+05	0.14802E-04	0.26956E-01
0.27900E+05	0.14824E-04	0.26065E-01
0.28100E+05	0.14845E-04	0.25311E-01
0.28300E+05	0.14866E-04	0.24556E-01
0.28500E+05	0.14887E-04	0.23802E-01
0.28700E+05	0.14909E-04	0.23048E-01
0.28900E+05	0.14930E-04	0.22293E-01
0.29100E+05	0.14951E-04	0.21653E-01
0.29300E+05	0.14972E-04	0.21012E-01
0.29500E+05	0.14993E-04	0.20371E-01
0.29700E+05	0.15015E-04	0.19731E-01
0.29900E+05	0.15036E-04	0.19090E-01
0.30100E+05	0.15057E-04	0.18546E-01
0.30300E+05	0.15078E-04	0.18002E-01
0.30500E+05	0.15099E-04	0.17459E-01
0.30700E+05	0.15120E-04	0.16915E-01
0.30900E+05	0.15141E-04	0.16371E-01
0.31100E+05	0.15162E-04	0.15908E-01
0.31300E+05	0.15183E-04	0.15446E-01
0.31500E+05	0.15204E-04	0.14983E-01
0.31700E+05	0.15225E-04	0.14520E-01
0.31900E+05	0.15246E-04	0.14058E-01
0.32100E+05	0.15270E-04	0.13661E-01
0.32300E+05	0.15295E-04	0.13264E-01
0.32500E+05	0.15319E-04	0.12867E-01
0.32700E+05	0.15343E-04	0.12470E-01
0.32900E+05	0.15367E-04	0.12073E-01
0.33100E+05	0.15392E-04	0.11735E-01
0.33300E+05	0.15417E-04	0.11396E-01
0.33500E+05	0.15442E-04	0.11058E-01

0.33700E+05	0.15467E-04	0.10719E-01
0.33900E+05	0.15492E-04	0.10381E-01
0.34100E+05	0.15517E-04	0.10093E-01
0.34300E+05	0.15542E-04	0.98043E-02
0.34500E+05	0.15566E-04	0.95159E-02
0.34700E+05	0.15591E-04	0.92276E-02
0.34900E+05	0.15616E-04	0.89392E-02
0.35100E+05	0.15641E-04	0.86931E-02
0.35300E+05	0.15666E-04	0.84471E-02
0.35500E+05	0.15690E-04	0.82010E-02
0.35700E+05	0.15715E-04	0.79549E-02
0.35900E+05	0.15740E-04	0.77089E-02
0.36100E+05	0.15764E-04	0.74988E-02
0.36300E+05	0.15789E-04	0.72887E-02
0.36500E+05	0.15814E-04	0.70785E-02
0.36700E+05	0.15838E-04	0.68684E-02
0.36900E+05	0.15863E-04	0.66583E-02
0.37100E+05	0.15887E-04	0.64785E-02
0.37300E+05	0.15912E-04	0.62986E-02
0.37500E+05	0.15936E-04	0.61187E-02
0.37700E+05	0.15961E-04	0.59388E-02
0.37900E+05	0.15985E-04	0.57590E-02
0.38100E+05	0.16010E-04	0.56048E-02
0.38300E+05	0.16034E-04	0.54506E-02
0.38500E+05	0.16058E-04	0.52963E-02
0.38700E+05	0.16083E-04	0.51421E-02
0.38900E+05	0.16107E-04	0.49879E-02
0.39100E+05	0.16131E-04	0.48557E-02
0.39300E+05	0.16155E-04	0.47234E-02
0.39500E+05	0.16180E-04	0.45911E-02
0.39700E+05	0.16204E-04	0.44588E-02
0.39900E+05	0.16228E-04	0.43265E-02
0.40100E+05	0.16253E-04	0.42128E-02
0.40300E+05	0.16277E-04	0.40990E-02
0.40500E+05	0.16301E-04	0.39853E-02
0.40700E+05	0.16325E-04	0.38715E-02
0.40900E+05	0.16349E-04	0.37578E-02
0.41100E+05	0.16373E-04	0.36598E-02
0.41300E+05	0.16397E-04	0.35619E-02
0.41500E+05	0.16421E-04	0.34639E-02
0.41700E+05	0.16445E-04	0.33660E-02
0.41900E+05	0.16469E-04	0.32680E-02
0.42100E+05	0.16493E-04	0.31837E-02
0.42300E+05	0.16517E-04	0.30993E-02
0.42500E+05	0.16541E-04	0.30149E-02
0.42700E+05	0.16565E-04	0.29305E-02
0.42900E+05	0.16589E-04	0.28461E-02
0.43100E+05	0.16613E-04	0.27732E-02
0.43300E+05	0.16637E-04	0.27004E-02
0.43500E+05	0.16661E-04	0.26275E-02
0.43700E+05	0.16684E-04	0.25546E-02
0.43900E+05	0.16708E-04	0.24817E-02
0.44100E+05	0.16732E-04	0.24187E-02
0.44300E+05	0.16756E-04	0.23557E-02
0.44500E+05	0.16779E-04	0.22927E-02
0.44700E+05	0.16803E-04	0.22297E-02
0.44900E+05	0.16827E-04	0.21667E-02
0.45100E+05	0.16850E-04	0.21122E-02
0.45300E+05	0.16874E-04	0.20576E-02
0.45500E+05	0.16898E-04	0.20031E-02

0.45700E+05	0.16921E-04	0.19486E-02
0.45900E+05	0.16945E-04	0.18941E-02
0.46100E+05	0.16968E-04	0.18468E-02
0.46300E+05	0.16992E-04	0.17995E-02
0.46500E+05	0.17016E-04	0.17522E-02
0.46700E+05	0.17039E-04	0.17049E-02
0.46900E+05	0.17063E-04	0.16576E-02
0.47100E+05	0.17072E-04	0.16180E-02
0.47300E+05	0.17082E-04	0.15784E-02
0.47500E+05	0.17092E-04	0.15389E-02
0.47700E+05	0.17102E-04	0.14993E-02
0.47900E+05	0.17111E-04	0.14597E-02
0.48100E+05	0.17111E-04	0.14258E-02
0.48300E+05	0.17111E-04	0.13919E-02
0.48500E+05	0.17111E-04	0.13580E-02
0.48700E+05	0.17111E-04	0.13240E-02
0.48900E+05	0.17111E-04	0.12901E-02
0.49100E+05	0.17111E-04	0.12602E-02
0.49300E+05	0.17111E-04	0.12302E-02
0.49500E+05	0.17111E-04	0.12003E-02
0.49700E+05	0.17111E-04	0.11703E-02
0.49900E+05	0.17111E-04	0.11403E-02

ENTERING LINK 6

UNDRFLOW AT 54221 IN MU

UNDRFLOW AT 54222 IN MU

UNDRFLOW AT 54222 IN MU

UNDRFLOW AT 54221 IN MU

THE FOLLOWING WIND VECTORS HAVE BEEN USED TO DEFINE AN ATMOSPHERE UP TO TIME 86400.

VECTOR HEIGHT	HORIZONTAL COORDINATES XS	YS	VECTOR COMPONENTS VX	VY	VZ
975.360	993804.000	999589.641	0.000	0.000	0.000
1219.200	993804.000	999589.641	0.000	0.000	0.000
1524.000	993804.000	999589.641	0.000	0.000	0.000
1828.800	993804.000	999589.641	0.000	0.000	0.000
2133.600	993804.000	999589.641	0.860	-0.155	0.000
2438.400	993804.000	999589.641	2.940	1.070	0.000
2743.200	993804.000	999589.641	5.420	3.129	0.000
3048.000	993804.000	999589.641	6.966	4.023	0.000
3657.600	993804.000	999589.641	3.484	2.011	0.000
4267.200	993804.000	999589.641	3.484	2.011	0.000
4876.800	993804.000	999589.641	3.484	2.011	0.000
5486.400	993804.000	999589.641	7.043	-1.242	0.000
6096.000	993804.000	999589.641	12.766	2.251	0.000
1219.200	1026389.000	1019976.125	0.000	0.000	0.000
1524.000	1026389.000	1019976.125	0.000	0.000	0.000
1828.800	1026389.000	1019976.125	0.000	0.000	0.000
2133.600	1026389.000	1019976.125	-1.771	0.249	0.000
2438.400	1026389.000	1019976.125	-0.981	0.915	0.000
2743.200	1026389.000	1019976.125	1.447	1.051	0.000
3048.000	1026389.000	1019975.125	3.484	0.804	0.000
3352.800	1026389.000	1019976.125	5.312	0.747	0.000
3657.600	1026389.000	1019976.125	7.082	0.995	0.000

3962.400	1026389.000	1019976.125	7.484	1.320	0.000
4267.200	1026389.000	1019976.125	5.461	1.987	0.000
4572.000	1026389.000	1019976.125	3.570	2.690	0.000
4876.800	1026389.000	1019976.125	3.215	3.105	0.000
5181.600	1026389.000	1019976.125	3.675	3.027	0.000
5486.400	1026389.000	1019976.125	6.234	0.545	0.000
5791.200	1026389.000	1019976.125	8.241	-2.055	0.000
6096.000	1026389.000	1019976.125	9.464	-4.017	0.000

REQUESTED GRID ARRANGEMENT

HEIGHT	INTERVAL	LIMITS WLLX	WLLY	WURX	WURY
914.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
1066.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
1219.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
1371.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
1524.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
1676.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
1828.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
1981.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
2133.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
2286.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
2438.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
2590.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
2743.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
2895.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
3048.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
3200.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
3352.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
3505.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
3657.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
3810.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
3962.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
4114.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
4267.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
4419.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
4572.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
4724.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
4876.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
5029.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
5181.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
5334.000	4500.000	997750.000	997750.000	1029200.000	1024700.000
5486.400	4500.000	997750.000	997750.000	1029200.000	1024700.000
5638.800	4500.000	997750.000	997750.000	1029200.000	1024700.000
5791.200	4500.000	997750.000	997750.000	1029200.000	1024700.000
5943.600	4500.000	997750.000	997750.000	1029200.000	1024700.000
6096.000	4500.000	997750.000	997750.000	1029200.000	1024700.000

COMPUTATION METHOD 1 WAS USED ON THE 4 NEAREST DATA POINTS.

ALPHA = 457.200METERS. BETA = 32000.000METERS.

WIND COMPONENTS

LEVEL 1 BASE AT 914.400 METERS

EAST-WEST ROW 1

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 2

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 3

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 4

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 5

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 6

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

LEVEL 2 BASE AT 1066.800 METERS

EAST-WEST ROW 1

0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-W ROW 2

0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-W ROW 3

0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-W ROW 4

0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-W ROW 5

0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0 00	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 6

0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

LEVEL 35 BASE AT 6090.000 METERS

EAST-WEST ROW 1

12.76600	12.40726	11.55457	11.25178	10.60150	9.88668	9.26819
-2.25103	-2.39996	-2.62927	-2.87952	-3.14944	-3.44614	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 2

12.56652	12.01592	11.44111	10.85199	10.26402	9.67492	9.26819
-2.33381	-2.55317	-2.80093	-3.04540	-3.28952	-3.53404	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 3

12.25293	11.65563	11.06309	10.50159	9.97319	9.47676	9.26818
-2.46395	-2.71134	-2.95576	-3.19090	-3.41323	-3.61629	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 4

11.94543	11.39031	10.70839	10.17429	9.70388	9.27948	9.26819
-2.59111	-2.85917	-3.10597	-3.32676	-3.52202	-3.69818	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 5

11.62454	10.92671	10.34295	9.84942	9.43366	9.26819	9.26819
-2.72477	-3.01362	-3.25675	-3.46160	-3.63418	-3.70286	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

EAST-WEST ROW 6

11.25204	10.50136	9.93556	9.49012	9.26819	9.26819	9.26819
-2.87941	-3.14069	-3.42581	-3.51074	-3.70286	-3.70286	-3.70286
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

ENTERING LINK 7

DAVIES EQUATIONS ARE INACCURATE FOR 12473.701 MICRONS AT 941.347 METERS

DAVIES EQUATIONS ARE INACCURATE FOR 12473.701 MICRONS AT 942.059 METERS

DAVIES EQUATIONS ARE INACCURATE FOR 12473.701 MICRONS AT 943.126 METERS

DAVIES EQUATIONS ARE INACCURATE FOR 12473.701 MICRONS AT 944.703 METERS

```
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
150   2   50   0    0   0   0   0
150   2    0   0   50   0   0   0
ENTERING LINK 5
142   2   58   0    0   0   0   0

TRANSPORT IS COMPLETED. INTERMEDIATE RESJLTS ARE ON TAPE  9
ENTERING LINK 8
```

APPENDIX A

THEORY OF OROGRAPHIC FLOW WITH APPLICATION TO TROPOSPHERIC FALLOUT

Introduction

It has been recognized for some time that terrain effects influence the ultimate distribution of local (less than 160 km from blast area) radioactivity resulting from a tropospheric nuclear explosion. The vertical lifting of light debris over mountains can extend the fallout range beyond the usual expectations, while gradual but extended depressions will shorten it. The need to develop a mathematical model for flow over variable terrain, which can be rendered compatible with such systems, arises principally from the lack of sufficient meteorological data at this time to yield a satisfactory time and space dependent picture of the wind field over short distances. Although sounding stations at 14-mi intervals are planned in the near future (Army Integrated Meteorological System), it is questionable whether even this will be sufficient to account for local variations of the wind field. The model of variable terrain flow developed in this investigation is conceived for the purpose of enabling one to predict the wind field in regions where meteorological data are not usually available.

Our model is based upon a perturbation treatment of the usual hydrodynamic-thermodynamic equations assuming an adiabatic atmosphere, and is predicated on the assumption of the existence of a uniform, steady velocity field, $\underset{\sim}{u}_0$, which would otherwise exist in the absence of the ground disturbance. The relationship between the change in the wind field $\Delta \underset{\sim}{v}(x, y, z)$ and the curvature of the terrain is deduced by first deriving the dispersion relationship for the system (which connects the vertical attenuation constant of the velocity field to the periodicity of the ground structure) and subsequently applying the boundary condition that the surface wind trajectory be parallel to the terrain. The resulting expressions become greatly simplified for short wavelengths and when the Coriolis effect is neglected. However, for most practical cases involving tropospheric fallout, the foregoing restrictions are not severe since sounding stations are presumed to exist at reasonable distances from each other.

From a theoretical point of view, this investigation is a modified extension of the earlier work of Queney,[A.1,A.2] but there are differences which render somewhat different results. Although both models utilize perturbation theory to include the effects of variable terrain, there is a distinct conceptual difference between them arising from the choice of the dependent variables. Queney deals with the displaced trajectories of the streamlines as the fundamental physical quantities of interest (which seems to introduce extra degrees of complexity into the problem), while we treat the changes in the velocity field. Our method of attack permits more refined criteria for establishing the validity of the calculation and leads quite naturally to a generalization to three-dimensional systems, which are more frequently encountered than the two-dimensional idealizations of Queney. Moreover, we show that a perturbation theory model for the hydrodynamics does not necessarily imply the applicability of superposition of ground disturbances, a result which does not seem to have been recognized earlier. This is a distinct problem. However, we are able to demonstrate that the superposition hypothesis can serve as the basis of an iterative scheme for computing the velocity field to an arbitrary degree of accuracy consistent with the initial premises of the perturbation method. In certain two-dimensional cases, there does not appear to be much difference between Queney's results and ours.

The overall validity of the model is based upon the applicability of the nonturbulent hydrodynamics equations together with the assumption of an adiabatic atmosphere in the unperturbed state. Consequently, the solutions do not yield lee waves when applied to the assumed small scale disturbances considered in this investigation. In addition, the results are not generally valid in the lower regions of the atmosphere where turbulent boundary layer effects may dominate the physical processes; however, this is not especially important for fallout since uncertainties attributed to lower atmosphere effects will be only a few hundred feet.

Soluble mathematical models of airflow in the troposphere must in some measure be removed from reality because of the enormous complexity of the actual physical system. Despite this inherent limitation, the nonturbulent models of airflow can be useful for fallout calculations if they at least semiquantitatively describe the salient features of the particular aerodynamics. The utility of such models can best be evaluated by comparison with suitable experiments.

Geometric Considerations

For mathematical simplicity the origin is located at a suitable point in the region where the airflow is to be computed. The x axis is established along the unperturbed wind direction, the y axis is perpendicular to the x axis, and the z-axis points in the direction of the zenith. If ϵ denotes the angle between the local west-east direction and the unperturbed wind velocity, then the components of $\underset{\sim}{\Omega}$ are given by

$$\Omega_y = \Omega \cos \Theta \cos \epsilon, \qquad \Omega_x = \Omega \cos \Theta \sin \epsilon, \qquad \Omega_z = \Omega_z ,$$

where Θ is the latitude, and Ω is the sideral day frequency which equals 7.3×10^{-5} sec^{-1}. For our problems all the components of $\underset{\sim}{\Omega}$ are assumed constant (i.e., the curvature of the earth is neglected).

Theory of Airflow

Airflow over variable terrain can be determined by assuming that the changes in wind velocity caused by the ground irregularities are a small perturbation on the wind field. It is postulated that if the ground were flat, the wind velocity, u, would be constant both in position and time. Orographic effects due to mountains and valleys then cause the wind field to change in a determined way as computed from the perturbation theory.

The origin of the coordinate system is established at a suitable point in the vicinity of the region where the wind field is to be computed. Assuming that for all times the thermodynamic process which describes the flow of air is isentropic, the relationship between pressure P and air-mass density ρ is given by

$$\left(P/P_e\right) = \left(\rho/\rho_e\right)^{\gamma} = \left(T/T_e\right)^{\gamma/(1-\gamma)} , \tag{A.1}$$

where P_e, ρ_e, and T_e are the pressure, mass density, and temperature at the origin in the unperturbed case and $\gamma = 1.4$. These quantities are further related to each other by the ideal gas law,

$$P_e = \left(\rho_e k T_e /m\right) , \tag{A.2}$$

where k is the Boltzmann constant ($k = 1.38 \times 10^{-16}$ erg deg^{-1}), and m is the mass of the air molecule. The two equations which describe the aerodynamics are the continuity equation and the momentum equation:

$$\partial\rho/\partial t + \nabla \cdot (\rho \underset{\sim}{v}) = 0 \tag{A.3}$$

and

$$d\underset{\sim}{v}/dt = 2\underset{\sim}{v} \times \underset{\sim}{\Omega} - \nabla\psi + \underset{\sim}{G} \ , \tag{A.4}$$

where $\underset{\sim}{G}$ is the gravity force and is equal to $-G\underset{\sim}{k}$, and ψ is a potential obtained by combining the $(1/\rho)\nabla P$ term with Eq. (A.1).

$$\psi = \left(P_e / \rho_e^{\gamma}\right) [\gamma/(\gamma - 1)] \, \rho^{(\gamma - 1)} \ . \tag{A.5}$$

We assume that a steady state exists in which there is only one uniform (spatially homogeneous) component of velocity, u_o, which, by construction, is parallel to the x direction. The system of equations then reduces to

$$0 = -\partial\psi/\partial x, \qquad 0 = -2u_o\Omega_z - \partial\psi/\partial y, \qquad 0 = 2u_o\Omega_y - \partial\psi/\partial z - G \ . \tag{A.6}$$

The general solution to Eq. (A.6) is given by

$$\psi = Ay + Bz + \psi_{or} \ , \tag{A.7}$$

which when substituted into the foregoing equations gives

$$A = -2u_o\Omega_z, \qquad B = -G + 2u_o\Omega_y \simeq -G \ , \tag{A.8}$$

$$\psi_{or} = \gamma/(\gamma - 1) \quad P_e / \rho_e = \gamma/(\gamma - 1) \quad \left(kT_e / m\right) = c_s^2 / (\gamma - 1) \ , \tag{A.9}$$

where c_s is the speed of sound and equals 3.4×10^4 cm sec^{-1} under STP conditions.

A measure of the distances over which changes in ψ are important in the equilibrium case can be determined by examining the ratios $\psi_{or}/|A| = y_c$ and $\psi_{or}/|B| = z_c$, which are respectively the distances over which the independent changes in ψ equal the value at the origin. We have

$$y_c = \frac{29 \times 10^8}{2u_o \Omega_z} \quad , \tag{A.10}$$

and

$$z_c = \frac{29 \times 10^8}{980} = 3 \times 10^6 \text{ cm} = 20 \text{ mi} \quad . \tag{A.11}$$

Using a maximum value of $\Omega_z = \Omega = 7\ 3 \times 10^{-5}$ sec^{-1} and a value of $u_o = 4400$ cm sec^{-1} (corresponding to a 100 mph wind) gives a value of $y_c \approx 4.5 \times 10^9$ cm $= 3 \times 10^4$ mi which signifies that for local fallout variations in y can be neglected altogether in the equilibrium case. This will not be true in general in the perturbed case.

The initial state of the system is thus specified by the velocity

$$\underset{\sim}{v} = \underset{\sim}{i} u_o \tag{A.12}$$

and density

$$\rho = \rho_o(z) = \rho_e\left(1 - z/z_c\right)^{1/(\gamma - 1)} = \rho_e(1 - \alpha z)^{1/(\gamma - 1)} \quad , \tag{A.13}$$

where

$$\alpha = 1/z_c = 1/\left(3 \times 10^6\right) = 0.33 \times 10^{-6} \text{ cm}^{-1} \quad . \tag{A.14}$$

If attention is further confined to the troposphere ($z \leq 2$ mi $= 3.0 \times 10^5$ cm), the variation of density with altitude is approximated by

$$\rho_o(z) \simeq \rho_e \left\{1 - [\alpha/(\gamma - 1)]\ z\right\} = \rho_e(1 - \beta z) \simeq \rho_e\ e^{-\beta z} \quad , \tag{A.15}$$

where β is defined as the tropospheric density attenuation constant,

$$\beta = \alpha/(\gamma - 1) = mG/\gamma kT_e = 2.5\alpha = 0.83 \times 10^{-6}\ \mathrm{cm}^{-1} \quad . \tag{A.16}$$

We now assume that the three components of velocity and density become modified by the terrain. The perturbed quantities are assumed to be related to the unperturbed ones by the equations

$$u_p = u_o + \bar{u}, \quad v_p = \bar{v}, \quad w_p = \bar{w}, \quad \rho_p = \rho_o + \bar{\rho}, \quad \psi_p = \psi_o + \bar{\psi} \quad . \tag{A.17}$$

Substituting Eq. (A.17) into Eqs. (A.3) and (A.4) and neglecting second order effects, such as $\bar{\rho}\bar{w}$ and $\bar{v}\bar{w}$, gives under stationary conditions ($\partial/\partial t = 0$)

$$\rho_o\left(\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z}\right) + u_o \frac{\partial}{\partial x}\bar{\rho} + \bar{w}\frac{\partial}{\partial z}\rho_o = 0 \quad , \tag{A.18}$$

$$u_o \frac{\partial \bar{u}}{\partial x} = 2\left(\bar{v}\Omega_z - \bar{w}\Omega_y\right) - \frac{\partial \bar{\psi}}{\partial x} \quad , \tag{A.19}$$

$$u_o \frac{\partial \bar{v}}{\partial x} = 2\left(\bar{w}\Omega_x - \bar{u}\Omega_z\right) - \frac{\partial \bar{\psi}}{\partial y} \quad , \tag{A.20}$$

and

$$u_o \frac{\partial \bar{w}}{\partial x} = 2\left(\bar{u}\Omega_y - \bar{v}\Omega_x\right) - \frac{\partial \bar{\psi}}{\partial z} \quad . \tag{A.21}$$

For mathematical convenience, it is desirable to deal with a function, η, related to the initial density ρ_o by the formula

$$\eta = \bar{\rho}/\rho_o \quad . \tag{A.22}$$

In terms of η, $\bar{\psi}$ is given by

$$\bar{\psi} = \psi\left(\rho_o + \bar{\rho}\right) - \psi\left(\rho_o\right) = \gamma\left(P_e/\rho_e^{\gamma}\right)\rho_o^{\gamma-1}\,\eta \quad . \tag{A.23}$$

Using Eq. (A.13) for $\rho_o(z)$ gives

$$\bar{\psi} = \left(\gamma kT_e/m\right)(1 - \alpha z)\eta \quad ; \tag{A.24}$$

while the derivatives of $\bar{\psi}$ are

$$\frac{\partial\bar{\psi}}{\partial x} = \frac{\gamma kT_e}{m}(1 - \alpha z)\frac{\partial\eta}{\partial x} \quad , \tag{A.25}$$

$$\frac{\partial\bar{\psi}}{\partial y} = \frac{\gamma kT_e}{m}(1 - \alpha z)\frac{\partial\eta}{\partial y} \quad , \tag{A.26}$$

and

$$\frac{\partial\bar{\psi}}{\partial z} = -(\gamma - 1)\,G + \frac{\gamma kT_e}{m}(1 - \alpha z)\frac{\partial\eta}{\partial z} \quad . \tag{A.27}$$

The object at this point is to reduce Eqs. (A.18)-(A.21) to a system of linear equations with constant coefficients. This can readily be accomplished by restricting the calculation to values of z much less than z_c so that $(1 - \alpha z) = (1 - z/z_c) \simeq 1$. Also, within this range, $\ln\rho_o = \ln\rho_e - \beta z$, thereby yielding

$$\left(\frac{\partial\bar{u}}{\partial x} + \frac{\partial\bar{v}}{\partial y} + \frac{\partial\bar{w}}{\partial z}\right) + u_o\frac{\partial}{\partial x}\eta - \beta\bar{w} = 0 \quad , \tag{A.28}$$

$$u_o\frac{\partial\bar{u}}{\partial x} = 2\left(\bar{v}\Omega_z - \bar{w}\Omega_y\right) - \frac{\gamma kT_e}{m}\frac{\partial\eta}{\partial x} \quad , \tag{A.29}$$

$$u_o\frac{\partial\bar{v}}{\partial x} = 2\left(\bar{w}\Omega_x - \bar{u}\Omega_z\right) - \frac{\gamma kT_e}{m}\frac{\partial\eta}{\partial y} \quad , \tag{A.30}$$

and

$$u_o \frac{\partial \bar{w}}{\partial x} = 2\left(\bar{u}\Omega_y - \bar{v}\Omega_x\right) + (\gamma - 1)G\eta - \frac{\gamma k T_e}{m}\frac{\partial \eta}{\partial z} . \tag{A.31}$$

Equations (A.28)-(A.31) relate the perturbed quantities to one another, but the absolute scale of the perturbations must be obtained from the new boundary conditions. We now assume that each of the perturbed quantities can be expressed by an expansion of plane waves.

$$\bar{u} = \int A(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d^3k, \qquad \bar{v} = \int B(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d^3k ,$$

and (A.32)

$$\bar{w} = \int C(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d^3k, \qquad \eta = \int D(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d^3k ,$$

where $d^3k = dk_x\, dk_y\, dk_z$. Substituting Eq. (A.32) into Eqs. (A.28)-(A.31) leads to the dispersion relationship between the components of the wave vector. Thus, we have

$$\begin{pmatrix} ik_x & ik_y & \left(ik_z - \beta\right) & iu_o k_x \\ ik_x u_o & -2\Omega_z & 2\Omega_y & \left(ik_x G/\beta\right) \\ 2\Omega_z & ik_x u_o & -2\Omega_x & \left(ik_y G/\beta\right) \\ -2\Omega_y & 2\Omega_x & ik_x u_o & \left[(1-\gamma)G + \left(ik_z G/\beta\right)\right] \end{pmatrix} \begin{pmatrix} A \\ B \\ C \\ D \end{pmatrix} = 0 . \tag{A.33}$$

The only nontrivial solutions to Eq. (A.33) occur when the determinant of the matrix equals zero. This establishes a connection between k_x, k_y, and k_z. The so-called dispersion relationship can be interpreted in several ways depending on which component(s) of the wave vector $\underset{\sim}{k}$ can be preassigned. It is at this point that the pertinent physical factors are introduced into the problem. Since the topography can be resolved into periodic components of x and y, we must necessarily regard k_x and k_y as real numbers. The dispersion relationship is then interpreted as

$$k_z = k_z\left(k_x, k_y\right) . \tag{A.34}$$

For each set of values (k_x, k_y) there will be two solutions of Eq. (A.34) that correspond to the roots of the equation which results when the determinant of the matrix Eq. (A.33) is set equal to zero. Since the number of solutions of Eq. (A.34) is finite, we can contract the description of the Fourier components of the field quantities. For example, $\tilde{w}$ now becomes

$$\bar{w}(x, y, z) = \sum_{\mu=1,2} \iint C_\mu\left(k_x, k_y\right) e^{ik_x x} e^{ik_y y} e^{ik_z^\mu\left(k_x, k_y\right)z} dk_y\, dk_x ;$$

where k_z^μ stands for the μ^{th} root of Eq. (A.34). Setting the determinant of the matrix of Eq. (A.33) equal to zero leads to the dispersion relationship

$$ak_z^2 + b\left(ik_z\right) + c = 0 \ , \tag{A.35}$$

where

$$a = \sigma\left(k_x^2 u_o^2 - \Omega^2 \omega_x^2\right) \ , \tag{A.36a}$$

$$b = \sigma\left[\gamma\beta k_x^2 u_o^2 + 2\Omega k_x k_y u_o \omega_x + \Omega^2\left(2ik_x\omega_z\omega_x + 2ik_y\omega_z\omega_y - \gamma\beta\omega_z^2\right)\right] , \tag{A.36b}$$

and

$$\begin{aligned} c = {} & k_x^2 u_o^2\left[\sigma\left(k_x^2 + k_y^2\right) - u_o^2 k_x^2 + (1 - \gamma)\sigma\beta^2\right] \\ & + \Omega\beta\sigma\left[2k_x^2\omega_y u_o - \gamma\left(\omega_y u_o k_x^2 - \omega_x u_o k_x k_y\right)\right] \\ & + \Omega^2\left[4k_x^2 u_o^2 - \sigma\left(k_x\omega_x + k_y\omega_y\right)^2 - \omega_z^2(1 - \gamma)\sigma\beta^2 - \beta\sigma\gamma\left(ik_x\omega_x\omega_y + ik_y\omega_y\omega_z\right)\right] , \end{aligned} \tag{A.36c}$$

in which

$$\omega_x = 2\Omega_x/\Omega, \quad \omega_y = 2\Omega_y/\Omega, \quad \omega_z = 2\Omega_z/\Omega, \quad \sigma = (G/\beta) \ . \tag{A.37}$$

Since Eq. (A.33) is homogeneous, the absolute magnitudes of the functions $A(\underset{\sim}{k})$, $B(\underset{\sim}{k})$, $C(\underset{\sim}{k})$, and $D(\underset{\sim}{k})$ cannot be determined; only their relationship to each other, as deduced from the boundary conditions, can. For mathematical convenience it is desirable to eliminate $D(\underset{\sim}{k})$ and deal only with the Fourier transforms of the velocity components of the wind velocity. Thus, we obtain the equation

$$\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix} \begin{pmatrix} A \\ B \\ C \end{pmatrix} = 0 \;, \tag{A.38}$$

where the b_{ik}'s are the elements of a matrix $\tilde{b}$ and are given by

$$b_{11} = ik_x\xi + iu_ok_x\Omega\omega_y, \quad b_{12} = ik_y\xi - iu_ok_x\Omega\omega_x, \quad b_{13} = \left(ik_z - \beta\right)\xi + \left(u_ok_x\right)^2$$

$$b_{21} = \left(ik_xu_o\right)\xi + ik_x\sigma\Omega\omega_y, \quad b_{22} = -\Omega\omega_z\xi - \left(ik_x\sigma\right)\Omega\omega_x, \quad b_{23} = \Omega\omega_y\xi - \left(ik_x\sigma\right)\left(ik_xu_o\right),$$

$$b_{31} = \Omega\omega_z\xi + \left(ik_y\sigma\right)\Omega\omega_y, \quad b_{32} = ik_xu_o\xi - \left(ik_y\sigma\right)\Omega\omega_x, \quad b_{33} = -\Omega\omega_x\xi + k_xk_y\sigma u_o, \tag{A.39}$$

in which $\xi = (1 - \gamma)G + (ik_z\sigma)$ and $\sigma = (G/\beta)$.

The dispersion relationship derived by setting the determinant of $\tilde{b}$ equal to zero is necessarily the same as that previously derived. Using Eq. (A.38), we deduce the general relationship between the Fourier transforms of the velocity components:

$$A = -\frac{\left(b_{12}b_{33} - b_{13}b_{32}\right)}{\left(b_{12}b_{31} - b_{11}b_{32}\right)}\, C \equiv T\left(k_x, k_y\right)\, C\left(k_x, k_y\right) \tag{A.40}$$

and

$$B = -\frac{\left(b_{13}b_{31} - b_{11}b_{33}\right)}{\left(b_{12}b_{31} - b_{11}b_{32}\right)}\, C \equiv U\left(k_x, k_y\right)\, C\left(k_x, k_y\right) \;. \tag{A.41}$$

Up to this point the analysis has been quite general, but henceforth we shall focus attention in the regime where the Coriolis effect is negligible. This is equivalent to setting $\Omega = 0$. The relationships between the physical parameters which must be satisfied to justify this step for fallout applications is discussed later in this appendix. Setting $\Omega = 0$ in the dispersion relationship gives

$$\sigma k_z^2 + \left(ik_z\right)\sigma\gamma\beta + \sigma\left(k_x^2 + k_y^2\right) - u_o^2 k_x^2 + (1 - \gamma)\sigma\beta^2 = 0 \ . \tag{A.42}$$

We now let

$$\lambda = -ik_z \ , \tag{A.43}$$

anticipating an exponential decay with altitude. Since $\sigma = G/\beta = 1.2 \times 10^9 \text{cm}^2\text{sec}^{-2} >> u_o^2$, we can neglect $u_o^2 k_x^2$ in Eq. (A.42) and thus obtain

$$\lambda^2 + \lambda\gamma\beta - k_x^2 - k_y^2 + (\gamma - 1)\beta^2 = 0 \ . \tag{A.44}$$

The roots of Eq. (A.44) are given by

$$\lambda = \frac{-\gamma\beta \pm \left\{(\gamma\beta)^2 + 4\left[k_x^2 + k_y^2 - \beta^2(\gamma - 1)\right]\right\}^{1/2}}{2} \ . \tag{A.45}$$

Since we are primarily interested in short range effects, we shall freely make use of the inequality $k_x^2 + k_y^2 >> \beta^2$ (this implies that the wavelength of the horizontal variations, $2\pi\beta^{-1}$, be less than 50 mi), which yields the following two roots of Eq. (A.45)

$$\lambda = \pm\left(k_x^2 + k_y^2\right)^{1/2} \ , \tag{A.46}$$

where only the positive root is acceptable on physical grounds since this guarantees that the perturbations will dampen at high altitudes. Using the plus root of Eq. (A.46) in Eqs. (A.40) and (A.41) gives

$$T\left(k_x, k_y\right) = -ik_x / \left(k_x^2 + k_y^2\right)^{1/2} \ , \tag{A.47}$$

and

$$U\left(k_x, k_y\right) = -ik_y \Big/ \left(k_x^2 + k_y^2\right)^{1/2} \quad . \tag{A.48}$$

From Eqs. (A.46)-(A.48) we deduce the following expressions for the perturbed velocity components:

$$\bar{w}(x, y, z) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} C\left(k_x, k_y\right) e^{i\left(k_x x + k_y y\right)} e^{-\left(k_x^2 + k_y^2\right)^{1/2} z} dk_x \, dk_y \; , \tag{A.49}$$

$$\bar{u}(x,y,z) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(-ik_x\right)\left(k_x^2 + k_y^2\right)^{-1/2} e^{i\left(k_x x + k_y y\right)} C\left(k_x, k_y\right) e^{-\left(k_x^2 + k_y^2\right)^{+1/2} z} dk_x \, dk_y \tag{A.50}$$

and

$$\bar{v}(x,y,z) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \left(-ik_y\right)\left(k_x^2 + k_y^2\right)^{-1/2} e^{i\left(k_x x + k_y y\right)} C\left(k_x, k_y\right) e^{-\left(k_x^2 + k_y^2\right)^{+1/2} z} dk_x \, dk_y \tag{A.51}$$

The function $C(k_x, k_y)$ is determined by application of the perturbed boundary condition. Let

$$\phi(x, y, z) = 0 = z - f(x, y) \tag{A.52}$$

be the equation of the earth's surface. The normal to this surface is $\nabla\phi$:

$$\nabla\phi = -\underset{\sim}{i} \frac{\partial f}{\partial x} - \underset{\sim}{j} \frac{\partial f}{\partial y} + \underset{\sim}{k} \quad . \tag{A.53}$$

On physical grounds, we must necessarily demand that the wind velocity be parallel to the surface $z = f(x, y)$ at every point. Thus, the boundary conditions are mathematically stated as

$$\left[\left(\underset{\sim}{v} \cdot \nabla\phi\right)\right] = 0 \quad \text{along } z = f(x, y) \; , \tag{A.54}$$

where

$$\underset{\sim}{v} = \underset{\sim}{i}u + \underset{\sim}{j}v + \underset{\sim}{k}w = \underset{\sim}{i}u_o + (\underset{\sim}{i}\tilde{u} + \underset{\sim}{j}\tilde{v} + \underset{\sim}{k}\tilde{w}) \ . \tag{A.55}$$

Inserting Eq. (A.55) into Eq. (A.54) gives

$$\tilde{w}(x, y, z = f) = u_o\left(\frac{\partial f}{\partial x}\right) + \tilde{u}(x, y, z = f)\left(\frac{\partial f}{\partial x}\right) + \tilde{v}(x, y, z = f)\left(\frac{\partial f}{\partial y}\right) \ . \tag{A.56}$$

Using Eqs. (A.47)-(A.51) in Eq. (A.56) yields the following integral equation for $C(k_x, k_y)$:

$$\int_{k_x}\int_{k_y}\left[1 - T\left(k_x, k_y\right)\left(\frac{\partial f}{\partial x}\right) - U\left(k_x, k_y\right)\left(\frac{\partial f}{\partial y}\right)\right]$$

$$C\left(k_x, k_y\right) e^{ik_x x} e^{ik_y y} e^{\left[-\left(k_x^2 + k_y^2\right)^{1/2} f(x, y)\right]} dk_x \, dk_y$$

$$= u_o \int_{k_x}\int_{k_y}\left(ik_x\right) F\left(k_x, k_y\right) e^{ik_x x} e^{ik_y y} dk_x \, dk_y \ , \tag{A.57}$$

where

$$F\left(k_x, k_y\right) = \left(\frac{1}{2\pi}\right)^2 \iint f(x, y) \, e^{-ik_x x} e^{-ik_y y} dx \, dy \ ,$$

and (A.58)

$$f(x, y) = \iint F\left(k_x, k_y\right) e^{ik_x x} e^{ik_y y} dk_x \, dk_y \ .$$

The solution for $C(k_x, k_y)$ is impossible to achieve by direct means because of the dependence of the integration of the left-hand side of Eq. (A.57) on f(x, y) and on the derivatives of $f(x, y)$. However, a systematic perturbation method for computing $C(k_x, k_y)$ can be deduced. The approximation to $C(k_x, k_y)$, achieved by setting $\exp - \left[\left(k_x^2 + k_y^2 \right)^{1/2} f(x, y) \right]$ equal to unity, is equivalent to assuming that the maximum elevation, f_{max}, is small compared to the wavelength of the horizontal oscillation, or in simpler terms the slope of the terrain is small. On the other hand, the neglect of $\bar{u}(\partial f/\partial x)$ as compared to $u_o(\partial f/\partial x)$ is necessarily consistent with the initial premise of the perturbation method used in this analysis, namely that the change in the velocity field be small compared to the initial velocity. This obviously must apply when comparing $\bar{u}$ to u_o. The neglect of $\bar{v}(\partial f/\partial y)$ as compared to $u_o(\partial f/\partial x)$ is somewhat difficult to justify under all cases. Although $\bar{v}$ is assumed small compared to u_o, we must also be sure that $(\partial f/\partial y)$ is not substantially greater than $(\partial f/\partial x)$.

The apriori assumption

$$\bar{u}\left(\frac{\partial f}{\partial x}\right) + \bar{v}\left(\frac{\partial f}{\partial y}\right) < u_o\left(\frac{\partial f}{\partial x}\right) \tag{A.59}$$

is equivalent to neglecting $-T(\partial f/\partial x) - U(\partial f/\partial y)$ as compared to unity in Eq. (A.57). The systematic method for computing $C(k_x, k_y)$ is based upon Eq. (A.59) coupled with the previously mentioned approximation

$$\exp \left[- \left(k_x^2 + k_y^2 \right)^{1/2} f(x, y) \right] \equiv \Gamma \approx 1 \quad . \tag{A.60}$$

Introduction of the functions

$$\xi = 1 - \Gamma = - \sum_{n=1} \frac{\phi^n}{n!} \quad , \tag{A.61}$$

where

$$\phi = - \left(k_x^2 + k_y^2 \right)^{1/2} f(x, y) \quad , \tag{A.62}$$

and

$$\tau = T\left(k_x, k_y\right)\left(\frac{\partial f}{\partial x}\right) + U\left(k_x, k_y\right)\left(\frac{\partial f}{\partial y}\right) \tag{A.63}$$

permits Eq. (A.57) to be written as

$$\int C(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d\underset{\sim}{k} = u_o \int H(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d\underset{\sim}{k} + \int \Delta C(\underset{\sim}{k})\, e^{i\underset{\sim}{k}\cdot\underset{\sim}{r}}\, d\underset{\sim}{k}\ . \tag{A.64}$$

where

$$\Delta = (\tau + \xi - \xi\tau) = \Delta(\underset{\sim}{k}, x, y), \quad H(\underset{\sim}{k}) = ik_x F(\underset{\sim}{k}), \quad \underset{\sim}{k} = \underset{\sim}{i}k_x + \underset{\sim}{j}k_y, \quad d\underset{\sim}{k} = dk_x\, dk_y\ .$$

Multiplying Eq. (A.64) by exp $(-i\underset{\sim}{k}'\cdot\underset{\sim}{r})$, and then integrating over $\underset{\sim}{r}$, gives

$$C(\underset{\sim}{k}') = u_o H(\underset{\sim}{k}') + \left(\frac{1}{2\pi}\right)^2 \int_{\underset{\sim}{k}} \int_{\underset{\sim}{r}} \Delta(\underset{\sim}{k}, \underset{\sim}{r})\, C(\underset{\sim}{k})\, e^{i(\underset{\sim}{k} - \underset{\sim}{k}')\cdot\underset{\sim}{r}}\, d\underset{\sim}{k}\, d\underset{\sim}{r}\ . \tag{A.65}$$

The perturbation scheme is developed by regarding the second term on the right-hand side of Eq. (A.65) as small. The first approximation to $C(\underset{\sim}{k})$, denoted by $C^{(1)}$, is deduced by completely disregarding the second term of the right-hand side. Thus

$$C^{(1)}(\underset{\sim}{k}') = u_o H(\underset{\sim}{k}') = u_o\left(ik_x'\right) F(\underset{\sim}{k}')\ . \tag{A.66}$$

Since $F(k)$ is the sum of individual contributions to the topography, we see that the principle of linear superposition is also reflected in $C^{(1)}(k)$. The second approximation is obtained by using Eq. (A.66) for $C(\underset{\sim}{k})$ in the integral expression

$$C^{(2)}(\underset{\sim}{k}') = u_o H(\underset{\sim}{k}') + \left(\frac{1}{2\pi}\right)^2 \int_{\underset{\sim}{k}} \int_{\underset{\sim}{r}} \Delta(\underset{\sim}{k}, \underset{\sim}{r}) \left[u_o H(\underset{\sim}{k})\right] e^{i(\underset{\sim}{k} - \underset{\sim}{k}')\cdot\underset{\sim}{r}}\, d\underset{\sim}{k}\, d\underset{\sim}{r}\ . \tag{A.67}$$

It follows by inspection that the nth approximation to $C(\underset{\sim}{k})$ is given by

$$C^{(n)}(\underset{\sim}{k'}) = u_o H(\underset{\sim}{k'}) + \left(\frac{1}{2\pi}\right)^2 \int_{\underset{\sim}{k}} \int_{\underset{\sim}{r}} \Delta(\underset{\sim}{k}, \underset{\sim}{r})\, C^{(n-1)}(k)\, e^{i(\underset{\sim}{k} - \underset{\sim}{k'}) \cdot \underset{\sim}{r}}\, d\underset{\sim}{k}\, d\underset{\sim}{r} \quad , \tag{A.68}$$

with

$$C^{(0)} = 0 \quad .$$

The corresponding Fourier transforms of the perturbed x and y components of the wind field, $A(\underset{\sim}{k})$ and $B(\underset{\sim}{k})$ respectively, are found from Eqs. (A.40) and (A.41) to the same order of approximation. The ultimate validity of this perturbation method can be evaluated only posteriorly — by comparing the calculated change in the magnitude of the wind field with u_o. Mathematically, the developed theory is valid so long as

$$\bar{u}^2 + \bar{v}^2 + \bar{w}^2 < u_o^2 \quad . \tag{A.69}$$

The prescription for calculating the wind field due to terrain effects is summarized as follows. Equation (A.68) is used to compute the Fourier transform of the vertical wind. The Fourier transform of the change in the horizontal components of the wind field is then determined by Eqs. (A.40) and (A.41). Finally, the inversion formula is employed to compute $\bar{u}(\underset{\sim}{r})$, $\bar{v}(\underset{\sim}{r})$, and $\bar{w}(\underset{\sim}{r})$.

The value of computing $C(\underset{\sim}{k})$ beyond the first approximation is worthwhile, even though the hydrodynamics model considers only the first correction to the flow, because the iteration scheme can more precisely establish the range of validity of the first approximation to $C(\underset{\sim}{k})$ and the dependence of $C(\underset{\sim}{k})$ on the characteristic features of the terrain. Most topography is complex and, as such, cannot always be represented by a simple periodic structure, but rather by a sum of frequencies. The higher corrections to $C(\underset{\sim}{k})$ take into account the interaction between the Fourier components of the ground structure and, consequently, must be evaluated to more firmly establish the validity of the superposition principle implicit in the first approximation.

In the next section, we shall apply the first-order theory to compute changes in the wind field caused by specific orographic effects. However, now we shall consider a simple two dimension periodic structure to exhibit the method for computing higher order corrections to $C(\underset{\sim}{k})$.

Let

$$f(x) = h\, e^{ik_o x} , \tag{A.70}$$

from which we have:

$$F(k) = \frac{h}{2\pi} \int e^{ik_o x}\, e^{-ikx}\, dx = h\delta\left(k - k_o\right) \tag{A.71}$$

and

$$C^{(1)}(k) = iku_o h\delta\left(k - k_o\right) = iu_o\left(k_o h\right)\, \delta\left(k - k_o\right) , \tag{A.72}$$

where $\delta(k)$ is the Dirac delta function. This yields

$$\bar{w}(x, z) = \int iku_o h\delta\left(k - k_o\right)\, e^{ikx}\, e^{-|k|z}\, dk = ik_o hu_o\, e^{ik_o x}\, e^{-|k_o|z} . \tag{A.73}$$

Since $T(k_x, k_y = 0) = -ik/|k|$, it follows that

$$\bar{u}(x, z) = +u_o\left(|k_o|h\right)\, e^{ik_o x}\, e^{-|k_o|z} . \tag{A.74}$$

Within the confines of the first approximation the inequality of Eq. (A.69) reduces to

$$(2)^{1/2}\left(|k_o|h\right) << 1 , \tag{A.75}$$

which basically shows that the slope of the terrain must be less than unity. The second approximation to $C(\underset{\sim}{k})$ is given by

$$C^{(2)}(\underset{\sim}{k}') = iu_o\left(k_o h\right)\delta\left(k' - k_o\right) + \left(\frac{1}{2\pi}\right) \iint\limits_{k\,x} \Delta(k, x)\left[iu_o\left(k_o h\right)\delta\left(k - k_o\right)\right] e^{i(k - k')}\, dk\, dx , \tag{A.76}$$

where

$$\Delta(k,x) = T(k)\left(\frac{\partial f}{\partial x}\right) - \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k|f)^n + T(k)\left(\frac{\partial f}{\partial x}\right) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k|f)^n . \tag{A.77}$$

Inserting Eq. (A.77) into Eq. (A.76) and performing the integration over k and x gives the following expression for $C^{(2)}(k')$:

$$\begin{aligned} C^{(2)}(k') = {} & iu_o\left(k_o h\right) \delta\left(k' - k_o\right) + iu_o\left(k_o h\right)^2 \delta\left(k' - 2k_o\right) \operatorname{sgn}\left(k_o\right) \\ & - iu_o\left(k_o h\right) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \left(|k_o| h\right)^n \delta\left[k' - (n+1)k_o\right] \\ & + iu_o\left(k_o h\right)^2 \operatorname{sgn}\left(k_o\right) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} \left(|k_o| h\right)^n \delta\left[k' - (n+2)k_o\right] , \end{aligned} \tag{A.78}$$

where

$$\operatorname{sgn}\left(k_o\right) \equiv |k_o| / k_o .$$

It is easy to show that the vertical component of velocity corresponding to $C^{(2)}(k')$ is given by

$$\begin{aligned} \bar{w}(x,z) = {} & iu_o\left(k_o h\right) \exp\left(ik_o x - |k_o| z\right)\left\{2 - \exp\left[-|k_o| h \exp\left(ik_o x - |k_o| z\right)\right]\right\} \\ & + \operatorname{sgn}\left(k_o\right) iu_o\left(k_o h\right)^2 \exp\left[2\left(ik_o x - |k_o| z\right]\right. \exp\left[-|k_o| h \exp\left(ik_o x - |k_o| z\right)\right] . \end{aligned} \tag{A.79}$$

Examination of the second term in Eq. (A.79) shows that the uncertainties introduced in the computation of $\bar{w}(x,z)$ by neglecting higher order terms are of the order of $(k_o h)^2$ for a one-dimensional periodic structure. The degree of accuracy to which one may choose to compute the velocity field for fallout computations should be consistent with the uncertainties introduced in other aspects of the calculation.

Application of the Theory to Specific Geometries

In this section we shall apply the theory to an infinite mountain ridge which makes an arbitrary angle with respect to the unperturbed flow, and to a mountain. Within the context of the theory a valley may be considered as an inverted mountain or mountain ridge. We have chosen these particular models because the terrain can be mathematically interpreted as a superposition of mountains and mountain ridges, and hence the general solution of airflow over variable terrain can be determined by superimposing the solutions for individual mountains, valleys, and ridges.

Mountain Ridge Not Perpendicular to Unperturbed Flow

In this case, the perpendicular to the line depicting the crest of the mountain makes an angle γ with respect to the direction of flow, as shown in Figure A.1.

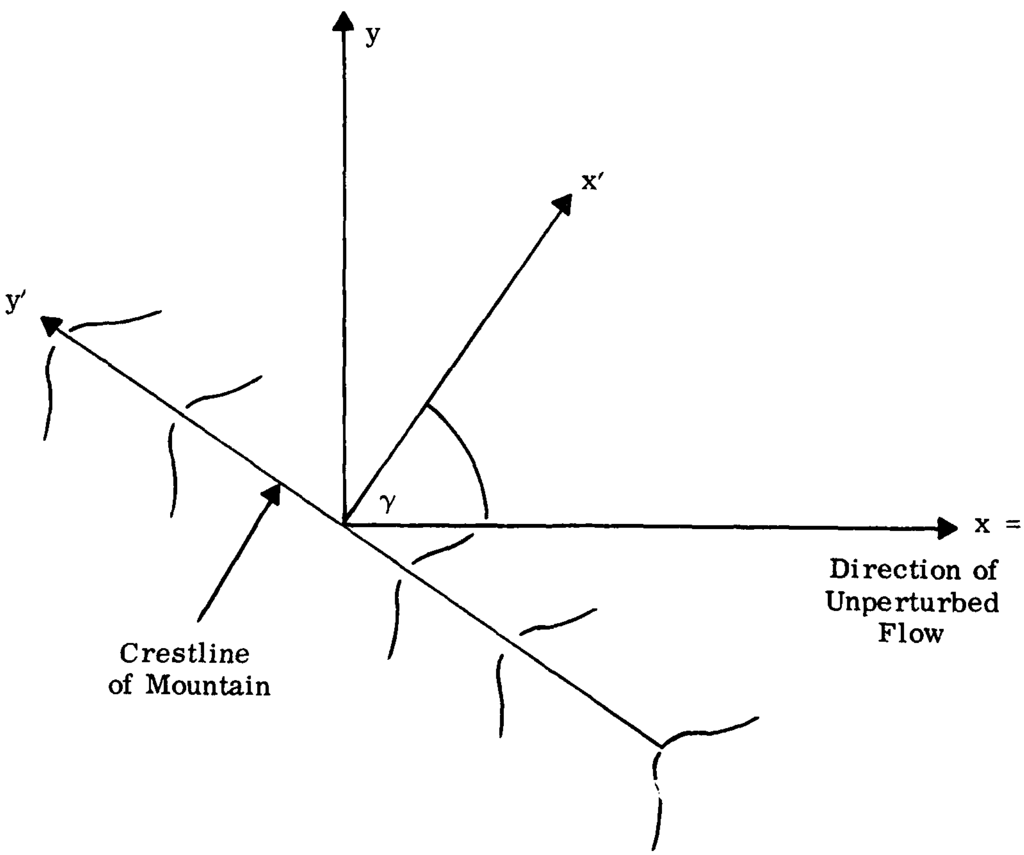


Figure A.1. Mountain Ridge Not Perpendicular to Flow

A suitable mathematical representation of a mountain ridge when viewed along the y' axis has been deduced by Queney[A.2] who showed that the topography could be represented by the equation

$$z = f'(x', y') = \frac{h}{1 + (x'/a)^2} \quad , \tag{A.80}$$

which as observed is independent of y'. However, the transformation equations

$$x = x' \cos\gamma - y' \sin\gamma, \qquad y = x' \sin\gamma + y' \cos\gamma \tag{A.81}$$

show that the topographical description in the x, y system, namely

$$f(x, y) = f'[x'(x, y), \; y'(x, y)] \quad , \tag{A.82}$$

will be a function of both x and y. The Fourier transform of the mountain ridge function in our system is

$$F\left(k_x, k_y\right) = \left(\frac{1}{2\pi}\right)^2 \iint f(x, y)\, e^{-ik_x x - ik_y y}\, dx\, dy \quad . \tag{A.83}$$

However, since $\underset{\sim}{k} \cdot \underset{\sim}{r}$ is invariant under an orthogonal transformation, we have

$$F\left(k_x, k_y\right) = F'\left[k'_x\left(k_x, k_y\right), k'_y\left(k_x, k_y\right)\right] = \left(\frac{1}{2\pi}\right)^2 \iint f'(x', y')\, e^{-ik'_x x' - ik'_y y'}\, dx'\, dy' \ , \tag{A.84}$$

where

$$k_x = k'_x \cos\gamma - k'_y \sin\gamma, \qquad k_y = k'_x \sin\gamma + k'_y \cos\gamma \quad . \tag{A.85}$$

When Eq. (A.80) is inserted in Eq. (A.84) we obtain

$$F'\left(k'_x, k'_y\right) = \left(\frac{ah}{2}\right) e^{-|k'_x| a}\, \delta\left(k'_y\right) \quad . \tag{A.86}$$

Using the inversion formula with $C(k_x, k_y) = ik_x u_o F(k_x, k_y)$ yields the following expression for $\bar{w}(x, y, z)$:

$$\bar{w}(x,y,z) = u_o \iint (ik_x)\, e^{ik_x x}\, e^{ik_y y}\, F(k_x, k_y)\, e^{-\left(k_x^2 + k_y^2\right)^{1/2} z}\, dk_x\, dk_y$$

$$= u_o \frac{\partial}{\partial x} \iint e^{i\left(k'_x x' + k'_y y'\right)} F'\left(k'_x, k'_y\right) e^{-\left(k'^2_x + k'^2_y\right)^{1/2} z}\, dk'_x\, dk'_y \quad . \tag{A.87}$$

Using Eq. (A.86) permits integration of Eq. (A.87) and we thus obtain

$$\bar{w}(x,y,z) = -2u_o(ah)\,\lambda \cos\gamma \frac{x'}{\left(x'^2 + \lambda^2\right)^2} \quad , \tag{A.88}$$

where

$$x' = x\cos\gamma + y\sin\gamma \quad ,$$

and (A.89)

$$\lambda = z + a \quad .$$

It is also easy to show that the x and y components of the perturbed velocity field are given by

$$\bar{u}(x,y,z) = -u_o(ah)\cos^2\gamma \left[\left(x'^2 - \lambda^2\right)/\left(x'^2 + \lambda^2\right)^2\right] \quad , \tag{A.90}$$

and

$$\bar{v}(x,y,z) = -u_o(ah)\cos\gamma\sin\gamma \left[\left(x'^2 - \lambda^2\right)/\left(x'^2 + \lambda^2\right)^2\right] \quad . \tag{A.91}$$

Since $\bar{u}$, $\bar{v}$, and $\bar{w}$ depend only on x', the origin of the system can conveniently be located anywhere along the crestline

An assessment of the range of validity of the theory can be rendered by examining the ratio, r, of the magnitude of the perturbed velocity to u_o when the flow is perpendicular to the crestline (i.e., $\gamma = 0$). In this case we have

$$\left(|\Delta \underset{\sim}{v}| / u_o \right) = \left(\bar{u}^2 + \bar{w}^2 \right)^{1/2} / u_o = r = ah / \left[x^2 + (z + a)^2 \right] , \qquad \text{(A.92)}$$

where the altitude z is defined in the range $z \geq f(x) = h/(1 + (x/a)^2)$. For a preassigned value of the half width, a, the requirement that r be much less than unity over all space establishes an upper bound to h which will render the results consistent with the perturbation theory. A good measure of this upper limit can be obtained by evaluating r at the top of the ridge ($x = 0$, $z = h$). Thus we have

$$r_t = ah/(h + a)^2 = (h/a) / \left[1 + (h/a)^2 \right] . \qquad \text{(A.93)}$$

Examination of the foregoing expression shows that r_t is less than unity regardless of the ratio (h/a). Thus one would conclude that the first-order perturbation theory would work under all cases, even including an infinitely steep mountain ridge. This obviously cannot be the case. Apparently, higher order corrections as computed by the iteration scheme are necessary to establish by analytical techniques the range of validity of the calculation. As we shall show in the next section, however, the limitations of the first-order theory can also be assessed by graphical means; that is, by examination of the computed wind streamlines.

It is interesting to note in passing that when the theory is applied to a valley, in which case h is replaced by $-|h|$ and the ratio, r, is computed at the bottom of the valley ($x = 0$, $z = -|h|$), we obtain

$$r_b = a|h|/(a - |h|)^2 , \qquad \text{(A.94)}$$

which clearly shows that the theory is valid only for a small slope, $|h| < a$.

A Single Mountain

Investigations have shown that a suitable repr^esentation for a mountain is

$$z = f(x, y) = \frac{h\left(a^3\right)}{\left(a^2 + r^2\right)^{3/2}} = \frac{A}{\left(a^2 + r^2\right)^{3/2}} , \qquad (A.95)$$

where h is the maximum elevation of the mountain, $r^2 = x^2 + y^2$, and a is its characteristic dropoff rate (when $r = a$, $f = 0.35\ h$).

Although one can construct several mountain functions which are similar to f(x, y), this particular function was selected because it ultimately yields analytic expressions for the perturbed components of the wind field. The Fourier transform of f(x, y) is

$$F\left(k_x, k_y\right) = \frac{A}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-i\underset{\sim}{k} \cdot \underset{\sim}{r}} \frac{1}{\left(a^2 + r^2\right)^{3/2}} \, dx\, dy ,$$

$$F\left(k_x, k_y\right) = F(k) = \frac{A}{(2\pi)k^{1/2}} \int_0^{\infty} \frac{r^{1/2}}{\left(a^2 + r^2\right)^{3/2}} J_o(kr)(kr)^{1/2} dr , \qquad (A.96)$$

where

$$k = \left(k_x^2 + k_y^2\right)^{1/2} .$$

The integral in Eq. (A.96) is recognized (Ref. A.3) as the Hankel transform of $r^{1/2}(r^2 + a^2)^{-3/2}$, so that F(k) becomes

$$F(k) = \frac{A}{2\pi a} e^{-ak} . \qquad (A.97)$$

If u_o is the unperturbed velocity, it follows that the first-order correction to the vertical component of the wind is given by

$$\bar{w}(x, y, z) = u_o \iint \left(ik_x\right) F(k)\, e^{i\left(k_x x + k_y y\right)} e^{-kz}\, dk_x\, dk_y \quad ,$$

$$\bar{w}(x, y, z) = \frac{u_o A}{a} \frac{\partial}{\partial x} \int_0^\infty e^{-(a + z)k} J_o(kr)\, kdk \quad ,$$

$$\bar{w}(x, y, z) = \frac{-3\lambda A u_o}{a} \frac{x}{\left(x^2 + y^2 + \lambda^2\right)^{5/2}} = \frac{-3\lambda a^2 h x u_o}{\left(r^2 + \lambda^2\right)^{5/2}} \quad , \tag{A.98}$$

where

$$\lambda = (z + a) \quad .$$

The changes in the x and y components of velocity are determined from Eqs. (A.50) and (A.51). Thus, we have:

$$\bar{u}(x, y, z) = -u_o \iint \frac{\left(ik_x\right)\left(ik_y\right)}{k} F(k)\, e^{i\left(k_x x + k_y y\right)} e^{-kz}\, dk_x\, dk_y$$

$$\bar{u}(x, y, z) = -u_o \frac{\partial}{\partial x^2} \int_0^{2\pi} \int_0^\infty F(k)\, e^{ikr \cos \phi} e^{-kz}\, dk\, d\phi \quad ,$$

$$\bar{u}(x, y, z) = u_o\left(a^2 h\right) \frac{\left(y^2 + \lambda^2 - 2x^2\right)}{\left(r^2 + \lambda^2\right)^{5/2}} \quad ; \tag{A.99}$$

and

$$\bar{v}(x, y, z) = -u_o \iint \frac{(ik_y)(ik_x)}{k} F(k)\, e^{i(k_x x + k_y y)}\, e^{-kz}\, dk_x\, dk_y \ ,$$

$$\bar{v}(x, y, z) = -u_o \frac{\partial^2}{\partial x\, \partial y} \int_0^{2\pi} \int_0^{\infty} F(k)\, e^{ikr \cos\phi}\, e^{-kz}\, dk\, d\phi \ .$$

$$\bar{v}(x, y, z) = -\frac{3u_o\left(a^2 h\right)xy}{\left(r^2 + \lambda^2\right)^{5/2}} \ . \qquad (A.100)$$

Wind Streamlines and Fallout Particle Streamlines in Two Dimensions

It is of interest to obtain analytic expressions and pictorial representations for the trajectories of fallout particles for the purpose of assessing the importance of the terrain effects. If u_p and w_p denote the horizontal and vertical components of the fallout particle velocity in a two-dimensional system, it is well known that these quantities are related to the wind velocity through the equations

$$u_p = u_o + \bar{u} \ , \qquad (A.101)$$

and

$$w_p = -V_F + \bar{w} \ , \qquad (A.102)$$

where V_F is the so-called fall velocity. Strictly speaking V_F is a function both of particle size and of altitude (Ref. A.4), but below 10,000 ft its variation with z can be neglected. Figure A.2 shows the average fall velocity, between 0 and 10,000 ft, plotted as a function of particle size for spherical particles with an assumed density of 2.5 g cm^{-3}.

The flow lines, or trajectories, of fallout particles can be determined from a quantity called the stream function,[A.5] $\Phi(x, z)$, defined by the partial differential equations

$$w_p = -V_F + \bar{w} = -\frac{\partial \Phi}{\partial x}(x, z) \ , \qquad (A.103)$$

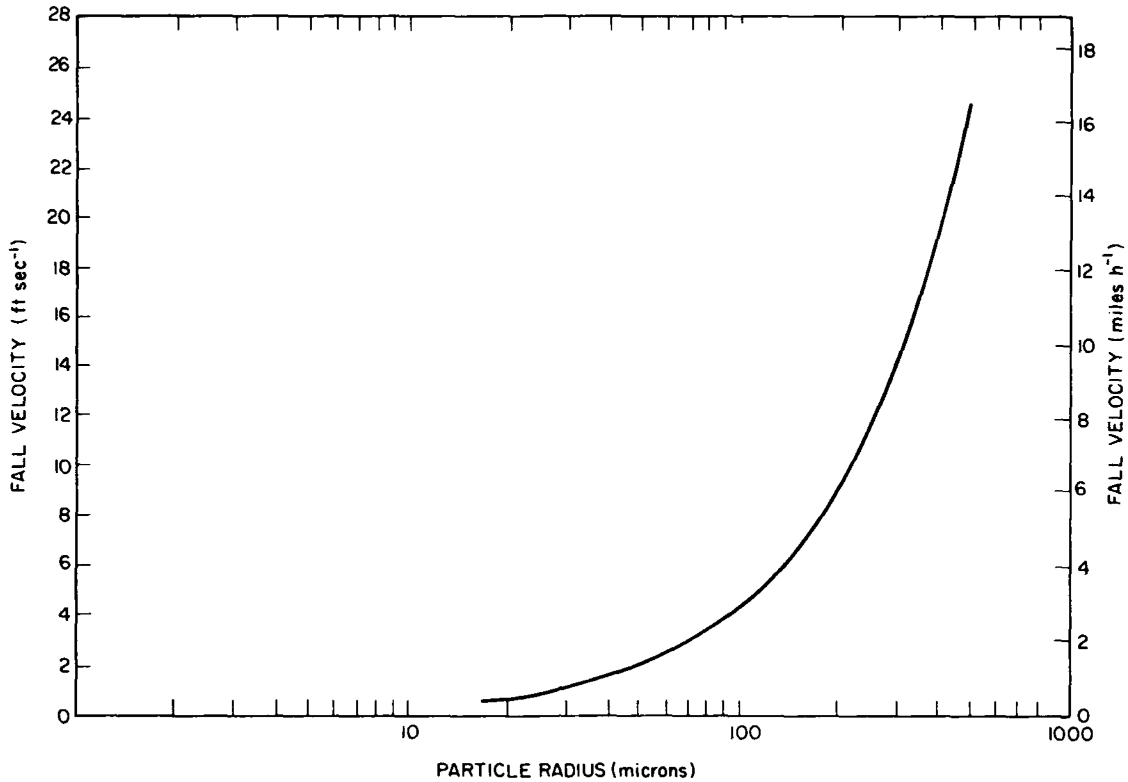


Figure A.2. Fall Velocity vs Particle Size

and

$$u_p = u_o + \bar{u} \equiv \frac{\partial \Phi}{\partial z}(x, z) \quad . \qquad \text{(A.104)}$$

On the other hand, the equation for the flow lines is

$$\frac{dx}{u_p} = \frac{dz}{w_p} \quad \text{or} \quad -w_p dx + u_p dz = 0 \quad , \qquad \text{(A.105)}$$

which when used with Eqs. (A.103) and (A.104) gives

$$\frac{\partial \Phi}{\partial x} dx + \frac{\partial \Phi}{\partial z} dz = 0 \quad \text{or} \quad d\Phi = 0 \quad . \qquad \text{(A.106)}$$

We thus see that the curves for which

$$\Phi = \text{constant} \tag{A.107}$$

are the streamlines. In the case of a mountain ridge perpendicular to the unperturbed flow it is easy to show that $\bar{u}$ and $\bar{w}$ are given by

$$\bar{u} = \frac{\partial \Psi}{\partial z} \ , \tag{A.108}$$

and

$$\bar{w} = -\frac{\partial \Psi}{\partial x} \ , \tag{A.109}$$

where the perturbed wind stream function $\Psi(x, z)$ is

$$\Psi = \frac{-u_o(ah)(z + a)}{(z + a)^2 + x^2} \ . \tag{A.110}$$

Using Eqs. (A.108) and (A.109), and recalling that u_o and V_F are both independent of position, enables us to construct the entire stream function, $\Phi(x, z)$. The function $\Phi(x, z)$, which has as its partial derivates w_p and u_p, is

$$\Phi = u_o z + \Psi + V_F x \ . \tag{A.111}$$

For computational purposes it is desirable to cast Eq. (A.111) in dimensionless form by dividing both sides by $(u_o a)$. The streamlines are then given by the equation

$$\text{constant} = C = \bar{z} + r\bar{x} - \alpha \frac{(1 + \bar{z})}{(1 + \bar{z})^2 + \bar{x}^2} \ , \tag{A.112}$$

where

$\alpha = h/a$ = ratio of height to half width of mountain ridge,

$r = V_F/u_o$ = ratio of the magnitude of fall velocity to the unperturbed wind velocity, u_o,

$\bar{x} = (x/a)$ = horizontal dimension in units of a,

$\bar{z} = (z/a)$ = vertical position in units of a.

In the absence of a mountain ridge, $\alpha = 0$ and Eq. (A.112) reduces to

$$\bar{z} = C' - r\bar{x} \quad , \tag{A.113}$$

which is the equation of the straight-line descent of a fallout particle with slope $-V_F/u_o$. On the other hand, the streamlines deduced from Eq. (A.112) when $r = 0$ depict the flow of the wind field over the mountain ridge. The solid lines in Figures A.3, A.4, and A.5 show the wind streamlines at various initial altitudes for values of $\alpha = 0.25$, 0.50, and 0.75. The dashed line on each figure is the mountain ridge function

$$\left(\frac{z}{a}\right) = \left(\frac{h}{a}\right) \frac{1}{1 + (x/a)^2} = \frac{\alpha}{1 + (x/a)^2} \quad .$$

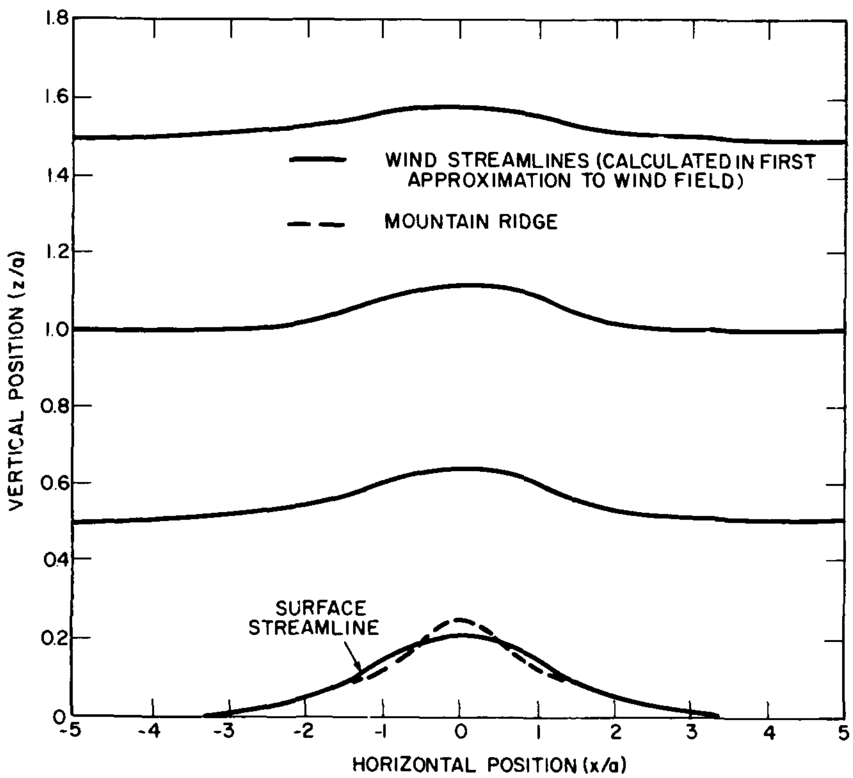


Figure A.3. Wind Streamlines for α = 0.25 (Note vertical scale is amplified 10 times.)

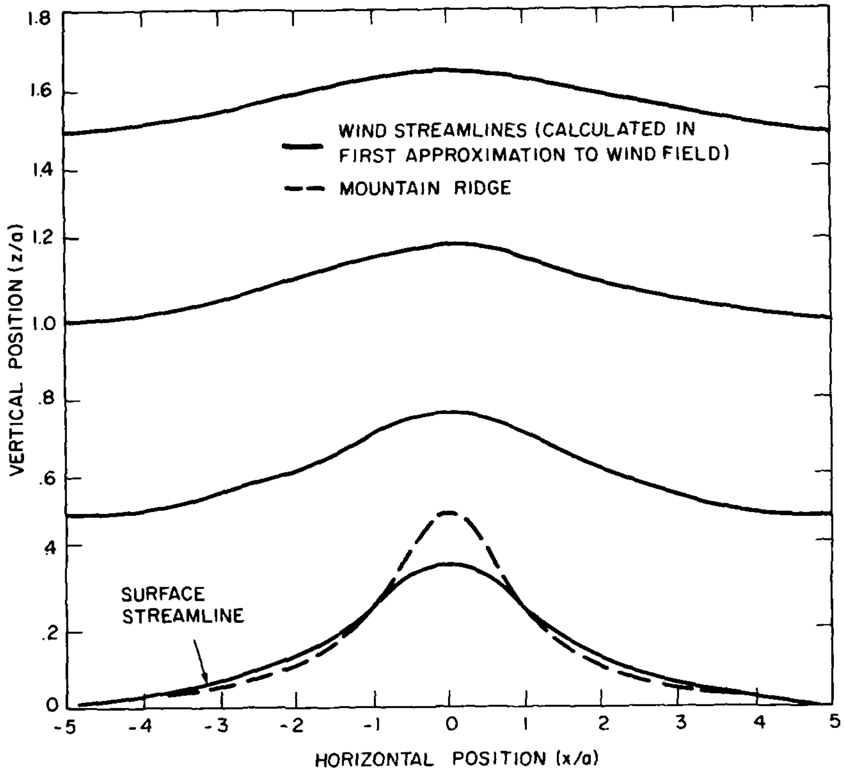


Figure A.4. Wind Streamlines for α = 0.50 (Note vertical scale is amplified 10 times.)

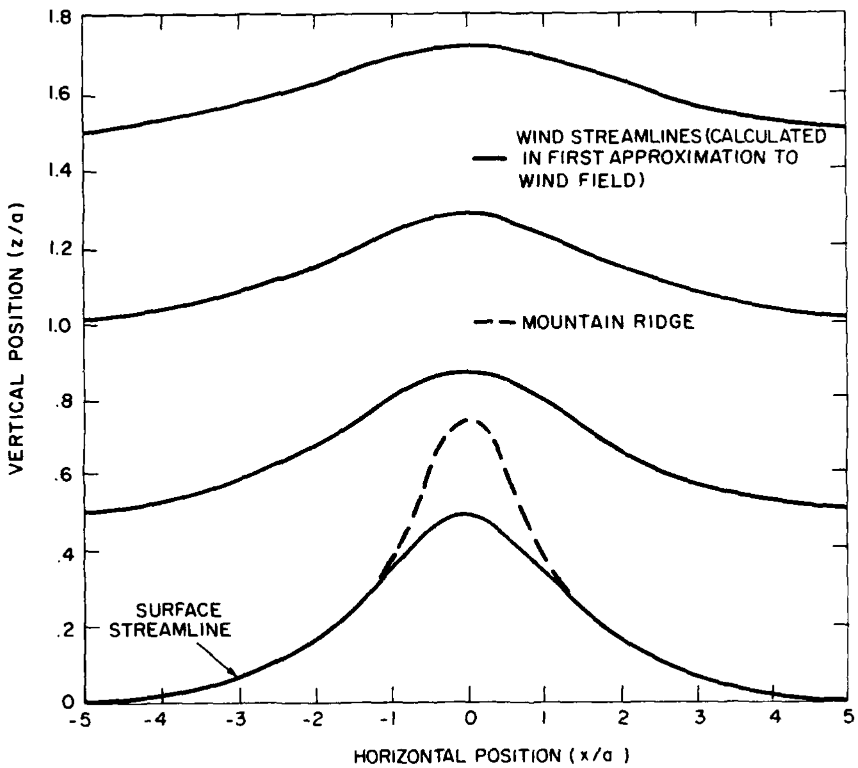


Figure A.5. Wind Streamlines for α = 0.75 (Note vertical scale is amplified 10 times.)

We can readily observe in all cases that the higher altitude streamlines are less affected by the mountains than the lower ones. This is to be expected. It is also interesting to note that the wind flow corresponding to the surface streamline actually hits the mountain ridge in all cases. This is not unexpected in view of the approximate nature of the first-order calculation of the Fourier transform of the vertical component, C(k), of the wind. It will be recalled that the perturbed boundary conditions were applied exactly (in which we required that the wind velocity be parallel to the earth's surface). However, in the process of determining the wind field, an iteration scheme was developed to compute C(k), and the results shown in this section correspond to the first approximation

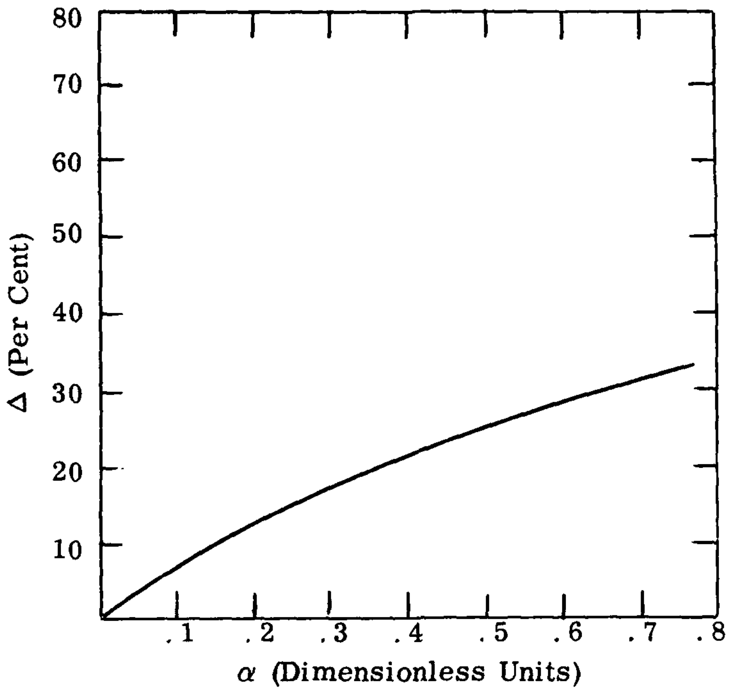
$$C^{(1)}(k) = \frac{u_o(ah)}{2}(ik)\, e^{-|k|a} \quad .$$

The uncertainties in the calculation (as noted earlier) should increase in proportion to the slope, which in the case of a mountain ridge is typified by the ratio (h/a). This is especially well borne out by the results which show that the relative difference, Δ, between the height of the mountain ridge and the maximum elevation increases with a corresponding increase in (h/a). We define Δ by the equation

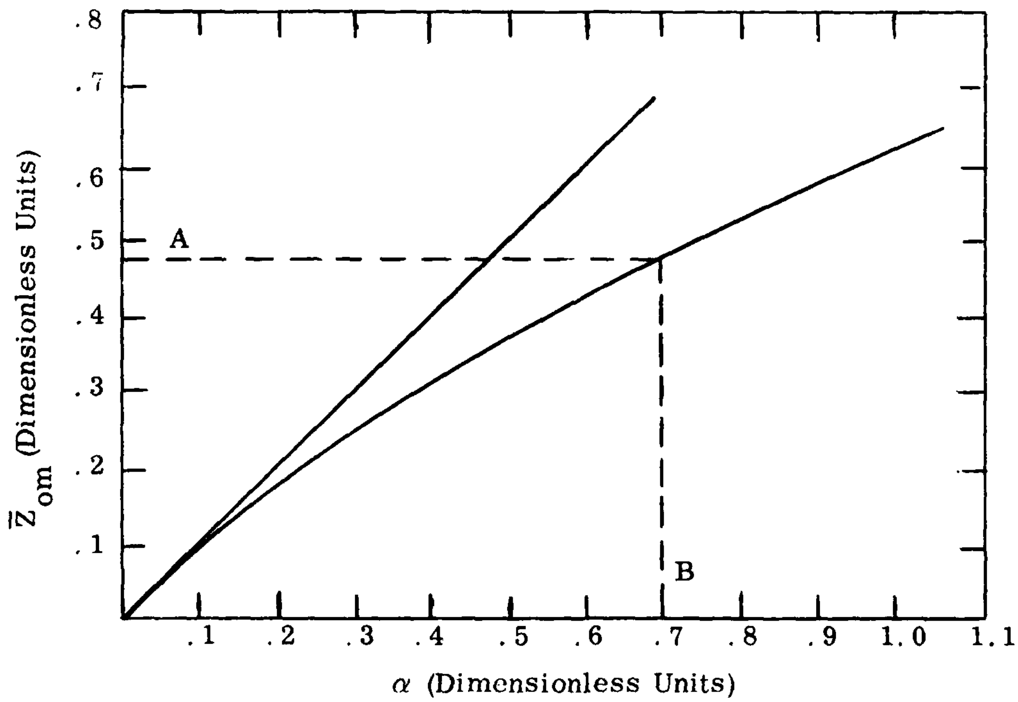
$$\Delta = \frac{(h/a) - \bar{Z}_{om}}{(h/a)} \times 100 = \frac{\alpha - \bar{Z}_{om}}{\alpha} \times 100 \quad , \tag{A.114}$$

where $\bar{Z}_{om}$ is the maximum elevation of the surface trajectory in units of a. Figure A.6(a) shows a plot of Δ vs $(h/a) = \alpha$. As observed, the relative difference of trajectories increases as (h/a) the average slope of the mountain ridge increases, thereby reflecting the uncertainties in the calculation attributable to the first-order approximation.

The results also show that the first approximation to the wind field underestimates the airlift due to the mountain ridge. This is perhaps better illustrated by a comparison of $\bar{Z}_{om}$ vs α (see Figure A.6(b)), which like Figure A.6(a) shows that the discrepancies between the actual maximum "lift" and the ideal lift increase with α. (Mathematically, this discrepancy is the difference between the 45° ideal



(a)



(b)

Figure A.6. Differences Between Surface Trajectory and Mountain Ridge lift line and the actual curve of $\bar{Z}_{om}$ vs α.) Figure A.6(b) illustrates the necessity of executing the iteration scheme for C(k) in order to ensure the proper evaluation of the wind field. Since this may be a complicated process, however, we can use the results of Figure A.6(b) to establish an empirical relationship between calculated trajectories and the true mountain profile. Thus, suppose we have a mountain ridge whose maximum elevation, also measured in units of its half width, is 0.48 (point A in Figure A.6(b)). The curve shows that to ensure that the calculated surface trajectory would actually rise to a maximum elevation of 0.48, it is necessary to perform the calculations for an α equal to 0.7 (point B in Figure A.6(b)).

The curves of the fallout particle trajectories shown in Figures A.7, A.8, and A.9 depict a mountain ridge corresponding to the maximum elevation $\bar{Z}_{om}$, although as in the case for the wind streamlines, the calculations were performed for the corresponding values of α. Since it is beyond the scope of this report to perform a parametric analysis of the stream function, we present the results for a fall-to-wind velocity ratio, V_F/u_o, equal to 0.1; the curves differ only in the choice of α,

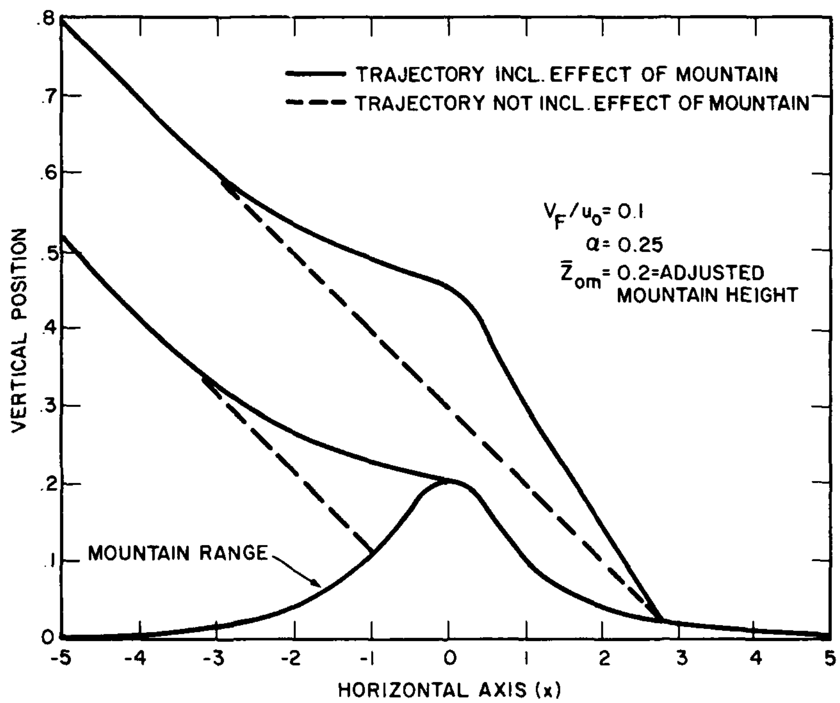


Figure A.7. Fallout Particle Trajectories (Note vertical scale is amplified 10 times; dimensions are in units of a, the half width of the mountain range.)

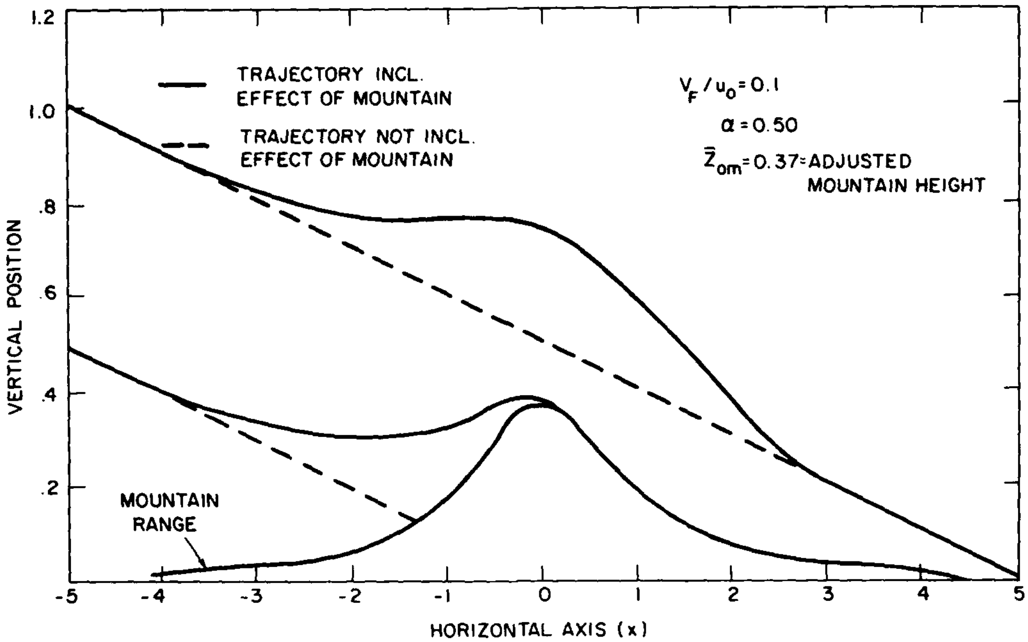


Figure A.8. Fallout Particle Trajectories (Note vertical scale is amplified 5 times; dimensions are in units of a, the half width of the mountain range.)

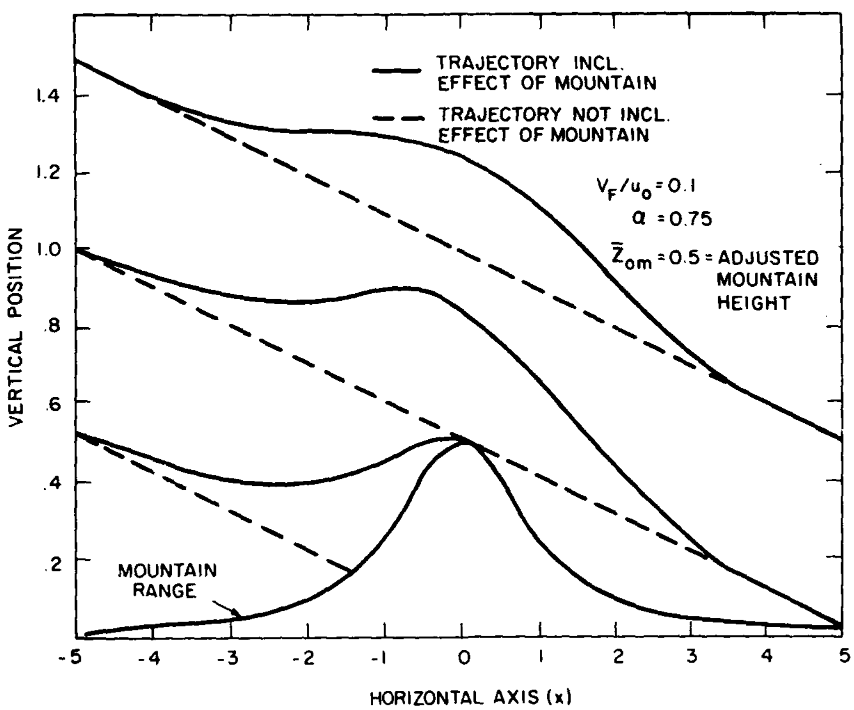


Figure A.9. Fallout Particle Trajectories (Note vertical scale amplified 5 times; dimensions are in units of a, the half width of the mountain range.)

or equivalently, $\bar{Z}_{om}$. All the curves originate at the dimensionless horizontal displacement, $(x/a) = -5$, sufficiently removed from the center of the mountain ridge so that terrain effects would not be felt. Examination of all three curves for the same initial vertical position of the streamlines shows that increases in the mountain ridge height lead to an enhanced downstream drift of the fallout particles. For those streamlines, whose unperturbed trajectories pass over the crest of the mountain ridge (e.g., the middle unperturbed trajectory of Figure A.9), the mountain ridge causes no permanent displacement of the streamlines. We have also constructed a digital program to compute the transport times for fallout particles traversing a two-dimensional mountain ridge for arbitrary γ and α and have applied the code for the cases $\gamma = 0.1$; $\alpha = 0.25$, 0.50, and 0.75 (between the limits $-5 \leq x \leq 5$). The results show no additional time delay due to the effect of the mountain ridge. On the contrary, we found a relative speedup of between 1 to 2%. Apparently, the increase in path length is slightly more than counterbalanced by the increase in velocity.

The fact that the perturbed trajectories which fail to intercept the mountain ridge always end up on the same unperturbed streamline is a general result, as can be seen by examination of the stream function,

$$C = \bar{z} + r\bar{x} - \left[\frac{\alpha(1 + \bar{z})}{(1 + \bar{z})^2 + \bar{x}^2}\right] .$$

The function in brackets is the contribution to the stream function attributable to the mountain ridge; for either large negative or positive values of $\bar{x}$ this goes to zero.

The middle set of curves in Figure A.9 can be used to obtain some estimate of the enhanced fallout range caused by the mountain ridge for $(V_F/u_o) = 0.1$ with $(\bar{Z}_{om}/a) = 0.5$. From Figure A.2 we see that this would correspond to a 100-μ particle with $u_o = 30$ mph, 150-μ particle with $u_o = 50$ mph, or a 230-μ particle with $u_o = 70$ mph, all of which are quite possible situations. If the actual mountain ridge peak were 1 mi, the intercept of the alluded-to trajectory would hit the horizontal axis at $\bar{x} = 5$ or $x = 10$ mi.

For heavy fallout particles, terrain effects are less important since the vertical lift decreases. At the other extreme, extremely light fallout particles will follow the wind streamlines.

Conditions for Neglecting the Coriolis Effect

We shall now examine the coefficients in the dispersion relationship — namely a, b, and c (see Eqs. (A.35) and (A.36)) — to determine the conditions on the velocity and wavelength for which the Coriolis effect can be neglected. Since we are primarily interested in short range effects, we shall freely make use of the assumed inequality

$$\left(k_x^2 + k_y^2\right)^{1/2} >> \beta = 0.75 \times 10^{-6}\ \mathrm{cm}^{-1} \ , \tag{A.115}$$

which signifies that the wavelength of the horizontal variations in the terrain is less than the $2\pi\beta^{-1}$, or approximately 50 mi.

The Coriolis effect can be neglected in Eq. (A.36a) if

$$|k_x| u_o >> \Omega \ , \tag{A.116}$$

or equivalently

$$(2\pi) \frac{|u_o|}{\Omega} >> L_x \ , \tag{A.117}$$

where L_x is a representative wavelength in the x direction. The distance (u_o/Ω) equals

$$\frac{u_o}{\Omega} = d = \left(3.8\, u_m\right) \mathrm{mi} \ , \tag{A.118}$$

where u_m is the wind velocity expressed in miles per hour. This restricts L_x to less than 24 u_m mi. We have assumed the inequality of Eq. (A.117) to hold in our analysis. Equation (A.117) will not be satisfied when the unperturbed velocity is too large or when both a small u_o and large L_x are combined. However, these conditions are not important relevant to fallout considerations. If L_x is large, the disturbance cannot be considered as a local effect; hence, additional meteorological information will be available, thus precluding the utility of the perturbation methods considered here. On the other hand if u_o is small, the perturbed wind velocity will also be small (as is shown in the analysis) and terrain effects will not be important since the motion of the fallout particle will be essentially vertical.

Examination of Eqs. (A.36b) and (A.36c) shows that the second and third coefficients in the dispersion relationship, b and c, are already expanded in powers of Ω and thus are in a suitable form for examining the conditions under which the Coriolis effect may be neglected. (Neglecting the Coriolis effect is equivalent to omitting those terms which are proportional to Ω and Ω^2.) The ratio of the first-to-second terms in Eq. (A.36b) is

$$\frac{\gamma\beta k_x^2 u_o^2}{2\Omega|k_x||k_y|u_o\omega_x} = \frac{\gamma\beta L_y u_o}{4\Omega L_x \cos\Theta \sin\epsilon} , \qquad \text{(A.119)}$$

where L_y and L_x are characteristic wavelengths for the y and x dimensions respectively. Neglecting minor numerical factors, the condition that the foregoing relationship be greater than unity is approximately given by

$$0.5\left(L_y/L_x\right) u_m > 1 . \qquad \text{(A.120)}$$

For physically interesting wind velocities (e.g., $u_m > 10$ mph) the inequality of Eq. (A.120) will break down only in unusual cases. Although such occurrences can be treated by the general theory we also assume the inequality of Eq. (A.120).

Using Eq. (A.115) together with the assumption $k_x > k_y$ gives for the ratio of the first-to-third terms in the expression for b (again neglecting minor numerical factors)

$$\left(k_x^2 u_o^2/\Omega^2\right)\frac{\beta}{|k_x|} = r_1 . \qquad \text{(A.121)}$$

Using Eq. (A.118) we find that the requirement that the foregoing expression is much greater than unity is given by

$$2\pi (1.8)\, u_m^2 >> L_x \text{ (miles)} \quad . \tag{A.122}$$

The cases for which $L_x > 2\pi (1.8)\, u_m^2$ are not relevant to local fallout. If we now use Eq. (A.115) but assume $k_y > k_x$, we have in lieu of Eq. (A.119)

$$\frac{k_x^2 u_o^2}{\Omega^2} \frac{\beta}{|k_y|} = r_1\left(L_y/L_x\right) = \left[4\pi (1.8)\, u_m\right] \left(0.5\, L_y/L_x\right) u_m \quad . \tag{A.123}$$

Since $(0.5\, L_y u_m/L_x)$ is assumed greater than unity, the foregoing expression will also be greater than unity.

Recalling that $\sigma = (\sigma/\beta) = 1.2 \times 10^9\ \text{cm}^2\ \text{sec}^{-2} >> u_o^2$, and neglecting minor numerical factors, gives the following approximate expression for c:

$$c \simeq k_x^2 u_o^2 \sigma \left(k_x^2 + k_y^2\right) + \Omega\beta\sigma u_o k_x^2 + \Omega^2 \sigma k_x^2 \quad . \tag{A.124}$$

By using the previously mentioned inequalities it is easy to show that Ω and Ω^2 can be neglected in the expression for c.

Conclusions

A perturbation type model has been developed to compute the airflow over variable terrain. The theory is based on the assumption of the existence of an unperturbed state characterized by an adiabatic atmosphere and a uniform velocity, u_o, which would otherwise exist in the absence of ground variations. When the general theory is addressed to small scale disturbances, which are of interest in local fallout applications, there are no lee waves and we can neglect the Coriolis force. In this regime the theory has been applied to compute the wind field over a mountain (valley) and a mountain (valley) ridge. Using the calculated wind streamline for a mountain ridge it becomes possible to assess the importance of variable terrain on the motion of particles typical of those encountered in the nuclear fallout regime.

References

A. 1 P. Queney, "Theory of Perturbations in Stratified Currents with Applications to Air Flow Over Mountain Barriers," Miscellaneous Reports of the Department of Meteorology of Chicago, 1947. (Available from MIT Library.)

A. 2 P. Queney, "The Problem of Air Flow Over Mountains: A Summary of Theoretical Studies," Bull. Am. Meteorol. Soc. 29, 16-26 (1948).

A. 3 A. Erdelyl, et al., Tables of Integral Transforms (New York, N. Y.: McGraw-Hill Book Company, Inc., 1954), Vol. 2.

A. 4 W. W. Kellogg, R. R. Rapp, and S. M. Greenfield, "Close-In Fallout," J. Meteorol. 14, 1-8 (1957).

A. 5 P. Morse and H. Feshback, Methods of Theoretical Physics (New York, N. Y.: McGraw-Hill Book Company, Inc., 1953).

APPENDIX B

THE INCORPORATION OF THE SEA BREEZE IN THE CALCULATION OF FALLOUT

Introduction

The effects of the sea breeze will be considered in our calculations of fallout. It is appreciated that this local circulation phenomenon can have an important effect on the lighter fallout particles, particularly on a clear day, when the temperature-induced circulating winds are larger than the so-called fall velocity, V_F.

The sea breeze is characterized by relatively large changes in the wind direction over short distances, and, as such, its internal features cannot be satisfactorily analyzed with existing installations because of the unavailability of sufficiently dense meteorological observation points in the vicinity of the coastline. We have developed, therefore, a suitable sea breeze model which can be applied in a digital computer program leading to the determination of fallout distribution. As in the overall DOD fallout model, space is divided into cells, with each compartment characterized by a distinct wind field. In the general case, the wind parameters for these cells are deduced (by suitable mathematical techniques) from sounding stations which are in close proximity to the geometric center of the cell. The construction of the cell's wind field by this method is appropriate throughout most of space where changes in the wind velocity occur over dimensions which are large compared to the distance between observation points. The sea breeze and other local circulation systems such as mountain and valley winds, however, cannot be treated by this method. Consequently, the geometric region enclosing the sea breeze is divided into a special cell which is treated separately. The nature of the problem dictates that analytic mathematical functions be used to generate the wind field in this cell. These functions, moreover, should be applicable for most situations.

Review of the Sea-Breeze Theories

The sea breeze is perhaps one of the best examples of an atmospheric process which can be treated analytically with a degree of success. Jeffreys[B.1] was the first to treat the problem in an exact way, although his results were not in full agreement with observations. As pointed out by Schmidt,[B. 2] the former's model led to a solution in which the daily wind variation was in phase with the daily temperature curve. This was not always consistent with the measured results. In the Jeffreys model, the only forces that are taken into account are the two due to friction and to the pressure gradient resulting from the unequal heating. Haurwitz[B. 3] classifies such a model as an equilibrium theory of the sea breeze — a theory which neglects the inertia of the wind and, consequently, the temporal changes of the wind that are of the order of $\Omega = 2\pi/(\text{sidereal day}) = 7.3 \times 10^{-5}$ sec. In retrospect, the main flaw in Jeffreys treatment was not so much his neglect of the inertia of the wind (as will be shown later, this can be justified in some cases) but rather his deletion of the Coriolis terms which account for the veering of the wind in the course of time. This effect was included in the subsequent papers.

The works of both Schmidt and Haurwitz were less concerned with rendering complete theory of the sea breeze than with clarifying the characteristic phenomena of the land and sea breezes, such as the phase shift between wind and temperature or the influence of the earth's rotation. These investigations did much to improve our understanding of the sea breeze, but they cannot be considered as complete in the usual sense. (A more thorough critique of their work is given by Defant,[B. 4] who also discusses research performed by other investigators.)

In analytical treatments of the sea breeze, it is necessary to make simplifying assumptions in order to obtain mathematically tractable equations. We can categorically say that all the analytical treatments are based upon linearization of the equations of motion which describe the sea-breeze circulation. The more complete analytical treatments of the sea breeze have been successful in accounting for the large scale characteristics of the sea-breeze circulation. Notable among this group are the investigations of Defant[B. 4, B. 5] and Haurwitz,[B. 6] which form the basis of the sea-breeze model used in our fallout computation. (A discussion of their work is given in the following section of this appendix.) Generally, the terms in the dynamical equations which deal with the horizontal advection of temperature are omitted, although in the Defant-Haurwitz models, vertical advection of temperature is retained, and the diffusion of heat upward by turbulent processes is included. Despite the approximations resulting from linearization, the linear models do yield a satisfactory reproduction of the fundamental field of motion of the sea breeze.

The development of high speed numerical methods has made possible a more refined treatment of the sea breeze which can account for not only horizontal advection but also the spatial variation of the viscosity and turbulent diffusion constants. Pierce[B.7] was perhaps the first who succeeded in integrating by numerical methods a set of nonlinear sea breeze equations. The main drawback in Pierce's model was his introduction of a somewhat artificial mechanism to transfer the heat absorbed by the earth to the atmosphere. The physical consequences of this are more fully discussed by Fisher.[B.8] As pointed out by Fisher, the most important feature of the numerical method lies in the fact that the nonlinear advective terms in the equations may be retained and thus allow the feedback effect of the wind field itself on the sea breeze to be studied. Fisher's model is conceptually identical to the linear model of Haurwitz and may be considered the most definitive work in the field inasmuch as it includes not only nonlinear horizontal advection but also the spatial variation of the transport parameters. This solution shows the sea breeze in the stages of development and decay and succeeds in reproducing the gross features of the wind system and many of its small details as well.

The main drawback in applying Fisher's model to the fallout problem is its sheer complexity, particularly in view of the fact that, as pointed out by Fisher himself, its principal contribution is its ability to describe the fine structure in the sea-breeze development. Although we can justify the incorporation of the sea breeze in fallout models, we are hard-pressed to justify the inclusion of its subleties. Other effects such as the irregularity of the coastline, the presence of a prevailing wind, and uncertainties in the transport coefficients would completely overshadow any improvement attributed to incorporation of the sea-breeze fine structure. (Recently, an attempt was made by Travelers Insurance Research Laboratory[B.9] to employ Fisher's observed data for calculation of fallout in a sea breeze In view of the extensive amount of "function-fitting" employed, it becomes difficult to appraise their model.)

Sea-Breeze Model

Wind-Field Parameters

In this section, we shall present the expressions for the components of the wind field that are used in our sea breeze calculations and are identical to those deduced by Defant.[B.5] The derivation of our final results, however, closely resembles Haurwitz's treatment of the sea-breeze circulation because it shows more clearly the assumptions which are made concerning the pressure variation.

Defant's approach to the sea-breeze problem is based on Lord Rayleigh's convection theory.[B.10] The dynamics of Defant's model are governed by the continuity equation, the three equations of motion, the equation of state, and the heat-diffusion equation. By neglecting variations in density except in so far as they modify the action of gravity, it becomes possible to construct a stream function which is used to describe the motion in the plane perpendicular to the coast. The mathematical equations are based on the assumption of an infinitely long coastline which we designate as the y axis. Variations of the meteorological equations in this direction are ignored. The x axis is perpendicular to the coast, and positive inland, while the z axis denotes the vertical. The equations which describe the system are:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad , \tag{B.1}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + w\frac{\partial u}{\partial z} - fv = -\frac{1}{\rho}\frac{\partial p}{\partial x} - \sigma u \quad , \tag{B.2}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + w\frac{\partial v}{\partial z} + fu = -\sigma v \quad , \tag{B.3}$$

$$\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + w\frac{\partial w}{\partial z} = -\frac{1}{\rho}\frac{\partial p}{\partial z} - g - \sigma w \quad . \tag{B.4}$$

$$p = \rho RT \quad . \tag{B.5}$$

and

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + w\frac{\partial T}{\partial z} = K\frac{\partial^2 T}{\partial z^2} \quad , \tag{B.6}$$

where u, v, and w are the velocity components along the x, y, and z axes, respectively; p denotes the pressure; T is the temperature; ρ is the mass density; g is the gravitational constant; K is the thermal diffusion constant; and R is the gas constant. The quantity $f = 2\Omega \sin \phi$ is the Coriolis parameter, while the effect of friction is taken in account through the Guldberg-Mohn friction parameter, σ. To be sure, this is the simplest way to incorporate the effect of viscosity into the theory. (Haurwitz offers an alternative approach to turbulent dissipation but, as we shall discuss later, this has its own drawbacks.) With the exception of Eq. (B.1) (the continuity equation derived by setting $(d\rho/dt) = 0$, see Ref. B.8) all the others are nonlinear in the sense that there are terms which involve multiplication of the dependent meteorological variables. Application of the following boundary conditions suffices to determine the problem in all cases:

$$
\begin{aligned}
w(z = 0) &= 0 \quad , \\
w(z \to \infty) &= 0 \quad , \qquad \text{(B.7)} \\
T(z = 0) &= T_o + T(x, t) \quad .
\end{aligned}
$$

The function $T(x, t)$ is the surface temperature differential, which is defined as the difference between the actual temperature above the water or land, and a suitable reference temperature, T_o, which we take to be the temperature along the coastline. In the theory of the sea breeze $T(x, t)$ performs the role of the "driving-force" in that it, alone, is responsible for the circulation.

Before proceeding with our discussion of the solution of the sea-breeze equations, it is appropriate to review a variation of the sea-breeze model as rendered by Haurwitz.[B.6] The difference between Haurwitz's model and Defant's lies in the method of treating turbulent friction. Instead of using the Guldberg-Mohn friction parameter, σ, to describe turbulent dissipation, Haurwitz employs kinematic viscosity. Thus, in lieu of the terms, $-\sigma u$ and $-\sigma v$, which appear in our Eqs. (B.2) and (B.3), his corresponding friction terms are $K\partial^2 u/\partial z^2$ and $K\partial^2 v/\partial z^2$, where the kinematic viscosity K is assumed to be independent of position. Haurwitz also neglects the viscous effects on the vertical wind component; we do not. When viscosity is introduced by the expressions $K\partial^2 u/\partial z^2$ and $K\partial^2 v/\partial z^2$, which are then used with the boundary conditions $u(z=0) = v(z=0) = 0$, we arrive at a sea-breeze model in which a boundary layer (in the sense of Schlicting and Prandtl) is built into the theory. In such a situation, the horizontal components of velocity increase with altitude from a minimum value of zero at the land and water surface. According to Haurwitz's model, the distance over which this buildup occurs is of the order of the characteristic height of the sea breeze. This seems to be somewhat inconsistent with the everyday experiences at the ocean front where strong horizontal winds are evident a few feet from the ground. Strictly speaking, when boundary layer theory is used, the temperature of the moving fluid at the boundary is the same as the surface temperature Thus, if the theory of the boundary layer were rigorously applied on a clear sunny day in the summertime, we would necessarily have to use a land temperature of about 90° - 100°F and a water temperature between 60° - 70°F. This corresponds to a temperature differential of about 20°C, which would produce wind velocities greater than those measured. In addition, according to the usual boundary layer theory, this is also the surface air temperature differential. Again, this is inconsistent with observations. Haurwitz's treatment of friction thus seems to lead to inconsistencies, at least in the lower regions of the sea breeze. It is also more complicated since it introduces a much more cumbersome expression for the vertical attenuation constant. Consequently, the fundamental equations which describe our system are based on the Defant sea breeze model.

Equations (B. 1) - (B. 6) can be simplified by introducing two new variables, the stream function ψ and the vorticity η, which are related to the x and z component of the velocity by

$$u = -\frac{\partial \psi}{\partial z}, \quad w = \frac{\partial \psi}{\partial x} \tag{B. 8}$$

$$\eta = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} = -\nabla^2 \psi \tag{B. 9}$$

By operating on Eq. (B.2) with $(\partial/\partial z)$ and on Eq. (B.4) with $(\partial/\partial x)$, and then subtracting the resulting second expression from the resulting first expression, gives us the following equation for η:

$$\frac{\partial\eta}{\partial t} + \left(u\frac{\partial}{\partial x} + w\frac{\partial}{\partial z}\right)\eta - f\frac{\partial v}{\partial z} = -\frac{1}{\rho^2}\left(\frac{\partial\rho}{\partial z}\frac{\partial p}{\partial x} - \frac{\partial\rho}{\partial x}\frac{\partial p}{\partial z}\right) - \sigma\eta \quad . \tag{B.10}$$

The first term on the right-hand side is what Haurwitz calls the solenoid term, S, which can be simplified by use of the ideal gas law $p = \rho RT$.

$$S = -\frac{1}{\rho^2}\left(\frac{\partial\rho}{\partial z}\frac{\partial p}{\partial x} - \frac{\partial\rho}{\partial x}\frac{\partial p}{\partial z}\right) = \frac{R}{p}\left(\frac{\partial T}{\partial x}\frac{\partial p}{\partial z} - \frac{\partial T}{\partial z}\frac{\partial p}{\partial x}\right) \quad . \tag{B.11}$$

Since the first part of S is much larger than the second (see Ref. B.6), we have

$$S = \frac{R}{p}\left(\frac{\partial T}{\partial x}\frac{\partial p}{\partial z}\right) = -\frac{g}{T}\left(\frac{\partial T}{\partial x}\right) \quad . \tag{B.12}$$

As is usual in dynamic meteorology, we now replace $(1/T)\ (\partial T/\partial x)$ in Eq. (B.12) by $(1/\tilde{\theta})\ (\partial\tilde{\theta}/\partial x)$, where $\tilde{\theta}$ is the so-called potential temperature. In Eq. (B.6) we replace T by $\tilde{\theta}$; thus,

$$\frac{\partial\tilde{\theta}}{\partial t} + u\frac{\partial\tilde{\theta}}{\partial x} + w\frac{\partial\tilde{\theta}}{\partial z} = \frac{\partial^2\tilde{\theta}}{\partial z^2} \quad . \tag{B.13}$$

Note in Eqs. (B.6) and (B.13) that only the vertical heat conduction has been taken into account since the vertical temperature gradient is generally much larger than the horizontal gradient.

At this point, the system of equations is linearized. That is, the meteorological variables are assumed to consist of an unperturbed part, that contribution which exists in the absence of the temperature differential T(x, t); and a smaller perturbed part, attributed to the driving force. Since in the system we consider, all the initial velocities equal to zero, u, v, and w are themselves the perturbed velocities. For the potential temperature we write

$$\tilde{\theta} = \theta_o(z) + \theta(x, z, t) \quad , \qquad (B.14)$$

where θ_o and θ are the perturbed and unperturbed parts, respectively. We then arrive at a set of linearized equations:

$$-\frac{\partial}{\partial t}\left(\nabla^2 \psi\right) - f\frac{\partial v}{\partial z} = -\frac{g}{\theta_o}\frac{\partial \theta}{\partial x} + \sigma\nabla^2 \psi \ , \qquad (B.15)$$

$$\frac{\partial v}{\partial t} - f\frac{\partial \psi}{\partial z} = -\sigma v \quad , \qquad (B.16)$$

and

$$\frac{\partial \theta}{\partial t} + \Gamma\frac{\partial \psi}{\partial x} = K \frac{\partial^2 \theta}{\partial z^2} \quad , \qquad (B.17)$$

where

$$\Gamma = \frac{\partial \theta_o}{\partial z} \quad .$$

Specifically, the convection terms such as $u(\partial u/\partial x)$, $u(\partial w/\partial x)$, and $w(\partial u/\partial z)$ have been neglected in the derivation of Eqs. (B.15) - (B.17). The justification for this can be examined by a comparison of their importance with the corresponding friction term. For example, let us compare the anticipated numerical value of the convection operator $D_u = u(\partial/\partial x) + w(\partial/\partial z)$ with σ, the Guldberg-Mohn parameter in Eq. (B.2). Roughly speaking, D_u can be assigned a value approximately equal to:

$$D_u \approx \frac{\bar{u}}{L_x} + \frac{\bar{w}}{L_z} \quad , \qquad (B.18)$$

where $\bar{u}$ and $\bar{w}$ are suitable average values of the respective velocity components, and L_x and L_z are characteristic dimensions of the horizontal and vertical extent of the sea breeze. L_x is a given quantity in that it is known a priori, while L_z is determined from the theory. The landward range of the sea breeze is estimated by many observers to lie between 15 - 50 km in the temperate zones, while in the tropical regions it can extend from 50 - 65 km and even as high as 124 - 145 km in the interior.[B.4] Representative values for different locations are included in Table B.1. The vertical extent of the sea breeze, L_z, varies with location, but it

TABLE B.1

TYPICAL SEA BREEZE VALUES

Range (km)	Location
16-32	New England
15	Flemish Coast
20-30	Baltic Sea
30-40	Holland
40-50	Sweden
up to 50	Jutland
40	Albania
>50	Northern Coast of Java

is substantially smaller than the horizontal dimension. Its altitude varies from 150 m over medium-sized lakes to 200 - 500 m over large lakes and the coastal regions and rises to more than 1000 m in warm climates. It is also a characteristic feature of the sea breeze that the horizontal velocity greatly exceeds the vertical component. Under a set of conditions which gave results consistent with observation, Defant found an average horizontal velocity component of $\bar{u} = 2$ m sec^{-1} for every centigrade degree of temperature difference as opposed to a corresponding value of $\bar{w} = 2$ cm sec^{-1} $^{o}C^{-1}$. If these results are used in Eq (B.18) with $L_x = 20$ km, $L_z = 500$ m, and a maximum temperature differential of $5^{o}C$ is assumed, we obtain the following value of D_u

$$D_u \approx \frac{10}{20 \times 10^3} + \frac{0.1}{500} \simeq 7 \times 10^{-4} \text{ sec}^{-1} \qquad (B.19)$$

Unfortunately, this is greater than a realistically high value of $\sigma = 2.5 \times 10^{-4}$, so that we cannot unequivocally disregard the nonlinear terms based upon the rough estimate of D_u. It is possible that phase differences between the constituents of the operator $D_u(u, \partial/\partial x, w, \partial/\partial z)$ can lead to cancellations, thereby precluding the use of a meaningful average. Despite this seeming contradiction, the remarkable feature of Defant's model is that it works. Apparently, the nonlinear terms do not significantly alter the main features of the sea breeze.

Within the altitude range for which the sea breeze is important, the potential temperature θ_o can be considered constant in Eq. (B.15), and its derivative at equilibrium, Γ, a constant in Eq. (B.17). This procedure renders Eqs. (B.15) - (B.17) linear with constant coefficients, and thus amenable to a solution by separation of variables.

The solution of Eqs. (B.15) - (B.17) is achieved by first assuming that x variation of the variables is given by

$$\theta = A(z,t) \sin \lambda x \qquad (B.20)$$

$$\psi = B(z,t) \cos \lambda x \qquad \begin{cases} u = -(\partial\psi/\partial z) = -\cos \lambda x (\partial B/\partial z) \\ w = \partial\psi/\partial x = -\lambda \sin \lambda x B \end{cases} \qquad (B.21)$$

$$v = C(z,t) \cos \lambda x \; . \qquad (B.22)$$

Since the surface temperature differential can in general be represented by a Fourier series in multiples of the sidereal day frequency Ω, it follows from the principle of linear superposition that A, B, and C will be given by

$$A(z,t) = \sum_{n=1}^{\infty} A_n(z)\, e^{in\Omega t} \; , \qquad (B.23)$$

$$B(z,t) = \sum_{n=1}^{\infty} B_n(z)\, e^{in\Omega t} \; , \qquad (B.24)$$

and

$$C(z,t) = \sum_{n=1}^{\infty} C_n(z)\, e^{in\Omega t} \; . \qquad (B.25)$$

Combining Eqs. (B.20) - (B.25), inserting them into Eqs. (B.15) - (B.17), and equating equal powers of exp (in Ωt) gives the following coupled equations for A_n, B_n, and C_n:

$$(\sigma + in\Omega)\left(B_n'' - \lambda^2 B_n\right) + f C_n' = \alpha\lambda A_n \quad , \tag{B.26}$$

and

$$(\sigma + in\Omega)\, C_n = f B_n' \quad , \tag{B.27}$$

$$in\Omega\, A_n - \lambda\Gamma B_n = K A_n'' \quad , \tag{B.28}$$

where

$$\alpha = g/\theta_o \quad .$$

For computational purposes it is more convenient to deal with functions $W_n(z)$ defined by the equation

$$W_n = -\lambda B_n \quad , \tag{B.29}$$

in terms of which the velocity components are given by

$$w(x,z,t) = \sin \lambda x \sum_{n=1}^{\infty} W_n(z)\, e^{in\Omega t} \quad , \tag{B.30}$$

$$u(x,z,t) = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} W_n'(z)\, e^{in\Omega t} \quad , \tag{B.31}$$

and

$$v(x,z,t) = \sum_{n=1}^{\infty} \left(f/q_n\right) U_n(x,z)\, e^{in\Omega t} \quad , \tag{B.32}$$

where

$$q_n = \sigma + in\Omega \quad . \tag{B.33}$$

Eliminating Eq. (B. 27) and using (B. 29) gives

$$W_n'' = a_n W_n - b_n A_n , \tag{B. 34}$$

and

$$A_n'' = c_n W_n + d_n A_n , \tag{B. 35}$$

where

$$a_n = \frac{q_n^2 \lambda^2}{\left(q_n^2 + f^2\right)} , \quad c_n = \frac{\Gamma}{K} ,$$

$$b_n = \frac{q_n \alpha \lambda^2}{\left(q_n^2 + f^2\right)} , \quad d_n = \frac{in\Omega}{K} . \tag{B. 36}$$

If we now let

$$A_n(z) = \hat{A}_n e^{\alpha_n z} , \tag{B. 37a}$$

and

$$W_n(z) = \hat{W}_n e^{\alpha_n z} , \tag{B. 37b}$$

where $\hat{A}_n$ and $\hat{W}_n$ are the values at the surface, and substitute Eq. (B. 37) into Eqs. (B. 34) and (B. 35), we derive the following matrix equation which must be satisfied in order to obtain a nontrivial solution:

$$\begin{pmatrix} b_n & \mu_n - a_n \\ \mu_n - d_n & -c_n \end{pmatrix} \begin{pmatrix} \hat{A}_n \\ \hat{W}_n \end{pmatrix} = 0 , \tag{B. 38}$$

where

$$\mu_n = \alpha_n^2 . \tag{B. 39}$$

The only nontrivial solutions to Eq. (B. 38) are those for which the determinant vanishes. This gives the two "allowed" values for μ_n:

$$\mu_{n1} = \frac{\left(a_n + d_n\right) + \left[\left(a_n + d_n\right)^2 - 4\left(a_n d_n + c_n b_n\right)\right]^{1/2}}{2} = E_{n1}\, e^{i\gamma_{n1}} ,$$

$$\mu_{n2} = \frac{\left(a_n + d_n\right) - \left[\left(a_n + d_n\right)^2 - 4\left(a_n d_n + c_n b_n\right)\right]^{1/2}}{2} = E_{n2}\, e^{i\gamma_{n2}} . \qquad \text{(B. 40)}$$

The roots of the dispersion relationship correspond to four values of α_n which are given by

$$\alpha_n = \pm E_{n1}^{1/2}\, e^{\left(i\gamma_{n1}/2\right)} = \pm U_{n1}\left[\cos\left(\eta_{n1}\right) + i \sin\left(\eta_{n1}\right)\right] , \qquad \text{(B. 41a)}$$

and

$$\alpha_n = \pm E_{n2}^{1/2}\, e^{\left(i\gamma_{n2}/2\right)} = \pm U_{n2}\left[\cos\left(\eta_{n2}\right) + i \sin\left(\eta_{n2}\right)\right] , \qquad \text{(B. 41b)}$$

where

$$\eta_{n1} = \gamma_{n1}/2, \quad \eta_{n2} = \gamma_{n2}/2, \quad U_{n1} = E_{n1}^{1/2} , \quad U_{n2} = E_{n2}^{1/2} . \qquad \text{(B. 41c)}$$

Of the four possible roots for α_n only two are acceptable — one from Eq. (B. 41a) and one from Eq. (B. 41b). The criterion for selecting the roots is that the real parts of α_n must be negative so as to insure exponential damping of the sea breeze. We define the two roots for α_n by the equations

$$\alpha_{n1} = \epsilon_{n1}\mu_{n1}^{1/2} = \epsilon_{n1} U_{n1}\left[\cos\left(\eta_{n1}\right) + i \sin\left(\eta_{n1}\right)\right] = k_{n1} + i\ell_{n1} , \qquad \text{(B. 42a)}$$

and

$$a_{n2} = \epsilon_{n2}\mu_{n2}^{1/2} = \epsilon_{n2} U_{n2}\left[\cos\left(\eta_{n1}\right) + i \sin\left(\eta_{n1}\right)\right] = k_{n2} + i\ell_{n2} , \qquad \text{(B. 42b)}$$

where

$$\begin{aligned}
\epsilon_{n1} &= +1 \text{ if } \cos\left(\eta_{n1}\right) < 0 , \\
\epsilon_{n1} &= -1 \text{ if } \cos\left(\eta_{n1}\right) > 0 , \\
\epsilon_{n2} &= +1 \text{ if } \cos\left(\eta_{n2}\right) < 0 , \\
\epsilon_{n2} &= -1 \text{ if } \cos\left(\eta_{n2}\right) > 0 .
\end{aligned} \qquad \text{(B. 43)}$$

In terms of the α_n the solution to the problem is given by

$$\theta = \sin \lambda x \sum_{n=1}^{\infty} \left(\hat{A}_{n1} e^{\alpha_{n1} z} + \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t} , \qquad \text{(B. 44)}$$

$$w = \sin x \sum_{n=1}^{\infty} \left(r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t} , \qquad \text{(B. 45)}$$

$$u = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} \left(\alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t} , \qquad \text{(B. 46)}$$

$$v = \lambda^{-1} \cos \lambda z \sum_{n=1}^{\infty} g_n \left(\alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t} . \qquad \text{(B. 47)}$$

where

$$g_n = -\left(f/q_n\right) \quad , \tag{B.48}$$

and

$$r_{n1} = -b_n/\left(\mu_{n1} - a_n\right), \; r_{n2} = -b_n/\left(\mu_{n2} - a_n\right) \quad . \tag{B.49}$$

Up to this point the analysis has been carried out in complex arithmetic. The actual physical meteorological quantities are obtained by first determining $\hat{A}_{n1}$ and $\hat{A}_{n2}$ from the boundary conditions, and then taking the real parts of Eqs. (B.44) - (B.47).

Boundary Conditions

In the theory of the sea breeze it is assumed that the shape of the temperature differential at the surface is given by

$$\theta(x, z=0, t) = \sin \lambda x \; T(t) \quad ,$$

where T(t) is a function of time. A positive value of T(t) corresponds to the surface temperature profile shown in Figure B.1, in which the land temperature

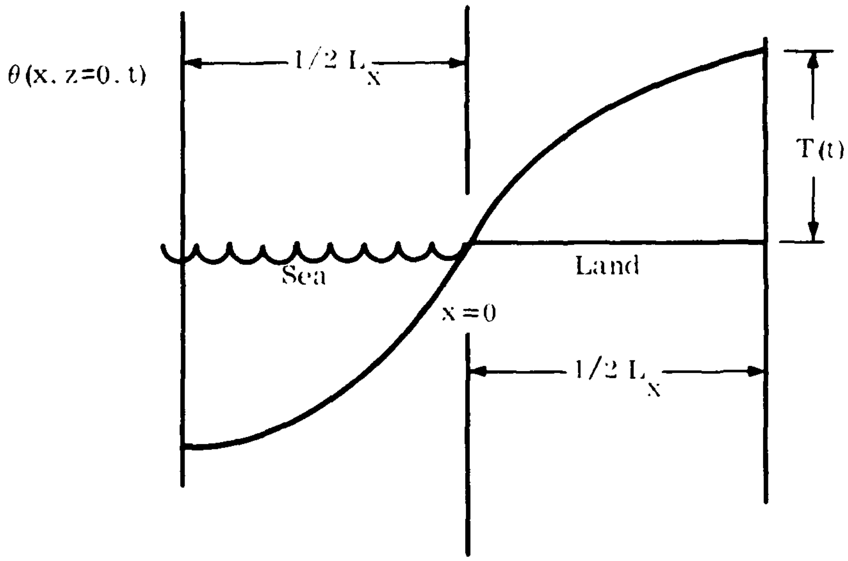


Figure B.1. Surface Temperature Variation is higher than the temperature over the water. T(t) is expressible as a Fourier series in multiples of the sidereal day frequency, Ω,

$$T(t) = \sum_{n=1}^{\infty} T_n e^{in\Omega t} \quad , \tag{B.50}$$

where

$$T_n = \Omega(2\pi)^{-1} \int_0^{(2\pi/\Omega)} T(t)\, e^{-in\Omega t} dt = \Omega(2\pi)^{-1} \left[\int_0^{2\pi/\Omega} T(t) \cos(n\Omega t)\, dt + i \int_0^{2\pi/\Omega} T(t) \sin(n\Omega t)\, dt \right] \tag{B.51}$$

$$T_n = T_n^* e^{i\tau_n} \quad ;$$

T_n^* is the magnitude of T_n and τ_n is its phase as computed from Eq. (B. 51). On the other hand, from Eq. (B. 44) we must have

$$T_n = \hat{A}_{n1} + \hat{A}_{n2} \quad . \tag{B.52}$$

The additional equation which is necessary to determine $\hat{A}_{n1}$ and $\hat{A}_{n2}$ is determined from the requirement that w = 0 at z = 0 for each vibrational mode. Thus,

$$r_{n1} \hat{A}_{n1} + r_{n2} \hat{A}_{n2} = 0 \quad . \tag{B.53}$$

Solving for $\hat{A}_{n1}$ and $\hat{A}_{n2}$ from Eqs. (B. 52) and (B. 53), inserting the results into Eqs. (B. 44) - (B. 47), and then taking the real parts of the latter equations will give us the expressions for the physical meteorological quantities. First, however, it is convenient to define the following quantities in polar form.

$$\hat{A}_{n1} = -\left[r_{n2} / (r_{n1} - r_{n2}) \right] T_n = S_{n1} e^{is_{n1}} T_n = S_{n1} T_n^* e^{i(s_{n1} + \tau_n)} \quad ;$$

$$\hat{A}_{n2} = \left[r_{n1} / (r_{n1} - r_{n2}) \right] T_n = S_{n2} e^{is_{n2}} T_n = S_{n2} T_n^* e^{i(s_{n2} + \tau_n)} \quad ;$$

$$r_{n1} = M_{n1} e^{im_{n1}} \quad ;$$

$$r_{n2} = M_{n2} e^{im_{n2}} \quad ;$$

$$g_n = G_n e^{i\nu_n} \quad ;$$

where

$$G_n = - f/\left(\sigma^2 + (n\Omega)^2\right)^{1/2} \quad ,$$

and

$$\nu_n = - \tan^{-1} (n\Omega/\sigma) \quad .$$

We then have

$$\theta = \sum_{n=1}^{\infty} \theta_n \quad , \tag{B. 54}$$

$$w = \sum_{n=1}^{\infty} w_n \quad , \tag{B. 55}$$

$$u = \sum_{n=1}^{\infty} u_n \quad , \tag{B. 56}$$

$$v = \sum_{n=1}^{\infty} v_n \quad , \tag{B. 57}$$

where

$$\theta_n = \sin \lambda x\, T_n^* \Big[S_{n1}\, e^{k_{n1} z} \cos\left(n\Omega t + \ell_{n1} z + s_{n1} + \tau_n\right)$$

$$+ S_{n2}\, e^{k_{n2} z} \cos\left(n\Omega t + \ell_{m2} z + s_{n2} + \tau_n\right)\Big] \quad , \qquad \text{(B. 58)}$$

$$w_n = \sin \lambda x\, T_n^* \Big[S_{n1} M_{n1}\, e^{k_{n1} z} \cos\left(n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \tau_n\right)$$

$$+ S_{n2} M_{n2}\, e^{k_{n2} z} \cos\left(n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \tau_n\right)\Big], \qquad \text{(B. 59)}$$

$$u_n = \lambda^{-1} \cos \lambda x\, T_n^* \Big[\epsilon_{n1} S_{n1} M_{n1} U_{n1}\, e^{k_{n1} z} \cos\left(n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \eta_{n1} + \tau_n\right)$$

$$+ \epsilon_{n2} S_{n2} M_{n2} U_{n2}\, e^{k_{n2} z} \cos\left(n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \eta_{n2} + \tau_n\right)\Big] \; , \qquad \text{(B. 60)}$$

$$v_n = \lambda^{-1} \cos \lambda x\, T_n^* G_n \Big[\epsilon_{n1} S_{n1} M_{n1} U_{n1}\, e^{k_{n1} z} \cos\left(n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \eta_{n1} + \tau_n + \nu_n\right)$$

$$+ \epsilon_{n2} S_{n2} M_{n2} U_{n2}\, e^{k_{n2} z} \cos\left(n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \eta_{n2} + \tau_n + \nu_n\right)\Big] . \qquad \text{(B. 61)}$$

In addition, the stream function $\psi = B(z, t) \cos \lambda x$ is given by

$$\psi = - \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} T_n^* \Big[S_{n1} M_{n1}\, e^{k_{n1} z} \cos\left(n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \tau_n\right)$$

$$+ S_{n2} M_{n2}\, e^{k_{n2} z} \cos\left(n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \tau_n\right)\Big] \; . \qquad \text{(B. 62)}$$

References

B. 1 H. Jeffreys, "On the Dynamics of Wind," Quart. J. Roy. Meteorol. Soc. 48, 29 (1922).

B. 2 F. H. Schmidt, "An Elementary Theory of the Land and Sea-Breeze Circulation," J. Meteorol. 4, 9 (1947).

B. 3 B. Haurwitz, "Comments on the Sea-Breeze Circulation," J. Meteorol. 4, 1 (1947).

B. 4 F. Defant, "Local Winds," in Compendium of Meteorology (Boston, Mass.: Am. Meteorol. Soc., 1951) pp. 655-671.

B. 5 F. Defant, "Theorie der Land-und Seewinde," Arch. Meteorol. Geophys. Bioklimatol. 2, 404 (1950).

B. 6 B. Haurwitz, "A Linear Sea Breeze Model," New York University, College of Engineering Research and Development, Quarterly Progress Report No. 3, Project Nr. 3-36-05-401, 1959.

B. 7 R. P. Pierce, "The Calculation of the Sea Breeze Circulation in Terms of the Differential Heating Across the Coast Line," Quart. J. Roy. Meteorol. Soc. 81, 351 (1955).

B. 8 E. L. Fisher, "A Theoretical Study of the Sea Breeze," J. Meteorol. 18, 216 (1960).

B. 9 Travelers Research Center, Inc., "The Influence of Local Winds on Fallout," Contract No. DA 36-039 AMC-03283(E), July, 1964.

B. 10 Lord Rayleigh, "On Convection Currents in a Horizontal Layer of Fluid When the Higher Temperature is on the Under Side," Phil. Mag. 32, 529 (1916).

APPENDIX C

TOPOGRAPHIC DATA INPUT PROGRAMS TOPIN AND DATERR

Introduction

The piecewise-planar topographic description system provided for use during particle transport (see p. 37, Figure 14, and p.131 ff) requires that topographic data be prepared and stored in a specific manner on magnetic tape prior to Transport Module execution. During transport, subroutines RDTOPO and HEIGHT serve to provide the transport program with the appropriate topographic data when it is needed. Two other programs TOPIN and DATERR, have been written to aid the researcher in the preparation of topographic data tapes for DELFIC. Working together, these two programs accept the user-prepared topographic description data from cards, perform many checks of data structure and consistency, and then, if the data set is adequate, prepare the input tape to be used by the Transport Module.

Description of Card Inputs

To explain the use of programs TOPIN and DATERR, we present in Table C 1 a description of the card inputs to TOPIN and DATERR. (A suggested procedure for encoding actual topographic data and descriptions of the operation of both TOPIN and DATERR along with flow charts and program listings are included in the sections that follow.)

TABLE C. 1

CARD INPUTS TO TOPIN AND DATERR

Card Number	Content	Variable Names and Format
1	Limiting coordinates of the area to be covered by this topographic data tape: lower X, upper X, lower Y, upper Y (m)	TXLL, TXLU, TYLL, TYLU (4E13. 6)
2	Topography identification card	(TOPID(J)J=1, 12) (12A6)
3	Control integer to indicate which data checking program is to be used. 0 indicates DATERR, other values are unassigned	ISUBR (I2)
4	Print control integer. 0 causes all inputs to be printed. 1 suppresses printing.	IPRNT (I2)
5	Grid interval and limiting coordinates for the first block of topo data (m)	GRINT, BXLL, BXLU, BYLL, BYLU (5E13. 6)
6	Number of grid squares in the X direction and in the Y direction, respectively, in the regular data array S(I, J)	II, JJ (2I12)
7	Regular grid data and address array of the current data block	((S(I, J), I=1, II), J=1, JJ) (5E13. 6)
8	Subsidiary data and address array of the current data block to be read five entires per card. The end of this data set is marked by a blank entry.	SUBSID(K) to SUBSID(K+ 5) (5E13. 6)
9	Same as card set 5 but for the second data block	
10	Same as card set 6 but for the second data block	
11	Same as card 7 but for the second block	
12	Same as card 8 but for the second data block	
. . .	. . .	
Last Card	blank	

A Recommended Procedure for the Encoding of Topographic Data

The piecewise-planar description system was designed to allow the user to provide when necessary a detailed topo description for DELFIC transport. In his initial planning for describing a topo surface the user must first settle upon the limiting coordinates of the area he wishes to describe. If this rectangular area is large in relation to the desired degree of detail within it, the user may wish to break the area up into a number of subareas. It is recommended that the number of subareas be kept as small as possible, preferably one, since program running time increases with the number of subarea blocks. The procedure for encoding the data of an individual block begins with the determination of the limiting coordinates of the topo subarea corresponding to the forthcoming data block. Like the complete topographic area, all subareas are rectangular with sides arranged north-south and east-west so that only four coordinates are required to define and locate the subarea. In addition a grid interval must be specified. This interval should be arranged so that the two-dimensional array S(I, J) is used extensively because the program running time is not adversely affected by having many entries in S(I, J).

Further subdivision of the grid squares represented in S(I, J) will add to program execution time and thus should be used only when necessary to achieve the desired degree of topo detail. Of course, the data set for further subdivisions of S(I, J) is restricted by the dimensioned size of array SUBSID(K).

The procedure recommended for actually encoding the topo data for arrays S(I, J) and SUBSID(K) is as follows:

1. Secure topo sheet(s) for the area to be encoded.
2. On the topo sheet(s) draw the limits of the subarea and the grid lines to subdivide the subarea. Note that in drawing these grid lines, the user should start in the south-west corner and work toward the north-east. For a prescribed grid interval the last row and column represented on the topo sheet may, and can, extend somewhat beyond the northern and eastern limits of the subarea. An automatic compensation is made by the program to adjust the area boundaries.

3 Next, the user should consider each grid square in turn to determine whether or not further subdivision is desired. Squares not to be subdivided simply have their elevation entered in the appropriately indexed elements which, incidentally, are read by the program row by row (west to east) from south to north. Whenever the user encounters a grid square that he wishes to subdivide, an address (index K) of the first of a group of four entries in the array SUBSID(K) must be entered into S(I, J) preceded by a minus sign. The array SUBSID(K) may be blocked off into sets of four before starting this encoding procedure; if these sets are filled in sequence from the top, no difficulty will arise.

It is recommended that the researcher draw subdividing lines on the topo sheet whenever a grid square is subdivided. Grid squares are always subdivided into four equal-sized squares.

It is recommended that the user proceed in a regular manner left to right within rows and bottom to top by rows until the basic two-dimensional grid has been passed over once. The sequence of blocks-of-four in SUBSID(K) will then be established as identical to the established sequence of addresses written into S(I, J).

4. Next, the user should return to the first grid square which was marked to be subdivided and assign either heights or further addresses to its four subdivisions. The sequence in which the four subdivisions are to be treated is established by convention as indicated by the following diagram:

2	3
1	4

Note that the sequence is clockwise from the south-west corner This sequence is used by subroutine HEIGHT in retrieving height data and must be observed by the user

It is recommended that the user adopt the procedure of passing across the complete map subarea at the level of first subdivisions of grid squares before further subdividing the subdivisions In this way he will be able to maintain the required sequences without conscious effort to explicitly relate entries in SUBSID(K) to particular subdivision areas on the map

Operation of Programs TOPIN and DATERR

TOPIN

After initializing itself and rewinding two tapes TOPIN begins by reading a card containing the limiting coordinates of the complete area for which the topographic heights are to be recorded. This area must always be rectangular in form with its sides aligned in east-west, north-south directions so that four coordinates suffice to define it Next, TOPIN reads an integer (ISUBR) which indicates the user's selection of a data checking program (Currently only one data checking program, DATERR, exists) Next another integer, IPRINT, is read to indicate whether or not the program should print a full copy of its results. If IPRINT is zero, results will be printed

Next, the program branches on the value of ISUBR to a data reading and checking program Currently DATERR is the only one available so that DATERR is called at this point DATERR reads and checks topographic data for one topographic data block each time it is called DATERR returns with parameter GRINT = 0.0 when it is entered after all topographic data have been processed

Upon return from DATERR or any other data reading and checking program TOPIN checks parameter GRINT for the termination condition (GRINT = 0 0) If termination is indicated, a transfer is made to statement number 11 (see the program listing) for final processing; if otherwise, parameter ITAPE is tested to see if a valid topo tape is still possible (ITAPE = 0) or if only a check of the remaining input deck can be made (ITAPE ≠ 0) If ITAPE equals zero, a block count and the arrays S(I, J) and SUBSID(K) are put temporarily onto tape ITEMPO and a return is made to statement 81 which is just before the calls to the data checking programs. If ITAPE does not equal zero, the writing out of the processed data records is skipped. Eventually the condition, GRINT = 0.0 will be encountered and processing will continue at statement 13. At 13 the parameter ITAPE is checked again and if errors have already been discovered in the data set, a comment is made to that effect and TOPIN stops. Otherwise, parameter ICHECK is set to 1 and DATERR is entered to carry out certain other tests on the data set as a whole. If errors are found, ITAPE is set positive so that when DATERR returns, a test of ITAPE can lead to either an error comment (if ITAPE $\neq$ 0) or the writing of the topography tape in final form (if ITAPE = 0) and then a final stop.

DATERR

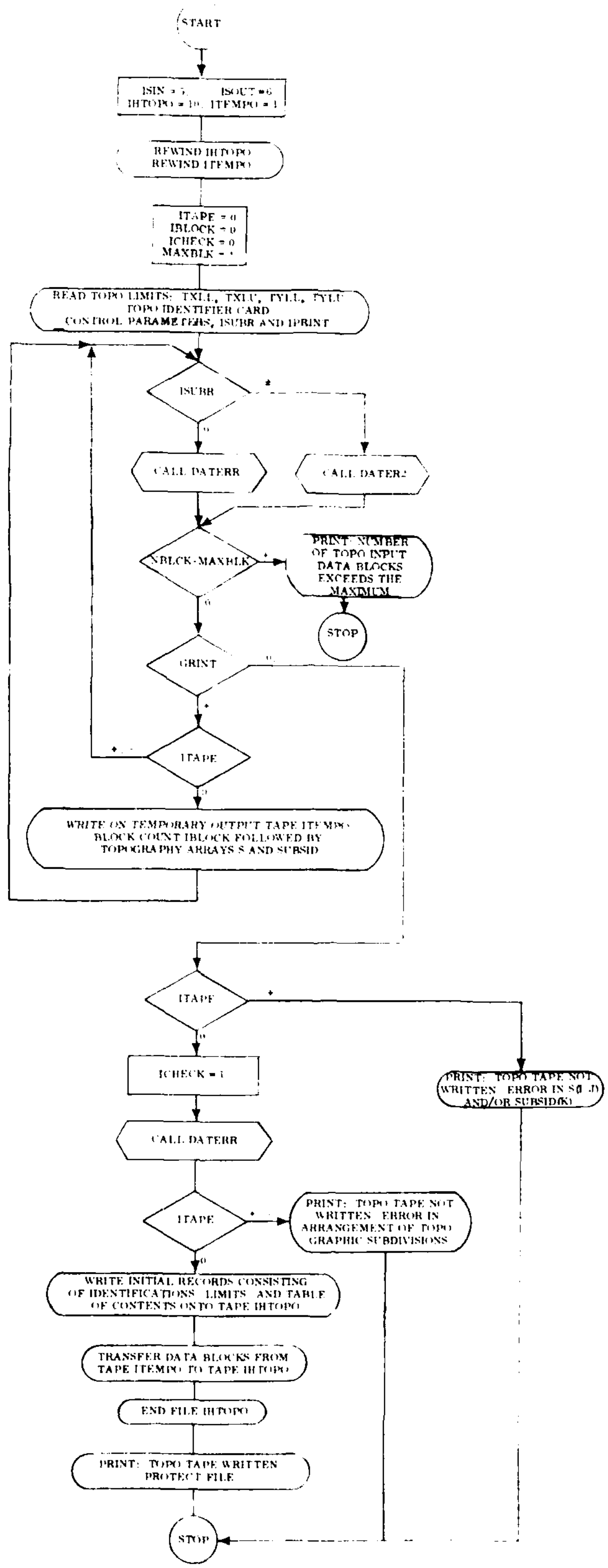
As indicated earlier this program has two different modes of operation. In the first (called when ICHECK = 0) it reads and checks a block of topographic data, and in the second (ICHECK $\neq$ 0) it performs tests on the complete topo table of contents and prepares the topo tape (IHTOPO) in its final form. The read-and-check mode begins at statement 16 by reading a card containing a grid interval and the limiting coordinates pertaining to the rectangular area that is to be documented in the current data block. A zero value of GRINT indicates that the last actual data block has already been processed and, therefore, if GRINT = 0.0, a return is made immediately; if not, the block counter IBLOCK is incremented and the data arrays S(I, J) and SUBSID(K) are read. Then, at statement 22 data checking begins. Between 22 and 40 the code ascertains that the addresses imbedded with S and SUBSID are indeed reasonable and matched by appropriate values or further addresses.

Next, after 40. the highest topo height is found and recorded in the topo table of contents along with lower coordinate limits, grid interval, and maximum array indices of the current data block.

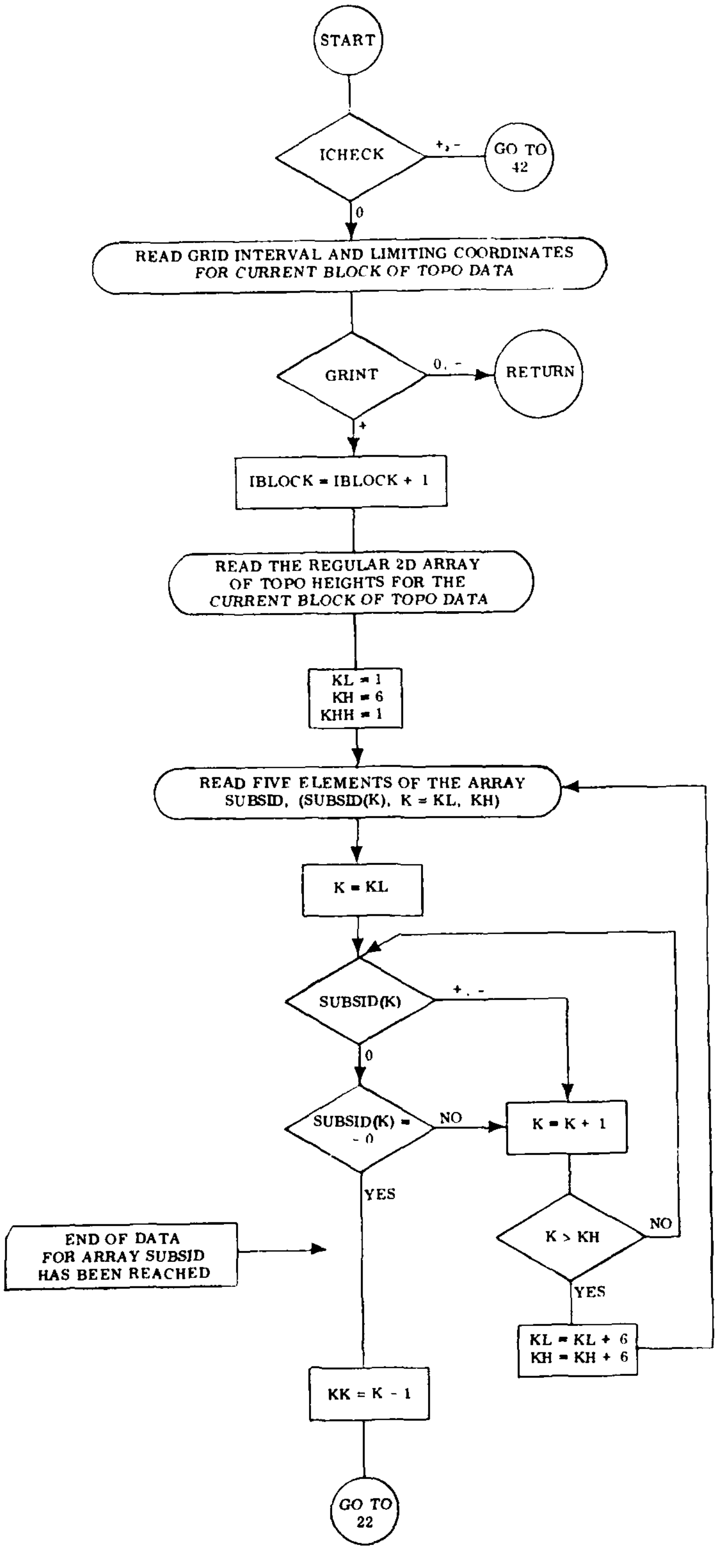
Successive tests are carried out as follows: (1) to ascertain that the number of entries in the subsidiary table is four times the total number of addresses in S(I, J) and SUBSID(K), (2) to ascertain that all height entries may be logically reached, (3) to check that the total area to be covered by the topo tape is not greater than the sum of the subareas covered by the individual data blocks, (4) to seek out any cases in which one subarea is totally included within another, and (5) to check that no gaps have been left between neighboring subareas. If any of these tests uncover an error, an explanatory comment is written and parameter ITAPE is set positive to indicate that a topo tape cannot be written in the desired final form.

Flow Charts and Program Listings

Flow charts of the main program TOPIN and subroutine DATERR are shown in FC-C.1 and FC-C.2, respectively. FORTRAN listings are included on p. 299 ff.

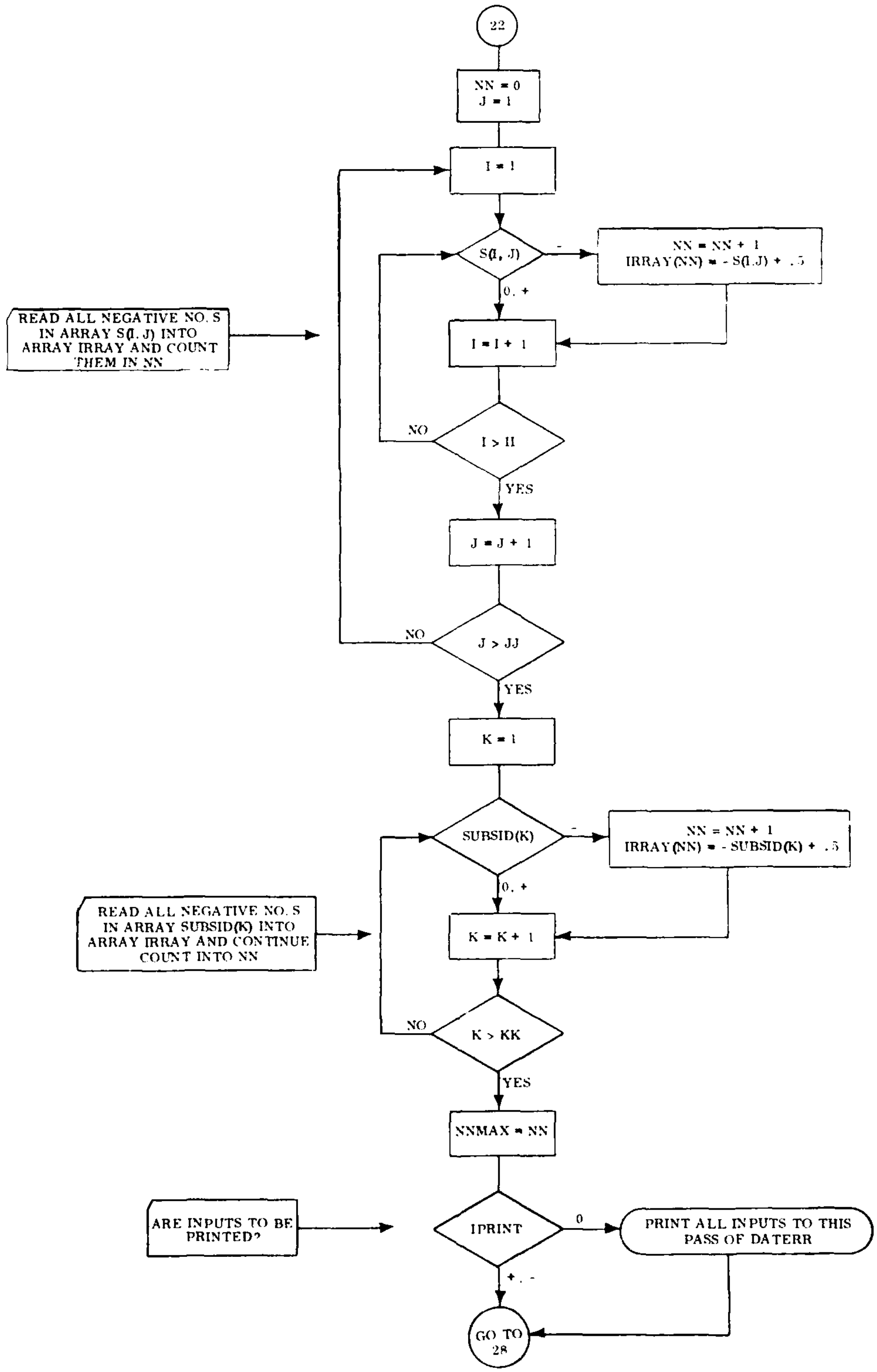


FC-C.1. Flow Chart of Main Program TOPIN



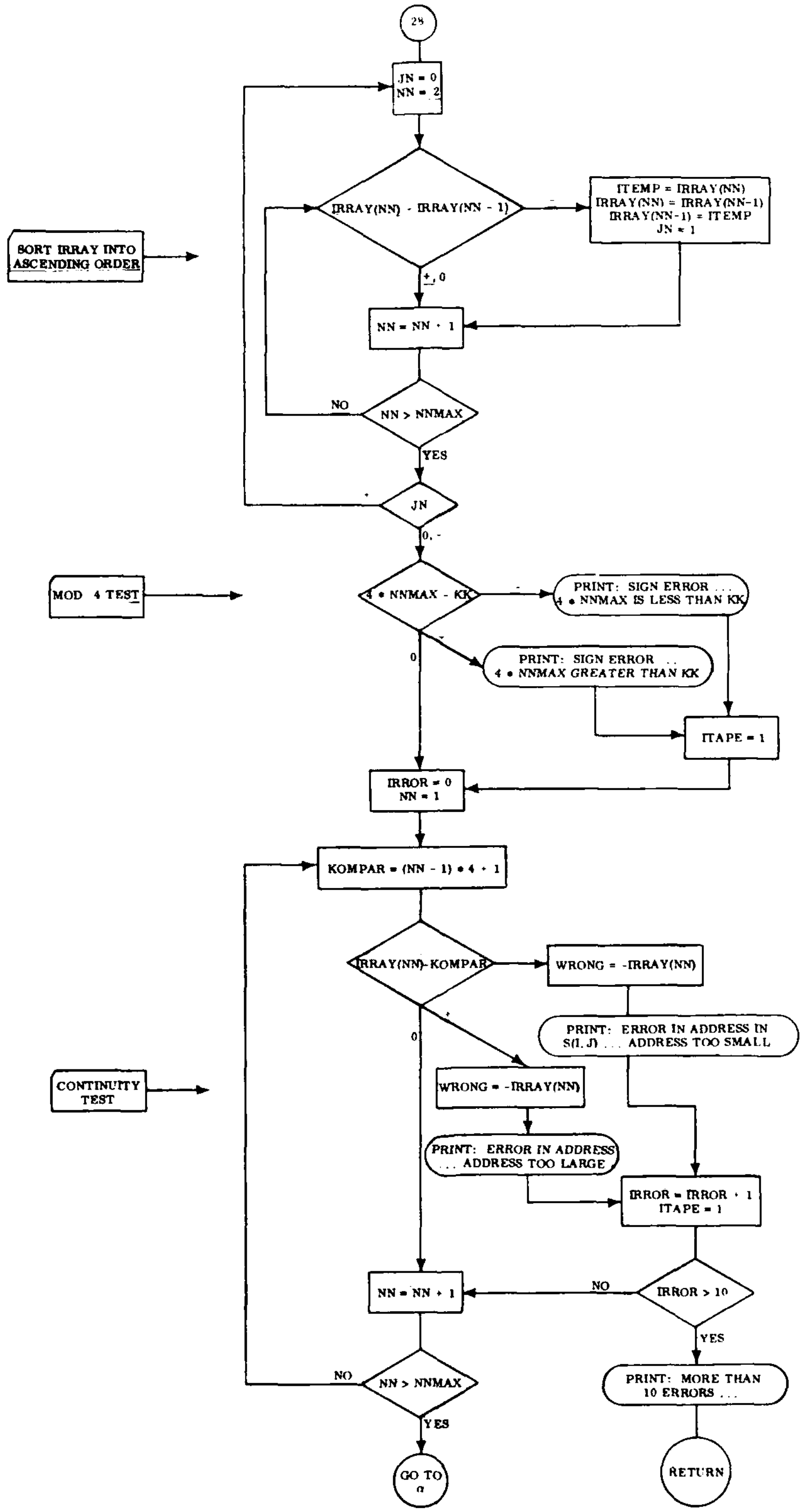
(a)

FC-C.2. Flow Charts of Subroutine DATERR



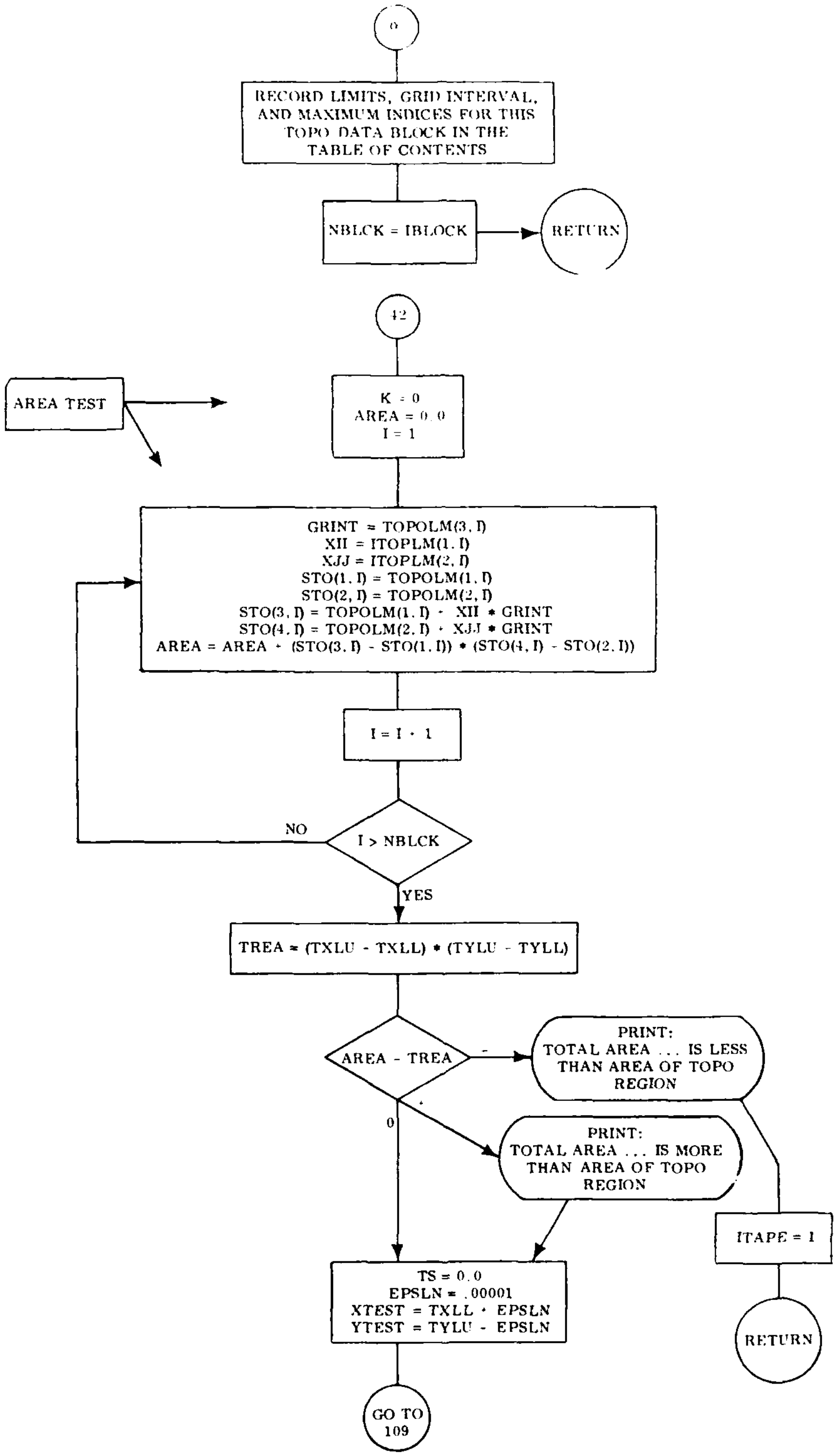
(b)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



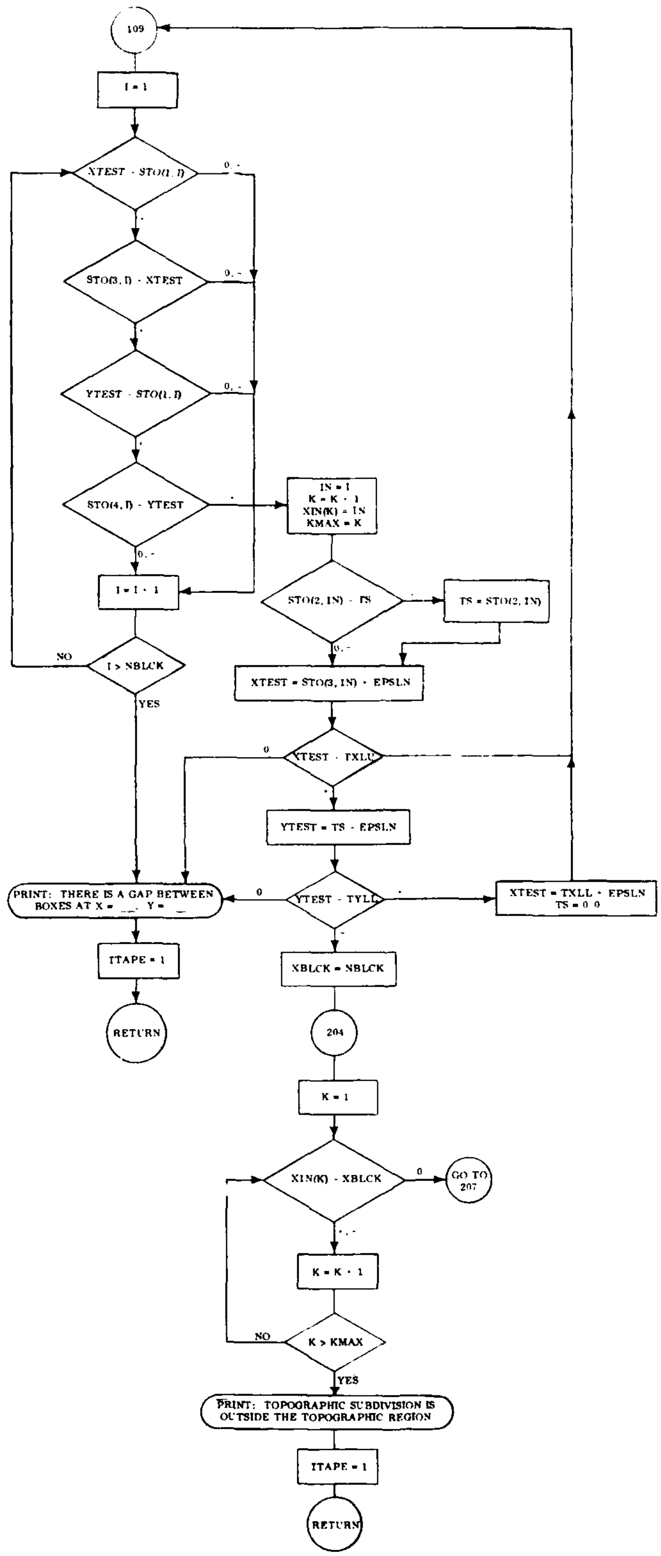
(c)

FC-C. 2. (Continued) Flow Charts of Subroutine DATERR



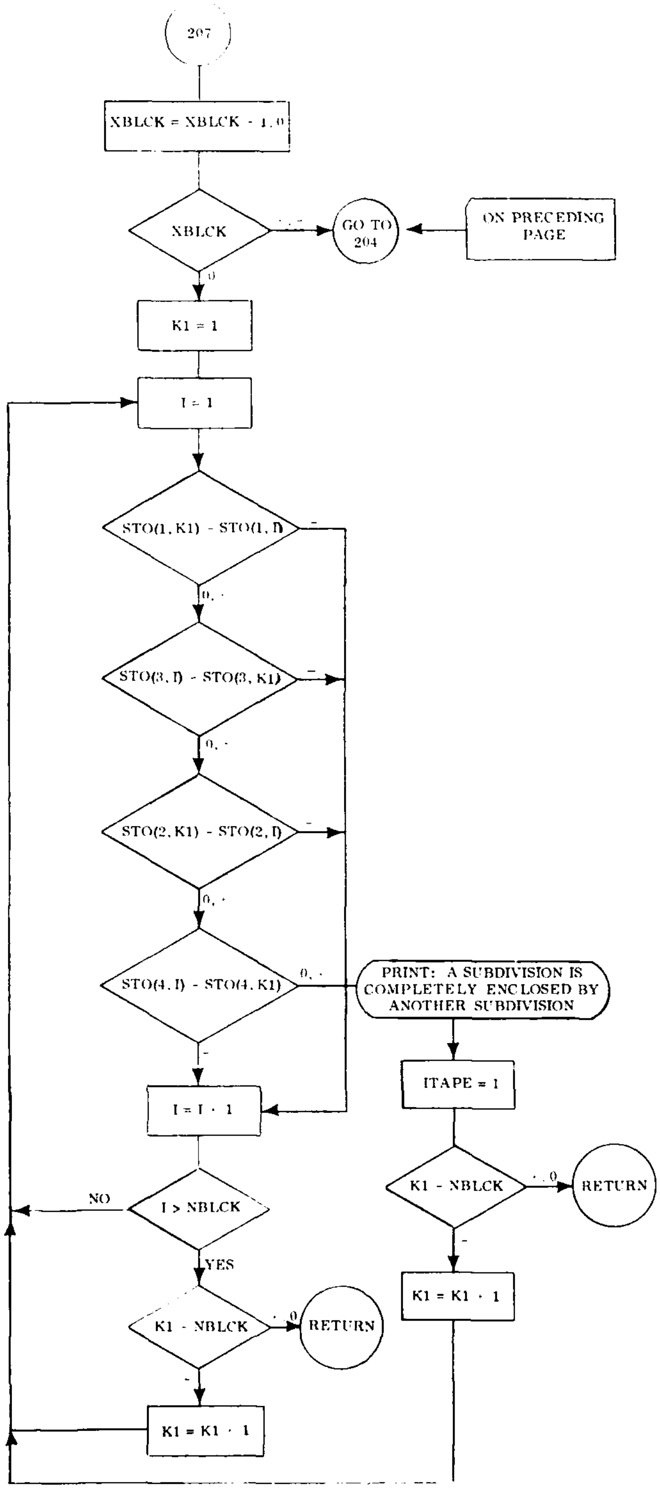
(d)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



(e)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



(f)

FC-C. 2. (Contined) Flow Charts of Subroutine DATERR

```
$IBFTC TOPINX  LIST,DECK,M94/2                                         TOPI
C     TOPIN                                                             TOPI
C     MAIN PROGRAM FOR CONSTRUCTING TOPOGRAPHY TAPE                     TOPI
C                                                                       TOPI
C ********************    GLOSSARY FOR TOPIN AND DATERR    *************TOPI
C                                                                       TOPI
C     BXLL           LOWER X BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI
C                    IN CORE                                            TOPI
C     BXLU           UPPER X BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI
C                    IN CORE                                            TOPI
C     BYLL           LOWER Y BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI
C                    IN CORE                                            TOPI
C     BYLU           UPPER Y BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI
C                    IN CORE                                            TOPI
C     GRINT          LENGTH OF THE STANDARD GRID INTERVAL  SEE S(I,J)   TOPI
C     IHTOPO         FINAL PREPARED TOPOGRAPHY TAPE NUMBER              TOPI
C     II             UPPER LIMIT OF I DIMENSION OF S(I,J) ARRAY         TOPI
C     IPRINT         INDICATES WHETHER INPUT (TOPO ARRAYS, BOUNDARIES,  TOPI
C                    LIMITS ETC. FOR EACH BLOCK) IS TO BE PRINTED.      TOPI
C                    0 CAUSES PRINTING , 1 CAUSES NONE.                 TOPI
C     ISIN.          SYSTEM INPUT TAPE NUMBER                           TOPI
C     ISOUT          SYSTEM OUTPUT TAPE NUMBER                          TOPI
C     ISUBR          INDICATES IF = 0 THAT SUBROUTINE DATERR IS TO BE USEDTOPI
C     ITAPE          NOT ZERO INDICATES THAT ERRORS HAVE BEEN FOUND IN  TOPI
C                    THE DATA SET AND THAT A VALID TOPO TAPE CANNOT BE  TOPI
C                    WRITTEN.    A ZERO INDICATES NO ERRORS ARE APPARENT TOPI
C     ITEMPO         TEMPORARY TOPO TAPE NUMBER                         TOPI
C     JJ             UPPER LIMIT OF J DIMENSION OF S(I,J) ARRAY         TOPI
C     S(I,J)         TWO DIMENSIONAL ARRAY CONTAINING EITHER TOPOGRAPHIC TOPI
C                    HEIGHT OF GRID SQUARE I,J ON THE MAP OR A NEGATIVE TOPI
C                    NUMBER WHICH, WHEN CONVERTED TO A POSITIVE INTEGER , TOPI
C                    IS THE INDEX, K, TO A NUMBER IN THE ONE-DIMENSIONAL TOPI
C                    ARRAY, SUBSID(K).  IF A NUMBER IN S(I,J) IS NEGATIVE,TOPI
C                    IT IS CALLED THE BASE ADDRESS TO THE SQUARE OF SIDE TOPI
C                    GRINT WHOSE LOWER-LEFT CORNER IS AT LOCATION       TOPI
C                    (I-1)*GRINT , (J-1)*GRINT  ON THE MAP              TOPI
C     SUBSID(K)      ONE-DIMENSIONAL ARRAY CONTAINING EITHER A TOPOGRAPHICTOPI
C                    HEIGHT OR A NEGATIVE NUMBER WHICH, WHEN CONVERTED TO TOPI
C                    A POSITIVE INTEGER, IS THE INDEX, K, TO ANOTHER    TOPI
C                    NUMBER IN SUBSID(K), WHICH MAY IN TURN BE EITHER   TOPI
C                    A TOPOGRAPHIC HEIGHT OR ANOTHER BASE ADDRESS       TOPI
C     TXLL           LOWER X BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA   TOPI
C     TXLU           UPPER X BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA   TOPI
C     TYLL           LOWER Y BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA   TOPI
C     TYLU           UPPER Y BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA   TOPI
C                                                                       TOPI
C **********************************************************************TOPI
C                                                                       TOPI
      DIMENSION S(3 ,30),SUBSID(10000),TOPOLM(4,100),ITOPLM(3,100)      TOPI
     1 ,TOPID(12)                                                       TOPI
C                                                                       TOPI
C **********************************************************************TOPI
C                                                                       TOPI
      COMMON  GRINT  , BXLL   , BXLU   , BYLL   , BYLU   , II           TOPI
      COMMON  JJ     , KK     , S      , SUBSID , ITAPE  , ISIN         TOPI
      COMMON  ISOUT  , IBLOCK , IPRINT , TOPOLM , ITOPLM , TXLL         TOPI
      COMMON  TXLU   , TYLL   , TYLU   , NBLCK  , ICHECK                TOPI
C                                                                       TOPI
C **********************************************************************TOPI
C                                                                       TOPI
    1 FORMAT (15H1IHTOPO   TXLL=,F6.1,8H   TXLU=,F6.1,8H   TYLL=,F6.1,  TOPI
```

```
     18H    TYLO=,F6.1)
    2 FORMAT (12A6)
    3 FORMAT (I2)
    4 FORMAT (7H ISUBR=,I2)
    5 FORMAT (8H IPRINT=,I2)
    6 FORMAT (6HIHTOPO,4F6.1)
    7 FORMAT (68H TOPO TAPE NOT WRITTEN. ERROR IN S(I,J) AND/OR
     1SUBSID(K).       )
    8 FORMAT (33H TOPO TAPE WRITTEN. PROTECT FILE.)
    9 FORMAT (4E13.6)
   22 FORMAT(120H TOPO TAPE NOT WRITTEN.  ERROR IN ARRANGEMENT OF TOPOGR
     1APHIC SUBDIVISIONS.                                            )
   33 FORMAT(120H THE NUMBER OF INPUT TOPO DATA BLOCKS EXCEEDS THE MAXI
     1UM NUMBER PERMITTED.                                           )
C
C ************************************************************************
C
      DATA DENTI/6HIHTOPO/
C
C ************************************************************************
C ************************************************************************
C
      ISOUT=6
      ISIN=5
      IHTOPO=4
      ITEMPO=1
      REWIND IHTOPO
      REWIND ITEMPO
      ITAPE=0
      IBLOCK=0
      ICHECK=0
      MAXBLK=100
      READ (ISIN,9)TALL,TALO,TILL,TILO
      READ (ISIN,2) (TOPID(J),J=1,12)
      READ (ISIN,3)ISUBR
      READ (ISIN,3)IPRINT
   81 IF(ISUBR)91,10,91
   91 WRITE (ISOUT,6)
C
C     INSERTION POINT FOR ALTERNATE DATA CHECKING PROGRAM  DATERZ
      GO TO 16
   10 CALL DATERR
      IF(NBLCK-MAXBLK)30,30,31
   30 IF(GRINT)13,15,11
   31 WRITE (ISOUT,33)
      GO TO 16
   11 IF(ITAPE)10,12,10
   12 WRITE (ITEMPO)IBLOCK
      WRITE (ITEMPO)((S(I,J),I=1,II),J=1,JJ)
      WRITE (ITEMPO)(SUBSID(K),K=1,KK)
      GO TO 81
   13 IF(ITAPE)14,15,14
   14 WRITE (ISOUT,7)
      GO TO 16
   15 ICHECK=1
      CALL DATERR
      IF(ITAPE)21,20,21
   21 WRITE (ISOUT,22)
```

```
      GO TO 16
   20 WRITE (IHTOPO) DENTI
      WRITE (IHTOPO)IXLL,IXLU,IYLL,IYLU,NBLCK
      WRITE (IHTOPO,2)
      WRITE (IHTOPO)TOPOLM
      WRITE (IHTOPO)ITOPLM
      REWIND ITEMPO
   18 READ (ITEMPO)IBLOCK
      IM=ITOPLM(1,IBLOCK)
      JM=ITOPLM(2,IBLOCK)
      KM=ITOPLM(3,IBLOCK)
      READ (ITEMPO)((S(I,J),I=1,IM),J=1,JM)
      READ (ITEMPO)(SUBSID(K),K=1,KM)
      WRITE (IHTOPO)((S(I,J),I=1,IM),J=1,JM)
      WRITE (IHTOPO)(SUBSID(K),K=1,KM)
      IF(IBLOCK-NBLCK)18,19,19
   19 END FILE IHTOPO
      WRITE (ISOUT,8)
   16 STOP
      END
```

```
$IBFTC DATER    LIST,DECK,M94/2                                          DATE
C                                                                        DATE
      SUBROUTINE DATERR                                                  DATE
C                                                                        DATE
C     SUBROUTINE TESTING FOR LOOPS AND INCORRECT ADDRESSES IN S(I,J)     DATE
C     AND SUBSID(K)                                                      DATE
C                                                                        DATE
C *********************************************************************DATE
C                                                                        DATE
C     SEE PROGRAM TOPIN FOR A GLOSSARY                                   DATE
C                                                                        DATE
C *********************************************************************DATE
C                                                                        DATE
      DIMENSION S(3 ,30),SUBSID(10000),IRRAY(1000),ARRAY(1000),AIN(100),DATE
     1TOPOLM(4,100),ITOPLM(5,100),STO(4,100)                             DATE
C                                                                        DATE
C *********************************************************************DATE
C                                                                        DATE
      COMMON  GRINT  , BXLL   , BXLU   , BYLL   , BYLU   , II            DATE
      COMMON  JJ     , KK     , S      , SUBSID , ITAPE  , ISIN          DATE
      COMMON  ISOUT  , IBLOCK , IPRINT , TOPOLM , ITOPLM , IXLL          DATE
      COMMON  IXLU   , IYLL   , IYLU   , NBLCK  , ICHECK                 DATE
C                                                                        DATE
C *********************************************************************DATE
C                                                                        DATE
    1 FORMAT (5E15.6)                                                    DATE
    2 FORMAT (2I12)                                                      DATE
    3 FORMAT (6F11.1)                                                    DATE
    4 FORMAT (10H0BLOCK NO.,I7)                                          DATE
    5 FORMAT (9H   GRINT=,F8.1,9H    BXLL=,F8.1,9H    BXLU=,F8.1,        DATE
     19H   BYLL=,F8.1,9H    BYLU=,F8.1)                                  DATE
    6 FORMAT (6H   II=,I12,6H   JJ=,I12)                                 DATE
    7 FORMAT (25H   (S(I,J),I=1,II),J=1,JJ)                              DATE
    8 FORMAT (6H   KK=,I12)                                              DATE
    9 FORMAT (19H   SUBSID(K),K=1,KK)                                    DATE
   10 FORMAT (9H   NNMAX=,I12)                                           DATE
   11 FORMAT (82H SIGN ERROR IN ADDRESS OR TOPOGRAPHIC HEIGHT IN S(I,J) DATE
     1AND/OR SUBSID(K), BLOCK NO.,I7,28H .(4*NNMAX IS LESS THAN KK.))    DATE
   12 FORMAT (82H SIGN ERROR IN ADDRESS OR TOPOGRAPHIC HEIGHT IN S(I,J) DATE
     1AND/OR SUBSID(K), BLOCK NO.,I7,31H .(4*NNMAX IS GREATER THAN KK.))DATE
   13 FORMAT (51H ERROR IN ADDRESS IN S(I,J) OR SUBSID(K), BLOCK NO.,   DATE
     1I7,19H . WRONG ADDRESS IS,F11.1,24H . ADDRESS IS TOO SMALL.)       DATE
   14 FORMAT (51H ERROR IN ADDRESS IN S(I,J) OR SUBSID(K), BLOCK NO.,   DATE
     1I7,19H . WRONG ADDRESS IS,F11.1,24H . ADDRESS IS TOO LARGE.)       DATE
   15 FORMAT (81H ADDRESSES NOT IN INCREMENTS OF 4. MORE THAN 10 ERRORS.DATE
     1 ERROR SEARCH IN BLOCK NO.,I7,14H DISCONTINUED.)                   DATE
  220 FORMAT(120H TOTAL AREA OF TOPOGRAPHIC SUBDIVISIONS IS LESS THAN ARDATE
     1EA OF TOPOGRAPHIC REGION.                                         )DATE
  221 FORMAT(120H TOTAL AREA OF TOPOGRAPHIC SUBDIVISIONS IS GREATER THANDATE
     1 AREA OF TOPOGRAPHIC REGION. CHECK FOR OVERLAPS.                  )DATE
  222 FORMAT(39H THERE IS A GAP BETWEEN BOXES AT XTEST=F15.9,7H YTEST=FIDATE
     115.9)                                                              DATE
  229 FORMAT(120H A TOPOGRAPHIC SUBDIVISION IS OUTSIDE THE TOPOGRAPHIC RDATE
     1EGION.                                                            )DATE
  216 FORMAT(120H A SUBDIVISION IS COMPLETELY ENCLOSED BY ANOTHER SUBDIVDATE
     1ISION.                                                            )DATE
C                                                                        DATE
C *********************************************************************DATE
C                                                                        DATE
      DATA Q000CT/010000000000/                                          DATE
C                                                                        DATE
```

```
C ************************************************************************************
C ************************************************************************************
C
      IF(ICHECK)42,16,42
   16 READ (ISIN,1)GRINT,BXLL,BXLU,BYLL,BYLU
      IF (GRINT) 41,41,17
   17 IBLOCK=IBLOCK+1
      READ (ISIN,2)II,JJ
      READ (ISIN,3)((S(I,J),I=1,II),J=1,JJ)
      KL=1
      KH=5
      DO 21 KHH=1,11000
      READ (ISIN,3)(SUBSID(K),K=KL,KH)
      DO 20 K=KL,KH
      IF(SUBSID(K))23,18,20
   18 A=OR(SUBSID(K),000007)
      IF(A)19,20,20
   19 KK=K-1
      GO TO 22
   20 CONTINUE
      KL=KL+5
      KH=KH+5
   21 CONTINUE
C
C
C     MOD 4 CHECK
C
C
   22 NN=0
      DO 24 J=1,JJ
      DO 24 I=1,II
      IF(S(I,J))23,24,24
   23 NN=NN+1
      IRRAY(NN)=-S(I,J)+0.5
   24 CONTINUE
      DO 26 K=1,KK
      IF(SUBSID(K))25,26,26
   25 NN=NN+1
      IRRAY(NN)=-SUBSID(K)+0.5
   26 CONTINUE
      NNMAX=NN
      IF(IPRINT)28,27,28
   27 WRITE (ISOUT,4)IBLOCK
      WRITE (ISOUT,5)GRINT,BXLL,BXLU,BYLL,BYLU
      WRITE (ISOUT,6)II,JJ
      WRITE (ISOUT,7)
      WRITE (ISOUT,3)((S(I,J),I=1,II),J=1,JJ)
      WRITE (ISOUT,8)KK
      WRITE (ISOUT,9)
      WRITE (ISOUT,3)(SUBSID(K),K=1,KK)
      WRITE (ISOUT,10)NNMAX
   28 JN=0
      DO 30 NN=2,NNMAX
      IF(IRRAY(NN)-IRRAY(NN-1))29,30,30
   29 ITEMP=IRRAY(NN)
      IRRAY(NN)=IRRAY(NN-1)
      IRRAY(NN-1)=ITEMP
      JN=1
```

```
30 CONTINUE
   IF(JN)31,31,28
31 IF(4*NNMAX-KK)32,35,33
32 WRITE (ISOUT,11)IBLOCK
   GO TO 34
33 WRITE (ISOUT,12)IBLOCK
34 ITAPE=1
35 IRROR=0
   DO 40 NN=1,NNMAX
   KOMPAR=(NN-1)*4+1
   IF(IRRAY(NN)-KOMPAR)36,40,37
36 WRONG=-IRRAY(NN)
   WRITE (ISOUT,13)IBLOCK,WRONG
   GO TO 38
37 WRONG=-IRRAY(NN)
   WRITE (ISOUT,14)IBLOCK,WRONG
38 IRROR=IRROR+1
   ITAPE=1
   IF(IRROR-10)40,40,39
39 WRITE (ISOUT,15)IBLOCK
   GO TO 41
40 CONTINUE
C
C
C     IDENTIFICATION OF LARGEST TOPO HEIGHT IN TOPO SUBDIVISION
C
C
   MM=0
   DO 240 J=1,JJ
   DO 240 I=1,II
   MM=MM+1
   ARRAY(MM)=B(I,J)
240 CONTINUE
   DO 242 K=1,KK
   MM=MM+1
   ARRAY(MM)=SUBSIB(K)
242 CONTINUE
   MMAX=MM
248 JM=0
   DO 245 MM=2,MMAX
   IF(ARRAY(MM)-ARRAY(MM-1))245,245,246
246 ATEMP=ARRAY(MM)
   ARRAY(MM)=ARRAY(MM-1)
   ARRAY(MM-1)=ATEMP
   JM=1
245 CONTINUE
   IF(JM)247,247,248
247 TOPOLM(1,IBLOCK)=BALL
   TOPOLM(2,IBLOCK)=BTLL
   TOPOLM(3,IBLOCK)=GRINT
   TOPOLM(4,IBLOCK)=ARRAY(MMAX)
   ITOPLM(1,IBLOCK)=II
   ITOPLM(2,IBLOCK)=JJ
   ITOPLM(3,IBLOCK)=KK
   NBLCK=IBLOCK
   GO TO 41
C
C
```

```
C     COMPARISON OF TOTAL AREA OF TOPO REGION WITH THE SUM OF THE AREAS
C     OF THE TOPO SUBDIVISIONS
C
C
   42 K=0
      AREA=0.0
      DO 100 I=1,NBLCK
      GRINT=TOPOLM(3,I)
      XII=ITOPLM(1,I)
      XJJ=ITOPLM(2,I)
      STO(1,I)=TOPOLM(1,I)
      STO(2,I)=TOPOLM(2,I)
      STO(3,I)=TOPOLM(1,I)+XII*GRINT
      STO(4,I)=TOPOLM(2,I)+XJJ*GRINT
      AREA=AREA+(STO(3,I)-STO(1,I))*(STO(4,I)-STO(2,I))
  100 CONTINUE
      TREA=(TXLU-TALL)*(TYLU-TYLL)
      IF(AREA-TREA)200,208,201
  200 WRITE (ISOUT,220)
      ITAPE=1
      GO TO 41
  201 WRITE (ISOUT,221)
C
C
C     CHECK FOR THE ERRONEOUS ENCLOSURE OF ONE TOPO SUBDIVISION BY
C     ANOTHER
C
C
  208 K1=1
  217 DO 210 I=1,NBLCK
      IF(I-K1)219,210,219
  219 IF(STO(1,K1)-STO(1,I))210,211,211
  211 IF(STO(3,I)-STO(3,K1))210,212,212
  212 IF(STO(2,K1)-STO(2,I))210,213,213
  213 IF(STO(4,I)-STO(4,K1))210,214,214
  210 CONTINUE
      IF(K1-NBLCK)215,101,101
  215 K1=K1+1
      GO TO 217
  214 WRITE (ISOUT,216)
      ITAPE=1
      GO TO 41
C
C
C     CHECK FOR GAPS BETWEEN TOPO SUBDIVISIONS
C
C
  101 TS=0.0
      EPSLN=1.0E-5
      XTEST=TALL+EPSLN
      YTEST=TYLU-EPSLN
  109 DO 105 I=1,NBLCK
      IF(XTEST-STO(1,I))105,105,102
  102 IF(STO(3,I)-XTEST)105,105,103
  103 IF(YTEST-STO(2,I))105,105,104
  104 IF(STO(4,I)-YTEST)105,105,106
  105 CONTINUE
      GO TO 202
```

```
  106 IN=I
      K=K+1
      XIN(K)=IN
      KMAX=K
      IF(STO(2,IN)-TS)108,108,107
  107 TS=STO(2,IN)
  108 XTEST=STO(3,IN)+EPSLN
      IF(XTEST-TXLL)109,202,110
  110 YTEST=TS-EPSLN
      IF(YTEST-TYLL)203,202,111
  202 WRITE (ISOUT,222)XTEST,YTEST
      ITAPE=1
      GO TO 41
  111 XTEST=TXLL+EPSLN
      TS=0.0
      GO TO 109
C
C
C     CHECK TO INSURE THAT ALL TOPO SUBDIVISIONS HAVE BEEN EXAMINED
C
C
  203 XBLCK=NBLCK
  204 DO 205 K=1,KMAX
      IF(XIN(K)-XBLCK)205,207,205
  205 CONTINUE
      WRITE (ISOUT,229)
      ITAPE=1
      GO TO 41
  207 XBLCK=XBLCK-1.0
      IF(XBLCK)204,41,204
   41 RETURN
      END
```