

AD815263

DASA-1800-IV



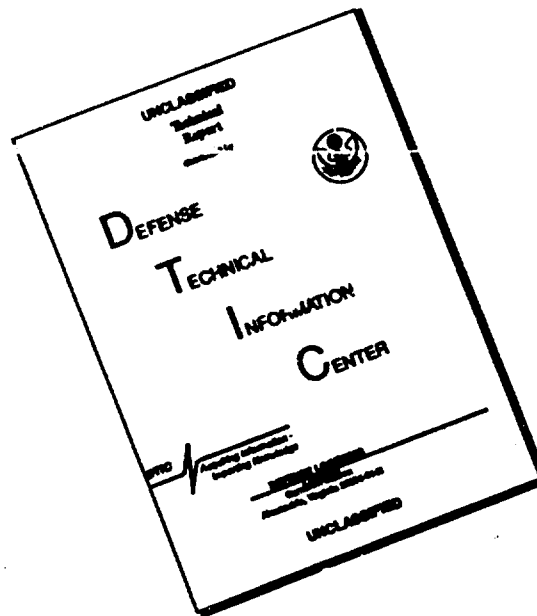
**DEPARTMENT OF DEFENSE
LAND FALLOUT
PREDICTION SYSTEM**

**Volume IV
ATMOSPHERIC TRANSPORT**

Reproduced by
**NATIONAL TECHNICAL
INFORMATION SERVICE**
U S Department of Commerce
Springfield VA 22151

DASA • NDL • NRDL • ECOM • TECH OPS

DISCLAIMER NOTICE



THIS DOCUMENT IS BEST QUALITY AVAILABLE. THE COPY FURNISHED TO DTIC CONTAINED A SIGNIFICANT NUMBER OF PAGES WHICH DO NOT REPRODUCE LEGIBLY.

N O T I C E

THIS DOCUMENT HAS BEEN REPRODUCED FROM THE BEST COPY FURNISHED US BY THE SPONSORING AGENCY. ALTHOUGH IT IS RECOGNIZED THAT CERTAIN PORTIONS ARE ILLEGIBLE, IT IS BEING RELEASED IN THE INTEREST OF MAKING AVAILABLE AS MUCH INFORMATION AS POSSIBLE.

For

DASA-1800-IV

DEPARTMENT OF DEFENSE
LAND FALLOUT PREDICTION SYSTEM

Volume IV - Atmospheric Transport

(14) TC-B-66-46
2 Feb 1967

~~W. Y. G. Ing~~
T. W. Schwenke, I. Kohlberg, H. G. Norment
~~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

	<u>Page</u>
INTRODUCTION	1
PHYSICAL AND MATHEMATICAL MODELS	4
<u>Fallout Particle Kinematics</u>	4
<u>Relationship Between Wind Field and Particle Velocity</u>	4
<u>Particle Settling Rates</u>	5
<u>Effect of Atmospheric Diffusion on Particle Transport</u>	7
<u>Wind-Field Description</u>	10
<u>Macrowind Fields</u>	10
<u>The Closest Datum Method</u>	12
<u>The Preferential-Weighting Method</u>	12
<u>The Least-Squares Method</u>	13
<u>Local Circulation Systems</u>	16
<u>Orographic Effects</u>	17
<u>The Sea Breeze</u>	28
<u>Transport In a Macrowind Cell</u>	36
<u>Transport In a Local Circulation System</u>	37
<u>Temporal Variation of the Wind Field</u>	37
<u>Topography Description</u>	37
COMPUTER PROGRAM OUTLINE	39
<u>Description</u>	39
<u>Program Discussion</u>	46
<u>Subroutine FALRAT</u>	46
<u>Subroutine DUMPP</u>	48
<u>Subroutine RDTOPO</u>	57
<u>Subroutine LINK5</u>	57
<u>Subroutine LINK6</u>	65
<u>Subroutine RDCIRS</u>	65

TABLE OF CONTENTS (Cont'd.)

	<u>Page</u>
<u>Subroutine MKWIND</u>	65
<u>Subroutine LINK7</u>	78
<u>Subroutine GETWND</u>	82
<u>Subroutine LOTRAN</u>	95
<u>Subroutine MTWND1</u>	97
<u>Subroutine RGWND1</u>	100
<u>Subroutine CBREZ1</u>	105
<u>Subroutine HEIGHT</u>	115
USER INFORMATION	124
<u>Input Description</u>	124
<u>General</u>	124
<u>MKWIND Data</u>	127
<u>Local Circulation System Data</u>	130
<u>Topography Data</u>	131
<u>Output Description</u>	136
<u>Printed Output</u>	136
<u>Binary Output</u>	136
<u>Data Structures for Secondary Memory Tapes</u>	138
REFERENCES	140
FORTRAN LISTINGS	141
<u>Subroutine FALRAT</u>	143
<u>Subroutine DUMPP</u>	145
<u>Subroutine RDTOPO</u>	153
<u>Subroutine LINK5</u>	155
<u>Subroutine LINK6</u>	166
<u>Subroutine RDCIRS</u>	167
<u>Subroutine MKWIND</u>	169
<u>Subroutine LINK7</u>	178

TABLE OF CONTENTS (Cont'd.)

	<u>Page</u>
<u>Subroutine GETWND</u>	189
<u>Subroutine LOTRAN</u>	192
<u>Subroutine MTWND1</u>	197
<u>Subroutine RGWND1</u>	201
<u>Subroutine CBREZ1</u>	205
<u>Subroutine HEIGHT</u>	211
SAMPLE TEST PROBLEM AND PRINTOUT	215
<u>Transport Module</u>	217
APPENDIX A. THEORY OF OROGRAPHIC FLOW WITH APPLICATION TO TROPOSPHERIC FALLOUT	227
<u>Introduction</u>	227
<u>Geometric Considerations</u>	229
<u>Theory of Airflow</u>	229
<u>Application of the Theory to Specific Geometries</u>	245
<u>Mountain Ridge Not Perpendicular to Unperturbed Flow</u>	245
<u>A Single Mountain</u>	249
<u>Wind Streamlines and Fallout Particle Streamlines in Two Dimensions</u>	251
<u>Conditions for Neglecting the Coriolis Effect</u>	261
<u>Conclusions</u>	263
<u>References</u>	264
APPENDIX B. THE INCORPORATION OF THE SEA BREEZE IN THE CALCULATION OF FALLOUT	265
<u>Introduction</u>	265
<u>Review of the Sea-Breeze Theories</u>	265
<u>Sea-Breeze Model</u>	268
<u>Wind-Field Parameters</u>	268
<u>Boundary Conditions</u>	279
<u>References</u>	283

TABLE OF CONTENTS (Cont'd.)

	<u>Page</u>
APPENDIX C. TOPOGRAPHIC DATA INPUT PROGRAMS	
TOPIN AND DATERR	285
<u>Introduction</u>	285
<u>Description of Card Inputs</u>	285
<u>A Recommended Procedure for the Encoding of Topographic Data</u> . . .	287
<u>Operation of Programs TOPIN and DATERR</u>	289
<u>TOPIN</u>	289
<u>DATERR</u>	290
<u>Flow Charts and Program Listings</u>	291
<u>TOPIN</u>	299
<u>DATERR</u>	302

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1 A Synopsis of the Programs of the Transport Module	45
2 Particle Classification Identifiers Used by DUMPP	49
3 Input Quantities for the Sea Breeze	110
4 Interpretative Output Description	111
5 Binary Input — Tape IPARIN	125
6 Description of the Transport Control Array — IC(I), I = 1, 8 . . .	126
7 Card Inputs for Identification and Control of the Transport Module	127
8 Wind-Field Computation Methods Specified by NCODE	128
9 Summary of Card Inputs to Subroutine MK'VIND	129
10 Card Inputs to Subroutine RDCIRS	130
11 Summary of Card Inputs to Subroutine MTWND1, RGWND1, and CBREZ1	132
12 The Binary Topography Tape Data	133
13 Printed Output of the Transport Module	137
14 The Grounded Particles Tape, IPOUT	139
B.1 Typical Sea Breeze Values	273
C.1 Card Inputs to TOPIN and DATERR	286

LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Operations of the Cloud Rise-Transport Interface Module	2
2	The Elementary Cloud Subdivision and Its Characterization	3
3	Segment of Cloud Volume	7
4	Strata Within the Specified Wind Field Volume	11
5	Wind Cells — Subdivisions of a Stratum	11
6	Wind-Field Volume with Superimposed Local Circulation System Cells	17
7	Mountain Ridge Not Perpendicular to Flow	19
8	Mountain Ridge Separations	23
9	Slope Compensation	27
10	Sea-Breeze Circulation	29
11	Temperature Differential Profile	29
12	Geometric Considerations Related to the Sea Breeze	35
13	Boundary-to-Boundary Transport and A Mountain Wind Cell	36
14	Piecewise-Planar Topography Specification Below the Macrowind-Field Volume	38
15	Program Arrangement for the Transport Module	40
16	General Flow Chart of the Transport Module (a), and Transport Module Loops (b)	41
17	Transport Module Data Flow	43
A. 1	Mountain Ridge Not Perpendicular to Flow	245
A. 2	Fall Velocity vs Particle Size	252
A. 3	Wind Streamlines for $\alpha = 0.25$	254

LIST OF ILLUSTRATIONS (Cont'd.)

<u>Figure</u>		<u>Page</u>
A. 4	Wind Streamlines for $\alpha = 0.50$	255
A. 5	Wind Streamlines for $\alpha = 0.75$	255
A. 6	Differences Between Surface Trajectory and Mountain Ridge	257
A. 7	Fallout Particle Trajectories	258
A. 8	Fallout Particle Trajectories	259
A. 9	Fallout Particle Trajectories	259
B. 1	Surface Temperature Variation	279

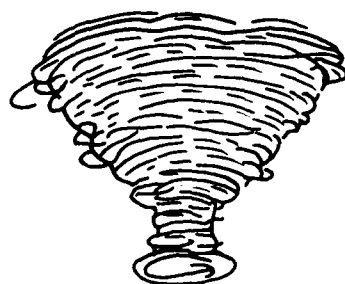
LIST OF FLOW CHARTS

<u>Flow Chart</u>	<u>Page</u>
FC-1 Flow Chart for Subroutine FALRAT	47
FC-2 Organizational Flow Chart for Subroutine DUMPP	51
FC-3 Detailed Flow Charts for Subroutine DUMPP	52
(a-e)	
FC-4 General Flow Chart for Subroutine LINK5	59
FC-5 Detailed Flow Charts for Subroutine LINK5	60
(a-e)	
FC-6 Flow Chart for Subroutine RDCIRS	66
FC-7 Organizational Flow Chart for Subroutine MKWIND	68
FC-8 Detailed Flow Charts for Subroutine MKWIND	69
(a-g)	
FC-9 General Flow Chart for Subroutine LINK7	83
FC-10 Detailed Flow Charts for Subroutine LINK7	84
(a-l)	
FC-11 Flow Chart for Subroutine GETWND	96
FC-12 Flow Charts for Subroutine LOTRAN	98
(a-b)	
FC-13 Flow Charts for Subroutine MTWND1	101
(a-c)	
FC-14 Flow Charts for Subroutine RGWND1	106
(a-c)	
FC-15 Flow Charts for Subroutine CBREZ1	116
(a-e)	
FC-16 Flow Charts for Subroutine HEIGHT	122
(a-b)	
FC-C.1 Flow Chart of Main Program TOPIN	292
FC-C.2 Flow Charts of Subroutine DATERR	293
(a-f)	

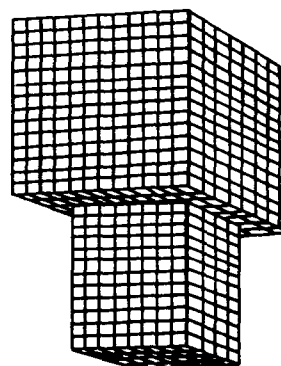
INTRODUCTION

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.

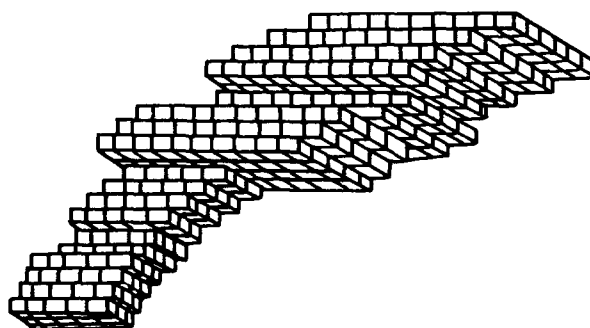
A set of fallout particles chosen as representative of the contents of cloud subdivisions is prepared by the Cloud Rise-Transport Interface program of the Cloud Rise Module. The generation of this input is described in detail in Volume III of this documentation; here we review only its essential highlights. 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

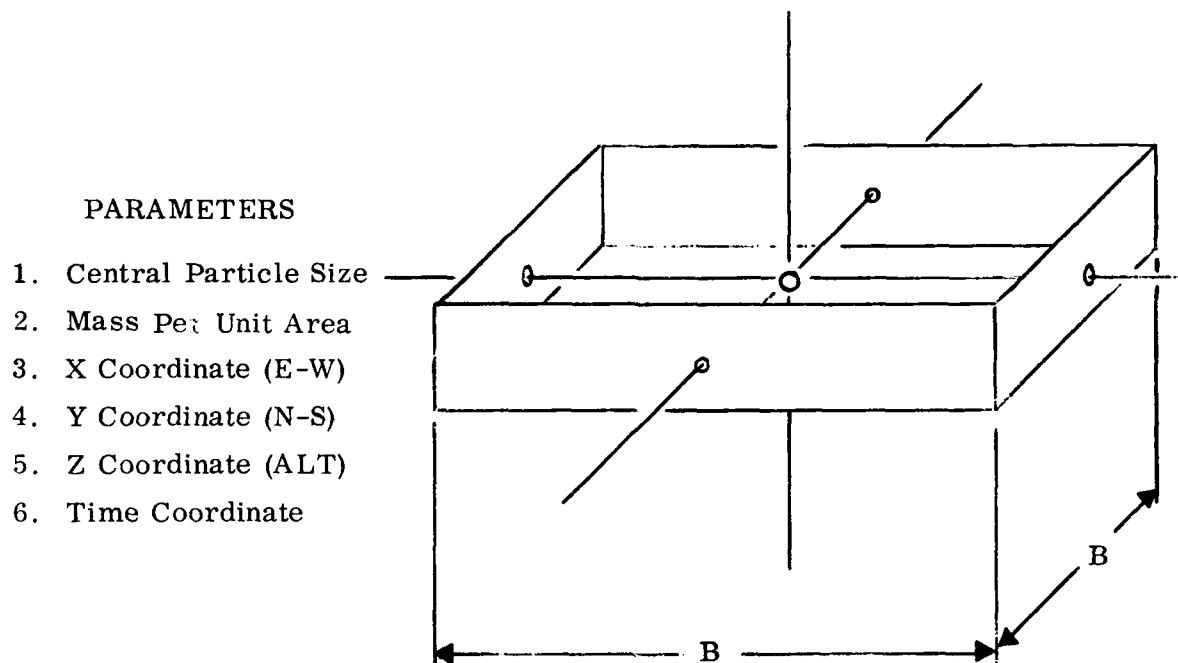


Figure 2. The Elementary Cloud Subdivision and Its Characterization
(B is the dimension of all cloud subdivisions at the time of their definition.)

(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

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

and the displacement equation

$$\frac{d\tilde{r}}{dt} = \tilde{V}_p, \quad (2)$$

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

$$\tilde{F} = - \left(\tilde{V}_p - \tilde{V}_w \right) \phi \left(|\tilde{V}_p - \tilde{V}_w| \right) \quad (3)$$

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

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

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

$$V_{px} = U , \quad (4)$$

$$V_{py} = V , \quad (5)$$

and

$$V_{pz} = -V_F + W , \quad (6)$$

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 Davies² 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

$$C_D R^2 = \frac{4G\rho_p d^3}{3\eta^2} , \quad (7)$$

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:

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

or

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

3. The settling velocity V_F is computed from

$$V_F = \frac{R\eta}{\rho d}, \quad (10)$$

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

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

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

We have concluded¹ 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 $2X_0$, $2Y_0$ (see Figure 3) is assumed to move

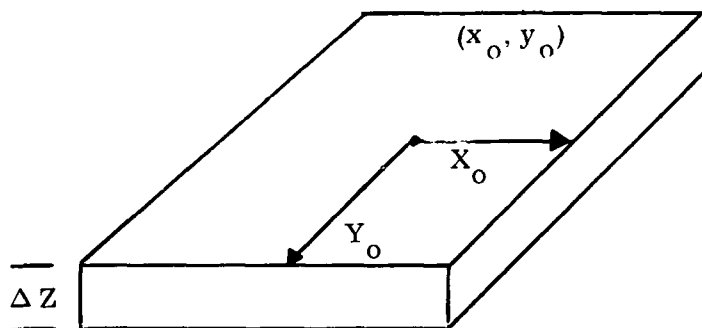


Figure 3. Segment of Cloud Volume

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

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

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

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

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

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

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:

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

On the other hand, we have

$$\overline{x^2} = \frac{\int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} x^2 P(x, y, t) dx dy}{4\sigma_o X_o Y_o} = Dt + 1/3 X_o^2, \quad (16)$$

and

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

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

$$X^2 = X_o^2 + 3Dt, \quad (17)$$

and

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

with corresponding lateral density

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

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

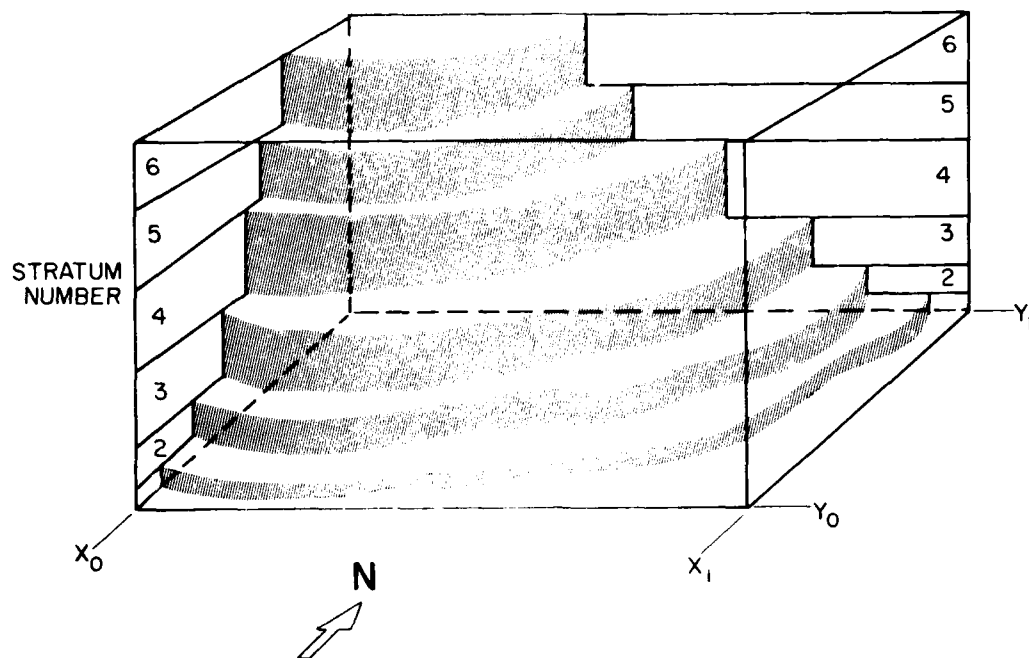


Figure 4. Strata within the Specified Wind Field Volume
(illustrated for six strata)

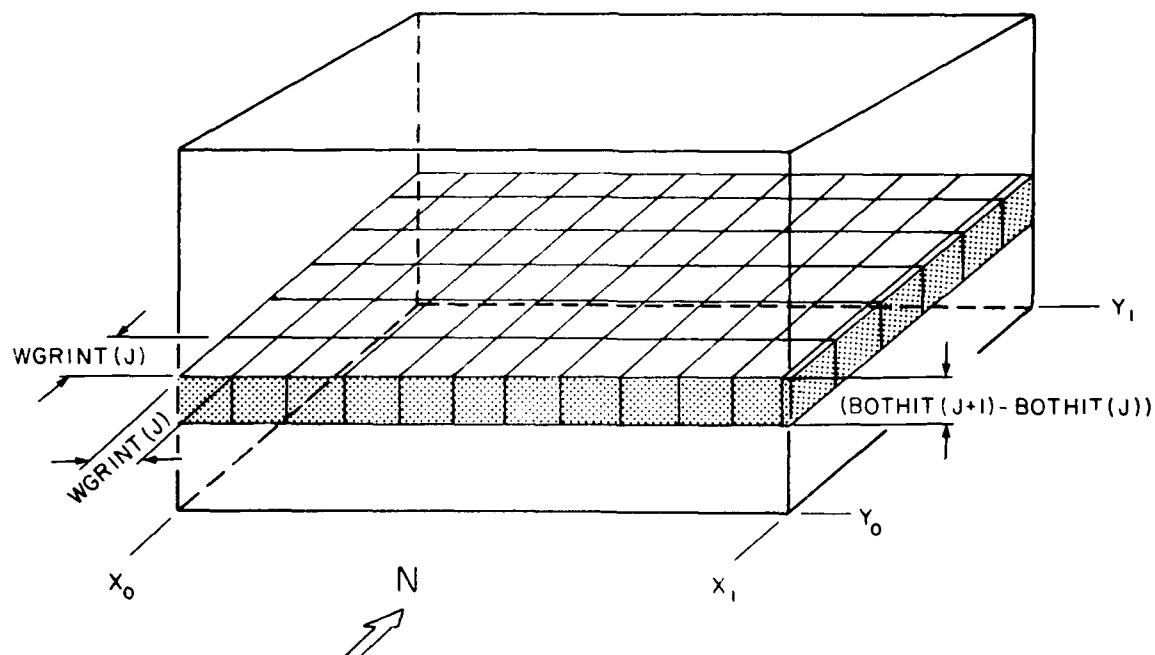


Figure 5. Wind Cells — Subdivisions of a Stratum (illustrated for the Jth stratum from the bottom)

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:

$\underline{R}_i$ = position of ith observed wind velocity vector relative to the wind-field-array grid point $\underline{R}_0$

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

$\underline{V}_0$ = wind velocity at a wind-field-array grid point $\underline{R}_0$. $\underline{V}_0$ is to be determined from $\underline{R}_i$ and $\underline{V}_i$.

The Closest Datum Method. 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.

The Preferential-Weighting Method. In the preferential weighting method $\underline{V}_0$ 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 $\underline{V}_0$ and $\underline{V}_i$ is given by

$$\underline{V}_0 = \sum_{i=1}^N f_i \underline{V}_i \quad (19)$$

where

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

A weighting method described by Cressman⁴ has been used in deriving an expression for f_i in the form

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

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

$$\begin{aligned} u &= u_o + (\nabla u)_o \cdot \underline{R} , \\ v &= v_o + (\nabla v)_o \cdot \underline{R} , \end{aligned} \quad (22)$$

and

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

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 , $(\nabla u)_o \equiv \underline{A}$, $(\nabla v)_o \equiv \underline{B}$, and $(\nabla w)_o \equiv \underline{C}$. Actually, the computation breaks down into three separate parts involving $(u_o, \underline{A})$, $(v_o, \underline{B})$, and $(w_o, \underline{C})$. 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 i th sounding station, the i th residual is given by

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

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

$$F(u_o, \underline{A}) = \sum_{i=1}^N \xi_i^2 \quad (24)$$

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

$$\frac{\partial F}{\partial u_0} = 0 = - \sum U_i + \sum (u_0 + \bar{A} \cdot \bar{R}_i) , \quad (25)$$

$$\frac{\partial F}{\partial A_x} = 0 = - \sum U_i x_i + \sum (u_0 + \bar{A} \cdot \bar{R}_i) x_i , \quad (26)$$

$$\frac{\partial F}{\partial A_y} = 0 = - \sum U_i y_i + \sum (u_0 + \bar{A} \cdot \bar{R}_i) y_i , \quad (27)$$

and

$$\frac{\partial F}{\partial A_z} = 0 = - \sum U_i z_i + \sum (u_0 + \bar{A} \cdot \bar{R}_i) z_i . \quad (28)$$

Introducing the averaged quantities,

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

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

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

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

and

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

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

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

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

$$u_o = \gamma_1 \bar{u} + \gamma_2 \overline{ux} + \gamma_3 \overline{uy} + \gamma_4 \overline{uz} \quad (31)$$

where

$$\begin{aligned} \gamma_1 &= \frac{B^{11}}{|B|} \quad , \\ \gamma_2 &= \frac{B^{21}}{|B|} \quad , \\ \gamma_3 &= \frac{B^{31}}{|B|} \quad , \end{aligned} \quad (32)$$

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

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

and

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

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.

Local Circulation Systems

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.

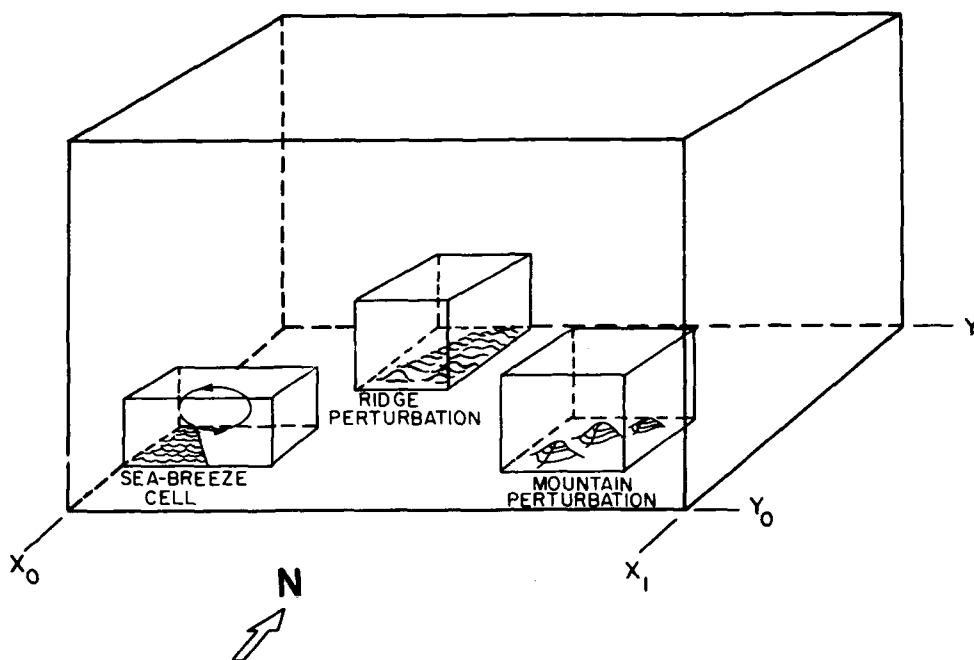


Figure 6. Wind-Field Volume with Superimposed Local Circulation System Cells

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_0 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_0 , and let the y axis be perpendicular to u_0 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

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

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

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

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:

$$u(x, y, z) = u_0 \left[1 + (a^2 h) \frac{(y^2 + \lambda^2 - 2x^2)}{(r^2 + \lambda^2)^{5/2}} \right] , \quad (37)$$

$$v(x, y, z) = -3u_0 (a^2 h) \frac{xy}{(r^2 + \lambda^2)^{5/2}} , \quad (38)$$

and

$$w(x, y, z) = -3u_0 (a^2 h) \frac{\lambda x}{(r^2 + \lambda^2)^{5/2}} , \quad (39)$$

where

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

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

$$z \geq f(x, y) \quad (41)$$

(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_0 . The pertinent geometric details are shown in Figure 7.

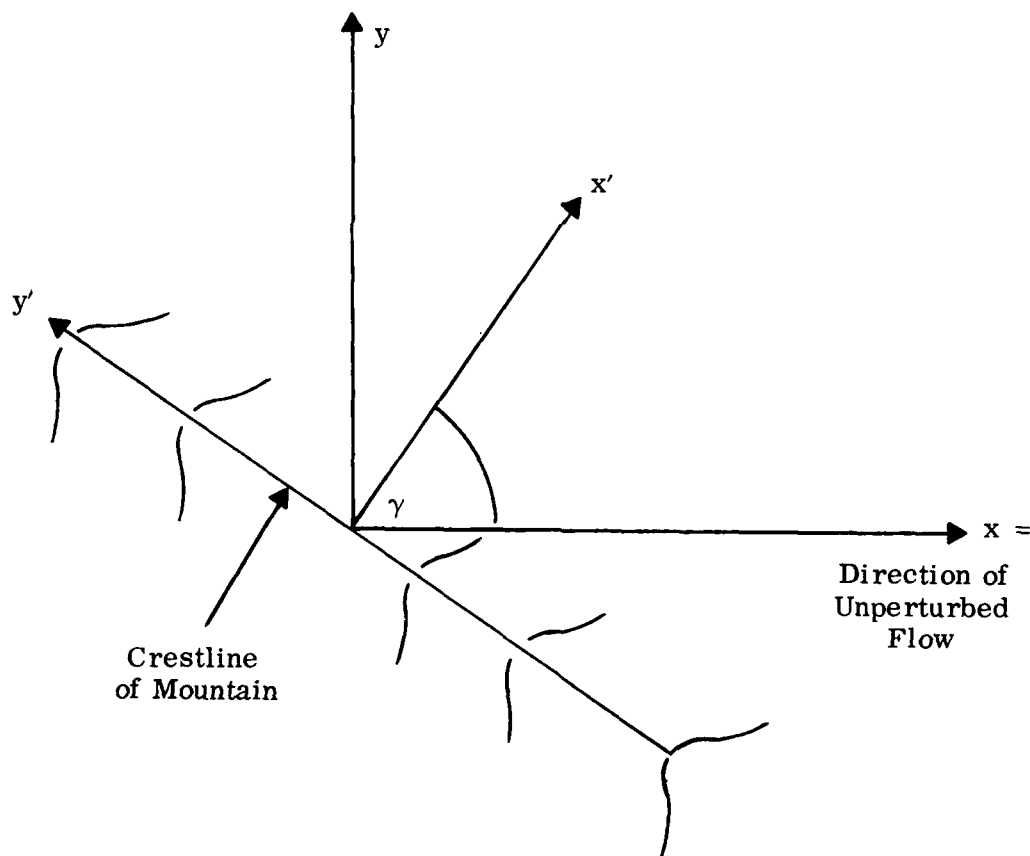


Figure 7. Mountain Ridge Not Perpendicular to Flow

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

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

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

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

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

$$u = u_0 - u_0(ah) \cos^2 \gamma \frac{(x \cos \gamma + y \sin \gamma)^2 - \lambda^2}{\left[(x \cos \gamma + y \sin \gamma)^2 + \lambda^2 \right]^2} , \quad (44)$$

$$v = -u_0(ah) \cos \gamma \sin \gamma \frac{(x \cos \gamma + y \sin \gamma)^2 - \lambda^2}{\left[(x \cos \gamma + y \sin \gamma)^2 + \lambda^2 \right]^2} , \quad (45)$$

and

$$w = -2u_0(ah) \lambda \cos \gamma \frac{(x \cos \gamma + y \sin \gamma)}{\left[(x \cos \gamma + y \sin \gamma)^2 + \lambda^2 \right]^2} , \quad (46)$$

where

$$\lambda = z + a .$$

It should be carefully noted that u , v , and w do not 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

$$(x_i, y_i) .$$

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

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

$$v_i(x, y, z) = v(x - x_i, y - y_i, z) , \quad (47)$$

and

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

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

$$z \geq z_i^* = f_i(x - x_i, y - y_i) \quad (48)$$

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

$$z_T^* = \sum_i f_i(x - x_i, y - y_i) , \quad (49)$$

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

$$\begin{aligned} u(x, y, z) &= \sum_i u_i(x - x_i, y - y_i, z) , \\ v(x, y, z) &= \sum_i v_i(x - x_i, y - y_i, z) , \end{aligned} \quad (50)$$

and

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

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).

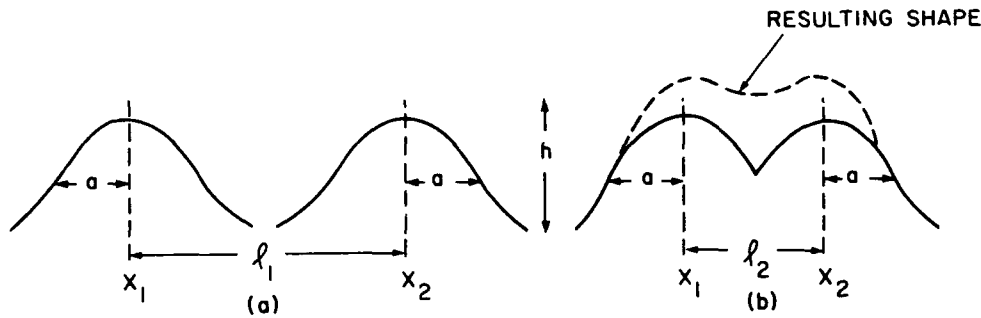


Figure 8. Mountain Ridge Separations

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

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

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

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

which if $\ell_1 \gg a$ 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

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

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. Size Limitations

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 $\Delta \underline{v}(x, y, z)$ and the curvature of the terrain is deduced by first expressing the three components of $\Delta \underline{v}$ (namely $\Delta v_i(x, y, z)$) in a spatial Fourier transform representation,

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

and then solving for the $A_i(\underline{k})$. The solution for the $A_i(\underline{k})$ 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

$$k_z = k_z(k_x, k_y) ,$$

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

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

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, Δu , as compared to u_0 diminishes. As shown in the analysis, Δu 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 .

$$S = (h/a) \text{ .}$$

However, by performing the calculations with a larger slope,

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

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

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

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

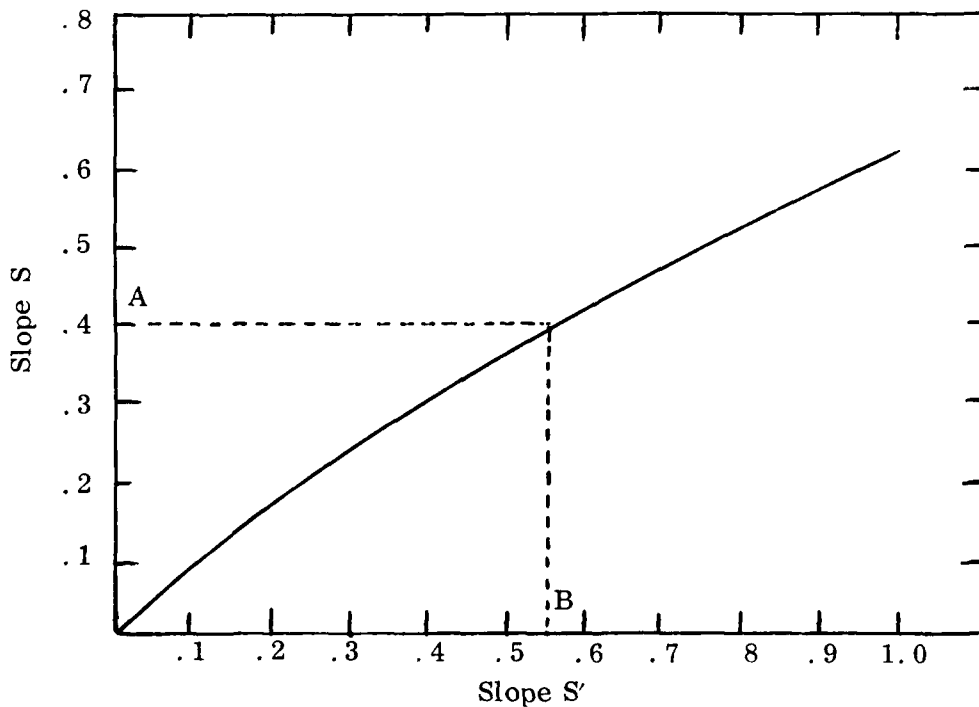


Figure 9. Slope Compensation

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. Defant⁵ 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

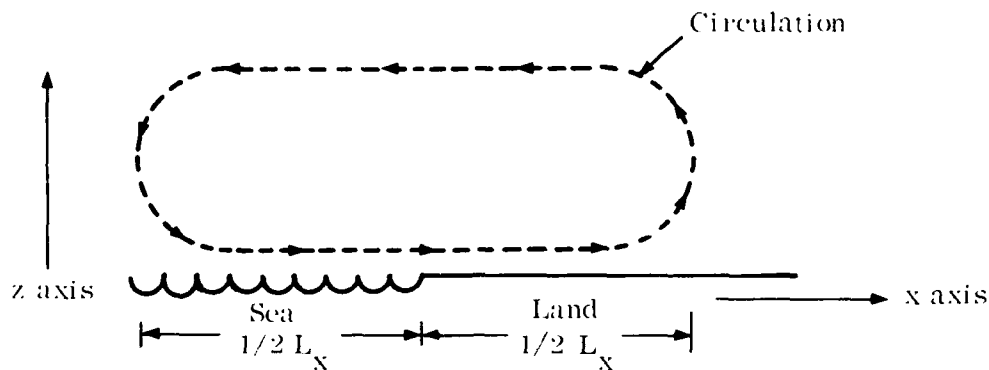


Figure 10. Sea-Breeze Circulation

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

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

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.

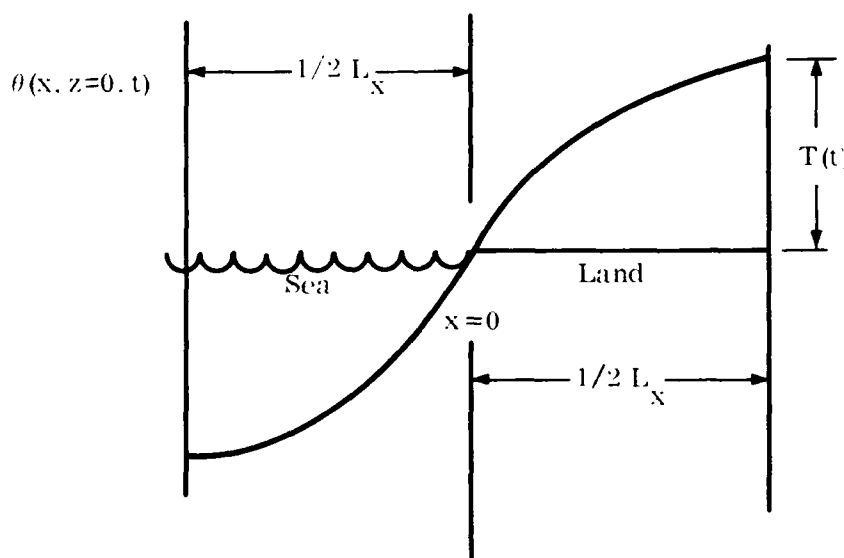


Figure 11. Temperature Differential Profile

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

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

where

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

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_0 , the average ground temperature; $\Gamma = (d\theta_0/dz)$, the initial unperturbed temperature gradient; and $\sin \phi$, 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:

$$u(x, y, z, t) = \sum u_n(x, y, z, t) \quad (57)$$

$$v(x, y, z, t) = \sum v_n(x, y, z, t) \quad (58)$$

and

$$w(x, y, z, t) = \sum w_n(x, y, z, t) \quad (59)$$

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:

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

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

and

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

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_0 , 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 $2.5 \times 10^{-4} \text{ sec}^{-1}$ 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

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

and

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

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

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

and

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

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 stippled 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 stippled 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 idealization, we have decided to define the wind field in the hatched region by the relationship

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

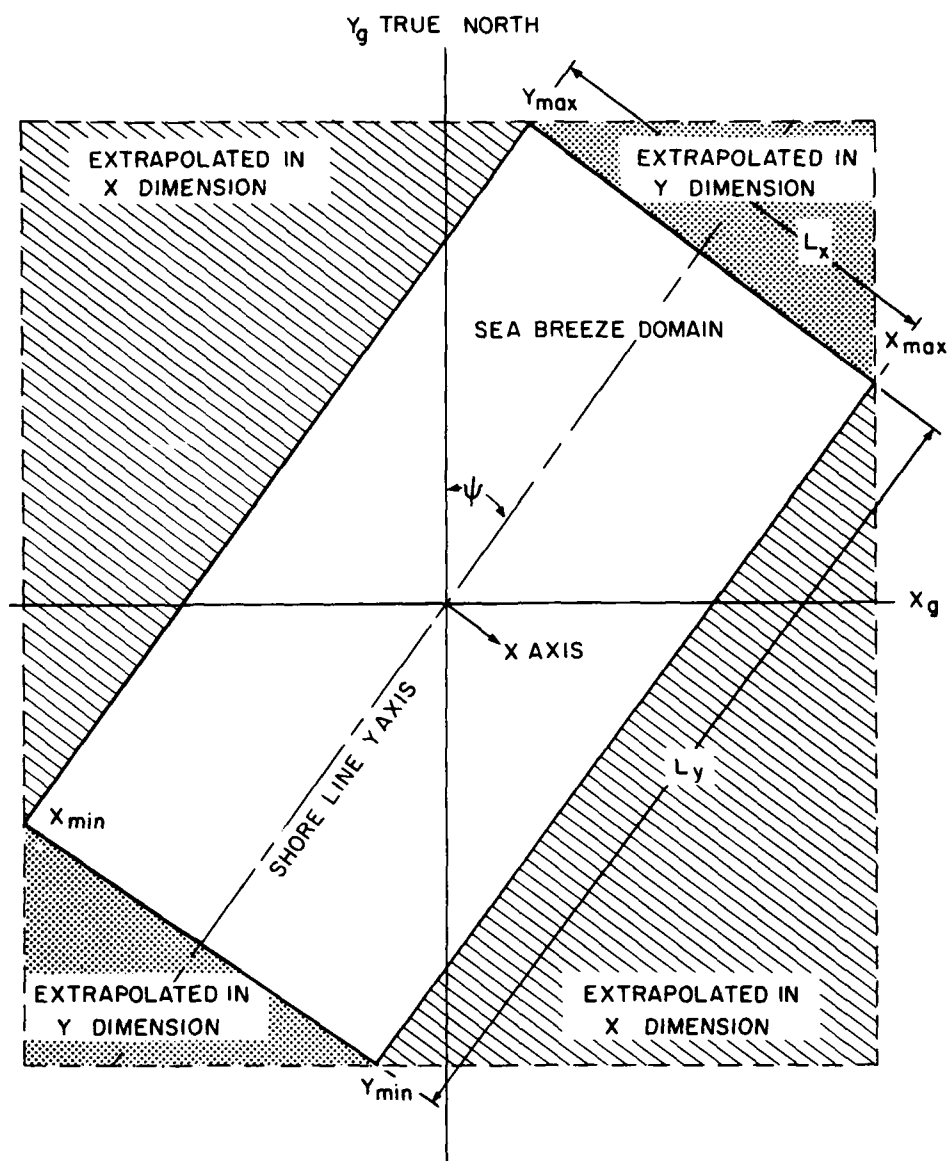


Figure 12. Geometric Considerations Related to the Sea Breeze

where $\tilde{V}_c$ 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

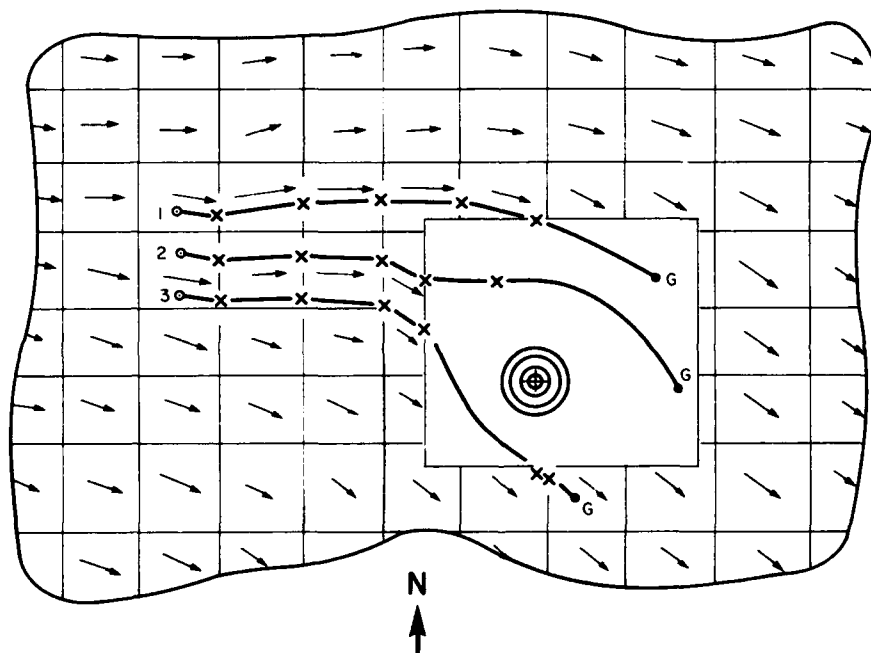


Figure 13. Boundary-to-Boundary Transport and a Mountain Wind Cell

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 $n + 1$ th 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

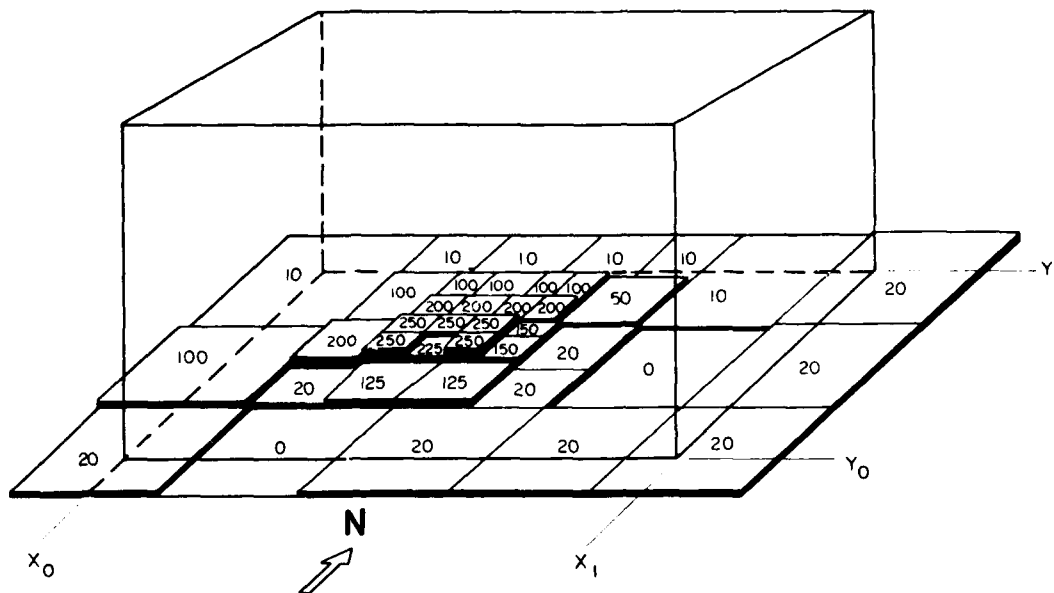


Figure 14. Piecewise-Planar Topography Specification Below the Macrowind-Field Volume (numbers are surface heights; vertical scale is exaggerated)

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 computations 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.

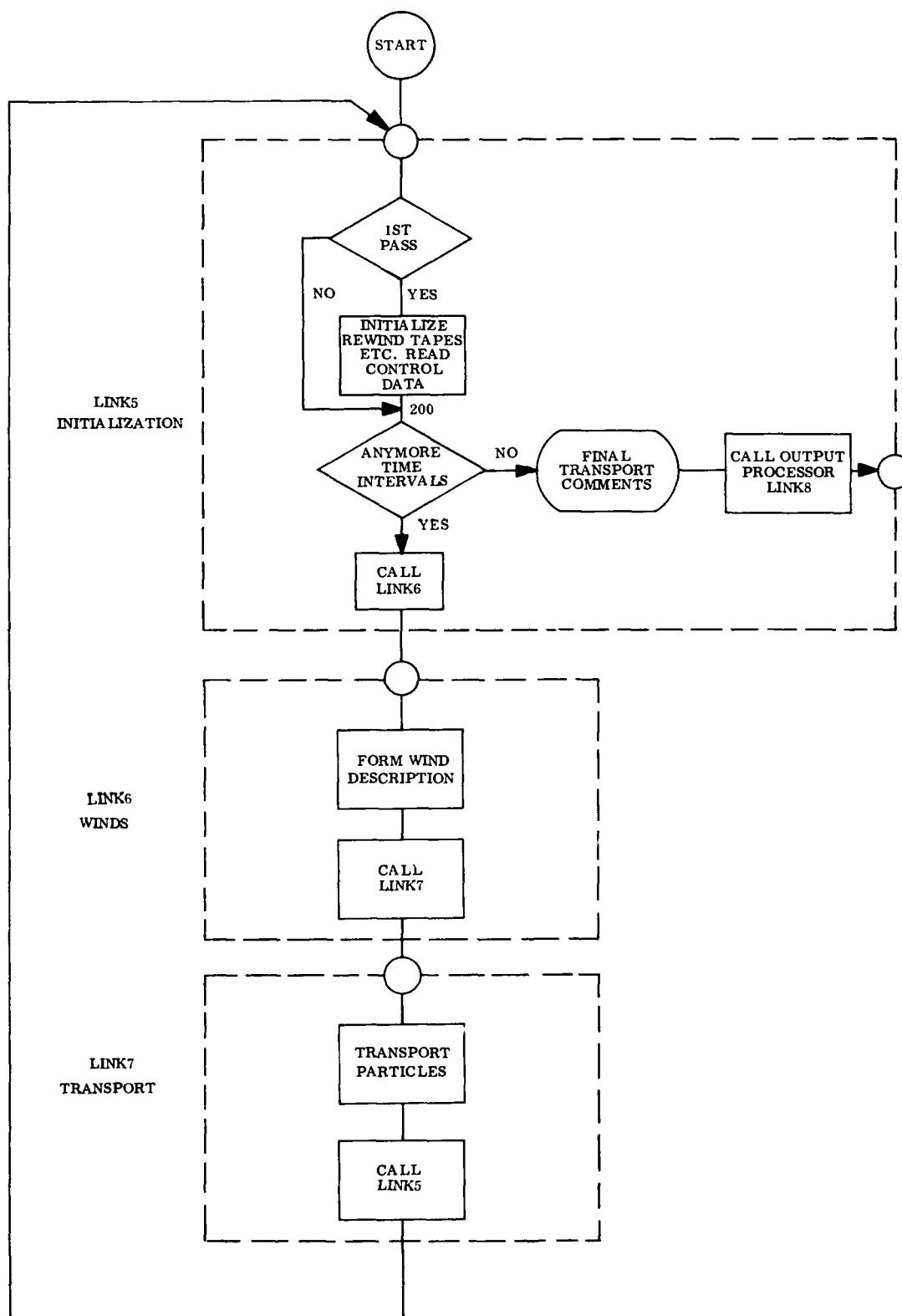


Figure 15. Program Arrangement for the Transport Module

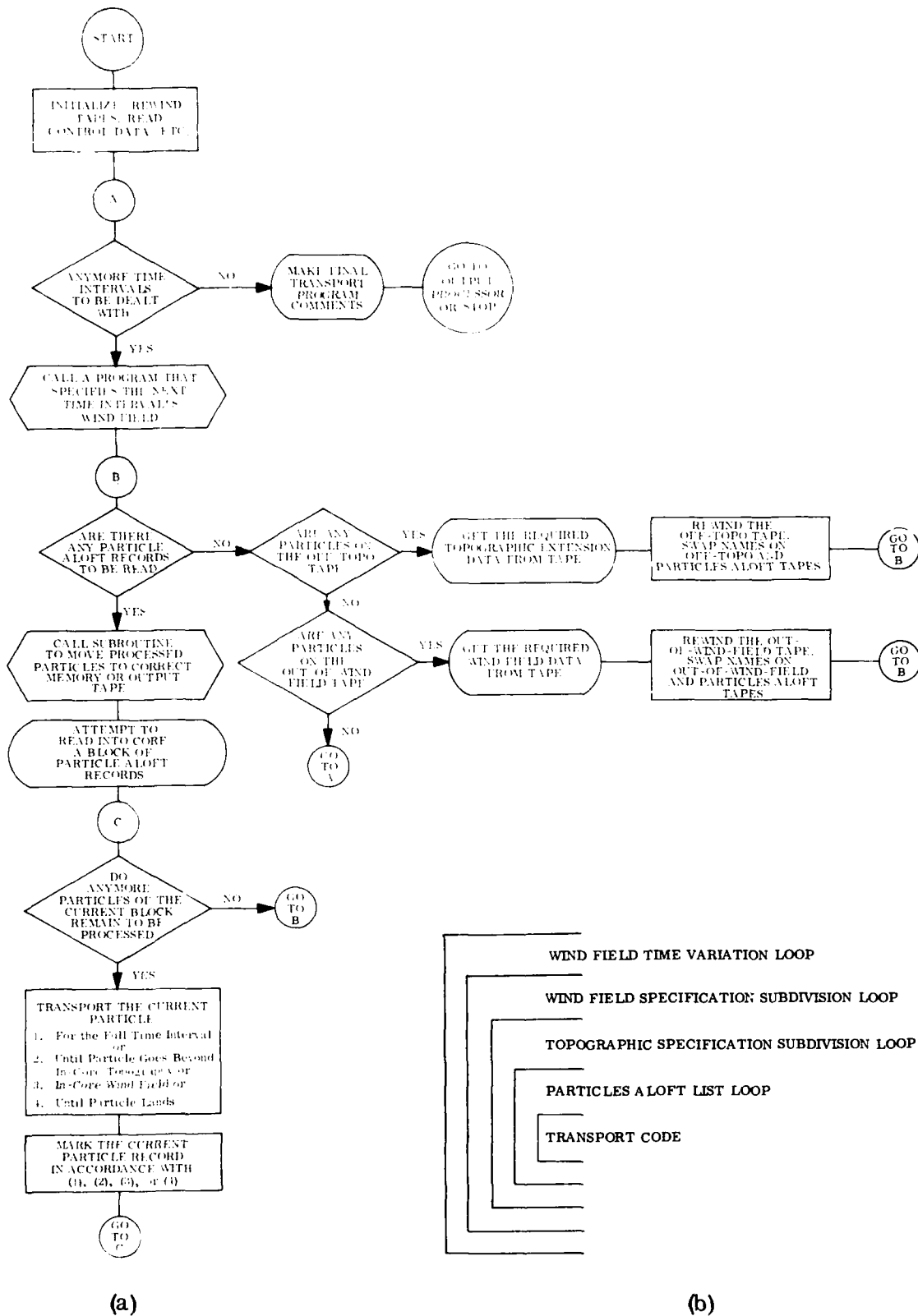


Figure 16. General Flow Chart of the Transport Module (a), and Transport Module Loops (b)

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.

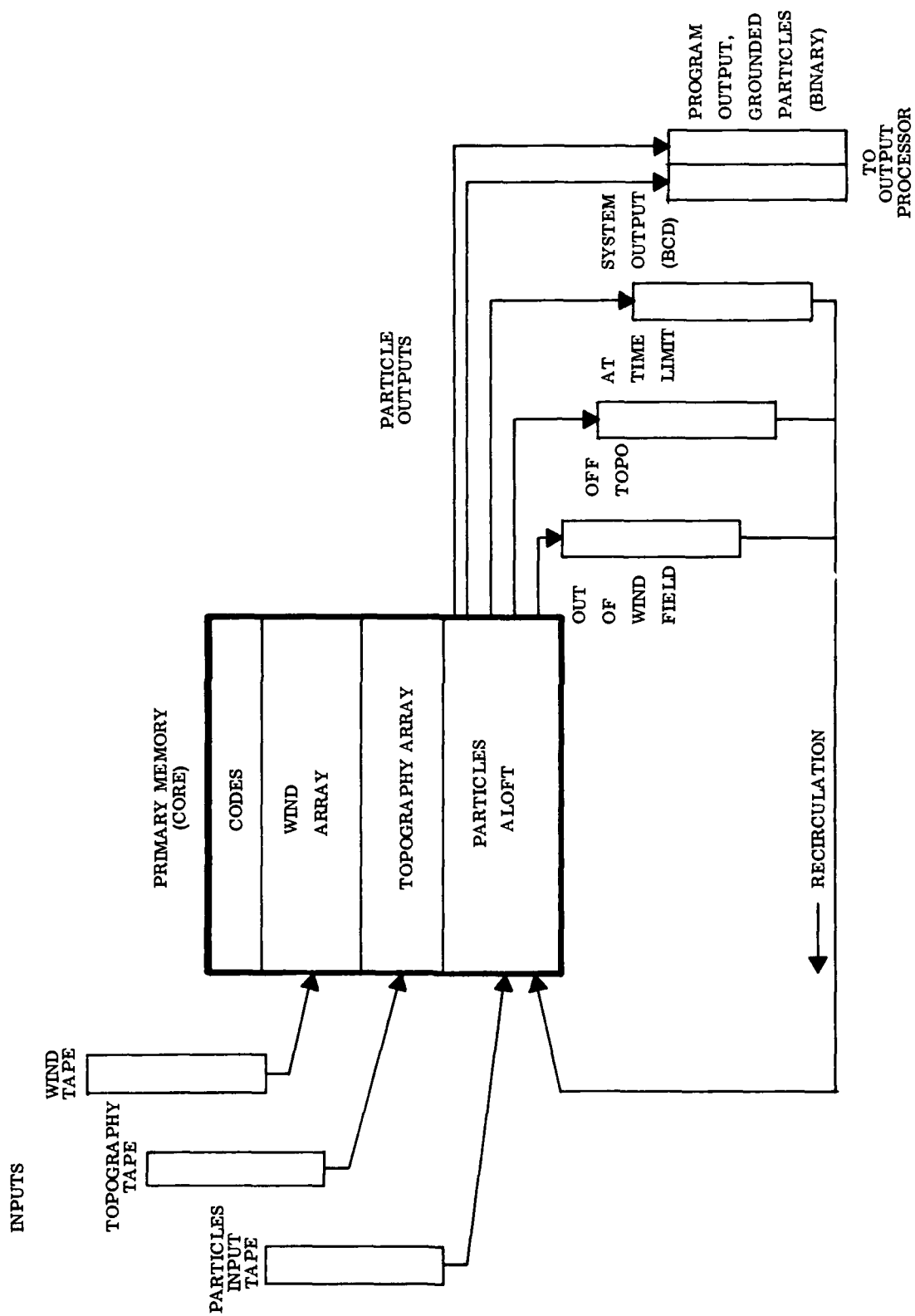


Figure 17. Transport Module Data Flow

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)

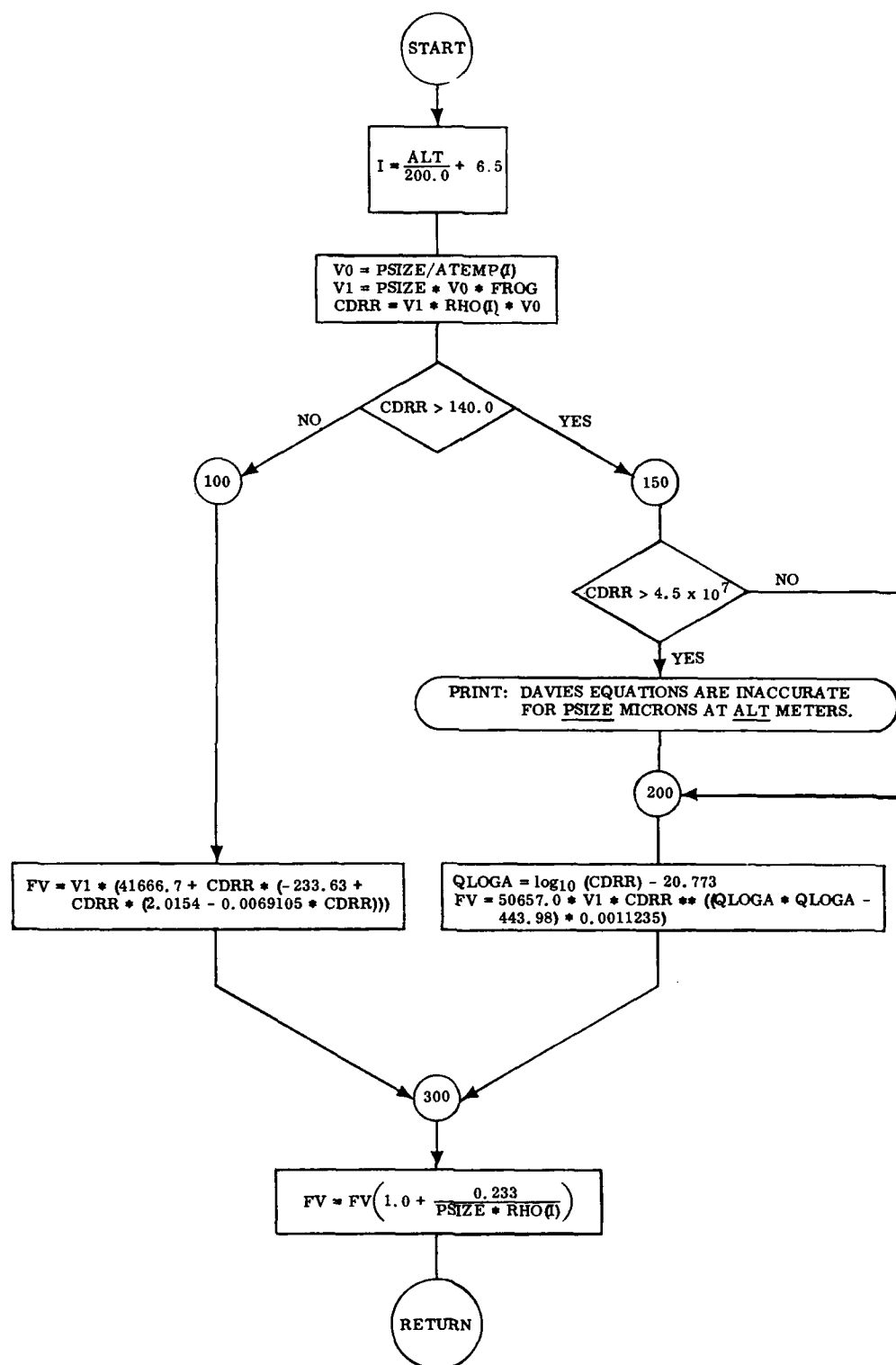
This subroutine computes the settling rate of a particle at height ALT in an atmosphere for which the density and dynamic viscosity are tabulated in arrays RHO and ATEMP respectively. These tabulations must be for 200 m intervals starting from 1000 m below MSL.† Fall rate equations derived by Davies² 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 $4/3 * g * \text{ROPART} * 10^{-8}$ 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 $C_D R^2$ are valid over separate ranges of $C_D R^2$. The separation occurs at $C_D R^2 = 140$. An overall upper limit of $C_D R^2 = 4.7 \times 10^7$ 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 $C_D R^2 > 4.7 \times 10^7$. 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 $\text{CDRR}(C_D R^2)$ 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 $\text{CDRR} > 4.7 \times 10^7$. 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 detailed 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.

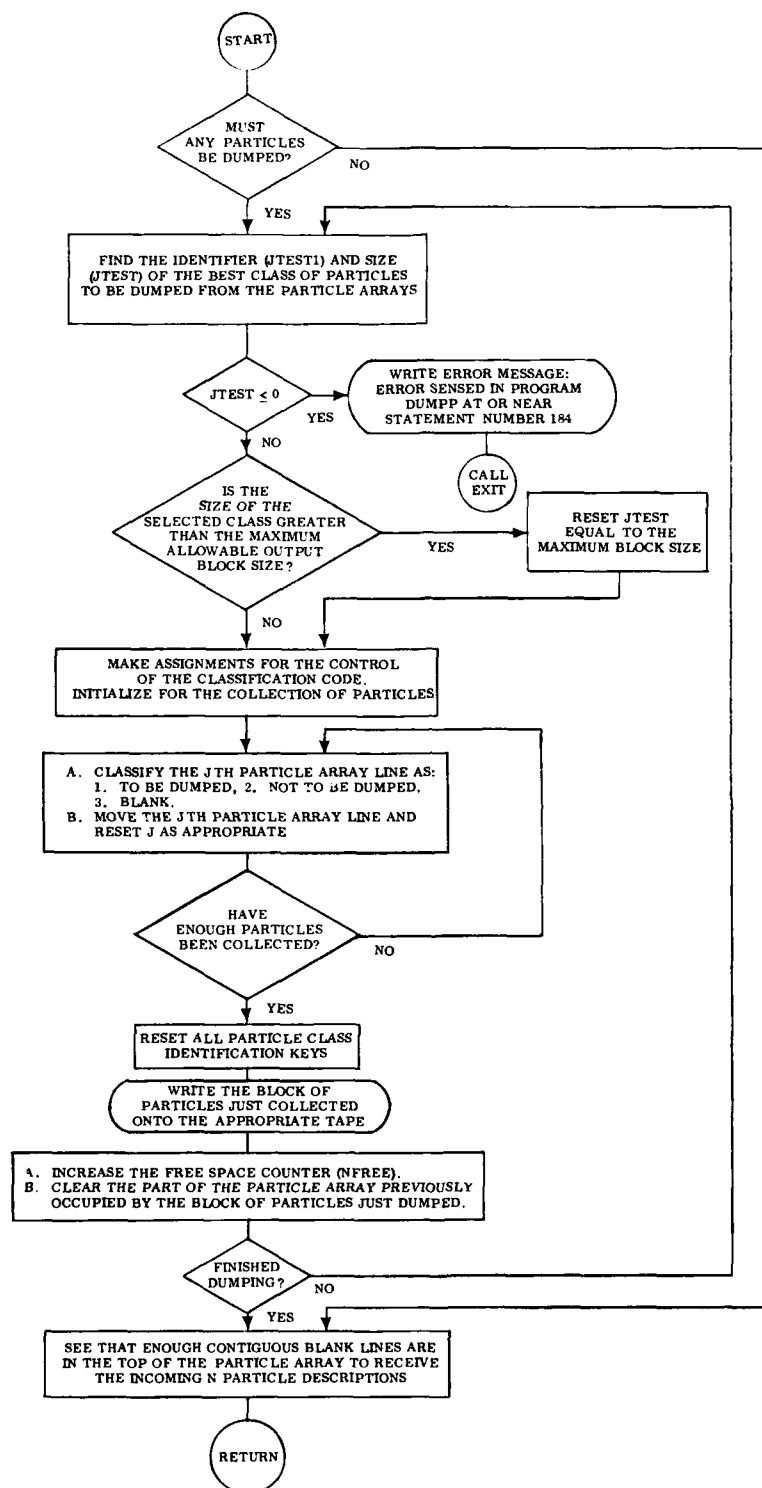
At 18 a safety test leading to an error stop is carried out followed by a threshold test on the size of the set to be dumped. Because a limit exists on the size of any particle block read by the output processor (see DASA-1800-VI), and also because we must impose block size control to allow for recirculation 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.

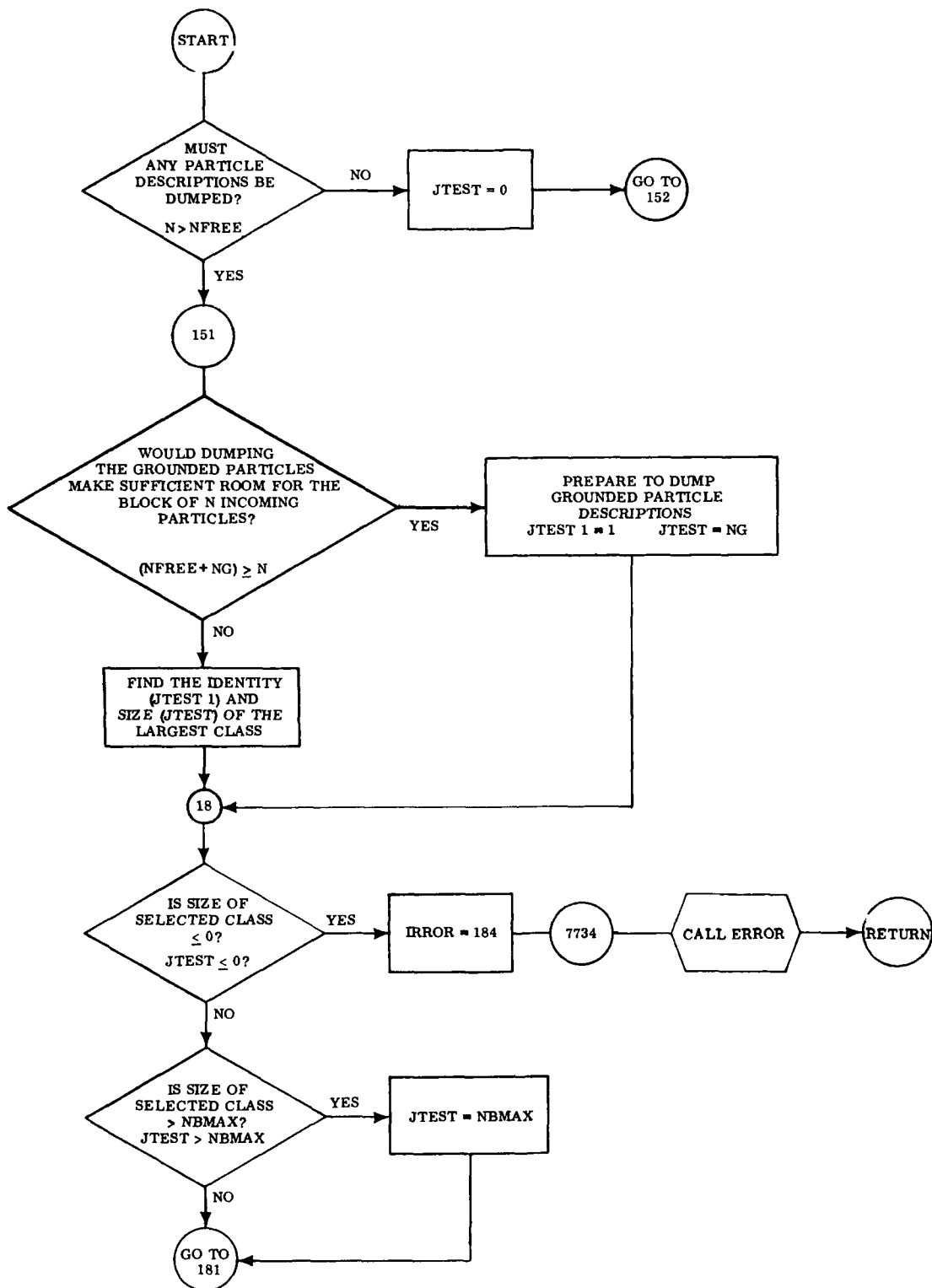
At 99 a one-line summary printout of information on the particle block to be dumped and of the particle counts is executed. Specifically, this output consists of the following data in order of printing from left to right: JTEST, JTEST1, and the current (predump) values of the in-core counts for blanks, grounded particles, lost particles, topography boundary particles, time boundary particles, and wind-field 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.

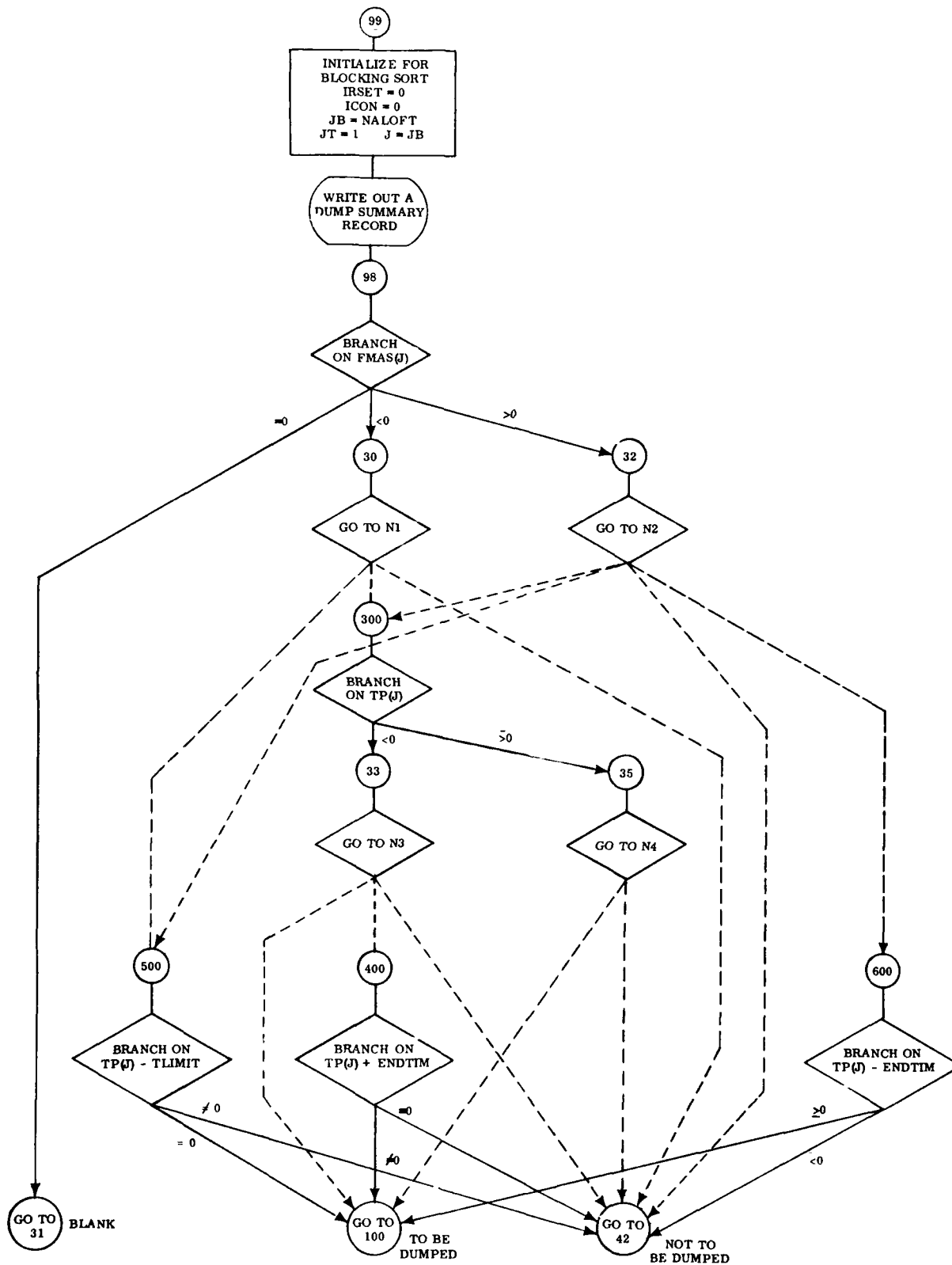


FC-2. Organizational Flow Chart for Subroutine DUMPP



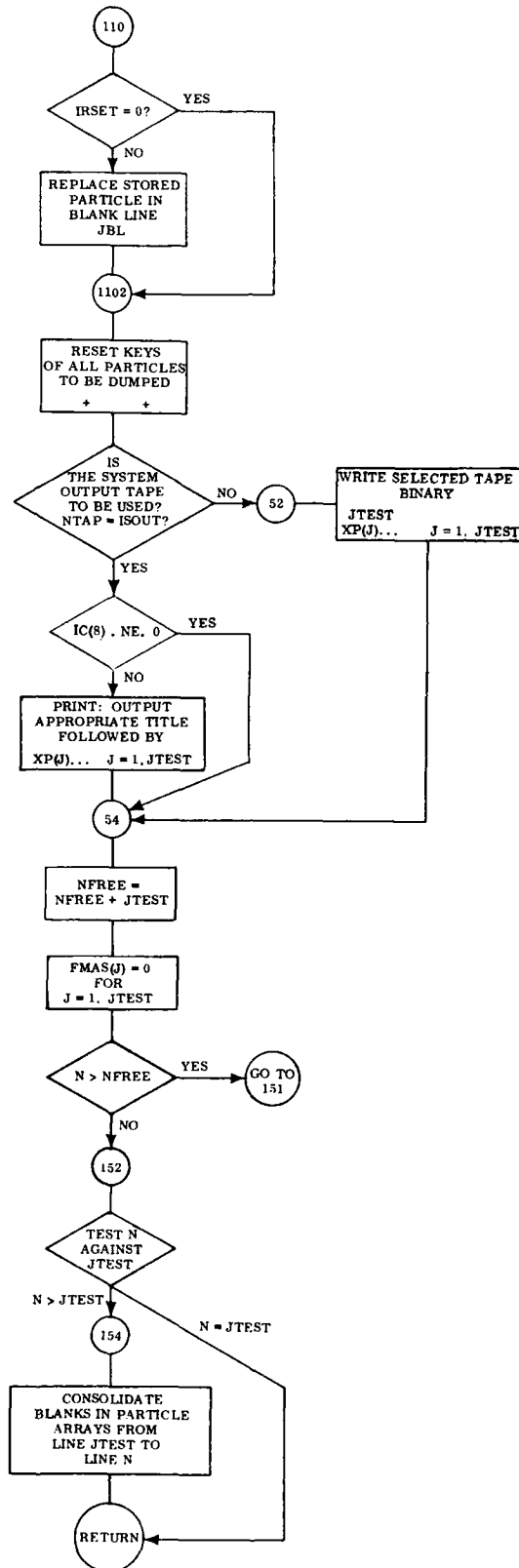
(a)

FC-3. Detailed Flow Charts for Subroutine DUMPP



(c)

FC-3. (Continued) Detailed Flow Charts 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.

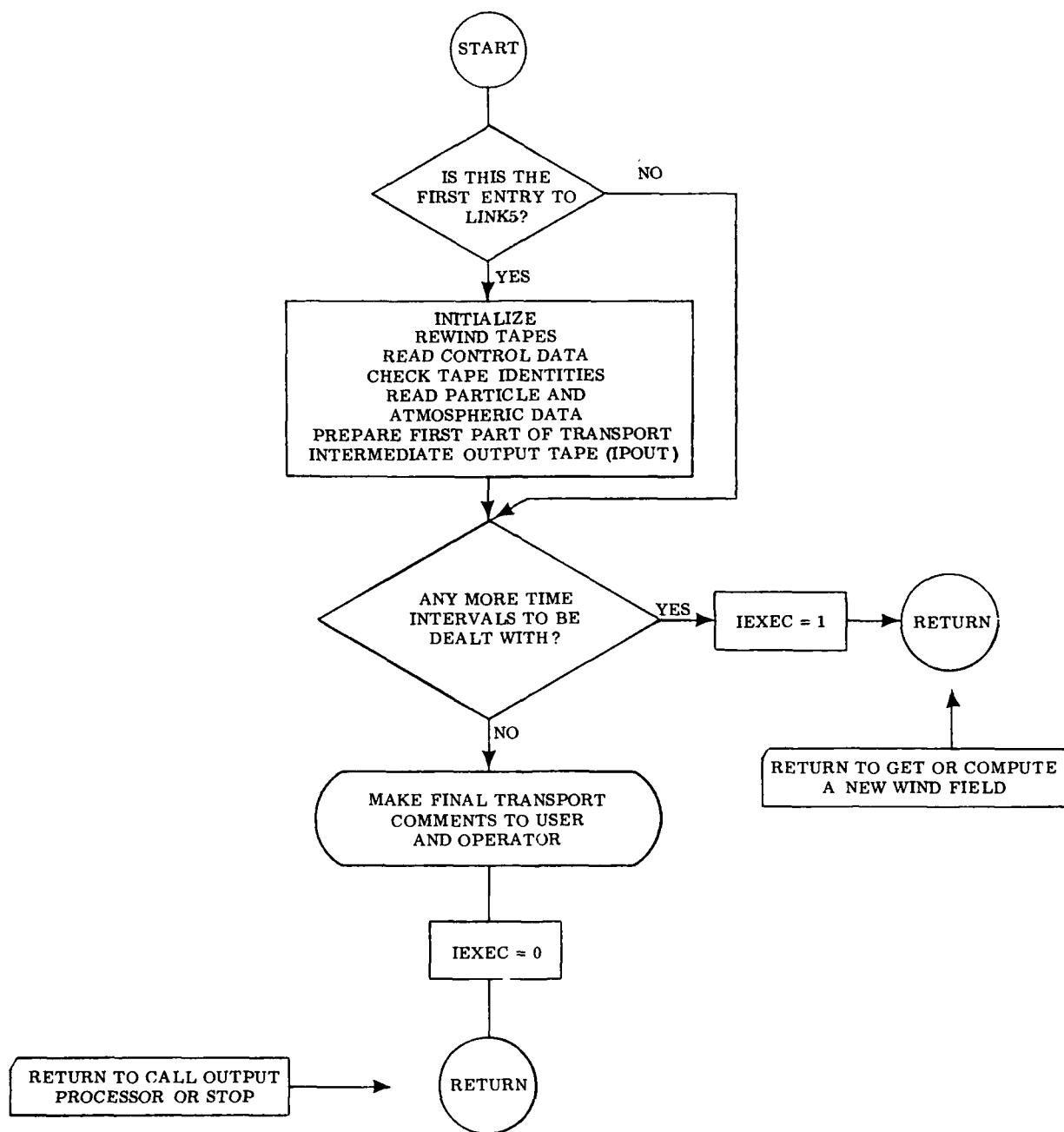
*Tape IPARIN has been prepared by the Cloud Rise-Transport Interface Module. See DASA-1800-III.

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.

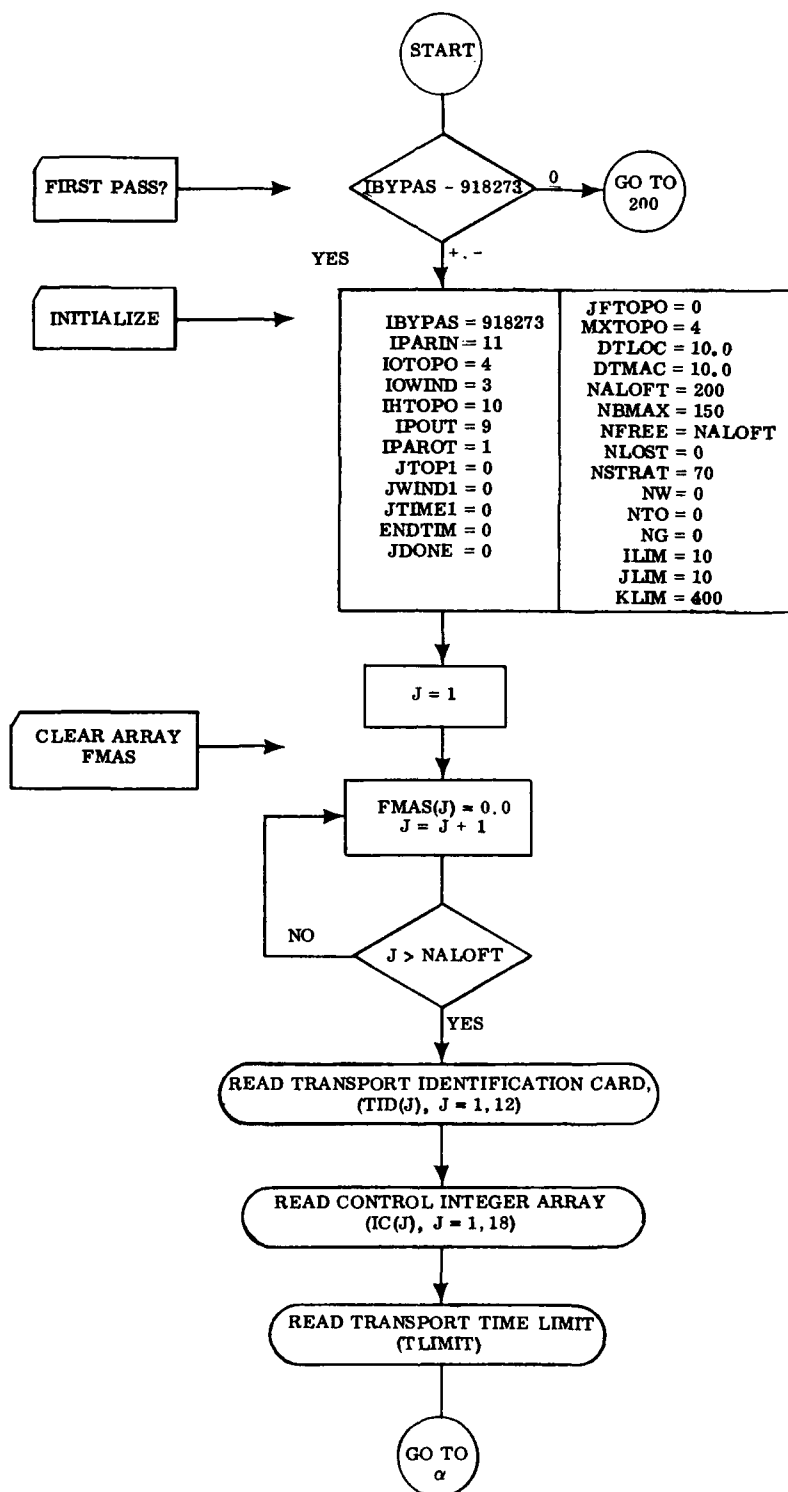
At 205 the program reads a wind-field data set identifier from the system input tape and transfers to the transport output tape (IPOUT) all identifiers and descriptive tables required by the output processor. Next LINK5 prints a title page for the transport run including identifiers and atmospheric data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

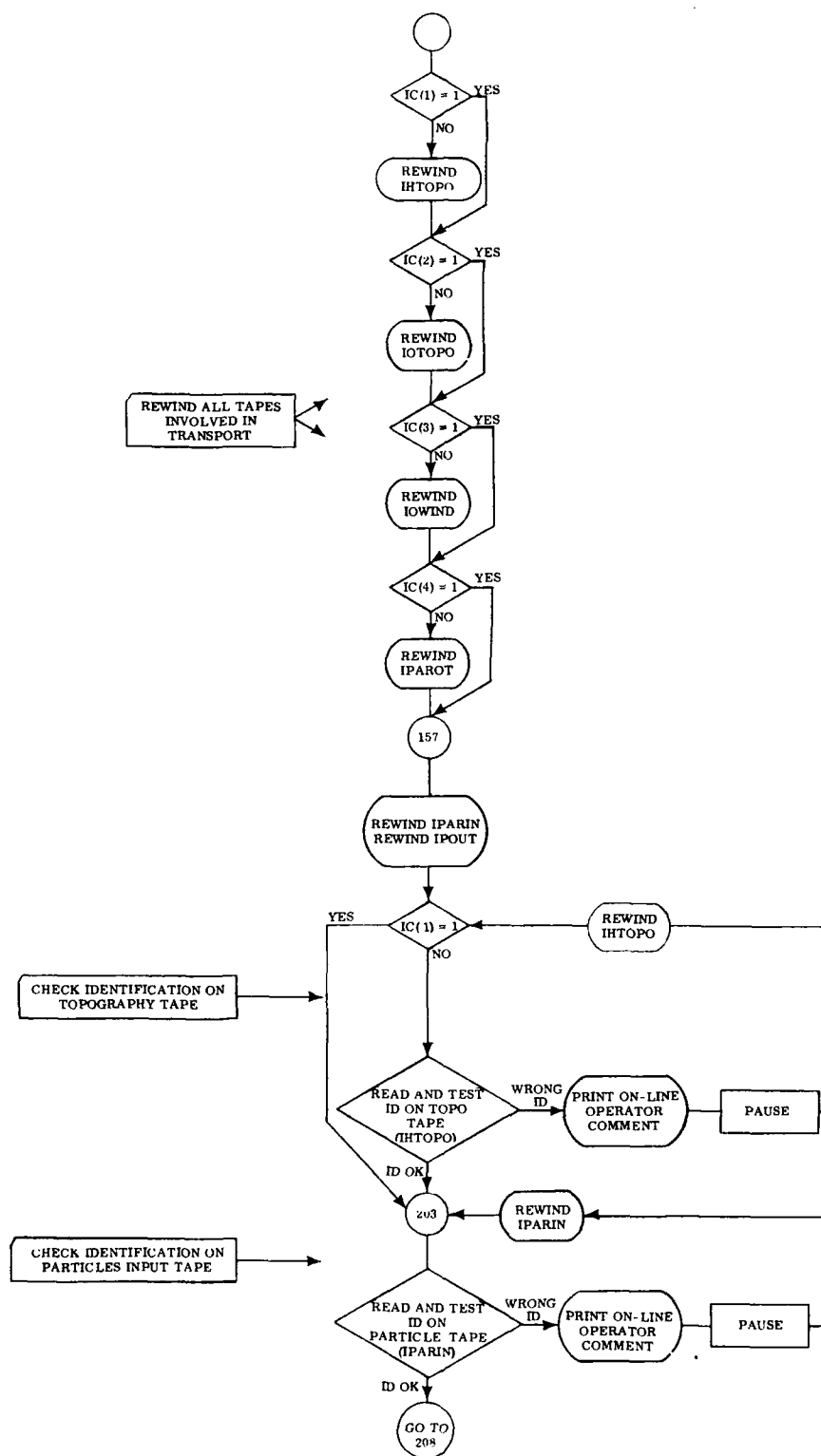


FC-4. General Flow Chart for Subroutine LINK5



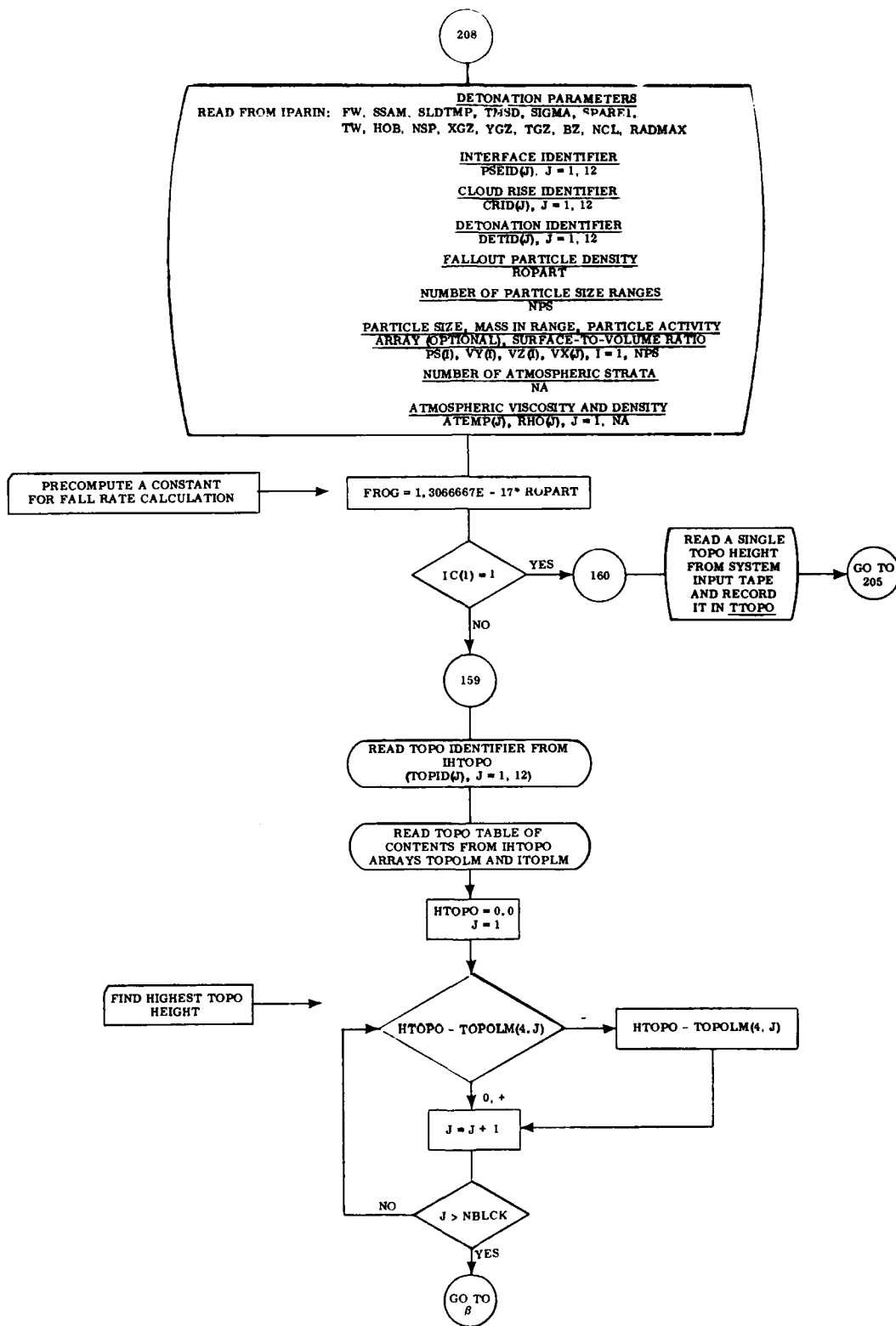
(a)

FC-5. Detailed Flow Charts for Subroutine LINK5



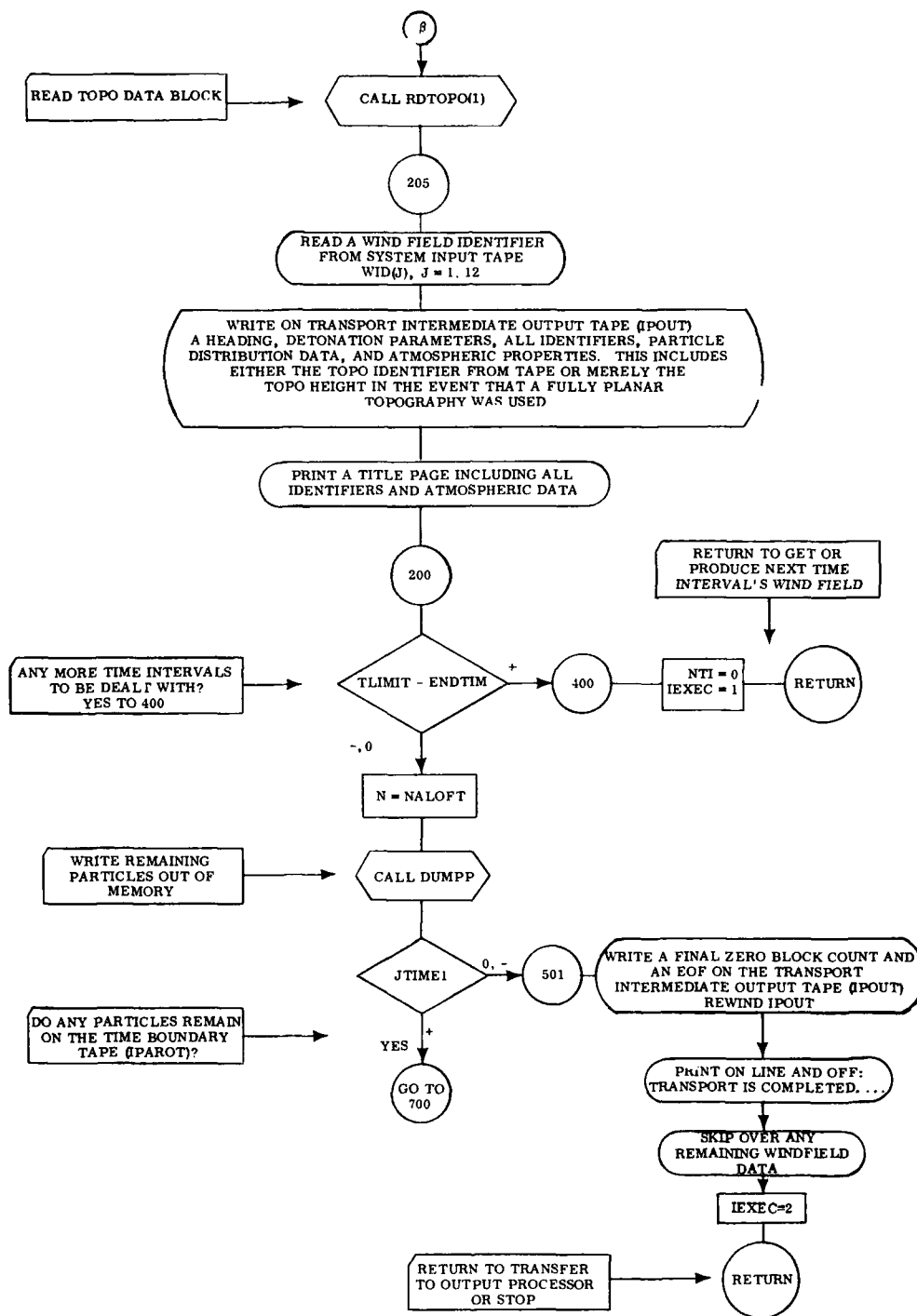
(b)

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



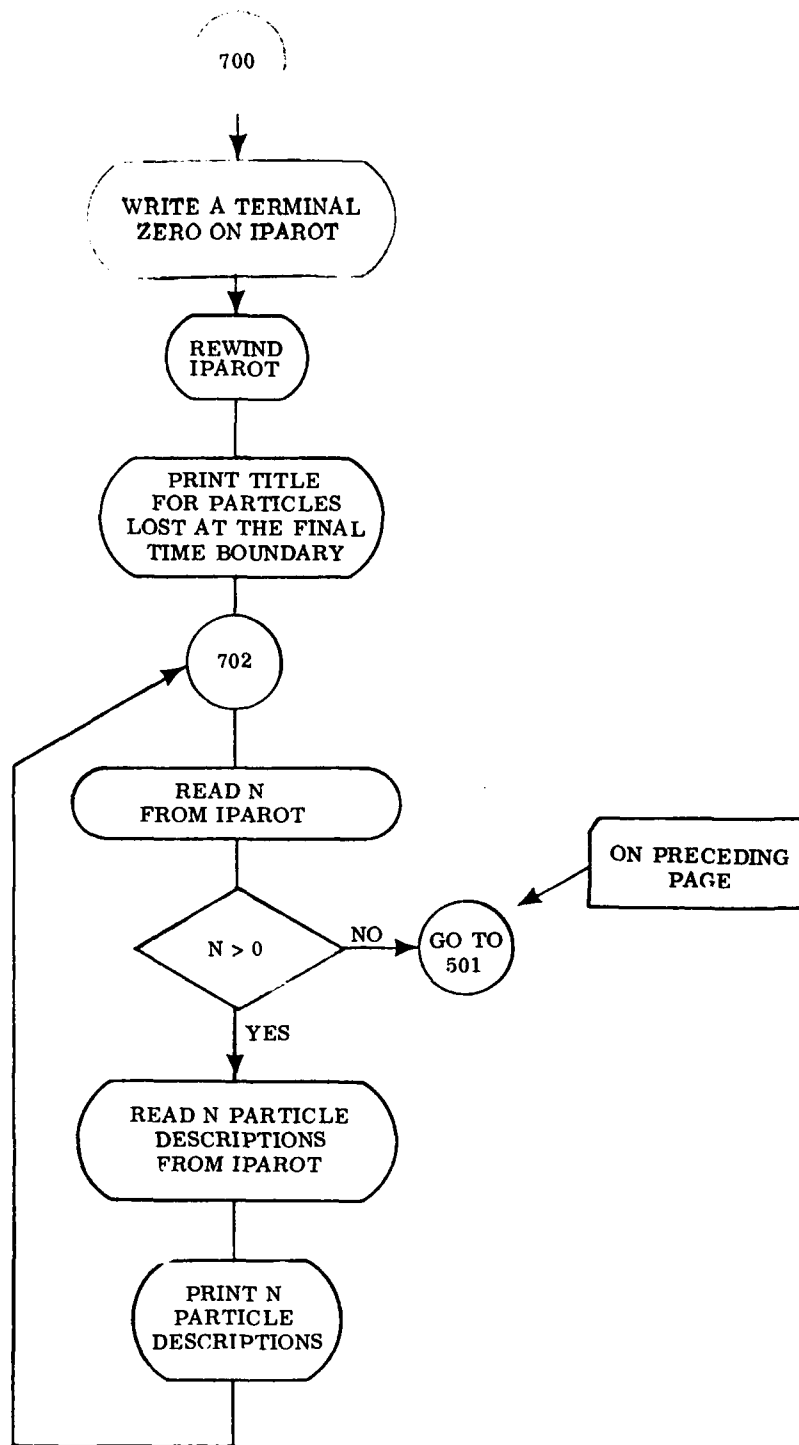
(c)

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)

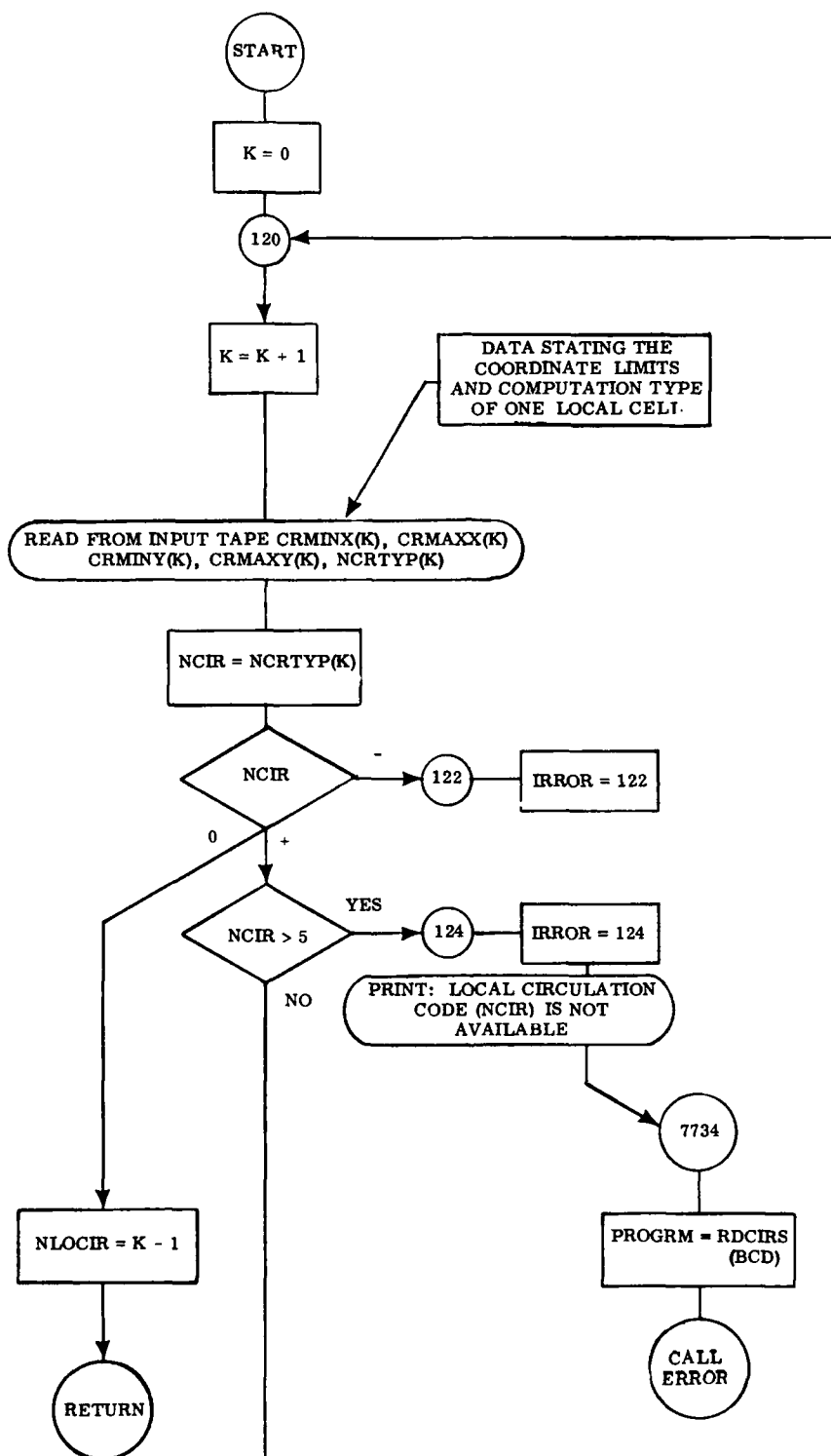
The purpose of this subroutine is to read a set of data which describes the geographical limits of the area covered by each of the local circulation systems that are to be used within the transport. Also, the identification number (NCRTYP(K)) for the computation code (local circulation model) applicable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

<u>Identification Number</u>	<u>Program to be Used</u>	<u>Model</u>
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)

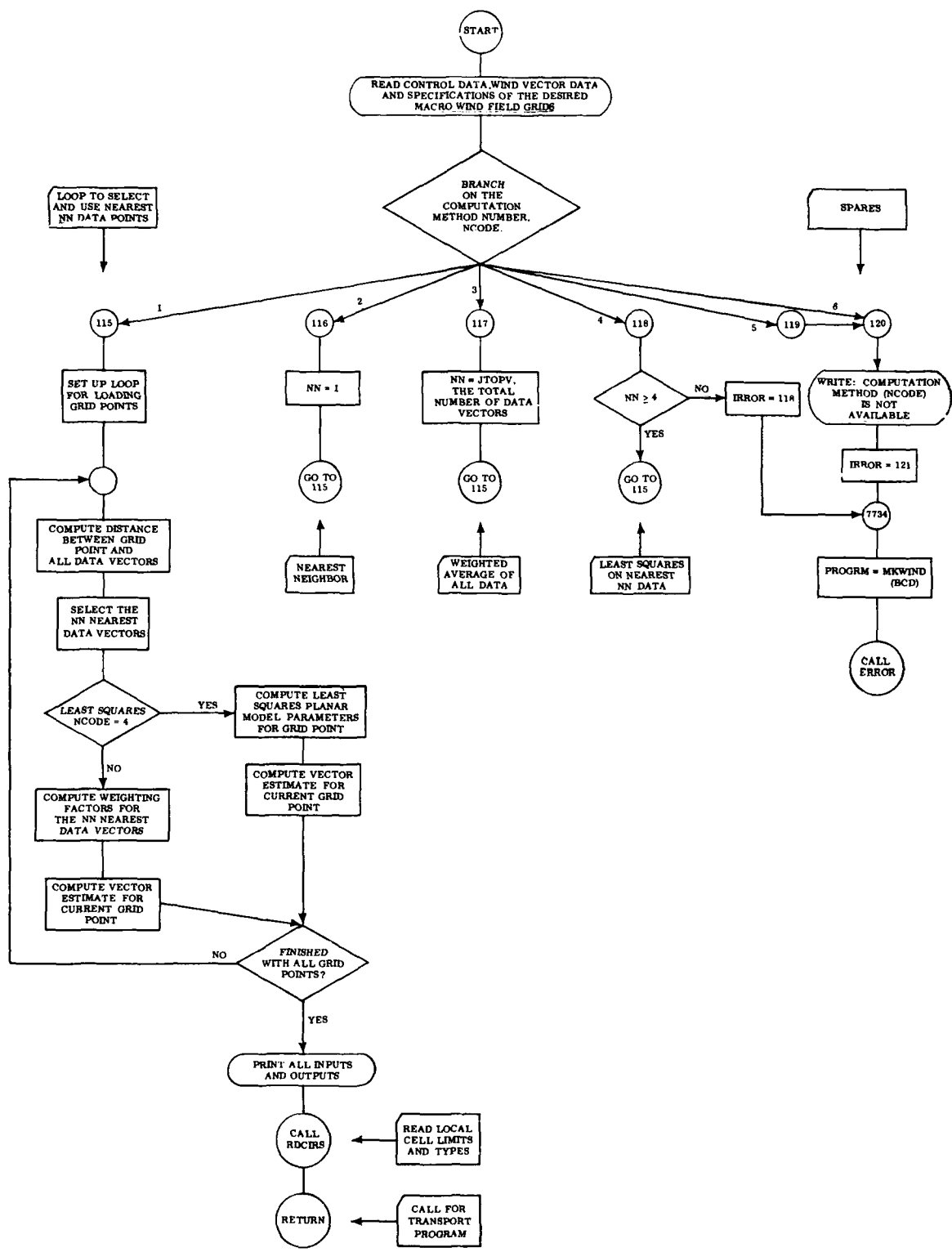
This subroutine forms and stores in core a horizontally and vertically variant wind description on the basis of inputs from the IBSYS input tape. Inputs are as follows:



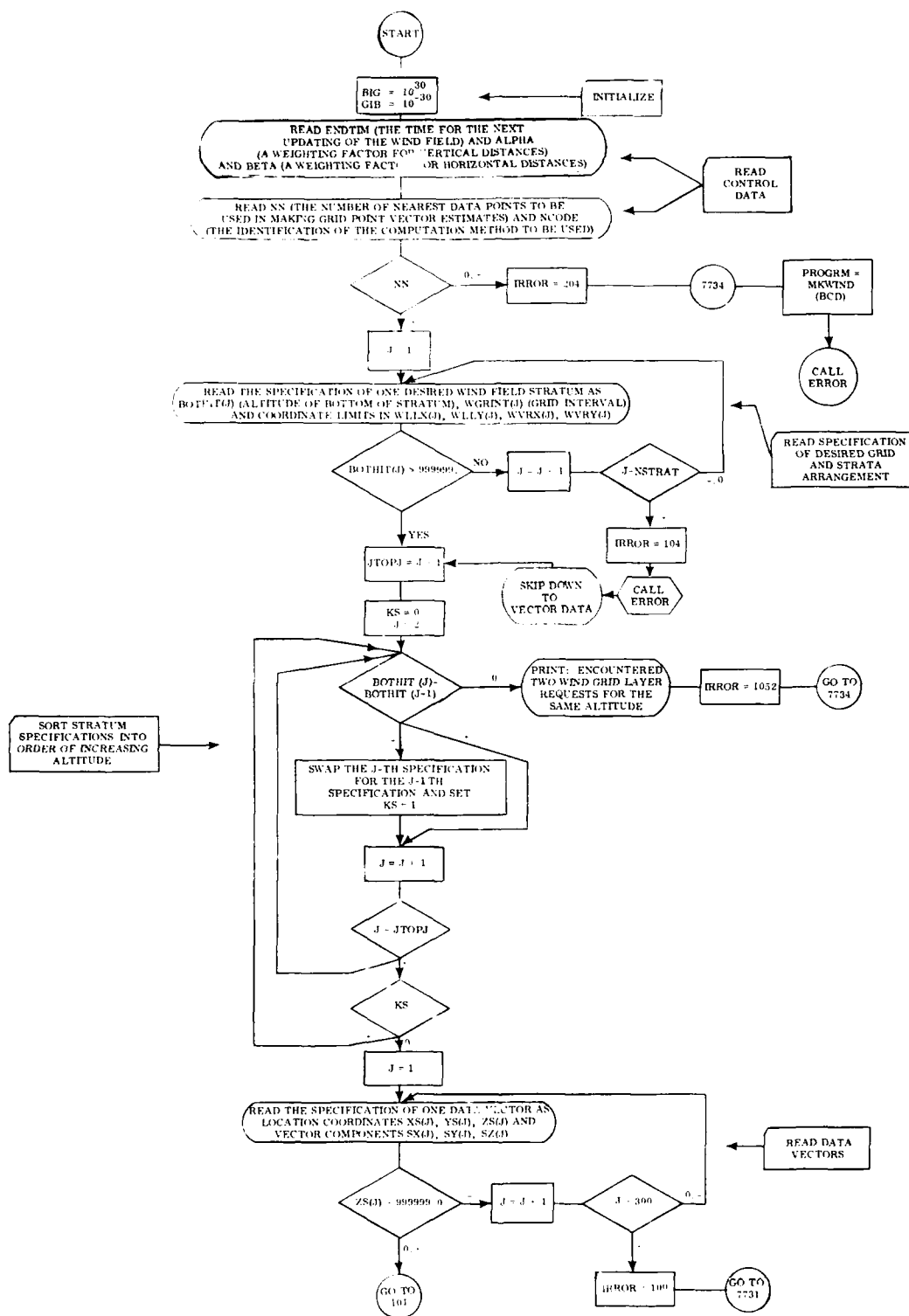
FC-6. Flow Chart for Subroutine RDCIRS

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.

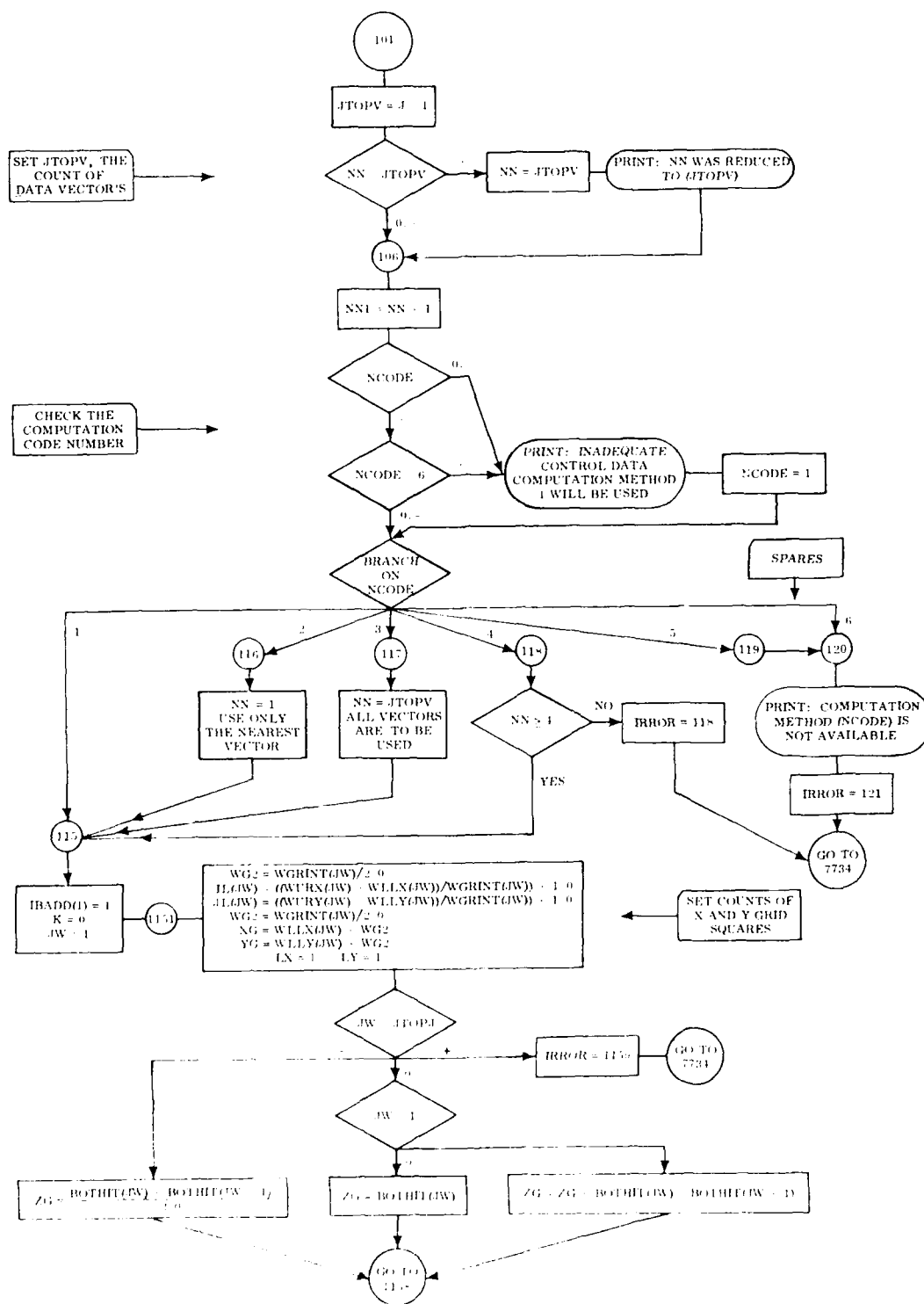


FC-7. Organizational Flow Chart for Subroutine MKWIND



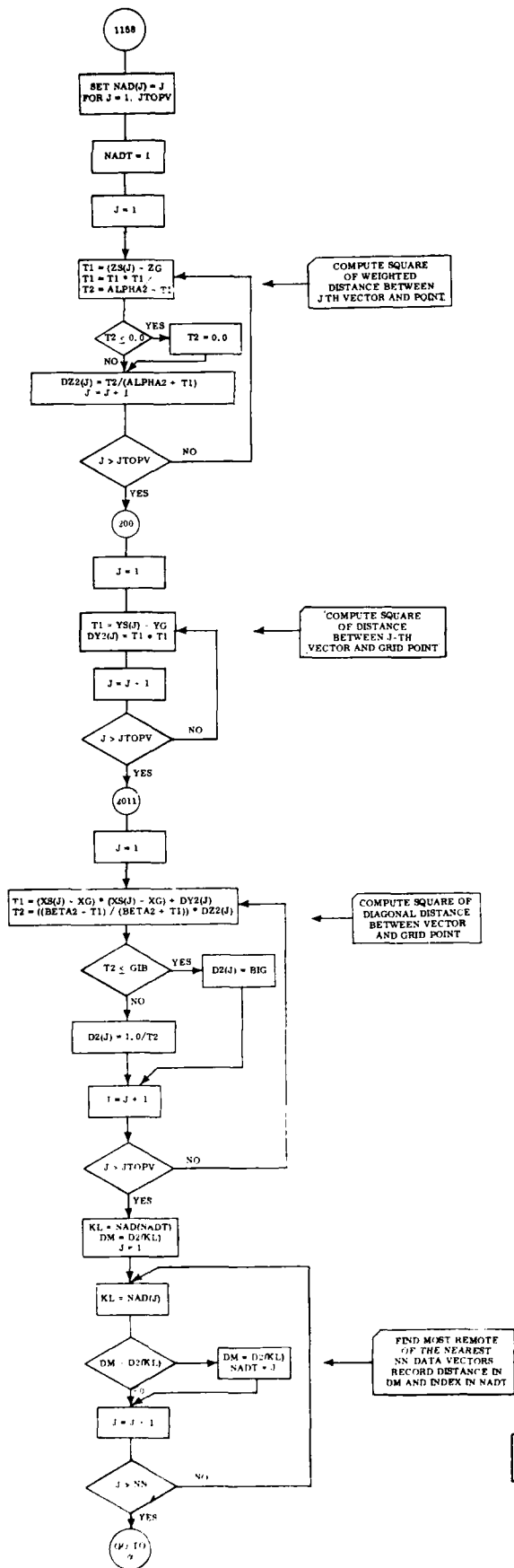
(a)

FC-8. Detailed Flow Charts for Subroutine MKWIND

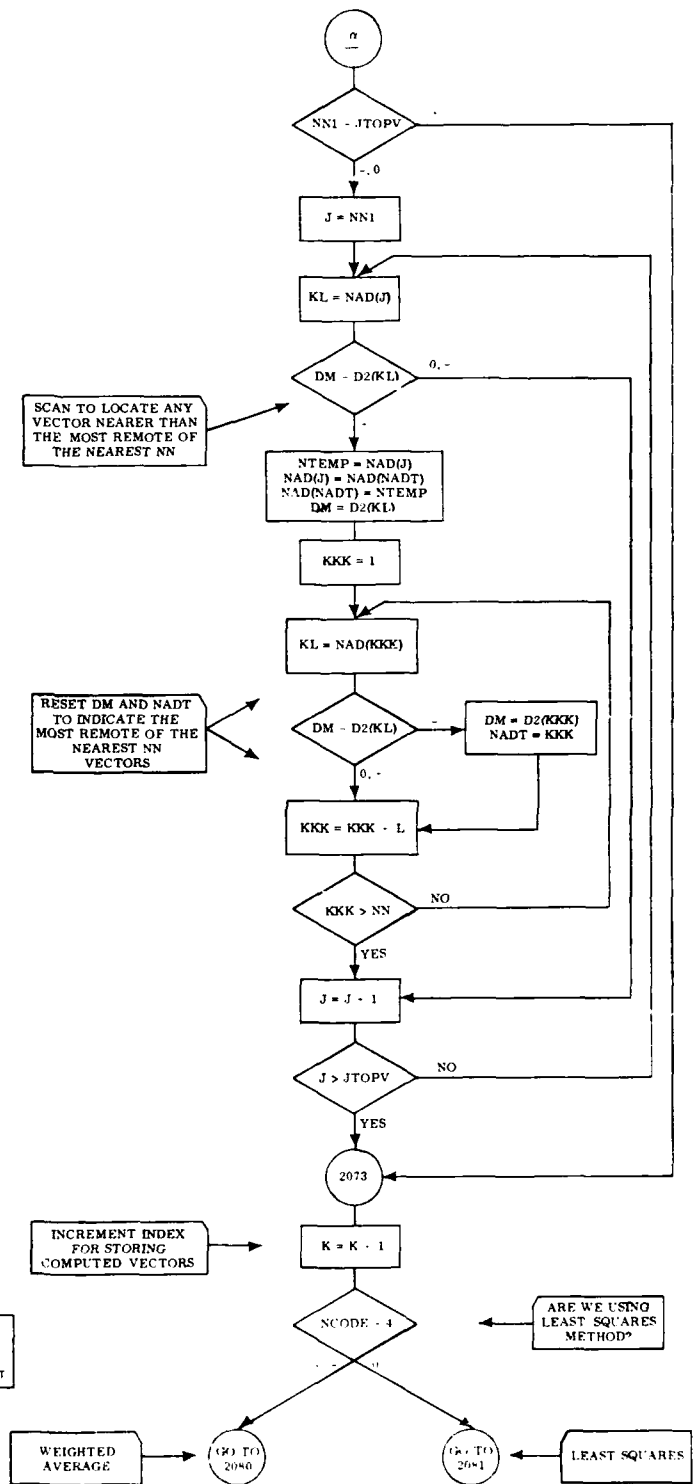


(b)

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

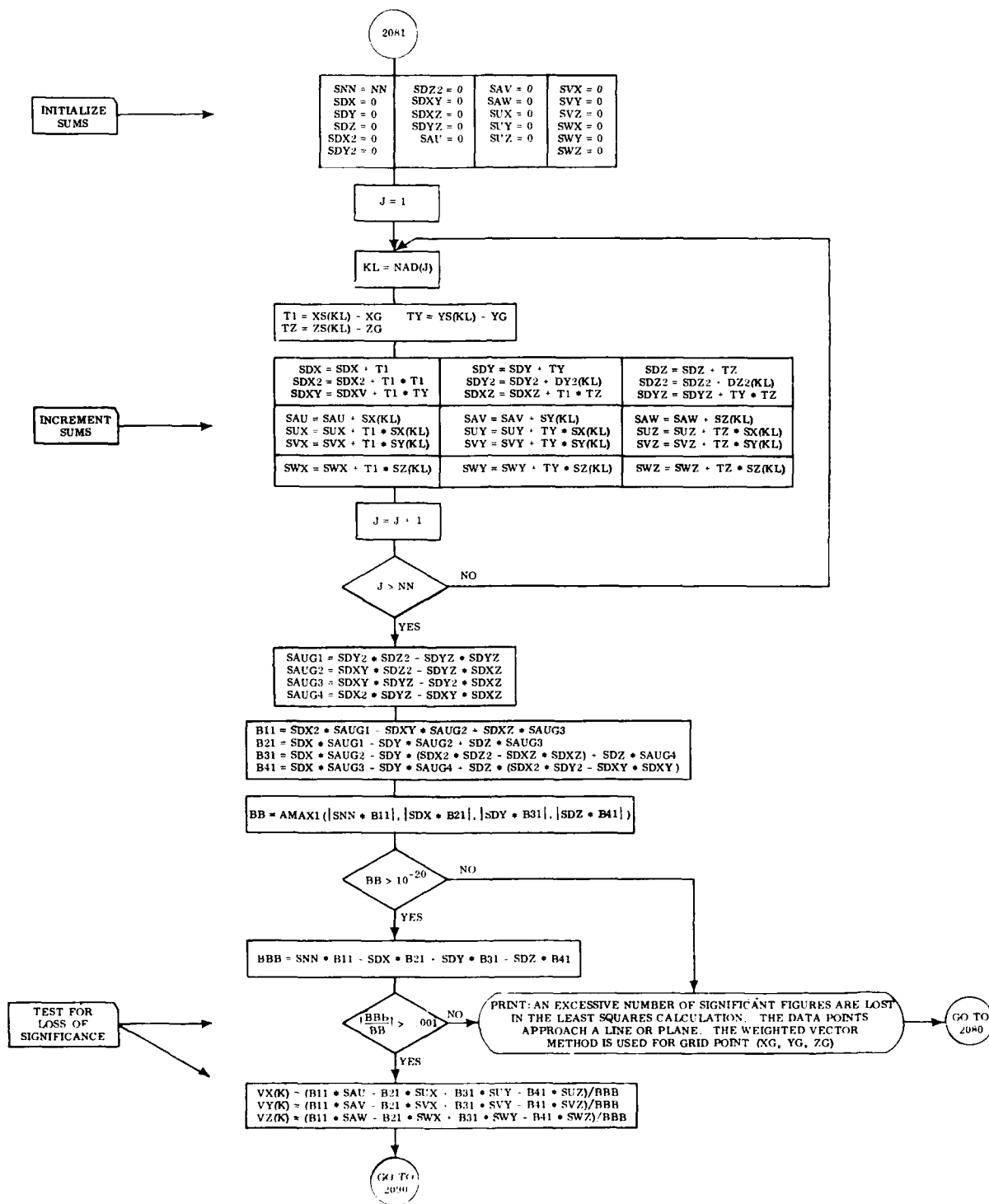


(c)



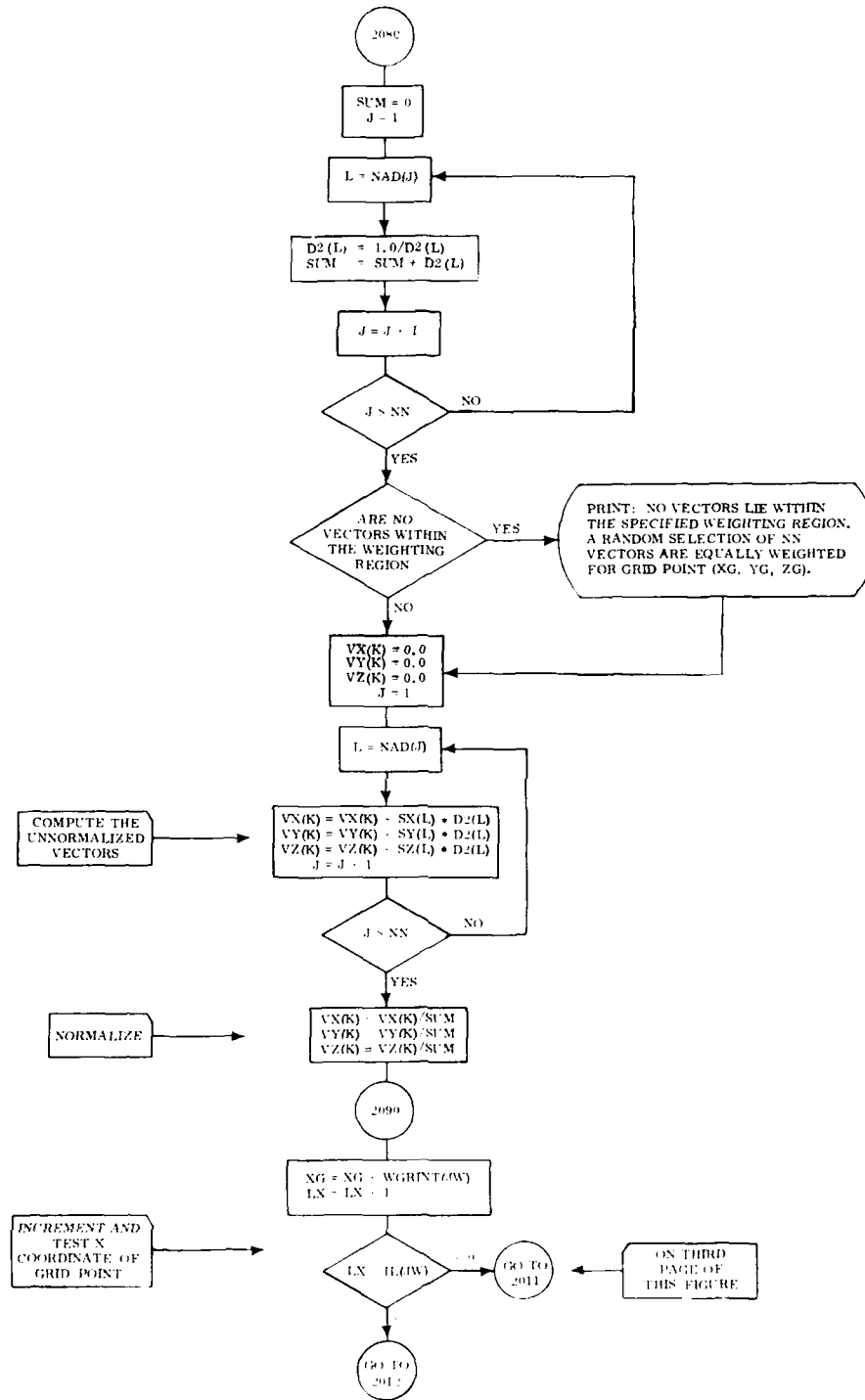
(d)

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



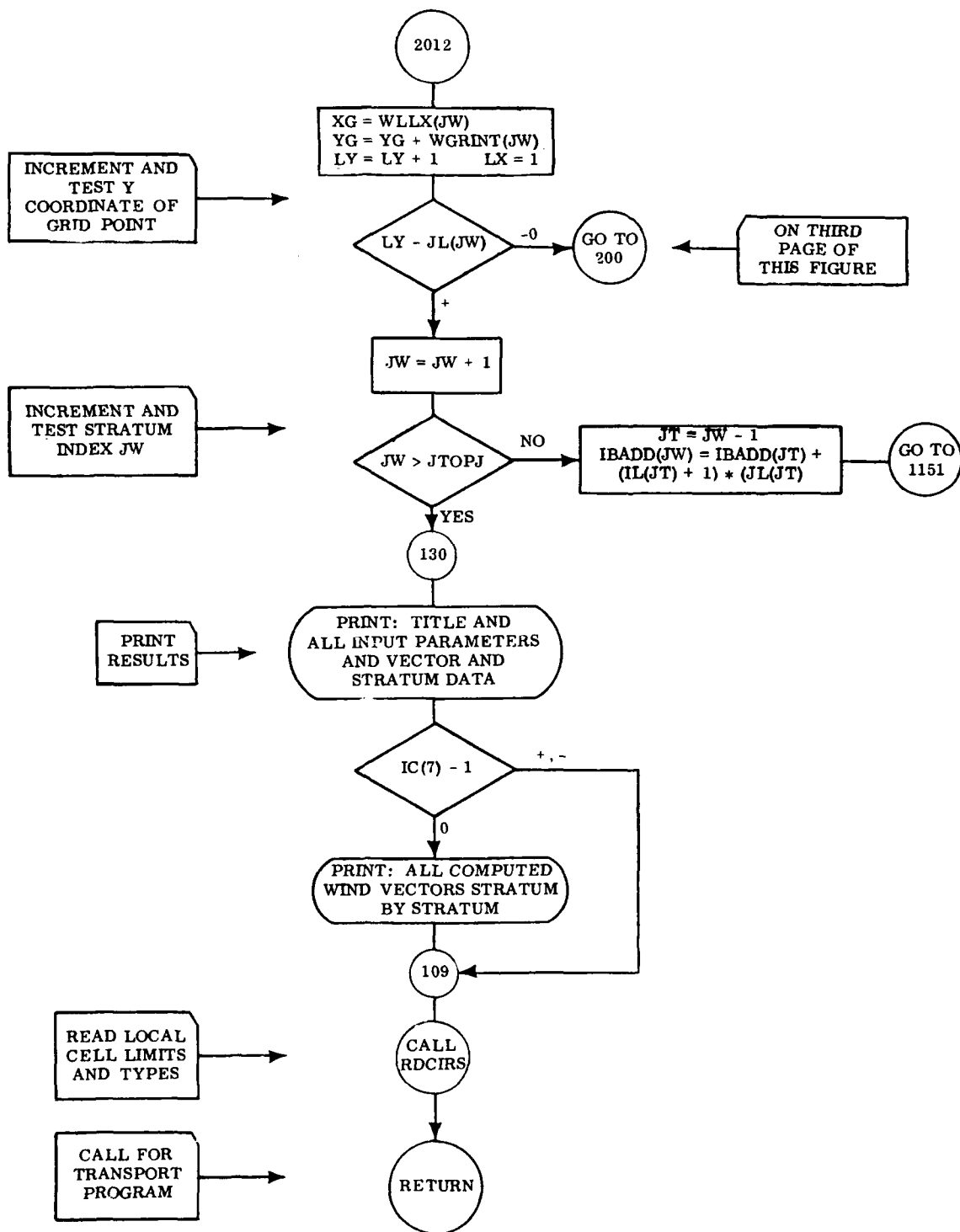
(e)

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 $0 < \text{NCODE} \leq 6$ 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 $\text{VX}(\text{J})$, $\text{VY}(\text{J})$, and $\text{VZ}(\text{J})$. The storage index of the first entry in the wind-field description arrays for the first stratum is set at 1 (i. e. , $\text{IBADD}(1) = 1$), 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 $\text{IL}(\text{JW})$ and $\text{JL}(\text{JW})$, 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 $\text{NAD}(\text{J}) = \text{J}$, $\text{J} = 1, \text{JTOPV}$, 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 $\left| \frac{BBB}{BB} \right|$ is found. If BB is less than 10^{-20} or the ratio $\left| \frac{BBB}{BB} \right|$ 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 ($JTOP1 \neq 0$), 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 $0 < TP(J) < TLIMIT$; 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

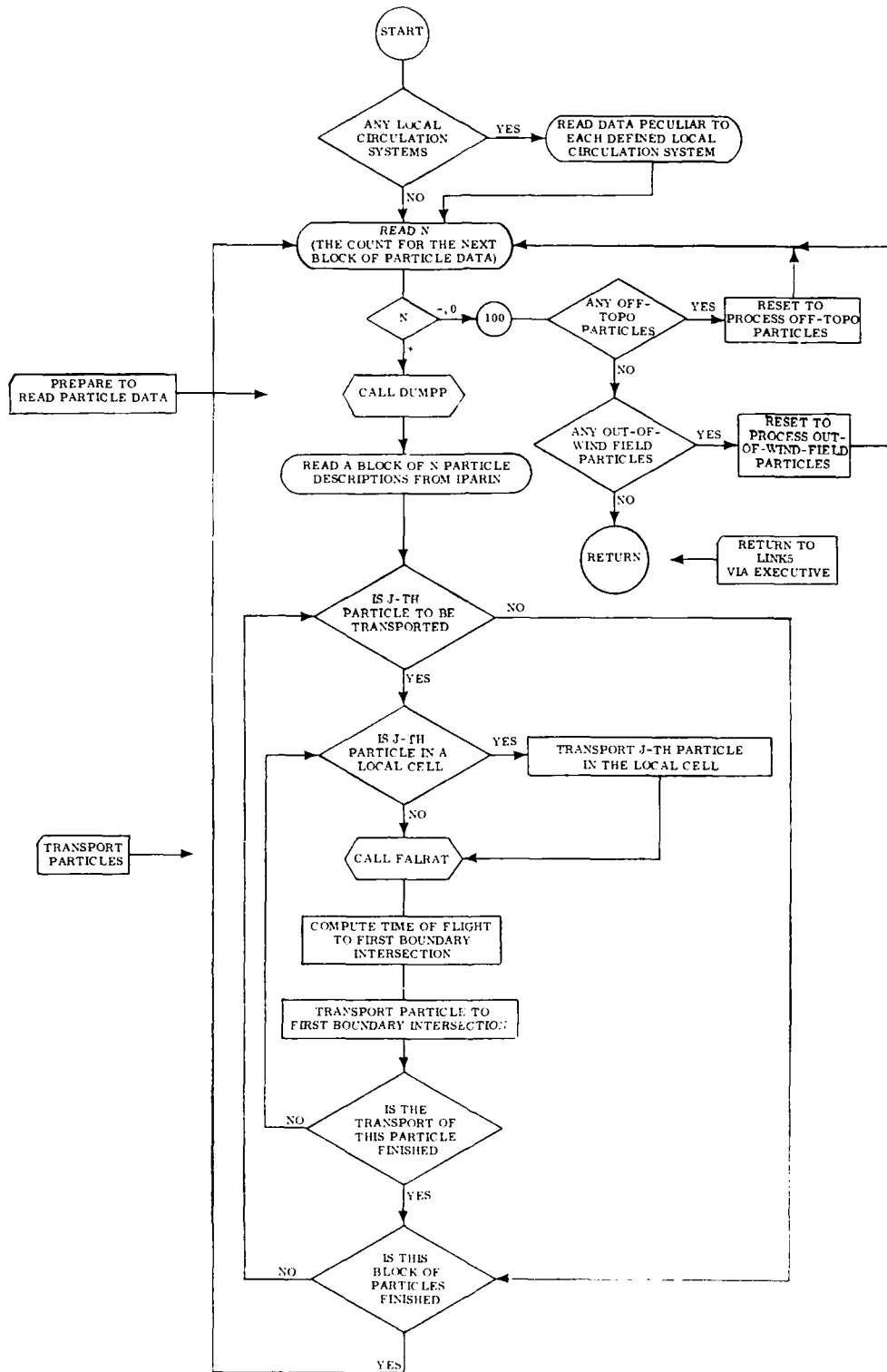
never moves any particle descriptions within the particle arrays. It does, however, mark the status of particles within the arrays using the sign of parameter FMAS(J) and the sign and value of TP(J) in accordance with the conventions described in Table 2.

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

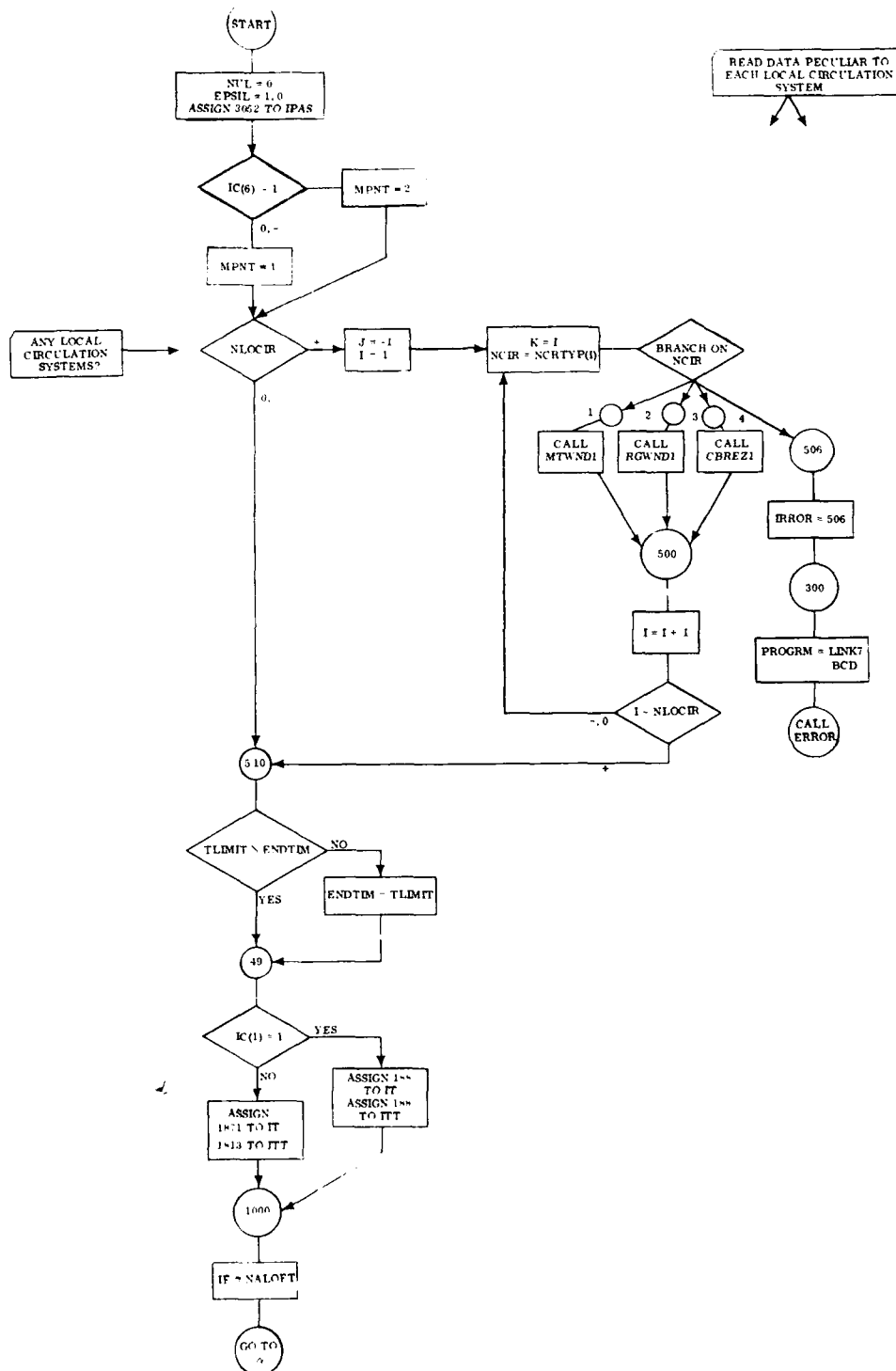
The purpose of this program is to determine the index to be used for retrieving the macrowind vector that applies at a particle position point XX, YY, ZZ. The desired index is stored in the argument JWAD upon return. JWAD is set negative in the event that the point XX, YY, ZZ is outside the volume for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

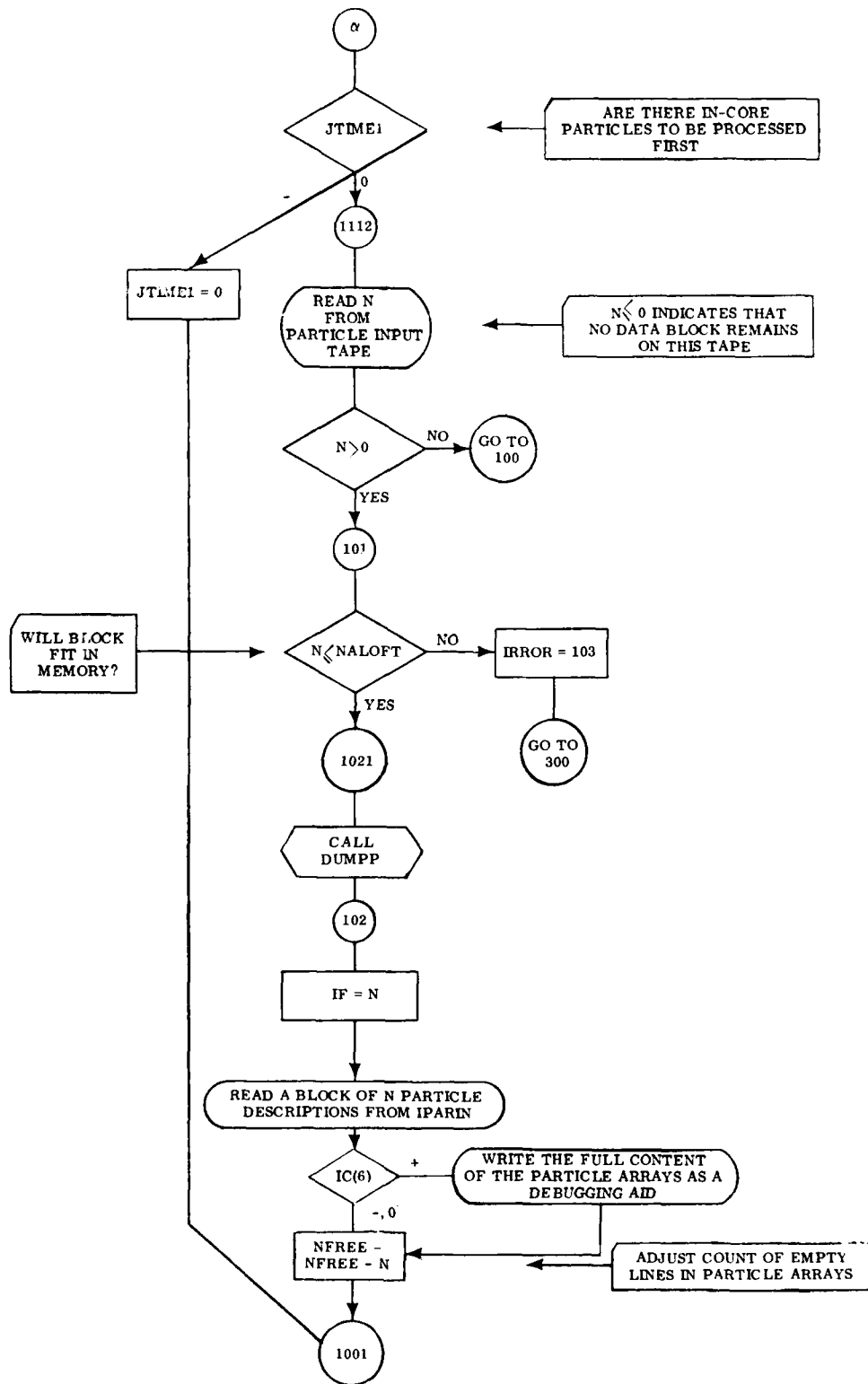


FC-9. General Flow Chart for Subroutine LINK7



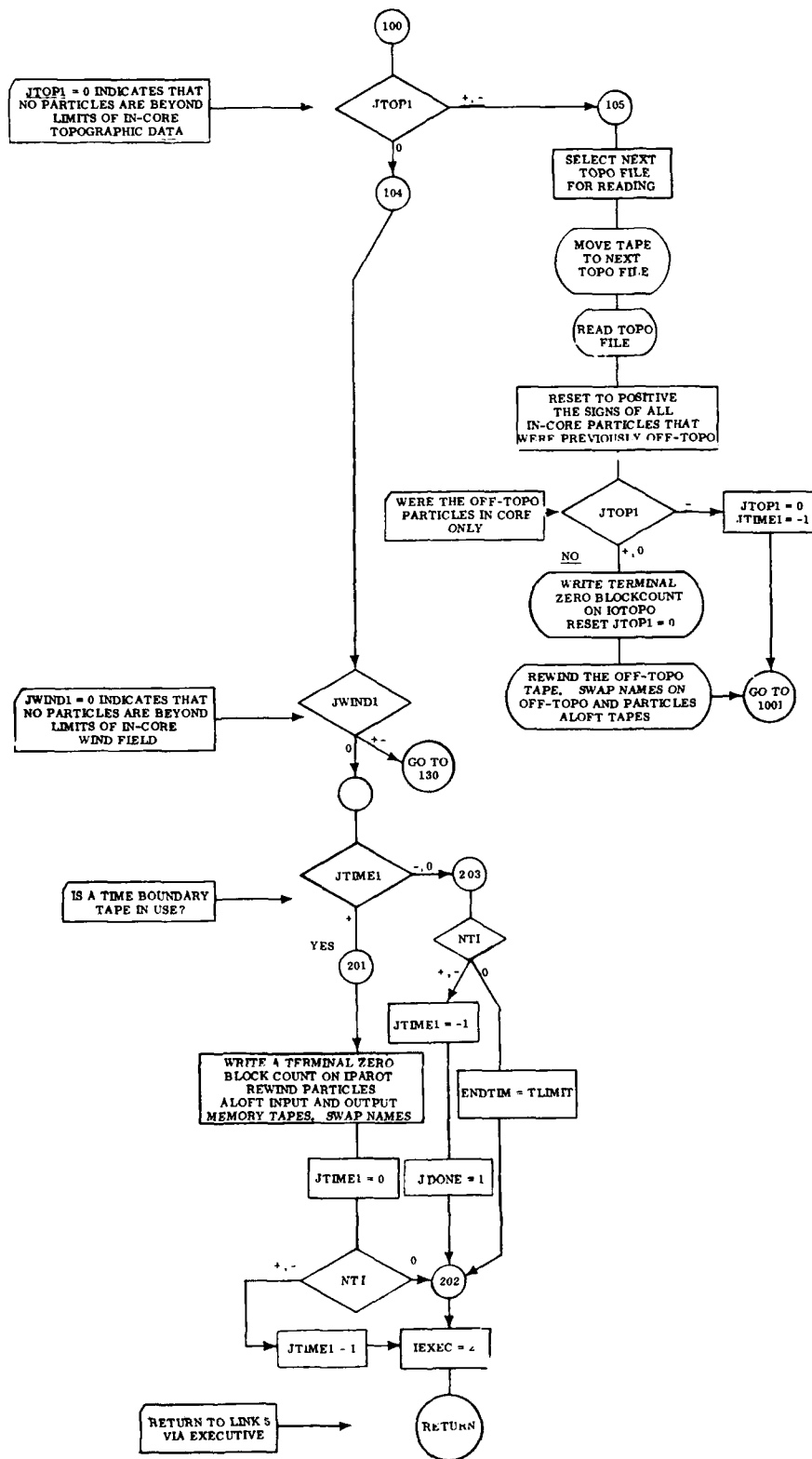
(a)

FC-10. Detailed Flow Charts for Subroutine LINK7



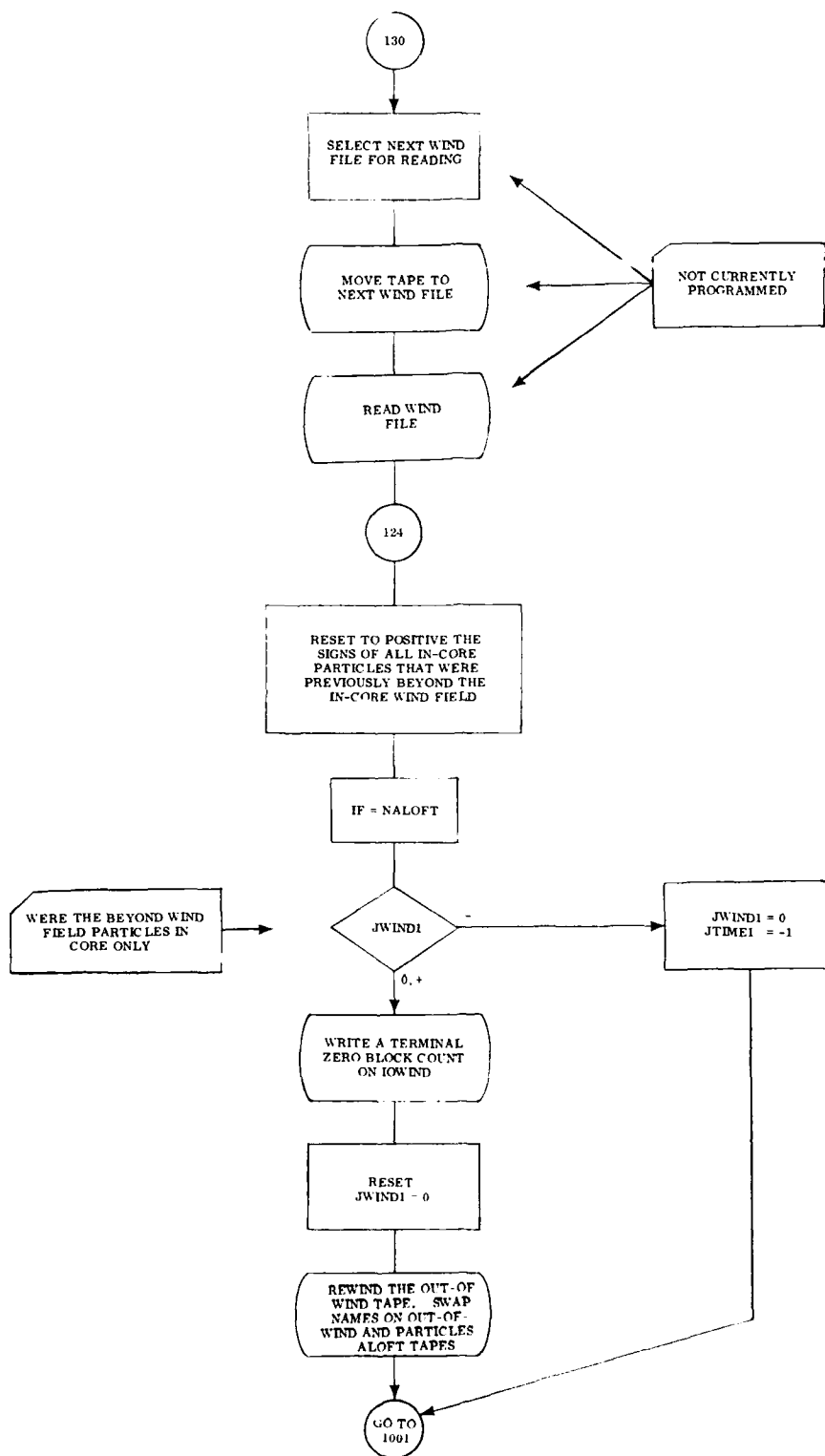
(b)

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



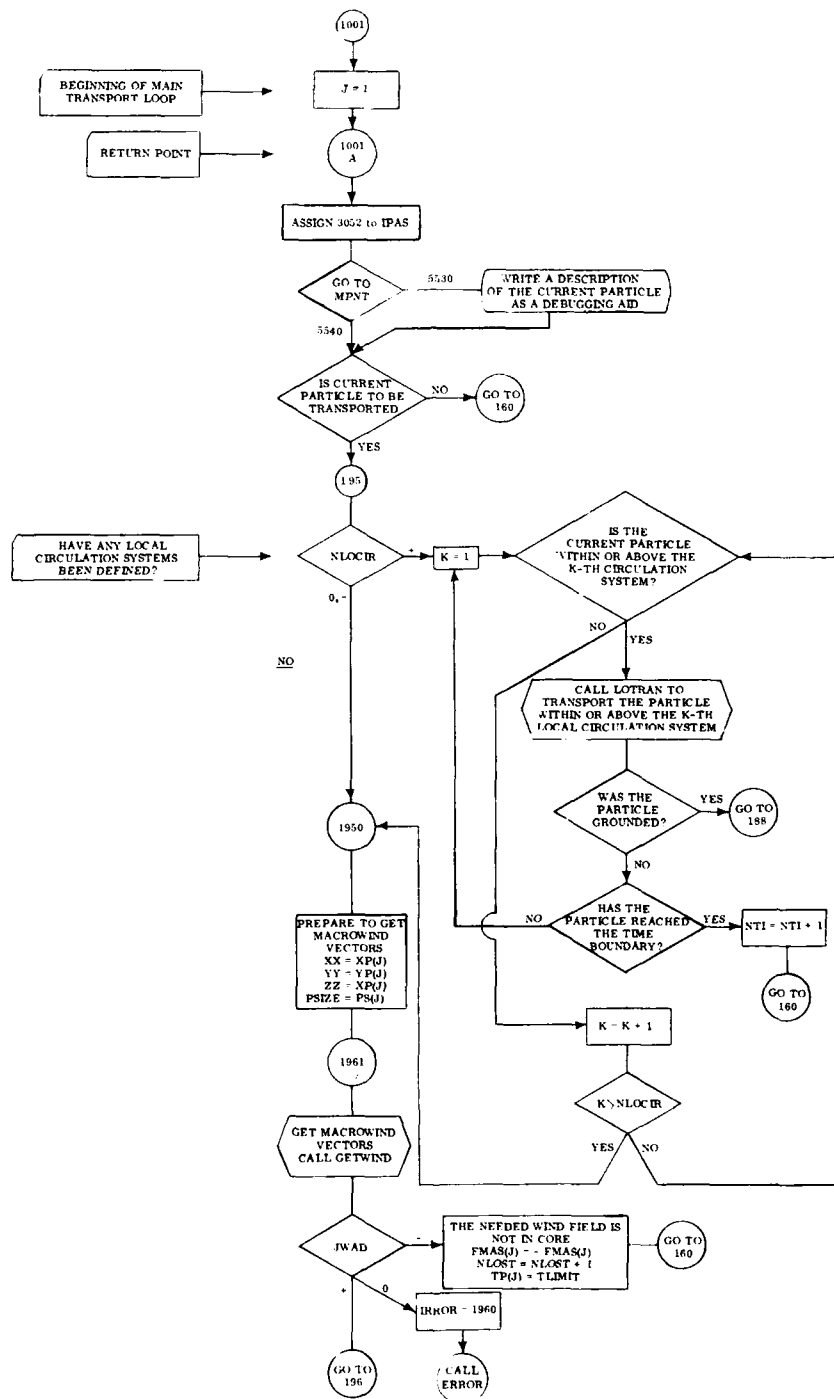
(c)

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



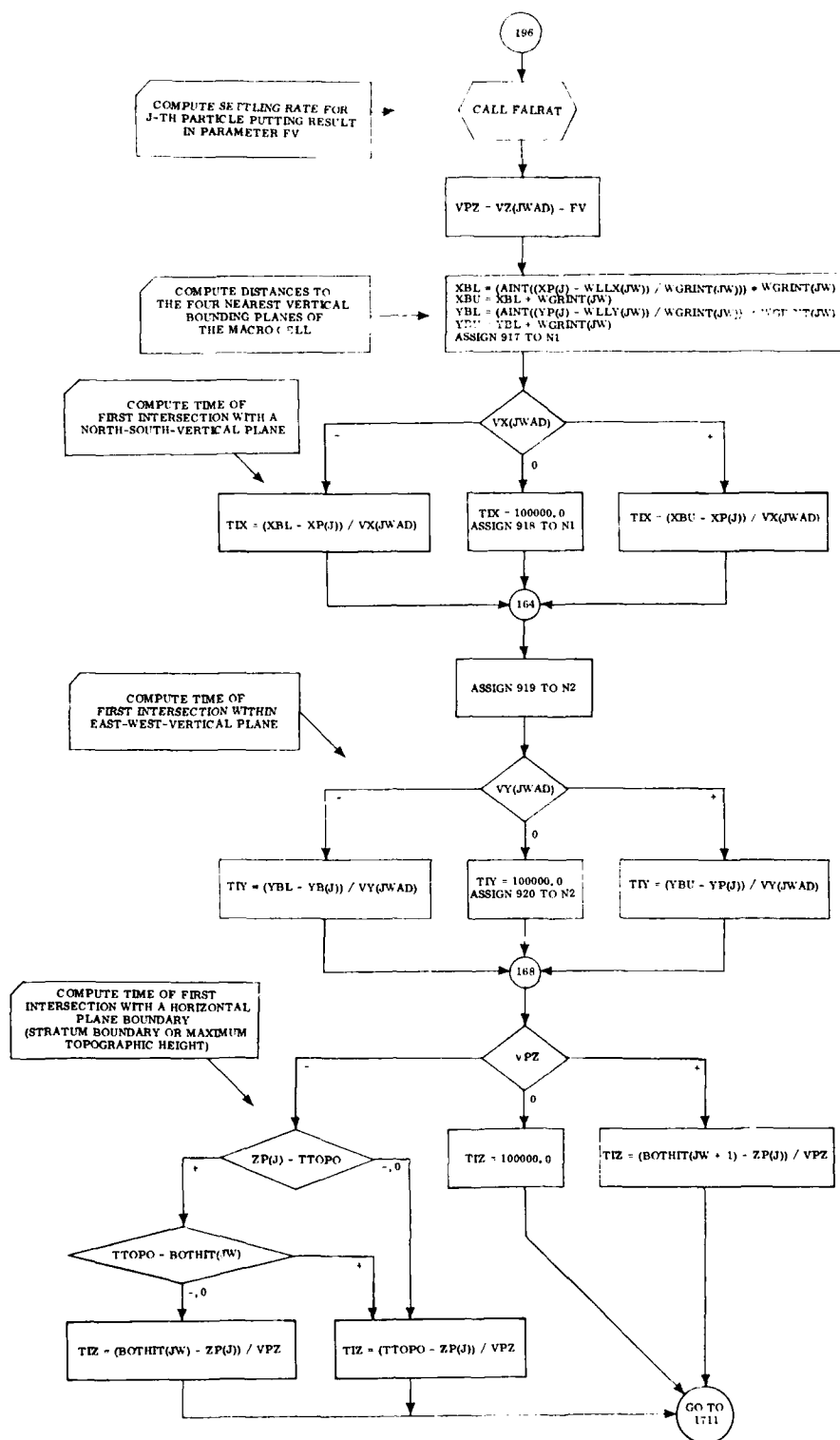
(d)

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



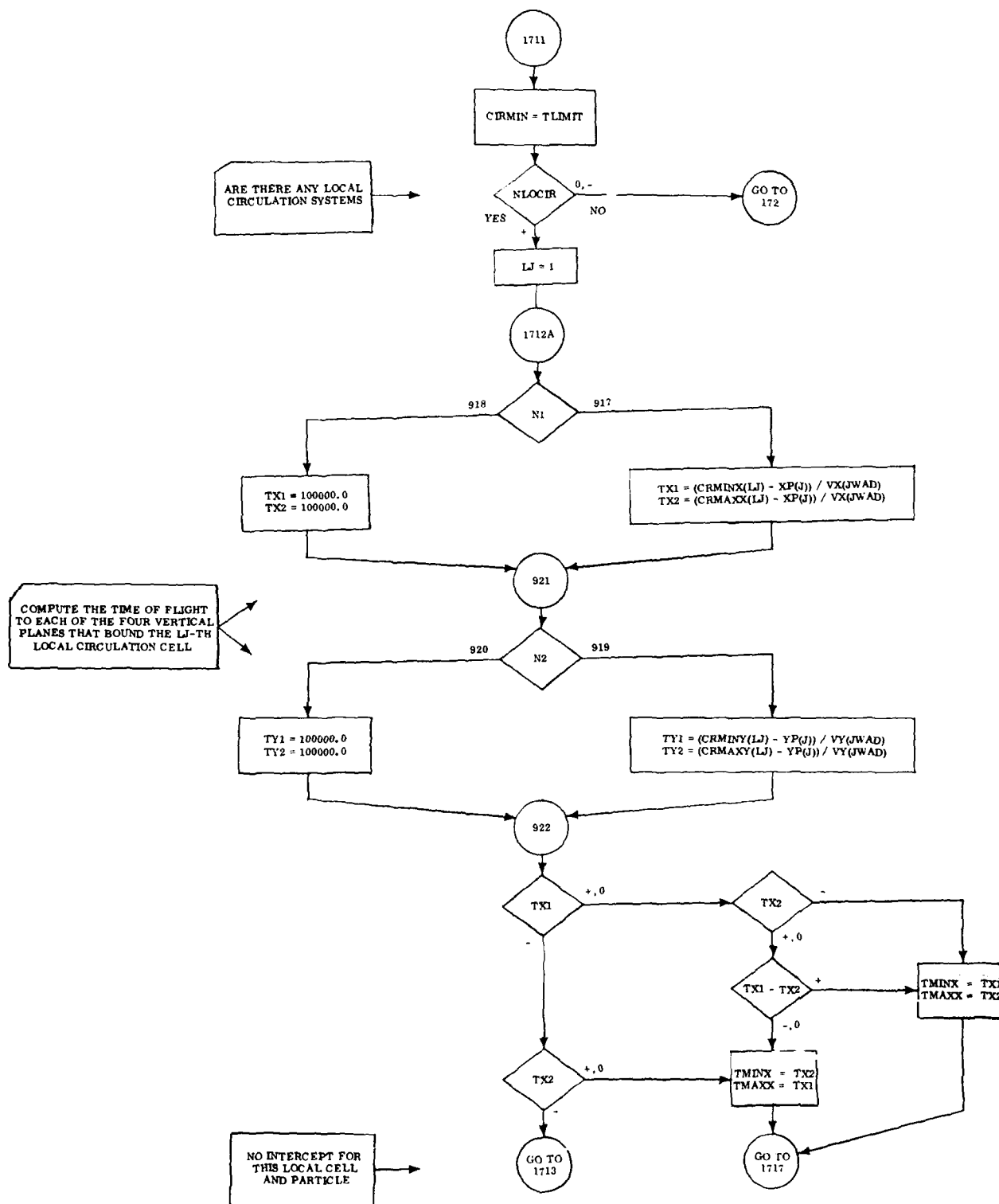
(e)

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



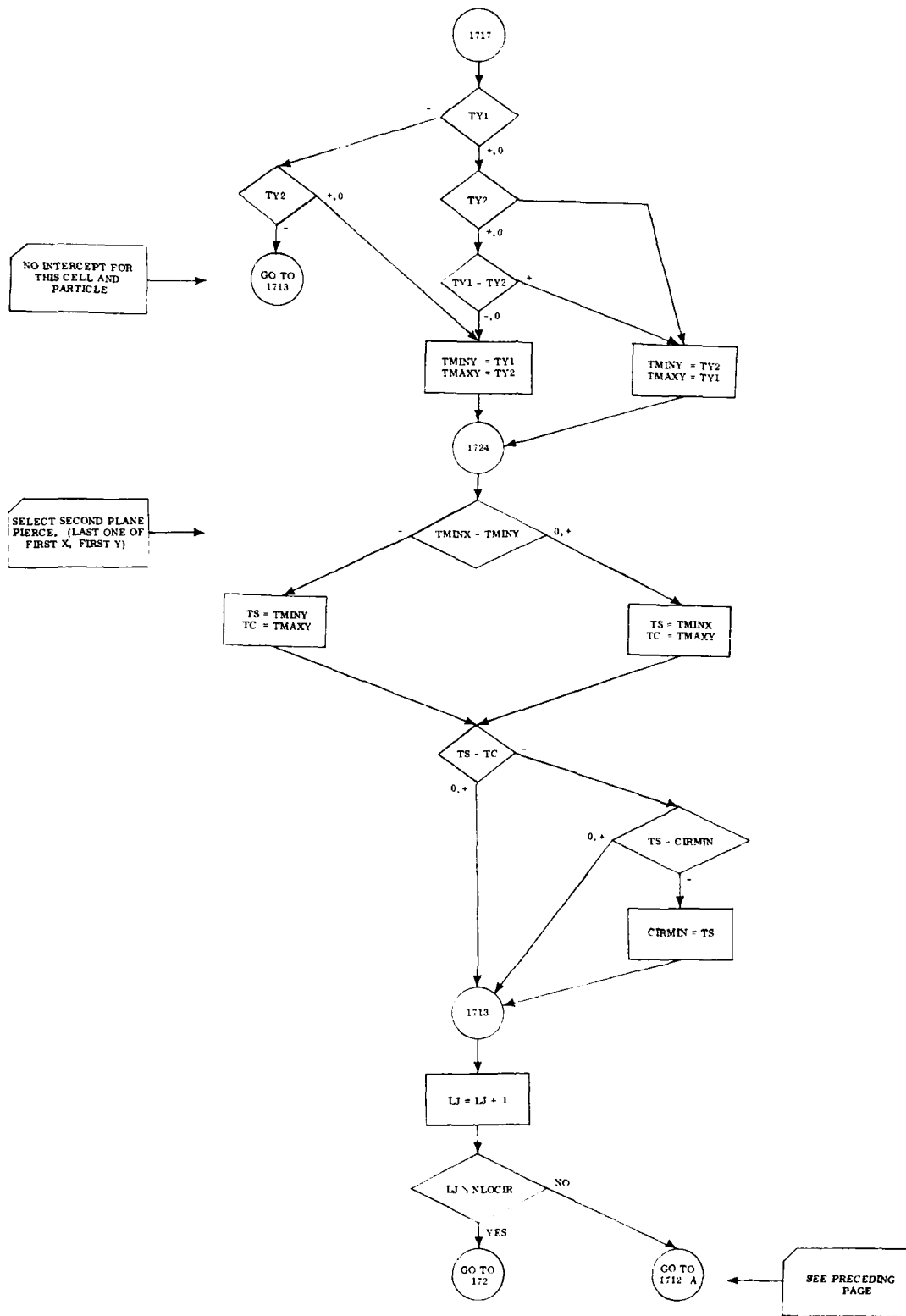
(f)

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



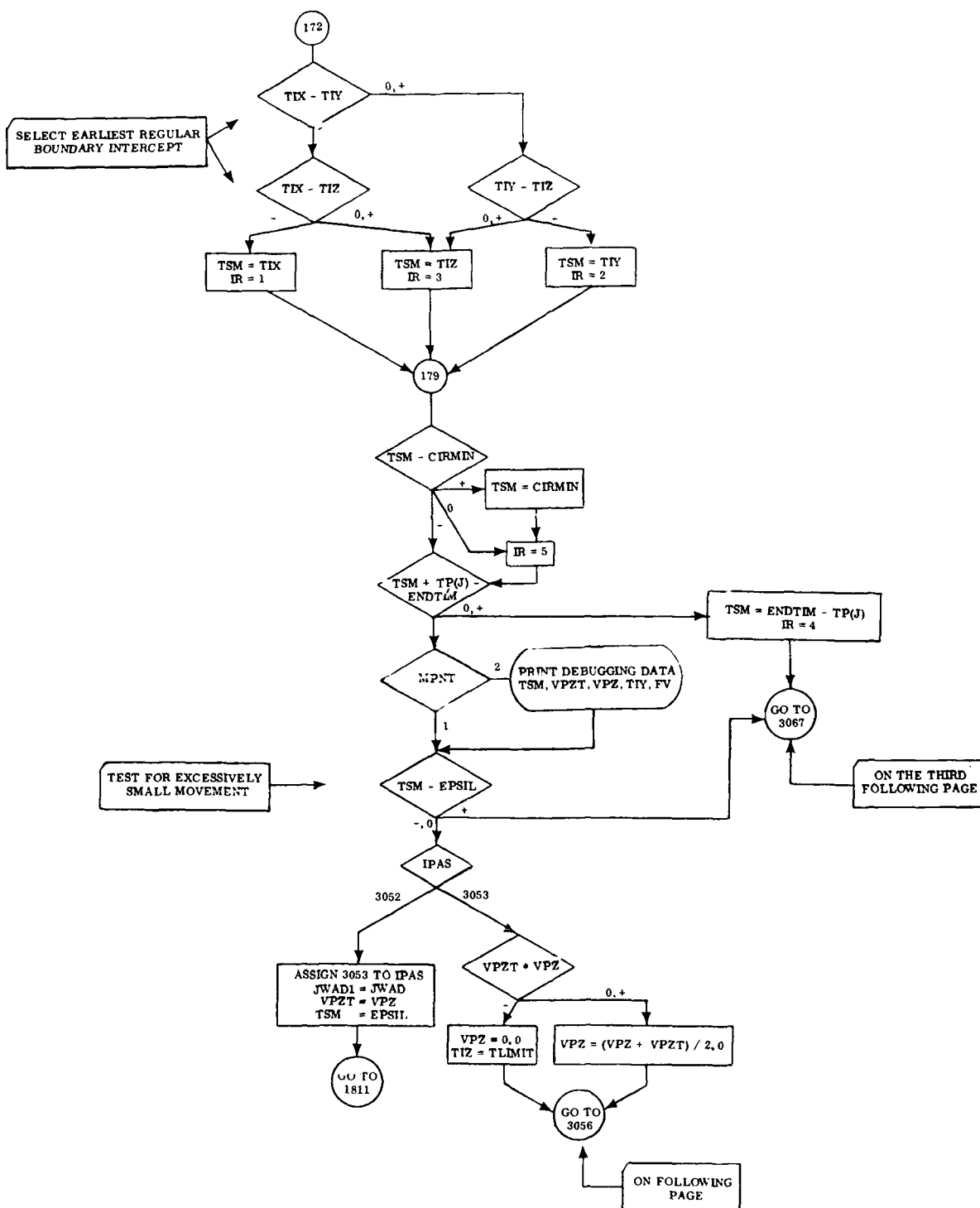
(g)

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



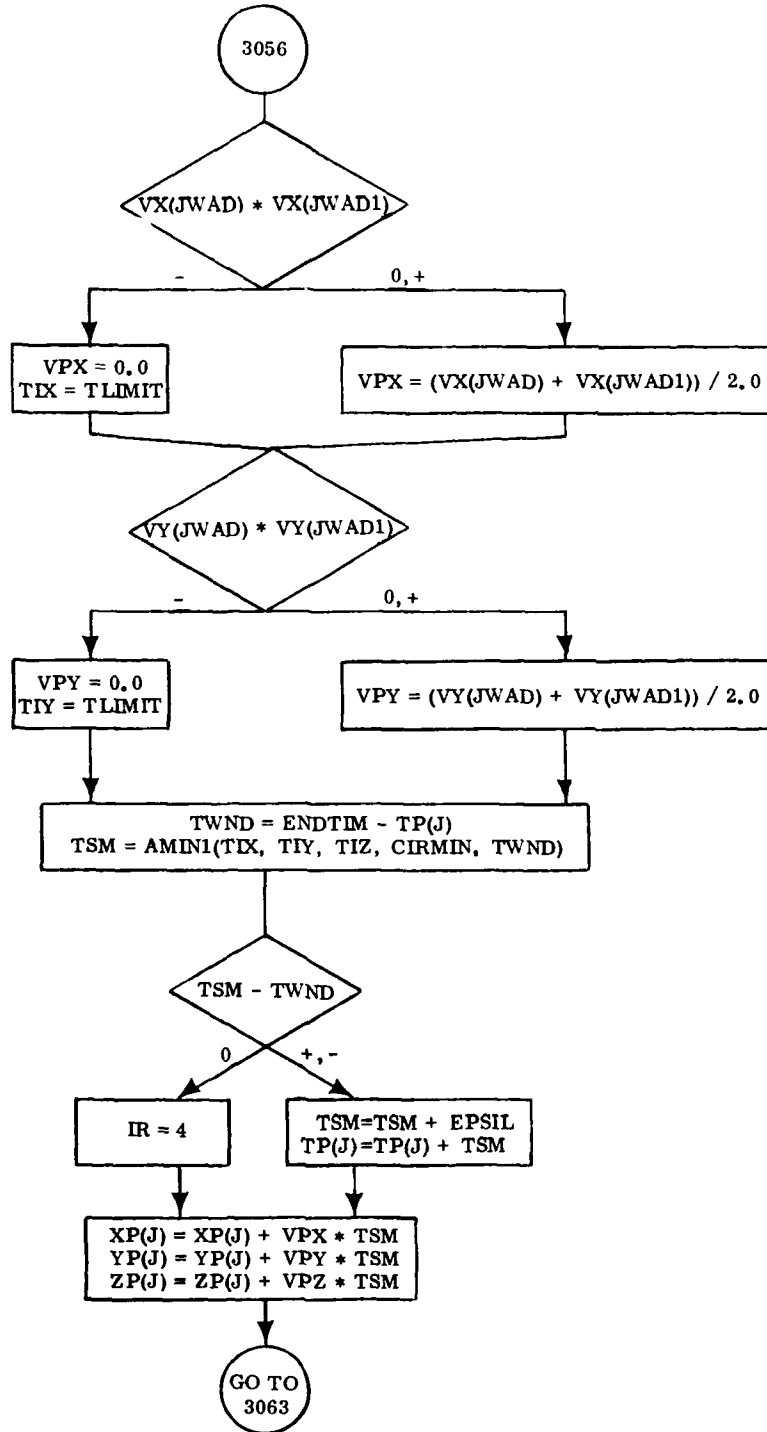
(h)

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



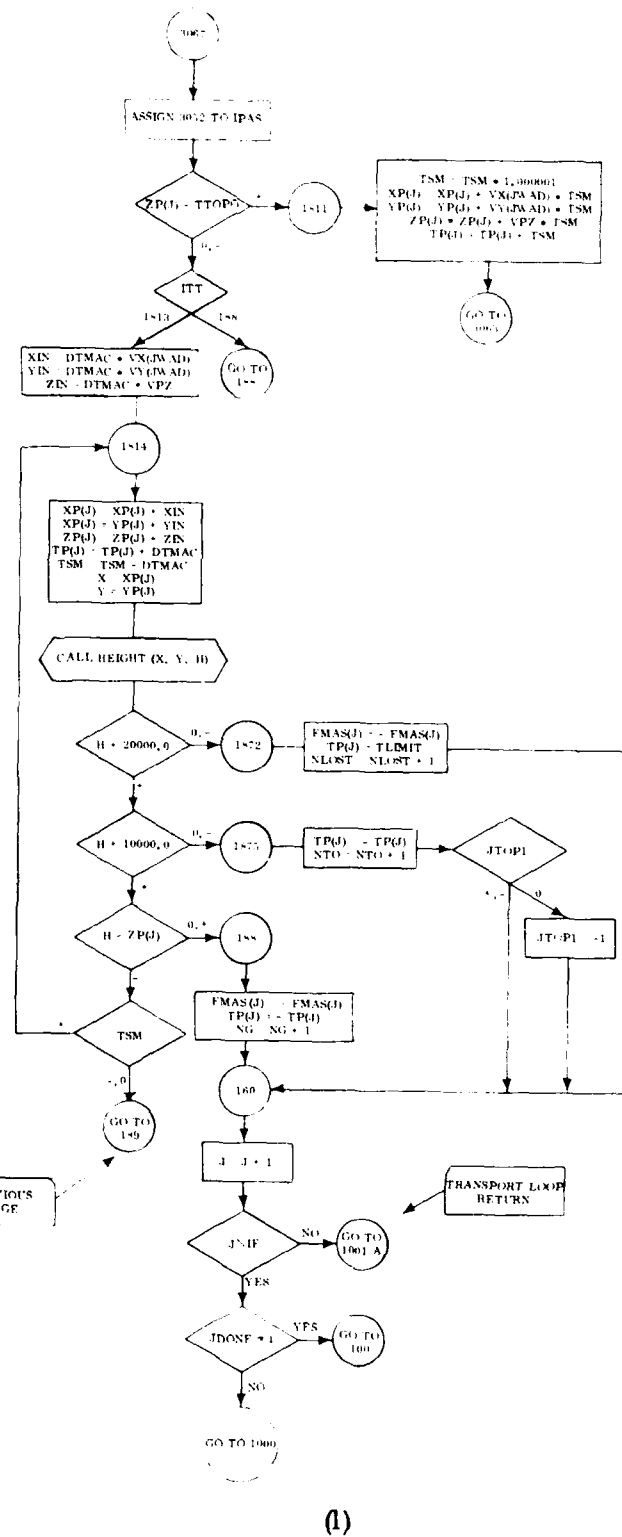
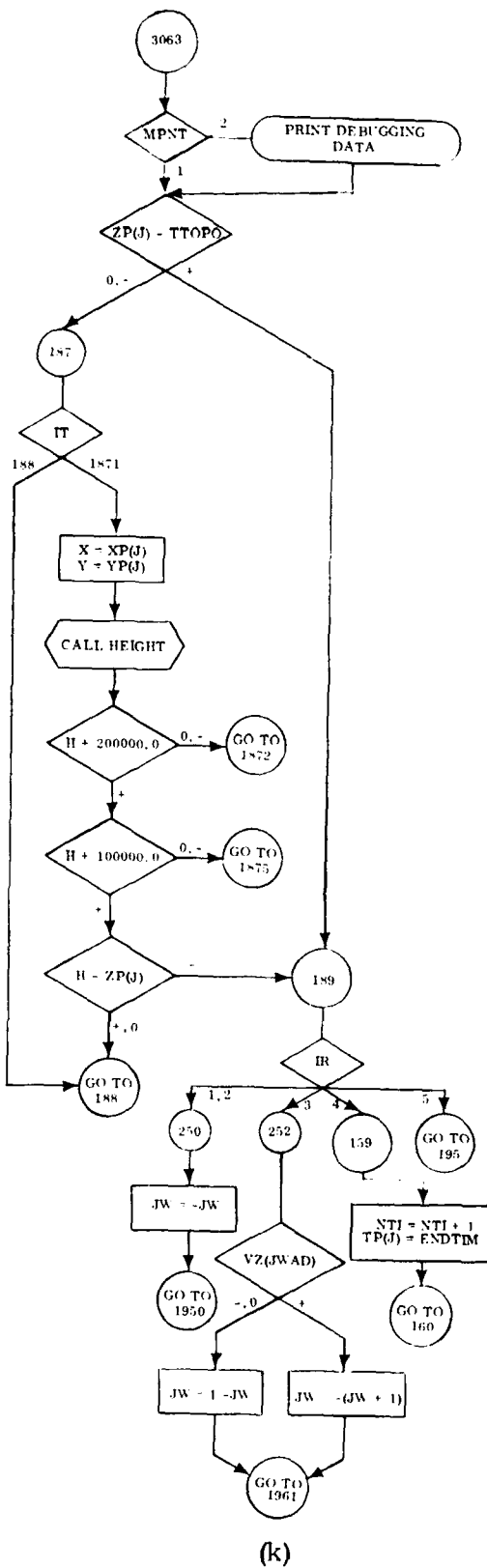
(i)

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

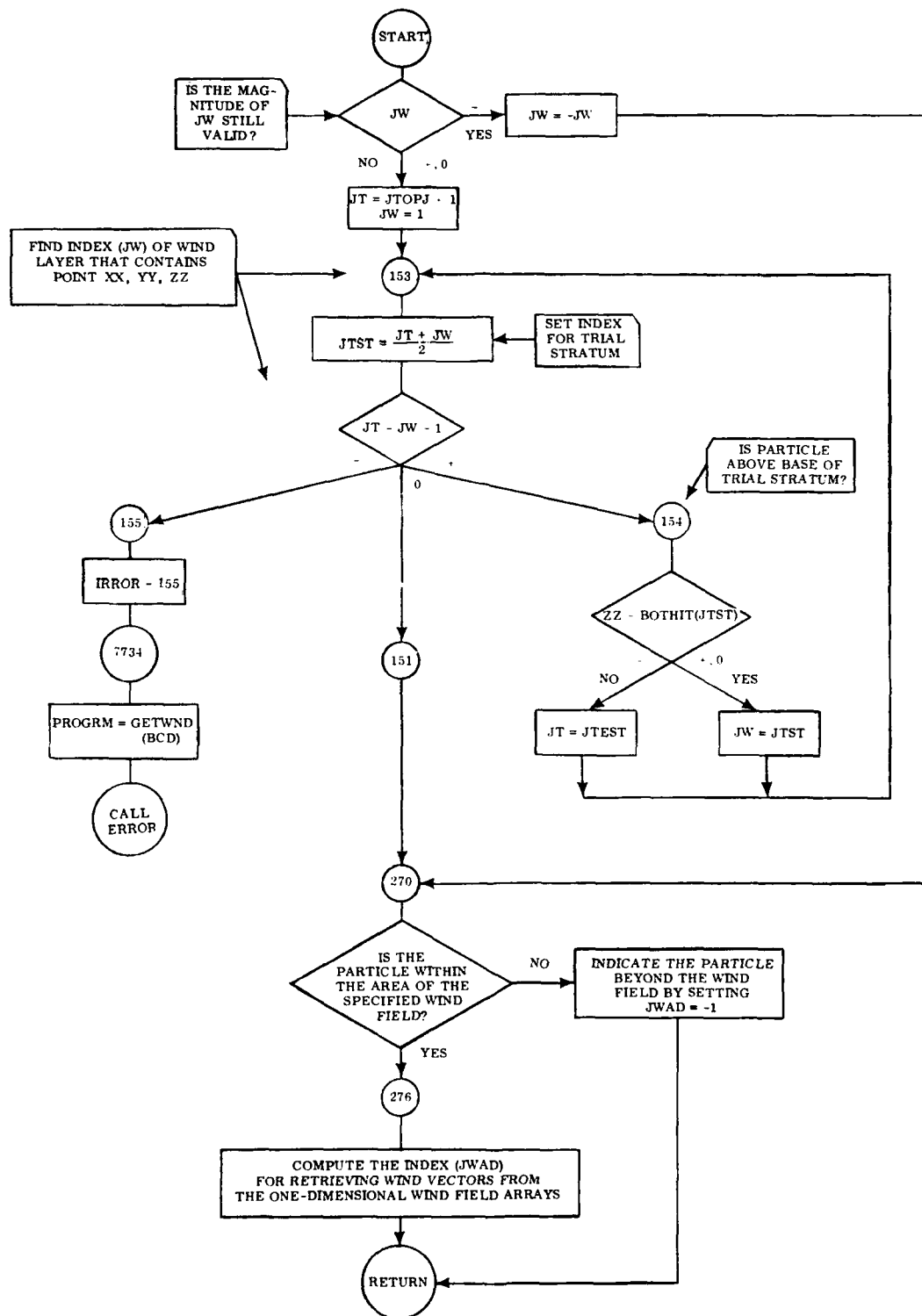
Next, at statement number 270 a check is made to see that the particle is within the bounds of the macrowind field. If it is not, JWAD is set -1 to indicate the problem to the calling program and GETWND returns. If the particle is in a satisfactory position, JWAD is computed to locate the desired vector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

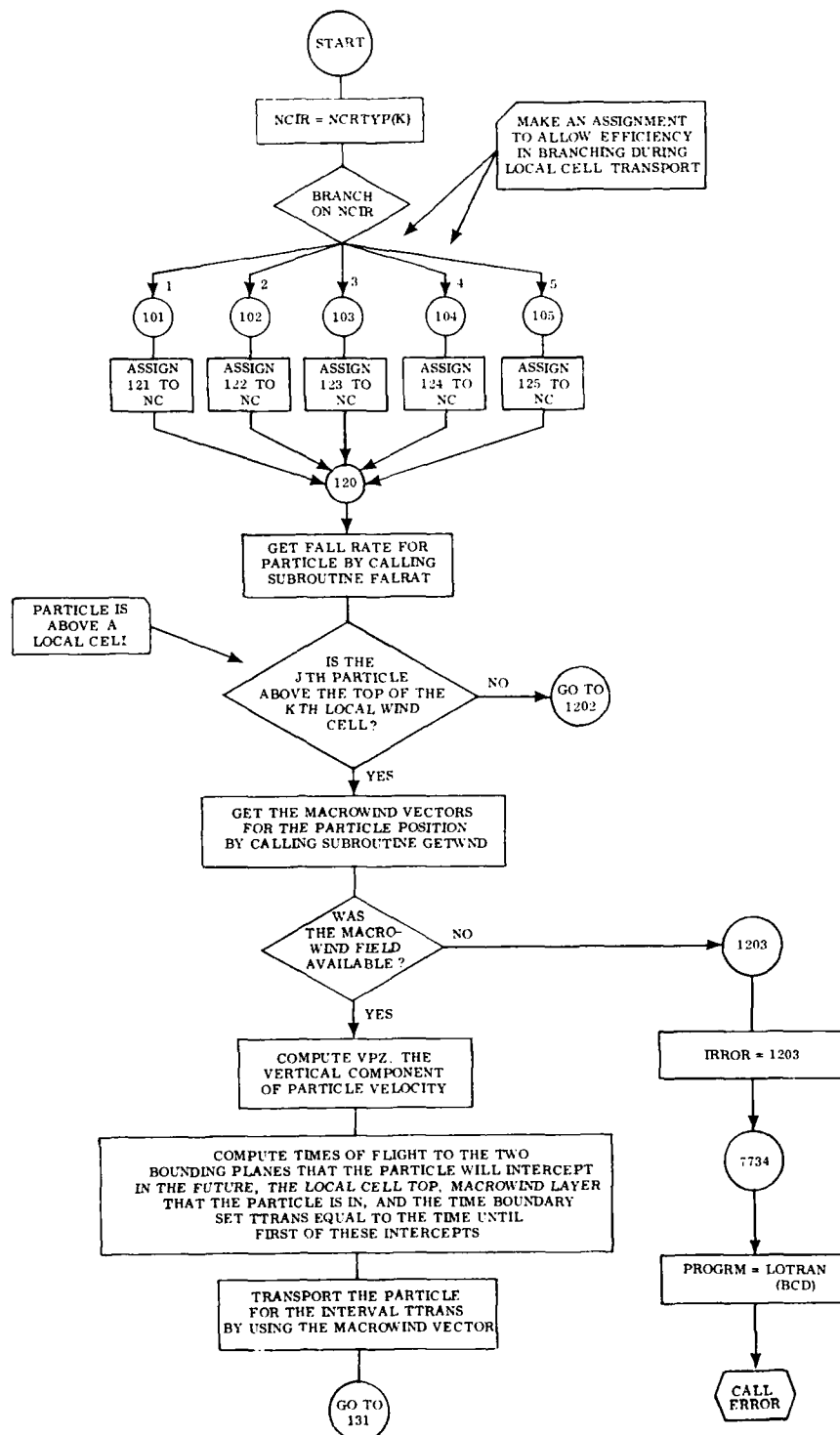
At 131 the program determines whether the particle is still in or above the local cell. If it is not, the program is finished and returns. If the particle is still in or above the local cell, the time boundary is checked and if not violated a return is made to the top of the loop at statement 120.

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)

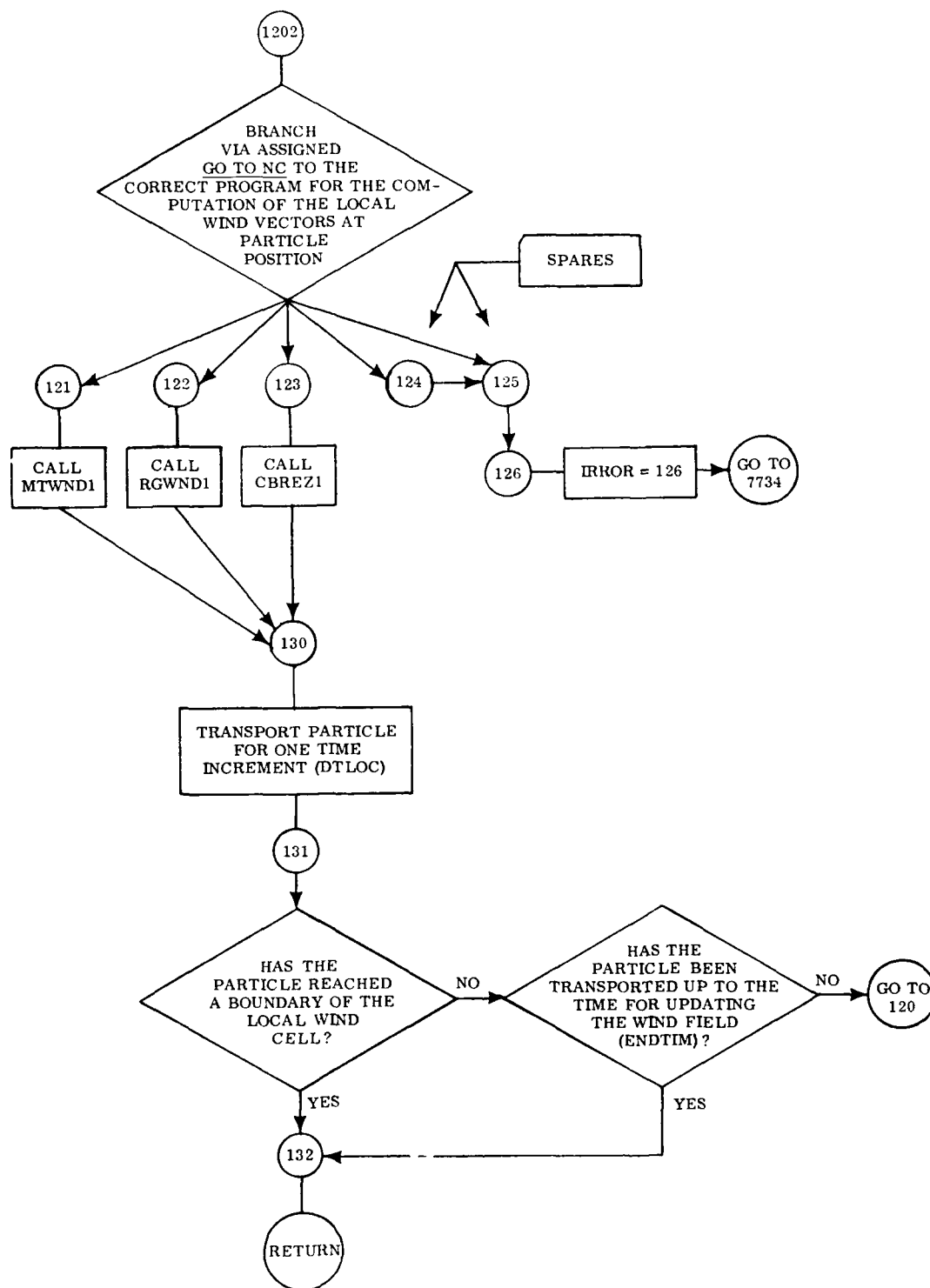
This subroutine, used for the Kth local cell, consists of two logical routes that serve the purpose of (1) reading mountain wind data and (2) computing the mountain wind components for the Jth particle at location XP(J), YP(J), ZP(J) after first checking for impact on the ground. If impact is sensed, 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 (CRMXY(K)), south (CRMINY(K)), east (CRMXX(K)), and west (CRMINX(K)).



(a)

FC-12. Flow Charts for Subroutine LOTRAN



(b)

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

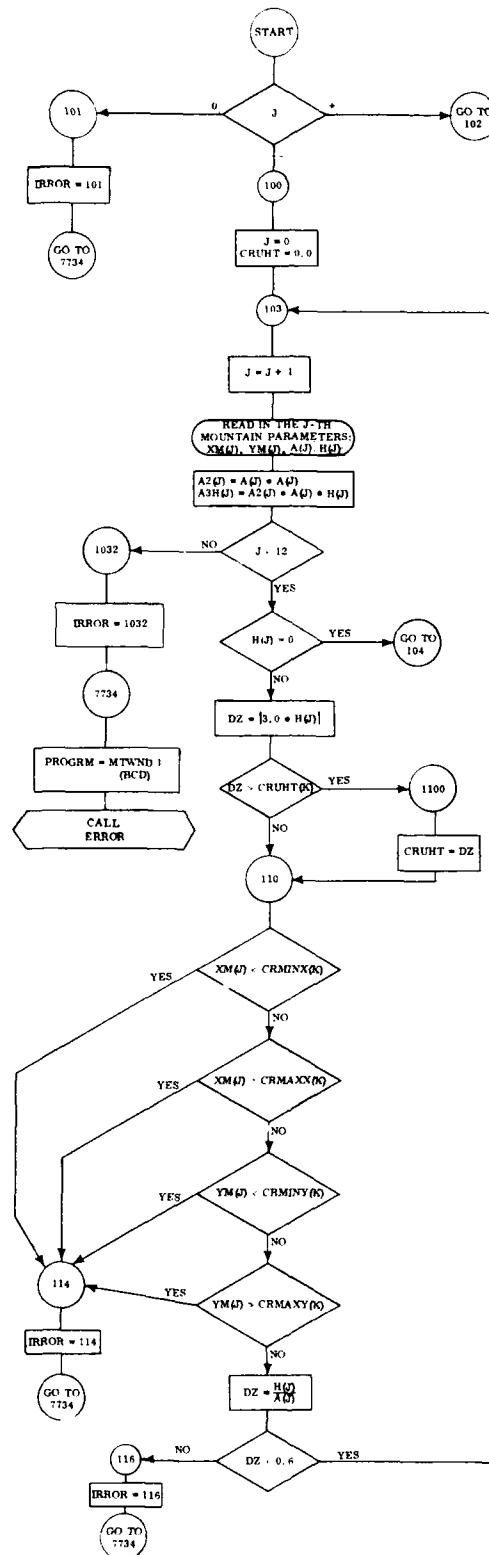
The height $CRUHT(K)$ of the Kth cell is defined as three times the height of the tallest mountain. The error subroutine is called if a mountain ratio exceeds 0.6, if a mountain does not lie within the boundaries of the Kth local cell, or if the number of mountains exceeds the maximum of twelve mountains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

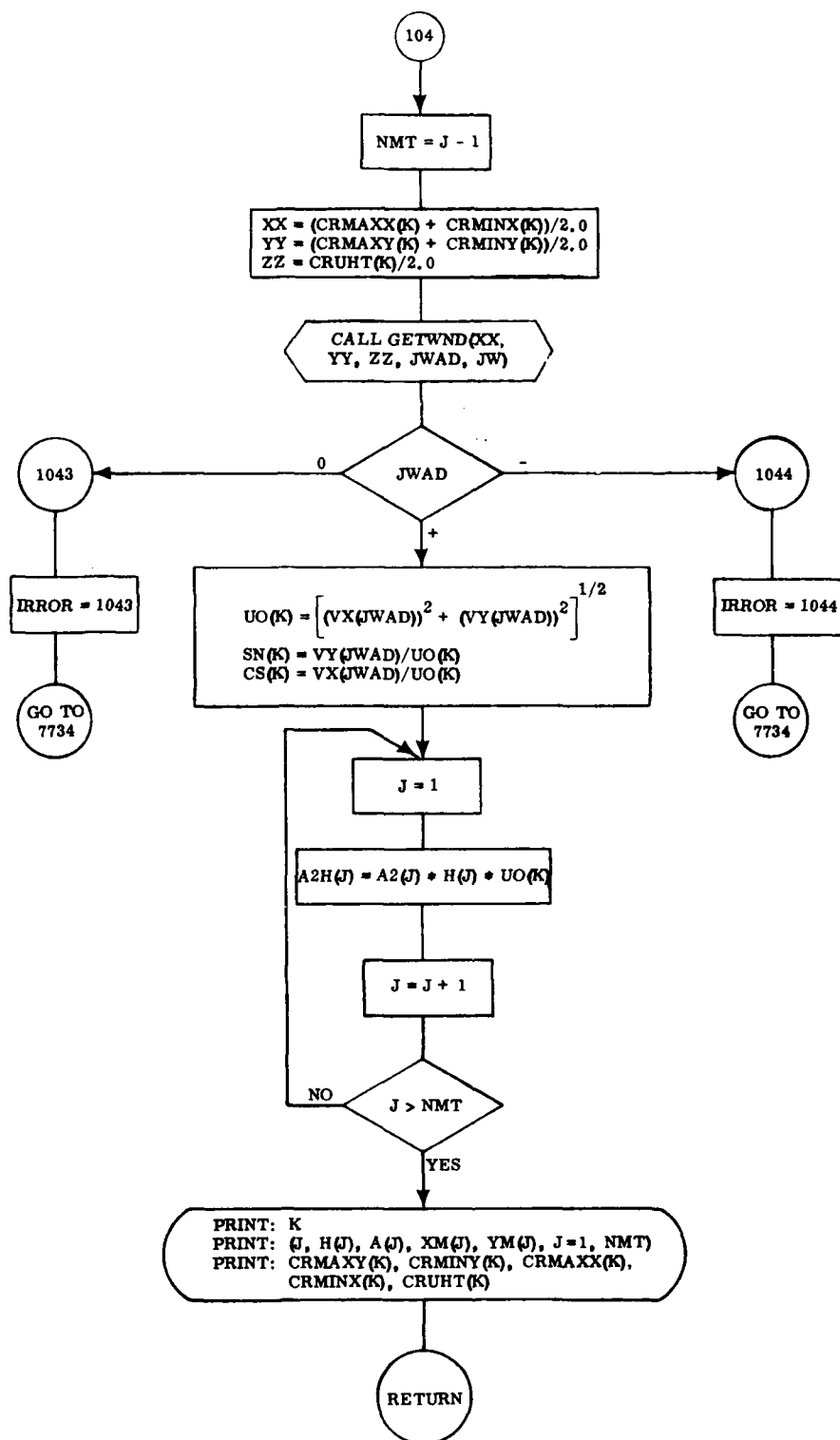
Subroutine RGWND1(J, K, AX, AY, AZ) (FC-14)

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



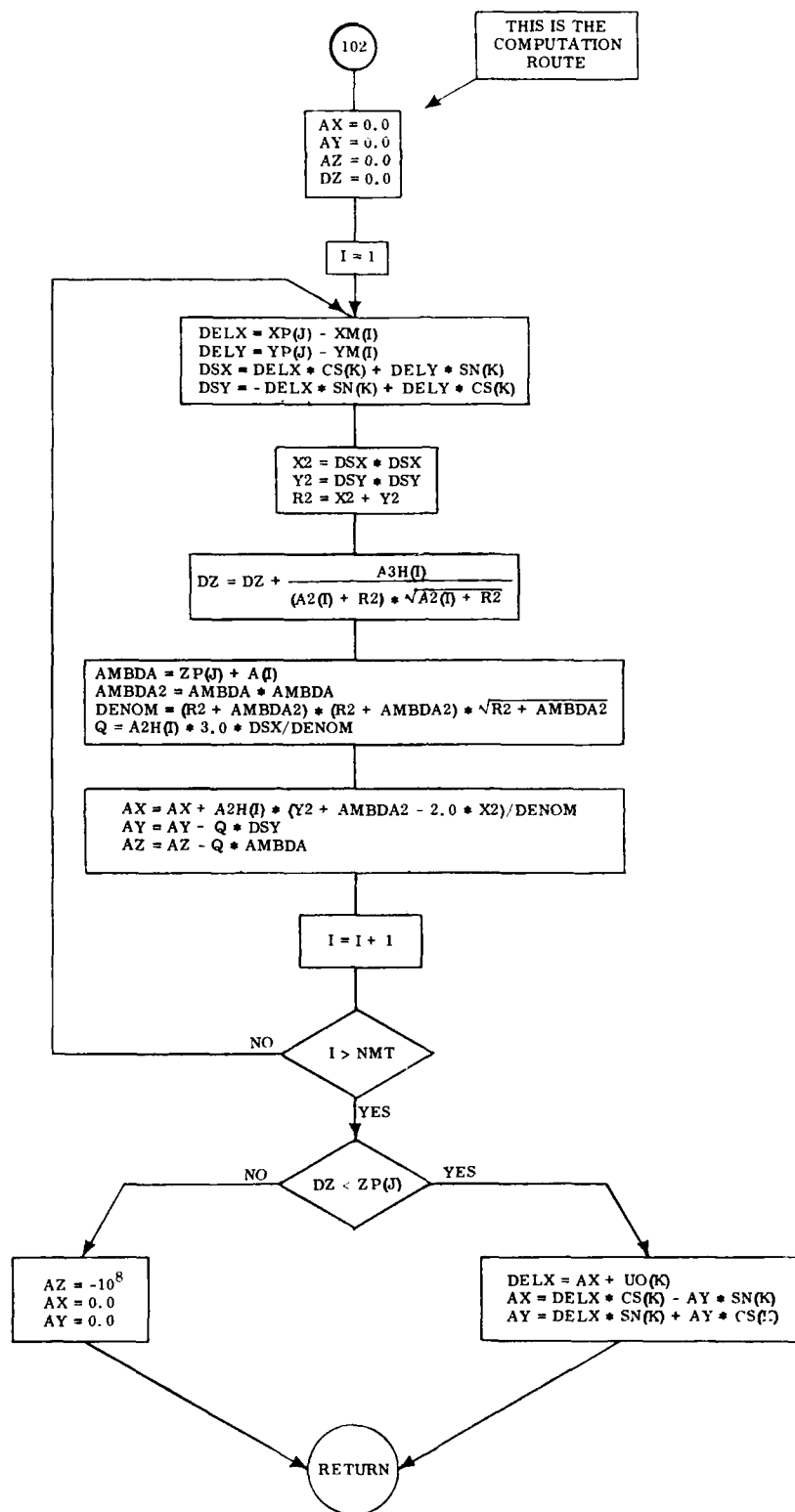
(a)

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 I th 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 K th local cell boundaries of north ($CRMXY(K)$), south ($CRMINY(K)$), east ($CRMXX(K)$) and west ($CRMINX(K)$). The height, $CRUHT(K)$, of the K th 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 K th cell, or the number of ridges exceeds the maximum of twelve ridges allowed in the K th cell.

Next, the read route computes the geometric center (XX, YY, ZZ) of the K th 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 J th particle from the I th 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

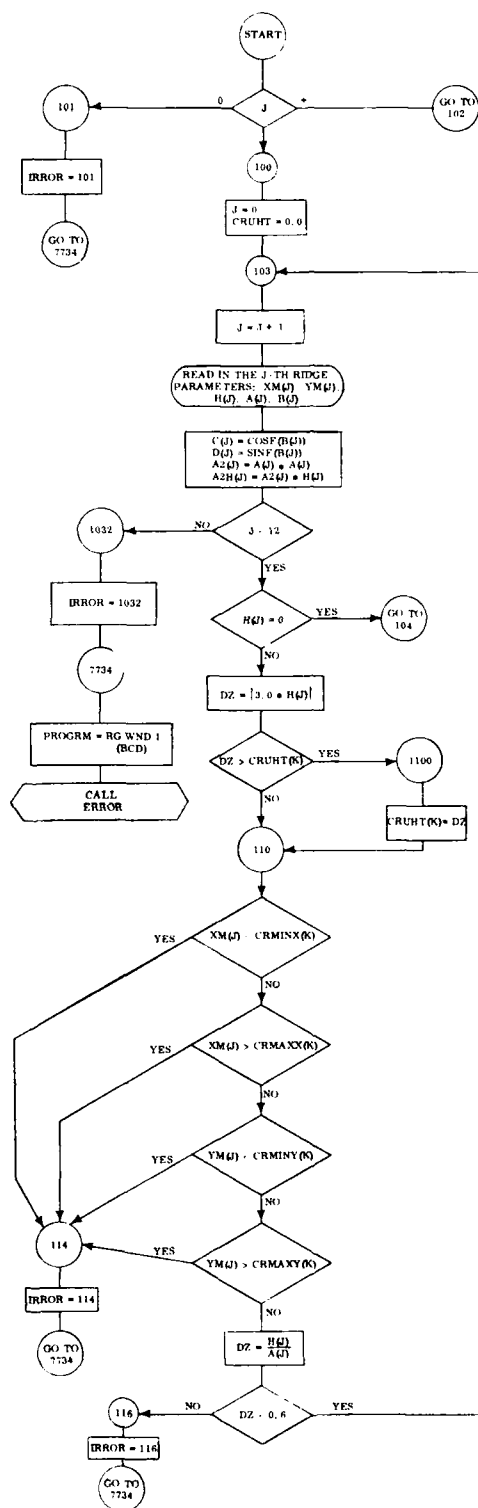
position. The results for each ridge are summed with the succeeding ridges to yield total height and perturbed wind vectors due to all the ridges at this point. Next, the particle height, $ZP(J)$, is checked against the total analytical ground height, DZ , to determine impact; upon which, the velocity $(0, 0, -10^8) \text{ msec}^{-1}$ 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.

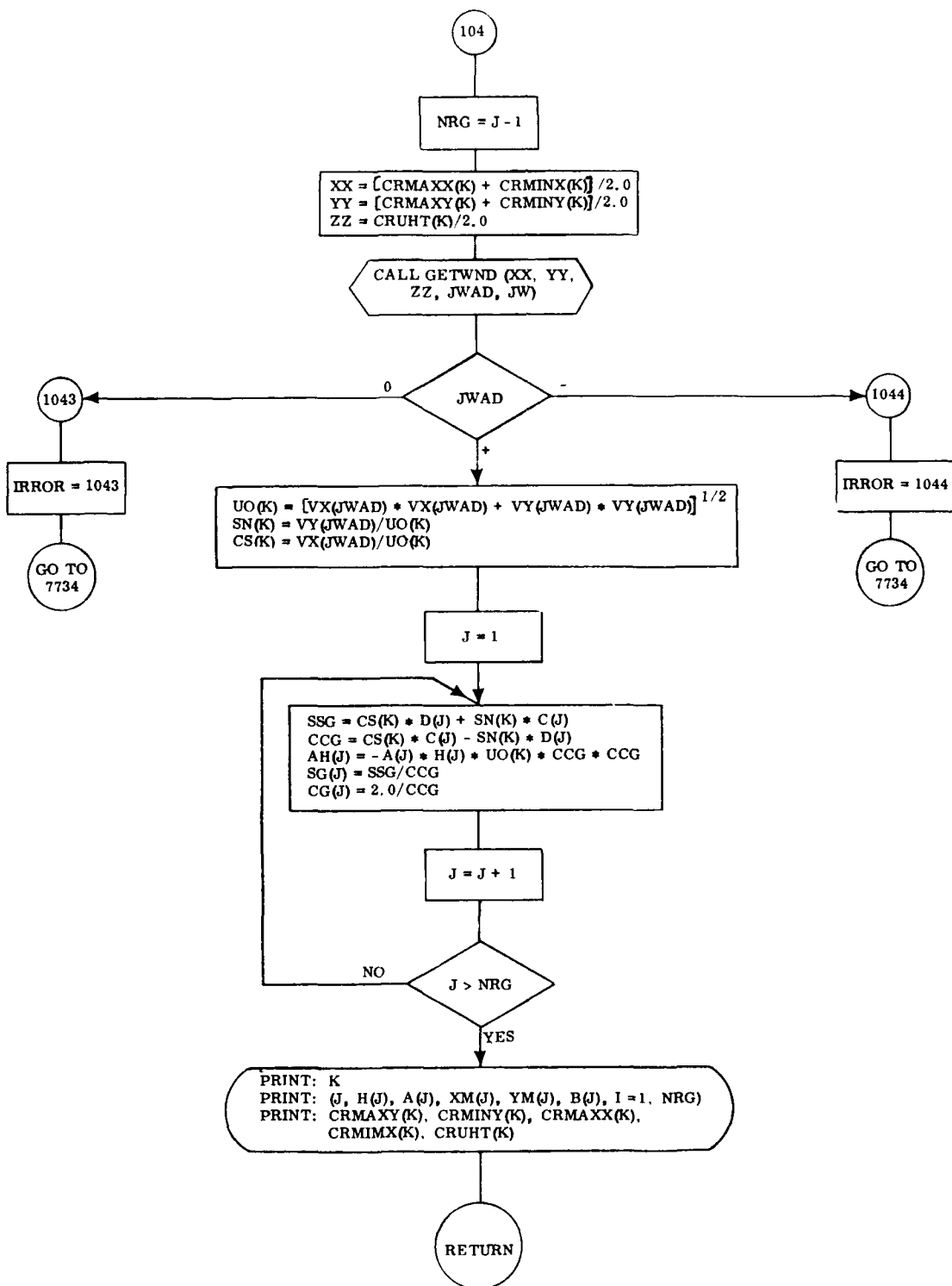
The CBREZ1 subroutine serves the dual purpose of reading the sea-breeze data and computing the sea-breeze velocity components for the J th particle at location $XP(J)$, $YP(J)$, $ZP(J)$ and time $TP(J)$, after first checking for particle impact at sea level. The programming of CBREZ1 is divided into 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 N th harmonic, and are printed out as the respective symbols: $OMGN(N)$, $AJZ(N)$, $AJX(N)$, $AJY(N)$, $AKN1(N)$, $ALN1(N)$, $AKN2(N)$, $ALN2(N)$, $BLOW1(N)$, $BLOW2(N)$,



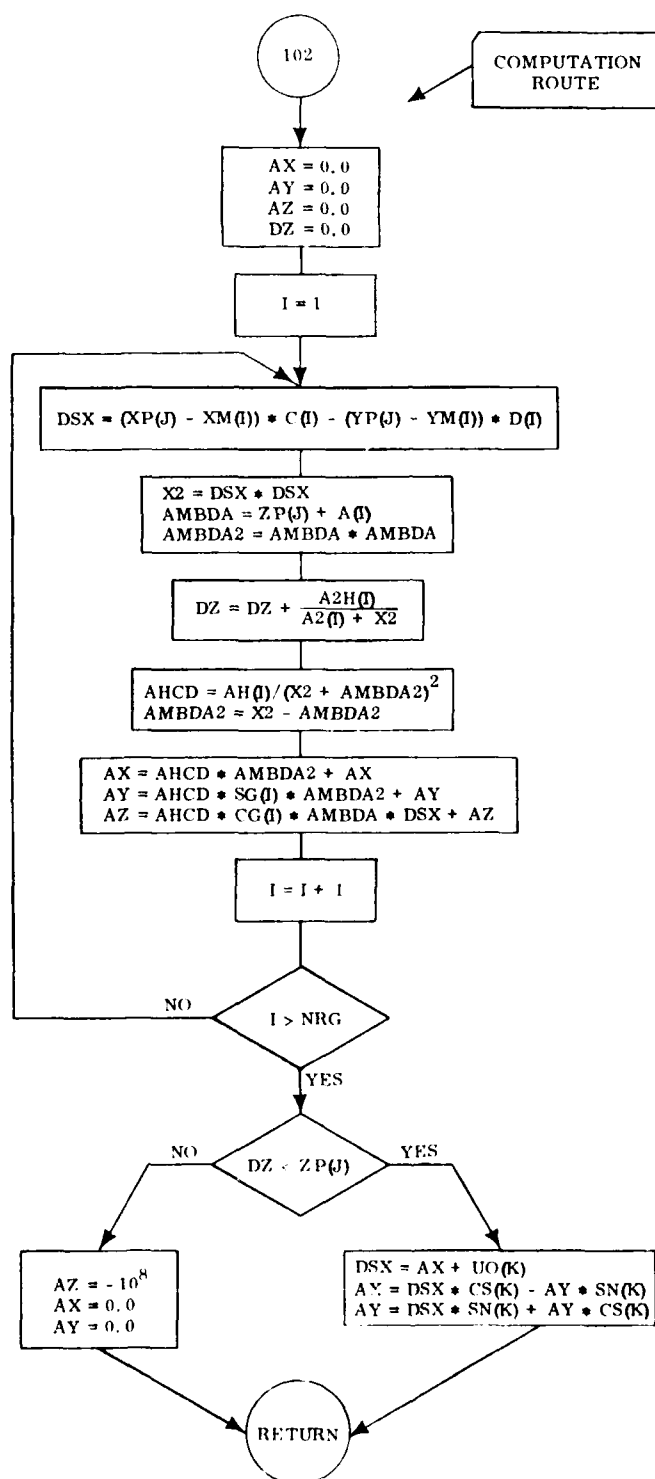
(a)

FC-14. Flow Charts for Subroutine RGWND1



(b)

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



(c)

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

ANN1(N), ANN2(N), PHIN(N), ENUN(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 (CRMXY(K)), south (CRMNY(K)), east (CRMXX(K)), and west (CRMNX(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 10^5 m
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	$^{\circ}K$	Expressed in degrees Kelvin: $\theta_o = 300^{\circ}K$
Unperturbed temperature gradient	$\Gamma = (d\theta_o/dz)$	GRAD	$^{\circ}K m^{-1}$	A constant z-independent positive value must be used; typical values $4 \times 10^{-3} \leq \Gamma \leq 7.5 \times 10^{-3}$
Thermal eddy diffusivity	K	AKY	$m^2 sec^{-1}$	A constant z-independent value must be used; typical values $25 \leq K \leq 75$
Magnitude of nth temperature differential	T_n^*	DELTX(N)	$^{\circ}K$	First harmonic will generally be less than $10^{\circ}K$, 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)	$\bar{K}_{n1}$	$\epsilon_1 U_1$
BLOW2(N)	$\bar{K}_{n2}$	$\epsilon_2 U_2$
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:

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

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

$$\alpha = g/\theta_0 .$$

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

$$\Omega = n\Omega .$$

When the foregoing substitution is made it becomes possible to drop the subscript n , it being understood that we are dealing with the n th 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:

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

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

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

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

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

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

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

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

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

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

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

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

where

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

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

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

and

$$B_1 = \xi_1^2 - \xi_2^2 - 4 (L A_7 \cos h - A_4 A_6 \sin \theta_4) ,$$

$$B_2 = 2 \xi_1 \xi_2 - 4 (A_4 A_6 \cos \theta_4 + L A_7 \sin h) ,$$

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

We define

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

$$m = \beta/2 .$$

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

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

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

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

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

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

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

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

where

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

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

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

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

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

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

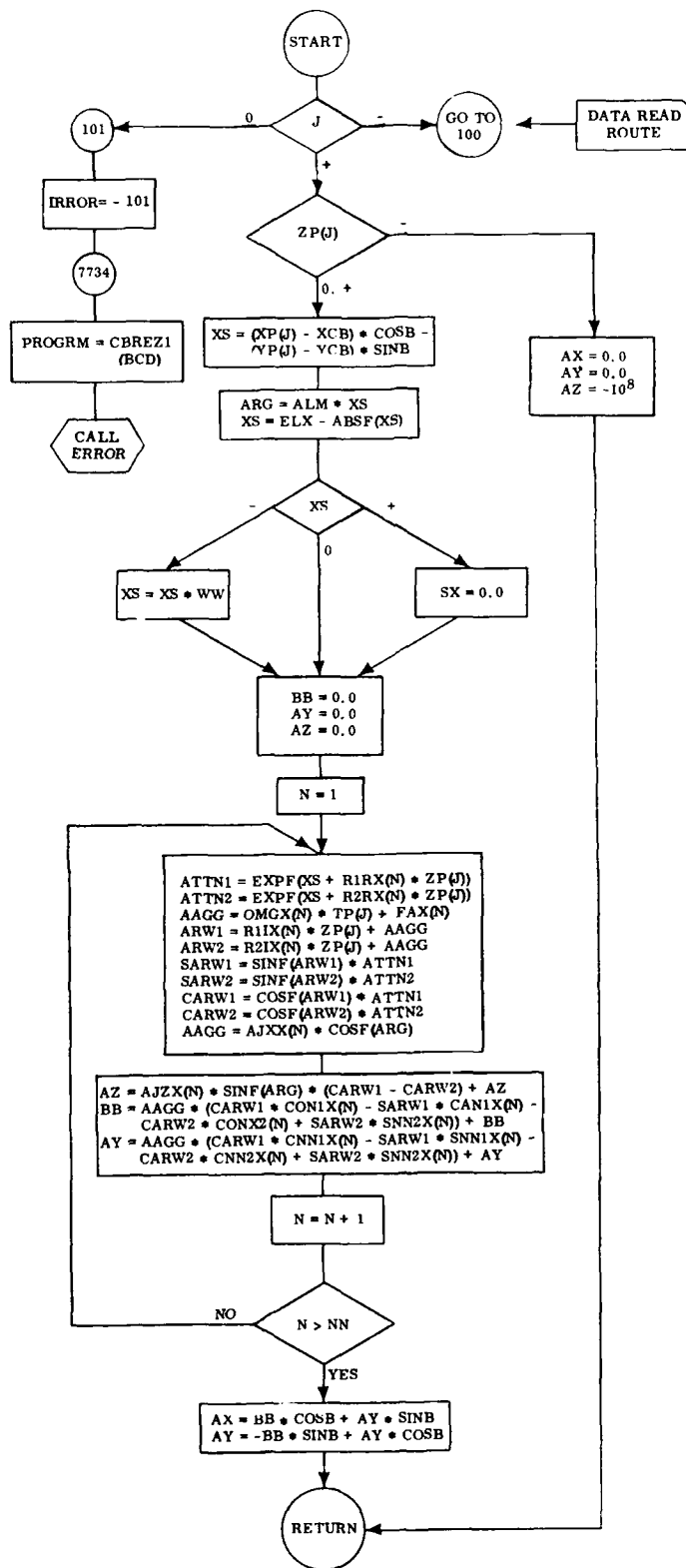
At this point all the mode-dependent constants necessary to describe the nth mode wind field have been computed.

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.)

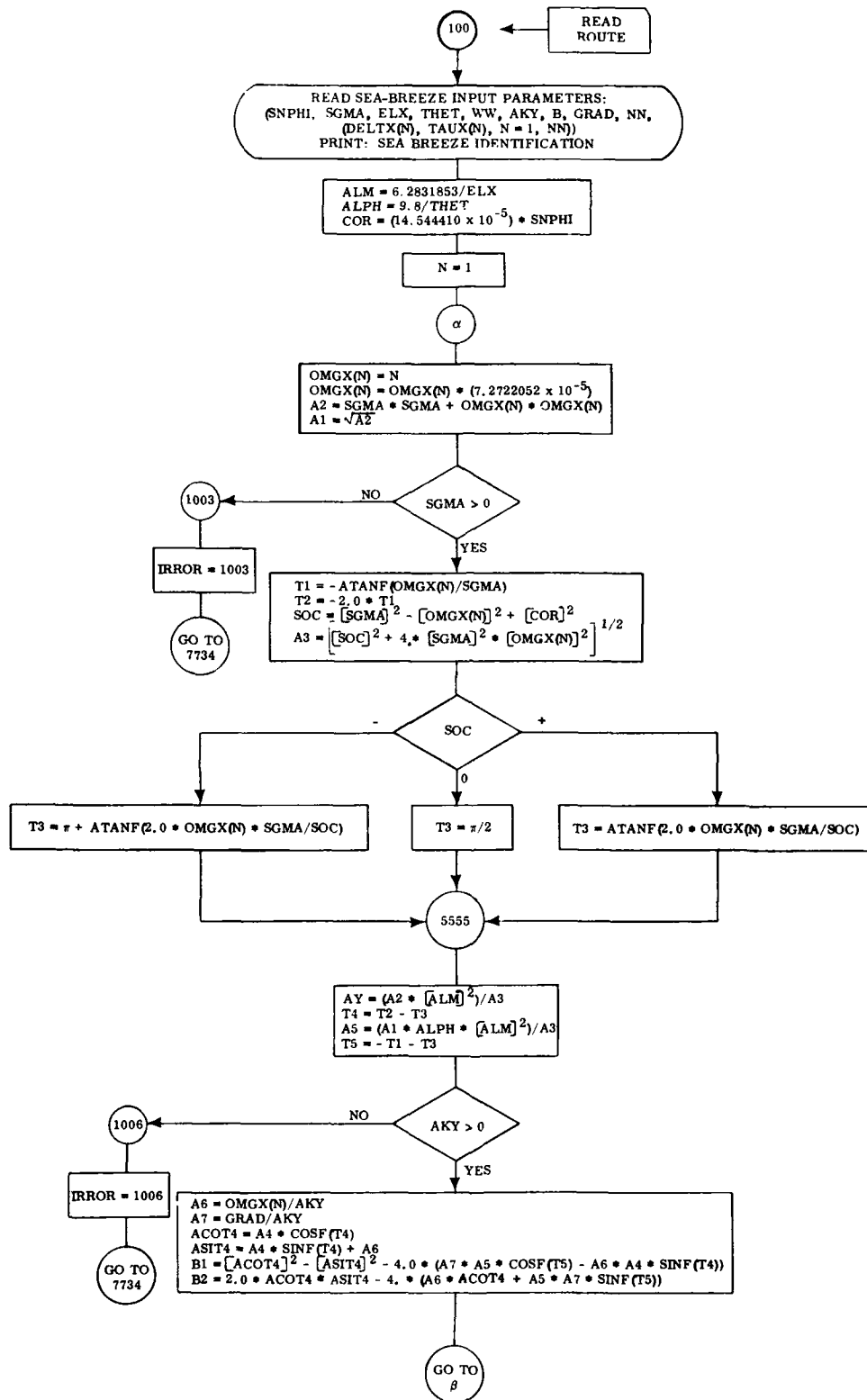
Upon entrance, the particle coordinates X and Y are checked first to determine if the particle is over the in-core topography block. If the particle is over in-core topo, a transfer is made to statement number 11 where retrieval begins. If it is not, a second check is made to determine if the particle 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



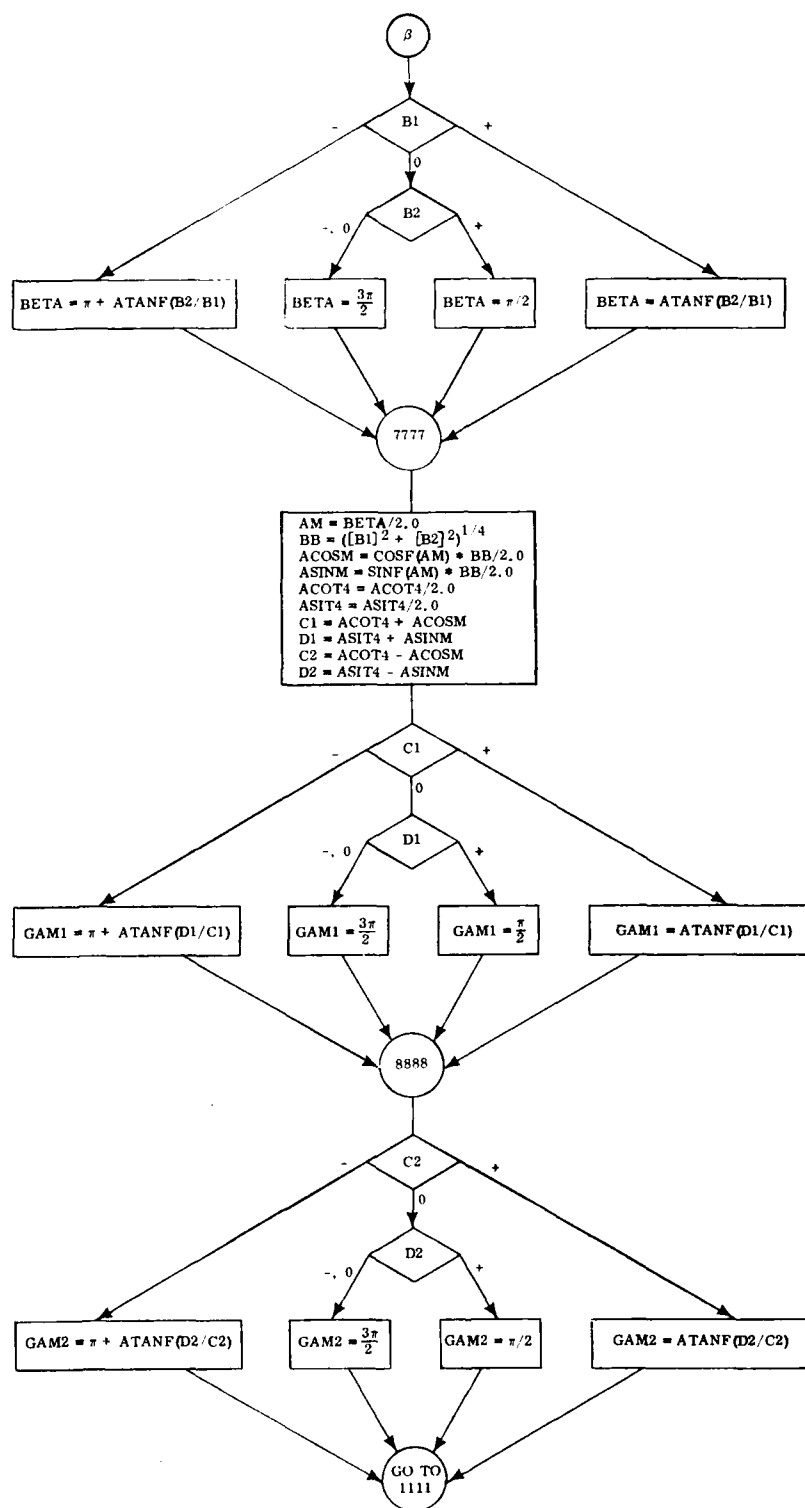
(a)

FC-15. Flow Charts for Subroutine CBREZ1



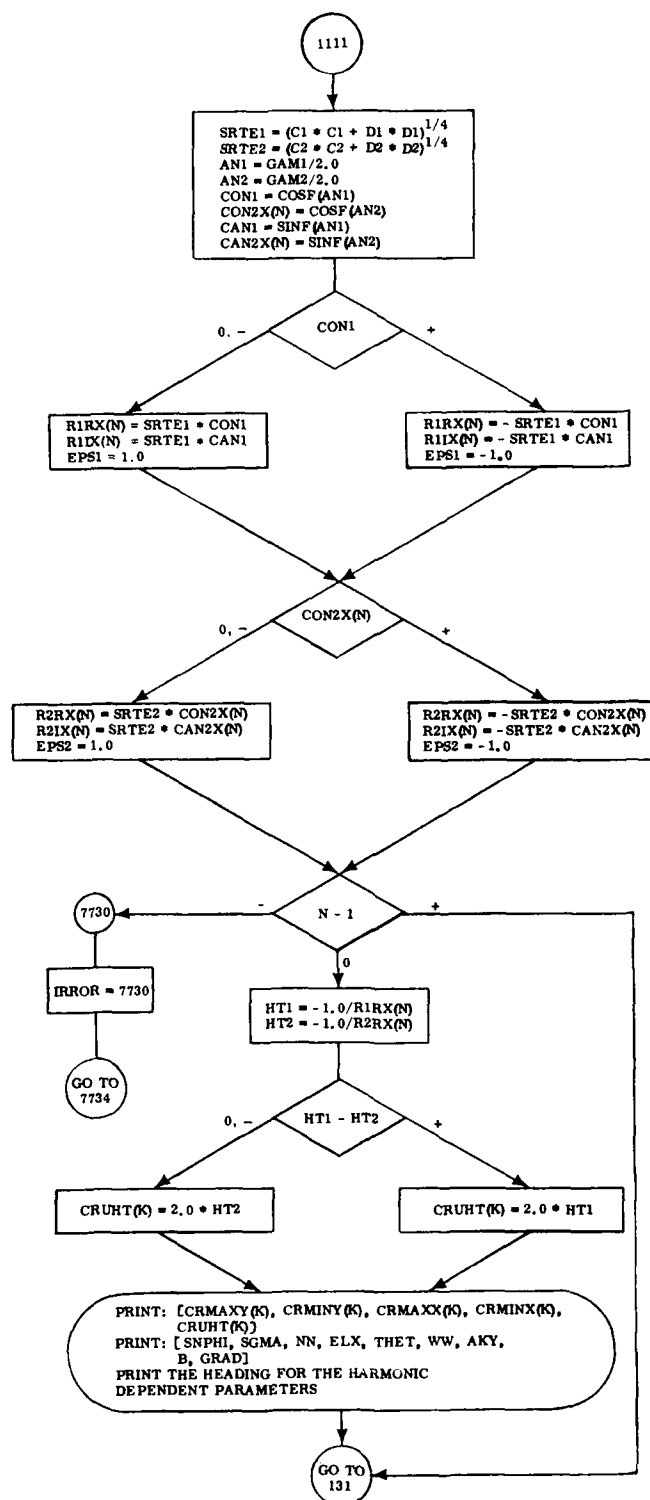
(b)

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



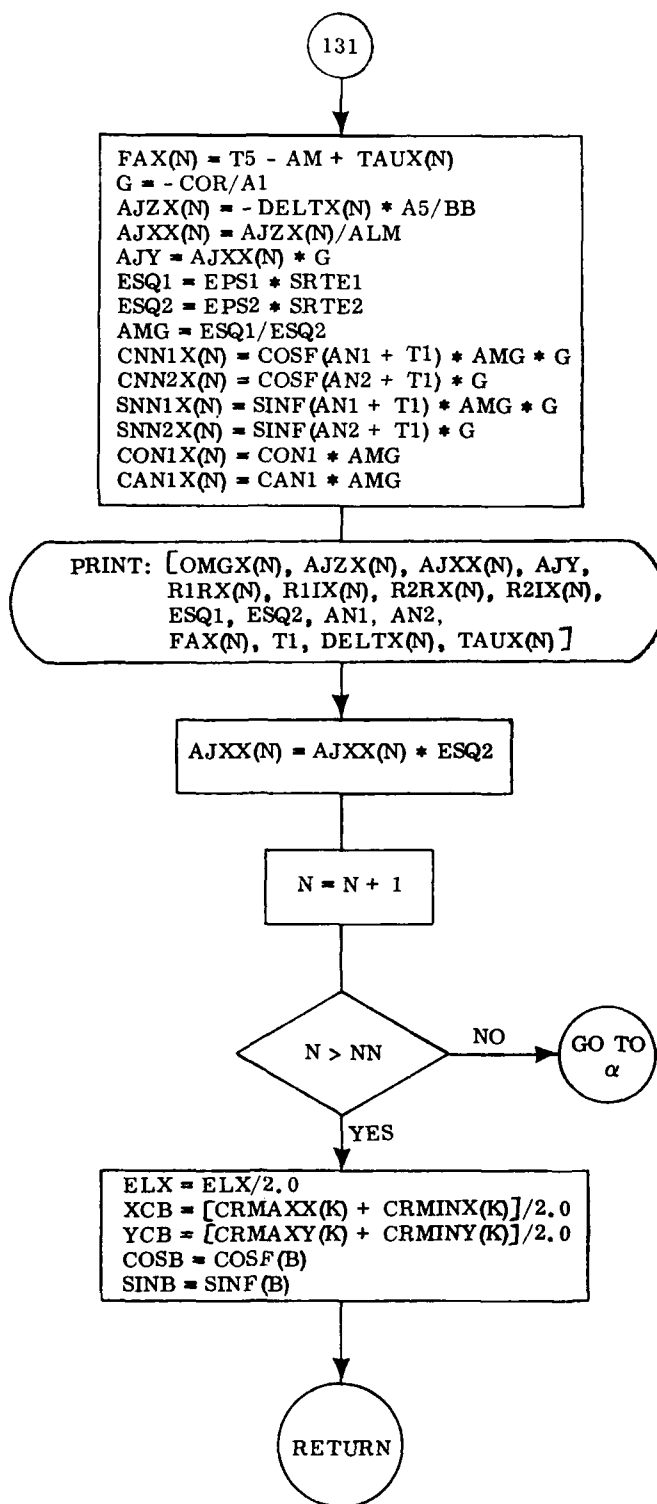
(c)

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



(d)

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

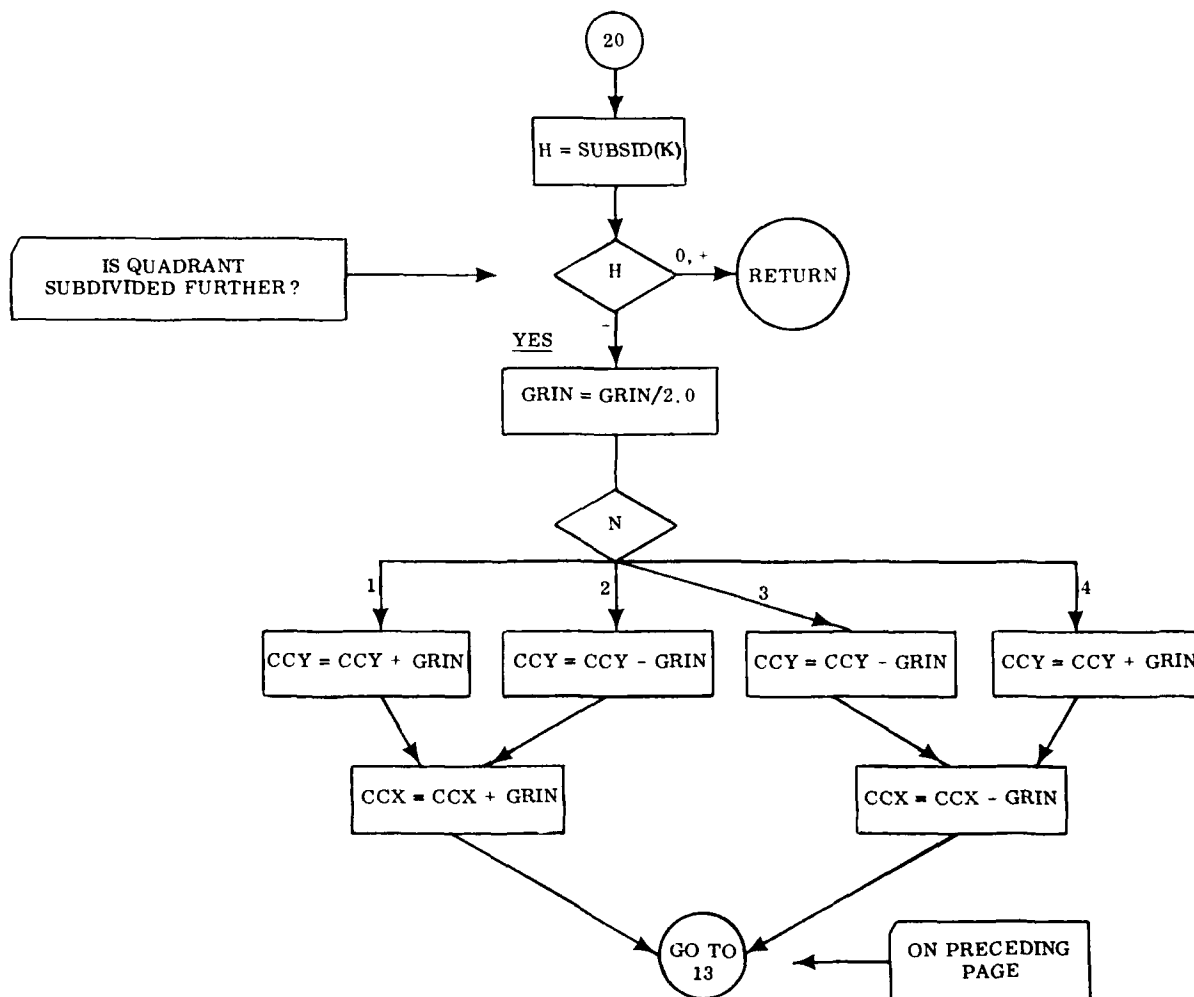


(e)

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.



(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 = 0
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 MEKWIN.

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 Sub-division 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:

<u>Identification Number</u>	<u>Local Circulation Type</u>
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	Same as records 5 and 6 but for second data block	
8		
.		
.		
.		
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

$$\begin{array}{cccccc}
 S(1, 1) & S(1, 2) & S(1, 3) & . & . & . & S(I, JJ) \\
 S(2, 1) & S(2, 2) & S(2, 3) & . & . & . & S(2, JJ) \\
 . & . & . & . & . & . & . \\
 . & . & . & . & . & . & . \\
 S(II, 1) & S(II, 2) & S(II, 3) & . & . & . & S(II, JJ)
 \end{array}$$

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, $\ln(SD)^{\dagger}$ 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.

```

$IBFIC FALRA LIST,DECK,M9472 FALR
C SUBROUTINE FALKAT FALR 1
C W.Y.G.ING TECHNICAL OPERATIONS RESEARCH FALR 2
C DEC 14 1965 FALR 3
C SUBROUTINE FALKAT(ALT,PSIZE,FV,ATEMP,RHO,FROG,ISOUT) FALR 4
C FALR 5
C ***** FALR 6
C FALR 7
C SUBROUTINE FALKAT,USING DAVIE'S EQUATIONS, COMPUTES THE SETTLING FALR 8
C RATE OF PARTICLES BY USING PARTICLE DIAMETER (MICRONS), SETTLING FALR 9
C (METERS ABOVE MSL) AND A STANDARD ATMOSPHERE FOR WHICH THE DENSITY FALR 10
C AND VISCOSITY HAVE BEEN TABULATED IN ARRAYS RHO(I) AND ATEMP(I) FALR 11
C RESPECTIVELY. FALR 12
C FALR 13
C ***** FALR 14
C FALR 15
C DIMENSION ATEMP(260),RHO(260) FALR 16
C REAL LOG10 FALR 17
C FALR 18
C ***** FALKAT GLOSSARY ***** FALR 19
C FALR 20
C ALT HEIGHT OF THE PARTICLE ABOVE MSL (METERS) FALR 21
C ATEMP(I) DYNAMIC VISCOSITY OF AIR AT ((I-1)*200) METERS ABOVE FALR 22
C MSL. (KILOGRAM/METER*SECOND) FALR 23
C CORR THE CORRECTION COEFFICIENT * SQUARE OF THE REYNOLDS FALR 24
C NUMBER. FALR 25
C FROG ((4/3)*PARTICLE DENSITY*GRAVITY*(CUBIC METER)/ CUBIC FALR 26
C MICRON). KILOGRAM-METER/(1000. SEC.)*(CUBIC MICRON)) FALR 27
C FV SETTLING RATE (METERS/SEC) FALR 28
C PSIZE PARTICLE DIAMETER (MICRONS) FALR 29
C RHO(I) AIR DENSITY AT ((I-1)*200 METERS ABOVE MSL. (KILO- FALR 30
C GRAMS/ CUBIC METER) FALR 31
C FALR 32
C ***** FALR 33
C FALR 34
C 2 FORMAT(//38H DAVIE'S EQUATIONS ARE INACCURATE FOR ,F12.3,12H MICROFALR 35
C INS AT ,F12.3,7H METERS) FALR 36
C FALR 37
C ***** FALR 38
C ***** FALR 39
C FALR 40
C I IS THE INDEX (IN THE ARRAYS RHO(I) AND ATEMP(I)) THAT IDENTIFIES FALR 41
C THE 200 METER THICK LAYER CONTAINING THE PARTICLE. THE ADDITION FALR 42
C OF 6.5 TO THE INDEX BEFORE TRUNCATION INSURES THAT PARTICLES FALR 43
C BETWEEN 100 AND 1100 METERS ABOVE MSL WILL LIE IN THE SIXTH LAYER FALR 44
C WHICH HAS ITS CENTER LOCATED AT MSL, PARTICLES BETWEEN 100 AND 300 FALR 45
C METERS ABOVE MSL WILL LIE IN THE 7TH LAYER WHICH HAS ITS CENTER FALR 46
C LOCATED AT 200 METERS ABOVE MSL, AND SO FORTH. FALR 47
C IF(ALT/200.0)+0.5 FALR 48
C V0=PSIZE/ATEMP(I) FALR 49
C V1=PSIZE*V0/RHO FALR 50
C CORR=V1/RHO(I)*V0 FALR 51
C IF(CORR-140.0)100,100,100 FALR 52
C 150 IF (CORR-+50.0)200,101,101 FALR 53
C 151 CORR EXCEEDS THE OPER. RANGE OF DAVIE'S EQUATIONS. HOWEVER, THE FALR 54
C NUMERICAL RESULT IS STILL USED. FALR 55
C 151 WRITE (1001,2)PSIZE,ALT FALR 56
C GO TO 200 FALR 57
C 100 CORR IS LESS THAN OR EQUAL TO 140. FALR 58
C 100 FV=V1*(41000.0 / +CORR*(1-2.9305E+2+CORR*(2.0154 -6.9105E-2*CORR))) FALR 59
C GO TO 300 FALR 60

```

```

C200  CORR IS GREATER THAN 140.
      WLOGA=ALOG10(CORR)-20.775
      FV=50637.0 *V1*CORR**((WLOGA+WLOGA+445.75)*0.001255)
C300  DRAG SLIP CORRECTED FALL RATE
      FV=FV*(1.0+2.35E-17*(PSIZE**NDS(1)))
C301  RETURN
      END

```

```

FALL 01
FALL 02
FALL 03
FALL 04
FALL 05
FALL 06
FALL 07

```

END

END

\$IBFTC DUMP	LIST,DECK,M94/2	DUMP	1
C	SUBROUTINE DUMPP	DUMP	2
C	T.W.SCHWENKE TECHNICAL OPERATIONS RESEARCH	SR DUMPP	3
C	28 NOVEMBER 1966	DUMP	4
C	*****	DUMP	5
C	THIS SUBROUTINE SELECTS THE BEST SET OF PARTICLES TO DUMP,	DUMP	6
C	SORTS IT INTO THE LOW NUMBERED END OF THE PARTICLE ARRAYS, WRITES	DUMP	7
C	IT OUT ONTO THE APPROPRIATE TAPE AND ADJUSTS PARTICLE SET COUNTERS	DUMP	8
C	THE SET SELECTED FOR DUMPING IS THE GROUNDED PARTICLES SET WHEN-	DUMP	9
C	EVER DUMPING IT WOULD MAKE SUFFICIENT ROOM FOR THE INCOMING BLOCK	DUMP	10
C	OF N PARTICLES. IF THIS IS NOT THE CASE, THE LARGEST PARTICLE SET	DUMP	11
C	IS SELECTED.	DUMP	12
C	***** GLOSSARY *****	DUMP	13
C	FOR ADDITIONAL GLOSSARY ENTRIES SEE SUBROUTINE LINK5	DUMP	14
C	FMAST SEE XPT	DUMP	15
C	IC() THE CONTROL INTEGER ARRAY. SEE LINK 5 GLOSSARY	DUMP	16
C	ICON BLOCKING SORT MODE INDICATOR. 0=FIRST PASS, +=BOTTOM LOOP	DUMP	17
C	,-=TOP LOOP	DUMP	18
C	IOTOPO THE OFF-TOPO MEMORY TAPE NUMBER	DUMP	19
C	IOWIND THE OUT-OF-WIND DATA MEMORY TAPE NUMBER	DUMP	20
C	IPAKOT THE TIME LIMIT BOUNDARY MEMORY TAPE NUMBER	DUMP	21
C	IPOUT THE TRANSPORT MODULE INTERMEDIATE OUTPUT TAPE NUMBER	DUMP	22
C	IRSET A MARKER FOR THE BLOCKING SORT WHICH INDICATES BY THE	DUMP	23
C	VALUE 1 THAT THE TEMPORARY STORAGE LINE FOR A PARTICLE	DUMP	24
C	IS LOADED AND MUST BE EVENTUALLY UNLOADED	DUMP	25
C	ISOUT THE FORTRAN SYSTEM OUTPUT TAPE NUMBER	DUMP	26
C	J A GENERAL INDEX. IN THE BLOCKING SORT, IT IS USED TO	DUMP	27
C	IDENTIFY THE PARTICLE THAT WAS JUST CLASSIFIED	DUMP	28
C	JB BOTTOM LOOP INDEX FOR THE BLOCKING SORT	DUMP	29
C	JBL BLANK LINE INDEX FOR THE BLOCKING SORT	DUMP	30
C	JFR USED TO RECORD THE INDEX OF A FREE (BLANK) LINE IN THE	DUMP	31
C	BOTTOM PART OF THE PARTICLE ARRAY DURING THE CONSOLIDATION	DUMP	32
C	OF N BLANKS INTO THE TOP OF THE ARRAY	DUMP	33
C	JT TOP LOOP INDEX FOR THE BLOCKING SORT	DUMP	34
C	JTEST A TEMPORARY STORAGE THAT EVENTUALLY CONTAINS THE NUMBER	DUMP	35
C	OF PARTICLE DESCRIPTIONS IN THE CLASS TO BE DUMPED	DUMP	36
C	JTEST1 A TEMPORARY STORAGE WHICH EVENTUALLY CONTAINS THE NUMBER	DUMP	37
C	INDICATING THE KIND (CLASS) OF PARTICLE DESCRIPTION TO BE	DUMP	38
C	DUMPED	DUMP	39
C	N THE NUMBER OF PARTICLES IN THE DATA BLOCK THAT IS WAITING	DUMP	40
C	TO BE READ NEXT AT THE TIME WHEN DUMPP IS CALLED	DUMP	41
C	NALOFT THE DIMENSIONED (MAXIMUM) SIZE OF THE PARTICLE ARRAY	DUMP	42
C	NBMAX THE MAXIMUM NUMBER OF PARTICLE DESCRIPTIONS THAT CAN BE	DUMP	43
C	INCLUDED IN A SINGLE BLOCK AS WRITTEN ON ANY MEMORY OR	DUMP	44
C	INTERMEDIATE OUTPUT TAPE	DUMP	45
C	NFREE THE NUMBER OF BLANK LINES (DENOTED BY FMAS()=0) IN THE	DUMP	46
C	PARTICLE ARRAYS	DUMP	47
C	NG A COUNT OF IN-CORE GROUNDED PARTICLES	DUMP	48
C	NLOST A COUNT OF THE PARTICLES IN THE PARTICLE ARRAY BUT LOCATED	DUMP	49
C	BEYOND THE COORDINATE LIMITS OF THE WIND OR TOPO DATA SETS	DUMP	50
C	NTAP A TEMPORARY STORAGE FOR THE NUMBER OF THE TAPE ONTO WHICH	DUMP	51
C	THE DUMP IS TO BE MADE	DUMP	52
C	NTI A COUNT OF IN-CORE PARTICLES THAT HAVE REACHED THE TIME	DUMP	53
C	BOUNDARY (ENDTIM)	DUMP	54
C	NTU A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE TOPO DATA	DUMP	55
C	NW A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE WIND DATA	DUMP	56

```

C      N1      AN ASSIGNED GO TO BRANCH POINT FOR THE CLASSIFYING CODE      DUMP 61
C      N2      SEE N1                                                         DUMP 62
C      N3      SEE N1                                                         DUMP 63
C      N4      SEE N1                                                         DUMP 64
C      PST     SEE XPT                                                         DUMP 65
C      TPT     SEE XPT                                                         DUMP 66
C      XPT     TEMPORARY STORAGE FOR XPT ( ) SOMETIMES USED TO START A SORT DUMP 67
C      YPT     SEE XPT                                                         DUMP 68
C      ZPT     SEE XPT                                                         DUMP 69
C                                                     DUMP 70
C ***** DUMP 71
C                                                     DUMP 72
C      COMMON /SET1/                                                         DUMP 73
C      1      DIAM      ,DETID(12),IRISE      , IEXEC      , ISIN      , ISOUT      , DUMP 74
C      2      SD        , SPAR      , SSAM      , TME        , TMP1      , TMP2      , DUMP 75
C      3      T2M       , U        , VPR       , W          , X          , Z          , DUMP 76
C      4      WHY(4)    , RMIN      , IDISTR    , SPAR1      , SPAR2      , SPAR3      , DUMP 77
C      5      SPAR4     , SPAR5     , SPAR6     , SPAR7      , SPAR8      , SPAR9      , DUMP 78
C                                                     DUMP 79
C ***** DUMP 80
C                                                     DUMP 81
C      COMMON /SET2/                                                         DUMP 82
C      1      S          , SUBSID    , GRINT     , BXLL      , BXLU      , BYLL      , DUMP 83
C      2      BYLU      , TXLL      , TXLU      , TYLL      , TYLU      , XGZ       , DUMP 84
C      3      YGZ       , NBLCK     , HTOPO    , TTOPU     , ILIM      , JLIM      , DUMP 85
C      4      KLIM      , II        , JU        , KK        , XP        , YP        , DUMP 86
C      5      ZP        , FMAS     , TP        , PS        , VX        , VY        , DUMP 87
C      6      VZ        , IL        , JL        , IBADD     , WGRINT    , NSTRAT    , DUMP 88
C      7      WLLX      , WLLY      , WURX      , WURY      , BOTHIT    , IPARIN    , DUMP 89
C      8      IOTOPU    , IOWIND    , IHTOPU    , IPOUT     , IPAROT    , JTOP1     , DUMP 90
C      9      JWIND1    , IRROR     , TLIMIT    , ENDTIM    , IC        , IBYPAS    , DUMP 91
C      1      JTOPJ     , NLOST     , N3        , NTU       , NTI       , NW        , DUMP 92
C      2      NALOFT    , JTIME1    , NBMAX     , NFREE     , N         , NCL       , DUMP 93
C      3      CRMXY     , CRUHT     , NCRTYP    , BZ        , CRMINX    , CRMINY    , DUMP 94
C      4      UO        , SN        , CS        , NLUCIR    , DTLOC     , ATEMP     , DUMP 95
C      5      RHU       , NA        , TGZ       , DTMAC     , FROG      , CRMAXX    , DUMP 96
C      6      ROPART                                         DUMP 97
C      DIMENSION TOPULM(4,4) ,NINTAR(4)      ,ITOPLM(3,4) DUMP 98
C      DIMENSION S(1,10)    ,SUBSID(400)    ,IC(18)      DUMP 99
C      DIMENSION XP(200)    ,YP(200)        ,ZP(200)      ,FMAS(200) DUMP 100
C      DIMENSION TP(200)    ,PS(200)        ,ATEMP(260)   ,RHO(260)  DUMP 101
C      DIMENSION VX(1500)   ,VY(1500)      ,VZ(1500)    ,IL(70)    DUMP 102
C      DIMENSION JL(70)     ,IBADD(70)      ,WURX(70)    DUMP 103
C      DIMENSION WGRINT(70) ,WLLX(70)    ,WLLY(70)    DUMP 104
C      DIMENSION WURY(70)   ,BOTHIT(70)   ,SN(6)       ,CS(6)     DUMP 105
C      DIMENSION CRMINX(6)  ,CRMAXX(6)   ,CRMINY(6)   ,CRMXY(6)  DUMP 106
C      DIMENSION CRUHT(6)   ,NCRTYP(6)    ,UO(6)       DUMP 107
C                                                     DUMP 108
C ***** DUMP 109
C                                                     DUMP 110
C      1      FORMAT(1H1,4X17,15H LOST PARTICLES) DUMP 111
C      2      FORMAT(/6X,2HXP,10X2HYP,10X,2HZP,10X,2HTP,10X,2HPS,8X,4HFMAS) DUMP 112
C      3      FORMAT(1X,6E12.5) DUMP 113
C      4      FORMAT(10I5) DUMP 114
C      5      FORMAT(5X11HBEYOND TOPO) DUMP 115
C      6      FORMAT(5X13HTIME BOUNDARY) DUMP 116
C      7      FORMAT(5X11HBEYOND WIND) DUMP 117
C      8      FORMAT(5X8H GROUNDED) DUMP 118

```

```

C                                     DUMP 119
C *****DUMP 120
C                                     DUMP 121
C      DATA PROGRAM/6H DUMPP/                                     DUMP 122
C                                     DUMP 123
C *****DUMP 124
C *****DUMP 125
C                                     DUMP 126
C      MUST ANY PARTICLES BE DUMPED TO MAKE ROOM FOR THE INCOMING BLOCK DUMP 127
C      OF N PARTICLES OR TO CLEAR THE PARTICLE ARRAYS. YES TO 151 DUMP 128
C      IF(N-NFREE)150,150,151 DUMP 129
150 JTEST=0 DUMP 130
    GO TO 152 DUMP 131
C DUMP 132
C 151 WOULD DUMPING THE GROUNDED PARTICLES PROVIDE SUFFICIENT ROOM FOR DUMP 133
C THE BLOCK OF N INCOMING PARTICLE DESCRIPTIONS. YES TO 1512 DUMP 134
151 IF(NFREE+NG-N)1511,1512,1512 DUMP 135
C DUMP 136
C 1512 PREPARE TO DUMP THE GROUNDED PARTICLE DESCRIPTIONS DUMP 137
1512 JTEST1=1 DUMP 138
    JTEST=NG DUMP 139
    GO TO 18 DUMP 140
C1511 DETERMINE WHICH SET OF PARTICLES TO DUMP DUMP 141
C FIND THE IDENTIFIER (JTEST1) AND SIZE (JTEST) OF THE MOST DUMP 142
C NUMEROUS CLASS OF PARTICLES IN THE PARTICLE ARRAYS DUMP 143
1511 IF(NLOST-NG)10,11,11 DUMP 144
10 JTEST=NG DUMP 145
    JTEST1=1 DUMP 146
    GO TO 12 DUMP 147
11 JTEST=NLOST DUMP 148
    JTEST1=2 DUMP 149
12 IF(JTEST-NT0)13,14,14 DUMP 150
13 JTEST=NT0 DUMP 151
    JTEST1=3 DUMP 152
14 IF(JTEST-NT1)15,16,16 DUMP 153
15 JTEST1=4 DUMP 154
    JTEST=NT1 DUMP 155
16 IF(JTEST-NW)17,18,18 DUMP 156
17 JTEST1=5 DUMP 157
    JTEST=NW DUMP 158
C 18 AT THIS POINT JTEST HAS MAX(NG,NLOST,NT0,NT1,NW). DUMP 159
C JTEST1 INDICATES THE KIND OF PARTICLE DESCRIPTION TO BE DUMPED DUMP 160
C SEE THE FOLLOWING CODE EXPLANATION FOR JTEST1=1 THRU 5 DUMP 161
C DUMP 162
C      JTEST1  NAME OF      KIND OF PARTICLE DESCRIPTIONS DUMP 163
C      VALUE   CLASS COUNTER TO BE DUMPED ONTO TAPE DUMP 164
C      1       NG          GROUNDED PARTICLES DUMP 165
C      2       NLOST       PARTICLES BEYOND THE AREAS FOR WHICH DUMP 166
C                          BOTH TOPO AND WINDS HAVE BEEN SPEC- DUMP 167
C                          IFIED DUMP 168
C      3       NT0         PARTICLES BEYOND THE LIMITS OF THE DUMP 169
C                          TOPO DATA CURRENTLY AVAILABLE IN CORE DUMP 170
C      4       NT1         PARTICLES THAT CANNOT BE VALIDLY DUMP 171
C                          TRANSPORTED FURTHER UNTIL THE WIND DUMP 172
C                          FIELD DESCRIPTION IS UPDATED DUMP 173
C      5       NW          PARTICLES BEYOND THE LIMITS OF THE DUMP 174
C                          WIND DATA CURRENTLY AVAILABLE IN CORE DUMP 175
C DUMP 176

```

C	TEST TO SEE THAT JTEST HAS AN ACCEPTABLE VALUE.	UNACCEPTABLE TO	DUMP 177
C	ERROR STOP AT 184		DUMP 178
C			DUMP 179
18	IF(JTEST)184,184,182		DUMP 180
184	IROR= 184		DUMP 181
7734	CALL ERROR(PROGRM,IROR,ISOUT)		DUMP 182
	GO TO 60		DUMP 183
C			DUMP 184
C 182	IS THE SIZE (JTEST) OF THE SELECTED CLASS GREATER THAN THE MAXIMUM		DUMP 185
C	ALLOWABLE OUTPUT BLOCK SIZE (NBMAX)). YES TO 183.		DUMP 186
182	IF(JTEST-NBMAX)181,181,183		DUMP 187
C			DUMP 188
C 183	RESET JTEST EQUAL TO THE MAXIMUM ALLOWABLE BLOCK SIZE SO THAT AN		DUMP 189
C	ACCEPTABLE BLOCK SIZE WILL BE DUMPED.		DUMP 190
183	JTEST=NBMAX		DUMP 191
C			DUMP 192
C 181	MAKE ASSIGNMENTS FOR THE EFFICIENT CONTROL OF THE CODE THAT WILL		DUMP 193
C	LATER BE USED TO CLASSIFY PARTICLE DESCRIPTIONS AS TO WHETHER		DUMP 194
C	THEY ARE TO BE DUMPED, NOT TO BE DUMPED, OR MERELY BLANK. ALSO		DUMP 195
C	DECREASE THE APPROPRIATE CLASS COUNTER BY THE NUMBER OF DESCRIPT-		DUMP 196
C	TIONS ABOUT TO BE DUMPED (JTEST) AND MAKE AN APPROPRIATE SETTING		DUMP 197
C	OF THE OUTPUT TAPE NAME NIAP.		DUMP 198
C 181	GO TO THE APPROPRIATE SELECTION CODE		DUMP 199
181	GO TO (19,20,21,22,23),JTEST1		DUMP 200
C			DUMP 201
C 19	CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF GROUNDED PARTICLES.		DUMP 202
C	GROUNDED PARTICLES ARE IDENTIFIED BY THE PATTERN -,- IN THE		DUMP 203
C	SIGNS OF FMAST() AND TPC() UNDER THE CONDITION THAT TPC()+LIMIT		DUMP 204
C	DOES NOT EQUAL ZEF).		DUMP 205
C	DESCRIPTIONS OF GROUNDED PARTICLES ARE ALWAYS WRITTEN UNTO THE		DUMP 206
C	TRANSPORT INTERMEDIATE OUTPUT TAPE IPOUT		DUMP 207
19	NG=NG-JTEST		DUMP 208
	NIAP=IPOUT		DUMP 209
	ASSIGN 300 TO N1		DUMP 210
	ASSIGN 400 TO N3		DUMP 211
	ASSIGN 42 TO N2		DUMP 212
	ASSIGN 42 TO N4		DUMP 213
	GO TO 99		DUMP 214
C			DUMP 215
C 20	CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT ARE		DUMP 216
C	LOST TO THE INVESTIGATION. THESE PARTICLES ARE IDENTIFIED BY A		DUMP 217
C	NEGATIVE FMAST() AND A TPC() WHICH EQUALS TLIMIT, THE TIME WHEN		DUMP 218
C	THE TRANSPORT OF PARTICLES IS TO CEASE. LOST PARTICLES ARE MERELY		DUMP 219
C	WRITTEN UNTO THE SYSTEM OUTPUT TAPE TO INFORM THE RESEARCHER OF		DUMP 220
C	THEIR REMOVAL FROM THE TRANSPORT.		DUMP 221
20	NLOST=NLOST-JTEST		DUMP 222
	NTAP=ISOUT		DUMP 223
	ASSIGN 500 TO N1		DUMP 224
	ASSIGN 42 TO N2		DUMP 225
	GO TO 99		DUMP 226
C			DUMP 227
C 21	CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT HAVE		DUMP 228
C	GONE BEYOND THE IN-CORE TOPOGRAPHY. THESE PARTICLES ARE IDENTI-		DUMP 229
C	FIED BY A POSITIVE FMAST() AND A NEGATIVE TPC(). THEY ARE		DUMP 230
C	WRITTEN UNTO THE OFF-TOPO TAPE (IOTOPO) BUT IF USE OF IOTOPO HAS		DUMP 231
C	BEEEN SUPPRESSED (BY SETTING IC(2)=1), THEY ARE WRITTEN UNTO THE		DUMP 232
C	SYSTEM OUTPUT TAPE INSTEAD. THIS IS TO LET THE RESEARCHER KNOW		DUMP 233
C	THAT HIS SUPPRESSION OF IOTOPO HAS LED TO A LOSS OF PARTICLES FROM		DUMP 234

C	THE TRANSPORT PROCESS.	DUMP 235
21	IF(IC(2)-1)211,212,211	DUMP 236
212	NTAP=ISOUT	DUMP 237
	GO TO 213	DUMP 238
C 211	JTOP1=1 INDICATES THAT THE ONLY OFF-TOPO PARTICLES THAT REMAIN IN	DUMP 239
C	THE TRANSPORT ARE THOSE THAT ARE IN CORE IN THE PARTICLE ARRAYS.	DUMP 240
C		DUMP 241
211	JTOP1=1	DUMP 242
	NTAP=IOTOPO	DUMP 243
213	NTU=NTU-JTEST	DUMP 244
	ASSIGN 300 TO N2	DUMP 245
	ASSIGN 100 TO N3	DUMP 246
	ASSIGN 42 TO N4	DUMP 247
	ASSIGN 42 TO N1	DUMP 248
	GO TO 99	DUMP 249
C		DUMP 250
C 22	CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT CAN	DUMP 251
C	NOT BE TRANSPORTED FURTHER UNTIL THE WIND FIELD IS UPDATED. THESE	DUMP 252
C	PARTICLES ARE IDENTIFIED BY A POSITIVE FMAST() AND A IP() EQUAL	DUMP 253
C	TO ENDLIM. NORMALLY THEY ARE WRITTEN ON TAPE IPAROT, BUT WHEN THE	DUMP 254
C	USER HAS SET IC(4)=1 TO SUPPRESS IPAROT, THEY ARE WRITTEN ON THE	DUMP 255
C	SYSTEM OUTPUT TAPE TO NOTIFY THE USER.	DUMP 256
22	IF(IC(4)-1)221,222,221	DUMP 257
222	NTAP=ISOUT	DUMP 258
	GO TO 223	DUMP 259
C		DUMP 260
C 221	JTIME1=1 INDICATES THAT THE ONLY OUT-OF-WIND PARTICLES THAT REMAIN	DUMP 261
C	IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS.	DUMP 262
221	JTIME1=1	DUMP 263
	NTAP=IPAROT	DUMP 264
223	NTI=NTI-JTEST	DUMP 265
	ASSIGN 600 TO N2	DUMP 266
	ASSIGN 42 TO N1	DUMP 267
	GO TO 99	DUMP 268
C		DUMP 269
C 23	CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT ARE	DUMP 270
C	BEYOND THE LIMITS OF THE WIND DATA CURRENTLY AVAILABLE IN CORE.	DUMP 271
C	THESE PARTICLES ARE IDENTIFIED BY A NEGATIVE FMAST() AND POSITIVE	DUMP 272
C	IP(). NORMALLY THEY ARE WRITTEN ON TAPE IOWIND, BUT WHEN THE	DUMP 273
C	USER HAS SET IC(3)=1 TO SUPPRESS IOWIND, THEY ARE WRITTEN ON THE	DUMP 274
C	SYSTEM OUTPUT TAPE TO NOTIFY THE USER.	DUMP 275
23	IF(IC(3)-1)231,232,231	DUMP 276
232	NTAP=ISOUT	DUMP 277
	GO TO 233	DUMP 278
C		DUMP 279
C 231	JWIND1=1 INDICATES THAT THE ONLY OUT-OF-WIND-FIELD PARTICLES THAT	DUMP 280
C	REMAIN IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS.	DUMP 281
231	JWIND1=1	DUMP 282
	NTAP=IOWIND	DUMP 283
233	NW=NW-JTEST	DUMP 284
	ASSIGN 300 TO N1	DUMP 285
	ASSIGN 100 TO N4	DUMP 286
	ASSIGN 42 TO N2	DUMP 287
	ASSIGN 42 TO N3	DUMP 288
C		DUMP 289
C 99	INITIALIZE FOR BLOCKING SORT	DUMP 290
99	IRSET=0	DUMP 291
	ICON=0	DUMP 292

JB=NALOFT	DUMP 293
JT=1	DUMP 294
J=JB	DUMP 295
C	DUMP 296
C WRITE OUT A DUMP SUMMARY	DUMP 297
WRITE (ISOUT,4)JTEST,JTEST1,NFREE,NG,NLOST,NTU,NTI,NW	DUMP 298
C NOW BEGIN THE BLOCKING SORT	DUMP 299
C 98 CLASSIFY THE JTH PARTICLE AS BLANK, TO BE DUMPED, OR NOT TO BE	DUMP 300
C DUMPED	DUMP 301
98 IF(FMAS(J))30,31,32	DUMP 302
30 GO TO N1,(300,500,42)	DUMP 303
32 GO TO N2,(42,300,500,600)	DUMP 304
300 IF(TP(J))33,33,33	DUMP 305
33 GO TO N3,(400,100,42)	DUMP 306
35 GO TO N4,(42,100)	DUMP 307
400 IF(IP(J)+ENDTIM)100,42,100	DUMP 308
500 IF(TP(J)-LIMIT) 42,100,42	DUMP 309
600 IF(TP(J)-ENDTIM)42,100,100	DUMP 310
C	DUMP 311
C 31 BLANK NOT TO BE DUMPED	DUMP 312
31 IF(ICON)422,901,424	DUMP 313
C 42 NON-BLANK NOT TO BE DUMPED	DUMP 314
42 IF(ICON)421,424,424	DUMP 315
C 100 TO BE DUMPED	DUMP 316
100 IF(ICON)903,9-4,900	DUMP 317
C	DUMP 318
C ICON=0 FIRST PASS	DUMP 319
C ICON=+1 BOTTOM LOOP	DUMP 320
C ICON=-1 TOP LOOP	DUMP 321
C	DUMP 322
C 900 MOVE THE JB-TH LINE TO THE BLANK LINE (JBL)	DUMP 323
900 XP(JBL)=XP(JB)	DUMP 324
YP(JBL)=YP(JB)	DUMP 325
ZP(JBL)=ZP(JB)	DUMP 326
TP(JBL)=TP(JB)	DUMP 327
PS(JBL)=PS(JB)	DUMP 328
FMAS(JBL)=FMAS(JB)	DUMP 329
JT=JT+1	DUMP 330
FMAS(JB)=0.0	DUMP 331
IF(JT-JTEST)901,901,1103	DUMP 332
901 JBL=JB	DUMP 333
ICON=-1	DUMP 334
J=JT	DUMP 335
902 JB=JB-1	DUMP 336
GO TO 98	DUMP 337
C 904 STORE THE JB-TH PARTICLE IN TEMPORARY STORAGE AND SET IRSET=1 TO	DUMP 338
C INDICATE THAT IT MUST BE PUT BACK INTO THE PARTICLE ARRAYS AT THE	DUMP 339
C END OF THIS DUMP OPERATION.	DUMP 340
904 XPT=XP(JB)	DUMP 341
YPT=YP(JB)	DUMP 342
ZPT=ZP(JB)	DUMP 343
TPT=TP(JB)	DUMP 344
PST=PS(JB)	DUMP 345
FMAS1=FMAS(JB)	DUMP 346
IRSET= 1	DUMP 347
FMAS(JB)=0.0	DUMP 348
GO TO 901	DUMP 349
903 JT=JT+1	DUMP 350

J=JT	DUMP 351
IF(JT-JTEST)98,98,110	DUMP 352
424 J=J-1	DUMP 353
JB=JB-1	DUMP 354
GO TO 1104	DUMP 355
C	DUMP 356
C 421 MOVE THE JT-TH LINE TO THE BLANK LINE (JBL)	DUMP 357
421 XP(JBL)=XP(JT)	DUMP 358
YP(JBL)=YP(JT)	DUMP 359
ZP(JBL)=ZP(JT)	DUMP 360
TP(JBL)=TP(JT)	DUMP 361
PS(JBL)=PS(JT)	DUMP 362
FMAS(JBL)=FMAS(JT)	DUMP 363
422 JBL=JT	DUMP 364
423 ICM=1	DUMP 365
J=JB	DUMP 366
1104 IF(JB-JTEST)110,110,98	DUMP 367
1103 JBL = JB	DUMP 368
C 110 IS THE TEMPORARY STORAGE LOADED. YES TO 1101	DUMP 369
110 IF(IRSET)1101,1102,1101	DUMP 370
C	DUMP 371
C 1101 REPLACE THE TEMPORARILY STORED PARTICLE IN THE BLANK LINE (JBL)	DUMP 372
1101 XP(JBL) = XPT	DUMP 373
YP(JBL) = YPT	DUMP 374
ZP(JBL) = ZPT	DUMP 375
TP(JBL) = TPT	DUMP 376
PS(JBL) = PST	DUMP 377
FMAS(JBL)=FMAST	DUMP 378
1102 CONTINUE	DUMP 379
C	DUMP 380
C RESET KEYS OF PARTICLES BEING DUMPED JUST BEFORE PRINTING OR	DUMP 381
C DUMPING THEM	DUMP 382
DO 131 J=1,JTEST	DUMP 383
IF(FMAS(J))101,111,111	DUMP 384
101 FMAS(J)=-FMAS(J)	DUMP 385
111 IF(TP(J))121,131,131	DUMP 386
121 TP(J)=-TP(J)	DUMP 387
131 CONTINUE	DUMP 388
C	DUMP 389
C	DUMP 390
C	DUMP 391
C NOW DUMP THE SELECTED DESCRIPTIONS	DUMP 392
C 50 IF THE SYSTEM OUTPUT TAPE IS TO BE WRITTEN, FIRST SELECT AND	DUMP 393
C WRITE AN APPROPRIATE TITLE.	DUMP 394
50 IF(INIAP-ISOUT)52,51,52	DUMP 395
C	DUMP 396
C IF THE PRINTING OF LOST PARTICLE DESCRIPTIONS IS TO BE SUPPRESSED,	DUMP 397
C GO TO 54	DUMP 398
51 IF(IC(8).NE.0) GO TO 54	DUMP 399
WRITE (ISOUT,1)JTEST	DUMP 400
WRITE (ISOUT,2)	DUMP 401
GO TO (511,516,513,514,515),JTEST	DUMP 402
511 WRITE (ISOUT,9)	DUMP 403
GO TO 516	DUMP 404
513 WRITE (ISOUT,6)	DUMP 405
GO TO 516	DUMP 406
514 WRITE (ISOUT,7)	DUMP 407
GO TO 516	DUMP 408

515	WRITE (ISOUT,8)	DUMP 409
516	WRITE (ISOUT,3)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)	DUMP 410
	GO TO 54	DUMP 411
52	WRITE (NTAP)JTEST	DUMP 412
	IF(NTAP-IPOUT)252,155,252	DUMP 413
155	WRITE (NTAP)(XP(J),YP(J), TP(J),PS(J),FMAS(J),J=1,JTEST)	DUMP 414
	GO TO 54	DUMP 415
252	WRITE (NTAP)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)	DUMP 416
	IF(IC(6)-1)54,2521,2521	DUMP 417
2521	WRITE (ISOUT,3)(XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)	DUMP 418
C		DUMP 419
C 54	ADD THE NUMBER OF LINES JUST DUMPED TO THE NUMBER OF LINES EMPTY	DUMP 420
C	PREVIOUSLY AND THEN ZERO OUT THE IDP OF THE LINES JUST DUMPED TO	DUMP 421
C	AVOID DOUBLE COUNTING	DUMP 422
54	NFREE=NFREE+JTEST	DUMP 423
	DO 541 J=1,JTEST	DUMP 424
541	FMAS(J)=0.0	DUMP 425
C		DUMP 426
	IF(NFREE-N)151,152,152	DUMP 427
C		DUMP 428
C 152	ARE THERE NOW ENOUGH CONTIGUOUS BLANK LINES IN THE TOP OF THE	DUMP 429
C	PARTICLE ARRAY TO RELIEVE THE N PARTICLES THAT ARE WAITING TO BE	DUMP 430
C	READ IN. YES TO 60	DUMP 431
152	IF(N-JTEST)60,60,154	DUMP 432
C		DUMP 433
C 154	CONSOLIDATE N BLANK LINES INTO THE TOP OF THE PARTICLE ARRAY	DUMP 434
154	JFR=NALOFT+1	DUMP 435
	K=JTEST+1	DUMP 436
	DO 56 J=K,N	DUMP 437
	IF(FMAS(J))57,56,57	DUMP 438
C 57	A PARTICLE MUST BE MOVED DOWN	DUMP 439
57	JFR=JFR-1	DUMP 440
	IF(FMAS(JFR))58,59,58	DUMP 441
58	IF(JFR-JTEST)60,60,57	DUMP 442
C 59	MOVE THE PARTICLE	DUMP 443
59	XP(JFR)=XP(J)	DUMP 444
	YP(JFR)=YP(J)	DUMP 445
	ZP(JFR)=ZP(J)	DUMP 446
	TP(JFR)=TP(J)	DUMP 447
	PS(JFR)=PS(J)	DUMP 448
	FMAS(JFR)=FMAS(J)	DUMP 449
	FMAS(J)=0.0	DUMP 450
56	CONTINUE	DUMP 451
60	RETURN	DUMP 452
	END	DUMP 453

454*

454 *

```

*IBFTC RDTOP LIST,DECK,M94/2                                RDT0
SUBROUTINE RDTOP0 (LB)                                       RDT0 1
C 11 OCT 66                                                  RDT0 2
C T. W. SCHWENKE TECHNICAL OPERATIONS RESEARCH SR OUTPRO CHAINRDT0 3
C THIS SUBROUTINE MERELY READS ONE TOPO BLOCK INTO ARRAYS J AND RDT0 4
C SUBSID. IT EXPECTS READ LIMITS TO BE IN COMMON WORDS II,JJ,KK. RDT0 5
C ERROR EXIT IF BAD LIMITS                                   RDT0 6
C                                                            RDT0 7
C *****                                                    RDT0 8
C                                                            RDT0 9
C COMMON /SET1/                                              RDT0 10
1 DIAM , DETID , IRISE , IEXEC , ISIN , ISOUT , RDT0 11
2 SU , SPAR , SSAM , TIME , TMP1 , TMP2 , RDT0 12
3 T2M , L , VPR , W , X , Z , RDT0 13
4 WHY , RMIN , IDISTR , SPAR1 , SPAR2 , SPAR3 , RDT0 14
5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 RDT0 15
DIMENSION DETID(12),WHY(40) RDT0 16
C                                                            RDT0 17
C *****                                                    RDT0 18
C                                                            RDT0 19
C COMMON /SET2/                                              RDT0 20
1 S , SUBSID , GRINT , BXLL , BXLU , BYLL RDT0 21
2, BYLU , TXLL , TXLU , TYLL , TYLU , XGZ RDT0 22
3, YGZ , INBLCK , HTOPO , TTOPU , ILIM , JLIM RDT0 23
4, KLIM , II , JJ , KK , XP , YP RDT0 24
5, ZP , FMAS , TP , PS , VX , VY RDT0 25
6, VZ , IL , JL , IBADD , WGRINT , NSTRAT RDT0 26
7, WLLX , WLLY , WORX , WORY , BOTHIT , IPARIN RDT0 27
8, IOTOPU , IOWIND , IHTOPU , IPOUT , IPAROT , JTOP1 RDT0 28
9, JWIND1 , IKKOR , TLIMIT , ENDTIM , IC , IBYPAS RDT0 29
1, JTOPU , NLOST , NG , NYO , NTI , NW RDT0 30
2, KALOFT , JTIME1 , NBMAX , NFREE , N , NCL RDT0 31
3, CRMAXY , CROFT , NCRITY , BZ , CRMINX , CRMINY RDT0 32
4, UC , SN , CS , NLOCIR , DTLOC , ATEMP RDT0 33
5, RHU , NA , IGZ , DTMAC , FROG , CRMAXX RDT0 34
6, ROPART RDT0 35
DIMENSION TOPOLM(4,4) ,NINTAR(4) ,ITOPLM(3,4) RDT0 36
DIMENSION S(1,10) ,SUBSID(400) ,IC(16) RDT0 37
DIMENSION XP(200) ,YP(200) ,ZP(200) ,FMAS(200) RDT0 38
DIMENSION TP(200) ,PS(200) ,ATEMP(260) ,RHU(260) RDT0 39
DIMENSION VX(1500) ,VY(1500) ,VZ(1500) ,IL(70) RDT0 40
DIMENSION JL(70) ,IBADD(70) ,WORX(70) RDT0 41
DIMENSION WGRINT(70) ,WLLX(70) ,WLLY(70) RDT0 42
DIMENSION WORY(70) ,BOTHIT(70) ,SN(6) ,CS(6) RDT0 43
DIMENSION CRMINX(6) ,CRMAXX(6) ,CRMINY(6) ,CRMAXY(6) RDT0 44
DIMENSION CROFT(6) ,NCRITY(6) ,JU(6) RDT0 45
C                                                            RDT0 46
C *****                                                    RDT0 47
C                                                            RDT0 48
C 9 FORMAT(33HOTOPO DATA TOO LARGE FOR PROGRAM.) RDT0 49
C 11 FORMAT(35H INCORRECT TOPO TABLE OF CONTENTS.) RDT0 50
C 100 FORMAT(10F10.3) RDT0 51
C                                                            RDT0 52
C *****                                                    RDT0 53
C *****                                                    RDT0 54
C *****                                                    RDT0 55
C                                                            RDT0 56
C II=ITOPLM(1,LB) RDT0 57
C JJ=ITOPLM(2,LB) RDT0 58
C KK=ITOPLM(3,LB) RDT0 59
C TTOPU=TOPOLM(4,LB) RDT0 60

```

```

      BXLL=TOPOLM(1,LB)
      BXLU=TOPOLM(3,LB)*FLOAT(II)+BXLL
      BYLL=TOPOLM(2,LB)
      BYLU=TOPOLM(3,LB)*FLOAT(JJ)+BYLL
      JFTOP0=LB+1
      IF(II)1,1,2
2     IF(JJ)1,1,3
3     IF(KK)1,4,4
4     IF(II-ILIM)5,5,6
5     IF(JJ-JLIM)7,7,6
7     IF(KK-KLIM)8,8,6
6     WRITE (ISOUT,9)
10    STOP
1     WRITE (ISOUT,11)
      GO TO 10
8     READ (IHTOP0)((S(1,J),I=1,I{}),J=1,JJ)
      READ (IHTOP0)(SUBSID(K),K=1,KK)
      WRITE (ISOUT,100)(SUBSID(K),K=1,KK)
      RETURN
      END

```

```

      RDTU 61
      RDTU 62
      RDTU 63
      RDTU 64
      RDTU 65
      RDTU 66
      RDTU 67
      RDTU 68
      RDTU 69
      RDTU 70
      RDTU 71
      RDTU 72
      RDTU 73
      RDTU 74
      RDTU 75
      RDTU 76
      RDTU 77
      RDTU 78
      RDTU 79
      RDTU 80

```

81*

81 *

```

$IBFTC LNK5 LIST,DECK,M94/2 LNK5
SUBROUTINE LNK5 LNK5 1
C T.W.SCHWENKE TECHNICAL OPERATIONS RESEARCH LINK 5 LNK5 2
C 15 OCTOBER 1966 LNK5 3
C FIRST OVERLAY LINK OF TRANSPORT PROGRAM. INITIALIZATION AND CALL LNK5 4
C OF WIND FIELD PREPARING SUBROUTINE MKWIND LNK5 5
C LNK5 6
C ***** LNK5 7
C LNK5 8
C ***** TAPE IDENTIFICATIONS AND ASSIGNMENTS ***** LNK5 9
C LNK5 10
C NAME CONTENT LNK5 11
C LNK5 12
C INTOPO TOPOGRAPHIC HEIGHT DATA TAPE LNK5 13
C IUTOPO PARTICLES ALOFT BUT BEYOND IN CORE TOPOGRAPHY LNK5 14
C IOWIND PARTICLES ALOFT BUT BEYOND IN CORE WIND FIELD LNK5 15
C IPARIN PARTICLES ALOFT AND TO BE PROCESSED LNK5 16
C IPARUT PARTICLES ALOFT AWAITING NEXT TIME PERIOD WINDS LNK5 17
C IPOUT PROGRAM SPECIFIC OUTPUT LNK5 18
C ISOUT SYSTEM (BCD) OUTPUT TAPE LNK5 19
C ISIN SYSTEM INPUT TAPE LNK5 20
C LNK5 21
C ***** PROGRAM GLOSSARY ***** LNK5 22
C LNK5 23
C ATEMP(J) ATMOSPHERIC VISCOSITY IN THE J-TH STRATUM LNK5 24
C AX,AY,AZ OUTPUT ARGUMENTS OF LOCAL CIRCULATION SYSTEM LNK5 25
C CODES. X,Y,AND Z COMPONENTS OF WIND AT THE LNK5 26
C POSITION OF THE J-TH PARTICLE. LNK5 27
C BLANK BLANK LITERAL ( ) LNK5 28
C BOTHIT(K) ALTITUDE OF BOTTOM OF KTH WIND DATA BLOCK LNK5 29
C BZ LENGTH OF THE SIDE OF THE SQUARE CLOUD SUBDIVIS- LNK5 30
C IONS AT THE TIME OF THEIR DEFINITION LNK5 31
C CBREZ1 SUBROUTINE SEA BREEZE CIRCULATION MODEL LNK5 32
C CIRMIN TIME UNTIL INTERSECTION WITH THE FIRST LOCAL LNK5 33
C WIND SYSTEM LNK5 34
C CIRKYP(J) THE LOCAL CIRCULATION TYPE OF THE JTH SYSTEM LNK5 35
C 1 MOUNTAIN WIND 1 LNK5 36
C 2 RIDGE WIND 1 LNK5 37
C 3 SEA BREEZE 1 LNK5 38
C 4 SEA BREEZE 2 LNK5 39
C 5 NOT ASSIGNED LNK5 40
C CRID( ) CLOUD RISE IDENTIFICATION LNK5 41
C CRMINX(J) SMALLEST X COORDINATE OF THE JTH LOCAL SYSTEM LNK5 42
C CRMXX(J) LARGEST X COORDINATE OF THE JTH LOCAL SYSTEM LNK5 43
C CRMINY(J) SMALLEST Y COORDINATE OF THE JTH LOCAL SYSTEM LNK5 44
C CRMXY(J) LARGEST Y COORDINATE OF THE JTH LOCAL SYSTEM LNK5 45
C CROHT(K) HEIGHT OF TOP SURFACE OF THE KTH LOCAL CIRCULATION LNK5 46
C SYSTEM CELL. LNK5 47
C DETID(J) INITIAL CONDITIONS (FIREBALL) IDENTIFICATION LNK5 48
C DENTI TOPOGRAPHY TAPE IDENTIFICATION LITERAL (INTOPO) LNK5 49
C AS READ FROM TAPE LNK5 50
C DENTT PARTICLES ALOFT INPUT TAPE IDENTIFICATION LNK5 51
C LITERAL AS READ FROM TAPE LNK5 52
C DTLOC TIME INCREMENT FOR USE IN TRANSPORT WITHIN LOCAL LNK5 53
C CIRCULATION SYSTEM CELLS LNK5 54
C DTMAC TIME INCREMENT FOR USE IN TRANSPORT WITHIN THE LNK5 55
C MACRO FIELD BUT BELOW MAXIMUM TOPOGRAPHIC HEIGHT LNK5 56
C DTST CANNED COPY OF PARTICLES ALOFT INPUT TAPE LNK5 57
C IDENTIFICATION LITERAL (IPARIN) LNK5 58
C DUMPP SUBROUTINE SELECTS AND DUMPS ONE OR MORE LNK5 59
C SETS OF PARTICLE DESCRIPTIONS ONTO TEMPORARY,OR LNK5 60

```

C		INTERMEDIATE OUTPUT TAPES.	LNK5	61
C	ENDTIM	TIME UP TO WHICH THE CURRENT WIND FIELD DESCRIPTION	LNK5	62
C		ION IS ASSUMED TO BE VALID.	LNK5	63
C	EPSIL	A SMALL NUMBER	LNK5	64
C	ERROR	SUBROUTINE ERROR COMMENT AND/OR STOP	LNK5	65
C	FALRAT	SUBROUTINE PARTICLE SETTLING RATE COMPUTATION	LNK5	66
C	FMAS(J)	MASS PER UNIT AREA (MKS) ASSOCIATED WITH THE	LNK5	67
C		J-TH CLOUD SUBDIVISION (PARTICLE) AT THE TIME OF	LNK5	68
C		ITS DEFINITION	LNK5	69
C	FRUG	A PRECOMPUTED CONSTANT INPUT FOR SUBROUTINE	LNK5	70
C		FALRAT	LNK5	71
C	FV	OUTPUT ARGUMENT OF FALRAT (MKS) A POSITIVE	LNK5	72
C		QUANTITY	LNK5	73
C	FW	FISSION YIELD	LNK5	74
C	GETWND	SUBROUTINE: GETS MACRO WIND VECTORS AT PARTICLE	LNK5	75
C		POSITION	LNK5	76
C	H	OUTPUT ARGUMENT OF SUBROUTINE HEIGHT	LNK5	77
C	HEIGHT	SUBROUTINE RETRIEVES TOPO HEIGHT AT X,Y	LNK5	78
C	HOB	HEIGHT OF BURST RELATIVE TO SURFACE HEIGHT	LNK5	79
C	HS	TEMPORARY STORAGE FOR STANDARD ATMOSPHERIC	LNK5	80
C		STRATUM BOTTOM HEIGHT	LNK5	81
C	HTOPU	HIGHEST TOPO HEIGHT ANYWHERE ON THE TOPO TAPE	LNK5	82
C	HTST	CANNED COPY OF TOPOGRAPHY TAPE IDENTIFICATION	LNK5	83
C		LITERAL (HTOPU)	LNK5	84
C	IBADD(K)	BASE ADDRESS OF KTH WIND DATA BLOCK IN ARRAYS	LNK5	85
C		VX, VY, AND VZ	LNK5	86
C	IBYPAS	INITIALIZE BYPASS FLAG	LNK5	87
C			LNK5	88
C	IC(J)	CONTROL VARIABLES INTERPRETATIONS AS FOLLOWS	LNK5	89
C	1	SUPPRESS TOPO TAPE USAGE IF = 1	LNK5	90
C	2	SUPPRESS OFF TOPO TAPE USAGE IF = 1	LNK5	91
C	3	SUPPRESS OUT OF WIND FIELD TAPE USAGE IF = 1	LNK5	92
C	4	SUPPRESS PARTICLES ALOFT SECONDARY MEMORY IF = 1	LNK5	93
C	5	SUPPRESS ALL TAPE SECONDARY MEMORY	LNK5	94
C	6	TRANSPORT TRACE CONTROL. 0 = NO TRACES,	LNK5	95
C		1 = A PARTIAL TRACE, 2 = A MORE COMPLETE TRACE	LNK5	96
C	7	PRINT WIND FIELD IF = 1	LNK5	97
C	8	SUPPRESS PRINT AND WRITE OF LOST PARTICLES IF = 1	LNK5	98
C			LNK5	99
C			LNK5	100
C	IEXEC	EXECUTIVE CONTROL WORD TO CONTROL BRANCHING	LNK5	101
C		BETWEEN CHAIN LINK SUBROUTINES	LNK5	102
C	IF	INDEX OF THE LAST LINE IN THE PARTICLE ARRAYS	LNK5	103
C	II	ITUPLM(1,J) FOR THE TOPO DATA BLOCK CURRENTLY IN	LNK5	104
C		CORE	LNK5	105
C	IL(K)	LIMITS ON INDICES OF WIND BLOCK DATA SET K	LNK5	106
C	ILIM		LNK5	107
C	IPAS	ASSIGNED GO TO VARIABLE FOR USE IN DEALING WITH	LNK5	108
C		EXCESSIVELY SMALL PARTICLE MOVEMENTS ARISING	LNK5	109
C		DURING TRANSPORT	LNK5	110
C	IR	INDICATOR OF TYPE OF BOUNDING PLANE ENCOUNTERED	LNK5	111
C		DURING TRANSPORT. 1=X-BOUNDARY, 2=Y-BOUNDARY,	LNK5	112
C		3=Z-BOUNDARY, 4= TIME BOUNDARY, 5= LOCAL CIRC-	LNK5	113
C		ULATION CELL BOUNDARY.	LNK5	114
C	IRRUR	NUMBER OF THE SOURCE STATEMENT NEAREST TO WHERE	LNK5	115
C		AN ERROR CONDITION WAS SENSED.	LNK5	116
C		THAT IS TO BE PROCESSED BY THE TRANSPORT LOOP	LNK5	117
C	IT	ASSIGNED GO TO VARIABLE FOR USE IN CODE SELECT-	LNK5	118

C		ION REGARDING USE OF PLANAR OR PIECEWISE PLANAR	LNK5	119
C		TOPOGRAPHIC DESCRIPTION.	LNK5	120
C	ITUPLM(1,J)	NUMBER OF CELLS IN THE X DIRECTION OF THE	LNK5	121
C		REGULAR GRID SECTION OF THE J-TH TOPO DATA BLOCK	LNK5	122
C	ITUPLM(2,J)	SEE ITUPLM(1,J) BUT FOR Y-DIRECTION	LNK5	123
C	ITUPLM(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	JFTOPU	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	ITUPLM(2,J) FOR THE TOPO DATA BLOCK CURRENTLY IN	LNK5	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 TRANSPORT	LNK5	143
C		LOOP WITHOUT FIRST READING MORE PARTICLES FROM	LNK5	144
C		TAPE. THIS ALL TRANSPORTABLE 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	0	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	JTOPU	INDEX OF HIGHEST WIND LAYER. SET BY MAXWIND 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 THE	LNK5	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 PRECEDING 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	ITUPLM(3,J) FOR THE TOPO DATA BLOCK CURRENTLY IN	LNK5	168
C		CORE	LNK5	169
C	LOTRAN	SUBROUTINE LOCAL SYSTEM PARTICLE TRANSPORT	LNK5	170
C		CODE	LNK5	171
C	MAXTOPO	MAX. NO. TOPO BLOCK CAPABILITY OF PROGRAM	LNK5	172
C	MPINT	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 READ	LNK5 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	NBLOCK	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	NINTAK(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 ACCOMMODATE	LNK5 201
C	NT1	NUMBER OF PARTICLES AT TIME BARRIER (IN CORE)	LNK5 202
C	NT0	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	PROGRAM	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 DISTRIBUTION	LNK5 226
C	SLDTMP	PARTICLE SOLIDIFICATION 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 (TY1,TY2)	LNK5	237
C			LNK5	236
C	TMINX,TMINY	TEMPORARY STORAGE FOR SMALLEST OF (TX1,TX2) AND (TY1,TY2) RESPECTIVELY	LNK5	239
C			LNK5	240
C	TMSD	TIME OF SOLIDIFICATION	LNK5	241
C	TOPID(J)	TOPOGRAPHY IDENTIFIER	LNK5	242
C	TOPULM	FLOATING POINT TOPOGRAPHIC DATA ARRAY	LNK5	243
C	TOPULM(1,J)	MIN. X-COORDINATE FOR J-TH TOPO DATA BLOCK	LNK5	244
C	TOPULM(2,J)	MIN. Y-COORDINATE FOR J-TH TOPO DATA BLOCK	LNK5	245
C	TOPULM(3,J)	INITIAL (REGULAR) GRID INTERVAL FOR J-TH TOPO DATA BLOCK	LNK5	246
C			LNK5	247
C	TOPULM(4,J)	HIGHEST TOPO HEIGHT IN THE J-TH TOPO DATA BLOCK	LNK5	248
C	TP	PARTICLE TIME STATUS	LNK5	249
C	TS	TIME OF FLIGHT TO POINT OF ENTRANCE TO CURRENT LOCAL WIND CELL	LNK5	250
C			LNK5	251
C	TTOPU	HIGHEST TOPO ELEVATION IN THE IN CORE DATA	LNK5	252
C	TW	TOTAL YIELD	LNK5	253
C	TWIND	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 THE	LNK5	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	VPA, 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 CONTAINING PARTICLE	LNK5	269
C			LNK5	270
C	XGZ, YGZ	COORDINATES OF GROUND ZERO	LNK5	271
C	XIN, YIN, ZIN	X, Y, AND Z INCREMENTAL DISTANCES FOR USE IN CONSTANT TIME INTERVAL STEPPING BELOW MAXIMUM TOPO HEIGHT.	LNK5	272
C			LNK5	273
C			LNK5	274
C	XP, YP, ZP	PARTICLE POSITION COORDINATES	LNK5	275
C	XX, YY, ZZ	TEMPORARY STORAGE FOR X, Y, AND Z PARTICLE COORDINATES	LNK5	276
C			LNK5	277
C	YBL, YBU	SEE XBL. FOR Y COORDINATE	LNK5	278
C			LNK5	279
C	*****		LNK5	280
C	COMMON /SET1/		LNK5	281
C	1 DIAM, DETID(12), IRISE, IEAEC, ISIN, ISOUT		LNK5	282
C	2 SD, SPAR, SPAM, TME, TMP1, TMP2		LNK5	283
C	3 T2M, U, VPR, W, X, Z		LNK5	284
C	4 WHY(40), RMIN, IDISIR, SPAR1, SPAR2, JDONE		LNK5	285
C	5 SPAR4, SPAR5, SPAR6, SPAR7, SPAR8, SPAR9		LNK5	286
C			LNK5	287
C	*****		LNK5	288
C			LNK5	289
C	COMMON /SET2/		LNK5	290
C	1 S, SUBSID, GRINT, BXLL, BXLU, BYLL		LNK5	291
C	2, BYLU, TXLL, TXLU, TYLL, TYLU, XGZ		LNK5	292

```

3,      YGZ      , NBLCK , HTOPO , TTOPU , ILIM , JLIM      LNK5 293
4,      KLIM     , II    , JJ    , KK    , XP    , YP      LNK5 294
5,      ZP       , FMAS  , TP    , PS    , VX    , VY      LNK5 295
6,      VZ       , IL    , JL    , IBADD , WGRINT , NSTRAT  LNK5 296
7,      WLLX     , WLLY  , WJRX , WURY  , BOTHIT , IPARIN  LNK5 297
8,      IOTOPU   , IOWIND , IHIOPU , IPOUT , IPAROT , JTOP1 !  LNK5 298
9,      JWIND1   , IRROR  , TLIMIT , ENDTIM , IC    , IBYPAS  LNK5 299
1,      JTOPJ    , NLOST  , NG    , NIO   , NTI   , NW      LNK5 300
2,      NALOFT   , JTIME1 , NBMAX , NFREE , N      , NCL     LNK5 301
3,      CRMAXY   , CROHI  , NCR1YP , BZ    , CRMINX , CRMINY  LNK5 302
4,      CU       , SN     , CS    , NLOCIR , DTLOC , ATEMP   LNK5 303
5,      RHU      , NA     , TGZ   , DTMAC , PROG  , CRMAXX  LNK5 304
6,      RUPART   ,         ,        ,        ,        ,        LNK5 305
      DIMENSION TOPULM(4,4) , NINTAR(4) , ITOPLM(3,4)      LNK5 306
      DIMENSION S(10,10) , SUBSID(400) , IC(18)          LNK5 307
      DIMENSION XP(200) , YP(200) , ZP(200) , FMAS(200)   LNK5 308
      DIMENSION TP(200) , PS(200) , ATEMP(260) , RHU(260)  LNK5 309
      DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70)   LNK5 310
      DIMENSION JL(70) , IBADD(70) , WJRX(70)            LNK5 311
      DIMENSION WGRINT(70) , WLLX(70) , WLLY(70)          LNK5 312
      DIMENSION WURY(70) , BOTHIT(70) , SN(6) , CS(6)      LNK5 313
      DIMENSION CRMINX(6) , CRMAXX(6) , CRMINY(6) , CRMAXY(6) LNK5 314
      DIMENSION CROHI(6) , NCR1YP(6) , CU(6)             LNK5 315
C ***** LNK5 316
C ***** LNK5 317
C ***** LNK5 318
      DIMENSION CRID(12) , PSEID(12) , TID(12)           LNK5 319
      DIMENSION WID(12) , TUPID(12)                     LNK5 320
1  FORMAT(12A6) LNK5 321
2  FORMAT(///25X,56H**** INITIAL CONDITIONS (FIREBALL) IDENTIFICATION LNK5 322
1N ****/25X,12A6,///25X,57H**** CLOUD RISE IDENTIFICATION ****/25 LNK5 323
2X,12A6,///25X,49H**** PARTICLE SET EXPANSION IDENTIFICATION **** LNK5 324
3  /25X,12A6,///25X,85H**** THIS RUN OF THE TRANSPORT MODULE WAS LNK5 325
4GIVEN THE FOLLOWING IDENTIFICATION ****/25X,12A6,///25X,28H**** LNK5 326
5THEIR INPUT DATA ****) LNK5 327
3  FORMAT(18X,12A6) LNK5 328
4  FORMAT(15) LNK5 329
5  FORMAT(2E12.5) LNK5 330
6  FORMAT(1H1,24X,48HATMOSPHERIC PROPERTIES FOR FALL RATE CALCULATION LNK5 331
1//25X16HHEIGHT OF BOTTOM4X,9HVISCOSITY12X,7HDENSITY/26X10HUF STRAI LNK5 332
2UM/25X16HMETERS ABOVE MSL6X5H(MKS)15X5H(MKS)/) LNK5 333
7  FORMAT(/15X71HTHE CONTROL VARIABLE ARRAY, IC(J), HAS BEEN GIVEN THE LNK5 334
1E FOLLOWING VALUES.) LNK5 335
8  FORMAT(15X,18I4) LNK5 336
9  FORMAT(/15X28HTHE TRANSPORT TIME LIMIT IS F12.3) LNK5 337
10 FORMAT(18X63H IN THIS RUN WE ASSUME A PLANAR DEPOSITION SURFACE AT LNK5 338
1 ELEVATIONF10.3) LNK5 339
11 FORMAT(42HOPARTICLES REMAINING ON TIME BOUNDARY TAPE) LNK5 340
12 FORMAT(6(1X,E13.6)) LNK5 341
13 FORMAT(A6,4F10.3,13) LNK5 342
14 FORMAT(29H0 WRONG TAPE REEL ON DRIVE 12) LNK5 343
15 FORMAT(42HOPLEASE MOUNT CORRECT TAPE AND PRESS START) LNK5 344
16 FORMAT(/18X,35HIDENTIFICATION FROM TOPOGRAPHY TAPE18X12A6) LNK5 345
17 FORMAT(25X,E13.5,5X,E13.5,6X,E13.5) LNK5 346
18 FORMAT(58HUTRANSPCR1 IS COMPLETED. INTERMEDIATE RESULTS ARE ON TAPE LNK5 347
1E 12) LNK5 348
19 FORMAT(44HOPLEASE FILE PROTECT THE REEL ON TAPE DRIVE 12,25H AT TLNK5 349
THE END OF THIS RUN.) LNK5 350

```

```

20  FORMAT(6F10.5)                                LNK5 351
21  FORMAT(18I4)                                    LNK5 352
22  FORMAT(A6,I3,4E12.5,I5)                        LNK5 353
23  FORMAT(1H1///51X19H* * * * * * * *///12X101HT H E   D E P A R T LNK5 354
    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 LNK5 355
    2 N   S Y S T E M,///51X,19H* * * * * * * *///52X,16HTRANSPØRT LNK5 356
    3MØDULE///55X,11HPREPARED BY/43X,34HTECHNICAL ØPERATIØNS RESEARCH,ILNK5 357
    4NC./52X,17HBURLINGTON, MASS.///29X,63H**** SUMMARY ØF INPUT IDENLNK5 358
    5TIFIERS AND INITIAL CØNDITIØNS ****)          LNK5 359

24  FORMAT(///15X16HTØPØGRAPHIC DATA)            LNK5 360
27  FORMAT(///15X13HPARTICLE DATA/18X28HDENSITY ØF FALLØUT PARTICLESF2ØLNK5 361
    1.3,2X,7HKG/M**3)                              LNK5 362
28  FORMAT(///15X9HWIND DATA/)                    LNK5 363
29  FORMAT(18X,A6,1X,I6,4(1X,E13.5),1X,I10)        LNK5 364
30  FORMAT(18X,A6,4(1X,F13.5),I6)                  LNK5 365
31  FORMAT(6F12.3)                                  LNK5 366
C                                     LNK5 367
C *****LNK5 368
C                                     LNK5 369
    DATA HTST,DTST,BLANK,PØUT,ENDWFD/6HIHTØPØ,6HIPARIN,6H      ,6HIPØULNK5 370
    1T ,6HEND ØF/                                             LNK5 371
C                                     LNK5 372
C *****LNK5 373
C *****LNK5 374
C                                     LNK5 375
C THIS BYPASSES INITIALIZATIØN CØDING AFTER THE FIRST PASS    LNK5 376
    NUL=Ø                                                    LNK5 377
    IF (IBYPAS-918273)2Ø1,2ØØ,2Ø1                          LNK5 378
2Ø1 IBYPAS=918273                                           LNK5 379
C INITIALIZE                                                 LNK5 380
    JØØNE=Ø                                                  LNK5 381
    IPARIN=11                                                LNK5 382
    IØTØPØ=4                                                 LNK5 383
    IØWIND=3                                                 LNK5 384
    IHTØPØ=1Ø                                                LNK5 385
    IPØUT= 9                                                 LNK5 386
    IPARØT=1                                                 LNK5 387
    JTØP1=Ø                                                  LNK5 388
    JWIND1=Ø                                                 LNK5 389
    JTIME1=Ø                                                 LNK5 390
    ENDTIM=Ø.Ø                                               LNK5 391
    JFTØPØ=1                                                 LNK5 392
    MXTØPØ=4                                                 LNK5 393
    DTMAC=1Ø.                                                LNK5 394
    DTLØC=1Ø.                                                LNK5 395
    NALØFT=2ØØ                                              LNK5 396
    NBMAX=15Ø                                               LNK5 397
    NFREE=NALØFT                                           LNK5 398
    NLØST=Ø                                                 LNK5 399
    NSTRAT =7Ø                                              LNK5 4ØØ
    NW=Ø                                                    LNK5 4Ø1
    NTØ=Ø                                                  LNK5 4Ø2
    NG=Ø                                                    LNK5 4Ø3
C                                     LNK5 4Ø4
C ILIM,JLIM,KLIM,ARE LIMITS ØN TØPØ ARRAYS. SEE DIMENSION.  LNK5 4Ø5
    ILIM=1Ø                                                LNK5 4Ø6
    JLIM=1Ø                                                LNK5 4Ø7
    KLIM=4ØØ                                              LNK5 4Ø8

```

DO 2011 J=1,NALOFT	LNK5 409
2011 FMAS(J)=0.0	LNK5 410
C	LNK5 411
C READ IDENTIFICATION FOR TRANSPORT	LNK5 412
READ (ISIN,1)(TID(J),J=1,12)	LNK5 413
C	LNK5 414
C READ CONTROL DATA FOR TRANSPORT	LNK5 415
C THESE CONTROL PARAMETERS ARE FOR USE AS SIMPLIFYING SWITCHES	LNK5 416
READ (ISIN,21)(IC(J),J=1,18)	LNK5 417
READ (ISIN,31)TLIMIT	LNK5 418
C	LNK5 419
C REWIND ALL TAPES INVOLVED IN TRANSPORT	LNK5 420
IF(IC(1)-1)150,151,151	LNK5 421
150 REWIND IHTOPO	LNK5 422
151 IF(IC(2)-1)152,153,153	LNK5 423
152 REWIND IOTOPO	LNK5 424
153 IF(IC(3)-1)154,155,155	LNK5 425
154 REWIND IOWIND	LNK5 426
155 IF(IC(4)-1)156,157,157	LNK5 427
156 REWIND IPAROT	LNK5 428
157 CONTINUE	LNK5 429
REWIND IPARIN	LNK5 430
REWIND IPOUT	LNK5 431
C	LNK5 432
C CHECK IDENTIFICATIONS ON TOPO AND PARTICLE INPUT TAPES	LNK5 433
206 IF(IC(1)-1)158,203,203	LNK5 434
158 READ (IHTOPO)DENTI	LNK5 435
RTST=AND(DENTI,COMPL(HTST))	LNK5 436
IF(RTST)202,2031,202	LNK5 437
C	LNK5 438
C 202 WRONG TAPE AS IHTOPO	LNK5 439
202 PRINT 14,IHTOPO	LNK5 440
WRITE (ISOUT,14)IHTOPO	LNK5 441
PRINT 15	LNK5 442
REWIND IHTOPO	LNK5 443
PAUSE	LNK5 444
REWIND IHTOPO	LNK5 445
GO TO 206	LNK5 446
C	LNK5 447
C 204 WRONG TAPE AS IPARIN	LNK5 448
204 PRINT 14,IPARIN	LNK5 449
WRITE (ISOUT,14)IPARIN	LNK5 450
PRINT 15	LNK5 451
REWIND IPARIN	LNK5 452
PAUSE	LNK5 453
REWIND IPARIN	LNK5 454
GO TO 207	LNK5 455
2031 READ(IHTOPO)TXLL,TXLU,TYLL,TYLU,NBLCK	LNK5 456
C	LNK5 457
203 CONTINUE	LNK5 458
207 READ (IPARIN)DENTT	LNK5 459
RTST=AND(DENTT,COMPL(DTST))	LNK5 460
IF(RTST)204,208,204	LNK5 461
C	LNK5 462
C 208 READ ARBITRARY 72 CHARACTER FIREBALL,CLOUD-RISE,AND PARTICLE	LNK5 463
C ACTIVITY IDENTIFICATIONS FROM IPARIN	LNK5 464
208 READ (IPARIN) FW,SSAM,SLDTMP,TMSD,SIGMA,TW,HOB,NSP,XGZ,YGZ,TGZ,BZ,	LNK5 465
1 NCL,RADMAX	LNK5 466

	READ (IPARIN)(PSEID(J),J=1,12)	LNK5 467
	READ (IPARIN)(CRID(J),J=1,12)	LNK5 468
	READ (IPARIN)(DETID(J),J=1,12)	LNK5 469
C		LNK5 470
C	READ DENSITY OF FALLOUT PARTICLES	LNK5 471
C	ROPART IS PARTICLE DENSITY IN KILOGRAMS PER CUBIC METER	LNK5 472
	READ (IPARIN)ROPART	LNK5 473
C		LNK5 474
C	READ PARTICLE SIZE MASS AND ACTIVITY DISTRIBUTIONS	LNK5 475
	READ (IPARIN)NPS	LNK5 476
C		LNK5 477
C	VX() IS USED TO TEMPORARILY STORE THE SURFACE TO VOLUME RATIO	LNK5 478
C	ARRAY SV	LNK5 479
C	VY() IS USED TO TEMPORARILY STORE THE A ARRAY FROM PSE (LINK4)	LNK5 480
C	VZ() IS USED TO TEMPORARILY STORE THE PACT ARRAY FROM PSE (LINK4)	LNK5 481
	READ (IPARIN)(PS(1),VY(1),VZ(1),VX(1),I=1,NPS)	LNK5 482
C		LNK5 483
C	READ ATMOSPHERIC DENSITY AND VISCOSITY	LNK5 484
C	A TABLE OF ATMOSPHERIC VISCOSITY (ATEMP(J)) AND DENSITY (RHU(J))	LNK5 485
C	STATED IN THE MKS SYSTEM FOR 200 METER STRATA STARTING FROM 1100	LNK5 486
C	METERS BELOW MSL	LNK5 487
	READ (IPARIN)NA	LNK5 488
	READ (IPARIN)(ATEMP(J),RHU(J),J=1,NA)	LNK5 489
C		LNK5 490
C	COMPUTE CONSTANT FOR FALL RATE CALCULATIONS	LNK5 491
	FRUG=1.3066667E-17*ROPART	LNK5 492
C		LNK5 493
C	READ ARBITRARY TOPO IDENTIFICATION	LNK5 494
	IF(IC(1)-1)159,160,160	LNK5 495
160	READ (ISIN,20)TTUPO	LNK5 496
	GO TO 205	LNK5 497
159	READ (IHTUPO)(TOPID(J),J=1,12)	LNK5 498
C		LNK5 499
C	READ TOPO TABLE OF CONTENTS	LNK5 500
	READ (IHTUPO)TOPULM	LNK5 501
	READ (IHTUPO)ITOPULM	LNK5 502
C		LNK5 503
C	FIND HIGHEST TOPO HEIGHT	LNK5 504
	HTOPO=0.0	LNK5 505
	DO 170 J=1,NBLCK	LNK5 506
	IF(HTOPO-ITOPULM(4,J))171,170,170	LNK5 507
171	HTOPO=ITOPULM(4,J)	LNK5 508
170	CONTINUE	LNK5 509
C		LNK5 510
C	READ FIRST TOPO DATA BLOCK	LNK5 511
	CALL RDTUPO (1)	LNK5 512
C		LNK5 513
C 205	POT AN IDENTIFICATION ON THE TRANSPORT INTERMEDIATE OUTPUT TAPE	LNK5 514
205	READ (ISIN,1)(WID(J),J=1,12)	LNK5 515
	WRITE (IPOUT)POUT	LNK5 516
	WRITE(IPOUT) FW,SSAM,SEDTMP,IMSD,SIGMA,FW,HUB,NCL,FLIMIT,0Z,	LNK5 517
1	ROPART,XGZ,YGZ,IC2,KADMAX	LNK5 518
	WRITE (IPOUT) (DETID(J),J=1,12),(CRID(J),J=1,12),(PSEID(J),J=1,12)	LNK5 519
1,	(IID(J),J=1,12),(WID(J),J=1,12)	LNK5 520
	WRITE (IPOUT)NPS	LNK5 521
	WRITE (IPOUT)(PS(J),VY(J),VZ(J),VX(J),J=1,NPS)	LNK5 522
	IF(IC(1)-1)2054,2055,2054	LNK5 523
2055	CONTINUE	LNK5 524

WRITE (IPOUT) (BLANK,J=1,12)	LNK5 525
GO TO 2056	LNK5 526
2054 WRITE (IPOUT) (TOPID(J),J=1,12)	LNK5 527
C PRINT TRANSPORT OUTPUT HEADING	LNK5 528
C 2056 WRITE (ISOUT,23)	LNK5 529
WRITE (ISOUT,2) (DETID(J),J=1,12),(CRID(J),J=1,12),(PSID(J),J=1,12),	LNK5 531
12),(TID(J),J=1,12)	LNK5 532
WRITE (ISOUT,7)	LNK5 533
WRITE (ISOUT,8) (IC(J),J=1,18)	LNK5 534
WRITE (ISOUT,9) TLIMIT	LNK5 535
WRITE (ISOUT,27) ROPART	LNK5 536
WRITE (ISOUT,29) DENI1,NSP,X3Z,YGZ,IGZ,BZ,NCL	LNK5 537
WRITE (ISOUT,24)	LNK5 538
IF (IC(1)-1) 2051,2052,2051	LNK5 539
2052 WRITE (ISOUT,10) TTOP	LNK5 540
GO TO 2053	LNK5 541
2051 WRITE (ISOUT,16) (TOPID(J),J=1,12)	LNK5 542
WRITE (ISOUT,30) DENI1,IXLL,FXLU,ITYL,ITYLU,NBLCK	LNK5 543
2053 WRITE (ISOUT,28)	LNK5 544
WRITE (ISOUT,3) (WID(J),J=1,12)	LNK5 545
WRITE (ISOUT,6)	LNK5 546
HS=-1100.0	LNK5 547
DO 2057 J=1,NA	LNK5 548
WRITE (ISOUT,17) HS,ATEMP(J),KHO(J)	LNK5 549
2057 HS=HS+200.0	LNK5 550
C *****	LNK5 551
C *****	LNK5 552
C 200 ANY MORE TIME INTERVALS TO BE DEALT WITH. NO TO 500	LNK5 553
200 IF (TLIMIT-ENDTIM) 500,500,400	LNK5 554
C 500 MAKE FINAL TRANSPORT PROGRAM OUTPUT AND COMMENTS	LNK5 555
C SET N=NALOFT TO CAUSE DUMPP TO CLEAR OUT THE ENTIRE PARTICLE AREA	LNK5 556
C 500 N=NALOFT	LNK5 557
CALL DUMPP	LNK5 558
C ARE ANY PARTICLES ON THE TIME BOUNDARY TAPE. YES TO 700	LNK5 559
C ***** TEMP *****	LNK5 560
C JTIME1=0	LNK5 561
IF (JTIME1) 501,501,700	LNK5 562
C 700 PRINT ANY PARTICLE DESCRIPTIONS THAT REMAIN ON THE TIME BOUNDARY	LNK5 563
C OVERFLOW TAPE, IPAROT	LNK5 564
700 WRITE (IPAROT) NUL	LNK5 565
REWIND IPAROT	LNK5 566
WRITE (ISOUT,11)	LNK5 567
READ (IPAROT) N	LNK5 568
IF (N) 501,501,701	LNK5 569
701 READ (IPAROT) (XP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,N)	LNK5 570
WRITE (ISOUT,12) (XP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,N)	LNK5 571
GO TO 702	LNK5 572
C 501 WRITE A FINAL ZERO BLOCK COUNT AND EOF ON IPOUT	LNK5 573
501 WRITE (IPOUT) NUL	LNK5 574
END FILE IPOUT	LNK5 575
REWIND IPOUT	LNK5 576
	LNK5 577
	LNK5 578
	LNK5 579
	LNK5 580
	LNK5 581
	LNK5 582

WRITE (ISOUT,18)IPOUT	LNK5 583
PRINT 18,IPOUT	LNK5 584
PRINT 19,IPOUT	LNK5 585
C	LNK5 586
C 5010 SKIP OVER ANY UNUSED WIND DATA	LNK5 587
C A CARD CONTAINING 'END OF WIND FIELD DATA' MUST MARK THE END OF	LNK5 588
C THE WIND FIELD DATA DECK	LNK5 589
5010 READ(ISIN,1)RTST	LNK5 590
RTST=AND(ENDWFD,COMPL(RTST))	LNK5 591
IF(RTST)5010,800,5010	LNK5 592
C	LNK5 593
C 800 PREPARE TO CALL OUTPUT PROCESSOR PROGRAM	LNK5 594
800 IEXEC=2	LNK5 595
RETURN	LNK5 596
C	LNK5 597
C 400 GET OR OTHERWISE PRODUCE THE NEXT TIME INTERVAL'S WIND FIELD.	LNK5 598
400 NTI=0	LNK5 599
IEXEC = 1	LNK5 600
RETURN	LNK5 601
END	LNK5 602

603*

603 *

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

```
LNK6
LNK6 1
LNK6 2
LNK6 3
LNK6 4
```

5 *

5 *

```

$IBFTC RDCIR LIST,DECK,M94/2                                RDCI
SUBROUTINE RDCIRS                                           RDCI 1
C 12 OCT 66                                                RDCI 2
C T.W.SCHWENKE TECHNICAL OPERATIONS RESEARCH, INC.       RDCI 3
C                                                         RDCI 4
C *****RDCI 5
C                                                         RDCI 6
C THIS PROGRAM READS LOCAL CIRCULATION SYSTEM INPUTS. IT READS RDCI 7
C SYSTEM COORDINATE LIMITS, CIRMIX( ),CRMXX( ),CRMINY( ),CRMXX(Y RDCI 8
C THE INDEX OF THE APPLICABLE COMPUTATION CODE FOR EACH LOCAL RDCI 9
C SYSTEM IS STORED IN NCRTYP( ) RDCI 10
C A COUNT OF THE NUMBER OF LOCAL SYSTEMS IS RECORDED IN NLOCIR RDCI 11
C                                                         RDCI 12
C *****RDCI 13
C                                                         RDCI 14
C COMMON /SET1/ RDCI 15
1 DIAM , DETID , IRISE , IEXEC , ISIN , ISOUT , RDCI 16
2 SD , SPAR , SSAM , TME , TMP1 , TMP2 , RDCI 17
3 T2M , U , VPR , W , X , Z , RDCI 18
4 WHY , RMIN , IDISTR , SPAR1 , SPAR2 , SPAR3 , RDCI 19
5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 RDCI 20
DIMENSION DETID(12),WHY(40) RDCI 21
C                                                         RDCI 22
C *****RDCI 23
C                                                         RDCI 24
C COMMON /SET2/ RDCI 25
1 S , SUBSID , GRINT , BXLL , BXLU , BYLL RDCI 26
2 BYLU , TXLL , TXLU , TYLL , TILU , XGZ RDCI 27
3 YGZ , NBLCK , HTOPU , TTOPU , TLIM , JLIM RDCI 28
4 KLIM , II , JJ , KK , KP , YP RDCI 29
5 ZP , FMAS , TP , PS , VX , VY RDCI 30
6 VZ , IL , JL , IBADD , WGRINT , NSTRAT RDCI 31
7 WLLX , WLLY , WURX , WURY , BOTHIT , IPARIN RDCI 32
8 IUTOPU , IOWIND , IHTOPU , IPOUT , IPARUT , JTOP1 RDCI 33
9 JWIND1 , IRROR , TLIMIT , ENDTIM , IC , IBYPAS RDCI 34
1 JTOPU , NLOST , NS , NTU , NTI , NW RDCI 35
2 NALOFT , JTIME1 , N3MAX , NFREE , N , NCL RDCI 36
3 CRMXX , CRUHT , NCRTYP , BZ , CRMINX , CRMINY RDCI 37
4 CU , SN , CS , NLOCIR , DTLOC , ATEMP RDCI 38
5 RHO , NA , IGZ , DTMAC , FRUG , CRMXX RDCI 39
6 RUPART RDCI 40
DIMENSION TOPLEM(4,4) , NINTAR(4) , ITOPLM(3,4) RDCI 41
DIMENSION S(10,10) , SUBSID(400) , IC(18) RDCI 42
DIMENSION XP(200) , YP(200) , ZP(200) , FMAS(200) RDCI 43
DIMENSION TP(200) , PS(200) , ATEMP(260) , RHO(260) RDCI 44
DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70) RDCI 45
DIMENSION JL(70) , IBADD(70) , WURX(70) RDCI 46
DIMENSION WGRINT(70) , WLLX(70) , WLLY(70) RDCI 47
DIMENSION WURY(70) , BOTHIT(70) , SN(6) , CS(6) RDCI 48
DIMENSION CRMINX(6) , CRMXX(6) , CRMINY(6) , CRMXX(6) RDCI 49
DIMENSION CRUHT(6) , NCRTYP(6) , CU(6) RDCI 50
C                                                         RDCI 51
C *****RDCI 52
C                                                         RDCI 53
C 1 FORMAT(4E12.5,I3) RDCI 54
2 FORMAT(//15X,22HLOCAL CIRCULATION CODE14,18H IS NOT AVAILABLE.) RDCI 55
C                                                         RDCI 56
C *****RDCI 57
C                                                         RDCI 58
C DATA PROGRAM /6HRDCIRS/ RDCI 59
C                                                         RDCI 60

```

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 USER	RDCI	78
C	IN TRANSPORTING PARTICLES	RDCI	79
	RETURN	RDCI	81
100	NLOCIR = K-1	RDCI	80
	END	RDCI	82

83*

83 *

```

SIBFTC MKWIN2 LIST,DECK,M94/2                                MKWI 0
SUBROUTINE MKWIND                                             MKWI 1
C 28 NOVEMBER 1966                                           MKWI 2
C T. W. SCHWENKE TECHNICAL OPERATIONS RESEARCH SR MKWIND MK2 MKWI 3
C                                                            MKWI 4
C *****                                                    MKWI 5
C                                                            MKWI 6
C THIS SUBROUTINE FORMS A HORIZONTALLY AND VERTICALLY VARIANT WIND MKWI 7
C DESCRIPTION IN CORE ON THE BASIS OF INPUTS FROM THE SYSTEM INPUT MKWI 8
C TAPE. INPUTS ARE AS FOLLOWS..                               MKWI 9
C 1. CONTROL VARIABLES ENDTIM, WHICH GIVES THE TIME AT WHICH MKWI 10
C THE FOLLOWING DATA CEASE TO BE VALID, ALPHA, WHICH IS A MKWI 11
C WEIGHTING FACTOR TO BE APPLIED TO VERTICAL DISTANCES, BETA, MKWI 12
C WHICH IS A WEIGHTING FACTOR TO BE APPLIED TO HORIZONTAL MKWI 13
C DISTANCES, AND NN, WHICH SPECIFIES THE NUMBER OF NEAREST MKWI 14
C VECTORS TO BE USED IN ESTIMATING THE WIND VECTOR AT A GRID MKWI 15
C POINT.                                                       MKWI 16
C 2. WIND GRID SPECIFICATIONS IN THE FORM BUTHIT(J),WGRINT(J), MKWI 17
C WLLX(J),WLLY(J),WURX(J),WURY(J) (6F10.3) MKWI 18
C WHERE BUTHIT(J) IS THE HEIGHT OF THE BOTTOM OF THE J-TH ARRAY, MKWI 19
C WGRINT(J) IS THE GRID INTERVAL TO BE USED IN THE J-TH LAYER, MKWI 20
C AND WLLX(J), WLLY(J),WURX(J),WURY(J) ARE LOWER LEFT CORNER AND MKWI 21
C UPPER RIGHT CORNER LIMIT COORDINATES. MKWI 22
C 3. WIND VECTOR DATA IN THE FORM (ZTHIT(J),ZS(J),XS(J),YS(J), MKWI 23
C SX(J),SY(J),SZ(J) MKWI 24
C (6F12.3) WHERE ZS(J) IS THE HEIGHT OF THE J-TH VECTOR, MKWI 25
C +SX(J) IS THE EASTWARD POINTING COMPONENT OF THE J-TH VECTOR, MKWI 26
C +SY(J) IS THE NORTHWARD POINTING COMPONENT OF THE J-TH VECTOR, MKWI 27
C +SZ(J) IS THE UPWARD COMPONENT OF THE J-TH VECTOR, XS(J) IS THE MKWI 28
C EAST-WEST COORDINATE OF THE J-TH VECTOR, AND YS(J) IS THE MKWI 29
C NORTH-SOUTH COORDINATE OF THE J-TH VECTOR. MKWI 30
C THE LAYER READING OPERATION IS TERMINATED WHEN BUTHIT(J) = MKWI 31
C 99999. OR MORE IS ENCOUNTERED. THE VECTOR READING OPERATION MKWI 32
C IS TERMINATED WHEN ZS(J)=99999.0 OR MORE IS ENCOUNTERED. MKWI 33
C A WIND FIELD TAPE IS NOT WRITTEN BY THIS PROGRAM. MKWI 34
C                                                            MKWI 35
C ***** GLOSSARY***** MKWI 36
C                                                            MKWI 37
C ALPHA A WEIGHTING FACTOR FOR THE VERTICAL DISTANCES MKWI 38
C BETA A WEIGHTING FACTOR FOR THE HORIZONTAL DISTANCES MKWI 39
C B10 AN ARBITRARILY LARGE NUMBER MKWI 40
C BY DISTANCE BETWEEN THE CURRENT GRID POINT AND THE MOST MKWI 41
C NEARBY OF THE NEAREST WIND DATA POINTS MKWI 42
C DZ1(J) SEE DZ2(J). FOR Y-DIRECTION BUT UNWEIGHTED MKWI 43
C DZ2(J) SQUARE OF WEIGHTED Z-DISTANCE BETWEEN GRID POINT AND MKWI 44
C THE J-TH DATA VECTOR MKWI 45
C G10 AN ARBITRARILY SMALL NUMBER MKWI 46
C IWIN SYSTEM INPUT TAPE NUMBER MKWI 47
C IOUT SYSTEM OUTPUT TAPE NUMBER MKWI 48
C JH FINAL (HIGHER) X INDEX. SEE JLL MKWI 49
C JXH NUMBER OF X ROWS IN OUTPUT GRID MKWI 50
C JLL INITIAL (LOWER) X-INDEX FOR PRINTING PLANE OF THE MKWI 51
C WIND FIELD ARRAY MKWI 52
C JSTPS THE NUMBER OF WIND STRATA IN THE DESIRED WIND FIELD MKWI 53
C DESCRIPTION MKWI 54
C JSTPV THE TOTAL NUMBER OF WIND DATA POINTS BEING USED MKWI 55
C K USED BY MKWIND AS A STRATUM INDEX AT PRINTING TIME MKWI 56
C KY Y-DIRECTION INDEX AT PRINTING TIME MKWI 57
C RAD(J) INDEX OF DISTANCES BETWEEN THE CURRENT GRID POINT MKWI 58
C AND THE JTH DATA POINT MKWI 59

```

```

C      NADT      INDEX OF THE NAD THAT CONTAINS THE ADDRESS OF THE D2      MKW1 60
C      WHICH IS THE LARGEST OF NEAREST NN DATA POINTS      MKW1 61
C      NCODE     IDENTIFICATION NUMBER FOR THE METHOD OF COMPUTATION TO      MKW1 62
C      BE USED IN TRANSLATING THE WIND DATA INTO THE WIND      MKW1 63
C      ARRAYS      MKW1 64
C      NN        THE NUMBER OF NEAREST DATA VECTORS THAT THE USER WISHES      MKW1 65
C      TO BE USED IN COMPUTATIONS      MKW1 66
C      T1        TEMPORARY STORAGE      MKW1 67
C      WG2       HALF OF GRID INTERVAL WGRINT(JW)      MKW1 68
C      XG        X COORDINATE OF GRID POINT      MKW1 69
C      XLIM      AN X-DIRECTION LIMIT FOR TESTING FOR THE COMPLETION OF      MKW1 70
C      A ROW IN THE WIND FIELD ARRAY      MKW1 71
C      YG        Y COORDINATE OF GRID POINT      MKW1 72
C      YLIM      SEE XLIM. FOR THE Y-DIRECTION      MKW1 73
C      ZG        Z COORDINATE OF GRID POINT      MKW1 74
C      MKW1 75
C *****
C      COMMON /SET1/
1      DIAM      , DETID      , IRICE      , IEXEC      , ISIN      , ISOUT      , MKW1 77
2      SD        , SPAR      , SSAM      , TME        , TMP1      , TMP2      , MKW1 78
3      T2M      , U          , VPR      , W          , X          , Z          , MKW1 79
4      JHY      , RMIN      , IDICTR     , SPAR1      , SPAR2      , SPAR3      , MKW1 80
5      SPAR4     , SPAR5     , SPAR6     , SPAR7      , SPAR8      , SPAR9      , MKW1 81
      DIMENSION DETID(12),JHY(40)      MKW1 82
      COMMON /SET2/
1      S         , SUBCID     , GRINT     , EXLL      , EXEC      , BYLL      , MKW1 83
2      BYLC      , TALL      , TXLC      , TYLL      , TYLC      , XGL      , MKW1 84
3      YGZ      , NSLCK     , DTSP2     , TTSP2     , TLIM      , ULIM      , MKW1 85
4      ALIM      , II        , JJ        , KK        , XP        , YP        , MKW1 86
5      ZP        , FMAX      , TP        , PU        , VX        , VY        , MKW1 87
6      VZ        , IL        , JL        , IBADD     , WGRINT     , NSTRAT     , MKW1 88
7      ALLX      , ALLY      , ACRX      , ACRY      , BTHIT     , IPARIN     , MKW1 89
8      ITSP2     , IWIND     , IHTSP2     , IPOUT     , IPARST     , JTCPI     , MKW1 90
9      JWIND1     , IRROR     , TLIMIT     , ENDTIM     , IC        , ILYPAL     , MKW1 91
1     JTSPJ      , NLST      , NG        , NT2       , NT1       , NA        , MKW1 92
2     NALFT      , JTIME1     , NBMAX     , NFREE      , N        , NCL       , MKW1 93
3     CRMAXY     , CRUHT      , ACRTP     , BZ        , CRMINX     , CRMINY     , MKW1 94
4     LZ         , SN        , CS        , NLBCIR     , DTSCC      , ATEMP      , MKW1 95
5     RH2        , NA        , TGZ      , DTMAC      , FRAG      , CRMAXX     , MKW1 96
6     RZPART      MKW1 100
      DIMENSION TSP2LM(4,4) , NINTAR(4) , ITSP2LM(3,4)      MKW1 101
      DIMENSION S(10,10) , SUBCID(400) , IC(18)      MKW1 102
      DIMENSION XP(200) , YP(200) , ZP(200) , FMAX(200)      MKW1 103
      DIMENSION TP(200) , PU(200) , ATEMP(260) , RH2(260)      MKW1 104
      DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70)      MKW1 105
      DIMENSION JL(70) , IBADD(70) , ACRX(70)      MKW1 106
      DIMENSION AGRINT(70) , ALLX(70) , ALLY(70)      MKW1 107
      DIMENSION ACRY(70) , BTHIT(70) , SN(6) , CS(6)      MKW1 108
      DIMENSION CRMINX(6) , CRMAXX(6) , CRMINY(6) , CRMAXY(6)      MKW1 109
      DIMENSION CRUHT(6) , ACRTP(6) , UZ(6)      MKW1 110
C      PARAMETERS PECULIAR TO MKWIND MK2      MKW1 111
      DIMENSION XS(300),YS(300),XA(300),YA(300),ZS(300),ZL(300),ZU(300),      MKW1 112
      ID42(300),DY2(300),D2(300),NAD(300)      MKW1 113
C      MKW1 114
C *****
C      1      FORMAT(6F10.3)      MKW1 115
      2      FORMAT(/78H)THE FOLLOWING WIND VECTORS HAVE BEEN USED TO DEFINE      MKW1 116
      1N ATMOSPHERE UP TO TIME F10.0)      MKW1 117
      MKW1 118
      MKW1 119

```

```

3   FORMAT(/7X6HVECT0R7X22HHORIZONTAL COORDINATES8X17HVECT0R COMPONENTMKW1 120
15/7X6HHEIGHT10X2HXS10X2HYS10X2HVX10X2HVV10X2HVZ) MKW1 121
4   FORMAT(4X,6F12.3) MKW1 122
5   FORMAT(/62H INADEQUATE CONTROL DATA. COMPUTATION METHOD 1 WILL BMKW1 123
1E USED.) MKW1 124
6   FORMAT(64H ENCOUNTERED TWO WIND GRID LAYER REQUESTS FOR THE SAME AMKW1 125
1LTITUDE.) MKW1 126
8   FORMAT(5I4) MKW1 127
9   FORMAT(19HCCOMPUTATION METHOD,14,17H IS NOT AVAILABLE) MKW1 128
10  FORMAT(/10X26HREQUESTED GRID ARRANGEMENT/7X6HHEIGHT9X8HINTERVAL22XMKW1 129
16HLIMITS/33X4HWLLX8X4HWLLY8X4HWJRX8X4HWURY/(4X6F12.3)) MKW1 130
11  FORMAT(16H1WIND COMPONENTS) MKW1 131
12  FORMAT(1X,13HEAST-WEST R0W,16) MKW1 132
13  FORMAT(1X,10F12.3) MKW1 133
14  FORMAT(19HCCOMPUTATION METHOD16,17H WAS USED ON THE 16,21H NEARESTMKW1 134
1 DATA POINTS.) MKW1 135
15  FORMAT(/76H LEVEL13,6X,6HBASE AT ,F12.3,7H METERS) MKW1 136
16  FORMAT(/19H NN WAS REDUCED TO 13) MKW1 137
17  FORMAT(6F12.3) MKW1 138
22  FORMAT(6E20.8) MKW1 139
23  FORMAT( / 119H AN EXCESSIVE NUMBER OF SIGNIFICANT FIGURES ARE L03MKW1 140
1T IN THE LEAST SQUARES CALCULATION. THE DATA POINTS APPROACH A LINEMKW1 141
2NE/63H OR A PLANE. THE WEIGHTED VECTOR METHOD IS USED FOR GRID POINTMKW1 142
3INT, 5X, 9H(X,Y,Z)=(, F12.3,1H,,F12.3,1H,,F12.3,1H)) MKW1 143
24  FORMAT(/76H NO VECTORS LIE WITHIN THE SPECIFIED WEIGHTING REGIONMKW1 144
1. A RANDOM SELECTION OF 14, 30H VECTORS ARE EQUALLY WEIGHTED , MKW1 145
2/ 5X, 15H FOR GRID POINT, MKW1 146
3 5X, 9H(X,Y,Z)=(, F12.3,1H,,F12.3,1H,,F12.3,1H)) MKW1 147
25  FORMAT ( / 10X,8HALPHA = F14.3, 7HMETERS,10X 7HDLTA = F14.3, MKW1 148
1 7HMETERS. ) MKW1 149
C MKW1 150
C ***** MKW1 151
C MKW1 152
DATA PR0GRM,BIG,NWIND,NWTST,GIS /6HMKWIND,1.0E+30,1.500,0,1.0E-30/ MKW1 153
C MKW1 154
C ***** MKW1 155
C ***** MKW1 156
C MKW1 157
READ (ISIN,1)ENDTIN,ALPHA,BETA MKW1 158
ALPHA2=ALPHA*ALPHA MKW1 159
BETA2=BETA*BETA MKW1 160
C READ SPECIFICATION OF DESIRED WIND ARRAY PROPERTIES MKW1 161
READ (ISIN,5)NN,NCODE MKW1 162
IF (NN)204,204,2041 MKW1 163
204 IRR0R=204 MKW1 164
7734 CALL ERROR(PR0GRM,IRR0R,IS0UT) MKW1 165
2041 D0 104 J=1,NSTRAT MKW1 166
READ (ISIN,1) B0THIT(J),AGRINT(J),WLLX(J),WLLY(J),WJRX(J),WJRY(J)MKW1 167
IF (B0THIT(J)-999999.0)104,105,105 MKW1 168
104 CONTINUE MKW1 169
IRR0R=104 MKW1 170
CALL ERROR(PR0GRM,IRR0R,IS0UT) MKW1 171
1041 READ (ISIN,1)XST MKW1 172
IF (XST-999999.0)1041,105,105 MKW1 173
105 J02PJ=J-1 MKW1 174
C MKW1 175
C NEW S0RT MKW1 176
1054 KS=0 MKW1 177
D0 1051 J=2,J02PJ MKW1 178
IF (B0THIT(J)-B0THIT(J-1))1153,1052,1051 MKW1 179

```

1153 KS=1	MKW1 180
HTST=B0THIT(J-1)	MKW1 181
VXT=WGRINT(J-1)	MKW1 182
VYT=WLLX(J-1)	MKW1 183
VZT=WLLY(J-1)	MKW1 184
XST=WURX(J-1)	MKW1 185
YST=WURY(J-1)	MKW1 186
B0THIT(J-1)=B0THIT(J)	MKW1 187
WGRINT(J-1)=WGRINT(J)	MKW1 188
WLLX(J-1)=WLLX(J)	MKW1 189
WLLY(J-1)=WLLY(J)	MKW1 190
WURX(J-1)=WURX(J)	MKW1 191
WURY(J-1)=WURY(J)	MKW1 192
B0THIT(J)=HTST	MKW1 193
WGRINT(J)=VXT	MKW1 194
WLLX(J)=VYT	MKW1 195
WLLY(J)=VZT	MKW1 196
WURX(J)=XST	MKW1 197
WURY(J)=YST	MKW1 198
1051 CONTINUE	MKW1 199
IF(KS)1054,1055,1054	MKW1 200
1052 WRITE (IS2UT,6)	MKW1 201
ERROR=1052	MKW1 202
GO TO 7734	MKW1 203
C	MKW1 204
C 1055 SORT OF THE REQUESTED LAYERS IS COMPLETE	MKW1 205
C NOW MAKE SURE THAT THERE IS SUFFICIENT SPACE FOR THE WIND FIELD	MKW1 206
1055 DO 1056 J=1,UTOPJ	MKW1 207
K1=(WURX(J)-WLLX(J))/WGRINT(J)+1.0	MKW1 208
K2=(WURY(J)-WLLY(J))/WGRINT(J)+1.0	MKW1 209
1056 NWTST=NWTST+K1*K2	MKW1 210
C	MKW1 211
C IS AVAILABLE WIND MEMORY EXCEEDED	MKW1 212
IF(NXIND .GT. NWTST) GO TO 1057	MKW1 213
1058 ERROR=-1058	MKW1 214
GO TO 7734	MKW1 215
1057 DO 100 J=1,300	MKW1 216
READ (ISIN,17)ZS(J),XS(J),YS(J),SX(J),SY(J),SZ(J)	MKW1 217
IF(ZS(J)-999999.0)100,101,101	MKW1 218
101 JTOPV=J-1	MKW1 219
IF((NN-JTOPV)106,106,2051	MKW1 220
2051 NN=JTOPV	MKW1 221
WRITE (IS2UT,16)JTOPV	MKW1 222
GO TO 106	MKW1 223
100 CONTINUE	MKW1 224
ERROR=100	MKW1 225
GO TO 7734	MKW1 226
C	MKW1 227
C VECTOR DATA ARE IN WIND ARRAYS ON INDICES J=1,JTOPV	MKW1 228
C FIRST USE NCODE AS A METHOD CONTROL VARIABLE. BRANCH ON NCODE VIA	MKW1 229
C A COMPUTED GO TO TO THE DESIRED COMPUTATION METHOD CODE.	MKW1 230
106 NNI=NN+1	MKW1 231
IF(NCODE)110,110,112	MKW1 232
112 IF(NCODE-6)113,113,110	MKW1 233
C	MKW1 234
C 110 NCODE IS INCORRECT	MKW1 235
110 WRITE (IS2UT,5)	MKW1 236
NCODE=1	MKW1 237
113 GO TO (115,116,117,118,119,120),NCODE	MKW1 238
C	MKW1 239

```

C 115 METHOD 1 USES THE NN NEAREST DATA POINTS. METHODS 2,3 AND 4 ALSO MKW1 240
C USE THIS CODE BUT FOR METHOD 2, NN=1 AND FOR METHOD 3, NN=JTOPV. MKW1 241
C FOR METHOD 4, THE NN SPECIFIED BY THE USER IS USED IN THE MKW1 242
C LEAST SQUARES METHOD (NN MUST BE GREATER THAN THREE). MKW1 243
115 ISAD(1)=1 MKW1 244
K=0 MKW1 245
JW=1 MKW1 246
C MKW1 247
C NOW FILL IN THE WIND GRID SIZE WORDS MKW1 248
1151 IL(JW)=((WGRX(JW)-ALLX(JW)))/WGRINT(JW) +.9999999 MKW1 249
JL(JW)=((WGRY(JW)-ALLY(JW)))/WGRINT(JW) +.9999999 MKW1 250
C MKW1 251
C NOW INITIALIZE FOR FILLING IN THE WIND GRID MKW1 252
AG2=WGRINT(JW)/2.0 MKW1 253
LX=1 MKW1 254
LY=1 MKW1 255
XG=ALLX(JW)+AG2 MKW1 256
YG=ALLY(JW)+AG2 MKW1 257
IF(JW-JTOPV)1154,1155,1159 MKW1 258
1159 IRROR=1159 MKW1 259
GO TO 7734 MKW1 260
1155 IF(JW-1)1156,1156,1157 MKW1 261
1156 ZG=UBTHIT(JW) MKW1 262
GO TO 1158 MKW1 263
1157 ZG=ZG+UBTHIT(JW)-UBTHIT(JW-1) MKW1 264
GO TO 1158 MKW1 265
1154 ZG=(UBTHIT(JW)+UBTHIT(JW+1))/2.0 MKW1 266
C MKW1 267
C SET ALL NAD(J) EQUAL TO J TO PROVIDE INDICES FOR THE FULL SET OF MKW1 268
C DATA POINTS AND TO PROVIDE AN INITIAL SET OF -NEAREST- DATA POINTS MKW1 269
C SET NADT=1 TO BEGIN THE SORT PROCEDURE THAT SELECTS THE MOST MKW1 270
C REMOTE OF THE SET OF -NEAREST- DATA POINTS. NOTE THAT FOR THE 1ST MKW1 271
C PASS ALL THE NN -NEAREST- POINTS ARE EQUALLY LIKELY TO BE THE MOST MKW1 272
C REMOTE OF THE SET. MKW1 273
1158 DO 203 J=1,JTOPV MKW1 274
203 NAD(J)=J MKW1 275
NADT=1 MKW1 276
C MKW1 277
C COMPUTE DISTANCES BETWEEN THE CURRENT GRID POINT (XG,YG,ZG) AND MKW1 278
C EACH OF THE DATA VECTOR LOCATIONS MKW1 279
C MKW1 280
C COMPUTE SQUARED Z DELTAS MKW1 281
DO 199 J=1,JTOPV MKW1 282
T1=(ZG(J)-ZG) MKW1 283
T1=T1*T1 MKW1 284
T2=(ALPHA2-T1) MKW1 285
IF(T2.LT.0.0) T2=0.0 MKW1 286
199 DZ2(J)=T2/(ALPHA2+T1) MKW1 287
C MKW1 288
C COMPUTE SQUARED Y DELTAS MKW1 289
DO 201 J=1,JTOPV MKW1 290
T1=YG(J)-YG MKW1 291
201 DY2(J)=T1*T1 MKW1 292
C MKW1 293
C COMPUTE SQUARED DISTANCES MKW1 294
DO 202 J=1,JTOPV MKW1 295
T1=(XG(J)-XG)*(XG(J)-XG)+DY2(J) MKW1 296
T2=((BETA2-T1)/(BETA2+T1))*DZ2(J) MKW1 297
IF(T2-G1) 2021,2021,2022 MKW1 298
2021 D2(J)=B1 MKW1 299

```

G0 T0 202	300
2022 D2(J)=1.0/T2	301
202 CONTINUE	302
C	303
C FIND THE ADDRESS OF AND DISTANCE TO THE MOST REMOTE POINT OF THE	304
C NN -NEAREST- POINTS (THE POINTS WHOSE ADDRESSES ARE GIVEN BY	305
C NAD(1),NAD(NN).) STORE THAT MAXIMUM DISTANCE IN THE WORD DM AND	306
C SET NADT SUCH THAT DM=D2(NAD(NADT)).	307
KL=NAD(NADT)	308
DM=D2(KL)	309
D0 207 J=1,NN	310
KL=NAD(J)	311
IF(DM-D2(KL))206,207,207	312
206 DM=D2(KL)	313
NADT=J	314
207 CONTINUE	315
C AT THIS POINT, DM IS THE LARGEST D2(J) FOR J=NAD(J),NAD(NN)	316
C	317
IF (NN1-JTOPV)2072,2072,2073	318
C	319
C2072 NOW SELECT BEST NN POINTS	320
C SCAN THE SET D2(J),J=NAD(NN+1,JTOPV) UNTIL A D2(J) LESS THAN DM	321
C IS FOUND. IF ONE IS FOUND, SWITCH NAD(NADT) WITH THE SELECTED NAD	322
C THEN RESET DM AND NADT TO INDICATE THE MOST REMOTE OF THE NEAREST	323
C NN POINTS. WHEN THE FULL SET D2(J),J=NAD(NN+1,JTOPV) HAS BEEN	324
C SCANNED, THE SET OF NEAREST DATA POINTS HAS BEEN SELECTED. ONLY	325
C ONE SCAN IS REQUIRED.	326
2072 D0 210 J=NN1,JTOPV	327
KL=NAD(J)	328
IF(DM-D2(KL))210,210,211	329
211 NTEMP=NAD(J)	330
NAD(J)=NAD(NADT)	331
NAD(NADT)=NTEMP	332
C	333
C NOW RESET DM AND NADT TO THE NEXT MOST REMOTE POINT	334
DM=D2(KL)	335
D0 212 KKK=1,NN	336
KL=NAD(KKK)	337
IF(DM-D2(KL))213,212,212	338
213 DM=D2(KL)	339
NADT=KKK	340
C	341
C DM AND NADT ARE SET WITH THE PARAMETERS OF THE MOST REMOTE OF	342
C THE NEAREST NN POINTS	343
212 CONTINUE	344
210 CONTINUE	345
2073 CONTINUE	346
C	347
C THE NEAREST NN HAVE BEEN FOUND	348
C *****SOME DAY INSERT HERE A BRANCH FOR WEIGHTING METHOD HERE**	349
C	350
C INCREMENT INDEX FOR STORING VECTOR COMPUTED FOR POINT (XG,YG,ZG)	351
K=K+1	352
C	353
C IS THE LEAST SQUARES METHOD TO BE USED. YES TO 2081	354
IF(NC0DE-4)2080,2081,2080	355
C	356
C2081 THIS IS THE LEAST SQUARES METHOD	357
C INITIALIZE FOR LEAST SQUARES METHOD	358
2081 SNN=NN	359

SDX=0.0	MKW1	360
SDY=0.0	MKW1	361
SDZ=0.0	MKW1	362
SDX2=0.0	MKW1	363
SDY2=0.0	MKW1	364
SDZ2=0.0	MKW1	365
SDXY=0.0	MKW1	366
SDXZ=0.0	MKW1	367
SDYZ=0.0	MKW1	368
SAX=0.0	MKW1	369
SAV=0.0	MKW1	370
SAW=0.0	MKW1	371
SUX=0.0	MKW1	372
SUY=0.0	MKW1	373
SVZ=0.0	MKW1	374
SVX=0.0	MKW1	375
SVY=0.0	MKW1	376
SVZ=0.0	MKW1	377
SWX=0.0	MKW1	378
SWY=0.0	MKW1	379
SWZ=0.0	MKW1	380
C	MKW1	381
C	MKW1	382
BEGIN LOOP TO EVALUATE INTERMEDIATE STEP FOR LEAST SQUARES CORREC.	MKW1	383
DO 3100 J=1,NN	MKW1	384
KL=NA2(J)	MKW1	385
C	MKW1	386
C	MKW1	387
COMPUTE DISTANCE BETWEEN KL-TH DATA POINT AND CURRENT GRID POINT	MKW1	388
T1=X5(KL)-XG	MKW1	389
TY=Y5(KL)-YG	MKW1	390
TZ=Z5(KL)-ZG	MKW1	391
C	MKW1	392
C	MKW1	393
COMPUTE ELEMENTS OF LEAST SQUARES MATRIX, B	MKW1	394
SDX=SDX+T1	MKW1	395
SDY=SDY+TY	MKW1	396
SDZ=SDZ+TZ	MKW1	397
SDX2=SDX2+T1*T1	MKW1	398
SDY2=SDY2+TY*TY	MKW1	399
SDZ2=SDZ2+TZ*TZ	MKW1	400
SDXY=SDXY+T1*TY	MKW1	401
SDXZ=SDXZ+T1*TZ	MKW1	402
SDYZ=SDYZ+TY*TZ	MKW1	403
SAX=SAX+X5(KL)	MKW1	404
SAV=SAV+Y5(KL)	MKW1	405
SAW=SAW+Z5(KL)	MKW1	406
SUX=SUX+T1*X5(KL)	MKW1	407
SUY=SUY+TY*Y5(KL)	MKW1	408
SVZ=SVZ+TZ*Z5(KL)	MKW1	409
SVX=SVX+T1*X5(KL)	MKW1	410
SVY=SVY+TY*Y5(KL)	MKW1	411
SVZ=SVZ+TZ*Z5(KL)	MKW1	412
SUX=SUX+T1*X5(KL)	MKW1	413
SUY=SUY+TY*Y5(KL)	MKW1	414
SVZ=SVZ+TZ*Z5(KL)	MKW1	415
3100	MKW1	416
SAUG1=SDYP*SDZ2-SDYZ*SDYZ	MKW1	417
SAUG2=SDXY*SDZ2-SDYZ*SDXZ	MKW1	418
SAUG3=SDXY*SDYZ-SDYP*SDXZ	MKW1	419
SAUG4=SDX2*SDYZ-SDXY*SDXZ	MKW1	420
C	MKW1	421
C	MKW1	422
COMPUTE COMPLEMENTARY MINORS OF MATRIX B	MKW1	423
B11=SDX2*SAUG1-SDXY*SAUG2+SDXZ*SAUG3	MKW1	424

B21=SDX*SAUG1-SDY*SAUG2+SDZ*SAUG3	MARKI 420
B31=SDX*SAUG2-SDY*(SDX2*SDZ2-SDXZ*SDXZ)+SDZ*SAUG4	MARKI 421
B41=SDX*SAUG3-SDY*SAUG4+SDZ*(SDX2*SDY2-SDXY*SDXY)	MARKI 422
C	MARKI 423
C TEST TO SEE IF A ROW OR COLUMN IS APPROXIMATELY ZERO	MARKI 424
RB= AMAX1(ABS(SNN*B11),ABS(SDX*B21),ABS(SDY*B31), ABS(SDZ*B41))	MARKI 425
IF(RB-1.0E-20)3800,3800,3700	MARKI 426
C	MARKI 427
C COMPUTE DETERMINANT OF B	MARKI 428
3700 BBB=SNN*B11-SDX*B21+SDY*B31-SDZ*B41	MARKI 429
C	MARKI 430
C TEST FOR LOSS OF PRECISION	MARKI 431
IF(ABS(BBB/BB)-0.001)3800,3800,3900	MARKI 432
C	MARKI 433
C3800 TOO MANY SIGNIFICANT FIGURES ARE LOST IN THE LEAST SQUARES	MARKI 434
C CALCULATION. THE DATA POINTS APPROACH A POINT, A LINE, OR A	MARKI 435
C PLANE. USE THE WEIGHTED VECTOR METHOD	MARKI 436
3800 WRITE (IS2UT,23)XG,YG,ZG	MARKI 437
G2 TO 2080	MARKI 438
C	MARKI 439
C COMPUTE WIND VECTORS	MARKI 440
3900 VX(K)=(B11*SAU-B21*SAU+B31*SAU-B41*SAU)/BBB	MARKI 441
VY(K)=(B11*SAV-B21*SAV+B31*SAV-B41*SAV)/BBB	MARKI 442
VZ(K)=(B11*SAW-B21*SAW+B31*SAW-B41*SAW)/BBB	MARKI 443
G2 TO 2090	MARKI 444
C	MARKI 445
C2080 COMPUTE AND SUM THE WEIGHTING FACTORS	MARKI 446
2080 SUM=0.0	MARKI 447
D2 214 J=1,NN	MARKI 448
L=NAD(J)	MARKI 449
2142 D2(L)=1.0/D2(L)	MARKI 450
214 SUM=SUM+D2(L)	MARKI 451
IF(SUM/FL2AT(NN) .LE. 0.18) WRITE(IS2UT,24) NN,XG,YG,ZG	MARKI 452
C	MARKI 453
C NOW COMPUTE VECTOR ESTIMATE AT GRID POINT	MARKI 454
C COMPUTE STORAGE INDEX	MARKI 455
C COMPUTE AND STORE WIND ESTIMATE AT GRID POINT	MARKI 456
VX(K)=0.0	MARKI 457
VY(K)=0.0	MARKI 458
VZ(K)=0.0	MARKI 459
D2 216 J=1,NN	MARKI 460
L=NAD(J)	MARKI 461
VX(K)=VX(K)+SX(L)*D2(L)	MARKI 462
VY(K)=VY(K)+SY(L)*D2(L)	MARKI 463
216 VZ(K)=VZ(K)+SZ(L)*D2(L)	MARKI 464
VX(K)=VX(K)/SUM	MARKI 465
VY(K)=VY(K)/SUM	MARKI 466
VZ(K)=VZ(K)/SUM	MARKI 467
2090 XG=XG+WGRINT(JW)	MARKI 468
LX=LX+1	MARKI 469
IF(LX-IL(JW))2011,2011,2012	MARKI 470
2012 XG=WLLX(JW)+WG2	MARKI 471
LY=LY+1	MARKI 472
LX=1	MARKI 473
YG=YG+WGRINT(JW)	MARKI 474
IF(LY-JL(JW))200, 200,1152	MARKI 475
1152 JW=JW+1	MARKI 476
IF(JW-JT&PJ)1160,1160,100	MARKI 477
1160 JT=JW-1	MARKI 478
IBADD(JW)=IBADD(JT)+(IL(JT))*(JL(JT))	MARKI 479

C	GO TO 1151	NRW1 480
C		NRW1 481
C 116	METHOD 2 NEAREST VECTOR	NRW1 482
116	NN=1	NRW1 483
	GO TO 115	NRW1 484
C		NRW1 485
C 117	METHOD 3 ALL VECTORS WEIGHTED	NRW1 486
117	NN=JTOPV	NRW1 487
	GO TO 115	NRW1 488
118	CONTINUE	NRW1 489
C		NRW1 490
C 119	METHOD 4 LEAST SQUARES	NRW1 491
C	USE BRANCH IN NCZDE=4 TO BRANCH TO LEAST SQUARES IN CODE	NRW1 492
C	NN MUST BE GREATER THAN 3 FOR THE LEAST SQUARES METHOD	NRW1 493
1191	IF (NN-4).161,115,115	NRW1 494
1191	NRW1=1161	NRW1 495
	GO TO 7734	NRW1 496
119	CONTINUE	NRW1 497
120	CONTINUE	NRW1 498
121	WRITE (ISCT,9)NRW1	NRW1 499
	NRW1=121	NRW1 500
	GO TO 7734	NRW1 501
130	WRITE (ISCT,2)NRW1	NRW1 502
	WRITE (ISCT,3)	NRW1 503
	WRITE (ISCT,4)(ZS(J),XL(J),YL(J),X(J),Y(J),Z(J),J=1,JTOPV)	NRW1 504
	WRITE (ISCT,10)(P,THIT(J),NRW1(J),ALLX(J), ALLY(J),ALLR(J),NRW1(J),J=1,JTOPV)	NRW1 505
	WRITE (ISCT,14)NRW1,NN	NRW1 507
	WRITE (ISCT,25) ALPHA, BETA	NRW1 508
	IF (IC(7)-1)109,109,109	NRW1 509
109	WRITE (ISCT,11)	NRW1 510
	GO 1,7 NR=1,JTOPV	NRW1 511
	WRITE (ISCT,15)P,SPRINT(N)	NRW1 512
	CH=COL(N)	NRW1 513
	JLL=ID/51(N)	NRW1 514
	GO 108 NR=1,CH	NRW1 515
	CH=JLL+IL(N)-1	NRW1 516
	WRITE (ISCT,12)NR	NRW1 517
	WRITE (ISCT,13)(VX(J),J=JLL,CH)	NRW1 518
	WRITE (ISCT,13)(VY(J),J=JLL,CH)	NRW1 519
	WRITE (ISCT,13)(VZ(J),J=JLL,CH)	NRW1 520
	JLL=CH+1	NRW1 521
108	CONTINUE	NRW1 522
107	CONTINUE	NRW1 523
109	CALL RDCIRC	NRW1 524
	RETURN	NRW1 525
	END	NRW1 526

```

*IBFTC LIN7 LIST,DECK,M94/2 LIN7
SUBROUTINE LINK7 LIN7 1
C T.W.SCHWENKE TECHNICAL OPERATIONS RESEARCH LINK 7 LIN7 2
C 19 OCT 66 LIN7 3
C THIRD PART OF TRANSPORT MODULE. PARTICLE TRANSPORT. LIN7 4
C LIN7 5
C ***** LIN7 6
C SEE SUBROUTINE LINK5 FOR A TRANSPORT GLOSSARY LIN7 7
C LIN7 8
C ***** LIN7 9
C ***** LIN7 10
C ***** LIN7 11
C COMMON /SET1/ LIN7 12
1 DIAM ,DETD(12),IRISE , IEXEC , ISIN , ISOUT , LIN7 13
2 SD , SPAR , SSAM , TIME , IMP1 , IMP2 , LIN7 14
3 TZM , U , VPR , W , X , Z , LIN7 15
4 WHY(4) , RMIN , IDISTR , SPAR1 , SPAR2 , JDONE , LIN7 16
5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 LIN7 17
C ***** LIN7 18
C ***** LIN7 19
C ***** LIN7 20
C COMMON /SET2/ LIN7 21
1 S , SUBSID , GRINI , BALL , BALO , BYLL LIN7 22
2 BYLO , TXLL , TXLU , TYLL , TYLE , AGZ LIN7 23
3 YGZ , NBLOCK , HTOPU , ITOPU , ILIN , JLIN LIN7 24
4 KLIM , II , JJ , KK , XP , YP LIN7 25
5 ZP , FMAS , TP , PS , VX , VY LIN7 26
6 VZ , IL , JL , IBADD , WGRINT , NSTRAT LIN7 27
7 WLLX , WLLY , WURX , WURY , BOTHIT , IPARIN LIN7 28
8 IOTOPU , IOWIND , IHTOPU , IPOUT , IPAROT , JTOPI LIN7 29
9 JWIND1 , IRKOR , TLIMIT , ENDTIM , IC , ISYPAS LIN7 30
1 JTOPJ , NLOST , NG , NTO , NTI , NW LIN7 31
2 NALOFI , JTIME1 , NBMAX , NFREE , N , NCL LIN7 32
3 CRMAXY , CRUMI , NCRITY , BZ , CRMINX , CRMINY LIN7 33
4 UU , SN , CS , NLOCIR , DLOC , ATEMP LIN7 34
5 RHO , NA , LGZ , DIMAC , ERUG , CRMAXX LIN7 35
6 ROPART LIN7 36
DIMENSION TOPLEM(4,4) , NINTAK(4) , ITOPLM(5,4) LIN7 37
DIMENSION S(10,10) , SUBSID(400) , IC(18) LIN7 38
DIMENSION XP(200) , YP(200) , ZP(200) , FMAS(200) LIN7 39
DIMENSION TP(200) , PS(200) , ATEMP(260) , RHO(260) LIN7 40
DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70) LIN7 41
DIMENSION JL(70) , IBADD(70) , WURX(70) LIN7 42
DIMENSION WGRINT(70) , WLLX(70) , WLLY(70) LIN7 43
DIMENSION WURY(70) , BOTHIT(70) , SN(6) , CS(6) LIN7 44
DIMENSION CRMINX(6) , CRMAXX(6) , CRMINY(6) , CRMAXY(6) LIN7 45
DIMENSION CRUMI(6) , NCRITY(6) , UU(6) LIN7 46
C ***** LIN7 47
C ***** LIN7 48
C ***** LIN7 49
1 FORMAT(3I5,6E12.5) LIN7 50
10 FORMAT (I5) LIN7 51
11 FORMAT(6E15.5) LIN7 52
12 FORMAT(5E15.5,6I5) LIN7 53
C ***** LIN7 54
C ***** LIN7 55
C ***** LIN7 56
DATA PROGRAM /6HLINK7 / LIN7 57
C ***** LIN7 58
C ***** LIN7 59
C ***** LIN7 60

```

C		LIN7	61
	JW=0	LIN7	62
	NUL=0	LIN7	63
	EPSIL=1.0	LIN7	64
C		LIN7	65
C	ARE TRANSPORT TRACES TO BE WRITTEN.. YES TO 5500	LIN7	66
	IF(IC(6)-1)5510,5510,5500	LIN7	67
5500	MPNT=2	LIN7	68
	GO TO 5520	LIN7	69
5510	MPNT =1	LIN7	70
C		LIN7	71
C	BRANCH TO READ ADDITIONAL DATA	LIN7	72
C	READ DATA PECULIAR TO EACH LOCAL WIND SYSTEM	LIN7	73
5520	IF(NLOCIR) 510,510,511	LIN7	74
C		LIN7	75
C	511 SETTING J NEGATIVE WILL CAUSE THE LOCAL CIRCULATION SYSTEM CODES	LIN7	76
C	TO READ THE DATA THAT THEY NEED WHEN THEY ARE FIRST ENTERED	LIN7	77
511	J=-1	LIN7	78
	DO 500 I=1,NLOCIR	LIN7	79
	K=I	LIN7	80
	NCIR=NCRTYP(I)	LIN7	81
	GO TO (501,502,503,504,505),NCIR	LIN7	82
501	CALL MTWND1(J,K,AX,AY,AZ)	LIN7	83
	GO TO 500	LIN7	84
502	CALL RGWND1(J,K,AX,AY,AZ)	LIN7	85
	GO TO 500	LIN7	86
503	CALL CBREZ1(J,K,AX,AY,AZ)	LIN7	87
	GO TO 500	LIN7	88
C	***** CODE INSERTION POINTS *****	LIN7	89
504	CONTINUE	LIN7	90
505	CONTINUE	LIN7	91
C	***** CODE INSERTION POINTS *****	LIN7	92
506	ERROR=-500	LIN7	93
	GO TO 300	LIN7	94
508	CONTINUE	LIN7	95
C		LIN7	96
510	IF(TELEF1-ENOTF1)48,47,47	LIN7	97
48	ENDF1=TELEF1	LIN7	98
49	IF(IC(1)-1)51,50,51	LIN7	99
50	ASSIGN 186 TO IT	LIN7	100
	ASSIGN 188 TO IT	LIN7	101
	GO TO 1000	LIN7	102
51	ASSIGN 1871 TO IT	LIN7	103
	ASSIGN 1813 TO IT	LIN7	104
1070	IF=NALOPT	LIN7	105
	IF(OTIME1)1113,1112,1112	LIN7	106
1113	OTIME1=0	LIN7	107
	GO TO 1101	LIN7	108
C		LIN7	109
C	ATTEMPT TO READ IN A BLOCK OF PARTICLES ALOFT RECORDS	LIN7	110
C	FIRST READ BLOCK SIZE	LIN7	111
1112	READ (IPARIN)	LIN7	112
C		LIN7	113
C	IF BLOCK SIZE IS NEGATIVE OR ZERO, NO BLOCK EXISTS	LIN7	114
	IF(N)100,100,101	LIN7	115
C		LIN7	116
C	CHECK TO SEE IF BLOCK WILL FIT IN ARRAY	LIN7	117
101	IF(N-NALOPT)1 21,1021,103	LIN7	118

C		LIN7	119
C103	ERROR - ALOFT LIST TOO LARGE. SHOULD NEVER HAPPEN. GO TO EXIT.	LIN7	120
103	ERROR=103	LIN7	121
C		LIN7	122
C300	GENERALIZED ERROR STOP	LIN7	123
300	CALL ERROR(PROGMM,ERROR,ISOUT)	LIN7	124
C		LIN7	125
1021	CALL DUMPP	LIN7	126
C		LIN7	127
C 102	NOW READ A BLOCK OF PARTICLE ALOFT DESCRIPTIONS	LIN7	128
102	IF=N	LIN7	129
	READ (IPARIN)(XP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,IF)	LIN7	130
C		LIN7	131
C	ARE TRANSPORT TRACES TO BE WRITTEN.. YES TO 5522	LIN7	132
	IF(IC(6))5521,5521,5522	LIN7	133
5522	WRITE (ISOUT,11)(AP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J),J=1,NALOFT)	LIN7	134
5521	NFREE=NFREE-N	LIN7	135
C		LIN7	136
C	*****	LIN7	137
C		LIN7	138
C	BEGINNING OF PARTICLES ALOFT LIST LOOP	LIN7	139
1001	DO 160 J=1,IF	LIN7	140
	ASSIGN 3052 TO IPAS	LIN7	141
	GO TO (5540,5550),MPNT	LIN7	142
5550	WRITE (ISOUT,11) AP(J),YP(J),ZP(J),IP(J),PS(J),FMAS(J)	LIN7	143
C		LIN7	144
C	SELECT PARTICLE TO BE TRANSPORTED -- +IP +FMAS	LIN7	145
5540	IF(FMAS(J))16,15,192	LIN7	146
192	IF(IP(J))160,193,194	LIN7	147
194	IF(IP(J)-ENDTIM)195,160,1971	LIN7	148
1971	IF(IP(J)-ILIMIT)160,160,197	LIN7	149
195	ERROR=-195	LIN7	150
	GO TO 300	LIN7	151
197	ERROR=-197	LIN7	152
	GO TO 300	LIN7	153
C		LIN7	154
C 195	A PARTICLE HAS BEEN SELECTED	LIN7	155
C	IS THE CURRENT PARTICLE WITHIN A LOCAL CIRCULATION SYSTEM	LIN7	156
195	IF(NLOCIR)1950,1950,1951	LIN7	157
1951	DO 1952 K=1,NLOCIR	LIN7	158
C	*****	LIN7	159
	WRITE (ISOUT,1)J,K,NLOCIR,FMAS(J),XP(J),YP(J),ZP(J),IP(J)	LIN7	160
	IF(XP(J)-CRMIX(K))1952,1953,1955	LIN7	161
1953	IF(CRMXXX(K)-XP(J))1952,1954,1954	LIN7	162
1954	IF(YP(J)-CRMINY(K))1952,1955,1955	LIN7	163
1955	IF(CRMXY(K)-YP(J))1952,1956,1956	LIN7	164
C		LIN7	165
C 1956	THE JTH PARTICLE IS WITHIN THE KTH LOCAL CIRCULATION SYSTEM	LIN7	166
C	SUBROUTINE LOTRAN(J,K) TRANSPORTS THE JTH PARTICLE IN THE KTH	LIN7	167
C	LOCAL WIND SYSTEM.	LIN7	168
1956	CALL LOTRAN(J,K)	LIN7	169
C		LIN7	170
C	188 IS WHERE GROUNDED PARTICLES ARE DEALT WITH.	LIN7	171
	IF(ZP(J)+9000.0)188,168,1957	LIN7	172
1957	IF(IP(J)-ENDTIM)1951,1950,1950	LIN7	173
1958	NTI=NTI+1	LIN7	174
	GO TO 160	LIN7	175
1952	CONTINUE	LIN7	176

C		LIN7 177
1950	XX=XP(J)	LIN7 178
	YY=YP(J)	LIN7 179
	ZZ=ZP(J)	LIN7 180
	PSIZE=PS(J)	LIN7 181
1961	CALL GETWIND(XX,YY,ZZ,JWAD,JW)	LIN7 182
	IF(JWAD)1959,1960,196	LIN7 183
C		LIN7 184
C 1959	THE NEEDED WIND FIELD IS NOT IN CORE	LIN7 185
1959	FMAS(J)=-FMAS(J)	LIN7 186
	TP(J)=TLIMIT	LIN7 187
	NLOST=NLOST+1	LIN7 188
	GO TO 160	LIN7 189
1960	IRROK=-1960	LIN7 190
	GO TO 300	LIN7 191
C		LIN7 192
C 196	GET PARTICLE FALL RATE. PUT IT IN FV WITH SIGN POSITIVE.	LIN7 193
196	CALL FALRAT(ZZ,PSIZE,FV,ATEMP,RHO,FROG,ISOUT)	LIN7 194
C		LIN7 195
C	COMPUTE VERTICAL PARTICLE MOTION COMPONENT	LIN7 196
C	A POSITIVE VPZ DENOTES AN UPWARD POINTING VECTOR	LIN7 197
	VPZ=VZ(JWAD)-FV	LIN7 198
C		LIN7 199
C	COMPUTE TIMES TO NEXT MACRO WIND FIELD BORDERS	LIN7 200
C	COMPUTE TIX -- TRANSIT TIME TO X BOUNDARY	LIN7 201
	XBL=(AINI((XP(J)-WLLX(JW))/WGRINT(JW)))*WGRINT(JW) +WLLX(JW)	LIN7 202
	XBU=XBL+WGRINT(JW)	LIN7 203
	YBL=(AINI((YP(J)-WLLY(JW))/WGRINT(JW)))*WGRINT(JW) +WLLY(JW)	LIN7 204
	YBU=YBL+WGRINT(JW)	LIN7 205
	ASSIGN 917 TO N1	LIN7 206
	IF(VX(JWAD))161,162,163	LIN7 207
161	TIX=(XBL-XP(J))/(VX(JWAD))	LIN7 208
	GO TO 164	LIN7 209
162	TIX=100000.0	LIN7 210
	ASSIGN 918 TO N1	LIN7 211
	GO TO 164	LIN7 212
163	TIX=(XBU-XP(J))/(VX(JWAD))	LIN7 213
C		LIN7 214
C	COMPUTE TIY -- TRANSIT TIME TO Y BOUNDARY	LIN7 215
164	ASSIGN 919 TO N2	LIN7 216
	IF(VY(JWAD))165,166,167	LIN7 217
165	TIY=(YBL-YP(J))/(VY(JWAD))	LIN7 218
	GO TO 168	LIN7 219
166	TIY=100000.0	LIN7 220
	ASSIGN 920 TO N2	LIN7 221
	GO TO 168	LIN7 222
167	TIY=(YBU-YP(J))/(VY(JWAD))	LIN7 223
C		LIN7 224
C	COMPUTE TIZ -- TRANSIT TIME TO Z BOUNDARY	LIN7 225
C 168	IS PARTICLE MOVING UP OR DOWN. UP TO 171	LIN7 226
168	IF(VPZ)169,170,171	LIN7 227
C		LIN7 228
C 169	IS PARTICLE BELOW MAX TOPO HEIGHT. YES TO 1691	LIN7 229
169	IF (ZP(J)-ITOPU)1691,1691,1693	LIN7 230
C		LIN7 231
C	1693 IS MAX TOPO HEIGHT ABOVE SLICE BOTTOM. YES TO 1692	LIN7 232
1693	IF (ITOPU-BOTHIT(JW))1691,1691,1692	LIN7 233
1692	TIZ = (ITOPU-ZP(J))/VPZ	LIN7 234

GO TO 1711	LIN7 235
1691 TI2=(BOTHIT(JW)-ZP(J))/VPZ	LIN7 236
GO TO 1711	LIN7 237
170 TI2=100000.0	LIN7 238
GO TO 1711	LIN7 239
171 TI2=(BOTHIT(JW+1)-ZP(J))/VPZ	LIN7 240
C	LIN7 241
C1711 FIND THE EARLIEST INTERSECTION WITH A LOCAL CIRCULATION SYSTEM	LIN7 242
1711 CIRMIN=TLIMIT	LIN7 243
C ARE THERE ANY LOCAL CIRCULATION SYSTEMS. YES TO 1712	LIN7 244
IF(NLOCIR)172,172,1712	LIN7 245
C	LIN7 246
C 1712 COMPUTE TIME OF FLIGHT TO EACH OF THE FOUR VERTICAL PLANES THAT	LIN7 247
C BOUND THE LJ-TH LOCAL CIRCULATION CELL.	LIN7 248
1712 DO 1713 LJ=1,NLOCIR	LIN7 249
GO TO (917,918),N1	LIN7 250
918 TX1=1000000.0	LIN7 251
TX2=1000000.0	LIN7 252
GO TO 921	LIN7 253
917 TX1=(CRMINX(LJ)-XF(J))/VX(JWAD)	LIN7 254
TX2=(CRMAXX(LJ)-XF(J))/VX(JWAD)	LIN7 255
921 GO TO (919,920),N2	LIN7 256
920 TY1=1000000.0	LIN7 257
TY2=1000000.0	LIN7 258
GO TO 922	LIN7 259
919 TY1=(CRMINY(LJ)-YP(J))/VY(JWAD)	LIN7 260
TY2=(CRMAXY(LJ)-YP(J))/VY(JWAD)	LIN7 261
C	LIN7 262
C TEST X INTERCEPTS	LIN7 263
C IS THE FIRST X DIRECTION INTERCEPT IN THE PAST. YES TO 1714	LIN7 264
922 IF(TX1)1714,1716,1716	LIN7 265
C	LIN7 266
C 1714 IS THE SECOND X DIRECTION INTERCEPT ALSO IN THE PAST. IF YES,	LIN7 267
C BOTH X DIRECTION INTERCEPTS ARE IN THE PAST AND THE PARTICLE WILL	LIN7 268
C NOT INTERSECT THIS CELL. GO TO 1713 TO CONSIDER THE NEXT CELL.	LIN7 269
1714 IF(TX2)1713,1715,1715	LIN7 270
1715 TMINX=TX1	LIN7 271
TMAXX=TX2	LIN7 272
GO TO 1717	LIN7 273
1716 IF(TX2)1716,1719,1719	LIN7 274
1718 TMINX=TX2	LIN7 275
TMAXX=TX1	LIN7 276
GO TO 1717	LIN7 277
C	LIN7 278
C 1719 BOTH X INTERCEPTS ARE IN THE FUTURE	LIN7 279
1719 IF(TX1-TX2)1715,1715,1718	LIN7 280
C	LIN7 281
C 1717 NOW TEST FOR Y INTERCEPTS	LIN7 282
1717 IF(TY1)1720,1721,1721	LIN7 283
1720 IF(TY2)1713,1723,1723	LIN7 284
1723 TMINY=TY1	LIN7 285
TMAXY=TY2	LIN7 286
GO TO 1724	LIN7 287
1721 IF(TY2)1725,1726,1726	LIN7 288
1725 TMINY=TY2	LIN7 289
TMAXY=TY1	LIN7 290
GO TO 1724	LIN7 291
C	LIN7 292

C 1726 BOTH Y INTERCEPTS ARE IN THE FUTURE	LIN7 293
1726 IF(TY1-TY2)1723,1725,1725	LIN7 294
C 1724 NOW SELECT FIRST INTERCEPT	LIN7 295
C	LIN7 296
C SELECT THE SECOND PLANE PIERCE (LAST OF FIRSTX,FIRSTY)	LIN7 297
1724 IF(TMINX-TMINY)1727,1728,1728	LIN7 298
1728 TS=TMINX	LIN7 299
TC=TMAXY	LIN7 300
GO TO 1729	LIN7 301
1727 TS=TMINY	LIN7 302
TC=TMAXX	LIN7 303
1729 IF(TS-TC)1730,1713,1713	LIN7 304
C	LIN7 305
C 1730 KEEP TIME OF EARLIEST INTERCEPT	LIN7 306
1730 IF(TS-CIRMIN)1731,1713,1713	LIN7 307
1731 CIRMIN=TS	LIN7 308
1713 CONTINUE	LIN7 309
C***** * * * * TEMP * * * * *	LIN7 310
WRITE (ISOUT,11)CIRMIN	LIN7 311
C	LIN7 312
C AT THIS POINT CIRMIN CONTAINS THE TIME OF THE FIRST INTERCEPT	LIN7 313
C BETWEEN PARTICLE AND A LOCAL CIRCULATION SYSTEM	LIN7 314
C 172 NOW SELECT EARLIEST BOUNDARY	LIN7 315
172 IF(TIX-TIY)175,175,175	LIN7 316
173 IF(TIX-TI2)176,176,176	LIN7 317
176 TSM=TIX	LIN7 318
IR=1	LIN7 319
GO TO 179	LIN7 320
C	LIN7 321
C 178 SET SMALLEST TIME OF FLIGHT STORAGE AT THE TIME OF FLIGHT TO THE	LIN7 322
C Z BOUNDARY	LIN7 323
178 TSM=TI2	LIN7 324
IR=3	LIN7 325
GO TO 179	LIN7 326
175 IF(TIY-TI2)180,176,176	LIN7 327
180 TSM=TIY	LIN7 328
IR=2	LIN7 329
C	LIN7 330
C 179 IS INTERSECTION WITH LOCAL SYSTEM PRIOR TO EARLIEST BOUNDARY	LIN7 331
C INTERCEPT. YES TO 1792	LIN7 332
179 IF(TSM-CIRMIN)1791,1795,1792	LIN7 333
1792 TSM=CIRMIN	LIN7 334
1793 IR=5	LIN7 335
C	LIN7 336
C 1791 DOES TIME LIMIT COME BEFORE EARLIEST OTHER BOUNDARY. YES TO 182	LIN7 337
1791 IF(TSM+TP(U)-ENDTIM)181,182,182	LIN7 338
C	LIN7 339
C 182 TRANSPORT PARTICLE UNTIL ENDTIM	LIN7 340
182 TSM=ENDTIM-TP(U)	LIN7 341
IR=4	LIN7 342
GO TO 3067	LIN7 343
C	LIN7 344
C 181 TEST FOR EXCESSIVELY SMALL MOVEMENT	LIN7 345
181 GO TO(5580,5570),MPNT	LIN7 346
5570 WRITE (ISOUT,11) TSM,VP2T,VP2,TIY,FV	LIN7 347
5580 IF(TSM-EP5IL)3050,3050,3067	LIN7 348
C	LIN7 349
C SPECIAL TRANSPORT IN THE EVENT OF EXCESSIVELY SMALL TSM	LIN7 350

3050 GO TO IPAS,(3052,3053)	LIN7 351
C	LIN7 352
3052 ASSIGN 3053 TO IPAS	LIN7 353
JWAD1=JWAD	LIN7 354
VPZT=VPZ	LIN7 355
TSM=EPSIL	LIN7 356
GO TO 1811	LIN7 357
3053 IF(VPZT*VPZ)3 54,3055,3055	LIN7 358
3054 VPZ=0.0	LIN7 359
TIZ=TLIMIT	LIN7 360
GO TO 3056	LIN7 361
3055 VPZ=(VPZ+VPZT)/2.0	LIN7 362
C	LIN7 363
C IF VECTORS ARE OF OPPOSITE SIGNS, USE A ZERO	LIN7 364
3056 IF(VX(JWAD)*VX(JWAD1))3057,3058,3058	LIN7 365
3057 VPX=0.0	LIN7 366
TIX=TLIMIT	LIN7 367
GO TO 3059	LIN7 368
3058 VPX=(VX(JWAD)+VX(JWAD1))/2.0	LIN7 369
C	LIN7 370
3059 IF(VY(JWAD)*VY(JWAD1))3060,3061,3061	LIN7 371
3060 VPY=0.0	LIN7 372
TIY=TLIMIT	LIN7 373
GO TO 3062	LIN7 374
C	LIN7 375
3061 VPY=(VY(JWAD)+VY(JWAD1))/2.0	LIN7 376
3062 TAND=ENDTIA-TP(J)	LIN7 377
TSM=ARIN(TIX, TIY, TIZ, CIRMIN, TAND)	LIN7 378
IF(TSM-TAND)3063,3064,3065	LIN7 379
3064 IR=4	LIN7 380
GO TO 3066	LIN7 381
3065 IR=5	LIN7 382
TSM=TSM+EPSIL	LIN7 383
TP(J)=TP(J)+TSM	LIN7 384
C	LIN7 385
3066 XP(J)=XP(J)+VPX*TSM	LIN7 386
YP(J)=YP(J)+VPY*TSM	LIN7 387
ZP(J)=ZP(J)+VPZ*TSM	LIN7 388
GO TO 3063	LIN7 389
3067 ASSIGN 3052 TO IPAS	LIN7 390
C	LIN7 391
C 3051 IS PARTICLE BELOW MAXIMUM TOPO HEIGHT NO TO 1811	LIN7 392
3051 IF(ZP(J)-ITOPU)1812,1812,1811	LIN7 393
1812 GO TO ITT,(188,1813)	LIN7 394
C	LIN7 395
C 1813 TRANSPORT PARTICLE FOR TSM BY STEPS OF DTMAC CHECKING TOPO AS WE	LIN7 396
C GO	LIN7 397
C COMPUTE MOVEMENT INCREMENTS FOR X,Y, AND Z DIRECTIONS	LIN7 398
1813 XIN=DTMAC*VX(JWAD)	LIN7 399
YIN=DTMAC*VY(JWAD)	LIN7 400
ZIN=DTMAC*VPZ	LIN7 401
1814 XP(J)=XP(J)+XIN	LIN7 402
YP(J)=YP(J)+YIN	LIN7 403
ZP(J)=ZP(J)+ZIN	LIN7 404
TP(J)=TP(J)+DTMAC	LIN7 405
TSM=TSM-DTMAC	LIN7 406
C	LIN7 407
C TEST FOR PARTICLE IMPACT ON TOPOGRAPHY	LIN7 408

X=XP(J)	LIN7 409
Y=YP(J)	LIN7 410
CALL HEIGHT(X,Y,H)	LIN7 411
IF(H+20000.0)1872,1872,1875	LIN7 412
1815 IF(H+10000.0)1875,1875,1816	LIN7 413
1816 IF(H-ZP(J))1817,188,188	LIN7 414
1817 IF(TSM)189,189,1814	LIN7 415
C	LIN7 416
C 1811 TRANSPORT PARTICLE FOR TSM	LIN7 417
C FIRST INCREASE TSM TO ASSURE THAT THE PARTICLE WILL ACHIEVE ITS	LIN7 418
C BOUNDARY	LIN7 419
1811 TSM=TSM*1.000 01	LIN7 420
XP(J)=XP(J)+VX(JWAD)*TSM	LIN7 421
YP(J)=YP(J)+VY(JWAD)*TSM	LIN7 422
ZP(J)=ZP(J)+VPZ*TSM	LIN7 423
TP(J)=TP(J)+TSM "	LIN7 424
3063 CONTINUE	LIN7 425
C	LIN7 426
C ARE TRANSPORT TRACES TO BE WRITTEN.. YES TO 5550	LIN7 427
GO TO (5560,5550),MPNT	LIN7 428
5550 WRITE(1500,12)XP(J),YP(J),ZP(J),TP(J),TSM,NTI,NG,NTG,NW,NLOST,	LIN7 429
1IR	LIN7 430
C	LIN7 431
C TEST FOR PARTICLE IMPACT ON TOPOGRAPHY	LIN7 432
5560 IF(ZP(J)-ITOP0)187,187,189	LIN7 433
C	LIN7 434
C 189 PARTICLE IS NOT G-OUNDED. ADJUST INDICES JW AND JWAD AND THEN	LIN7 435
C RECYCLE.	LIN7 436
189 GO TO (250,250,252,189,189),IR	LIN7 437
159 NTI=NTI+1	LIN7 438
TP(J)=ENDTIM	LIN7 439
GO TO 160	LIN7 440
250 JW=-JW	LIN7 441
GO TO 1950	LIN7 442
252 IF(VZ(JWAD))258,258,259	LIN7 443
258 JW=1-JW	LIN7 444
GO TO 1961	LIN7 445
259 JW=-(JW+1)	LIN7 446
272 GO TO 1961	LIN7 447
187 GO TO IF,(186,1871)	LIN7 448
1871 X=XP(J)	LIN7 449
Y=YP(J)	LIN7 450
CALL HEIGHT(X,Y,H)	LIN7 451
IF(H+20000.0)1872,1872,1874	LIN7 452
1874 IF(H+10000.0)1875,1875,1876	LIN7 453
C	LIN7 454
C 1872 F=-20000.0 PARTICLE BEYOND SPECIFIED TOPO	LIN7 455
1872 FMAS(J)=-FMAS(J)	LIN7 456
TP(J)=TLIMIT	LIN7 457
NLOST=NLOST+1	LIN7 458
GO TO 160	LIN7 459
C	LIN7 460
C 1875 H=-10000.0 PARTICLE BEYOND IN-CORE TOPO	LIN7 461
1875 TP(J)=-TP(J)	LIN7 462
NTG=NTG+1	LIN7 463
IF(JTOP1)180,1877,180	LIN7 464
1877 JTOP1=-1	LIN7 465
GO TO 160	LIN7 466

1876 IF(H-ZP(J))189,188,188	LIN7 467
C	LIN7 468
C 188 TAKES CARE OF GROUNDED PARTICLES	LIN7 469
188 FMAS(J)=-FMAS(J)	LIN7 470
TP(J)=-TP(J)	LIN7 471
NG=NG+1	LIN7 472
160 CONTINUE	LIN7 473
C	LIN7 474
C *****	LIN7 475
C	LIN7 476
C END OF MAIN TRANSPORT LOOP	LIN7 477
C	LIN7 478
C *****	LIN7 479
C	LIN7 480
IF(JDONE-1)1005,100,1000	LIN7 481
C	LIN7 482
C ARE ANY PARTICLES ON THE OFF THE TOPO TAPE	LIN7 483
100 IF(JTOP1)103,104,103	LIN7 484
C JTOP1 OR ZERO INDICATES SOME PARTICLES ARE ON OFF TOPO TAPE	LIN7 485
C A NEGATIVE JTOP1 INDICATES PARTICLES IN OFF TOPO BUFFER BUT NOT	LIN7 486
C ON THE OFF TOPO TAPE	LIN7 487
C	LIN7 488
C ARE ANY PARTICLES ON THE OUT OF WIND FIELD TAPE	LIN7 489
104 IF(JWIND1)130,200,130	LIN7 490
C 200 IS PARTICLES ALOFT TIME BOUNDARY TAPE IN USE 33	LIN7 491
200 IF(JTIME1,203,203,201	LIN7 492
203 IF(NT1)2021,2031,2021	LIN7 493
2021 JTIME1=-1	LIN7 494
JDONE = 1	LIN7 495
GO TO 202	LIN7 496
2031 ENDTIME=ELIMIT	LIN7 497
GO TO 202	LIN7 498
201 WRITE(IPAROT)AOL	LIN7 499
REWIND IPAROT	LIN7 500
REWIND IPARIN	LIN7 501
ITEMP=IPARIN	LIN7 502
IPARIN =IPAROT	LIN7 503
IPAROT=ITEMP	LIN7 504
JTIME1=0	LIN7 505
IF(NT1)2022,202,2022	LIN7 506
2022 JTIME1=-1	LIN7 507
202 IEXEC=2	LIN7 508
RETURN	LIN7 509
C	LIN7 510
C A NEGATIVE JWIND1 INDICATES SOME PARTICLES ARE IN THE OUT OF THE	LIN7 511
C WIND FIELD BUFFER BUT NOT ON TAPE	LIN7 512
C JWIND1 GREATER THAN ZERO INDICATES PARTICLES ARE ON THE OUT OF	LIN7 513
C THE WIND FIELD TAPE	LIN7 514
C	LIN7 515
C	LIN7 516
C 105 GET THE REQUIRED TOPO DATA FROM TAPE	LIN7 517
C 105 TO 107 SCANS BUFFER PARTICLE COUNTERS TO DETERMINE NEXT NEEDED	LIN7 518
C TOPO DATA BLOCKS AND CHOOSES THE NEAREST ONE FOR READING.	LIN7 519
C	LIN7 520
C NBLCK IS SET BY THE INITIALIZING PROGRAM WHICH READS THE TOPO	LIN7 521
C TAPES IDENTIFICATION RECORDS	LIN7 522
C NINTAR(J) IS SET BY THE TRANSPORT LOOP WHEN PARTICLES LEAVE TOPO	LIN7 523
C AND RESET WHEN OFF-TOPO BUFFER IS EMPTIED.	LIN7 524

C		LIN7 525
105	JTEST=1000	LIN7 526
	DO 107 J=1,NBLCK	LIN7 527
	JTEST1=NINJAR(J)	LIN7 528
	IF(JTEST1)107,107,108	LIN7 529
108	JTEST2=JFTOPU-J+1	LIN7 530
	IF(JTEST2)109,110,111	LIN7 531
110	ERROR=-110	LIN7 532
C		LIN7 533
C	SEEKS THE FILE IN CORE. SHOULD NEVER HAPPEN. GO TO A STOP EXIT.	LIN7 534
	GO TO 300	LIN7 535
111	IF(JTEST2-JTEST)112,107,107	LIN7 536
112	JTEST=JTEST2	LIN7 537
	GO TO 140	LIN7 538
109	IF(JTEST+JTEST2)107,107,113	LIN7 539
113	JTEST=-JTEST2	LIN7 540
140	JF=J	LIN7 541
107	CONTINUE	LIN7 542
C		LIN7 543
C	AT THIS POINT JF HAS THE NUMBER OF THE DESIRED FILE	LIN7 544
C	NOW MOVE TAPE TO SELECTED FILE	LIN7 545
	IF(JF-JFTOPU)1072,1071,1071	LIN7 546
C		LIN7 547
C	1071 PREPARE TO MOVE FORWARD ON INTOPU. COMPUTE NUMBER OF BLOCKS TO	LIN7 548
C	READ IN	LIN7 549
	1071 JR=JF-JFTOPU+1	LIN7 550
	GO TO 1074	LIN7 551
C		LIN7 552
C	1072 DESIRED FILE IS BEHIND READ HEAD. BACK UP TO GET IT.	LIN7 553
	1072 REWIND INTOPU	LIN7 554
C		LIN7 555
C	NOW SKIP OVER INITIAL RECORDS	LIN7 556
	READ (INTOPU)1ST	LIN7 557
	READ (INTOPU)3ST,1ST,1ST,1ST,1ST	LIN7 558
	READ (INTOPU)TOPULM	LIN7 559
	READ (INTOPU)ITOPULM	LIN7 560
	JR=JF	LIN7 561
C		LIN7 562
C	1074 NOW READ UP THROUGH THE DESIRED BLOCK	LIN7 563
	1074 DO 1075 J=1,JR	LIN7 564
	1075 CALL ROTOPU(J)	LIN7 565
C		LIN7 566
C	116 RESET ALL OFF-TOPU PARTICLES	LIN7 567
116	DO 118 J=1,NALOFT	LIN7 568
	IF(FMAS(J))117,118,118	LIN7 569
117	IF(IP(J))119,118,118	LIN7 570
119	IF(IP(J)-LIMIT)120,118,120	LIN7 571
120	FMAS(J)=-FMAS(J)	LIN7 572
	IP(J)=-IP(J)	LIN7 573
118	CONTINUE	LIN7 574
	IF=NALOFT	LIN7 575
	IF(JTOP1)1151,115,115	LIN7 576
1151	JTOP1=J	LIN7 577
	JTIME1=-1	LIN7 578
	GO TO 1001	LIN7 579
C		LIN7 580
C	115 REWIND OFF TOPU AND PARTICLES ALOFT TAPES AND SWAP NAMES	LIN7 581
115	WRITE(IOTOPU)NUL	LIN7 582

```

      REWIND IOTUPO
      REWIND IPARIN
      ITEMP=IOTUPO
      IOTUPO=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,NALJFT
      IF(FMAS(J)) 1241,122,122
1241 IF(TP(J))122,122,1242
1242 IF(TP(J)-LIMIT)1243,122,1243
1243 FMAS(J)=FMAS(J)
122 CONTINUE
      IF=NALJFT
      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,15)NOL
      JWIND1=0
      REWIND IOWIND
      REWIND IPARIN
      ITEMP=IOWIND
      IOWIND=IPARIN
      IPARIN=ITEMP
      GO TO 1001
C
      END

```

```

LIN7 583
LIN7 584
LIN7 585
LIN7 586
LIN7 587
LIN7 588
LIN7 589
LIN7 590
LIN7 591
LIN7 592
LIN7 593
LIN7 594
LIN7 595
LIN7 596
LIN7 597
LIN7 598
LIN7 599
LIN7 600
LIN7 601
LIN7 602
LIN7 603
LIN7 604
LIN7 605
LIN7 606
LIN7 607
LIN7 608
LIN7 609
LIN7 610
LIN7 611
LIN7 612
LIN7 613
LIN7 614
LIN7 615
LIN7 616
LIN7 617

```

616*

618 *

```

$IBFTC GETWN LIST,DECK,M94/2                                GETW
SUBROUTINE GETWND(XX,YY,ZZ,JWAD,JW)                          GETW  1
C 28 NOVEMBER 1966                                           GETW  2
C T.W. SCHWENKE TECHNICAL OPERATIONS RESEARCH                GETW  3
C                                                             GETW  4
C *****                                                    GETW  5
C THIS SUBROUTINE RETRIEVES THE MACRO WIND FIELD VECTOR WHICH GETW  6
C APPLIES AT THE POINT WHOSE COORDINATES ARE IN THE ARGUMENT WORDS GETW  7
C XX,YY, ANDZZ AT ENTRANCE. THE X,Y, AND Z WIND VECTOR COMPONENTS GETW  8
C ARE IN THE COMMON VARIABLES VX(JWAD),VY(JWAD), AND VZ(JWAD) WHEN GETW  9
C THE SUBROUTINE RETURNS. IF AN ENTRANCE IS MADE WITH ARGUMENT JW GETW 10
C SET NEGATIVE, JW IS SET POSITIVE AND ITS VALUE IS USED RATHER THAN GETW 11
C RECOMPUTED. UPON EXIT JWAD IS SET NEGATIVE IN THE EVENT THAT THE GETW 12
C MACRO WIND FIELD PERTAINING TO THE DESIRED POINT IS NOT AVAILABLE GETW 13
C IN CORE.                                                    GETW 14
C                                                             GETW 15
C ***** GLOSSARY ***** GETW 17
C JW INDEX OF THE WIND ARRAY STRATA IN THE MACRO WIND FIELD. GETW 18
C THIS INDEX INCREASES FROM BOTTOM TO TOP OF THE FIELD. GETW 19
C IF NEGATIVE AT ENTRANCE, IT IS SET POSITIVE AND USED BY GETW 20
C GETWND. IF ZERO OR POSITIVE AT ENTRANCE, IT IS RE- GETW 21
C COMPUTED BY GETWND. GETW 22
C JWAD AN INDEX FOR THE STORAGE ARRAYS (ONE DIMENSIONAL) FOR GETW 23
C MACRO WIND FIELD VECTORS. JWAD IS SET NEGATIVE BY GETW 24
C GETWND IF THE NEEDED MACRO WIND FIELD VECTORS ARE NOT GETW 25
C IN CORE. GETW 26
C JI INDEX OF WIND LAYER USED IN A SEARCH FOR THE LAYER CONTAINING GETW 27
C THE J-IH PARTICLE. TOP INDEX GETW 28
C JIOPS INDEX OF THE TOP LAYER IN THE MACRO WIND FIELD DESCRIPTION GETW 29
C TION GETW 30
C JISI TEMPORARY STORAGE GETW 31
C ZZ Z COORDINATE OF THE POINT FOR WHICH A MACRO WIND FIELD GETW 32
C VECTOR IS SOUGHT GETW 33
C XX X COORDINATE. SEE ZZ GETW 34
C YY Y COORDINATE. SEE ZZ GETW 35
C IRROR NUMBER OF THE STATEMENT NEAR THE POINT WHERE AN ERROR GETW 36
C WAS DISCOVERED GETW 37
C WELX(JW) LOWER LIMIT FOR X COORDINATES IN THE MACRO WIND FIELD GETW 38
C WELY(JW) LOWER LIMIT FOR Y COORDINATES IN THE MACRO WIND FIELD GETW 39
C WORX(JW) UPPER LIMIT FOR X COORDINATES IN THE MACRO WIND FIELD GETW 40
C WORY(JW) UPPER LIMIT FOR Y COORDINATES IN THE MACRO WIND FIELD GETW 41
C I1W X DIRECTION WIND FIELD RETRIEVAL INDEX GETW 42
C J1W Y DIRECTION WIND FIELD RETRIEVAL INDEX GETW 43
C WDIRIV(JW) GRID INTERVAL FOR THE WIND FIELD IN JW-TH STRATUM IN GETW 44
C METERS GETW 45
C I1(JW) THE NUMBER OF GRID DIVISIONS IN THE X DIRECTION OF THE GETW 46
C WIND FIELD IN STRATUM JW GETW 47
C IBADE(JW) BASE ADDRESS FOR STORING DATA FROM THE JW-TH STRATUM OF GETW 48
C THE WIND FIELD. THIS IS AN INDEX IN THE 1-D WIND ARRAY GETW 49
C ***** GETW 50
C ***** GETW 51
C ***** GETW 52
C ***** GETW 53
C ***** GETW 54
COMMON /SET1/
1 DIAM , DEID , IRIS , IEXE , ISIN , ISOT , GETW 55
2 SD , SPAR , CSAM , IME , IMPI , IMPZ , GETW 56
3 IZM , J , VPR , W , X , Z , GETW 57
4 WHY , RMIN , IDISTK , SPAR1 , SPAR2 , SPAR3 , GETW 58
5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 , GETW 59
DIMENSION DETID(12),WHY(40) GETW 60

```

```

C
C *****
C
COMMON /SET2/
1      S      , SUBSID , GRINT , BALL , BALS , BYLL , GETW 61
2      BYLU , TXLL , TXLU , TYLL , TYLU , XGL , GETW 62
3      YGZ , NBULK , HTOPU , ITOPU , ILIN , ULIN , GETW 63
4      KLIM , II , JJ , KK , KK , KK , KK , GETW 64
5      ZP , FMAS , IP , PS , VX , VY , GETW 65
6      VZ , IL , JL , IBADU , WGRINT , NGRINT , GETW 66
7      WLLX , WLLY , WOKX , WOKY , BUTHIT , IPARIN , GETW 67
8      IOTOPU , IOWIND , IHTOPU , IPOUT , IPAROT , IOTOPU , GETW 68
9      JWIND1 , IRROR , ILLIMIT , ENDITL , IC , IOTOPU , GETW 69
1     IOTOPU , NLOSI , NG , NTO , NTH , NW , GETW 70
2     NALCOT , JTIME1 , NGRAX , NGRY , NGL , GETW 71
3     CRMAXY , CROHI , NCRITP , BZ , CRMINX , CRMINY , GETW 72
4     SS , SN , CS , NLOCIR , STECC , ATERP , GETW 73
5     RHO , RA , IGZ , JTIME , PROG , CRMAXX , GETW 74
6     REPART
1     DIMENSION TOPPLN(4,4) , NINTAX(4) , ITOPPLN(3,4) , GETW 75
2     DIMENSION S(10,10) , SUBSID(400) , IC(18) , GETW 76
3     DIMENSION XP(200) , TP(200) , ZP(200) , FMAS(200) , GETW 77
4     DIMENSION IP(200) , PS(200) , ATERP(200) , RHO(200) , GETW 78
5     DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70) , GETW 79
6     DIMENSION UL(70) , IBADU(70) , OKX(70) , GETW 80
7     DIMENSION WGRINT(70) , WLLX(70) , WLLY(70) , GETW 81
8     DIMENSION WOKY(70) , BUTHIT(70) , SN(6) , CS(6) , GETW 82
9     DIMENSION CRMINX(6) , CRMAXX(6) , CRMINY(6) , CRMAXY(6) , GETW 83
10    DIMENSION CROHI(6) , NCRITP(6) , SS(6) , GETW 84
C
C *****
C
DATA PRGR476HGETWND/
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
AND STORE IT IN JW. USE A TWO-BOUNDED BINARY SEARCH.
102 JT=JTOPJ+1
JW=1
C
C 103 COMPUTE TRIAL INDEX NUMBER
103 JTST=(JT+JW)/2
C HAVE TRIAL TOP AND BOTTOM INDICES CONVERGED TO INDICATE THE
DESIRED LAYER. NO TO 104
104 IF(JT-JW-1)105,270,104
C 104 IS PARTICLE ABOVE THE BOTTOM OF THE TRIAL LAYER. NO TO 102
104 IF(ZZ-BUTHIT(JTST))102,105,105
C 105 PARTICLE IS IN OR ABOVE SLICE JTST.
105 JW=JTST

```

```

      GO TO 155
C 152 PARTICLE IS BELOW SLICE JTSI
      JTSI=JTSI
      GO TO 153
      155 ERROR=-155
      7734 CALL ERROR(PROGRAM,ERROR,IJOUT)
C
C
C 270 IS THE PARTICLE WITHIN THE SPECIFIED WIND FIELD.      GO TO 271
      270 IF(XA-MLEA(JW))271,272,272
C 271 MARK PARTICLE BEYOND SPECIFIED WIND FIELD
      271 JWAQ=-1
      GO TO 160
      272 IF(XA-MLEA(JW))273,273,271
      273 IF(YI-MLEI(JW))271,273,273
      275 IF(YI-MLEI(JW))274,274,271
C
C 274 IS SPECIFIED WIND FIELD IN CORE      REMARKED IN THIS PROGRAM
      274 CONTINUE
C 276 NOW COMPUTE JWAQ = THE INDEX OF THE NEAREST WIND CELL
      276 IIA= (XA-MLEA(JW))/MSX(JW)
      JWAQ= (YI-MLEI(JW))/MSY(JW)
      JWAQ=IBAD(JWAQ+(JWAQ-1)*IC(JW)+1,IA
      160 RETURN
      END

```

```

GETW 119
GETW 120
GETW 121
GETW 122
GETW 123
GETW 124
GETW 125
GETW 126
GETW 127
GETW 128
GETW 129
GETW 130
GETW 131
GETW 132
GETW 133
GETW 134
GETW 135
GETW 136
GETW 137
GETW 138
GETW 139
GETW 140
GETW 141
GETW 142
GETW 143

```

144*

144 *

Reproduced from
best available copy. 

```

5IBFTC LUTRA LIST, DECK, 99472                                LUTR
SUBROUTINE LUTRA(N)                                           LUTR  1
C 11 OCT 66                                                  LUTR  2
C T.W. SCHWENKE TECHNICAL OPERATIONS RESEARCH              LUTR  3
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX LUTR  4
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX LUTR  5
C THIS IS THE EXECUTIVE ROUTINE TO TRANSPORT PARTICLES WITH LUTR  6
C ALL LOCAL WIND SYSTEMS.                                     LUTR  7
C IF THE 0TH PARTICLE IS ABOVE THE TOP OF THE LOCAL CIRCULATION LUTR  8
C CELL, LUTRA TRANSPORTS IT TO THE WIND VECTORS ASSIGNED TO THE LUTR  9
C MACRO WIND FIELD ARRAYS. FOR PARTICLES ASSIGNED WITHIN THE LOCAL LUTR 10
C CIRCULATION CELL IT CIRCLES THE APPROPRIATE LOCAL CIRCULATION CELL LUTR 11
C TO OBTAIN AN ESTIMATE OF THE WIND VECTOR AT PARTICLE POSITION. LUTR 12
C IN EITHER CASE THE PARTICLE IS TRANSPORTED FOR ONE TIME STEP OF LUTR 13
C DURATION OF  $\Delta T$  SECONDS. THIS PROCEDURE IS REPEATED UNTIL THE LUTR 14
C PARTICLE LEAVES THE LOCAL CELL OR ENDS UP IN THE WIND FIELD. LUTR 15
C DESCRIBED TOPOGRAPHY OR UNTIL THE WIND VECTOR FOR SYSTEM IS THE LUTR 16
C ENTIRE WIND FIELD IS REACHED. PARTICLES ARE FREE TO ENTER THE LOCAL LUTR 17
C CELL FROM ABOVE DURING TRANSPORT.                          LUTR 18
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX LUTR 19
C XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX LUTR 20
C RA X COMPONENT OF A WIND VECTOR RETRIEVED FROM A LOCAL LUTR 21
C WIND FIELD DESCRIBING A WIND VECTOR. LUTR 22
C AY Y COMPONENT - SEE RA LUTR 23
C AZ Z COMPONENT - SEE RA LUTR 24
C CRN(X) COORDINATE OF THE WESTERN BOUNDARY OF PARTICLE FOR THE KTH LUTR 25
C LOCAL CIRCULATION SYSTEM CELL LUTR 26
C CRN(X) SEE CRN(X) + SOUTHERN BOUNDARY LUTR 27
C CRN(X) SEE CRN(X) + NORTHERN BOUNDARY LUTR 28
C CRN(X) SEE CRN(X) + EASTERN BOUNDARY LUTR 29
C CRN(X) ALTITUDE (METERS ABOVE SEA) OF THE TOP BOUNDARY OF THE LUTR 30
C KTH LOCAL CIRCULATION CELL LUTR 31
C DTLOC TIME STEP DURATION (SECONDS) FOR TRANSPORT WITHIN EACH OF LUTR 32
C LOCAL CIRCULATION CELLS LUTR 33
C ENDIF TIME FOR THE NEXT OPERATION OF THE WIND FIELD. SECONDS LUTR 34
C FV SCALING RATE (ABSOLUTE VALUE IN PERCENTAGE) OF THE LUTR 35
C 0TH PARTICLE AS COMPUTED BY SUBROUTINE FAKRA LUTR 36
C IEXRA STATEMENT NEAR THE POINT WHERE A LEXON WAS DISCOVERED LUTR 37
C I INDEX FOR PARTICLE DESCRIPTIONS LUTR 38
C J INDEX OF THE WIND ARRAY STRAITS IN THE MACRO WIND FIELD. LUTR 39
C THIS INDEX INCREASES FROM BOTTOM TO TOP OF THE FIELD. LUTR 40
C IF NEGATIVE AT ENTRANCE, IT IS SET POSITIVE AND USED BY LUTR 41
C GETWIND. IF ZERO OR POSITIVE AT ENTRANCE, IT IS SET LUTR 42
C COMPUTED BY GETWIND. LUTR 43
C JWL J INDEX FOR THE STORAGE ARRAYS (ONE DIMENSIONAL) FOR LUTR 44
C MACRO WIND FIELD VECTORS. JWD IS SET NEGATIVE BY LUTR 45
C GETWIND IF THE NEEDED MACRO WIND FIELD VECTORS ARE NOT LUTR 46
C IN CORE. LUTR 47
C JWF BOUNDARY INDEX OF THE BOUNDARY BEYOND LUTR 48
C N ARGUMENT OF LUTRA THAT COMMUNICATES THE INDEX OF THE LUTR 49
C LOCAL SYSTEM CELL IN OR ABOVE WHICH THE 0TH PARTICLE LUTR 50
C IS LOCATED WHEN LUTRA IS CALLED LUTR 51
C NC A SEARCH POINT VARIABLE FOR USE WITH AN ASSIGNED GO TO LUTR 52
C NC THAT EFFICIENTLY TRANSPORTS TO THE APPROPRIATE LOCAL LUTR 53
C CIRCULATION SYSTEM PROGRAM DURING EXECUTION OF THE LUTR 54
C LOCAL TRANSPORT LOOP LUTR 55
C NCIR TEMPORARY NON-DECLARED STORAGE FOR LOCAL CIRCULATION LUTR 56
C SYSTEM TYPE NUMBER LUTR 57
C NCRTYPEK TYPE DESIGNATION FOR THE KTH LOCAL CIRCULATION SYSTEM LUTR 58

```

C		SEE THE EXPLANATION OF TYPE NUMBERS BELOW.	LOTR	61
C		1 MTWIND1	LOTR	62
C		2 RGWIND1	LOTR	63
C		3 CBREZ1	LOTR	64
C		4 NOT ASSIGNED	LOTR	65
C		5 NOT ASSIGNED	LOTR	66
C	ICELLT	SEE IXX. FOR THE PLANE DEFINING LOCAL CELL TOP	LOTR	67
C	ILAYER	SEE IXX. FOR THE HORIZONTAL PLANES BOUNDING THE MACRO	LOTR	68
C		WIND FIELD LAYER THAT THE PARTICLE IS IN	LOTR	69
C	IF(U)	TIME COORDINATE FOR THE U-TH PARTICLE	LOTR	70
C		SINCE DESIGNATION	LOTR	71
C	IXX	TIME OF FLIGHT TO NEXT (IN TIME) INTERCEPT BETWEEN	LOTR	72
C		PARTICLE TRAJECTORY AND AN ACONSTANT PLANE BOUNDING	LOTR	73
C		THE LOCAL WIND CELL	LOTR	74
C	IYY	SEE IXX. FOR A Y-BOUNDARY PLANE	LOTR	75
C	VPZ	VERTICAL PARTICLE VELOCITY, M/SEC.	LOTR	76
C	VX(I)	X COMPONENT ARRAY OF THE MACRO WIND FIELD DESCRIPTION	LOTR	77
C	VY(I)	Y COMPONENT - SEE VX(I)	LOTR	78
C	VZ(I)	Z COMPONENT - SEE VX(I)	LOTR	79
C	XP(I)	X COORDINATE OF THE U-TH PARTICLE. A COMMON ARRAY	LOTR	80
C	XX	X COORDINATE OF THE POINT FOR WHICH DELTAX IS TO SET	LOTR	81
C		THE APPLICABLE MACRO WIND VECTOR	LOTR	82
C	YP(U)	Y COORDINATE OF THE U-TH PARTICLE. A COMMON ARRAY	LOTR	83
C	YY	Y COORDINATE - SEE XX	LOTR	84
C	ZP(U)	Z COORDINATE OF THE U-TH PARTICLE. A COMMON ARRAY	LOTR	85
C		ALTITUDE OF U-TH PARTICLE IN METERS ABOVE GLL	LOTR	86
C	ZZ	Z COORDINATE - SEE XX	LOTR	87
C			LOTR	88
C			LOTR	89
C			LOTR	90
C			LOTR	91
C			LOTR	92
C			LOTR	93
C			LOTR	94
C			LOTR	95
C			LOTR	96
C			LOTR	97
C			LOTR	98
C			LOTR	99
C			LOTR	100
C			LOTR	101
C			LOTR	102
C			LOTR	103
C			LOTR	104
C			LOTR	105
C			LOTR	106
C			LOTR	107
C			LOTR	108
C			LOTR	109
C			LOTR	110
C			LOTR	111
C			LOTR	112
C			LOTR	113
C			LOTR	114
C			LOTR	115
C			LOTR	116
C			LOTR	117
C			LOTR	118

C	COMMON /SUB17/									
1	DIAT	DEIID	TRICE	TEACE	ICIN	ISOUT				
2	CD	SPAR	SHAT	TE	ICIN	ISOUT				
3	IZ	C	VPR	W	A	Z				
4	AMT	KMIN	IDISIX	SPAX1	SPAX2	SPAX3				
5	SPAX4	SPAX5	SPAX6	SPAX7	SPAX8	SPAX9				
	DIMENSION DEIID(12),AMT(40)									

C	COMMON /SUB12/									
1	Q	SUBSID	GRINT	EXEL	EXEL	BYEL				
2	FILE	IXEL	IXEL	IYEL	IYEL	AGE				
3	ISZ	WELK	HICPS	ITOPC	ILIN	ULIN				
4	KELT	II	IS	KK	XP	YP				
5	ZP	EXHS	IE	ES	VX	VY				
6	VE	IE	SE	ISADD	WOKINI	NSIRAI				
7	ALCA	WELT	WORA	WORT	SOHIT	IPARIN				
8	ITOPC	ISWIND	ITOPC	ISOUT	IFAROT	STCHI				
9	SWIND1	ISWIND	ITOPC	ISOUT	IFAROT	STCHI				
10	STOPC	ISWIND	ITOPC	ISOUT	IFAROT	STCHI				
11	STOPC	ISWIND	ITOPC	ISOUT	IFAROT	STCHI				
12	VALOFF	STLINE1	KSMAX	NEKEE	N	NCL				
13	CRHAT	CRONI	NRKTE	BZ	CRMINX	CRMINY				
14	CS	SN	CS	NECCIR	DLOC	ATEMP				
15	KHO	NA	ISZ	DIAC	FRGB	CRMAXX				
16	KSPART									
	DIMENSION FORCEL(4,4) ,MINIAR(4) ,ITOPEN(3,4)									
	DIMENSION STCHI(10,10) ,SUBSID(400) ,IC(18)									

DIMENSION XP(200)	,YP(200)	,ZP(200)	,FHAS(200)	LOTR 117
DIMENSION TP(200)	,PS(200)	,ATEMP(200)	,RHU(200)	LOTR 120
DIMENSION VX(1500)	,VY(1500)	,VZ(1500)	,IL(70)	LOTR 121
DIMENSION JL(70)	,IBADD(70)	,WORA(70)		LOTR 122
DIMENSION WGRINT(70)		,WELX(70)	,WELLY(70)	LOTR 123
DIMENSION WGRY(70)	,BOTHT(70)	,BN(6)	,CB(6)	LOTR 124
DIMENSION CRMINX(6)	,CRMAXX(6)	,CRMINY(6)	,CRMAXY(6)	LOTR 125
DIMENSION CROHT(6)	,NCRTYP(6)	,CC(6)		LOTR 126
C				LOTR 127
C *****				LOTR 128
C				LOTR 129
C DATA PROGRAM/6HELOTRAN/				LOTR 130
C				LOTR 131
C *****				LOTR 132
C *****				LOTR 133
C				LOTR 134
C FOR LOCAL CIRCULATION TYPE NUMBER IN TEMPORARY STORAGE				LOTR 135
C NCIR=NCRTYP(K)				LOTR 136
C MAKE AN ASSIGNMENT TO ALLOW AN EFFICIENT				LOTR 137
C BRANCH TO THE APPROPRIATE LOCAL CIRCULATION TRANSPORT CODE				LOTR 138
C IN THE ACTUAL LOCAL TRANSPORT LOOP				LOTR 139
C GO TO (101,102,103,104,105),NCIR				LOTR 140
101 ASSIGN 121 TO NC				LOTR 141
GO TO 120				LOTR 142
102 ASSIGN 122 TO NC				LOTR 143
GO TO 120				LOTR 144
103 ASSIGN 123 TO NC				LOTR 145
GO TO 120				LOTR 146
104 ASSIGN 124 TO NC				LOTR 147
GO TO 120				LOTR 148
105 ASSIGN 125 TO NC				LOTR 149
C				LOTR 150
C 120 BEGIN THE LOCAL TRANSPORT LOOP.				LOTR 151
C FIRST DETERMINE IF THE PARTICLE IS BELOW THE LEVEL OF THE TOP				LOTR 152
C OF THE K-TH LOCAL CIRCULATION SYSTEM CELL. IF IT IS NOT, CALL				LOTR 153
C GETIND TO GET THE MACRO WIND VECTOR AT PARTICLE POSITION AND THEN				LOTR 154
C MOVE THE PARTICLE BY TRANSFERRING TO 130. IF PARTICLE IS WITHIN THE				LOTR 155
C LOCAL CELL, A BRANCH MUST BE MADE TO THE APPROPRIATE SUBROUTINE.				LOTR 156
120 ZZ=ZP(J)				LOTR 157
PSIZE=PS(J)				LOTR 158
CALL FALCAT(ZZ,PSIZE,FV,ATEMP,RHU,PROG,ISOUT)				LOTR 159
IF(ZP(J)-CROHT(K))1202,1202,1201				LOTR 160
C 1201 PARTICLE IS ABOVE LOCAL CELL				LOTR 161
1201 XX = XP(J)				LOTR 162
YY = YP(J)				LOTR 163
ZZ = ZP(J)				LOTR 164
CALL GETWAD(XX,YY,ZZ,UWAD,UW)				LOTR 165
C WAS THE MACRO WIND FIELD SPECIFICATION AVAILABLE FOR THE PARTICLE				LOTR 166
C POSITION. NO TO 1203				LOTR 167
IF(UWAD)1203,1203,1204				LOTR 168
C 1203 MACRO WIND FOR THIS PARTICLE IS NOT AVAILABLE.				LOTR 169
1203 IRROR = 1203				LOTR 170
GO TO 7734				LOTR 171
C				LOTR 172
C 1204 COMPUTE VERTICAL COMPONENT OF PARTICLE VELOCITY				LOTR 173
1204 VPZ=VZ(UWAD)-FV				LOTR 174
C				LOTR 175
C COMPUTE TIMES OF FLIGHT TO ALL BOUNDARY INTERCEPTS THAT ARE IN THE				LOTR 176

C	FUTURE. FIRST COMPUTE TIME TO COMING X BOUNDARY OF LOCAL CELL	LOTR 177
	IF(VX(JWAD))1205,1206,1206	LOTR 178
1205	TXX=(XP(J)-CRMIX(K))/VX(JWAD)	LOTR 179
	GO TO 1207	LOTR 180
1206	TXX=(CRMXX(K)-XP(J))/VX(JWAD)	LOTR 181
C		LOTR 182
C	1207 COMPUTE TIME TO COMING Y BOUNDARY INTERCEPT	LOTR 183
1207	IF(VY(JWAD))1208,1209,1209	LOTR 184
1208	TTY=(YP(J)-CRMINY(K))/VY(JWAD)	LOTR 185
	GO TO 1210	LOTR 186
1209	TTY=(CRMAY(K)-YP(J))/VY(JWAD)	LOTR 187
C		LOTR 188
C	1210 COMPUTE TIMES TO COMING Z BOUNDARY (HORIZONTAL) OF MACRO WIND	LOTR 189
C	FIELD AND TO THE TOP OF THE LOCAL WIND CELL	LOTR 190
1210	IF(VPZ)1211,1212,1212	LOTR 191
1211	ICELLT=(CRGHT(K)-ZP(J))/VPZ	LOTR 192
	JWT=JW	LOTR 193
	GO TO 1213	LOTR 194
1212	ICELLT=1.0E+08	LOTR 195
	JWT=JW+1	LOTR 196
1213	FLAYER=(BOGHT(JWT)-ZP(J))/VPZ	LOTR 197
C		LOTR 198
C	COMPUTE TIME UNTIL TIME FOR UPDATING THE WIND FIELD	LOTR 199
	ITIME=ENDTIM-TP(J)	LOTR 200
C		LOTR 201
C	NOW SELECT THE TIME UNTIL THE FIRST VALID BOUNDARY INTERCEPT	LOTR 202
C	ADD A SMALL INCREMENT TO PUSH THE PARTICLE PAST THE BOUNDARY OF	LOTR 203
C	THE LOCAL CELL.	LOTR 204
	ITRANS=AMINI(TXX,TTY,ICELLT,FLAYER,ITIME) +.01	LOTR 205
C		LOTR 206
C	NOW TRANSPORT THE PARTICLE FOR THAT PERIOD OF TIME	LOTR 207
	XP(J)=XP(J)+ITRANS*VX(JWAD)	LOTR 208
	YP(J)=YP(J)+ITRANS*VY(JWAD)	LOTR 209
	ZP(J)=ZP(J)+ITRANS*VPZ	LOTR 210
	TP(J)=TP(J)+ITRANS	LOTR 211
	GO TO 131	LOTR 212
C		LOTR 213
C	1202 PARTICLE IS WITHIN LOCAL CELL. BRANCH TO CALL APPROPRIATE WIND	LOTR 214
C	PROGRAM	LOTR 215
C	BRANCH TO CALL A LOCAL WIND PROGRAM TO TEST PARTICLE FOR IMPACT	LOTR 216
C	TOPOGRAPHY. IF IMPACTED, IT ASSIGNS A LARGE DOWNWARD WIND COM-	LOTR 217
C	POONENT. IF NOT IMPACTED, IT COMPUTES CORRECT WIND COMPONENTS AT	LOTR 218
C	THE PARTICLE POSITION	LOTR 219
1202	GO TO NC,(121,122,123,124,125)	LOTR 220
121	CALL HTWIND(J,K,AX,AY,AZ)	LOTR 221
	GO TO 130	LOTR 222
122	CALL RGWIND(J,K,AX,AY,AZ)	LOTR 223
	GO TO 130	LOTR 224
123	CALL CBREZ1(J,K,AX,AY,AZ)	LOTR 225
	GO TO 130	LOTR 226
C	***** CODE INSERTION POINTS *****	LOTR 227
124	CONTINUE	LOTR 228
125	CONTINUE	LOTR 229
C	***** CODE INSERTION POINTS *****	LOTR 230
126	IROR=-126	LOTR 231
7734	CALL ERROR(PROGM,IROR,ISOUT)	LOTR 232
C		LOTR 233
C	130 TRANSPORT THE PARTICLE FOR ONE TIME INCREMENT (DTLOC).	LOTR 234

130	ZP(J)=ZF(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)-CRMIX(K))132,132,133	LOTR	243
133	IF(CRMXX(K)-XP(J))132,132,134	LOTR	244
134	IF(YP(J)-CRMINY(K))132,132,135	LOTR	245
135	IF(CRMXY(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 BOUNDARY	LOTR	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)-END114)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 MTWN
SUBROUTINE MTWND1(J,K,AX,AY,AZ) MTWN 1
C 11 OCT 66 MTWN 2
C 1. KOHLBERG, T.W.SCHWENKE TECHNICAL OPERATIONS RESEARCH, INC. MTWN 3
C MTWN 4
C THIS SUBROUTINE SERVES THE DUAL PURPOSE OF READING MOUNTAIN WIND MTWN 5
C DATA WHEN THE SIGN OF ARGUMENT J IS MINUS) AND COMPUTING THE MTWN 6
C MOUNTAIN WIND FOR THE JTH PARTICLE AFTER FIRST CHECKING FOR IMPACT MTWN 7
C ON THE ANALYTICAL GROUND. IF IMPACT IS SENSED THE PARTICLE IS MTWN 8
C ASSIGNED A LARGE DOWNWARD VELOCITY COMPONENT. MTWN 9
C MTWN 10
C ***** MTWN 11
C MTWN 12
COMMON /SET1/ MTWN 13
1 DIAM , DETID , IRISE , IEXEC , ISPN , ISOUT , MTWN 14
2 SD , SPAR , SSAM , TAE , IMPI , IMPZ , MTWN 15
3 IZM , U , VPR , W , X , Z , MTWN 16
4 WHY , RMIN , IDISTR , SPAR1 , SPAR2 , SPAR3 , MTWN 17
5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 MTWN 18
C MTWN 19
C ***** MTWN 20
C MTWN 21
DIMENSION DETID(12),WHY(40) MTWN 22
COMMON /SET2/ MTWN 23
1 S , SUBSID , GRINT , BALL , BALU , BYLL MTWN 24
2, HYLU , TALL , TKLU , TYLL , TLEU , XGZ MTWN 25
3, YGZ , NBECK , HTOFU , ITOFU , ILIN , JLIN MTWN 26
4, ALIM , II , JU , KK , AP , YP MTWN 27
5, ZP , FMAS , PS , PS , VX , VT MTWN 28
6, VZ , IL , JL , IBADD , WGRINT , NSIRAI MTWN 29
7, WLLX , WLLY , WORX , WURY , BOTHIT , IPARIN MTWN 30
8, ITOFU , IOWIND , IHTOPU , IPOUT , IPAROT , JTOP1 MTWN 31
9, JWINDI , IKROR , ILINH , ENDIM , IC , IBYPAS MTWN 32
1, JTOPJ , NEOST , NG , NTO , NTI , NW MTWN 33
2, NALOFT , JTIMEI , NDMAX , NKEE , N , NCL MTWN 34
3, CRMXY , CROHT , NCRTYP , SZ , CRMIX , CRMINY MTWN 35
4, CC , SN , CS , NCCIR , JTEOC , ATEPP MTWN 36
5, RHC , NA , IGZ , JTRAC , FRUG , CRMXXX MTWN 37
6, RUPART MTWN 38
DIMENSION TOPLEM(4,4) , NINIR(4) , ITOPLM(3,4) MTWN 39
DIMENSION S(1,10) , SUBSID(400) , IC(18) MTWN 40
DIMENSION XP(200) , YP(200) , ZP(200) , FMAS(200) MTWN 41
DIMENSION TP(200) , PS(200) , ATEMP(200) , RHU(200) MTWN 42
DIMENSION VX(1500) , VY(1500) , VZ(1500) , IL(70) MTWN 43
DIMENSION JL(70) , IBADD(70) , WORX(70) MTWN 44
DIMENSION WGRINI(70) , WLLX(70) , WLLY(70) MTWN 45
DIMENSION WORB(70) , BOTHIT(70) , SN(6) , CS(6) MTWN 46
DIMENSION CRMIX(6) , CRMXXX(6) , CRMINY(6) , CRMXY(6) MTWN 47
DIMENSION CROHT(6) , NCRTYP(6) , CC(6) MTWN 48
C ***** MTWN 49
C MTWN 50
PARAMETERS PECULIAR TO SUBROUTINE MTWND1 MTWN 51
C MTWN 52
DIMENSION AM(12),IM(12),H(12),A(12),AZ(12),AZH(12),ASH(12) MTWN 53
C MTWN 54
C ***** MOUNTAIN WIND1 SUBROUTINE ***** MTWN 55
C MTWN 56
C A(J) HALF-WIDTH OF THE J-TH MOUNTAIN MTWN 57
C AX WIND VECTOR EAST MTWN 58
C AY WIND VECTOR NORTH MTWN 59
C AZ VERTICAL WIND VECTOR MTWN 60

```

```

C      CN(K)      COSINE OF COUNTER CLOCKWISE ANGLE FROM EAST OF      MTWN  61
C      UNPERTURBED WIND      MTWN  62
C      CRMAXY(K)  NORTH      BOUNDARY OF THE K-TH LOCAL CELL.      MTWN  63
C      CRMINY(K)  SOUTH      BOUNDARY OF THE K-TH LOCAL CELL.      MTWN  64
C      CRMAXX(K)  EAST       BOUNDARY OF THE K-TH LOCAL CELL.      MTWN  65
C      CRMINX(K)  WEST       BOUNDARY OF THE K-TH LOCAL CELL.      MTWN  66
C      CROHT(K)   THE LOCAL CELL TOP HEIGHT (METERS)      MTWN  67
C      DELX      X COORDINATE OF MACROSYSTEM TRANSLATED INTO LOCAL      MTWN  68
C      DELX      CELL COORDINATES      MTWN  69
C      DELY      Y COORDINATE      SEE DELX      MTWN  70
C      DSA      THE X PRIME RESULT OF ROTATION OF (DELX,DELY) INTO      MTWN  71
C      THE K-TH LOCAL COORDINATE SYSTEM.      MTWN  72
C      DSY      THE Y PRIME RESULT      SEE DSA      MTWN  73
C      DZ      SUM OF TOPO HEIGHT INCREMENTS      MTWN  74
C      DZ      MOUNTAIN RATIO OF H(J)/A(J)      MTWN  75
C      H(J)      HEIGHT OF THE J-TH MOUNTAIN      MTWN  76
C      NMJ      THE NUMBER OF MOUNTAINS REPRESENTED IN THIS MOUNTAIN      MTWN  77
C      WIND SYSTEM      MTWN  78
C      SN(K)      SINE OF ANGLE COUNTER-CLOCKWISE FROM EAST OF      MTWN  79
C      UNPERTURBED WIND      MTWN  80
C      SO(K)      MAGNITUDE OF UNPERTURBED WIND VECTOR      MTWN  81
C      XM(J)      X LOCATION COORDINATE OF THE J-TH MOUNTAIN      MTWN  82
C      YM(J)      Y LOCATION COORDINATE OF THE J-TH MOUNTAIN      MTWN  83
C      XX      X COORDINATE OF CENTER OF LOCAL CELL      MTWN  84
C      YY      Y COORDINATE OF CENTER OF LOCAL CELL      MTWN  85
C      ZZ      Z COORDINATE OF CENTER OF LOCAL CELL      MTWN  86
C      JW      INDEX OF THE WIND STRATUM CONTAINING THE PARTICLE      MTWN  87
C      JWAD      INDEX OF MACRO WIND CELL CONTAINING PARTICLE      MTWN  88
C      XP(J)      PARTICLE POSITION COORDINATE      MTWN  89
C      YP(J)      PARTICLE POSITION COORDINATE      MTWN  90
C      ZP(J)      PARTICLE POSITION COORDINATE      MTWN  91
C      MTWN  92
C      MTWN  93
C      MTWN  94
C      MTWN  95
1      FORMAT(4F10.3)      MTWN  96
2      FORMAT(//20A24HLOCAL CIRCULATION NUMBER10/30A10H MOUNTAIN WIND 1//10H      MTWN  97
10A10H MOUNTAIN14A10H MOUNTAIN14A10H MOUNTAIN14A20H LOCATION COORDINATES/10A6H      MTWN  98
2HNUMBER6X10HHEIGHT6X10HWIDTH10H/X1H X11H Y1)      MTWN  99
3      FORMAT(//12A,10,6X,4F11.3//)      MTWN 100
4      FORMAT(//20A,36HCOORDINATES OF LOCAL CELL BOUNDARIES/,10A,3HNORTH,      MTWN 101
110A,3H SOUTH,11A,4H EAST,11A,4H WEST,10A,3H HEIGHT//,10A,5F10.3//)      MTWN 102
5      FORMAT(//20X,46HCHARACTERISTICS OF THE UNPERTURBED WIND VECTOR/20X,      MTWN 103
116HUNPERTURBED WIND,4X,17HCOSINE OF ANGULAR,4X,15HSINE OF ANGULAR,      MTWN 104
2/20X,16HVECTOR MAGNITUDE,5X,15HDEV. FROM NORTH,5X,15HDEV. FROM NORTH      MTWN 105
3TH//,18X,3F20.3//)      MTWN 106
C      MTWN 107
C      MTWN 108
C      MTWN 109
C      DATA PROGRAM/6HMTWIND1/      MTWN 110
C      MTWN 111
C      MTWN 112
C      MTWN 113
C      MTWN 114
C      IF(J)100,101,102      MTWN 115
101  IRRUR=-101      MTWN 116
7734 CALL ERROR(PROGRM,IRRUR,ISOUT)      MTWN 117
C      MTWN 118

```

C 100 THIS IS THE DATA READING ROUTE	MTWN 119
100 J=0	MTWN 120
CRUHT(K)=0.0	MTWN 121
103 J=J+1	MTWN 122
READ (ISIN,1) XM(J),YM(J),H(J),A(J)	MTWN 123
A2(J)=A(J)*A(J)	MTWN 124
A*H(J)=A2(J)*A(J)*H(J)	MTWN 125
IF(J-12)1031,1032,1032	MTWN 126
1032 IRROR=1032	MTWN 127
GO TO 7734	MTWN 128
1031 IF(H(J))1099,104,1099	MTWN 129
C1099 COMPUTE THE KTH LOCAL CELL HEIGHT.	MTWN 130
1099 DZ=ABS(3.0*H(J))	MTWN 131
IF (DZ-CRUHT(K))110,110,1100	MTWN 132
1100 CRUHT(K)=DZ	MTWN 133
C 110 CHECK TO SEE THAT THE MOUNTAIN JUST READ IS WITHIN THE LIMITS OF	MTWN 134
C THE KTH LOCAL WIND SYSTEM.	MTWN 135
110 IF(XM(J)-CRMINX(K)) 114,111,111	MTWN 136
111 IF(XM(J)-CRMAXX(K)) 112,112,114	MTWN 137
112 IF (YM(J)-CRMINY(K)) 114,113,113	MTWN 138
113 IF (YM(J)-CRMAXY(K)) 115,115,114	MTWN 139
C	MTWN 140
C 114 THE MOUNTAIN IS NOT WITHIN THE LIMITS OF THE KTH LOCAL WIND SYSTEM	MTWN 141
114 IRROR= 114	MTWN 142
GO TO 7734	MTWN 143
C	MTWN 144
C 115 CHECK TO SEE THAT THE MOUNTAIN RATIO H(J)/A(J) IS LESS THAN 0.6	MTWN 145
115 DZ = H(J)/A(J)	MTWN 146
IF (DZ-0.6)103,116,116	MTWN 147
C	MTWN 148
C 116 THE MOUNTAIN RATIO H(J)/A(J) IS NOT LESS THAN 0.6	MTWN 149
116 IRROR = 116	MTWN 150
GO TO 7734	MTWN 151
C	MTWN 152
104 NMT=J-1	MTWN 153
C NMT	MTWN 154
C THE NUMBER OF MOUNTAINS REPRESENTED IN THIS	MTWN 155
C MOUNTAIN WIND SYSTEM	MTWN 156
C 1042 COMPUTE UNPERTURBED VECTOR HERE	MTWN 157
C THE FOLLOWING THREE CARDS CONSTITUTE THE LOCATION COORDINATES OF	MTWN 158
C THE UNPERTURBED WIND VECTOR	MTWN 159
YY=(CRMAXY(K)+CRMINY(K))/2.0	MTWN 160
XX=(CRMAXX(K)+CRMINX(K))/2.0	MTWN 161
ZZ=CRUHT(K)/2.0	MTWN 162
CALL GETWND(XX,YY,ZZ,JWAD,JW)	MTWN 163
IF(JWAD)1043,1044,1045	MTWN 164
1043 IRROR=1043	MTWN 165
GO TO 7734	MTWN 166
1044 IRROR=1044	MTWN 167
GO TO 7734	MTWN 168
C THE FOLLOWING THREE CARDS CONSTITUTE THE MAGNITUDE AND DIRECTION	MTWN 169
C OF THE UNPERTURBED WIND VECTOR	MTWN 170
1045 UO(K)=SQRT(VX(JWAD)*VX(JWAD)+VY(JWAD)*VY(JWAD))	MTWN 171
SN(K)=VY(JWAD)/UO(K)	MTWN 172
CS(K)=VX(JWAD)/UO(K)	MTWN 173
DO 1049 J=1,NMT	MTWN 174
1049 A2H(J)=A2(J)*H(J)*UO(K)	MTWN 175
WRITE (ISOUT,2)K	MTWN 176
WRITE (ISOUT,3)(J,H(J),A(J),XM(J),YM(J),J=1,NMT)	

WRITE (ISOUT,4)CRMXY(K),CRMINY(K),CRMXX(K),CRMIX(K),CROHT(K)	MTWN 177
WRITE (ISOUT,5)UO(K),SN(K),CS(K)	MTWN 178
105 RETURN	MTWN 179
C	MTWN 180
C 102 THIS IS THE TESTING AND COMPUTING ROUTE	MTWN 181
C COMPUTE THE TOPOGRAPHIC INCREMENT AT POSITION OF THE JTH PARTICLE	MTWN 182
C COMPUTE THE PERTURBED WIND COMPONENTS, SUM THEM, AND ADD THEM	MTWN 183
C TO THE UNPERTURBED WIND VECTOR.	MTWN 184
102 AX=0.0	MTWN 185
AY=0.0	MTWN 186
AZ=0.0	MTWN 187
DZ=0.0	MTWN 188
DO 106 I=1,NMT	MTWN 189
C THE FOLLOWING TWO CARDS TRANSLATE THE PARTICLE INTO THE MOUNTAIN	MTWN 190
C COORDINATE SYSTEM.	MTWN 191
DELX=XP(J)-XM(I)	MTWN 192
DELY=YP(J)-YM(I)	MTWN 193
C THE FOLLOWING TWO CARDS ROTATE THE PARTICLE INTO THE MOUNTAIN	MTWN 194
C COORDINATE SYSTEM.	MTWN 195
DSX = DELX * CS(K) + DELY * SN(K)	MTWN 196
DSY = -DELX * SN(K) + DELY * CS(K)	MTWN 197
Y2 = DSY * DSY	MTWN 198
X2 = DSX * DSX	MTWN 199
R2=X2+Y2	MTWN 200
C NOW COMPUTE TOPO HEIGHT INCREMENT RESULTING FROM THE ITH MOUNTAIN	MTWN 201
C AND ADD IT TO SUM.	MTWN 202
DZ = DZ + AZH(I)/((A2(I)+R2)*SQRT(A2(I)+R2))	MTWN 203
C	MTWN 204
C COMPUTE PERTURBATION WIND INCREMENTS	MTWN 205
AMBD A = ZP(J)+A(I)	MTWN 206
AMBD A2=AMBD A*AMBD A	MTWN 207
DENOM = (R2+AMBD A2)*(R2+AMBD A2)*SQRT(R2+AMBD A2)	MTWN 208
Q = AZH(I)*3.0*DSX/DENOM	MTWN 209
C	MTWN 210
C1061 AX, THE PERTURBED WIND COMPONENT IN THE DIRECTION OF THE	MTWN 211
C UNPERTURBED WIND.	MTWN 212
1061 AX=AX+AZH(I)*(Y2+AMBD A2-2.0*X2)/DENOM	MTWN 213
C AY, AZ, THE PERTURBED WIND COMPONENTS PERPENDICULAR TO THE	MTWN 214
C DIRECTION OF THE UNPERTURBED WIND.	MTWN 215
AY= AY - Q*DSY	MTWN 216
106 AZ= AZ - Q*AMBD A	MTWN 217
C	MTWN 218
C NOW TEST FOR IMPACTED PARTICLE	MTWN 219
IF(DZ-ZP(J))109,108,108	MTWN 220
C 108 PARTICLE HAS IMPACTED	MTWN 221
108 AZ=-1.0E+08	MTWN 222
AX=0.0	MTWN 223
AY=0.0	MTWN 224
GO TO 105	MTWN 225
C 109 THE PARTICLE IS ALOFT. NOW ADD THE UNPERTURBED WIND VECTOR TO	MTWN 226
C THE PERTURBED COMPONENT IN THE SAME DIRECTION.	MTWN 227
109 DELX = AX+UO(K)	MTWN 228
C THE FOLLOWING TWO CARDS DEROTATE THE WIND VECTOR INTO THE	MTWN 229
C MACRO SYSTEM.	MTWN 230
AX=DELX*CS(K)-AY*SN(K)	MTWN 231
AY=DELX*SN(K)+AY*CS(K)	MTWN 232
GO TO 105	MTWN 233
END	MTWN 234

235*

235 *

```

$IBFTC RGWN1 LIST,DECK,M94/2
SUBROUTINE RGWND1(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 RIDGE WIND
C DATA WHEN THE SIGN OF ARGUMENT J IS MINUS) AND COMPUTING THE
C RIDGE 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 COMMON /SET1/
C 1 DIAM , DETID , IRIDE , IEXEC , IOIN , ISOUT , RGWN 14
C 2 SD , SPAR , SSAN , TME , IMP1 , IMP2 , RGWN 15
C 3 T2R , C , VPK , W , X , Z , RGWN 16
C 4 WHY , KNIN , IDISTR , SPAR1 , SPAR2 , SPAR3 , RGWN 17
C 5 SPAR4 , SPAR5 , SPAR6 , SPAR7 , SPAR8 , SPAR9 , RGWN 18
C *****
C DIMENSION DETID(12),WHY(40)
C COMMON /SET2/
C 1 S , SUBSID , GRINT , EXLL , EXLU , BYLL , RGWN 24
C 2 PYLU , IXLL , IXLU , IYLL , IYLU , XGZ , RGWN 25
C 3 YGZ , NBLOCK , HTOPU , ITOPU , ILIM , JLim , RGWN 26
C 4 CLIM , II , JJ , KK , XP , YP , RGWN 27
C 5 ZP , FMAS , IP , PS , VA , VY , RGWN 28
C 6 VZ , IL , JL , ISADD , WGRINT , NSTRAT , RGWN 29
C 7 WLLX , WLLY , WURX , WURY , BOTHII , IPARIN , RGWN 30
C 8 IOTOPU , IOWIND , IHTOPU , IPOUT , IPAROT , JTOP1 , RGWN 31
C 9 JWINBI , IRACK , ILMII , ENDTIM , IC , IBYPAS , RGWN 32
C 10 JTOPU , NEOST , NG , NTO , NTL , NW , RGWN 33
C 11 NALOFF , OTIME1 , N3MAX , NFREE , N , NCL , RGWN 34
C 12 CRMAXY , CRUHI , NCRITY , SZ , CRMINX , CRMINY , RGWN 35
C 13 CU , CN , CS , NLOCIR , DTLOC , ATEMP , RGWN 36
C 14 RHO , NA , T2Z , DTMAC , FRUG , CRMAXX , RGWN 37
C 15 ROPART , RGWN 38
C DIMENSION TORLEN(4,4) , NIRPAR(4) , ITOPEN(3,4) , RGWN 39
C DIMENSION S10(10) , SUBSID(400) , IC(18) , RGWN 40
C DIMENSION AP(200) , YP(200) , ZP(200) , FMAS(200) , RGWN 41
C DIMENSION IP(200) , PS(200) , ATEMP(200) , RHO(200) , RGWN 42
C DIMENSION VA(1000) , VY(1000) , VZ(1000) , IL(70) , RGWN 43
C DIMENSION JL(70) , ISADD(70) , WURX(70) , RGWN 44
C DIMENSION WGRINT(70) , WLLX(70) , WLLY(70) , RGWN 45
C DIMENSION WURX(70) , BOTHII(70) , CN(6) , CS(6) , RGWN 46
C DIMENSION CRMINX(6) , CRMAXX(6) , CRMINY(6) , CRMAXY(6) , RGWN 47
C DIMENSION CRUHI(6) , NCRITY(6) , CU(6) , RGWN 48
C *****
C PARAMETERS PECULIAR TO SUBROUTINE RGWND1
C DIMENSION XH(12),YM(12),H(12),A(12),B(12),C(12),SG(12),CG(12),
C IAZ(12),AH(12),D(12),A2H(12)
C *****
C ***** RIDGE WIND 1 GLOSSARY *****
C A(K) THE HALF WIDTH OF THE KTH RIDGE
C AX WIND VECTOR EAST

```

C	AY	WIND VECTOR NORTH	RGWN	61
C	AZ	VERTICAL WIND VECTOR	RGWN	62
C	AX	PERTURBED WIND COMPONENT IN THE DIRECTION OF THE	RGWN	63
C		UNPERTURBED WIND	RGWN	64
C	B(K)	CLOCKWISE ANGULAR DEVIATION OF KTH RIDGE FROM	RGWN	65
C		TRUE NORTH	RGWN	66
C	CCG	COSINE OF ANGLE UNPERTURBED WIND MAKES WITH	RGWN	67
C		PERPENDICULAR FROM KTH RIDGE	RGWN	68
C	CN(K)	COSINE OF COUNTER-CLOCKWISE ANGLE FROM EAST OF	RGWN	69
C		UNPERTURBED WIND	RGWN	70
C	CRMAXY(K)	NORTH BOUNDARY OF THE K-TH LOCAL CELL.	RGWN	71
C	CRMINY(K)	SOUTH BOUNDARY OF THE K-TH LOCAL CELL.	RGWN	72
C	CRMAXX(K)	EAST BOUNDARY OF THE K-TH LOCAL CELL.	RGWN	73
C	CRMINX(K)	WEST BOUNDARY OF THE K-TH LOCAL CELL.	RGWN	74
C	CRH(K)	THE LOCAL CELL TOP HEIGHT (METERS)	RGWN	75
C	DSX	THE PERPENDICULAR DISTANCE OF THE JTH PARTICLE FROM	RGWN	76
C		THE JTH RIDGE (METERS)	RGWN	77
C	DZ	SUM OF TOPO HEIGHT INCREMENTS	RGWN	78
C	DZ	RIDGE RATIO OF H(J)/A(J)	RGWN	79
C	H(K)	THE HEIGHT OF THE KTH RIDGE	RGWN	80
C	NRG	NUMBER OF RIDGES IN THIS RIDGE WIND SYSTEM	RGWN	81
C	SN(K)	SINE OF ANGLE COUNTER-CLOCKWISE FROM EAST OF	RGWN	82
C		UNPERTURBED WIND	RGWN	83
C	SSG	SINE OF ANGLE UNPERTURBED WIND MAKES WITH	RGWN	84
C		PERPENDICULAR FROM KTH RIDGE	RGWN	85
C	UU(K)	MAGNITUDE OF UNPERTURBED WIND VECTOR	RGWN	86
C	XN(J)	X LOCATION COORDINATE OF THE J-TH RIDGE	RGWN	87
C	XX	X COORDINATE OF CENTER OF LOCAL CELL	RGWN	88
C	YN(J)	Y LOCATION COORDINATE OF THE J-TH RIDGE	RGWN	89
C	YY	Y COORDINATE OF CENTER OF LOCAL CELL	RGWN	90
C	ZZ	Z COORDINATE OF CENTER OF LOCAL CELL	RGWN	91
C	JW	INDEX OF THE WIND SITUATION CONTAINING THE PARTICLE	RGWN	92
C	JWAD	INDEX OF MACRO WIND CELL CONTAINING PARTICLE	RGWN	93
C	XP(J)	PARTICLE POSITION COORDINATE	RGWN	94
C	YP(J)	PARTICLE POSITION COORDINATE	RGWN	95
C	ZP(J)	PARTICLE POSITION COORDINATE	RGWN	96
C			RGWN	97
C	*****		RGWN	98
C			RGWN	99
1	FORMAT(5F10.3)		RGWN	100
2	FORMAT(//25X,24HLOCAL CIRCULATION NUMBER16/30X16H RIDGE WIND 1//1RGWN		RGWN	101
	10X5HRLDGE/40HRLDGE/40HRLDGE/420HLOCATION COORDINATES30X10HRLDGE 421RGWN		RGWN	102
	2501H/10X6HNUMBER30HHEIGHT10X5HWIDTH5X10H11X1HY)		RGWN	103
3	FORMAT(//12X,10,3X,4F12.3,3X,F11.3//)		RGWN	104
4	FORMAT(//25X,36HCOORDINATES OF LOCAL CELL BOUNDARIES/15X,5HNORTH,RGWN		RGWN	105
	110X,5HSOUTH,11X,4HEAST,11X,4HWEST,10X,6HHEIGHT//,7X,5F10.3//)		RGWN	106
5	FORMAT(//20X,46HCHARACTERISTICS OF THE UNPERTURBED WIND VECTOR/20X,RGWN		RGWN	107
	110HUNPERTURBED WIND,4X,1HCOSINE OF ANGULAR,4X,1HSINE OF ANGULAR,RGWN		RGWN	108
	2/20X,10HVECTOR MAGNITUDE,5X,1HDEV. FROM NORTH,5X,1HDEV. FROM NORTH,RGWN		RGWN	109
	3TH//,13X,3F20.3//)		RGWN	110
C			RGWN	111
C	*****		RGWN	112
C			RGWN	113
C	DATA PROGRAM/6HRGWIND1/		RGWN	114
C			RGWN	115
C	*****		RGWN	116
C	*****		RGWN	117
C			RGWN	118

IF(J)100,101,102	RGWN 119
101 ERROR=-101	RGWN 120
7734 CALL ERROR(PROGRAM,ERROR,ISOUT)	RGWN 121
C 100 THIS IS THE DATA READING ROUTE	RGWN 122
100 J=0	RGWN 123
CRUHT(K)=0.0	RGWN 124
103 J=J+1	RGWN 125
READ (ISIN,1) XM(J),YM(J),H(J),A(J),B(J)	RGWN 126
C(J) = COS(B(J))	RGWN 127
S(J) = SIN(B(J))	RGWN 128
A2(J) = A(J)*A(J)	RGWN 129
A2H(J)=A2(J)*H(J)	RGWN 130
IF(J-12)1031,1032,1032	RGWN 131
1032 ERROR=1032	RGWN 132
GO TO 7734	RGWN 133
1031 IF(H(J))1099,104,1099	RGWN 134
C 1099 COMPUTE THE KTH LOCAL CELL HEIGHT.	RGWN 135
1099 DZ=ABS(3.0*H(J))	RGWN 136
IF (DZ-CRUHT(K))110,115,115	RGWN 137
110 CRUHT(K)=DZ	RGWN 138
C 110 CHECK TO SEE THAT THE RIDGE JUST READ IS WITHIN THE LIMITS OF	RGWN 139
C THE KTH LOCAL WIND SYSTEM.	RGWN 140
110 IF(XM(J)-CRMINX(K)) 114,111,111	RGWN 141
111 IF(XM(J)-CRMAXX(K)) 112,112,114	RGWN 142
112 IF (YM(J)-CRMINY(K)) 114,113,113	RGWN 143
113 IF (YM(J)-CRMAXY(K)) 115,115,114	RGWN 144
C	RGWN 145
C 114 THE RIDGE IS NOT WITHIN THE LIMITS OF THE KTH LOCAL WIND SYSTEM	RGWN 146
114 ERROR= 114	RGWN 147
GO TO 7734	RGWN 148
C	RGWN 149
C 115 CHECK TO SEE THAT THE RIDGE RATIO H(J)/A(J) IS LESS THAN 0.6	RGWN 150
115 DZ = H(J)/A(J)	RGWN 151
IF (DZ-0.6) 105,116,116	RGWN 152
C	RGWN 153
C 116 THE RIDGE RATIO H(J)/A(J) IS NOT LESS THAN 0.6	RGWN 154
116 ERROR = 116	RGWN 155
GO TO 7734	RGWN 156
C	RGWN 157
114 NPG=J-1	RGWN 158
C NRG	RGWN 159
C THE NUMBER OF RIDGES REPRESENTED IN THIS	RGWN 160
C RIDGE WIND SYSTEM	RGWN 161
C 1042 COMPUTE UNPERTURBED VECTOR HERE	RGWN 162
C THE FOLLOWING THREE CARDS CONSTITUTE THE LOCATION COORDINATES OF	RGWN 163
C THE UNPERTURBED WIND VECTOR	RGWN 164
YY=(CRMAXY(K)+CRMINY(K))/2.0	RGWN 165
XX=(CRMAXX(K)+CRMINX(K))/2.0	RGWN 166
ZZ=CRUHT(K)/2.0	RGWN 167
CALL GETWIND(XX,YY,ZZ,UWAD,UW)	RGWN 168
IF(UWAD)1043,1044,1043	RGWN 169
1043 ERROR=1043	RGWN 170
GO TO 7734	RGWN 171
1044 ERROR=1044	RGWN 172
GO TO 7734	RGWN 173
C THE FOLLOWING THREE CARDS CONSTITUTE THE MAGNITUDE AND DIRECTION	RGWN 174
C OF THE UNPERTURBED WIND VECTOR	RGWN 175
1045 UO(K)=SQRT(VX(UWAD)*VX(UWAD)+VY(UWAD)*VY(UWAD))	RGWN 176
SN(K)=VY(UWAD)/UO(K)	

CS(K)=VX(JWAD)/UO(K)	RGWN 177
DO 1049 J=1,NRG	RGWN 178
SSG = CS(K)*D(J)+SN(K)*C(J)	RGWN 179
CCG = CS(K)*C(J)-SN(K)*D(J)	RGWN 180
AH(J)= -A(J)*H(J)*UO(K)*CCG*CCG	RGWN 181
SG(J)=SSG/CCG	RGWN 182
1049 CG(J)= 2.0/CCG	RGWN 183
C	RGWN 184
WRITE (ISOUT,2)K	RGWN 185
WRITE (ISOUT,3)(J,H(J),A(J),AH(J),H(J),D(J),C(J),VAD)	RGWN 186
WRITE (ISOUT,4)(CRMAA(K),CRMIN(K),CRMAXA(K),CRMINA(K),CRUM(K))	RGWN 187
WRITE (ISOUT,5)CG(K),SN(K),CC(K)	RGWN 188
105 RETURN	RGWN 189
C	RGWN 190
C 102 THIS IS THE TESTING AND COMPUTING ROUTE	RGWN 191
C COMPUTE THE TOPOGRAPHIC INCREMENT AT POSITION OF THE JTH PARTICLE	RGWN 192
C COMPUTE THE PERTURBED WIND COMPONENTS, SOFTEN, AND ADD THEM	RGWN 193
C TO THE UNPERTURBED WIND VECTOR.	RGWN 194
102 AX=0.0	RGWN 195
AY=0.0	RGWN 196
AZ=0.0	RGWN 197
DZ=0.0	RGWN 198
DO 105 I=1,NRG	RGWN 199
C 107 THIS CARD TRANSLATES AND ROTATES THE PARTICLE INTO THE WIND	RGWN 200
C WIND SYSTEM. DDX IS THE PERPENDICULAR DISTANCE OF THE ITH	RGWN 201
C PARTICLE FROM THE I-TH RIDGE.	RGWN 202
107 DDX = (X(I)-X(I1))*(Y(I)-Y(I1))*R(I1)	RGWN 203
X2 = DDX*DDX	RGWN 204
AMDDA = ZP(J) + A(I1)	RGWN 205
AMDDA2 = AMDDA*AMDDA	RGWN 206
C NOW COMPUTE TOP TOPOGRAPHIC INCREMENT RESULTING FROM THE ITH RIDGE	RGWN 207
C AND ADD IT TO DDX.	RGWN 208
DZ = DZ + AZH(I1)/(X2(I1)+X2)	RGWN 209
C	RGWN 210
C THE FOLLOWING CARDS COMPUTE THE PERTURBATION WIND INCREMENTS	RGWN 211
AMDDA = AH(I1)/(X2+AMDDA2)*(X2+AMDDA2)	RGWN 212
AMDDA2 = X2+AMDDA2	RGWN 213
C AX IS THE COMPONENT IN THE DIRECTION OF THE UNPERTURBED WIND	RGWN 214
AX=AMDDA*AMDDA2 + AX	RGWN 215
AY= AMDDA*SG(I)*AMDDA2 + AY	RGWN 216
105 AZ= AMDDA*CG(I)*AMDDA*DDX + AZ	RGWN 217
C	RGWN 218
C NOW TEST FOR IMPACTED PARTICLE	RGWN 219
IF(DZ-ZP(J))109,108,108	RGWN 220
C 108 PARTICLE HAS IMPACTED	RGWN 221
108 AZ=-1. E+08	RGWN 222
AX=0.0	RGWN 223
AY=0.0	RGWN 224
GO TO 105	RGWN 225
C 109 THE PARTICLE IS ALOFT. NOW ADD THE UNPERTURBED WIND VECTOR TO	RGWN 226
C THE PERTURBED COMPONENT IN THE SAME DIRECTION.	RGWN 227
109 DDX=AX+UO(K)	RGWN 228
C THE FOLLOWING TWO CARDS DEGRATE THE WIND SYSTEM INTO MACRO FIELD	RGWN 229
C COORDINATES	RGWN 230
AX= DDX*CG(K)-AY*SN(K)	RGWN 231
AY= DDX*SN(K)+AY*CG(K)	RGWN 232
GO TO 105	RGWN 233
END	RGWN 234

235*

235 *

```

$IBFTC CBRE11 LIST,DECK,M94/2
SUBROUTINE CBREZ1(J,K,AX,AY,AZ)
C      11 OCT 66
C      I. KOHLBERG, I.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-IN PARTICLE.
C      *****
C      COMMON /SET1/
C      1 DIAM , DETID , IRICE , IELEC , IGIN , IISUJ , ,
C      2 UB , SPAR , UAH , UME , JAP1 , JAP2 , ,
C      3 IZ , U , VPS , W , A , Z , ,
C      4 ART , RAIN , IDICK , SPART , SPARE , SPARS , ,
C      5 SPAR4 , SPAR5 , SPAR6 , SPART , SPAR8 , SPAR9
C      *****
C      DIMENSION DETID(12),WHY(40)
C      1 XN /SET2/
C      1 S , SUBSID , GRINI , BALL , BALU , STEL
C      2 BYLO , TALL , TALO , TTEL , TTEL , AGE
C      3 YGZ , NBLOCK , HICPU , TICPU , ILIN , ULIN
C      4 XLIN , IL , JU , NA , AP , IF
C      5 ZP , FRAS , IF , FO , VA , VI
C      6 VL , IL , JL , ISADU , WORKIN , NSTRAI
C      7 WELX , VLEY , WORA , WORT , SCIMP , IPASIN
C      8 IOTCPU , IOWIND , IOTCPU , IPOOT , IOWIND , IOTCPU
C      9 SWIND1 , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      10 IOTCPU , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      11 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      12 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      13 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      14 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      15 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      16 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      17 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      18 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      19 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      20 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      21 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      22 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      23 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      24 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      25 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      26 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      27 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      28 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      29 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      30 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      31 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      32 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      33 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      34 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      35 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      36 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      37 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      38 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      39 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      40 IOWIN , IOWIN , IOWIN , IOWIN , IOWIN , IOWIN
C      *****
C      PARAMETER PECULIA TO COREZ
C      DIMENSION HUA(4),HUA(4),HUA(4),HUA(4),HUA(4),
C      IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4),
C      IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4),
C      IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4),IOWIN(4)
C      *****
C      GRAVITY=9.80665 (METERS/SECOND**2)
C      THERMAL=0.001 (METERS/SECOND**2)
C      AX=0.0 (METERS/SECOND)
C      AY=0.0 (METERS/SECOND)
C      AZ=0.0 (METERS/SECOND)
C      B=0.0 (THE ANGLE (MEASURED CLOCKWISE FROM TRUE NORTH) THAT

```


7734 CALL ERROR(PROGRAM, IAXOR, 13001)	CBRE 119
C 100 THIS IS THE DATA READING ROUTE	CBRE 120
C	CBRE 121
100 READ (10IN,1)SINPHI,SGMA,ELX,THET	CBRE 122
READ (10IN,1)W,AXI,B,GRAD,NN	CBRE 123
READ (10IN,4)(DELIAIN),IAXIN),N=1,NN)	CBRE 124
WRITE (13001,2)X	CBRE 125
ALN=B*.2831623/ELX	CBRE 126
ALPH=9.8/THET	CBRE 127
CON=(14.0444+10E-05)*SINPHI	CBRE 128
DO 2000 N=1,NN	CBRE 129
CONX(N)=N	CBRE 130
CONX(N)=CONX(N)*7.2722092E-05	CBRE 131
A2=SGMA*CONX(N)+CONX(N)*CONX(N)	CBRE 132
A1=SQRT(A2)	CBRE 133
IF(SGMA)1003,1003,1004	CBRE 134
C1003 SGMA IS ZERO OR NEGATIVE. THIS IS NOT ALLOWED	CBRE 135
1003 IRRX=1003	CBRE 136
GO TO 7734	CBRE 137
1004 I1=ATA(1,CONX(N)/SGMA)	CBRE 138
I2=2.0*I1	CBRE 139
SQC=SGMA**2-CONX(N)**2+CON**2	CBRE 140
A3=SQRT(SQC**2+4.0*SGMA**2*CONX(N)**2)	CBRE 141
IF(SQC)1.05,1.55,1500	CBRE 142
1005 I3=ATAN(2.0*CONX(N)*SGMA/A3)+3.1415927	CBRE 143
GO TO 1500	CBRE 144
1005 I3=1.070700	CBRE 145
GO TO 1500	CBRE 146
1005 I3=ATAN(2.0*CONX(N)/SGMA/SQC)	CBRE 147
1005 A4=(A2+ALN**2)/A3	CBRE 148
I4=I2-I3	CBRE 149
A5=(A1-ALPH*A4**2)/A3	CBRE 150
I5=I1-I3	CBRE 151
IF(ANY)1006,1006,1000	CBRE 152
C1006 ANY IS ZERO OR NEGATIVE. THIS IS NOT ALLOWED	CBRE 153
1006 IRRY=1006	CBRE 154
GO TO 7734	CBRE 155
16 6 A6=CONX(N)/AKY	CBRE 156
A7=SGMA/IKY	CBRE 157
ACOT4 = A4*COS(I4)	CBRE 158
ASIT4 = A4*SIN(I4)+A5	CBRE 159
C1=ACOT4*ACOT4+ASIT4*ASIT4+4.0*(A1*A5*COS(I3)+A6*A4*SIN(I4))	CBRE 160
C2=2.0*ACOT4*ASIT4+4.0*(A6*ACOT4+A5*A4*SIN(I3))	CBRE 161
IF(C1)10 7,1077,1777	CBRE 162
1007 BETA=ATAN(C2/C1) + 3.1415927	CBRE 163
GO TO 1777	CBRE 164
1077 IF(B2) 10,9077,9077	CBRE 165
9077 BETA=4.712389	CBRE 166
GO TO 1777	CBRE 167
9077 BETA=1.5707963	CBRE 168
GO TO 1777	CBRE 169
1777 BETA=ATAN(B2/B1)	CBRE 170
7777 AN=BETA/2.0	CBRE 171
B3=(B1*B1+B2*B2)*.0.25	CBRE 172
ACOSN=COS(AN)*BB/2.0	CBRE 173
ASINN=SIN(AN)*BB/2.0	CBRE 174
ACOT4=ACOT4/2.0	CBRE 175
ASIT4=ASIT4/2.0	CBRE 176

C1= ACOT4+ACOSM	CBRE 177
D1= ASIT4+ASINM	CBRE 178
C2=ACOT4-ACOSM	CBRE 179
D2= ASIT4-ASINM	CBRE 180
IF(C1)1000,1000,1000	CBRE 181
1000 GAM1=ATAN(D1/C1) + 0.1415927	CBRE 182
GO TO 5888	CBRE 183
1088 IF(D1)+0.00,0.00,0.00	CBRE 184
9078 GAM1=4.7123890	CBRE 185
GO TO 8888	CBRE 186
9088 GAM1=1.57 7367	CBRE 187
GO TO 8888	CBRE 188
1848 GAM1=ATAN(D1/C1)	CBRE 189
5888 IF(C2)1000,1000,1000	CBRE 190
1009 GA12=ATAN(D2/C2) + 0.1415927	CBRE 191
GO TO 1111	CBRE 192
1099 IF(D2)+0.0,0.0,0.0	CBRE 193
9099 GA12=4.7123890	CBRE 194
GO TO 1111	CBRE 195
4099 GA12=1.57 7367	CBRE 196
GO TO 1111	CBRE 197
1099 GA12=ATAN(D2/C2)	CBRE 198
1111 C*IE1=(C1*C1+D1*D1)*0.25	CBRE 199
C*IE2=(C2*C2+D2*D2)*0.25	CBRE 200
AN1=GAM1/2.0	CBRE 201
AN2=GA12/2.0	CBRE 202
CON1AC = (C*IE1)	CBRE 203
CON2X(N)=COS(AN2)	CBRE 204
CAN1=SIN(AN1)	CBRE 205
CAN2X(N)=SIN(AN2)	CBRE 206
IF(CON1)0.0,0.0	CBRE 207
5 R1RX(N)=C*IE1*CON1	CBRE 208
R1IX(N)=C*IE1*CAN1	CBRE 209
EPS1=1.0	CBRE 210
GO TO 7	CBRE 211
66 R1RX(N)=-R1IF1*CON1	CBRE 212
R1IX(N)=-R1IF1*CAN1	CBRE 213
EPS1=-1.0	CBRE 214
7 IF(CON2X(N))0.0,0.0	CBRE 215
8 R2RX(N)=C*IE2*CON2X(N)	CBRE 216
R2IX(N)=C*IE2*CAN2X(N)	CBRE 217
EPS2=1.0	CBRE 218
GO TO 10	CBRE 219
9 R2RX(N)=-C*IE2*CON2X(N)	CBRE 220
R2IX(N)=-C*IE2*CAN2X(N)	CBRE 221
EPS2=-1.0	CBRE 222
10 IF(N-1)/730,110,101	CBRE 223
7730 IRXOR=7730	CBRE 224
GO TO 7734	CBRE 225
0 DETERMINATION OF HEIGHT OF SEA BREEZE	CBRE 226
11 HT1=-1.0/R1RX(N)	CBRE 227
HT2=-1.0/R2RX(N)	CBRE 228
0 CROMI(N) IS THE HEIGHT OF THE SEA BREEZE CELL.	CBRE 229
IF(HT1-HT2)11,11,12	CBRE 230
11 CROMI(K)=2.0*HT2	CBRE 231
GO TO 13	CBRE 232
12 CROMI(K)=2.0*HT1	CBRE 233
13 WRITE (10001,20)CROMAX(N),CROMIN(N),CROMAX(N),CROMIN(N),CROMI(N)	CBRE 234

	WRITE (10001,3) SWPHI,CGMA, NW,ELX,THEI,WW,ANT,B,GRAD	CBRE 230
	WRITE (10001,21)	CBRE 236
131	FAX(N)=15-AM+IAUX(N)	CBRE 237
	G=-COR/AL	CBRE 238
C		CBRE 239
C	COMPUTATIONS FOR CONSTANT COEFFICIENTS OF THE WIND FIELD	CBRE 240
	AJZX(N)=-DELTA(N)*AD/BS	CBRE 241
	AJXX(N)=AJZX(N)/ALH	CBRE 242
	AJY=AJXX(N)*G	CBRE 243
	ESQ1=EPS1*SRTE1	CBRE 244
	ESQ2=EPS2*SRTE2	CBRE 245
	ANG=ESQ1/ESQ2	CBRE 246
	CNN1X(N)=COS(AN1+T1)*ANG*G	CBRE 247
	CNN2X(N)=COS(AN2+T1)*G	CBRE 248
	SNN1X(N)=SIN(AN1+T1)*ANG*G	CBRE 249
	SNN2X(N)=SIN(AN2+T1)*G	CBRE 250
	CON1X(N)=CON1*ANG	CBRE 251
	CAN1X(N)=CAN1*ANG	CBRE 252
	WRITE (10001,22) N,ONGX(N),N,AJZX(N),N,AJXX(N),N,AJY,N,CNN1X(N),N,CNN2X(N),N,SNN1X(N),N,SNN2X(N),N,CON1X(N),N,CAN1X(N),N,DELTA(N),N,IAUX(N)	CBRE 253
2000	AJXX(N)=AJXX(N)*ESQ2	CBRE 256
	ELX=ELX/2.0	CBRE 257
C1013	COMPUTE CENTER OF SEA BREEZE CELL.	CBRE 258
1013	ACB=(CXM1X(N)+CXM1X(N))/2.0	CBRE 259
	YCB=(CXM1Y(N)+CXM1Y(N))/2.0	CBRE 260
	CCSB=COS(B)	CBRE 261
	SINB=SIN(B)	CBRE 262
105	RETURN	CBRE 263
C		CBRE 264
C	THIS IS THE COMPUTING ROUTE	CBRE 265
C		CBRE 266
C102	TEST FOR IMPACTED PARTICLE	CBRE 267
102	IF(ZP(J)) 1019,1023,1023	CBRE 268
1019	AX=0.0	CBRE 269
	AY=0.0	CBRE 270
	AZ=-1.0E+8	CBRE 271
	GO TO 105	CBRE 272
C		CBRE 273
C1020	PARTICLE IS ALOFT	CBRE 274
C	THIS CARD ROTATES AND TRANSLATES THE MACRO SYSTEM COORDINATES	CBRE 275
C	INTO THE SEA BREEZE CELL COORDINATES.	CBRE 276
1020	XS=(XP(J)-XCB)*CCSB-(YP(J)-YCB)*SINB	CBRE 277
	ARG=ALM*XS	CBRE 278
C	THE FOLLOWING CARDS ATTENUATE THE WIND FIELD IN THE ADJACENT	CBRE 279
C	REGION.	CBRE 280
	XS=ELX-ABS(XS)	CBRE 281
	IF(XS) 100,103,103	CBRE 282
100	XS=XS*WA	CBRE 283
	GO TO 1030	CBRE 284
103	XS=0.0	CBRE 285
1030	SB=0.0	CBRE 286
	AY=0.0	CBRE 287
	AZ=0.0	CBRE 288
	DO 3000 N=1,NN	CBRE 289
C	THE FOLLOWING TEN CARDS COMPUTE WIND FIELD COEFFICIENTS.	CBRE 290
	ATTN1=EXP(XS+R1RX(N)*ZP(J))	CBRE 291
	ATTN2=EXP(XS+R2RX(N)*ZP(J))	CBRE 292

AAGG=UMGX(N)*TP(J)+FAX(N)	CBRE 293
ARW1=R1IX(N)*ZP(J)+AAGG	CBRE 294
ARW2=R2IX(N)*ZP(J)+AAGG	CBRE 295
SARW1=SIN(ARW1)*ATTN1	CBRE 296
SARW2=SIN(ARW2)*ATTN2	CBRE 297
CARW1=COS(ARW1)*ATTN1	CBRE 298
CARW2=COS(ARW2)*ATTN2	CBRE 299
AAGG=AUXX(N)*COS(ARW1)	CBRE 300
C THE FOLLOWING FIVE CARDS ARE THE WIND FIELD	CBRE 301
AZ=AUX(N)*SIN(ARW1)+(CARW1-CARW2)+AZ	CBRE 302
BB=AAGG*(CARW1*CONIX(N)-SARW1*CANIX(N)-CARW2*CONIX(N)+SARW2*	CBRE 303
ICANIX(N))+BB	CBRE 304
3000 AY=AAGG*(CARW1*CONIX(N)-SARW1*CANIX(N)-CARW2*CONIX(N)+SARW2*	CBRE 305
ICANIX(N))+AY	CBRE 306
C15 ROTATION OF THE WIND VECTORS INTO THE MACROSYSTEM	CBRE 307
15 AX=BB*COSD+AY*SIND	CBRE 308
AY=-BB*SIND+AY*COSD	CBRE 309
GO TO 105	CBRE 310
END	CBRE 311

312*

312 *

```

>IBFTC HEIGH LIST,DECK,M94/2
SUBROUTINE HEIGHT(XA,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
COMMON /SET1/
1 DIAM , DETID , IRIDE , IEXEC , ISIN , ICOSI ,
2 SD , SPAR , SSAM , IAL , IMP1 , IMP2 ,
3 T2M , U , VPR , A , X , Z ,
4 WHY , RMIN , IDISTK , SPARK1 , SPARK2 , SPARK3 ,
5 SPARK4 , SPARK5 , SPARK6 , SPARK7 , SPARK8 , SPARK9
DIMENSION DETID(12),WHY(40)
C *****
C
COMMON /SET2/
1 U , SUBSID , GRINT , BXLL , BXLG , BYLL ,
2 BYLG , IALL , IALG , IYLL , IYLG , XGZ ,
3 YGZ , NBLOCK , HTOPU , ITOPU , ILLM , ULIM ,
4 KLIK , IL , JJ , KK , KP , YP ,
5 ZP , FNAS , IF , PS , VX , VT ,
6 VL , IL , JL , ISADD , WGRINT , NSIRAI ,
7 WLLX , WLLY , WURA , WURY , BOTHII , IPARIN ,
8 IOTOPU , IORIND , IITOPU , IPOSI , IPAROI , ITOPI ,
9 SWINDI , IRROR , ILLIMI , ENDTIM , IC , ISTRAS ,
1 CRASHI , NEOSI , NS , NIO , NII , NW ,
2 NALGHI , JTIMEI , NDRAX , NFREE , N , NCL ,
3 CRMAXI , CRGHI , NCRITP , BL , CRMINA , CRGHIY ,
4 CC , CN , CS , NCCIR , DTECC , ALLEN ,
5 RHO , NA , IGZ , DTHAC , FROG , CRMAXX ,
6 RUPART
DIMENSION TOPLEM(4,4) , INIAR(4) , ITOPLM(3,4)
DIMENSION S(10,10) , SUBSID(400) , IC(18)
DIMENSION XP(200) , IP(200) , ZP(200) , FNAS(200)
DIMENSION IF(200) , PS(200) , ALLEN(200) , RHO(200)
DIMENSION VAI(1500) , VI(1500) , VZ(1500) , IEL(70)
DIMENSION JL(70) , ISADD(70) , WURA(70)
DIMENSION WGRINT(70) , WLLX(70) , WLLY(70)
DIMENSION WURY(70) , BOTHII(70) , SN(6) , CS(6)
DIMENSION CRMINX(6) , CRMAXX(6) , CRGHI(6) , CRMAXI(6)
DIMENSION CRGHI(6) , NCRITP(6) , JOC(6)
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-BXLG)2,4,4
2 IF(YY-BYLL)4,11,3
3 IF(YY-BYLG)11,4,4
C IS PARTICLE OVER RANGE OF TOPOGRAPHY UNDER STUDY
4 IF(XA-IALL)10,6,5
5 IF(XA-IALG)6,10,10
6 IF(YY-IYLL)10,9,7
7 IF(YY-IYLG)9,10,10
8 FORMAT (2F12.4)
C PARTICLE IS BEYOND IN-CORE TOPO SPECIFICATION BUT WITHIN THE TOPO

```


GO TO 23
22 CCY=CCY-GRIN
26 CCX=CCX-GRIN
GO TO 13
27 CCF=CCF-GRIN
GO TO 26
28 XE10X
END

HEIG 119
HEIG 120
HEIG 121
HEIG 122
HEIG 123
HEIG 124
HEIG 125
HEIG 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.

Preceding page blank

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(IJ), 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	0.18200E-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	0.13514E-04	0.11592E+01
0.10000E+03	0.18443E-04	0.11460E+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	0.18162E-04	0.10679E+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	0.18070E-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	0.17610E-04	0.87841E+00
0.31000E+04	0.17527E-04	0.86372E+00
0.33000E+04	0.17420E-04	0.84615E+00
0.35000E+04	0.17329E-04	0.83352E+00
0.37000E+04	0.17250E-04	0.82040E+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	0.16831E-04	0.74462E+00
0.49000E+04	0.16820E-04	0.72667E+00
0.51000E+04	0.16760E-04	0.71151E+00
0.53000E+04	0.16723E-04	0.69447E+00
0.55000E+04	0.16682E-04	0.67882E+00
0.57000E+04	0.16641E-04	0.66342E+00
0.59000E+04	0.16598E-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	0.15829E-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	0.43430E+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	0.14654E-04	0.38443E+00
0.10900E+05	0.14579E-04	0.37444E+00
0.11100E+05	0.14504E-04	0.36545E+00
0.11300E+05	0.14429E-04	0.35535E+00
0.11500E+05	0.14354E-04	0.34773E+00
0.11700E+05	0.14279E-04	0.33711E+00
0.11900E+05	0.14204E-04	0.33040E+00
0.12100E+05	0.14129E-04	0.32180E+00
0.12300E+05	0.14054E-04	0.31324E+00
0.12500E+05	0.13979E-04	0.30462E+00
0.12700E+05	0.13904E-04	0.29500E+00
0.12900E+05	0.13829E-04	0.28735E+00
0.13100E+05	0.13754E-04	0.27875E+00
0.13300E+05	0.13679E-04	0.26950E+00
0.13500E+05	0.13604E-04	0.26027E+00
0.13700E+05	0.13529E-04	0.25204E+00
0.13900E+05	0.13454E-04	0.24380E+00
0.14100E+05	0.13379E-04	0.23552E+00
0.14300E+05	0.13304E-04	0.23176E+00
0.14500E+05	0.13229E-04	0.22500E+00
0.14700E+05	0.13154E-04	0.21825E+00
0.14900E+05	0.13079E-04	0.21149E+00
0.15100E+05	0.13004E-04	0.20473E+00
0.15300E+05	0.12929E-04	0.19890E+00
0.15500E+05	0.12854E-04	0.19230E+00
0.15700E+05	0.12779E-04	0.18615E+00
0.15900E+05	0.12704E-04	0.18000E+00
0.16100E+05	0.12629E-04	0.17385E+00
0.16300E+05	0.12554E-04	0.16770E+00
0.16500E+05	0.12479E-04	0.16155E+00
0.16700E+05	0.12404E-04	0.15574E+00
0.16900E+05	0.12329E-04	0.15146E+00
0.17100E+05	0.12254E-04	0.14719E+00
0.17300E+05	0.12179E-04	0.14292E+00
0.17500E+05	0.12104E-04	0.13865E+00
0.17700E+05	0.12029E-04	0.13430E+00
0.17900E+05	0.11954E-04	0.13011E+00
0.18100E+05	0.11879E-04	0.12584E+00
0.18300E+05	0.11804E-04	0.12177E+00
0.18500E+05	0.11729E-04	0.11770E+00
0.18700E+05	0.11654E-04	0.11396E+00
0.18900E+05	0.11579E-04	0.11000E+00
0.19100E+05	0.11504E-04	0.10615E+00
0.19300E+05	0.11429E-04	0.10225E+00
0.19500E+05	0.11354E-04	0.98341E-01
0.19700E+05	0.11279E-04	0.94430E-01
0.19900E+05	0.11204E-04	0.91075E-01
0.20100E+05	0.11129E-04	0.87182E-01
0.20300E+05	0.11054E-04	0.84180E-01
0.20500E+05	0.10979E-04	0.81187E-01
0.20700E+05	0.10904E-04	0.78090E-01
0.20900E+05	0.10829E-04	0.76193E-01
0.21100E+05	0.10754E-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	0.66887E-01
0.22300E+05	0.14221E-04	0.64912E-01
0.22500E+05	0.14242E-04	0.62918E-01
0.22700E+05	0.14264E-04	0.60924E-01
0.22900E+05	0.14286E-04	0.58829E-01
0.23100E+05	0.14308E-04	0.56711E-01
0.23300E+05	0.14329E-04	0.54571E-01
0.23500E+05	0.14351E-04	0.52404E-01
0.23700E+05	0.14373E-04	0.50195E-01
0.23900E+05	0.14394E-04	0.47946E-01
0.24100E+05	0.14416E-04	0.45745E-01
0.24300E+05	0.14438E-04	0.43513E-01
0.24500E+05	0.14459E-04	0.41272E-01
0.24700E+05	0.14481E-04	0.39032E-01
0.24900E+05	0.14502E-04	0.36795E-01
0.25100E+05	0.14524E-04	0.34571E-01
0.25300E+05	0.14545E-04	0.32351E-01
0.25500E+05	0.14567E-04	0.30137E-01
0.25700E+05	0.14588E-04	0.27928E-01
0.25900E+05	0.14610E-04	0.25724E-01
0.26100E+05	0.14631E-04	0.23525E-01
0.26300E+05	0.14653E-04	0.21331E-01
0.26500E+05	0.14674E-04	0.19142E-01
0.26700E+05	0.14696E-04	0.16958E-01
0.26900E+05	0.14717E-04	0.14779E-01
0.27100E+05	0.14738E-04	0.12605E-01
0.27300E+05	0.14760E-04	0.10436E-01
0.27500E+05	0.14781E-04	0.08272E-01
0.27700E+05	0.14802E-04	0.06113E-01
0.27900E+05	0.14824E-04	0.03959E-01
0.28100E+05	0.14845E-04	0.01810E-01
0.28300E+05	0.14866E-04	0.00000E-01
0.28500E+05	0.14887E-04	0.00000E-01
0.28700E+05	0.14909E-04	0.00000E-01
0.28900E+05	0.14930E-04	0.00000E-01
0.29100E+05	0.14951E-04	0.00000E-01
0.29300E+05	0.14972E-04	0.00000E-01
0.29500E+05	0.14993E-04	0.00000E-01
0.29700E+05	0.15015E-04	0.00000E-01
0.29900E+05	0.15036E-04	0.00000E-01
0.30100E+05	0.15057E-04	0.00000E-01
0.30300E+05	0.15078E-04	0.00000E-01
0.30500E+05	0.15099E-04	0.00000E-01
0.30700E+05	0.15120E-04	0.00000E-01
0.30900E+05	0.15141E-04	0.00000E-01
0.31100E+05	0.15162E-04	0.00000E-01
0.31300E+05	0.15183E-04	0.00000E-01
0.31500E+05	0.15204E-04	0.00000E-01
0.31700E+05	0.15225E-04	0.00000E-01
0.31900E+05	0.15246E-04	0.00000E-01
0.32100E+05	0.15267E-04	0.00000E-01
0.32300E+05	0.15288E-04	0.00000E-01
0.32500E+05	0.15319E-04	0.00000E-01
0.32700E+05	0.15340E-04	0.00000E-01
0.32900E+05	0.15361E-04	0.00000E-01
0.33100E+05	0.15382E-04	0.00000E-01
0.33300E+05	0.15403E-04	0.00000E-01
0.33500E+05	0.15424E-04	0.00000E-01

0.33730E+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	0.86511E-02
0.35300E+05	0.15666E-04	0.83631E-02
0.35500E+05	0.15691E-04	0.80750E-02
0.35700E+05	0.15715E-04	0.77869E-02
0.35900E+05	0.15740E-04	0.74988E-02
0.36100E+05	0.15764E-04	0.72107E-02
0.36300E+05	0.15789E-04	0.69226E-02
0.36500E+05	0.15814E-04	0.66345E-02
0.36700E+05	0.15838E-04	0.63464E-02
0.36900E+05	0.15863E-04	0.60583E-02
0.37100E+05	0.15887E-04	0.57702E-02
0.37300E+05	0.15912E-04	0.54821E-02
0.37500E+05	0.15936E-04	0.51940E-02
0.37700E+05	0.15961E-04	0.49059E-02
0.37900E+05	0.15985E-04	0.46178E-02
0.38100E+05	0.16010E-04	0.43297E-02
0.38300E+05	0.16034E-04	0.40416E-02
0.38500E+05	0.16058E-04	0.37535E-02
0.38700E+05	0.16083E-04	0.34654E-02
0.38900E+05	0.16107E-04	0.31773E-02
0.39100E+05	0.16131E-04	0.28892E-02
0.39300E+05	0.16155E-04	0.26011E-02
0.39500E+05	0.16180E-04	0.23130E-02
0.39700E+05	0.16204E-04	0.20249E-02
0.39900E+05	0.16228E-04	0.17368E-02
0.40100E+05	0.16253E-04	0.14487E-02
0.40300E+05	0.16277E-04	0.11606E-02
0.40500E+05	0.16301E-04	0.08725E-02
0.40700E+05	0.16325E-04	0.05844E-02
0.40900E+05	0.16349E-04	0.02963E-02
0.41100E+05	0.16373E-04	0.00082E-02
0.41300E+05	0.16397E-04	0.00001E-02
0.41500E+05	0.16421E-04	0.00000E-02
0.41700E+05	0.16445E-04	0.00000E-02
0.41900E+05	0.16469E-04	0.00000E-02
0.42100E+05	0.16493E-04	0.00000E-02
0.42300E+05	0.16517E-04	0.00000E-02
0.42500E+05	0.16541E-04	0.00000E-02
0.42700E+05	0.16565E-04	0.00000E-02
0.42900E+05	0.16589E-04	0.00000E-02
0.43100E+05	0.16613E-04	0.00000E-02
0.43300E+05	0.16637E-04	0.00000E-02
0.43500E+05	0.16661E-04	0.00000E-02
0.43700E+05	0.16685E-04	0.00000E-02
0.43900E+05	0.16709E-04	0.00000E-02
0.44100E+05	0.16733E-04	0.00000E-02
0.44300E+05	0.16757E-04	0.00000E-02
0.44500E+05	0.16781E-04	0.00000E-02
0.44700E+05	0.16805E-04	0.00000E-02
0.44900E+05	0.16829E-04	0.00000E-02
0.45100E+05	0.16853E-04	0.00000E-02
0.45300E+05	0.16877E-04	0.00000E-02
0.45500E+05	0.16901E-04	0.00000E-02

0.45700E+05	0.16721E-04	0.19488E-02
0.45900E+05	0.16745E-04	0.18441E-02
0.46100E+05	0.16768E-04	0.18488E-02
0.46300E+05	0.16992E-04	0.17995E-02
0.46500E+05	0.17018E-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	0.14201E-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

UNDERFLOW AT 54221 IN M2

UNDERFLOW AT 54222 IN M2

UNDERFLOW AT 54222 IN M2

UNDERFLOW AT 54221 IN M2

THE FOLLOWING WIND VECTORS HAVE BEEN USED TO DEFINE AN ATMOSPHERE UP TO TIME 06400.

VECTOR HEIGHT	HORIZONTAL COORDINATES		VECTOR COMPONENTS		
	X	Y	VX	VY	VZ
475.500	993804.000	999589.541	0.000	0.300	0.000
1219.200	993804.000	999589.541	0.000	0.300	0.000
1524.000	993804.000	999589.541	0.000	0.000	0.000
1828.800	993804.000	999589.541	0.000	0.000	0.000
2133.600	993804.000	999589.541	0.000	0.155	0.000
2438.400	993804.000	999589.541	2.940	1.070	0.000
2743.200	993804.000	999589.541	5.420	3.129	0.000
3048.000	993804.000	999589.541	6.960	4.023	0.000
3352.800	993804.000	999589.541	3.484	-2.011	0.000
3657.600	993804.000	999589.541	3.484	-2.011	0.000
3962.400	993804.000	999589.541	7.043	-1.242	0.000
4267.200	993804.000	999589.541	12.766	2.251	0.000
4572.000	993804.000	999589.541	0.000	0.000	0.000
4876.800	993804.000	999589.541	0.000	0.000	0.000
5181.600	993804.000	999589.541	0.000	0.000	0.000
5486.400	993804.000	999589.541	0.000	0.000	0.000
5791.200	993804.000	999589.541	0.000	0.000	0.000
6096.000	993804.000	999589.541	0.000	0.000	0.000
6400.000	1026389.000	1019976.125	0.000	0.000	0.000
6704.000	1026389.000	1019976.125	0.000	0.000	0.000
7008.000	1026389.000	1019976.125	0.000	0.000	0.000
7312.000	1026389.000	1019976.125	-1.771	0.249	0.000
7616.000	1026389.000	1019976.125	-0.981	0.915	0.000
7920.000	1026389.000	1019976.125	1.447	1.051	0.000
8224.000	1026389.000	1019976.125	3.484	0.804	0.000
8528.000	1026389.000	1019976.125	5.312	0.747	0.000
8832.000	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	3.875	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	WLLX	WLLY	WURX	WURY
914.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
1066.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
1219.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
1371.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
1524.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
1676.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
1828.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
1981.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
2133.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
2286.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
2438.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
2590.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
2743.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
2895.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
3048.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
3200.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
3352.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
3505.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
3657.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
3810.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
3962.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
4114.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
4267.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
4419.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
4572.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
4724.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
4876.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
5029.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
5181.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
5334.000	4500.000	997750.000	997750.000	1024200.000	1024700.000
5486.400	4500.000	997750.000	997750.000	1024200.000	1024700.000
5638.800	4500.000	997750.000	997750.000	1024200.000	1024700.000
5791.200	4500.000	997750.000	997750.000	1024200.000	1024700.000
5943.600	4500.000	997750.000	997750.000	1024200.000	1024700.000
6096.000	4500.000	997750.000	997750.000	1024200.000	1024700.000

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

ALPHA = 457.200 METERS, BETA = 32000.000 METERS.

WIND COMPONENTS-----

[illegible]

LEVEL	2	BASE AT	1000.800 METERS
EAST-WEST	ROW	1	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000
EAST-W	ROW	2	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000
EAST-W	ROW	3	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000
EAST-W	ROW	4	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000
EAST-W	ROW	5	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000
EAST-WEST	ROW	6	
C	300	0.00000	0.00000
C	300	0.00000	0.00000
C	300	0.00000	0.00000

LEVEL 35	BASE AT	6090.000 METERS				
EAST-WEST ROW	1					
12.76600	12.40720	11.55457	11.25178	10.60150	9.88608	9.26819
2.25100	2.39990	2.62927	2.87952	3.14944	3.44614	3.70280
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EAST-WEST ROW	2					
12.56550	12.01592	11.44111	10.85199	10.26402	9.67492	9.26819
2.34341	2.55317	2.80993	3.04540	3.28952	3.53404	3.70280
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EAST-WEST ROW	3					
12.25293	11.65553	11.06309	10.50159	9.97319	9.47070	9.26818
2.46395	2.71147	2.95575	3.19990	3.41023	3.51629	3.70280
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EAST-WEST ROW	4					
11.94543	11.39041	10.70839	10.17429	9.70388	9.27948	9.26819
2.59112	2.85917	3.10597	3.32670	3.52202	3.69818	3.70280
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EAST-WEST ROW	5					
11.62457	10.92671	10.34295	9.84942	9.43360	9.26819	9.26819
2.72477	3.01362	3.25675	3.46160	3.63418	3.70280	3.70280
0.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
EAST-WEST ROW	6					
11.29207	10.50137	9.93555	9.45012	9.26819	9.26819	9.26819
2.87941	3.14044	3.42581	3.61074	3.70280	3.70280	3.70280
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 943.120 METERS

DAVIES EQUATIONS ARE INACCURATE FOR 12473.701 MICRONS AT 944.703 METERS

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, u_0 , which would otherwise exist in the absence of the ground disturbance. The relationship between the change in the wind field $\Delta u(x, y, z)$ 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 non-turbulent 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 Ω are given by

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

where Θ is the latitude, and Ω is the sidereal day frequency which equals $7.3 \times 10^{-5} \text{ sec}^{-1}$. For our problems all the components of Ω 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

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

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,

$$P_e = (\rho_e k T_e / m), \quad (\text{A.2})$$

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:

$$\partial \rho / \partial t + \nabla \cdot (\rho \underline{y}) = 0 \quad (\text{A.3})$$

and

$$d\underline{y}/dt = 2\underline{y} \times \underline{\Omega} - \nabla \psi + \underline{G} , \quad (\text{A.4})$$

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

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

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

$$0 = -\partial \psi / \partial x, \quad 0 = -2u_o \Omega_z - \partial \psi / \partial y, \quad 0 = 2u_o \Omega_y - \partial \psi / \partial z - G . \quad (\text{A.6})$$

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

$$\psi = Ay + Bz + \psi_{or} , \quad (\text{A.7})$$

which when substituted into the foregoing equations gives

$$A = -2u_o \Omega_z, \quad B = -G + 2u_o \Omega_y \approx -G , \quad (\text{A.8})$$

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

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 $\psi_{\text{or}}/|A| = y_c$ and $\psi_{\text{or}}/|B| = z_c$, which are respectively the distances over which the independent changes in ψ equal the value at the origin. We have

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

and

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

Using a maximum value of $\Omega_z = \Omega = 7.3 \times 10^{-5} \text{ sec}^{-1}$ and a value of $u_o = 4400 \text{ cm sec}^{-1}$ (corresponding to a 100 mph wind) gives a value of $y_c = 4.5 \times 10^9 \text{ cm} = 3 \times 10^4 \text{ mi}$ 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

$$\vec{v} = \vec{i}u_o \quad (\text{A.12})$$

and density

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

where

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

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

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

where β is defined as the tropospheric density attenuation constant,

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

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

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

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 ($\partial/\partial t = 0$)

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

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

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

and

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

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

$$\eta = \bar{\rho}/\rho_o . \quad (\text{A.22})$$

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

$$\bar{\psi} = \psi(\rho_0 + \bar{\rho}) - \psi(\rho_0) = \gamma \left(P_e / \rho_e^\gamma \right) \rho_0^{\gamma-1} \eta . \quad (\text{A. 23})$$

Using Eq. (A.13) for $\rho_0(z)$ gives

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

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

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

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

and

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

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, $\ln \rho_0 = \ln \rho_e - \beta z$, thereby yielding

$$\left(\frac{\partial \bar{u}}{\partial x} + \frac{\partial \bar{v}}{\partial y} + \frac{\partial \bar{w}}{\partial z} \right) + u_0 \frac{\partial}{\partial x} \eta - \beta \bar{w} = 0 , \quad (\text{A. 28})$$

$$u_0 \frac{\partial \bar{u}}{\partial x} = 2 \left(\bar{v} \Omega_z - \bar{w} \Omega_y \right) - \frac{\gamma k T_e}{m} \frac{\partial \eta}{\partial x} , \quad (\text{A. 29})$$

$$u_0 \frac{\partial \bar{v}}{\partial x} = 2 \left(\bar{w} \Omega_x - \bar{u} \Omega_z \right) - \frac{\gamma k T_e}{m} \frac{\partial \eta}{\partial y} , \quad (\text{A. 30})$$

and

$$u_o \frac{\partial \bar{w}}{\partial x} = 2(\bar{u}\Omega_y - \bar{v}\Omega_x) + (\gamma - 1)G\eta - \frac{\gamma kT}{m} e \frac{\partial \eta}{\partial z} \quad (A.31)$$

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.

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

and

(A.32)

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

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

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

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 $\underline{k}$ 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

$$k_z = k_z(k_x, k_y) \quad (A.34)$$

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, $\bar{w}$ now becomes

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

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

$$ak_z^2 + b(ik_z) + c = 0 , \quad (\text{A.35})$$

where

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

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

and

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

in which

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

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

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

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

$$\begin{aligned} b_{11} &= ik_x \xi + iu_o k_x \Omega \omega_y, & b_{12} &= ik_y \xi - iu_o k_x \Omega \omega_x, & b_{13} &= (ik_z - \beta) \xi + (u_o k_x)^2 \\ b_{21} &= (ik_x u_o) \xi + ik_x \sigma \Omega \omega_y, & b_{22} &= -\Omega \omega_z \xi - (ik_x \sigma) \Omega \omega_x, & b_{23} &= \Omega \omega_y \xi - (ik_x \sigma) (ik_x u_o), \\ b_{31} &= \Omega \omega_z \xi + (ik_y \sigma) \Omega \omega_y, & b_{32} &= ik_x u_o \xi - (ik_y \sigma) \Omega \omega_x, & b_{33} &= -\Omega \omega_x \xi + k_x k_y \sigma u_o, \end{aligned} \quad (\text{A } 39.)$$

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:

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

and

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

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

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

We now let

$$\lambda = -ik_z , \quad (\text{A.43})$$

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

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

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

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

Since we are primarily interested in short range effects, we shall freely make use of the inequality $k_x^2 + k_y^2 \gg \beta^2$ (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)

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

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

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

and

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

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

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

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

and

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

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

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

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

$$\nabla\phi = -\hat{i} \frac{\partial f}{\partial x} - \hat{j} \frac{\partial f}{\partial y} + \hat{k} \quad (\text{A. 53})$$

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

$$[(\mathbf{v} \cdot \nabla\phi)] = 0 \quad \text{along } z = f(x, y), \quad (\text{A. 54})$$

where

$$\underline{v} = \underline{j}u + \underline{j}v + \underline{k}w = \underline{j}u_o + (\underline{j}\bar{u} + \underline{j}\bar{v} + \underline{k}\bar{w}) . \quad (A.55)$$

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

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

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

$$\begin{aligned} & \int_{k_x} \int_{k_y} \left[1 - T(k_x, k_y) \left(\frac{\partial f}{\partial x} \right) - U(k_x, k_y) \left(\frac{\partial f}{\partial y} \right) \right] \\ & C(k_x, k_y) e^{ik_x x} e^{ik_y y} e^{\left[- (k_x^2 + k_y^2)^{1/2} f(x, y) \right]} dk_x dk_y \\ & = u_o \int_{k_x} \int_{k_y} (ik_x) F(k_x, k_y) e^{ik_x x} e^{ik_y y} dk_x dk_y , \end{aligned} \quad (A.57)$$

where

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

and

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

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 $\exp - \left[(k_x^2 + k_y^2)^{1/2} f(x, y) \right]$ 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 $u_0(\partial f/\partial x)$ 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_0 . The neglect of $\bar{v}(\partial f/\partial y)$ as compared to $u_0(\partial f/\partial x)$ is somewhat difficult to justify under all cases. Although $\bar{v}$ is assumed small compared to u_0 , we must also be sure that $(\partial f/\partial y)$ is not substantially greater than $(\partial f/\partial x)$.

The a priori assumption

$$\bar{u} \left(\frac{\partial f}{\partial x} \right) + \bar{v} \left(\frac{\partial f}{\partial y} \right) < u_0 \left(\frac{\partial f}{\partial x} \right) \quad (\text{A. 59})$$

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

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

Introduction of the functions

$$\xi = 1 - \Gamma = - \sum_{n=1} \frac{\phi^n}{n!} \quad (\text{A. 61})$$

where

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

and

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

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

$$\int C(\underline{k}) e^{i\underline{k} \cdot \underline{r}} d\underline{k} = u_0 \int H(\underline{k}) e^{i\underline{k} \cdot \underline{r}} d\underline{k} + \int \Delta C(\underline{k}) e^{i\underline{k} \cdot \underline{r}} d\underline{k} \quad (\text{A.64})$$

where

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

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

$$C(\underline{k}') = u_0 H(\underline{k}') + \left(\frac{1}{2\pi} \right)^2 \int_{\underline{k}} \int_{\underline{r}} \Delta(\underline{k}, \underline{r}) C(\underline{k}) e^{i(\underline{k} - \underline{k}') \cdot \underline{r}} d\underline{k} d\underline{r} \quad (\text{A.65})$$

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 $C(\underline{k})$, denoted by $C^{(1)}$, is deduced by completely disregarding the second term of the right-hand side. Thus

$$C^{(1)}(\underline{k}') = u_0 H(\underline{k}') = u_0 (i k'_x) F(\underline{k}') \quad (\text{A.66})$$

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

$$C^{(2)}(\underline{k}') = u_0 H(\underline{k}') + \left(\frac{1}{2\pi} \right)^2 \int_{\underline{k}} \int_{\underline{r}} \Delta(\underline{k}, \underline{r}) \left[u_0 H(\underline{k}) \right] e^{i(\underline{k} - \underline{k}') \cdot \underline{r}} d\underline{k} d\underline{r} \quad (\text{A.67})$$

It follows by inspection that the n th approximation to $C(\underline{k})$ is given by

$$C^{(n)}(\underline{k}') = u_0 H(\underline{k}') + \left(\frac{1}{2\pi}\right)^2 \int_{\underline{k}} \int_{\underline{r}} \Delta(\underline{k}, \underline{r}) C^{(n-1)}(\underline{k}) e^{i(\underline{k} - \underline{k}') \cdot \underline{r}} d\underline{k} d\underline{r} , \quad (\text{A. 68})$$

with

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

The corresponding Fourier transforms of the perturbed x and y components of the wind field, $A(\underline{k})$ and $B(\underline{k})$ 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_0 . Mathematically, the developed theory is valid so long as

$$\bar{u}^2 + \bar{v}^2 + \bar{w}^2 < u_0^2 . \quad (\text{A. 69})$$

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 $\bar{u}(\underline{r})$, $\bar{v}(\underline{r})$, and $\bar{w}(\underline{r})$.

The value of computing $C(\underline{k})$ 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 $C(\underline{k})$ and the dependence of $C(\underline{k})$ 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 $C(\underline{k})$ 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 $C(\underline{k})$.

Let

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

from which we have:

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

and

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

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

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

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

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

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

$$(2)^{1/2}(|k_o|h) \ll 1, \quad (A.75)$$

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

$$C^{(2)}(\underline{k}') = iu_o(k_o h)\delta(k' - k_o) + \left(\frac{1}{2\pi}\right) \int \int_{k, x} \Delta(k, x) \left[iu_o(k_o h)\delta(k - k_o) \right] e^{i(k - k')} dk dx, \quad (A.76)$$

where

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

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

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

where

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

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

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

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.

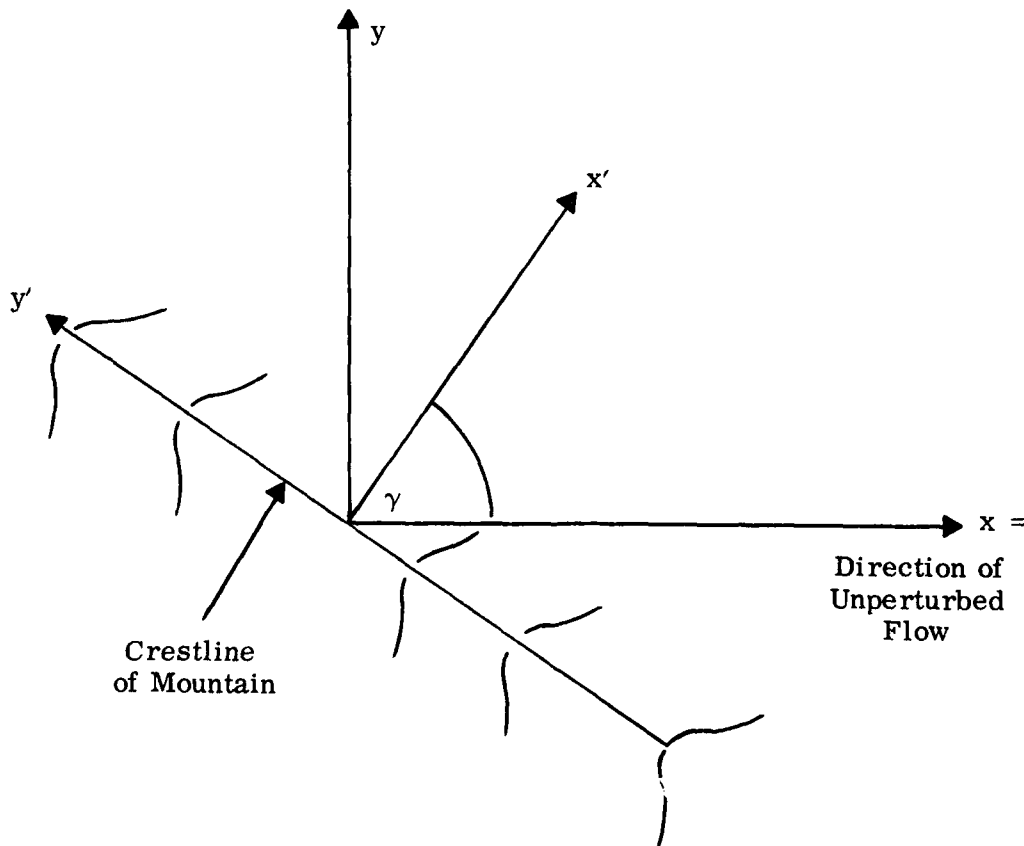


Figure A.1. Mountain Ridge Not Perpendicular to Flow

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

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

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

$$x = x' \cos \gamma - y' \sin \gamma, \quad y = x' \sin \gamma + y' \cos \gamma \quad (\text{A. 81})$$

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

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

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

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

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

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

where

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

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

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

Using the inversion formula with $C(k_x, k_y) = ik_x u_0 F(k_x, k_y)$ yields the following expression for $\bar{w}(x, y, z)$:

$$\begin{aligned}\bar{w}(x, y, z) &= u_0 \iint (ik_x) e^{ik_x x} e^{ik_y y} F(k_x, k_y) e^{-\left(k_x^2 + k_y^2\right)^{1/2} z} dk_x dk_y \\ &= u_0 \frac{\partial}{\partial x} \iint e^{i(k'_x x' + k'_y y')} F'(k'_x, k'_y) e^{-\left(k'^2_x + k'^2_y\right)^{1/2} z} dk'_x dk'_y.\end{aligned}\quad (\text{A.87})$$

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

$$\bar{w}(x, y, z) = -2u_0(ah) \lambda \cos \gamma \frac{x'}{\left(x'^2 + \lambda^2\right)^{3/2}}, \quad (\text{A.88})$$

where

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

and

$$\lambda = z + a. \quad (\text{A.89})$$

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

$$\bar{u}(x, y, z) = -u_0(ah) \cos^2 \gamma \left[\frac{(x'^2 - \lambda^2)}{(x'^2 + \lambda^2)^2} \right], \quad (\text{A.90})$$

and

$$\bar{v}(x, y, z) = -u_0(ah) \cos \gamma \sin \gamma \left[\frac{(x'^2 - \lambda^2)}{(x'^2 + \lambda^2)^2} \right]. \quad (\text{A.91})$$

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_0 when the flow is perpendicular to the crestline (i.e., $\gamma = 0$). In this case we have

$$(|\Delta \mathbf{v}|/u_0) = (\bar{u}^2 + \bar{w}^2)^{1/2}/u_0 = r = ah/\left[x^2 + (z+a)^2\right], \quad (\text{A.92})$$

where the altitude z is defined in the range $z \geq f(x) = h/(1 + (x/a)^2)$. For a pre-assigned 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

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

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

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

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

A Single Mountain

Investigations have shown that a suitable representation for a mountain is

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

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

$$F(k_x, k_y) = \frac{A}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-ik \cdot r} \frac{1}{(a^2 + r^2)^{3/2}} dx dy, \\ F(k_x, k_y) = F(k) = \frac{A}{(2\pi)k^{1/2}} \int_0^{\infty} \frac{r^{1/2}}{(a^2 + r^2)^{3/2}} J_0(kr) (kr)^{1/2} dr, \quad (\text{A. 96})$$

where

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

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

$$F(k) = \frac{A}{2\pi a} e^{-ak}. \quad (\text{A. 97})$$

If u_0 is the unperturbed velocity, it follows that the first-order correction to the vertical component of the wind is given by

$$\begin{aligned}\bar{w}(x, y, z) &= u_0 \iint (ik_x) F(k) e^{i(k_x x + k_y y)} e^{-kz} dk_x dk_y, \\ \bar{w}(x, y, z) &= \frac{u_0 A}{a} \frac{\partial}{\partial x} \int_0^\infty e^{-(a+z)k} J_0(kr) k dk, \\ \bar{w}(x, y, z) &= \frac{-3\lambda A u_0}{a} \frac{x}{(x^2 + y^2 + \lambda^2)^{5/2}} = \frac{-3\lambda a^2 h x u_0}{(r^2 + \lambda^2)^{5/2}}, \quad (A.98)\end{aligned}$$

where

$$\lambda = (z + a).$$

The changes in the x and y components of velocity are determined from Eqs. (A.50) and (A.51). Thus, we have:

$$\begin{aligned}\bar{u}(x, y, z) &= -u_0 \iint \frac{(ik_x)(ik_y)}{k} F(k) e^{i(k_x x + k_y y)} e^{-kz} dk_x dk_y \\ \bar{u}(x, y, z) &= -u_0 \frac{\partial}{\partial x} \int_0^\infty \int_0^{2\pi} F(k) e^{ikr \cos \phi} e^{-kz} dk d\phi, \\ \bar{u}(x, y, z) &= u_0 (a^2 h) \frac{(y^2 + \lambda^2 - 2x^2)}{(r^2 + \lambda^2)^{5/2}}; \quad (A.99)\end{aligned}$$

and

$$\bar{v}(x, y, z) = -u_0 \iint \frac{(ik_y)(ik_x)}{k} F(k) e^{i(k_x x + k_y y)} e^{-kz} dk_x dk_y ,$$

$$\bar{v}(x, y, z) = -u_0 \frac{\partial^2}{\partial x \partial y} \int_0^{2\pi} \int_0^\infty F(k) e^{ikr \cos \phi} e^{-kz} dk d\phi ,$$

$$\bar{v}(x, y, z) = - \frac{3u_0 (a^2 h)_{xy}}{(r^2 + \lambda^2)^{5/2}} . \quad (A.100)$$

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

$$u_p = u_0 + \bar{u} , \quad (A.101)$$

and

$$w_p = -V_F + \bar{w} , \quad (A.102)$$

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

$$w_p = -V_F + \bar{w} = -\frac{\partial \Phi}{\partial x}(x, z) , \quad (A.103)$$

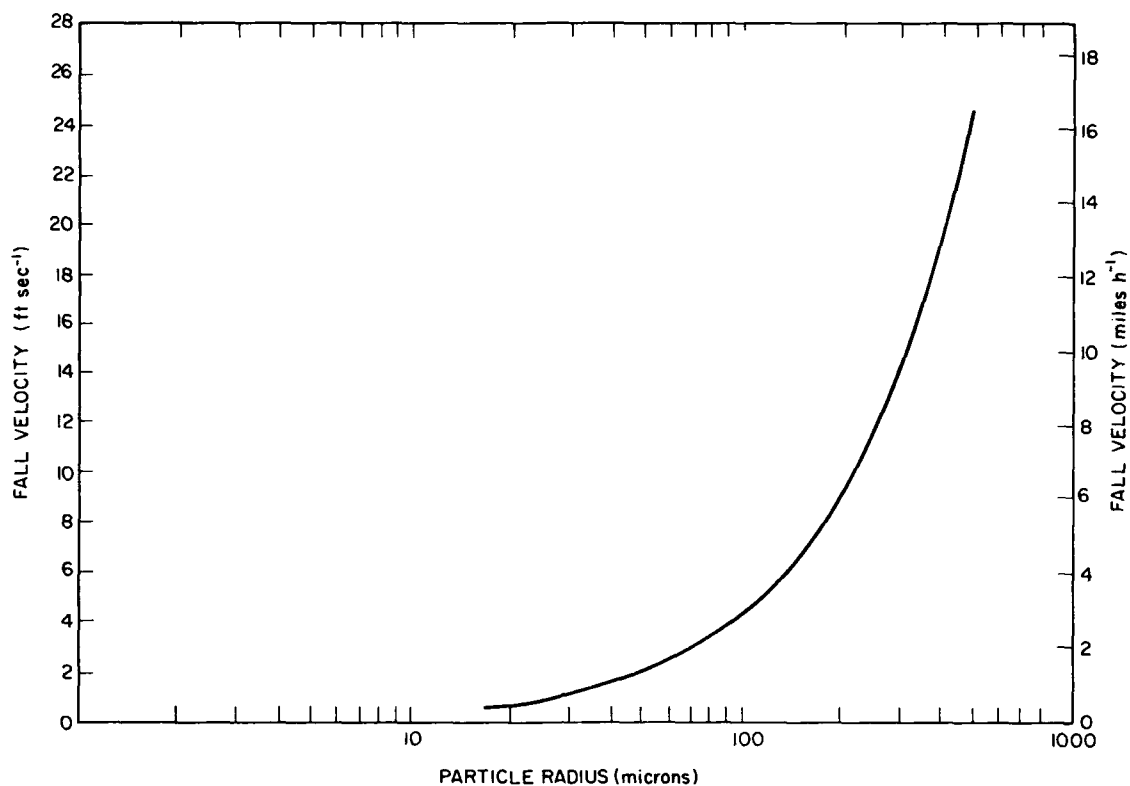


Figure A.2. Fall Velocity vs Particle Size

and

$$u_p = u_o + \bar{u} \equiv \frac{\partial \Phi}{\partial z}(x, z) \quad . \quad (\text{A.104})$$

On the other hand, the equation for the flow lines is

$$\frac{dx}{u_p} = \frac{dz}{w_p} \quad \text{or} \quad -w_p dx + u_p dz = 0 \quad , \quad (\text{A.105})$$

which when used with Eqs. (A.103) and (A.104) gives

$$\frac{\partial \Phi}{\partial x} dx + \frac{\partial \Phi}{\partial z} dz = 0 \quad \text{or} \quad d\Phi = 0 \quad . \quad (\text{A.106})$$

We thus see that the curves for which

$$\Phi = \text{constant} \quad (\text{A.107})$$

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

$$\bar{u} = \frac{\partial \Psi}{\partial z} \quad , \quad (\text{A.108})$$

and

$$\bar{w} = -\frac{\partial \Psi}{\partial x} \quad , \quad (\text{A.109})$$

where the perturbed wind stream function $\Psi(x, z)$ is

$$\Psi = \frac{-u_o (ah)(z + a)}{(z + a)^2 + x^2} \quad . \quad (\text{A.110})$$

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 derivatives w_p and u_p , is

$$\Phi = u_o z + \Psi + V_F x \quad . \quad (\text{A.111})$$

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

$$\text{constant} = C = \bar{z} + r\bar{x} - \alpha \frac{(1 + \bar{z})}{(1 + \bar{z})^2 + \bar{x}^2} \quad , \quad (\text{A.112})$$

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

$$\bar{z} = C' - r\bar{x} , \quad (\text{A.113})$$

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

$$\left(\frac{z}{a}\right) = \left(\frac{h}{a}\right) \frac{1}{1 + (x/a)^2} = \frac{\alpha}{1 + (x/a)^2} .$$

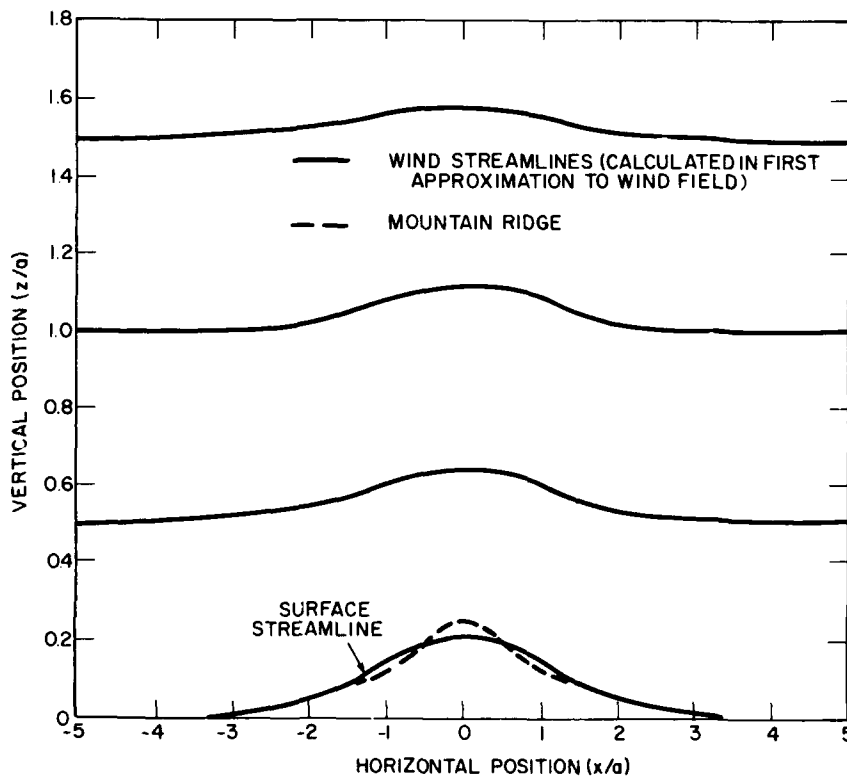


Figure A.3. Wind Streamlines for $\alpha = 0.25$ (Note vertical scale is amplified 10 times.)

Figure A.4. Wind Streamlines for $\alpha = 0.50$ (Note vertical scale is amplified 10 times.)

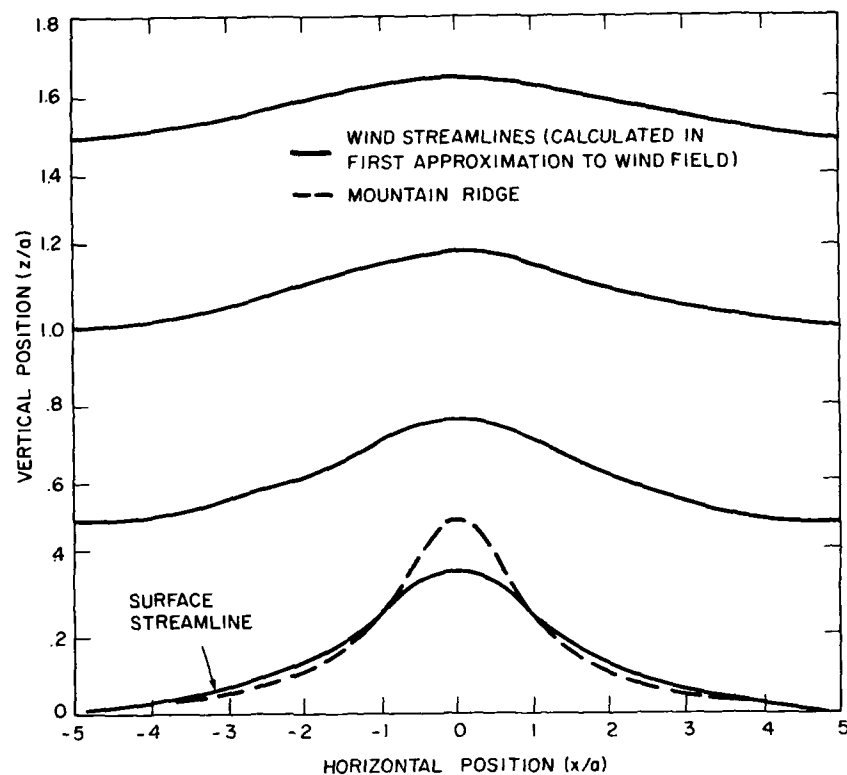
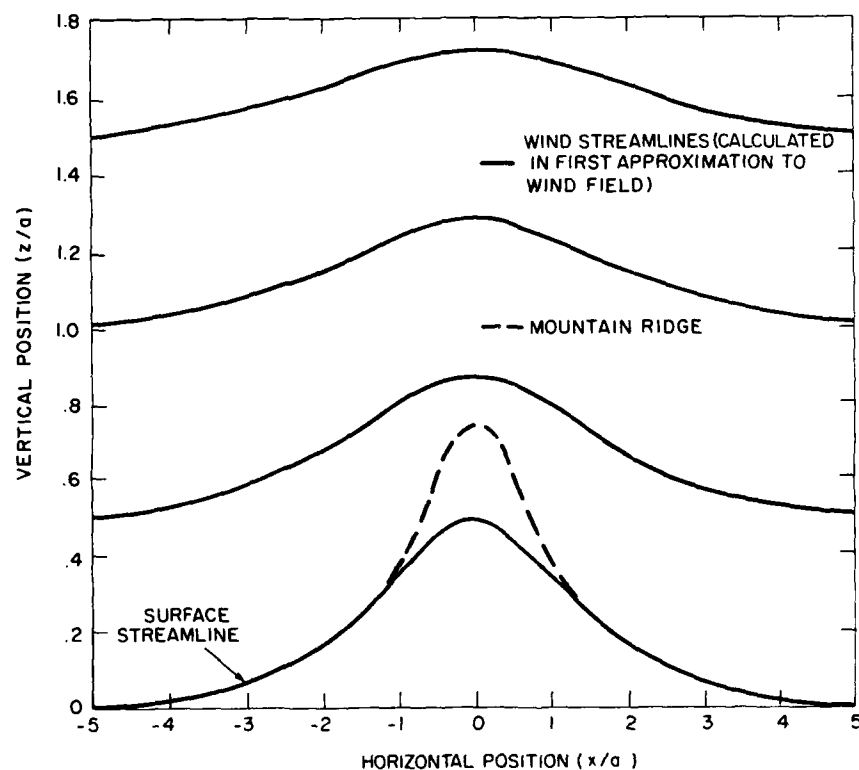


Figure A.5. Wind Streamlines for $\alpha = 0.75$ (Note vertical scale is amplified 10 times.)



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

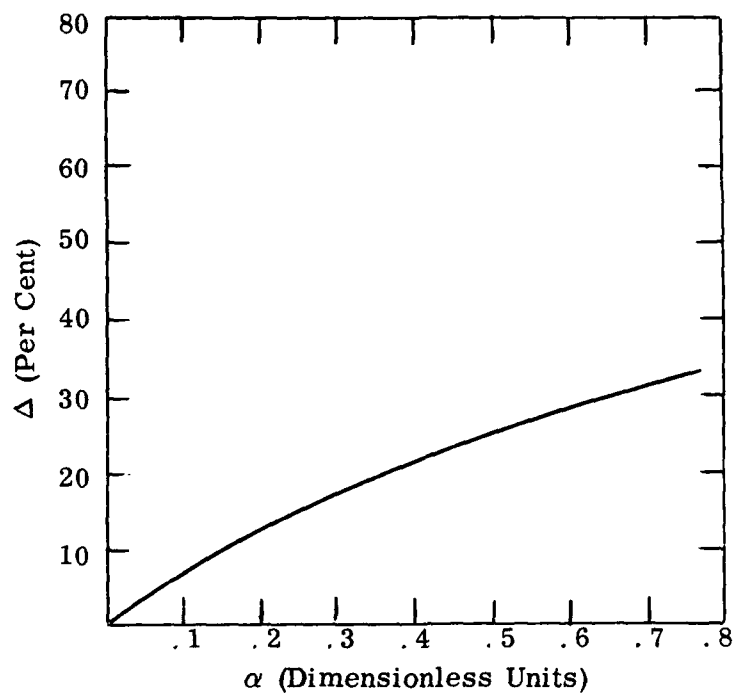
$$C^{(1)}(k) = \frac{u_o(ah)}{2} (ik) e^{-|k|a} .$$

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

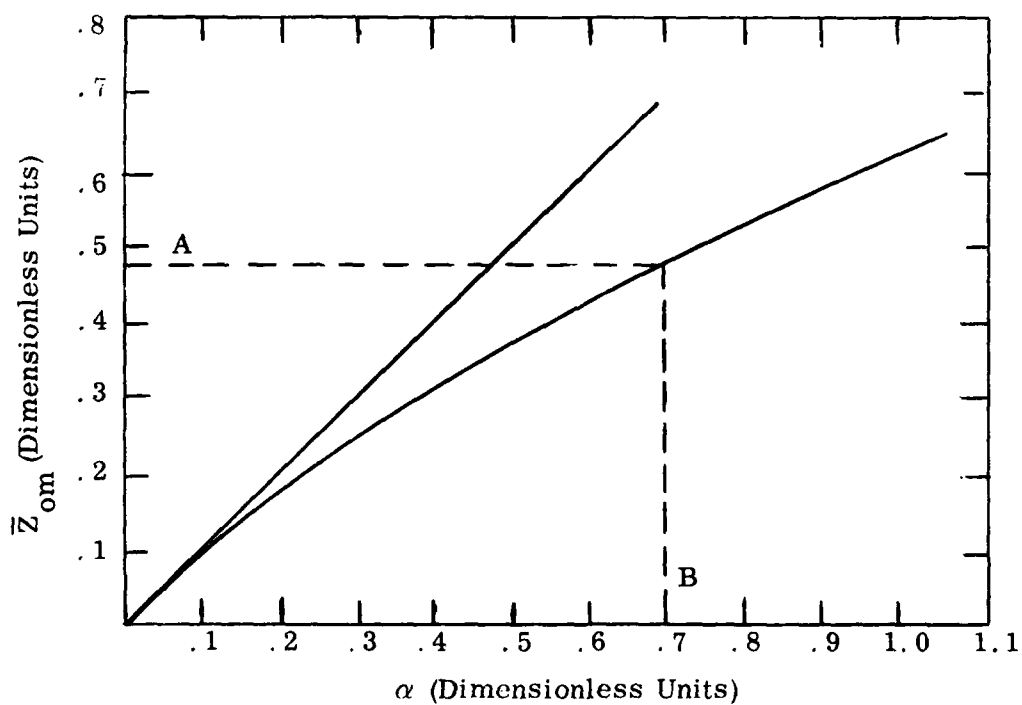
$$\Delta = \frac{(h/a) - \bar{Z}_{om}}{(h/a)} \times 100 = \frac{\alpha - \bar{Z}_{om}}{\alpha} \times 100 , \quad (A.114)$$

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_0 , equal to 0.1; the curves differ only in the choice of α ,

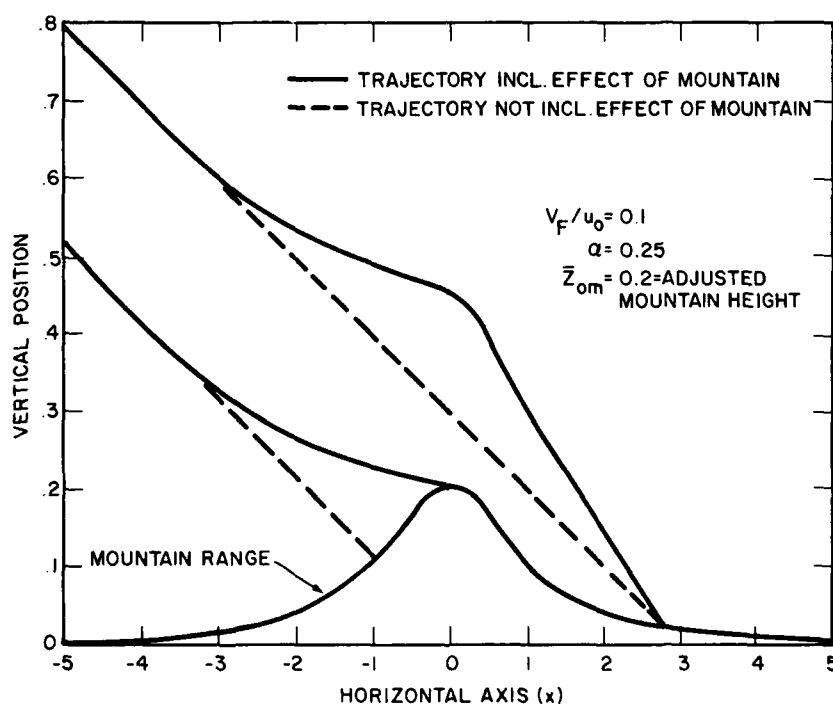


Figure A.7. Fallout Particle Trajectories (Note vertical scale is amplified 10 times; dimensions are in units of a , the half width of the mountain range.)

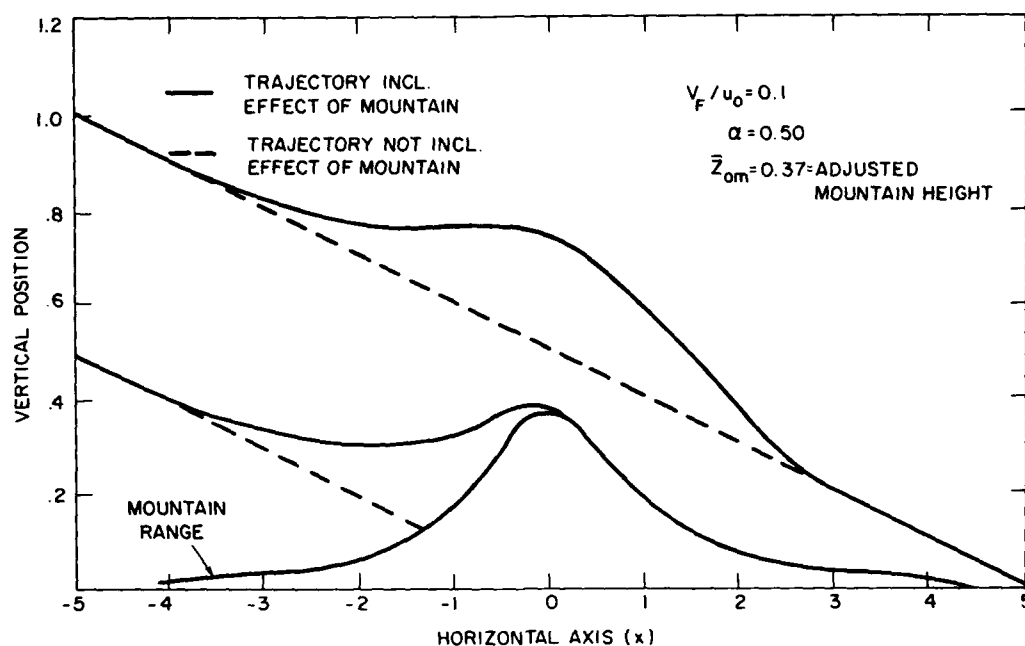


Figure A.8. Fallout Particle Trajectories (Note vertical scale is amplified 5 times; dimensions are in units of a , the half width of the mountain range.)

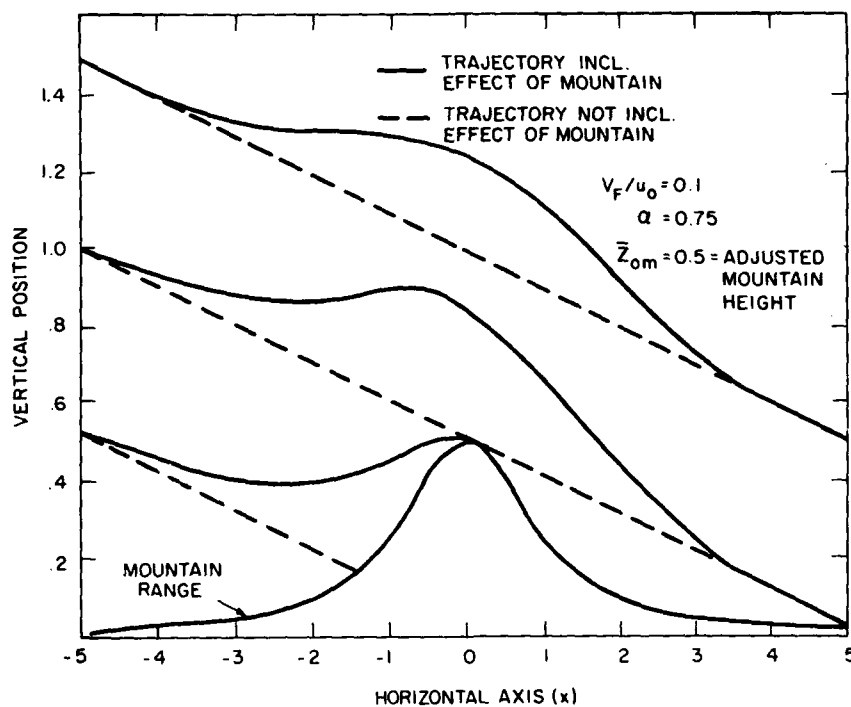


Figure A.9. Fallout Particle Trajectories (Note vertical scale amplified 5 times; dimensions are in units of a , the half width of the mountain range.)

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 $(V_F/u_0) = 0.1$ with $(\bar{z}_{om}/a) = 0.5$. From Figure A.2 we see that this would correspond to a 100- μ particle with $u_0 = 30$ mph, 150- μ particle with $u_0 = 50$ mph, or a 230- μ particle with $u_0 = 70$ 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

$$\left(k_x^2 + k_y^2\right)^{1/2} \gg \beta = 0.75 \times 10^{-6} \text{ cm}^{-1}, \quad (\text{A.115})$$

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

$$|k_x|u_o \gg \Omega, \quad (\text{A.116})$$

or equivalently

$$(2\pi) \frac{|u_o|}{\Omega} \gg L_x, \quad (\text{A.117})$$

where L_x is a representative wavelength in the x direction. The distance (u_o/Ω) equals

$$\frac{u_o}{\Omega} = d = (3.8 u_m) \text{ mi}, \quad (\text{A.118})$$

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

$$\frac{\gamma \beta k_x^2 u_o^2}{2\Omega |k_x| |k_y| u_o \omega_x} = \frac{\gamma \beta L_y u_o}{4\Omega L_x \cos \Theta \sin \epsilon} \quad , \quad (A.119)$$

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

$$0.5 \left(L_y / L_x \right) u_m > 1 \quad . \quad (A.120)$$

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)

$$\left(k_x^2 u_o^2 / \Omega^2 \right) \frac{\beta}{|k_x|} = r_1 \quad . \quad (A.121)$$

Using Eq. (A.118) we find that the requirement that the foregoing expression is much greater than unity is given by

$$2\pi (1.8) u_m^2 \gg L_x \text{ (miles)} . \quad (\text{A.122})$$

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)

$$\frac{k_x^2 u_o^2}{\Omega^2} \frac{\beta}{|k_y|} = r_1 (L_y/L_x) = \left[4\pi (1.8) u_m \right] (0.5 L_y/L_x) u_m . \quad (\text{A.123})$$

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 $\sigma = (\sigma/\beta) = 1.2 \times 10^9 \text{ cm}^2 \text{ sec}^{-2} \gg u_o^2$, and neglecting minor numerical factors, gives the following approximate expression for c :

$$c \approx k_x^2 u_o^2 \sigma \left(k_x^2 + k_y^2 \right) + \Omega \beta \sigma u_o k_x^2 + \Omega^2 \sigma k_x^2 . \quad (\text{A.124})$$

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. Erdelyi, 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. Feshbach, 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 subtleties. 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:

$$\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad , \quad (\text{B. 1})$$

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} - fv = -\frac{1}{\rho} \frac{\partial p}{\partial x} - \sigma u \quad , \quad (\text{B. 2})$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + w \frac{\partial v}{\partial z} + fu = -\sigma v \quad , \quad (\text{B. 3})$$

$$\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - g - \sigma w \quad , \quad (\text{B. 4})$$

$$p = \rho RT \quad . \quad (\text{B. 5})$$

and

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + w \frac{\partial T}{\partial z} = K \frac{\partial^2 T}{\partial z^2} \quad , \quad (\text{B. 6})$$

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:

$$\begin{aligned} w(z = 0) &= 0 \quad , \\ w(z \rightarrow \infty) &= 0 \quad , \\ T(z = 0) &= T_0 + T(x, t) \quad . \end{aligned} \tag{B. 7}$$

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_0 , 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 Schlichting 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 $90^\circ - 100^\circ\text{F}$ and a water temperature between $60^\circ - 70^\circ\text{F}$. 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

$$u = -\frac{\partial \psi}{\partial z}, \quad w = \frac{\partial \psi}{\partial x} \quad (\text{B. 8})$$

$$\eta = \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x} = \nabla^2 \psi \quad (\text{B. 9})$$

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 :

$$\frac{\partial \eta}{\partial t} + \left(u \frac{\partial}{\partial x} + w \frac{\partial}{\partial z} \right) \eta - f \frac{\partial v}{\partial z} = - \frac{1}{\rho} \left(\frac{\partial \rho}{\partial z} \frac{\partial p}{\partial x} - \frac{\partial \rho}{\partial x} \frac{\partial p}{\partial z} \right) - \sigma \eta \quad . \quad (\text{B. 10})$$

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 $p = \rho RT$.

$$S = - \frac{1}{\rho} \left(\frac{\partial \rho}{\partial z} \frac{\partial p}{\partial x} - \frac{\partial \rho}{\partial x} \frac{\partial p}{\partial z} \right) = \frac{R}{p} \left(\frac{\partial T}{\partial x} \frac{\partial p}{\partial z} - \frac{\partial T}{\partial z} \frac{\partial p}{\partial x} \right) \quad . \quad (\text{B. 11})$$

Since the first part of S is much larger than the second (see Ref. B. 6), we have

$$S = \frac{R}{p} \left(\frac{\partial T}{\partial x} \frac{\partial p}{\partial z} \right) = - \frac{g}{T} \left(\frac{\partial T}{\partial x} \right) \quad . \quad (\text{B. 12})$$

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,

$$\frac{\partial \tilde{\theta}}{\partial t} + u \frac{\partial \tilde{\theta}}{\partial x} + w \frac{\partial \tilde{\theta}}{\partial z} = \frac{\partial^2 \tilde{\theta}}{\partial z^2} \quad . \quad (\text{B. 13})$$

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

$$\tilde{\theta} = \theta_0(z) + \theta(x, z, t) , \quad (\text{B. 14})$$

where θ_0 and θ are the perturbed and unperturbed parts, respectively. We then arrive at a set of linearized equations:

$$-\frac{\partial}{\partial t} (\nabla^2 \psi) - f \frac{\partial v}{\partial z} = -\frac{g}{\theta_0} \frac{\partial \theta}{\partial x} + \sigma \nabla^2 \psi , \quad (\text{B. 15})$$

$$\frac{\partial v}{\partial t} - f \frac{\partial \psi}{\partial z} = -\sigma v , \quad (\text{B. 16})$$

and

$$\frac{\partial \theta}{\partial t} + \Gamma \frac{\partial \psi}{\partial x} = K \frac{\partial^2 \theta}{\partial z^2} , \quad (\text{B. 17})$$

where

$$\Gamma = \frac{\partial \theta_0}{\partial z} .$$

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:

$$D_u \approx \frac{\bar{u}}{L_x} + \frac{\bar{w}}{L_z} , \quad (\text{B. 18})$$

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 $\bar{u} = 2 \text{ m sec}^{-1}$ for every centigrade degree of temperature difference as opposed to a corresponding value of $\bar{w} = 2 \text{ cm sec}^{-1} \text{ } ^\circ\text{C}^{-1}$. If these results are used in Eq (B. 18) with $L_x = 20 \text{ km}$, $L_z = 500 \text{ m}$, and a maximum temperature differential of 5°C is assumed, we obtain the following value of D_u

$$D_u \approx \frac{10}{20 \times 10^3} + \frac{0.1}{500} = 7 \times 10^{-4} \text{ sec}^{-1} \quad (\text{B. 19})$$

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_0 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

$$\theta = A(z, t) \sin \lambda x \quad (B.20)$$

$$\psi = B(z, t) \cos \lambda x \quad \begin{cases} u = -(\partial\psi/\partial z) = -\cos \lambda x (\partial B/\partial z) \\ w = \partial\psi/\partial x = -\lambda \sin \lambda x B \end{cases} \quad (B.21)$$

$$v = C(z, t) \cos \lambda x \quad (B.22)$$

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

$$A(z, t) = \sum_{n=1}^{\infty} A_n(z) e^{in\Omega t} \quad (B.23)$$

$$B(z, t) = \sum_{n=1}^{\infty} B_n(z) e^{in\Omega t} \quad (B.24)$$

and

$$C(z, t) = \sum_{n=1}^{\infty} C_n(z) e^{in\Omega t} \quad (B.25)$$

Combining Eqs. (B.20) - (B.25), inserting them into Eqs. (B.15) - (B.17), and equating equal powers of $\exp(in\Omega t)$ gives the following coupled equations for A_n , B_n , and C_n :

$$(\sigma + in\Omega) \left(B_n'' - \lambda^2 B_n \right) + f C_n' = \alpha \lambda A_n, \quad (B.26)$$

and $(\sigma + in\Omega) C_n = f B_n', \quad (B.27)$

$$in\Omega A_n - \lambda \Gamma B_n = K A_n'', \quad (B.28)$$

where

$$\alpha = g/\theta_0.$$

For computational purposes it is more convenient to deal with functions $W_n(z)$ defined by the equation

$$W_n = -\lambda B_n, \quad (B.29)$$

in terms of which the velocity components are given by

$$w(x, z, t) = \sin \lambda x \sum_{n=1}^{\infty} W_n(z) e^{in\Omega t}, \quad (B.30)$$

$$u(x, z, t) = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} W_n'(z) e^{in\Omega t}, \quad (B.31)$$

and

$$v(x, z, t) = \sum_{n=1}^{\infty} (f/q_n) U_n(x, z) e^{in\Omega t}, \quad (B.32)$$

where

$$q_n = \sigma + in\Omega. \quad (B.33)$$

Eliminating Eq. (B. 27) and using (B. 29) gives

$$W_n'' = a_n W_n - b_n A_n, \quad (\text{B. 34})$$

and

$$A_n'' = c_n W_n + d_n A_n, \quad (\text{B. 35})$$

where

$$a_n = \frac{q_n^2 \lambda^2}{(q_n^2 + f^2)}, \quad c_n = \frac{\Gamma}{K}, \quad (\text{B. 36})$$

$$b_n = \frac{q_n \alpha \lambda^2}{(q_n^2 + f^2)}, \quad d_n = \frac{i n \Omega}{K}.$$

If we now let

$$A_n(z) = \hat{A}_n e^{\alpha_n z}, \quad (\text{B. 37a})$$

and

$$W_n(z) = \hat{W}_n e^{\alpha_n z}, \quad (\text{B. 37b})$$

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:

$$\begin{pmatrix} b_n & \mu_n - a_n \\ \mu_n - d_n & -c_n \end{pmatrix} \begin{pmatrix} \hat{A}_n \\ \hat{W}_n \end{pmatrix} = 0, \quad (\text{B. 38})$$

where

$$\mu_n = \alpha_n^2. \quad (\text{B. 39})$$

The only nontrivial solutions to Eq. (B.38) are those for which the determinant vanishes. This gives the two "allowed" values for μ_n :

$$\begin{aligned}\mu_{n1} &= \frac{(a_n + d_n) + (a_n + d_n)^2 - 4(a_n d_n + c_n b_n)^{1/2}}{2} = E_{n1} e^{i\gamma_{n1}}, \\ \mu_{n2} &= \frac{(a_n + d_n) - (a_n + d_n)^2 - 4(a_n d_n + c_n b_n)^{1/2}}{2} = E_{n2} e^{i\gamma_{n2}}.\end{aligned}\quad (\text{B.40})$$

The roots of the dispersion relationship correspond to four values of α_n which are given by

$$\alpha_n = \pm E_{n1}^{1/2} e^{(i\gamma_{n1}/2)} = \pm U_{n1} \left[\cos(\eta_{n1}) + i \sin(\eta_{n1}) \right], \quad (\text{B.41a})$$

and

$$\alpha_n = \pm E_{n2}^{1/2} e^{(i\gamma_{n2}/2)} = \pm U_{n2} \left[\cos(\eta_{n2}) + i \sin(\eta_{n2}) \right], \quad (\text{B.41b})$$

where

$$\eta_{n1} = \gamma_{n1}/2, \quad \eta_{n2} = \gamma_{n2}/2, \quad U_{n1} = E_{n1}^{1/2}, \quad U_{n2} = E_{n2}^{1/2}. \quad (\text{B.41c})$$

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

$$\alpha_{n1} = \epsilon_{n1} \mu_{n1}^{1/2} = \epsilon_{n1} U_{n1} \left[\cos(\eta_{n1}) + i \sin(\eta_{n1}) \right] = k_{n1} + i\ell_{n1}, \quad (\text{B.42a})$$

and

$$\alpha_{n2} = \epsilon_{n2} \mu_{n2}^{1/2} = \epsilon_{n2} U_{n2} \left[\cos(\eta_{n1}) + i \sin(\eta_{n1}) \right] = k_{n2} + i\ell_{n2}, \quad (\text{B.42b})$$

where

$$\begin{aligned}
 \epsilon_{n1} &= +1 \text{ if } \cos(\eta_{n1}) < 0, \\
 \epsilon_{n1} &= -1 \text{ if } \cos(\eta_{n1}) > 0, \\
 \epsilon_{n2} &= +1 \text{ if } \cos(\eta_{n2}) < 0, \\
 \epsilon_{n2} &= -1 \text{ if } \cos(\eta_{n2}) > 0.
 \end{aligned} \tag{B.43}$$

In terms of the α_n the solution to the problem is given by

$$\theta = \sin \lambda x \sum_{n=1}^{\infty} \left(\hat{A}_{n1} e^{\alpha_{n1} z} + \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t}, \tag{B.44}$$

$$w = \sin x \sum_{n=1}^{\infty} \left(r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t}, \tag{B.45}$$

$$u = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} \left(\alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t}, \tag{B.46}$$

$$v = \lambda^{-1} \cos \lambda z \sum_{n=1}^{\infty} g_n \left(\alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{in\Omega t}, \tag{B.47}$$

where

$$g_n = - \left(f/q_n \right) , \quad (B.48)$$

and

$$r_{n1} = - b_n / (\mu_{n1} - a_n), \quad r_{n2} = - b_n / (\mu_{n2} - a_n) . \quad (B.49)$$

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

$$\theta(x, z=0, t) = \sin \lambda x T(t) ,$$

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

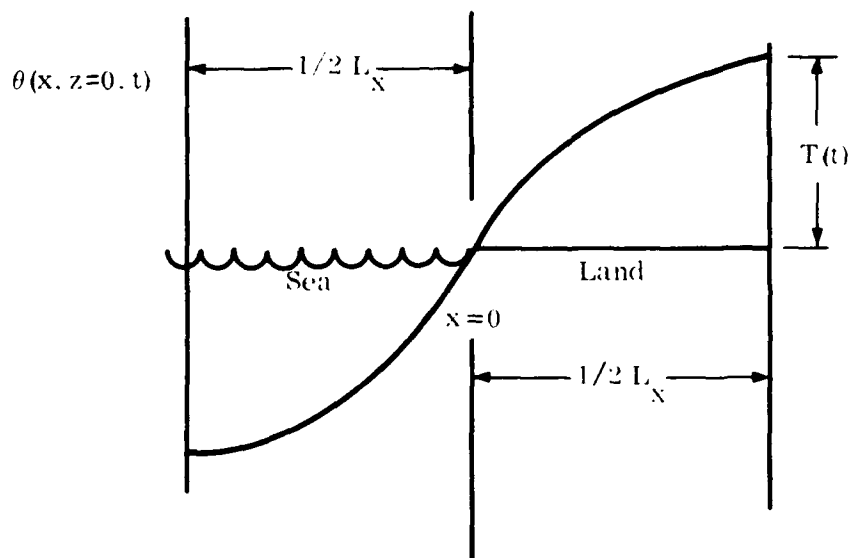


Figure B.1. Surface Temperature Variation

is higher than the temperature over the water. $T(t)$ is expressible as a Fourier series in multiples of the sidereal day frequency, Ω ,

$$T(t) = \sum_{n=1}^{\infty} T_n e^{in\Omega t}, \quad (B. 50)$$

where

$$T_n = \Omega (2\pi)^{-1} \int_0^{2\pi/\Omega} T(t) e^{-in\Omega t} dt = \Omega (2\pi)^{-1} \left[\int_0^{2\pi/\Omega} T(t) \cos(n\Omega t) dt + i \int_0^{2\pi/\Omega} T(t) \sin(n\Omega t) dt \right] \quad (B. 51)$$

$$T_n = T_n^* e^{i\tau_n};$$

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

$$T_n = \hat{A}_{n1} + \hat{A}_{n2}. \quad (B. 52)$$

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,

$$r_{n1} \hat{A}_{n1} + r_{n2} \hat{A}_{n2} = 0. \quad (B. 53)$$

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.

$$\hat{A}_{n1} = - \left[r_{n2} / (r_{n1} - r_{n2}) \right] T_n = S_{n1} e^{is_{n1}} T_n = S_{n1} T_n^* e^{i(s_{n1} + \tau_n)};$$

$$\hat{A}_{n2} = \left[r_{n1} / (r_{n1} - r_{n2}) \right] T_n = S_{n2} e^{is_{n2}} T_n = S_{n2} T_n^* e^{i(s_{n2} + \tau_n)};$$

$$r_{n1} = M_{n1} e^{im_{n1}} ;$$

$$r_{n2} = M_{n2} e^{im_{n2}} ;$$

$$g_n = G_n e^{i\nu_n} ;$$

where

$$G_n = -f/(\sigma^2 + (n\Omega)^2)^{1/2} ,$$

and

$$\nu_n = -\tan^{-1} (n\Omega/\sigma) .$$

We then have

$$\theta = \sum_{n=1}^{\infty} \theta_n , \tag{B. 54}$$

$$w = \sum_{n=1}^{\infty} w_n , \tag{B. 55}$$

$$u = \sum_{n=1}^{\infty} u_n , \tag{B. 56}$$

$$v = \sum_{n=1}^{\infty} v_n , \tag{B. 57}$$

where

$$\begin{aligned} \theta_n = \sin \lambda x T_n^* \left[S_{n1} e^{k_{n1} z} \cos (n\Omega t + \ell_{n1} z + s_{n1} + \tau_n) \right. \\ \left. + S_{n2} e^{k_{n2} z} \cos (n\Omega t + \ell_{n2} z + s_{n2} + \tau_n) \right], \end{aligned} \quad (\text{B. 58})$$

$$\begin{aligned} w_n = \sin \lambda x T_n^* \left[S_{n1} M_{n1} e^{k_{n1} z} \cos (n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \tau_n) \right. \\ \left. + S_{n2} M_{n2} e^{k_{n2} z} \cos (n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \tau_n) \right], \end{aligned} \quad (\text{B. 59})$$

$$\begin{aligned} u_n = \lambda^{-1} \cos \lambda x T_n^* \left[\epsilon_{n1} S_{n1} M_{n1} U_{n1} e^{k_{n1} z} \cos (n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \eta_{n1} + \tau_n) \right. \\ \left. + \epsilon_{n2} S_{n2} M_{n2} U_{n2} e^{k_{n2} z} \cos (n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \eta_{n2} + \tau_n) \right], \end{aligned} \quad (\text{B. 60})$$

$$\begin{aligned} v_n = \lambda^{-1} \cos \lambda x T_n^* G_n \left[\epsilon_{n1} S_{n1} M_{n1} U_{n1} e^{k_{n1} z} \cos (n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \eta_{n1} + \tau_n + \nu_n) \right. \\ \left. + \epsilon_{n2} S_{n2} M_{n2} U_{n2} e^{k_{n2} z} \cos (n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \eta_{n2} + \tau_n + \nu_n) \right]. \end{aligned} \quad (\text{B. 61})$$

In addition, the stream function $\psi = B(z, t) \cos \lambda x$ is given by

$$\begin{aligned} \psi = - \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} T_n^* \left[S_{n1} M_{n1} e^{k_{n1} z} \cos (n\Omega t + \ell_{n1} z + s_{n1} + m_{n1} + \tau_n) \right. \\ \left. + S_{n2} M_{n2} e^{k_{n2} z} \cos (n\Omega t + \ell_{n2} z + s_{n2} + m_{n2} + \tau_n) \right]. \end{aligned} \quad (\text{B. 62})$$

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 $\neq$ 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.

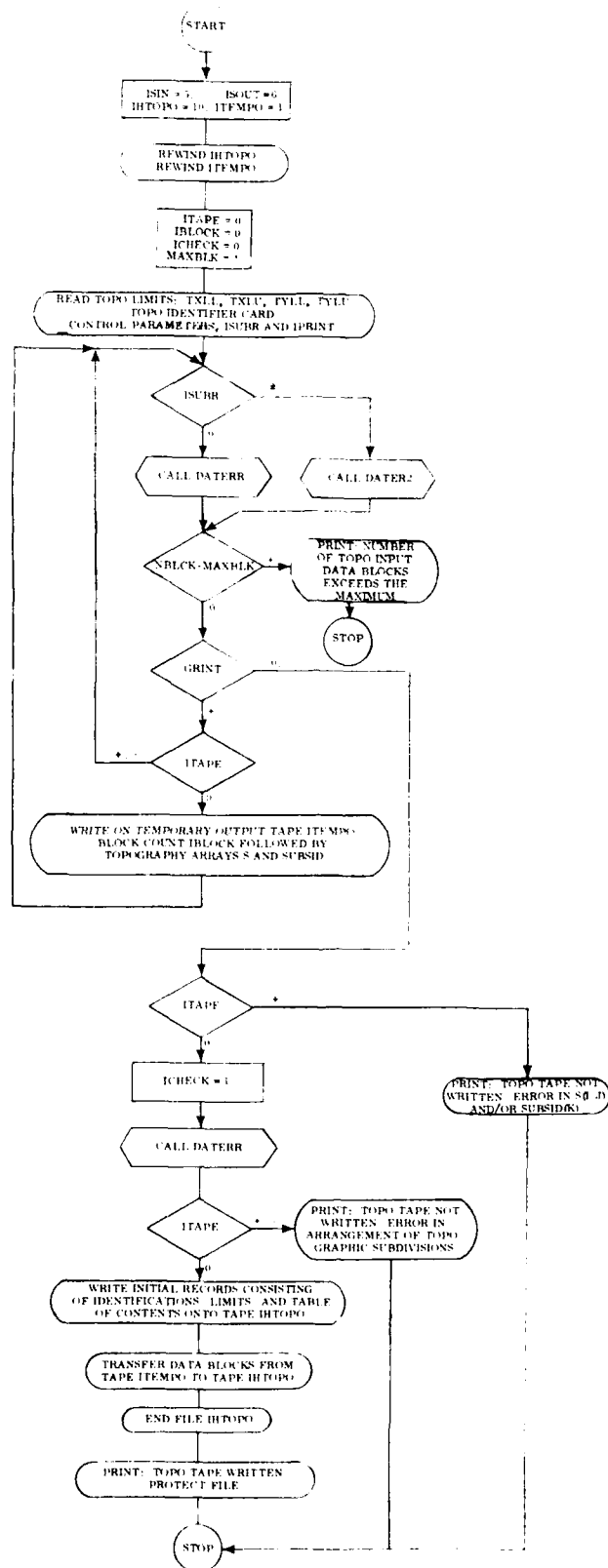
Next, after 40. the highest topo height is found and recorded in the topo table of contents along with lower coordinate limits, grid interval, and maximum array indices of the current data block.

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

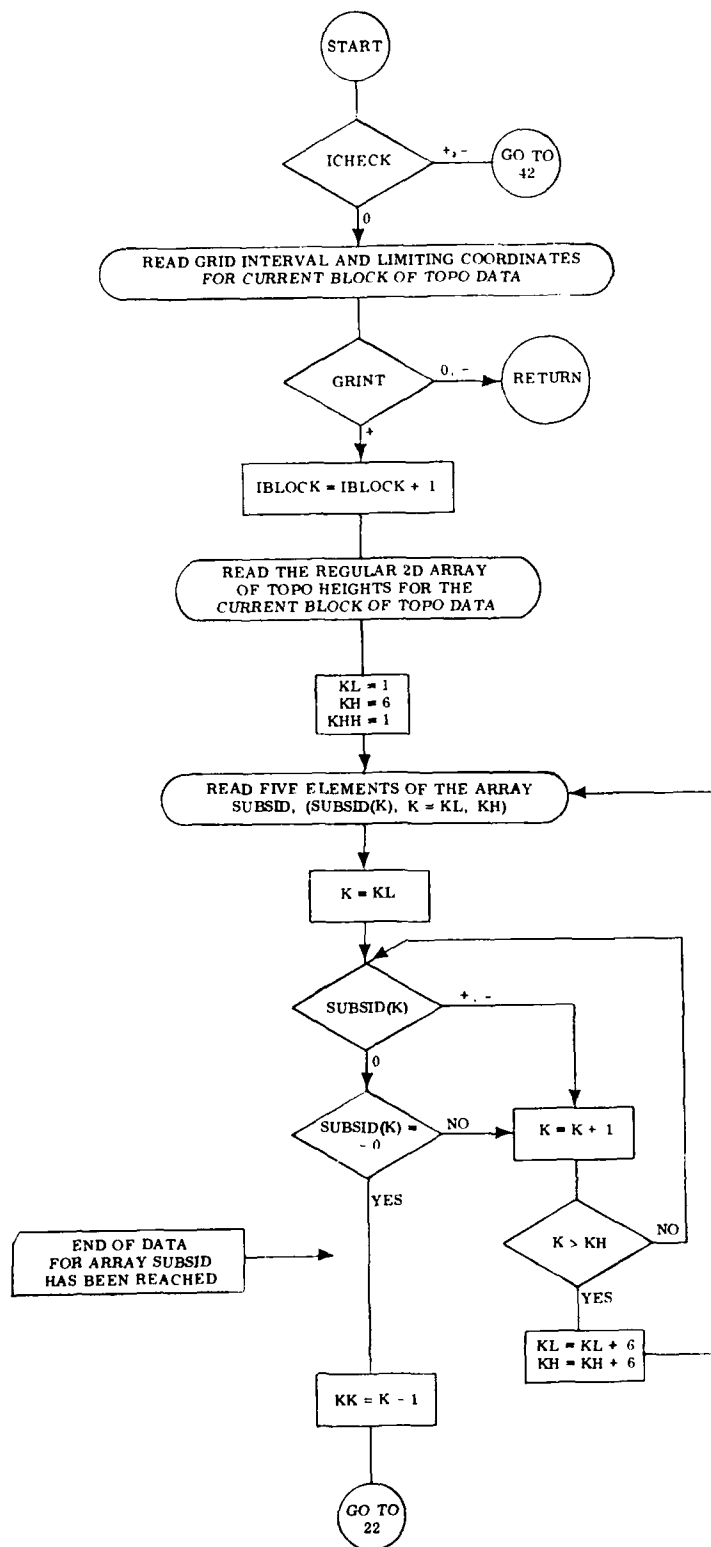
greater than the sum of the subareas covered by the individual data blocks, (4) to seek out any cases in which one subarea is totally included within another, and (5) to check that no gaps have been left between neighboring subareas. If any of these tests uncover an error, an explanatory comment is written 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.

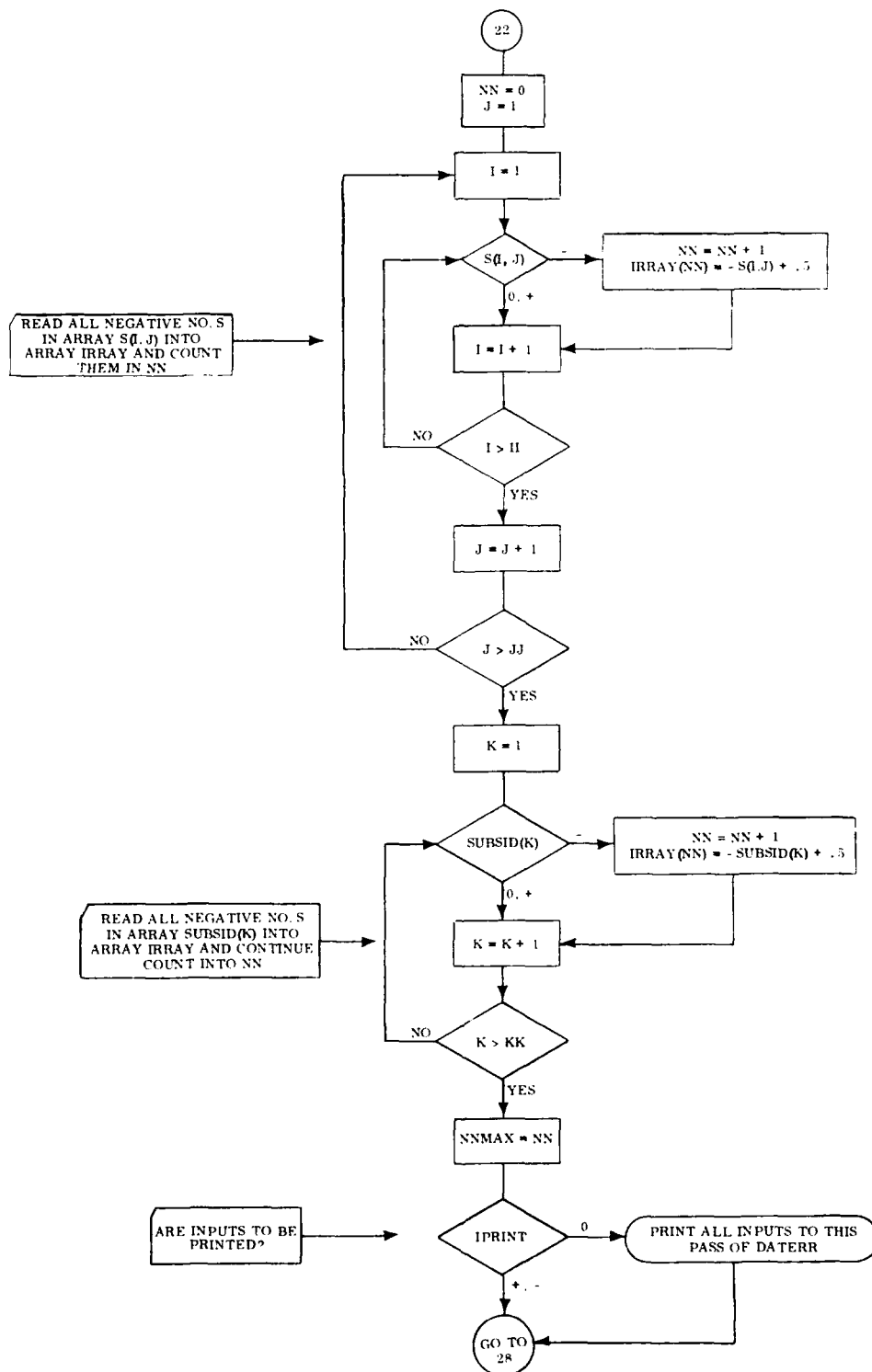


FC-C.1. Flow Chart of Main Program TOPIN



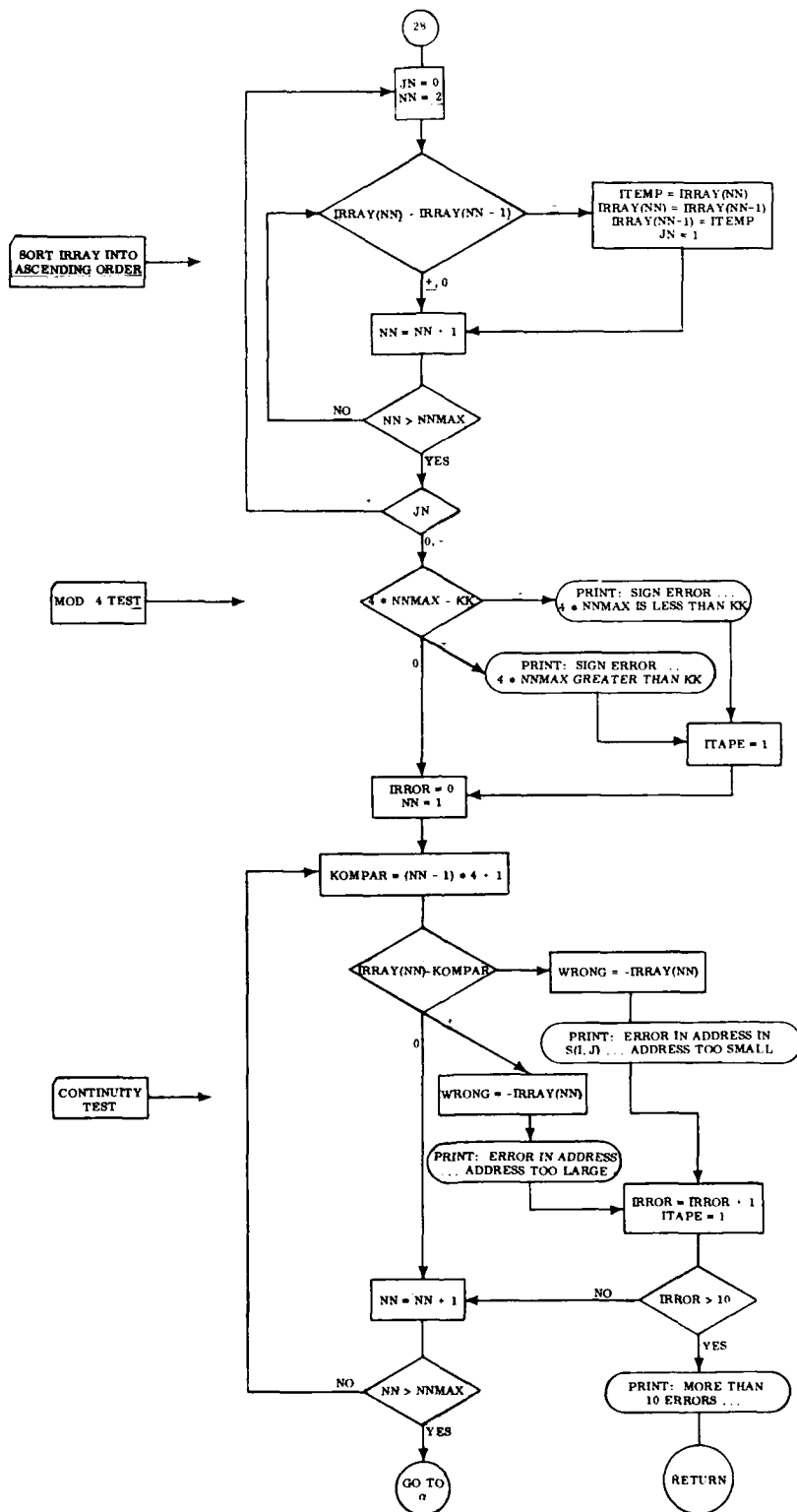
(a)

FC-C.2. Flow Charts of Subroutine DATERR



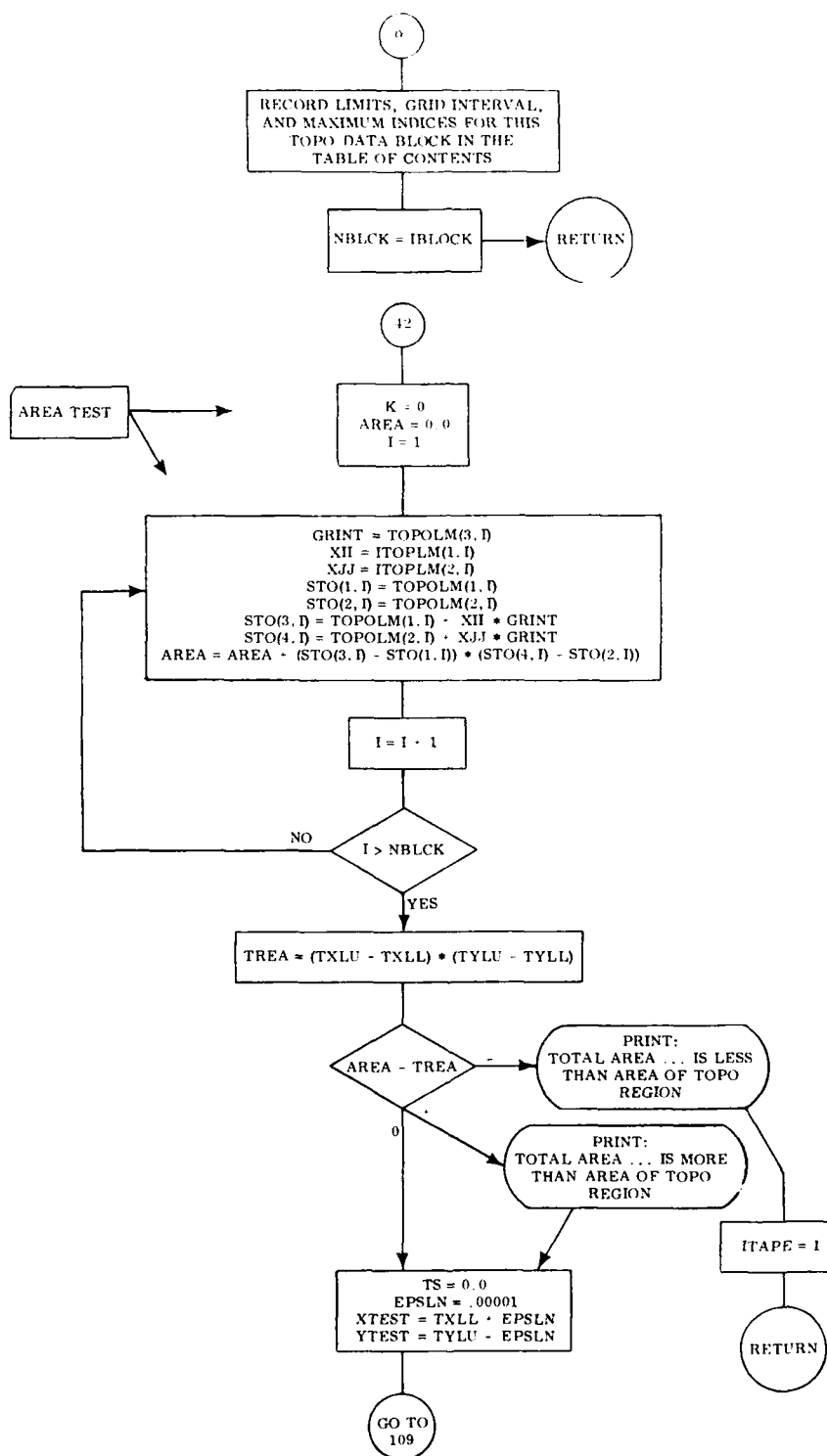
(b)

FC-C. 2. (Continued) Flow Charts of Subroutine DATERR



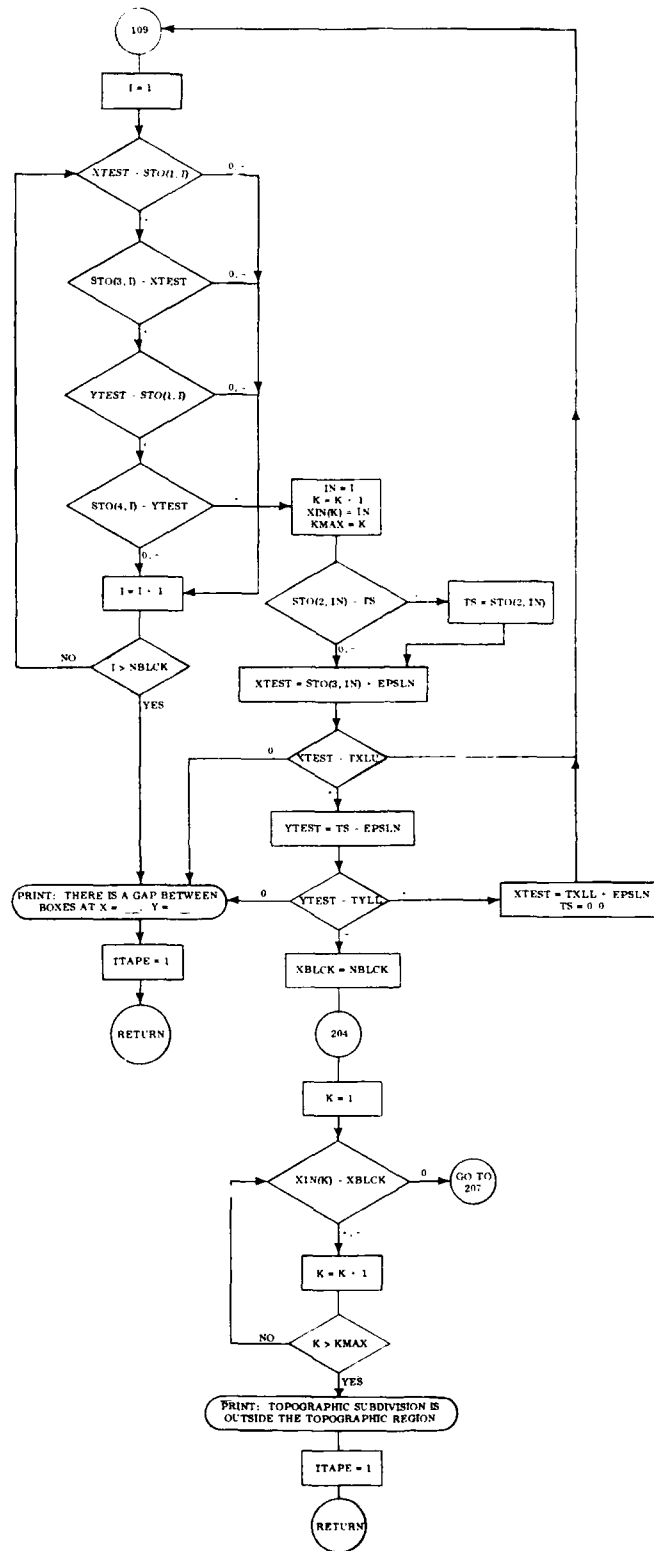
(c)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



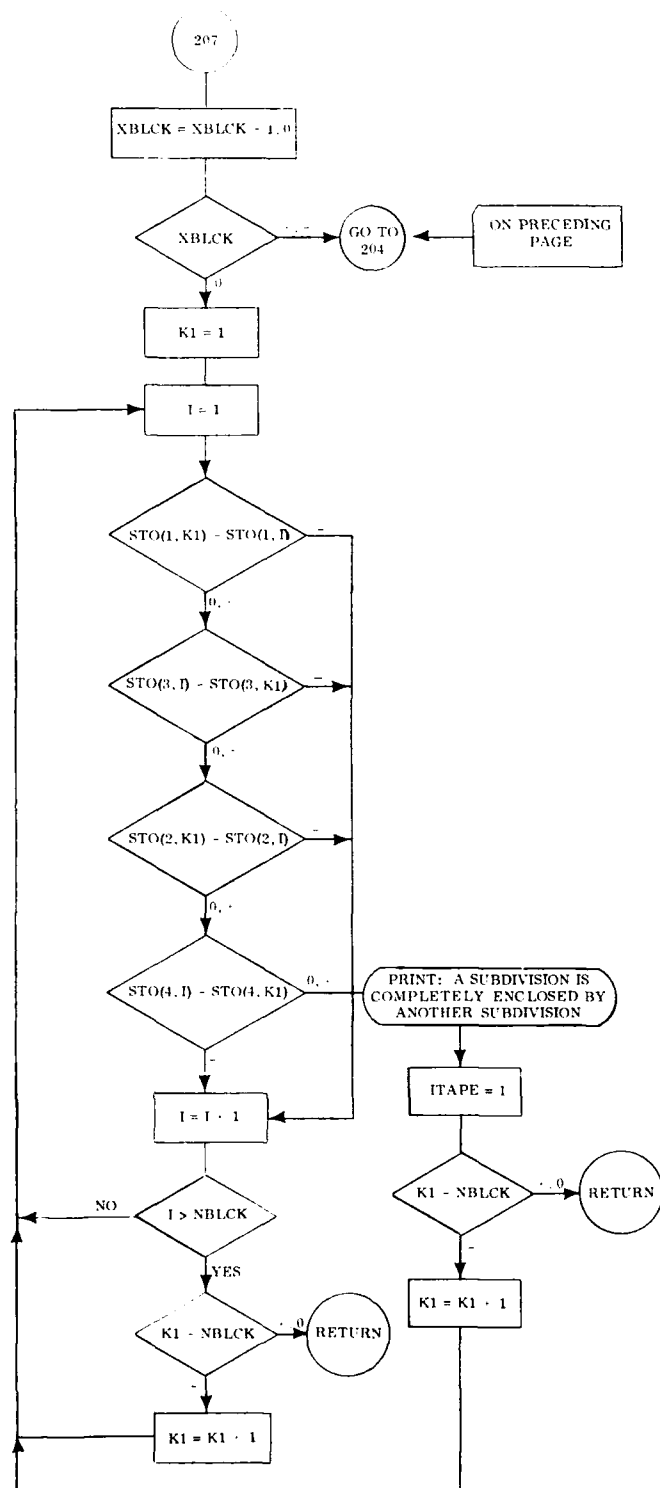
(d)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



(e)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR



(f)

FC-C.2. (Continued) Flow Charts of Subroutine DATERR

```

$IBFTC TOPINX LIST,DECK,M94/2
C TOPIN TOPI 1
C MAIN PROGRAM FOR CONSTRUCTING TOPOGRAPHY TAPE TOPI 2
C TOPI 3
C ***** GLOSSARY FOR TOPIN AND DATERR ***** TOPI 4
C TOPI 5
C BXLL LOWER X BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI 6
C IN CORE TOPI 7
C BXLU UPPER X BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI 8
C IN CORE TOPI 9
C BYLL LOWER Y BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI 10
C IN CORE TOPI 11
C BYLU UPPER Y BOUNDARY OF TOPOGRAPHIC DATA BLOCK CURRENTLY TOPI 12
C IN CORE TOPI 13
C GRINT LENGTH OF THE STANDARD GRID INTERVAL SEE S(I,J) TOPI 14
C IHTOPO FINAL PREPARED TOPOGRAPHY TAPE NUMBER TOPI 15
C II UPPER LIMIT OF I DIMENSION OF S(I,J) ARRAY TOPI 16
C IPRINT INDICATES WHETHER INPUT (TOPO ARRAYS, BOUNDARIES, TOPI 17
C LIMITS ETC. FOR EACH BLOCK) IS TO BE PRINTED. TOPI 18
C 0 CAUSES PRINTING, 1 CAUSES NONE. TOPI 19
C ISIN SYSTEM INPUT TAPE NUMBER TOPI 20
C ISOUT SYSTEM OUTPUT TAPE NUMBER TOPI 21
C ISUBR INDICATES IF = 0 THAT SUBROUTINE DATERR IS TO BE USED TOPI 22
C ITAPE NOT ZERO INDICATES THAT ERRORS HAVE BEEN FOUND IN TOPI 23
C THE DATA SET AND THAT A VALID TOPO TAPE CANNOT BE TOPI 24
C WRITTEN. A ZERO INDICATES NO ERRORS ARE APPARENT TOPI 25
C ITEMPO TEMPORARY TOPO TAPE NUMBER TOPI 26
C JJ UPPER LIMIT OF J DIMENSION OF S(I,J) ARRAY TOPI 27
C S(I,J) TWO DIMENSIONAL ARRAY CONTAINING EITHER TOPOGRAPHIC TOPI 28
C HEIGHT OF GRID SQUARE I,J ON THE MAP OR A NEGATIVE TOPI 29
C NUMBER WHICH, WHEN CONVERTED TO A POSITIVE INTEGER, TOPI 30
C IS THE INDEX, K, TO A NUMBER IN THE ONE-DIMENSIONAL TOPI 31
C ARRAY, SUBSID(K). IF A NUMBER IN S(I,J) IS NEGATIVE, TOPI 32
C IT IS CALLED THE BASE ADDRESS TO THE SQUARE OF SIDE TOPI 33
C GRINT WHOSE LOWER-LEFT CORNER IS AT LOCATION TOPI 34
C (I-1)*GRINT, (J-1)*GRINT ON THE MAP TOPI 35
C SUBSID(K) ONE-DIMENSIONAL ARRAY CONTAINING EITHER A TOPOGRAPHIC TOPI 36
C HEIGHT OR A NEGATIVE NUMBER WHICH, WHEN CONVERTED TO TOPI 37
C A POSITIVE INTEGER, IS THE INDEX, K, TO ANOTHER TOPI 38
C NUMBER IN SUBSID(K), WHICH MAY IN TURN BE EITHER TOPI 39
C A TOPOGRAPHIC HEIGHT OR ANOTHER BASE ADDRESS TOPI 40
C TXLL LOWER X BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA TOPI 41
C TXLU UPPER X BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA TOPI 42
C TYLL LOWER Y BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA TOPI 43
C TYLU UPPER Y BOUNDARY OF THE COMPLETE TOPOGRAPHY AREA TOPI 44
C TOPI 45
C ***** TOPI 46
C TOPI 47
C DIMENSION S(3,30),SUBSID(10000),TOPOLM(4,100),IHTOPLM(3,100) TOPI 48
C 1,TOPIID(12) TOPI 49
C TOPI 50
C ***** TOPI 51
C TOPI 52
C COMMON GRINT, BXLL, BXLU, BYLL, BYLU, II TOPI 53
C COMMON JJ, KK, S, SUBSID, ITAPE, ISIN TOPI 54
C COMMON ISOUT, IBLOCK, IPRINT, ITOPLM, IHTOPLM, TXLL TOPI 55
C COMMON TXLU, TYLL, TYLU, NBLCK, ICHECK TOPI 56
C TOPI 57
C ***** TOPI 58
C TOPI 59
C 1 FORMAT (15H1IHTOPO TXLL=,F6.1,8H TXLU=,F6.1,8H TYLL=,F6.1, TOPI 60

```

18H TYLU=F6.1)	TOP1 81
2 FORMAT (12A6)	TOP1 82
3 FORMAT (12)	TOP1 83
4 FORMAT (7H ISUBR=,12)	TOP1 84
5 FORMAT (6H IPRINT=,12)	TOP1 85
6 FORMAT (6H ITOPO,4F6.1)	TOP1 86
7 FORMAT (68H ITOPO TAPE NOT WRITTEN. ERROR IN S(1,5) AND/OR	TOP1 87
ISUBSID(K).)	TOP1 88
8 FORMAT (33H ITOPO TAPE WRITTEN. PROJECT FILE.)	TOP1 89
9 FORMAT (4E13.6)	TOP1 90
22 FORMAT(120H ITOPO TAPE NOT WRITTEN. ERROR IN ARRANGEMENT OF ITOPO	TOP1 91
IAPIC SUBDIVISIONS.)	TOP1 92
33 FORMAT(120H THE NUMBER OF INPUT ITOPO DATA BLOCKS EXCEEDS THE MAXIMUM	TOP1 93
10H NUMBER PERMITTED.)	TOP1 94
C	TOP1 95
C *****	TOP1 96
C	TOP1 97
DATA DENI1/6H IHTOPU/	TOP1 98
C	TOP1 99
C *****	TOP1 100
C *****	TOP1 101
C	TOP1 102
ISOUT=6	TOP1 103
ISIN=5	TOP1 104
IHTOPU=4	TOP1 105
ITEMPO=1	TOP1 106
REWIND IHTOPU	TOP1 107
REWIND ITEMPO	TOP1 108
ITAPE=1	TOP1 109
IBLOCK=0	TOP1 110
ICHECK=1	TOP1 111
MAXBLK=100	TOP1 112
READ (ISIN,9) IALL,IALS,IFLL,IFLS	TOP1 113
READ (ISIN,2) (IOPID(J),J=1,12)	TOP1 114
READ (ISIN,3) ISUBR	TOP1 115
READ (ISIN,3) IPRINT	TOP1 116
81 IF (ISUBR) 91,10,91	TOP1 117
91 WRITE (ISOUT,6)	TOP1 118
C	
C	
INSERTION POINT FOR ALTERNATE DATA CHECKING PROGRAM DATERR	
GO TO 16	
10 CALL DATERR	
IF(NBLCK-MAXBLK) 30,30,31	
30 IF(GRINT) 13,13,11	
31 WRITE (ISOUT,33)	
GO TO 16	
11 IF(ITAPE) 10,12,10	
12 WRITE (ITEMPO) IBLOCK	
WRITE (ITEMPO) ((S(I,J),I=1,11),J=1,55)	
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)	

Reproduced from
best available copy.

```

      GO TO 16
20  WRITE (IHTOPU) DENTI
      WRITE (IHTOPU) IXLL,IXLU,ITYLL,ITYLU,NBLCK
      WRITE (IHTOPU,2)
      WRITE (IHTOPU)ITOPLM
      WRITE (IHTOPU)ITOPLM
      REWIND ITEMPO
18  READ (ITEMPO)IBLOCK
      IM=ITOPLM(1,IBLOCK)
      JM=ITOPLM(2,IBLOCK)
      KM=ITOPLM(3,IBLOCK)
      READ (ITEMPO)((S(1,J),I=1,IM),J=1,JM)
      READ (ITEMPO)(SUBSID(K),K=1,KM)
      WRITE (IHTOPU)((S(1,J),I=1,IM),J=1,JM)
      WRITE (IHTOPU)(SUBSID(K),K=1,KM)
      IF (IBLOCK-NBLCK)18,17,17
19  END FILE IHTOPU
      WRITE (ISOUT,8)
16  STOP
      END

```

```

TOP1 119
TOP1 120
TOP1 121
TOP1 122
TOP1 123
TOP1 124
TOP1 125
TOP1 126
TOP1 127
TOP1 128
TOP1 129
TOP1 130
TOP1 131
TOP1 132
TOP1 133
TOP1 134
TOP1 135
TOP1 136
TOP1 137
TOP1 138

```

139*

139 *

IBFTC DATER	LIST,DECK,M94/2	DATE	
C		DATE	1
	SUBROUTINE DATERR	DATE	2
C		DATE	3
C	SUBROUTINE TESTING FOR LOOPS AND INCORRECT ADDRESSES IN S(1,J)	DATE	4
C	AND SUBSID(K)	DATE	5
C		DATE	6
C	*****	DATE	7
C		DATE	8
C	SEE PROGRAM TOPIN FOR A GLOSSARY	DATE	9
C		DATE	10
C	*****	DATE	11
C		DATE	12
C	DIMENSION S(3,30),SUBSID(1000),IRRA(100),ARRA(100),WIN(100),	DATE	13
C	1TOPLEN(4,100),1TOFL(3,100),SIS(4,100)	DATE	14
C		DATE	15
C	*****	DATE	16
C		DATE	17
C	COMMON GRINI, BALE, BALO, BYLL, BYLO, 11	DATE	18
C	COMMON JU, KK, S, SUBSID, 1TAPE, 10TH	DATE	19
C	COMMON 10001, 1BLOCK, 1PRINT, 1TOPLEN, 1TOFLN, 1ALL	DATE	20
C	COMMON 1ALE, 1TLE, 1TLO, 1NLEN, 1CHECK	DATE	21
C		DATE	22
C	*****	DATE	23
C		DATE	24
C	1 FORMAT (5E15.6)	DATE	25
C	2 FORMAT (2I12)	DATE	26
C	3 FORMAT (6F11.1)	DATE	27
C	4 FORMAT (10HUBLOCK NO.,17)	DATE	28
C	5 FORMAT (9H GRINI=F0.1,9H BALE=F0.1,9H BALO=F0.1,	DATE	29
C	10H BYLL=F0.1,9H BYLO=F0.1)	DATE	30
C	6 FORMAT (6H II=F11.2,6H JJ=F11.2)	DATE	31
C	7 FORMAT (25H (S(1,J),I=1,11),J=1,30)	DATE	32
C	8 FORMAT (6H KK=F11.2)	DATE	33
C	9 FORMAT (19H SUBSID(K),K=1,KK)	DATE	34
C	10 FORMAT (9H MINMAX=F11.2)	DATE	35
C	11 FORMAT (82H SIGN ERROR IN ADDRESS OR TOPOGRAPHIC HEIGHT IN S(1,J)	DATE	36
C	1AND/OR SUBSID(K), BLOCK NO.,17,20H .14MINMAX IS LESS THAN KK.)	DATE	37
C	12 FORMAT (82H SIGN ERROR IN ADDRESS OR TOPOGRAPHIC HEIGHT IN S(1,J)	DATE	38
C	1AND/OR SUBSID(K), BLOCK NO.,17,31H .14MINMAX IS GREATER THAN KK.)	DATE	39
C	13 FORMAT (51H ERROR IN ADDRESS IN S(1,J) OR SUBSID(K), BLOCK NO.,	DATE	40
C	117,19H . WRONG ADDRESS IS,F11.1,24H . ADDRESS IS TOO SMALL.)	DATE	41
C	14 FORMAT (51H ERROR IN ADDRESS IN S(1,J) OR SUBSID(K), BLOCK NO.,	DATE	42
C	117,19H . WRONG ADDRESS IS,F11.1,24H . ADDRESS IS TOO LARGE.)	DATE	43
C	15 FORMAT (81H ADDRESSES NOT IN INCREMENTS OF 4. MORE THAN 10 ERRORS.	DATE	44
C	1 ERROR SEARCH IN BLOCK NO.,17,14H DISCONTINUED.)	DATE	45
C	220 FORMAT(120H TOTAL AREA OF TOPOGRAPHIC SUBDIVISIONS IS LESS THAN ARDATE	DATE	46
C	1EA OF TOPOGRAPHIC REGION.)	DATE	47
C	221 FORMAT(120H TOTAL AREA OF TOPOGRAPHIC SUBDIVISIONS IS GREATER THANDATE	DATE	48
C	1 AREA OF TOPOGRAPHIC REGION. CHECK FOR OVERLAPS.)	DATE	49
C	222 FORMAT(39H THERE IS A GAP BETWEEN BOXES AT XTEST=F15.9,7H YTEST=F1DATE	DATE	50
C	15.9)	DATE	51
C	229 FORMAT(120H A TOPOGRAPHIC SUBDIVISION IS OUTSIDE THE TOPOGRAPHIC RDATE	DATE	52
C	1EGION.)	DATE	53
C	216 FORMAT(120H A SUBDIVISION IS COMPLETELY ENCLOSED BY ANOTHER SUBDIVDATE	DATE	54
C	1ISION.)	DATE	55
C		DATE	56
C	*****	DATE	57
C		DATE	58
C	DATA QUUCT/01000000000000/	DATE	59
C		DATE	60

Reproduced from
best available copy.

```

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=0
      DO 21 KHH=1,11000
      READ (ISIN,3)(SUBSD(K),K=KL,KH)
      DO 20 K=KL,KH
      IF(SUBSD(K))23,18,20
18  A=OR(SUBSD(K),.000001)
      IF(A)19,20,20
19  KK=K-1
      GO TO 22
20  CONTINUE
      KL=KL+0
      KH=KH+0
21  CONTINUE

C *****
C *****
C *****
      MOD 4 CHECK
C *****
22  NN=1
      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(SUBSD(K))25,26,26
25  NN=NN+1
      IRRAY(NN)=-SUBSD(K)+0.5
26  CONTINUE
      NNMAX=NN
      IF(IPRINT)25,27,26
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)(SUBSD(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
      DATE 61
      DATE 62
      DATE 63
      DATE 64
      DATE 65
      DATE 66
      DATE 67
      DATE 68
      DATE 69
      DATE 70
      DATE 71
      DATE 72
      DATE 73
      DATE 74
      DATE 75
      DATE 76
      DATE 77
      DATE 78
      DATE 79
      DATE 80
      DATE 81
      DATE 82
      DATE 83
      DATE 84
      DATE 85
      DATE 86
      DATE 87
      DATE 88
      DATE 89
      DATE 90
      DATE 91
      DATE 92
      DATE 93
      DATE 94
      DATE 95
      DATE 96
      DATE 97
      DATE 98
      DATE 99
      DATE 100
      DATE 101
      DATE 102
      DATE 103
      DATE 104
      DATE 105
      DATE 106
      DATE 107
      DATE 108
      DATE 109
      DATE 110
      DATE 111
      DATE 112
      DATE 113
      DATE 114
      DATE 115
      DATE 116
      DATE 117
      DATE 118

```

www

$$\frac{1}{2} = \frac{1}{2}$$

24. 1=1, 11

$$\frac{1}{2} = \frac{1}{2} + \frac{1}{2}$$
$$A \otimes B \otimes C = \sum_{i,j} A_{ij} \otimes B_{ij} \otimes C_{ij}$$

24. CONTINUE

$$U \cup V \quad \angle \rightarrow \angle \quad \angle = i, \angle \angle$$
$$f(z) = f_1(z) + i f_2(z)$$
$$\Delta \chi(\Delta T) \approx \Delta \chi_{\text{eff}}(\Delta T)$$

242 CONTINUED

$$\therefore \Delta X = 0.0001$$

248 $J' =$

00 245 1120, MAX

```
IF (ARRAY (0,0) - ARRAY (0,0 - 1)) 2+0,2+0,2+0
```

246 ATEMP=APR-AY(M%)

$$ARRAY(0) = ARRAY(0) - 1$$
$$A \leq AY \left(\frac{1}{\lambda} - \epsilon \right) = A \left(\frac{1}{\lambda} - \epsilon \right)$$
$$J^*A = 1$$

245 CONTINUE

$$12(2.1) = 4 + 7 + 4 + 1 + 4 + 0$$

44 / TOPLEVEL(4,1)BLOCK=BASE

POPUL (Z, BLOCK) = C1LL

$$\text{FORMULA (3, BLOCK)} = \text{exit}$$
$$POPUL(4, IBLCK) = AR \times (1 - 0.75 \times X)$$

1709011, 1500K

$$ITUPLE(2, I_BLOCK) = JJ$$

```
ITOPLEN(3,IBLOCK)=KK
```

```

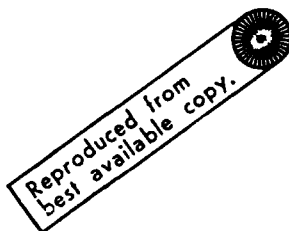
NBLCK=IBLOCK

```

GO TO 41

()

DATE 117
DATE 120
DATE 121
DATE 122
DATE 123
DATE 124
DATE 125
DATE 126
DATE 127
DATE 128
DATE 129
DATE 130
DATE 131
DATE 132
DATE 133
DATE 134
DATE 135
DATE 136
DATE 137
DATE 138
DATE 139
DATE 140
DATE 141
DATE 142
DATE 143
DATE 144
DATE 145
DATE 146
DATE 147
DATE 148
DATE 149
DATE 150
DATE 151
DATE 152
DATE 153
DATE 154
DATE 155
DATE 156
DATE 157
DATE 158
DATE 159
DATE 160
DATE 161
DATE 162
DATE 163
DATE 164
DATE 165
DATE 166
DATE 167
DATE 168
DATE 169
DATE 170
DATE 171
DATE 172
DATE 173
DATE 174
DATE 175



COMPARISON OF TOTAL AREA OF TOPO REGION WITH THE SUM OF THE AREAS	DATE 177
OF THE TOPO SUBDIVISIONS	DATE 178
	DATE 179
	DATE 180
42 K=0	DATE 181
AREA=0.0	DATE 182
DO 100 I=1,NBLCK	DATE 183
GRINT=TOPOLM(3,I)	DATE 184
XII=1TOPLM(1,I)	DATE 185
XJJ=1TOPLM(2,I)	DATE 186
STO(1,I)=TOPOLM(1,I)	DATE 187
STO(2,I)=TOPOLM(2,I)	DATE 188
STO(3,I)=TOPOLM(1,I)+XII*GRINT	DATE 189
STO(4,I)=TOPOLM(2,I)+XJJ*GRINT	DATE 190
AREA=AREA+(STO(3,I)-STO(1,I))*(STO(4,I)-STO(2,I))	DATE 191
100 CONTINUE	DATE 192
TREA=(TXLU-TALL)*(TYLU-TYLL)	DATE 193
IF(ABS(AREA-TREA)200,200,201)	DATE 194
200 WRITE (ISOUT,220)	DATE 195
ITAPE=1	DATE 196
GO TO 41	DATE 197
201 WRITE (ISOUT,221)	DATE 198
	DATE 199
	DATE 200
C CHECK FOR THE ERRONEOUS ENCLOSURE OF ONE TOPO SUBDIVISION BY	DATE 201
C ANOTHER	DATE 202
	DATE 203
	DATE 204
208 K1=1	DATE 205
217 DO 210 I=1,NBLCK	DATE 206
IF(I-K1)219,210,219	DATE 207
219 IF(STO(1,K1)-STO(1,I))210,211,211	DATE 208
211 IF(STO(3,K1)-STO(3,I))210,212,212	DATE 209
212 IF(STO(2,K1)-STO(2,I))210,213,213	DATE 210
213 IF(STO(4,K1)-STO(4,I))210,214,214	DATE 211
210 CONTINUE	DATE 212
IF(K1-NBLCK)215,101,101	DATE 213
215 K1=K1+1	DATE 214
GO TO 217	DATE 215
214 WRITE (ISOUT,216)	DATE 216
ITAPE=1	DATE 217
GO TO 41	DATE 218
	DATE 219
	DATE 220
C CHECK FOR GAPS BETWEEN TOPO SUBDIVISIONS	DATE 221
	DATE 222
	DATE 223
101 TS=0.0	DATE 224
EPSLN=1.0E-5	DATE 225
XTEST=TXLU+EPSLN	DATE 226
YTEST=TYLU-EPSLN	DATE 227
109 DO 105 I=1,NBLCK	DATE 228
IF(XTEST-STO(1,I))105,105,102	DATE 229
102 IF(STO(3,I)-XTEST)105,105,103	DATE 230
103 IF(YTEST-STO(2,I))105,105,104	DATE 231
104 IF(STO(4,I)-YTEST)105,105,106	DATE 232
105 CONTINUE	DATE 233
GO TO 202	DATE 234

```

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-TXLS)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 TOPS SUBDIVISIONS HAVE BEEN EXAMINED
C
C
203 XBLOCK=NBLOCK
204 DO 205 K=1,KMAX
    IF(XIN(K)-XBLOCK)205,207,204
205 CONTINUE
    WRITE (ISOUT,229)
    ITAPE=1
    GO TO 41
207 XBLOCK=XBLOCK-1.0
    IF(XBLOCK)204,41,204
41 RETURN
END

```

```

DATE 235
DATE 236
DATE 237
DATE 238
DATE 239
DATE 240
DATE 241
DATE 242
DATE 243
DATE 244
DATE 245
DATE 246
DATE 247
DATE 248
DATE 249
DATE 250
DATE 251
DATE 252
DATE 253
DATE 254
DATE 255
DATE 256
DATE 257
DATE 258
DATE 259
DATE 260
DATE 261
DATE 262
DATE 263
DATE 264
DATE 265
DATE 266

```

267*

267 *

