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
DISCLAIMER NOTICE

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

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.
DEPARTMENT OF DEFENSE
LAND FALLOUT PREDICTION SYSTEM

Volume IV - Atmospheric Transport

TO-B-66-46
2 February 1967

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
ACKNOWLEDGMENTS

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

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

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

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>MKWIND</td>
<td>65</td>
</tr>
<tr>
<td>LINK7</td>
<td>78</td>
</tr>
<tr>
<td>GETWND</td>
<td>82</td>
</tr>
<tr>
<td>LOTRAN</td>
<td>95</td>
</tr>
<tr>
<td>MTWND1</td>
<td>97</td>
</tr>
<tr>
<td>RGWND1</td>
<td>100</td>
</tr>
<tr>
<td>CBREZ1</td>
<td>105</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>115</td>
</tr>
</tbody>
</table>

**USER INFORMATION** ........................................ 124

**Input Description** ........................................ 124

**General** .................................................. 124

**MKWIND Data** .............................................. 127

**Local Circulation System Data** .......................... 130

**Topography Data** .......................................... 131

**Output Description** ....................................... 136

**Printed Output** ........................................... 136

**Binary Output** ............................................ 136

**Data Structures for Secondary Memory Tapes** ........ 138

**REFERENCES** ............................................... 140

**FORTRAN LISTINGS** ......................................... 141

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>FALRAT</td>
<td>143</td>
</tr>
<tr>
<td>DUMPP</td>
<td>145</td>
</tr>
<tr>
<td>RDTOPO</td>
<td>153</td>
</tr>
<tr>
<td>RDCIRS</td>
<td>155</td>
</tr>
<tr>
<td>LINK5</td>
<td>155</td>
</tr>
<tr>
<td>LINK6</td>
<td>166</td>
</tr>
<tr>
<td>RDCIRS</td>
<td>167</td>
</tr>
<tr>
<td>MKWIND</td>
<td>169</td>
</tr>
<tr>
<td>LINK7</td>
<td>178</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Cont'd.)

<table>
<thead>
<tr>
<th>Subroutine</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GETWND</td>
<td>189</td>
</tr>
<tr>
<td>LOTRAN</td>
<td>192</td>
</tr>
<tr>
<td>MTWND1</td>
<td>197</td>
</tr>
<tr>
<td>RGWND1</td>
<td>201</td>
</tr>
<tr>
<td>CBREZ1</td>
<td>205</td>
</tr>
<tr>
<td>HEIGHT</td>
<td>211</td>
</tr>
<tr>
<td>SAMPLE TEST PROBLEM AND PRINTOUT</td>
<td>215</td>
</tr>
<tr>
<td>Transport Module</td>
<td>217</td>
</tr>
<tr>
<td>APPENDIX A. THEORY OF OROGRAPHIC FLOW WITH APPLICATION TO TROPOSPHERIC FALLOUT</td>
<td>227</td>
</tr>
<tr>
<td>Introduction</td>
<td>227</td>
</tr>
<tr>
<td>Geometric Considerations</td>
<td>229</td>
</tr>
<tr>
<td>Theory of Airflow</td>
<td>229</td>
</tr>
<tr>
<td>Application of the Theory to Specific Geometries</td>
<td>245</td>
</tr>
<tr>
<td>Mountain Ridge Not Perpendicular to Unperturbed Flow</td>
<td>245</td>
</tr>
<tr>
<td>A Single Mountain</td>
<td>249</td>
</tr>
<tr>
<td>Wind Streamlines and Fallout Particle Streamlines in Two Dimensions</td>
<td>251</td>
</tr>
<tr>
<td>Conditions for Neglecting the Coriolis Effect</td>
<td>261</td>
</tr>
<tr>
<td>Conclusions</td>
<td>263</td>
</tr>
<tr>
<td>References</td>
<td>264</td>
</tr>
</tbody>
</table>

APPENDIX B. THE INCORPORATION OF THE SEA BREEZE IN THE CALCULATION OF FALLOUT | 265 |

| Introduction | 265 |
| Review of the Sea-Breeze Theories | 265 |
| Sea-Breeze Model | 268 |
| Wind-Field Parameters | 268 |
| Boundary Conditions | 279 |
| References | 283 |
# TABLE OF CONTENTS (Cont'd.)

APPENDIX C. TOPOGRAPHIC DATA INPUT PROGRAMS

<table>
<thead>
<tr>
<th>TOPIN AND DATERR</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>285</td>
</tr>
<tr>
<td>Description of Card Inputs</td>
<td>285</td>
</tr>
<tr>
<td>A Recommended Procedure for the Encoding of Topographic Data</td>
<td>287</td>
</tr>
<tr>
<td>Operation of Programs TOPIN and DATERR</td>
<td>289</td>
</tr>
<tr>
<td><strong>TOPIN</strong></td>
<td>289</td>
</tr>
<tr>
<td><strong>DATERR</strong></td>
<td>290</td>
</tr>
<tr>
<td>Flow Charts and Program Listings</td>
<td>291</td>
</tr>
<tr>
<td><strong>TOPIN</strong></td>
<td>299</td>
</tr>
<tr>
<td><strong>DATERR</strong></td>
<td>302</td>
</tr>
</tbody>
</table>
# List of Tables

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

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

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A. 4</td>
<td>Wind Streamlines for $\alpha = 0.50$</td>
<td>255</td>
</tr>
<tr>
<td>A. 5</td>
<td>Wind Streamlines for $\alpha = 0.75$</td>
<td>255</td>
</tr>
<tr>
<td>A. 6</td>
<td>Differences Between Surface Trajectory and Mountain Ridge</td>
<td>257</td>
</tr>
<tr>
<td>A. 7</td>
<td>Fallout Particle Trajectories</td>
<td>258</td>
</tr>
<tr>
<td>A. 8</td>
<td>Fallout Particle Trajectories</td>
<td>259</td>
</tr>
<tr>
<td>A. 9</td>
<td>Fallout Particle Trajectories</td>
<td>259</td>
</tr>
<tr>
<td>B. 1</td>
<td>Surface Temperature Variation</td>
<td>279</td>
</tr>
</tbody>
</table>
# LIST OF FLOW CHARTS

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

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

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

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

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

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

\[ \frac{dV_p}{dt} = - \left[ V_p(t) - V_w(\mathbf{r}(t), t) \right] \cdot \phi \left( |V_p - V_w| \right) + G \]  

(1)

and the displacement equation

\[ \frac{dr}{dt} = V_p, \]  

(2)

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

\[ F = - (V_p - V_w) \cdot \phi \left( |V_p - V_w| \right). \]  

(3)

* A commonly used expression for \( \phi \) in the pressure flow regime is

\[ \phi = \frac{1}{2} \frac{C_D}{m} \rho A |V_p - V_w| = K |V_p - V_w|, \]

while in the Stokes law regime \( \phi \) 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 Davies\(^2\) for spheres are most appropriate for use in the DOD Land Fallout Prediction System. The following procedure is used in computing particle settling rates:

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

\[
C_D R^2 = \frac{4 \rho p d^3}{3 \eta^2}, \quad (7)
\]

where \( G \) is the acceleration of gravity, \( \rho \) and \( p_p \) are the densities of air and particle, \( d \) is the particle diameter, and \( \eta \) is the dynamic viscosity of the air.
2. The Reynolds number is evaluated from the Davies polynomials:

\[ R = \frac{C_D R^2}{24} - 2.3363 \times 10^{-4} \left( C_D R^2 \right)^2 + 2.0154 \times 10^{-6} \left( C_D R^2 \right)^3 \]

\[- 6.9105 \times 10^{-9} \left( C_D R^2 \right)^4, \quad C_D R^2 < 140 \]

or

\[ \log_{10} R = -1.29536 + 0.986 \left( \log_{10} C_D R^2 \right) - 0.0467 \left( \log_{10} C_D R^2 \right)^2 \]

\[ + 0.0011235 \left( \log_{10} C_D R^2 \right)^3, \quad 100 < C_D R^2 < 4.5 \times 10^7 \]

3. The settling velocity \( V_F \) is computed from

\[ V_F = \frac{R \eta}{\rho d} \]

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

\[ f = 1 + \frac{2.33 \times 10^{-4}}{\rho} \]
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 $\Delta Z$ and lateral dimensions $2X_0$, $2Y_0$ (see Figure 3) is assumed to move 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)).

![Figure 3. Segment of Cloud Volume](imageURL)
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

\[
\sigma_0 = \frac{N}{4Y_0 X_0} \quad \text{(particles - m}^{-2}\text{).} \tag{12}
\]

During transport, turbulent diffusion tends to disperse the particles of the cloud subdivision so that by the time the subdivision reaches the ground, its shape will have changed and its particle density, \( \sigma \), 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

\[
P(x, y, t) = \sigma_0 \int_{-X_0}^{+X_0} \int_{-Y_0}^{+Y_0} G(x-x', y-y', t) \, dx' \, dy' , \tag{13}
\]

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

\[
G = (2\pi D t)^{-1} \exp \left\{ -\left[ (x-x')^2 + (y-y')^2 \right]/2Dt \right\} , \tag{14}
\]

with \( D \) being the diffusion constant. Consideration of Eqs. (13) and (14) show that \( P(x, y, t) \) is defined over the entire \( x, y \) plane, but as an approximation to the theoretical result for computational purposes we have chosen to construct an equivalent rectangular segment of uniform surface density \( \sigma \) 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 $x^2$, $y^2$ 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, $x^2$ and $y^2$ are related to the limiting dimensions via the formulas:

$$x^2 = \frac{1}{3} X^2; \quad y^2 = \frac{1}{3} Y^2.$$  \hspace{1cm} (15)

On the other hand, we have

$$\bar{x}^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^2 P(x, y, t) \, dx \, dy = \frac{4\sigma X Y}{4\sigma X Y} = Dt + \frac{1}{3} X^2,$$

and

$$\bar{y}^2 = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} y^2 P(x, y, t) \, dx \, dy = \frac{4\sigma X Y}{4\sigma X Y} = Dt + \frac{1}{3} Y^2.$$  \hspace{1cm} (16)

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

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

and

$$Y^2 = Y_o^2 + 3Dt,$$  \hspace{1cm} (17)

with corresponding lateral density

$$\sigma = \frac{N}{4XY}.$$  \hspace{1cm} (18)

The user is referred to Pasquill's "Atmospheric Diffusion" for a discussion on reasonable estimates of $D$.  \hspace{1cm} (3)
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:

\[ R_i = \text{position of } i\text{th observed wind velocity vector relative to the wind-field-array grid point } R_0 \]

\[ V_i = \text{measured wind velocity at position } R_i \]

\[ V_0 = \text{wind velocity at a wind-field-array grid point } R_0 \]. \( V_0 \) is to be determined from \( R_i \) and \( V_i \).

**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 \( V_0 \) is computed as a weighted average of the velocities from observations that lie within distance \( \beta \) from the grid point in the horizontal plane and distance \( \alpha \) from the grid point in the vertical direction. Specifically, the relationship between \( V_0 \) and \( V_i \) is given by

\[ V_0 = \sum_{i=1}^{N} f_i \cdot V_i \]

where

\[ \sum_{i=1}^{N} f_i = 1 \]

A weighting method described by Cressman\(^4\) has been used in deriving an expression for \( f_i \) in the form

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

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

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

\[ u = u_o + (\nabla u)_o \cdot R, \]
\[ v = v_o + (\nabla v)_o \cdot R, \]
\[ w = w_o + (\nabla w)_o \cdot R, \]

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 \( u_o, v_o, w_o, (\nabla u)_o = A_x, (\nabla v)_o = B_y, \) and \( (\nabla w)_o = C_z. \)

Actually, the computation breaks down into three separate parts involving \( (u_o, A) \), \( (v_o, B) \), and \( (w_o, C) \). To illustrate the procedure, we shall outline the method for computing \( u_o. \) If \( U_i \) denotes the x component of wind velocity at the ith sounding station, the ith residual is given by

\[ \xi_i = U_i - u_i = U_i - (u_o + A_x x_i + A_y y_i + A_z z_i). \]

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

\[ F(u_o, A) = \sum_{i=1}^{N} \xi_i^2. \]
with respect to these four parameters. The four linear equations so deduced are

\[
\frac{\partial F}{\partial u_0} = 0 = - \sum U_1 + \sum (u_o + A \cdot R_1),
\]

(25)

\[
\frac{\partial F}{\partial A_x} = 0 = - \sum U_i x_i + \sum (u_o + A \cdot R_1) x_i,
\]

(26)

\[
\frac{\partial F}{\partial A_y} = 0 = - \sum U_i y_i + \sum (u_o + A \cdot R_1) y_i,
\]

(27)

and

\[
\frac{\partial F}{\partial A_z} = 0 = - \sum U_i z_i + \sum (u_o + A \cdot R_1) z_i.
\]

(28)

Introducing the averaged quantities,

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

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

\[
\bar{u} z = \left( \frac{1}{N} \right) \sum U_i z_i, \quad \bar{x}^2 = \left( \frac{1}{N} \right) \sum x_i^2, \quad \bar{x} y = \left( \frac{1}{N} \right) \sum x_i y_i,
\]

(29)

\[
\bar{x} z = \left( \frac{1}{N} \right) \sum x_i z_i, \quad \bar{y} z = \left( \frac{1}{N} \right) \sum y_i z_i, \quad \bar{y}^2 = \left( \frac{1}{N} \right) \sum y_i^2,
\]

and

\[
\bar{z}^2 = \left( \frac{1}{N} \right) \sum z_i^2.
\]
gives the following matrix equation for $u_0$ and $A$:

\[
\begin{pmatrix}
1 & x & y & z \\
-x & x^2 & xy & xz \\
y & xy & y^2 & yz \\
z & xz & yz & z^2 \\
\end{pmatrix}
\begin{pmatrix}
u_0 \\
ul \\
u_x \\
u_y \\
u_z \\
\end{pmatrix}
=
\begin{pmatrix}
u \\
u_x \\
u_y \\
u_z \\
\end{pmatrix}.
\]

By use of conventional matrix inversion techniques, Eq. (30) can be solved for $u_0$. We have

\[
u_0 = \gamma_1 \ddot{u} + \gamma_2 \dddot{u} + \gamma_3 \ddot{u} + \gamma_4 \dddot{u},
\]

where

\[
\gamma_1 = \frac{B_{11}}{|B|},
\]

\[
\gamma_2 = \frac{B_{21}}{|B|},
\]

\[
\gamma_3 = \frac{B_{31}}{|B|},
\]

and

\[
\gamma_4 = \frac{B_{41}}{|B|},
\]
in which \(|B|\) denotes the determinant of the matrix Eq. (30). The quantities \(B_{ki}^{ki}\) 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

\[
v_o = \gamma_1 \bar{v} + \gamma_2 \bar{vx} + \gamma_3 \bar{vy} + \gamma_4 \bar{vz},
\]

and

\[
w_o = \gamma_1 \bar{w} + \gamma_2 \bar{wx} + \gamma_3 \bar{wy} + \gamma_4 \bar{wz},
\]

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.
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

\[
z = f(x, y) = \frac{ha^3}{(a^2 + r^2)^{3/2}}
\]

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

\[
r = \left( x^2 + y^2 \right)^{1/2}
\]

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

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

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

and

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

where

\[
\lambda = (z + a).
\]
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 crest-line makes an arbitrary angle \( \gamma \) with respect to the direction of the unperturbed flow \( u_o \). The pertinent geometric details are shown in Figure 7.

![Figure 7. Mountain Ridge Not Perpendicular to Flow](image)
The mathematical description of the elevation of the mountain ridge when viewed along the $y'$ axis is given by the expression

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

(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

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

$$y = x' \sin \gamma + y' \cos \gamma, \quad y' = -x \sin \gamma + y \cos \gamma.$$  \hspace{1cm} (43)

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

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

(44)

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

(45)

and

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

(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

\[
\begin{align*}
    u_i(x, y, z) &= u(x - x_i, y - y_i, z), \\
    v_i(x, y, z) &= v(x - x_i, y - y_i, z), \\
    w_i(x, y, z) &= w(x - x_i, y - y_i, z),
\end{align*}
\]

(47)

and

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

\[
z \geq z_i^* = f_i(x - x_i, y - y_i)
\]

(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

$$z^*_T = \sum_i f_i(x - x_i, y - y_i)$$  \hspace{1cm} (49)$$

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

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

$$v(x, y, z) = \sum_i v_i(x - x_i, y - y_i, z)$$  \hspace{1cm} (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 $\ell_1$, as shown in Figure 8(a).

![Figure 8. Mountain Ridge Separations](image)

If $\ell_1$ is large compared to $a$, then with good approximation the topography can be represented by the equation

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

(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

$$z_T^*(x = x_2) = \frac{h}{1 + \left(\frac{\ell_1^2}{a^2}\right)} + h,$$
which if \( \theta_1 \gg a \) approximately equals \( h \), the contribution from the second ridge only. Now consider the same ridges, but this time separated by a smaller distance \( \theta_2 \) (Figure 8(b)). Equation (51) will no longer be adequate because \( z_T^*(x = x_2) \) becomes

\[
z_T^*(x = x_2) = \frac{h}{1 + \left( \frac{\theta_2^2}{a^2} \right)} + h,
\]

which can be significantly greater than \( h \) if \( \theta_2 \) 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

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

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

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

and then solving for the $A_i(k)$. The solution for the $A_i(k)$ 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, \( \Delta v \), as compared to \( u_0 \) diminishes. As shown in the analysis, \( \Delta y \) 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 \( \left| S \right|^2 \), 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, \( a \).

\[ S = \left( \frac{h}{a} \right) \]
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

\[ |h'| = |h|(S'/S), \tag{53} \]

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

![Figure 9. Slope Compensation](image-url)
conjecture have not been rendered. The modification of elevation does not alleviate the shortcomings of first-order perturbation theory; consequently, we further suggest and, moreover, stipulate in the program itself that $S' \leq 0.6$.

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

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

Figure 10 shows the typical circulation pattern after sunrise when viewed along the direction of the coastline (positive $y$ axis). In addition to the $x$-$z$ circulation there is an accompanying $y$ component of velocity which is related to the other components in a determined way, but is not shown in the figure. The driving force is
of course the potential temperature differential at the surface, whose behavior with $x$ and $t$ is assumed to be given by

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

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

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

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 $\tau_n$. 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: $\sigma$, the Guldberg-Mohn friction parameter; $K$, the thermal eddy diffusivity; $\bar{\theta}_0$, the average ground temperature; $\Gamma = (d\bar{\theta}_0/dz)$, the initial unperturbed temperature gradient; and $\sin \phi$, where $\phi$ 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 $n$th 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:

$$w_n = \sin \lambda x J_{n_z} \left[ e^{k_{n_1} z} \cos (n \Omega t + \ell n_1 z + \phi_n) - e^{k_{n_2} z} \cos (n \Omega t + \ell n_2 z + \phi_n) \right] \quad (60)$$

$$u_n = \cos \lambda x J_{n_x} \left[ \bar{K}_{n_1} e^{k_{n_1} z} \cos (n \Omega t + \ell n_1 z + \phi_n + \eta_{n_1}) - \bar{K}_{n_2} e^{k_{n_2} z} \cos (n \Omega t + \ell n_2 z + \phi_n + \eta_{n_2}) \right] \quad (61)$$

and

$$v_n = \cos \lambda x J_{n_y} \left[ \bar{K}_{n_1} e^{k_{n_1} z} \cos (n \Omega t + \ell n_1 z + \phi_n + \eta_{n_1} + \nu_n) - \bar{K}_{n_2} e^{k_{n_2} z} \cos (n \Omega t + \ell n_2 z + \phi_n + \eta_{n_2} + \nu_n) \right] \quad (62)$$

The constants $J_{n_z}$, $J_{n_x}$, and $J_{n_y}$ are each proportional to $T_n^*$, the magnitude of the $n$th 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 $k_{n_1}$, $\ell_{n_1}$, $k_{n_2}$, $\ell_{n_2}$, $\bar{K}_{n_1}$, $\bar{K}_{n_2}$, $\eta_{n_1}$, $\eta_{n_2}$, and $\nu_n$ are completely independent of $T_n^*$ or $\tau_n$, while $\phi_n = \alpha_n + \tau_n$ where $\alpha_n$ is mode-dependent but otherwise independent of $T_n^*$ or $\tau_n$. 

31
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.

$\theta_0$, the average ground temperature, can be determined by standard techniques.

Although $\sigma$, 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 $\sigma$ 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, it was shown that holding all other parameters fixed and increasing $\sigma$ from 0 to $2.5 \times 10^{-4}$ sec$^{-1}$ shortened the time lag between maximum
temperature differential and the maximum wind velocity from 6.7 to 1.4 hr. Concurrently, for the same temperature differential, a factor of 3 decrease in wind velocity occurred. The value of $\sigma$ 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 $\sigma$. There are to our knowledge no known experimental methods which yield $\sigma$; however, suggested values are given in Table 3 (p. 110).

$\Gamma$, 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 $\tau_n$, the amplitude and phases of the temperature harmonics, are input quantities calculated from the following formulas. Defining certain quantities $\delta_n$ and $\Delta_n$ by the equations

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

and

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

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

\[
\tau_n = \tan^{-1} \left( \frac{\Delta_n}{\delta_n} \right) \quad . \quad (66)
\]

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

$$V(x,y,z,t) = V_c(x,y,z,t) \exp \left\{-ka \left[|x| - (L_x/2)\right]\right\}$$,
where $\vec{v}_c$ 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](image)
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 \(\Delta t\) is determined by the user. The point-slope method of integration, including restriction on values of \(\Delta t\), 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

<table>
<thead>
<tr>
<th>Name</th>
<th>Called By</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>LINK5</td>
<td>Executive Program M3*</td>
<td>Transport initialization and control.</td>
</tr>
<tr>
<td>RDTOPO</td>
<td>LINK5 and LINK7</td>
<td>Reads a block of topographic data into core memory.</td>
</tr>
<tr>
<td>LINK6</td>
<td>Executive Program M3*</td>
<td>Calls subroutine MKWIND</td>
</tr>
<tr>
<td>DUMPP</td>
<td>LINK5 and LINK7</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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. |

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 \( 4/3 \times g \times \text{ROPART} \times 10^{-8} \) 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 \( C_D R^2 \) and several intermediate parameters. Next \( C_D R^2 \) 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 \( C_D R^2 > 7 \times 10^7 \). 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.
START

\( I = \text{ALT} \times 200.0 + 6.5 \)

\( V_0 = \text{PSIZE} / \text{ATEMP} \)

\( V_1 = \text{PSIZE} \times V_0 \times \text{FROG} \)

\( \text{CDRR} = V_1 \times \text{RHO} \times V_9 \)

\( \text{CDRR} > 146.0 \)

NO

100

YES

150

\( \text{CDRR} > 4.5 \times 10^7 \)

NO

YES

PRINT: DAVIES EQUATIONS ARE INACCURATE FOR PSIZE MICRONS AT ALT METERS.

200

\( FV = V_1 \times (41666.7 \times \text{CDRR} \times (-233.63 \times \text{CDRR} \times (2.0154 - 0.0069105 \times \text{CDRR}))) \)

\( QLOGA = \log_{10} (\text{CDRR}) - 20.773 \)

\( FV = 50657.0 \times V_1 \times \text{CDRR} \times (QLOGA \times QLOGA - 443.98) \times 0.0011235 \)

300

\( FV = FV(1.0 + 0.233 \times \text{PSIZE} \times \text{RHO}) \)

RETURN

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 \( J \)th particle, of three spatial coordinates \( X_P(J), Y_P(J), \) and \( Z_P(J) \); a time coordinate \( T_P(J) \); a particle size \( P_S(J) \); and a mass per unit area \( F_M(J) \). 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 \( F_M(J) \) and the sign and magnitude of the time coordinate \( T_P(J) \) are used to record the class of the description as indicated in Table 2.
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 \(N_{\text{FREE}}\), 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 \(J_{\text{TEST}} = 0\) 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 \(J_{\text{TEST}} = 1\) and \(J_{\text{TEST}} = N_{G}\) 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 \(J_{\text{TEST1}}\) (see Table 2) and \(J_{\text{TEST}}\) 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 \(N_{\text{BMAX}}\) will be written by DUMPP.

### Table 2

<table>
<thead>
<tr>
<th>Class</th>
<th>FMAS(J)</th>
<th>TP(J)</th>
<th>JTEST1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blank</td>
<td>0</td>
<td>Not Used</td>
<td>1</td>
</tr>
<tr>
<td>Grounded particles</td>
<td>-FMAS(J)</td>
<td>-TP(J)</td>
<td>1</td>
</tr>
<tr>
<td>Lost particles. These are particles that have gone beyond the complete wind field or topographic description</td>
<td>-FMAS(J)</td>
<td>TLIMIT</td>
<td>2</td>
</tr>
<tr>
<td>Topography boundary particles. These are particles at the limit of the in-core topography</td>
<td>+FMAS(J)</td>
<td>-TP(J)</td>
<td>3</td>
</tr>
<tr>
<td>Time boundary particles. These are particles at the time limit for the in-core wind field</td>
<td>+FMAS(J)</td>
<td>ENDTIM</td>
<td>4</td>
</tr>
<tr>
<td>Wind-field boundary particles. These are particles at the spatial limit of the in-core wind field</td>
<td>-FMAS(J)</td>
<td>+TP(J)</td>
<td>5</td>
</tr>
</tbody>
</table>
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) ≠ 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
MUST ANY PARTICLE DESCRIPTIONS BE DUMPED?  
N > NFREE  
NO  
JTEST = 0  
GO TO 152  
YES  
151  
WOULD DUMPING THE GROUNDED PARTICLES MAKE SUFFICIENT ROOM FOR THE BLOCK OF N INCOMING PARTICLES?  
(NFREE + NG) > N  
NO  
FIND THE IDENTITY (JTEST 1) AND SIZE (JTEST) OF THE LARGEST CLASS  
YES  
PREPARE TO DUMP GROUNDED PARTICLE DESCRIPTIONS  
JTEST 1 = 1  
JTEST = NG  
NO  
IS SIZE OF SELECTED CLASS ≤ 07  
JTEST ≤ 07  
YES  
IRROR = 184  
7734  
CALL ERROR  
RETURN  
NO  
IS SIZE OF SELECTED CLASS > NBMAX?  
JTEST > NBMAX  
YES  
JTEST = NBMAX  
NO  
GO TO 181  
(a)  
FC-3. Detailed Flow Charts for Subroutine DUMPP
FC-3. (Continued) Detailed Flow Charts for Subroutine DUMPP
FC-3. (Continued) Detailed Flow Charts for Subroutine DUMPP

54
FC-3. (Continued) Detailed Flow Chart for Subroutine DUMPP
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 L1NK5 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 L1NK5 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 L1NK5. It is assigned its true value by the wind description program MKWIND which is called by L1NK6, subsequent to L1NK5, when L1NK5 has set IEXEC = 1 at statement number 400.)

When transport has been completed, L1NK5 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 L1NK8 of the Output Processor Module (see DASA-1800-VI.)
IS THIS THE FIRST ENTRY TO LINK5?

YES

INITIALIZE
REWIND TAPES
READ CONTROL DATA
CHECK TAPE IDENTITIES
READ PARTICLE AND ATMOSPHERIC DATA
PREPARE FIRST PART OF TRANSPORT INTERMEDIATE OUTPUT TAPE (IPOUT)

ANY MORE TIME INTERVALS TO BE DEALT WITH?

YES

IEXEC = 1
RETURN

NO

MAKE FINAL TRANSPORT COMMENTS TO USER AND OPERATOR

RETURN TO GET OR COMPUTE A NEW WIND FIELD

IEXEC = 0

RETURN TO CALL OUTPUT PROCESSOR OR STOP

FC-4. General Flow Chart for Subroutine LINK5
FC-5. Detailed Flow Charts for Subroutine LINK5
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
WRITE A TERMINAL ZERO ON IPAROT

REWIND IPAROT

PRINT TITLE FOR PARTICLES LOST AT THE FINAL TIME BOUNDARY

READ N FROM IPAROT

\[ N > 0 \]

NO

GO TO 501

YES

READ N PARTICLE DESCRIPTIONS FROM IPAROT

PRINT N PARTICLE DESCRIPTIONS

ON PRECEDING PAGE

(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:

<table>
<thead>
<tr>
<th>Identification Number</th>
<th>Program to be Used</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MTWND1</td>
<td>Mountain wind</td>
</tr>
<tr>
<td>2</td>
<td>RGWND1</td>
<td>Ridge wind</td>
</tr>
<tr>
<td>3</td>
<td>CBREZ1</td>
<td>Sea breeze</td>
</tr>
</tbody>
</table>

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:
START

K = 0

K = K + 1

DATA STATING THE COORDINATE LIMITS AND COMPUTATION TYPE OF ONE LOCAL CELI.

READ FROM INPUT TAPE CRMINX(K), CRMAXX(K)
CRMINY(K), CRMAXY(K), NCRTYP(K)

NCIR = NCRTYP(K)

NCIR

NCIR > 5

PRINT: LOCAL CIRCULATION CODE (NCIR) IS NOT AVAILABLE

NO

YES

NO

CALL ERROR

RETURN

ERROR = 122

ERROR = 124

NLOCIR = K - 1

PROGRM = RDCIRS (BCD)

7734

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) > 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) > 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
FC-8. Detailed Flow Charts for Subroutine MKWIND
FC-8. (Continued) Detailed Flow Charts for Subroutine MKWIND
(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
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 \(0 < \text{NCODE} < 6\) 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 NCCDE, 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 \(\text{VX}(J)\), \(\text{VY}(J)\), and \(\text{VZ}(J)\). The storage index of the first entry in the wind-field description arrays for the first stratum is set at 1 (i.e., \(\text{IBADD}(1) = 1\)), the stratum index \(\text{JW}\) 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 \(\text{JW}\) 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 \(\text{XG}, \text{YG}, \text{and ZG}\) 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 \(\text{NAD}\), 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 \(\text{NAD}(J) = J, \ J = 1, \ J_{TOPV}\), 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 \(\text{NAD}\) do not reference data vectors in order of increasing distance from the grid point \((\text{XG}, \text{YG}, \text{ZG})\), but merely provide an initial input to a sort procedure that will provide such an ordering. Initially, we set \(\text{NADT}\), the index of the \(\text{NAD}\) 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 ≥ 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 $\frac{BBB}{BB}$ is found. If BB is less than 10^{-20} or the ratio $\frac{BBB}{BB}$ is less than 10^{-3}, an excessive number of significant figures are lost in the least-squares calculation for this particular grid point (i.e., the normal equations matrix is essentially singular), and the code prints this information and then branches
to the preferential weighting method (as though NCODE = 1). If neither of these cases occurs, the wind velocity vectors are computed and stored using index K.

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

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

Subroutine LINK7 (FC-9 and FC-10)

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

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

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

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

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

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

The main transport loop (between statement numbers 1001 and 160) uses the index J to identify the current particle description. It begins by determining if the current (Jth) particle is to be transported. To be transportable it must be identified by a positive FMAS(J) and $0 < TP(J) < TLIMIT$; 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.
General Flow Chart for Subroutine LINK7

FC-9.
FC-10. Detailed Flow Charts for Subroutine LINK7
(b)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
STOP = 0 indicates that no particles are beyond limits of in-core topographic data.

STOP

SELECT NEXT TOPO FILE FOR READING

MOVE TAPE TO NEXT TOPO FILE

READ TOPO FILE

RESET TO POSITIVE THE SIGNS OF ALL IN-CORE PARTICLES THAT WERE PREVIOUSLY OFF-TOPO

WERE THE OFF-TOPO PARTICLES IN CORE ONLY

STOP

STOP = 0 JTIME1 = -1

WRITE TERMINAL ZERO BLOCK COUNT ON SOUTOCORE REWIND PARTICLES ALOFT INPUT AND OUTPUT MEMORY TAPE FILE NAME

ENDTM = TLMEM

JTIME1 = 0 JDONE = 1

WRITE A TERMINAL BLOCK COUNT ON IN/BOT REWIND PARTICLES ALOFT INPUT AND OUTPUT MEMORY TAPE FILE NAME

ENDTM = TLMEM

JTIME1 = -1 JTIME2 =

RETURN TO LINKS VIA EXECUTIVE

(c)

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
130

SELECT NEXT WIND FILE FOR READING

MOVE TAPE TO NEXT WIND FILE

READ WIND FILE

124

RESET TO POSITIVE THE SIGNS OF ALL IN-CORE PARTICLES THAT WERE PREVIOUSLY BEYOND THE IN-CORE WIND FIELD

IF = NALOFT

JWIND = 0

JTIME1 = -1

WRITE A TERMINAL ZERO BLOCK COUNT ON JWIND

RESET JWIND = 0

REWIND THE OUT-OF-WIND TAPE, SNAP NAMES ON OUT-OF-WIND TAPE AND PARTICLES ALOFT TAPES

50 TO 1081

(d)

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

FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
ARE THERE ANY LOCAL CIRCULATION SYSTEMS

KLOCIR

LD = 1

TMEAN = TLIMIT

NO LOCAL CIRCULATION SYSTEMS

GO TO 172

COMPUTE THE TIME OF FLIGHT TO EACH OF THE FOUR VERTICAL PLANES THAT BOUND THE LD-TH LOCAL CIRCULATION CELL.

TXI = 100000.0
TX2 = 100000.0

TY1 = (CRMIN(L2) - TY0) / VY(YMAX - YMIN)
TY2 = (CRMAX(L2) - Y0) / VY(YMAX - YMIN)

(g)
FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
(b) FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
FC-10. (Continued) Detailed Flow Charts for Subroutine LINK7
FC-10. (Continued) Detailed Flow Charts of Subroutine LINK7
FC-10. (Continued) Detailed Flow Charts of Subroutine LINK?
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
START

IS THE MAGNITUDE OF JW STILL VALID?

JW

NO

0

JT = JT + JW

JW = 1

FIND INDEX (JW) OF WIND LAYER THAT CONTAINS POINT XX, YY, ZZ

153

SET INDEX FOR TRIAL STRATUM

JT = JT + JW

JW = 1

IS PARTICLE ABOVE BASE OF TRIAL STRATUM?

0

ERROR = 155

JT = JT + JW

151

ZZ = BOTH

NO

YES

PROG = GETWND (BCD)

ERROR

270

IS THE PARTICLE WITHIN THE AREA OF THE SPECIFIED WIND FIELD?

NO

INDICATE THE PARTICLE BEYOND THE WIND FIELD BY SETTING JWAD = -1

YES

COMPUTE THE INDEX (JWAD) FOR RETRIEVING WIND VECTORS FROM THE ONE-DIMENSIONAL WIND FIELD ARRAYS

RETURN

FC-11. Flow Chart for Subroutine GETWND
local transport volume, or resting at a macrowind layer boundary or the time
boundary. In any case, a transfer is made to statement number 131.

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

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

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

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

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

NCIR = NCRTYP(K)

MAKE AN ASSIGNMENT TO ALLOW EFFICIENCY IN BRANCHING DURING LOCAL CELL TRANSPORT

 Branch on NCIR

1  2  3  4  5

ASSIGN 121 TO NC  ASSIGN 123 TO NC  ASSIGN 124 TO NC  ASSIGN 125 TO NC

GET FALL RATE FOR PARTICLE BY CALLING SUBROUTINE FALRAT

PARTICLE IS ABOVE A LOCAL CELL!

IS THE JTH PARTICLE ABOVE THE TOP OF THE KTH LOCAL WIND CELL?

NO  GO TO 1293

YES

GET THE MACROWIND VECTORS FOR THE PARTICLE POSITIONS BY CALLING SUBROUTINE GETWND

DO WAS THE MACROWIND FIELD AVAILABLE?

NO

YES

COMPUTE VPZ, THE VERTICAL COMPONENT OF PARTICLE VELOCITY

COMPUTE TIMES OF FLIGHT TO THE TWO BOUNDING PLANES THAT THE PARTICLE WILL INTERCEPT IN THE FUTURE, THE LOCAL CELL TOP/MACROWIND LAYER THAT THE PARTICLE IS IN, AND THE TIME BOUNDARY SET ITTRANS EQUAL TO THE TIME UNTIL FIRST OF THESE INTERCEPTS

TRANSPORT THE PARTICLE FOR THE INTERVAL ITTRANS BY USING THE MACROWIND VECTORS

GO TO 131

(a)

FC-12. Flow Charts for Subroutine LOTRAN
FC-12. (Continued) Flow Charts for Subroutine LOTRAN
The height $CRHT.K.$ of the $K$th 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 $K$th local cell, or if the number of mountains exceeds the maximum of twelve mountains 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 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 $J$th particle from the $I$th mountain in terms of a component parallel and a component perpendicular to the direction of the unperturbed wind. The analytical height of the $I$th 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 $\{0, 0, -1E8\}$ 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 $K$th local cell consists of two logical routes that serve the purpose of reading ridge wind data and computing the ridge wind components for the $J$th particle at location $XP(J), YP(J), ZP(J)$, after first checking for impact on
FC-13. Flow Charts for Subroutine MTWND1
(Continued) Flow Charts for Subroutine MTWND1

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

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

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

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

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

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

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

The read route starts by reading from the system input tape values for parameters $B$, $\text{GRAD}$, $\text{NN}$ and the pairs $\text{DELTX}(N)$ and $\text{TAUX}(N)$ for $N$ ranging from 1 to $\text{NN}$. The maximum value of $\text{NN}$ is nine (see Table 3). $N$ represents the order of the harmonic described by $\text{DELTX}(N)$ and $\text{TAUX}(N)$. These parameters serve to calculate $\text{OMGX}(N)$, $\text{AJZX}(N)$, $\text{AJXX}(N)$, $\text{AJY}(N)$, $\text{R1RX}(N)$, $\text{R1IX}(N)$, $\text{R2RX}(N)$, $\text{R2IX}(N)$, $\text{ESQ1}$, $\text{ESQ2}$, $\text{AN1}$, $\text{AN2}$, $\text{FAX}(N)$, $T1$, $\text{DELTX}(N)$, and $\text{TAUX}(N)$, for each $N$th harmonic, and are printed out as the respective symbols: $\text{OMGN}(N)$, $\text{AJZ}(N)$, $\text{AJX}(N)$, $\text{AJY}(N)$, $\text{ALKN1}(N)$, $\text{ALN1}(N)$, $\text{AKN2}(N)$, $\text{ALN2}(N)$, $\text{BLOW1}(N)$, $\text{BLOW2}(N)$,
FC-14. Flow Charts for Subroutine RGWND1
NRG = J - 1

XX = (CRMAXX(K) + CRMINX(K)) / 2.0
YY = (CRMAXY(K) + CRMINY(K)) / 2.0
ZZ = CRUHT(K) / 2.0

CALL GETWND (XX, YY, ZZ, JWAD, JP)

JWAD

0

UO(K) = [VX(JWAD) * VX(JWAD) + VY(JWAD) * VY(JWAD)] 1/2
SN(K) = VY(JWAD) / UO(K)
CS(K) = VX(JWAD) / UO(K)

J = 1

SSG = CS(K) * D(j) + SN(K) * C(j)
CCG = CS(K) * C(j) - SN(K) * D(j)
AI(J) = -A(j) * H(j) * UO(K) * CCG * CCG
SG(J) = SSG / CCG
CG(J) = 2.0 / CCG

J = J + 1

NO

J > NRG

YES

PRINT: K
PRINT: J, H(J), A(J), XM(J), YM(J), B(J), I = 1, NRG
PRINT: CRMAXX(K), CRMINX(K), CRMAXY(K), CRMINY(K), CRUHT(K)

RETURN

(b)

FC-14. (Continued) Flow Charts for Subroutine RGWND1
FC-14. (Contined) Flow Charts for Subroutine RGWND1
ANN1(N), ANN2(N), PHIN(N), ENU(N), DELTX(N), and TAUX(N). Correspondences of these mnemonics with the symbols defined in the Physical and Mathematical Models section are given in Tables 3 and 4.

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

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

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

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

To aid the user in evaluating the properties of the sea-breeze circulation system generated from the input data, an "interpretative output" of key model parameters is provided. This output is described in Table 4. These parameters are...
<table>
<thead>
<tr>
<th>Physical Quantity</th>
<th>Text Designation</th>
<th>Program Designation</th>
<th>Dimension Units</th>
<th>Typical Values and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of harmonics</td>
<td>n</td>
<td>NN</td>
<td></td>
<td>Approximately two or three</td>
</tr>
<tr>
<td>Total extent of sea breeze</td>
<td>( L_x )</td>
<td>ELX</td>
<td>m</td>
<td>Less than ( 10^5 ) m</td>
</tr>
<tr>
<td>Sine of latitude</td>
<td>( \sin \phi )</td>
<td>SNI PHI</td>
<td></td>
<td>(-1 \leq \sin \phi \leq +1)</td>
</tr>
<tr>
<td>Angle of coastline relative to y axis of grid</td>
<td>( \psi )</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wind field extrapolation attenuation constant</td>
<td>( k_a )</td>
<td>WW</td>
<td>m(^{-1})</td>
<td>(0 \leq k_a \leq \infty)</td>
</tr>
<tr>
<td>Guldberg-Mohn friction parameter</td>
<td>( \sigma )</td>
<td>SGMA</td>
<td>sec(^{-1})</td>
<td>A value of zero is not allowed; typical values (0.5 \times 10^{-4} \leq \sigma \leq 2.5 \times 10^{-4})</td>
</tr>
<tr>
<td>Average ground temperature</td>
<td>( \theta_0 )</td>
<td>THET</td>
<td>(^0)K</td>
<td>Expressed in degrees Kelvin: (\theta_0 = 300^0)K</td>
</tr>
<tr>
<td>Unperturbed temperature gradient</td>
<td>( \Gamma^* = (d\theta_0 / dz) )</td>
<td>GRAD</td>
<td>(^0)Km(^{-1})</td>
<td>A constant z-independent positive value must be used; typical values (4 \times 10^{-3} \leq \Gamma \leq 7.5 \times 10^{-3})</td>
</tr>
<tr>
<td>Thermal eddy diffusivity</td>
<td>( K )</td>
<td>AKY</td>
<td>m(^2)sec(^{-1})</td>
<td>A constant z-independent value must be used; typical values (25 \leq K \leq 75)</td>
</tr>
<tr>
<td>Magnitude of nth temperature differential</td>
<td>( T_n^* )</td>
<td>DELTX(N)</td>
<td>(^0)K</td>
<td>First harmonic will generally be less than (10^0)K, with subsequent harmonics decreasing in magnitude</td>
</tr>
<tr>
<td>Phase of nth temperature differential</td>
<td>( r_n )</td>
<td>TAUX(N)</td>
<td></td>
<td>Phase of first harmonic should correspond to about 1 hr, or (\tau_1 \sim \Omega (8.6) \times 10^3 = 0.26)</td>
</tr>
<tr>
<td>Lag time between sea-breeze local time and Greenwich time</td>
<td>( \Delta t_s )</td>
<td>ELAG</td>
<td>sec</td>
<td></td>
</tr>
</tbody>
</table>
**TABLE 4**

**INTERPRETATIVE OUTPUT DESCRIPTION**

<table>
<thead>
<tr>
<th>Output Designation</th>
<th>Text Designation</th>
<th>Interpretative Output Description Designation</th>
</tr>
</thead>
<tbody>
<tr>
<td>OMGN(N)</td>
<td>nΩ</td>
<td>nΩ</td>
</tr>
<tr>
<td>AJZ (N)</td>
<td>J_{nz}</td>
<td>-T* L/B</td>
</tr>
<tr>
<td>AJX (N)</td>
<td>J_{nx}</td>
<td>λ^{-1} J_{nz}</td>
</tr>
<tr>
<td>AJY (N)</td>
<td>J_{ny}</td>
<td>G J_{nx}</td>
</tr>
<tr>
<td>AKN1 (N)</td>
<td>k_{n1}</td>
<td>k_{1}</td>
</tr>
<tr>
<td>ALN1 (N)</td>
<td>l_{n1}</td>
<td>l_{1}</td>
</tr>
<tr>
<td>AKN2 (N)</td>
<td>k_{n2}</td>
<td>k_{2}</td>
</tr>
<tr>
<td>ALN2 (N)</td>
<td>l_{n2}</td>
<td>l_{2}</td>
</tr>
<tr>
<td>BLOW1 (N)</td>
<td>K_{n1}</td>
<td>ε_{1} U_{1}</td>
</tr>
<tr>
<td>BLOW2 (N)</td>
<td>K_{n2}</td>
<td>ε_{2} U_{2}</td>
</tr>
<tr>
<td>ANN1 (N)</td>
<td>η_{n1}</td>
<td>η_{1}</td>
</tr>
<tr>
<td>ANN2 (N)</td>
<td>η_{n2}</td>
<td>η_{2}</td>
</tr>
<tr>
<td>PHIN (N)</td>
<td>φ_{n}</td>
<td>h - m + τ_{n}</td>
</tr>
<tr>
<td>ENU (N)</td>
<td>ν_{n}</td>
<td>-θ_{1}</td>
</tr>
</tbody>
</table>
are sufficient to calculate the wind velocity components $w_n$, $u_n$, and $v_n$ as given by Eqs. (30), (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:

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

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

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

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 $\tau_n$ 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 nth mode. This permits many of the dummy analytical variables subsequently defined to bear a one-to-one correspondence with the mode-dependent variables. For instance, defined quantities such as $q$, $a$, $b$, and $\epsilon_1$ correspond to $q_n$, $a_n$, $b_n$, $\epsilon_{1n}$.

The next group of calculations is:

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

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

$$q^2 + f^2 = A_3 e^{i\theta_3}, A_3 = \left(\left(\sigma^2 + f^2 - \Omega^2\right)^2 + 4\Omega^2\sigma^2\right)^{1/2}, \theta_3 = \tan^{-1}\left(\frac{2\Omega\sigma}{\sigma^2 + f^2 - \Omega^2}\right).$$
We next turn our attention to the coefficients a, b, c, and d:

\[ a = q^2 \frac{X^2}{(q^2 + f^2)} = A_4 e^{i\theta_4}, \quad A_4 = (A_2 \frac{X^2}{A_3}), \quad \theta_4 = \theta_2 - \theta_3; \]

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

\[ c = (\Gamma / K) = A_7; \]

\[ d = i (\Omega / K) = iA_6, \quad A_6 = (\Omega / K). \]

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

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

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

\[ R = \left( (a + d)^2 - 4(ad + bc) \right)^{1/2} = B e^{i \phi}, \]

where

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

\[ = \xi_1 + i \xi_2, \]

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

and

\[ B_1 = \xi_1^2 - \xi_2^2 - 4 \left( \Lambda_7 \cos h - A_4 A_6 \sin \theta_4 \right), \]

\[ B_2 = 2 \xi_1 \xi_2 - 4 \left( A_4 A_6 \cos \theta_4 + \Lambda_7 \sin h \right), \]

\[ \beta = \tan^{-1} \left( \frac{B_2}{B_1} \right). \]
We define

\[ B = \left( B_1^2 + B_2^2 \right)^{1/4}, \]
\[ m = \beta/2. \]

Using the foregoing expressions, \( C_1, D_1, C_2, D_2, E_1, E_2, \gamma_1, \) and \( \gamma_2 \) are then calculated.

\[ C_1 = \frac{1}{2} \left[ A_4 \cos \theta_4 + B \cos m \right]; D_1 = \frac{1}{2} \left[ A_4 \sin \theta_4 + A_6 + B \sin m \right]; \]
\[ C_2 = \frac{1}{2} \left[ A_4 \cos \theta_4 - B \cos m \right]; D_2 = \frac{1}{2} \left[ A_4 \sin \theta_4 + A_6 - B \sin m \right]; \]

\[ \gamma_1 = \tan^{-1} \left( \frac{D_1}{C_1} \right); \quad \gamma_2 = \tan^{-1} \left( \frac{D_2}{C_2} \right); \]
\[ E_1 = \left( C_1^2 + D_1^2 \right)^{1/2}; \quad E_2 = \left( C_2^2 + D_2^2 \right)^{1/2}. \]

The attenuation constants for the nth mode (symbolized here by \( \alpha_1 \) and \( \alpha_2 \)) are given by

\[ \alpha_1 = \pm \left( \mu_1 \right)^{1/2} = \epsilon_1 U_1 e^{i \eta_1} = k_1 + i \ell_1, \]
\[ \alpha_2 = \pm \left( \mu_2 \right)^{1/2} = \epsilon_2 U_2 e^{i \eta_2} = k_2 + i \ell_2, \]

where

\[ U_1 = E_1^{1/2}, \quad U_2 = E_2^{1/2}, \quad \eta_1 = \gamma_1/2, \quad \eta_2 = \gamma_2/2, \]
\[ k_1 = \epsilon_1 U_1 \cos \eta_1, \quad \ell_1 = \epsilon_1 U_1 \sin \eta_1, \]
\[ k_2 = \epsilon_2 U_2 \cos \eta_2, \quad \ell_2 = \epsilon_2 U_2 \sin \eta_2, \]

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

114
Next, the following quantities are calculated:

\[
\phi' = h - m ,
\]
\[
g = - (f/q) = G e^{i\nu} ,
\]

where

\[
G = -f/\left(\sigma^2 + \Omega^2\right)^{1/2} = -\left(f/A_1\right) ,
\]
\[
\nu = -\theta_1 .
\]

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
FC-15. Flow Charts for Subroutine CBREZ1
FC-15. (Continued) Flow Charts for Subroutine CBREZ1
(c) FC-15. (Continued) Flow Charts for Subroutine CBREZ1
FC-15. (Continued) Flow Charts for Subroutine CBREZ1
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.
FC-16. Flow Charts for Subroutine HEIGHT
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 identi-
fier 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 dur-
ing 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 sub-
divisions 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>
<thead>
<tr>
<th>Logical Binary Record</th>
<th>Record Contents</th>
<th>Variable Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tape identification word (IPARIN), spare, x and y coordinates of ground zero,</td>
<td>DENTI, NSP, XGZ, YGZ, TGZ, BZ, NCL</td>
</tr>
<tr>
<td></td>
<td>shot time, cloud subdivision edge length, spare</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Detonation parameters: yield, cloud soil burden, soil solidification temperature,</td>
<td>FW, SSAM, SLDTMP, TMSD, SIGMA, SPARE1, SPARE2, SPARE3</td>
</tr>
<tr>
<td></td>
<td>soil solidification time, ln(SD), † spares</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>LINK4 run identification</td>
<td>PSEID(J), J = 1, 12</td>
</tr>
<tr>
<td>4</td>
<td>Cloud-rise identification</td>
<td>CRID(J), J = 1, 12</td>
</tr>
<tr>
<td>5</td>
<td>Initial-conditions run identification</td>
<td>DETID(J), J = 1, 12</td>
</tr>
<tr>
<td>6</td>
<td>Fallout particle density</td>
<td>ROPART</td>
</tr>
<tr>
<td>7</td>
<td>Number of particle size ranges</td>
<td>NPS</td>
</tr>
<tr>
<td>8</td>
<td>Central particle size, associated mass, associated activity, * and surface-to-</td>
<td>PS(I), A(I), PACT(I), SV(I), I = 1, NPS</td>
</tr>
<tr>
<td></td>
<td>volume ratio for each size range</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Number of atmospheric strata</td>
<td>NAT</td>
</tr>
<tr>
<td>10</td>
<td>Atmospheric viscosity and density for each stratum</td>
<td>ATEMP(I), RHO(I), I = 1, NAT</td>
</tr>
<tr>
<td>11</td>
<td>Number of particles described in the first data block</td>
<td>NP</td>
</tr>
<tr>
<td>12</td>
<td>Particle data for first data block: x, y, z, and time coordinates, particle size,</td>
<td>XPAR(I), YPAR(I), ZPAR(I), TP(I), PSIZ(I),</td>
</tr>
<tr>
<td></td>
<td>mass per unit area of cloud subdivision bottom</td>
<td>SMAS(I), I = 1, NP</td>
</tr>
<tr>
<td>13</td>
<td>Same as record 11 for the second data block</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Same as record 12 for the second data block</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>M</td>
<td>Tape termination indicator</td>
<td>NP = 0</td>
</tr>
<tr>
<td>M + 1</td>
<td>End of file</td>
<td></td>
</tr>
</tbody>
</table>

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

†See LINK1 glossary in DASA-1800-II.
<table>
<thead>
<tr>
<th>I</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>IC(1) &gt; 0, suppresses usage of topography tape, IHTOPO, and a planar topography is assumed.</td>
</tr>
<tr>
<td>2</td>
<td>IC(2) &gt; 0, suppresses usage of off-to-po secondary tape, IOTOPO</td>
</tr>
<tr>
<td>3</td>
<td>IC(3) &gt; 0, suppresses usage of out-of-wind-field secondary tape, IOWIND</td>
</tr>
<tr>
<td>4</td>
<td>IC(4) &gt; 0, suppresses usage of time-boundary secondary tape, IPAROT</td>
</tr>
<tr>
<td>5</td>
<td>IC(5) &gt; 0, suppresses usage of all secondary tapes</td>
</tr>
<tr>
<td>6</td>
<td>IC(6) &lt; 1, no transport traces are printed</td>
</tr>
<tr>
<td></td>
<td>IC(6) = 1, in-core particle arrays are printed following read-in of each block of particles from IPARIN (see p. 78)</td>
</tr>
<tr>
<td></td>
<td>IC(6) &gt; 1, in addition, a print-out is executed following each transport increment (see p. 78)</td>
</tr>
<tr>
<td>7</td>
<td>IC(7) = 1, causes the computed wind field to be printed each time it is updated (see Table 13)</td>
</tr>
<tr>
<td>8</td>
<td>IC(8) = 0, causes a listing of lost particles (see Table 2) whenever a group of lost particles are discarded by subroutine DUMPP.</td>
</tr>
</tbody>
</table>
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>
<thead>
<tr>
<th>Card Number</th>
<th>Content</th>
<th>Variable Names and Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transport model run identification</td>
<td>TID(J), J = 1, 12 (12A6)</td>
</tr>
<tr>
<td>2</td>
<td>Control integer array</td>
<td>IC(J), J = 1, 18 (18I4)</td>
</tr>
<tr>
<td>3</td>
<td>Transport time limit (sec)</td>
<td>TLIMIT (F10.5)</td>
</tr>
<tr>
<td>4</td>
<td>Altitude of planar topography. This card is to be omitted if a topography input tape is used (m).</td>
<td>TTOPO (F10.5)</td>
</tr>
<tr>
<td>5</td>
<td>Wind-field data set identification card</td>
<td>WID(J); J = 1, 12 (12A6)</td>
</tr>
</tbody>
</table>

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**

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

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 WIND.
<table>
<thead>
<tr>
<th>Card Number</th>
<th>Content</th>
<th>Variable Names and Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Time (seconds) at which the forthcoming wind data set is to be updated, α, β (Eq. 21)</td>
<td>ENDTIM, ALPHA, BETA (3F10.3)</td>
</tr>
<tr>
<td>2</td>
<td>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).</td>
<td>NN, NCODE (2I4)</td>
</tr>
<tr>
<td>3</td>
<td>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.)</td>
<td>BOTHIT(J), WGRIDNT(J), WLLX(J), WLLY(J), WURX(J), WURY(J), J = 1 (6F10.3)</td>
</tr>
<tr>
<td>4</td>
<td>Same as card 3 but for second stratum</td>
<td>Same as card 3 but for J = 2</td>
</tr>
<tr>
<td></td>
<td>The end of the subdivision specifications is marked by the number 999999.0 in the stratum base altitude place</td>
<td></td>
</tr>
<tr>
<td>First</td>
<td>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)</td>
<td>ZS(J), XS(J), YS(J), SX(J), SY(J), SZ(J), J = 1 (6F12.3)</td>
</tr>
<tr>
<td>Second</td>
<td>Same as preceding card but for second data vector</td>
<td>Same as preceding card but for J = 2</td>
</tr>
<tr>
<td></td>
<td>The end of the deck of data vectors is marked by the number 999999.0 in the vector altitude position</td>
<td></td>
</tr>
</tbody>
</table>
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 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:

Table 10  
CARD INPUTS TO SUBROUTINE RDCIRS

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Content</th>
<th>Variable Names and Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Coordinates of planes that bound the Jth local circulation system cell on the west, east, south, and north, respectively, and the circulation type identifier.</td>
<td>CRMINX(J), CRMAXX(J), CRMINY(J), CRMAXY(J), NCRTYP(J) (4E12.5, I3)</td>
</tr>
<tr>
<td>Last Card</td>
<td>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.</td>
<td>Blank</td>
</tr>
</tbody>
</table>
Identification Number | Local Circulation Type
--- | ---
0 | Marks the end of the set of local circulation cell descriptions
1 | Mountain wind (MTWND1)
2 | Ridge wind (RGWND1)
3 | See breeze (CBREZ1)
4 | Not assigned
5 | Not assigned

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

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

**Topography Data**

Two basically different forms of topography may be specified for use by the Transport Module in regions not covered by local circulation systems. They are referred to here as fully planar topography (a single plane) and piecewise-planar topography (many segments of planes). The choice of method of topographic description is communicated to the Transport Module by the user in the control parameter IC(1) (see Table 6) which must be given the value 1 if the fully planar option is desired and 0 if not. In the fully planar option, the program merely reads from a
### TABLE 11

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

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

132
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**

<table>
<thead>
<tr>
<th>THE BINARY TOPOGRAPHY TAPE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Record Number</td>
</tr>
<tr>
<td>----------------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>

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 |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

\[
\begin{align*}
S(1, 1) & \quad S(1, 2) & \quad S(1, 3) & \quad \ldots & \quad S(I, JJ) \\
S(2, 1) & \quad S(2, 2) & \quad S(2, 3) & \quad \ldots & \quad S(2, JJ) \\
& \vdots & \quad \vdots & \quad \ddots & \quad \vdots \\
S(I, 1) & \quad S(I, 2) & \quad S(I, 3) & \quad \ldots & \quad S(I, JJ)
\end{align*}
\]

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:

<table>
<thead>
<tr>
<th>I</th>
<th>TOPOLM</th>
<th>ITOPLM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BXLL</td>
<td>II</td>
</tr>
<tr>
<td>2</td>
<td>BYLL</td>
<td>JJ</td>
</tr>
<tr>
<td>3</td>
<td>GRINT</td>
<td>KK</td>
</tr>
<tr>
<td>4</td>
<td>TTOPO</td>
<td></td>
</tr>
</tbody>
</table>

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>
<thead>
<tr>
<th>Output Sequence</th>
<th>Content</th>
<th>Variable Names</th>
</tr>
</thead>
</table>
| 1               | Run identifiers for LINK1, LINK2, LINK4, and transport | DETTD(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. | TTOPO  
|                 | b. If a piecewise planar topography is specified, the topography tape (INTTOPO) identifier is printed. | 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  

137
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 written 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)

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

*Not yet calculated unless the user has provided a LINK3 particle activity calculation
†See LINK1 glossary in DASA-1800-II.
REFERENCES


FORTRAN LISTINGS

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

REAL WYNE

DIMENSION ATMP(260), XNUM(260)

REAL LOGIC

**** FALRAT GLOSSARY ****

LOG  = LOGICAL VALUE

I = INTEGER VALUE

ATMP(i)  = DYNAMIC VISCOSITY OF AIR AT (i-1)*200 METER ABOVE
 WEIGHTED AVERAGE OF VISCOSITY VALUES

CURR = CURRENT COEFFICIENT OF VISCOSITY

NUM = NUMBER

FRG = FOG

PGA = PARTICLE DIAMETER

XNUM = X NUMBER

**** FORMAT FALRAT INPUT DATA ****

INDEP = INDEPENDENT VARIABLE

1 = LOGICAL VALUE

I = INTEGER VALUE

ATMP(i)  = DYNAMIC VISCOSITY OF AIR AT (i-1)*200 METER ABOVE
 WEIGHTED AVERAGE OF VISCOSITY VALUES

CURR = CURRENT COEFFICIENT OF VISCOSITY

NUM = NUMBER

FRG = FOG

PGA = PARTICLE DIAMETER

XNUM = X NUMBER

**** FALRAT OUTPUT RESULTS ****

LOG  = LOGICAL VALUE

I = INTEGER VALUE

ATMP(i)  = DYNAMIC VISCOSITY OF AIR AT (i-1)*200 METER ABOVE
 WEIGHTED AVERAGE OF VISCOSITY VALUES

CURR = CURRENT COEFFICIENT OF VISCOSITY

NUM = NUMBER

FRG = FOG

PGA = PARTICLE DIAMETER

XNUM = X NUMBER

**** FALRAT RUN CHECK ****

LOG  = LOGICAL VALUE

I = INTEGER VALUE

ATMP(i)  = DYNAMIC VISCOSITY OF AIR AT (i-1)*200 METER ABOVE
 WEIGHTED AVERAGE OF VISCOSITY VALUES

CURR = CURRENT COEFFICIENT OF VISCOSITY

NUM = NUMBER

FRG = FOG

PGA = PARTICLE DIAMETER

XNUM = X NUMBER

**** FALRAT STOP 14 ****
C200  CURR IS GREATER THAN 1400
C200  ULOGA=ALOG10(CURR)-20*13.5
C200  RV=5057.66*INV(CURR)**(1.2836-2.933*10**-4*RV=
C300  DRAG SLIP CORRECTED FALL RATE
C300  RV=RV*(1.0+0.0381/1/14.1=AND(1))
C300  RETURN
C300  END
SUBROUTINE DUMP

THIS SUBROUTINE SELECTS THE BEST SET OF PARTICLES TO DUMP,
SORTS IT INTO THE LOW NUMBERED END OF THE PARTICLE ARRAYS, WRITES
IT OUT ONTO THE APPROPRIATE TAPE AND ADJUSTS PARTICLE SET COUNTERS.
THE SET SELECTED FOR DUMPING IS THE GROUNDING PARTICLES SET WHEN
EVER DUMPING IT WOULD MAKE SUFFICIENT ROOM FOR THE INCOMING BLOCK
OF N PARTICLES. IF THIS IS NOT THE CASE, THE LARGEST PARTICLE SET IS
SELECTED.

FOR ADDITIONAL GLOSSARY ENTRIES SEE SUBROUTINE LINKS

Glossary

FMAST SEE XPT
IC( ) THE CONTROL INTEGER ARRAY. SEE LINK 5 GLOSSARY
ICGQ BLOCKING SORT MODE INDICATOR. =FIRST PASS, #OUTOUT LOOP
STOP LOUP =TOP LOOP
IOTOP0 THE OFF-TOPU MEMORY TAPE NUMBER
IOWIND THE OUT-OF-WIND DATA MEMORY TAPE NUMBER
IPAKUT THE TIME LIMIT BOUNDARY MEMORY TAPE NUMBER
IPOUT THE TRANSPORT MODULE INTERMEDIATE OUTPUT TAPE NUMBER
IRSET A MARKER FOR THE BLOCKING SORT WHICH INDICATES THE
VALUE I THAT THE TEMPORARY STORAGE LINE FOR A PARTICLE
IS LOADED AND MUST BE EVENTUALLY UNLOADED
JSET THE FORTRAN SYSTEM OUTPUT TAPE NUMBER
J A GENERAL INDEX. IN THE BLOCKING SORT, IT IS USED TO
IDENTIFY THE PARTICLE THAT WAS JUST CLASSIFIED
JB BOTTOM LOOP INDEX FOR THE BLOCKING SORT
JBL BLANK LINE INDEX FOR THE BLOCKING SORT
JFR USED TO RECORD THE INDEX OF A FREE (BLANK) LINE IN THE
BOTTOM PART OF THE PARTICLE ARRAY DURING THE CONSOLIDATION
OF N BLANKS INTO THE TOP OF THE ARRAY
JT TOP LOOP INDEX FOR THE BLOCKING SORT
JTEST A TEMPORARY STORAGE THAT EVENTUALLY CONTAINS THE NUMBER
OF PARTICLE DESCRIPTIONS IN THE CLASS TO BE DUMPED
JTEST1 A TEMPORARY STORAGE WHICH EVENTUALLY CONTAINS THE NUMBER
INDICATING THE EARLIEST (CLASS) OF PARTICLE DESCRIPTION TO BE DUMPED
N THE NUMBER OF PARTICLES IN THE DATA BLOCK THAT IS WAITING
TO BE DUMPED AT THE TIME WHEN DUMP IS CALLED
NALIFT THE DIMENSIONED (MAXIMUM) SIZE OF THE PARTICLE ARRAY
NBMAX THE MAXIMUM NUMBER OF PARTICLE DESCRIPTIONS THAT CAN BE
INCLUDED IN A SINGLE BLOCK AS WRITTEN ON ANY MEMORY OR
INTERMEDIATE OUTPUT TAPE
NFREE THE NUMBER OF BLANK LINES (DENOTED BY FMAST #0) IN THE
PARTICLE ARRAYS
NG A COUNT OF IN-CORE GROUNDING PARTICLES
NLUS1 A COUNT OF THE PARTICLES IN THE PARTICLE ARRAY BUT LOCATED
BEYOND THE COORDINATE LIMITS OF THE WIND OR TOPU DATA SET
NTAP A TEMPORARY STORAGE FOR THE NUMBER OF THE TAPE UNTIL WHICH DUMP
THE DUMP IS TO BE MADE
NT1 A COUNT OF IN-CORE PARTICLES THAT HAVE REACHED THE TIME
BOUNDARY (ENDTIM);
NTU A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE WIND DATA DUMP
NW A COUNT OF IN-CORE PARTICLES BEYOND THE IN-CORE WIND DATA DUMP
C N1 AN ASSIGNED GO TO BRANCH POINT FOR THE CLASSIFYING CODE 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 XPI ) SOMETIMES USED TO START A SURT DUMP 67
C YPT SEE XPT DUMP 68
C ZPT SEE XPT DUMP 69
C
C******************************************************************************
C COMMON /SET1/
1 D1AM *DETIU(12)*IRISE *IEDEC *ISIN *ISWUT *, DUMP 71
2 SD *SPAP *SSAP *TM1 *TMP1 *TMP2 *, DUMP 72
3 TTM *U *VPR *W *X *Z *, DUMP 73
4 WHY(4) *RMIN *IDISTK *SPAP1 *SPAP2 *SPAP3 *, DUMP 74
5 SPAR4 *SPAR5 *SPAR6 *SPAR7 *SPAR8 *SPAR9 *, DUMP 75
C******************************************************************************
C COMMON /SET2/
1 S *SUBSID *GRINT *BXL *BXYL *BYLL *, DUMP 76
2 BYLU *TXLU *TYLU *XC *, DUMP 77
3 YGZ *NGL *HTRU *ITOPU *ITOPU *ILIM *, DUMP 78
4 KLI K *II *JJ *KK *XP *, DUMP 79
5 ZP *FMAS *TP *PS *VX *VY *, DUMP 80
6 VZ *IL *JL *I9ADU *GRINT *, DUMP 81
7 WLLX *WLLY *WURX *WURY *BOTHIT *, DUMP 82
8 JTUPU *IOM *INSWIT *ITOPU *ITOPU *, DUMP 83
9 JNMD1 *MRR *TLIMIT *ENDTIM *IC *, DUMP 84
1 JTUPJ *NLUST *N4 *NTU *NTI *, DUMP 85
2 NALUFT *JTINE1 *NBAX *NKFKE *, DUMP 86
3 CRM *CRAU *NCRTIP *BZ *CRMINX *, DUMP 87
4 *UU *SN *CS *NLUCIR *DTLC *, DUMP 88
5 RHU *NA *TGZ *DTMAC *, DUMP 89
6 ROPART DIMENSION TOPULM(4*4) *NINTAR(4) *ITOPULM(3*4) *, DUMP 90
DIMENSION SI1(10) *SUBSID(400) *IC(18) *, DUMP 91
DIMENSION XP(200) *TP(200) *LP(200) *FMAS(200) *, DUMP 92
DIMENSION TP(200) *PS(200) *TEM(260) *RHO(126) *, DUMP 93
DIMENSION VX(1500) *VY(1500) *VZ(1500) *IL(70) *, DUMP 94
DIMENSION JL(70) *IBADD(70) *WURX(70) *, DUMP 95
DIMENSION WGRINT(73) *MLLX(70) *WLLY(70) *, DUMP 96
DIMENSION WURX(70) *BOTHIT(70) *SN(6) *CS(6) *, DUMP 97
DIMENSION CRMINX(6) *CRM(6) *CRMINY(6) *CRMAXY(6) *, DUMP 98
DIMENSION CRMAXY(6) *NCRTYP(6) *JO(6) *, DUMP 99
C******************************************************************************
1 FORMAT(I12,3X) *X17*13H LUST PARTICLES *, DUMP 100
2 FORMAT(16X,2HXP,1G2HYP,10X,2HFP,10X,2HFP,10X,2HPS,8X,4HFMAS) *, DUMP 101
3 FORMAT(1X,6E12.5) *, DUMP 102
4 FORMAT(I1U15) *, DUMP 103
5 FORMAT(5X,HBEYOND TOPO) *, DUMP 104
6 FORMAT(5X,HBEYOND TIME BOUNDARY) *, DUMP 105
7 FORMAT(5X,HBEYOND MDO) *, DUMP 106
9 FORMAT(5X,HBEYONND GROUNDED) *, DUMP 107
C
C******************************************************************************
C
C******************************************************************************
DATA PROGRAM/6H DUMP/ DUMP 119

C

***wl***i.---*-,**w iu-.

DUMP 120

C

MUST ANY PARTICLES BE DUMPED TO MAKE ROOM FOR THE INCOMING BLOCK

DUMP 121

C

OF N PARTICLES OR TO CLEAR THE PARTICLE ARRAYS. YES TO 151

DUMP 122

IF(N-FREE=150)150,150,151

DUMP 123

150 JTEST=0

DUMP 124

GO TO 152

C

DUMP 125

C

151 WOULD DUMPING THE GROUNDED PARTICLES PROVIDE SUFFICIENT ROOM FOR

DUMP 126

C

THE BLOCK OF N INCOMING PARTICLE DESCRIPTIONS. YES TO 1512

DUMP 127

151 IF(N-FREE+NG-N)1511,1512,1512

DUMP 128

C

DUMP 129

C

PREPARE TO DUMP THE GROUNDED PARTICLE DESCRIPTIONS

DUMP 130

1512 JTEST=1

C DUMP 131

JTEST=NG

GO TO 12

C DUMP 132

1911 DETERMINE WHICH SET OF PARTICLES TO DUMP

DUMP 133

C

IF THE IDENTIFIER (JTEST) AND SIZE (JTEST) OF THE MOST

DUMP 134

C

NUMEROUS CLASS OF PARTICLES IN THE PARTICLE ARRAYS

DUMP 135

1511 IF(N-LOST=NG)10,11,11

DUMP 136

10 JTEST=NG

DUMP 137

.JTEST1=1

DUMP 138

GO TO :2

DUMP 139

11 JTEST=NG

C

GO TO 12

DUMP 140

12 IF(JTEST=NLOST)13,14,14

DUMP 141

13 JTEST=NLOST

DUMP 142

JTEST1=3

DUMP 143

14 IF(JTEST=NIL)15,16,16

DUMP 144

15 JTEST1=4

DUMP 145

JTEST=NI

DUMP 146

16 IF(JTEST=NL)17,18,18

DUMP 147

17 JTEST=NI

DUMP 148

JTEST1=5

DUMP 149

18 AT THIS POINT JTEST HAS XAMING (NLST/NT/LND/FIELD)

DUMP 150

C

JTEST INDICATES THE KIND OF PARTICLE DESCRIPTION TO BE DUMPED

DUMP 151

C

SEE THE FOLLOWING CODE EXPLANATION FOR JTEST1 THRU 5

DUMP 152

C

JTEST1 NAME OF KIND OF PARTICLE DESCRIPTIONS DUMP 153

C

VALUE CLASS COUNTER TO BE DUMPED onto TAPE DUMP 154

C

1 NG GROUNDED PARTICLES DUMP 155

C

2 NLOST PARTICLES BEYOND THE AREAS FOR WHICH DUMP 156

C

3 NTU TOPU DATA CURRENTLY AVAILABLE IN CORE DUMP 157

C

4 NFI PARTICLES THAT CANNOT BE VALIDLY DUMP 158

C

5 NW PARTICLES BEYOND THE LIMITS OF THE DUMP 159

C

DUMP 160

C

DUMP 161

C

DUMP 162

C

DUMP 163

C

DUMP 164

C

DUMP 165

C

DUMP 166

C

DUMP 167

C

DUMP 168

C

DUMP 169

C

DUMP 170

C

DUMP 171

C

DUMP 172

C

DUMP 173

C

DUMP 174

C

DUMP 175

C

DUMP 176
TEST TO SEE THAT JTEST HAS AN ACCEPTABLE VALUE. UNACCEPTABLE TO DUMP 177
ERROR STOP AT 184
184 IF( JTEST ) 185,186,187
185 IRNOK= 184
186 CALL ERROR(IFRUNK=IRNOK=180)
GO TO 60
DUMP 178
DUMP 179
DUMP 180
DUMP 181
DUMP 182
182 IS THE SIZE (JTEST) OF THE SELECTED CLASS GREATER THAN THE MAXIMUM ALLOWABLE BLOCK SIZE (NO_MAX)? YEO TO 183
183 IF( JTEST/NOMAX ) 182,180,185
184 JTEST=NBMAX
DUMP 184
DUMP 185
DUMP 186
DUMP 187
DUMP 188
DUMP 189
DUMP 190
DUMP 191
DUMP 192
DUMP 193
DUMP 194
DUMP 195
DUMP 196
DUMP 197
DUMP 198
DUMP 199
DUMP 200
DUMP 201
DUMP 202
DUMP 203
DUMP 204
DUMP 205
DUMP 206
DUMP 207
DUMP 208
DUMP 209
DUMP 210
DUMP 211
DUMP 212
DUMP 213
DUMP 214
DUMP 215
DUMP 216
DUMP 217
DUMP 218
DUMP 219
DUMP 220
DUMP 221
DUMP 222
DUMP 223
DUMP 224
DUMP 225
DUMP 226
DUMP 227
DUMP 228
DUMP 229
DUMP 230
DUMP 231
DUMP 232
DUMP 233
DUMP 234
148
C THE TRANSPORT PROCESS
21 IF1IC(2)-1)211,212,211 DUMP 235
212 NTAP=ISOUT DUMP 236
GO TO 213 DUMP 237
C 211 JTPO=1 INDICATES THAT THE ONLY OFF-TOPO PARTICLES THAT REMAIN IN DUMP 238
C THE TRANSPORT ARE THOSE THAT ARE IN CORE IN THE PARTICLE ARRAYS. DUMP 239
C
C 211 JTPO=1 DUMP 240
211 NTAP=IOTopo DUMP 241
212 NTN=I1-JTEST DUMP 242
ASSIGN 3UU TO N2 DUMP 243
ASSIGN 1U0 TO N3 DUMP 244
ASSIGN 42 TO N4 DUMP 245
ASSIGN 42 TO N5 DUMP 246
GO TO 99 DUMP 247
C DUMP 248
C 26 CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT CAN DUMP 249
C NOT BE TRANSPORTED FURTHER UNTIL THE WIND FIELD IS UPDATED. INCLUD G DUMP 250
C PARTICLES ARE IDENTIFIED BY A POSITIVE MASS ) AND A IP( ) EQUAL DUMP 251
C TO ENDING. NORMALLY THEY ARE WRITTEN ON TAPE JINAPUT BUT WHEN THE DUMP 252
C USER HAS SET IC(4)=1 TO SUPPRESS JINAPUT THEY ARE WRITTEN ON THE DUMP 253
C SYSTEM OUTPUT TAPE TO NOTIFY THE USER. DUMP 254
C
C 22 IF1IC(4)=1)221,222,223 DUMP 255
222 NTAP=ISOUT DUMP 256
GO TO 223 DUMP 257
C DUMP 258
C 221 JTONE=1 INDICATES THAT THE ONLY OUT-OF-WIND PARTICLES THAT REMAIN DUMP 259
C IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS. DUMP 260
C 221 JTONE=1 DUMP 261
221 NTAP=IPARUT DUMP 262
223 NTN=I1-JTEST DUMP 263
ASSIGN 6U0 TO N2 DUMP 264
ASSIGN 42 TO N1 DUMP 265
GO TO 99 DUMP 266
C DUMP 267
C 23 CODE TO MAKE ASSIGNMENTS FOR THE SELECTION OF PARTICLES THAT ARE DUMP 268
C BEYOND THE LIMITS OF THE WIND DATA CURRENTLY AVAILABLE IN CORE. DUMP 269
C THESE PARTICLES ARE IDENTIFIED BY A NEGATIVE MASS ) AND POSITIVE DUMP 270
C IP( ) NORMALLY THEY ARE WRITTEN ONLY TAPE JINAPUT BUT WHEN THE DUMP 271
C USER HAS SET IC(3)=1 TO SUPPRESS JINAPUT THEY ARE WRITTEN ON THE DUMP 272
C SYSTEM OUTPUT TAPE TO NOTIFY THE USER. DUMP 273
C 23 IF1IC(3)=1)231,232,233 DUMP 274
232 NTAP=ISOUT DUMP 275
GO TO 233 DUMP 276
C DUMP 277
C 231 JWIND=1 INDICATES THAT THE ONLY OUT-OF-WIND-FIELD PARTICLES THAT REMAIN DUMP 278
C IN THE TRANSPORT ARE THOSE THAT ARE IN THE PARTICLE ARRAYS. DUMP 279
231 JWIND=1 DUMP 280
231 NTAP=IOWIN DUMP 281
233 N4=NW-JTEST DUMP 282
ASSIGN 3UU TO N1 DUMP 283
ASSIGN 1U0 TO N4 DUMP 284
ASSIGN 42 TO N2 DUMP 285
ASSIGN 42 TO N3 DUMP 286
GO TO 233 DUMP 287
C DUMP 288
C 99 INITIALIZE FOR BLOCKING SORT DUMP 289
99 IRSET=0 DUMP 290
ICON=0 DUMP 291
Jb=NALOFT
JT=1
JB=JB

C

WRITE OUT A DUMP SUMMARY

C

NOW BEGIN THE BLOCKING SORT

C

CLASSIFY THE JIN PARTICLE AS BLANK, TO BE DUMPED OR NOT TO BE DUMPED

C

DUMP

98 IF(JICON1=3J131332
30 GT 31(140130013142
32 GO TO 294(1501500150013142
50 IF(JICON13301332
53 GT 31(1401500150013142
55 GO TO 7914(1501500150013142
40 IF(JICON1=END1111100142142
50 IF(JICON1=END111142142142

C

31 BLANK NOT TO BE DUMPED

C

32 IF(JICON14225001424
34 GT 31(1401500150013142
42 IF(JICON142142142142

C

100 TO BE DUMPED

C

100 IF(JICON14254249000

C

ICON=0 FIRST PASS

C

ICON=1 BOTTOM LOOP

C

ICON=1 TOP LOOP

C

YOU MOVE THE JQ-IN LINE TO THE BLANK LINE (JBL)

C

900 XP(JBL1)*XP(JB)
YP(JBL1)=YP(JB)
ZP(JBL1)=ZP(JB)
TP(JBL1)=TP(JB)
PS(JBL1)=PS(JB)
FMAS(JBL1)=FMAS(JB)
JT=JT+1
FMAS(JBL1)=0

C

901 JBL=JB
ICON=1
JB=JB-1
GO TO 902

C

STORE THE JQ-IN PARTICLE IN TEMPORARY STORAGE AND SET JQUC=1 TO

C

INDICATE THAT IT MUST BE PUT BACK INTO THE PARTICLE ARRAYS AT THE

C

END OF THIS DUMP OPERATION

C

904 XP=XP(JB)
YPT=YP(JB)
ZPT=ZP(JB)
TP=TP(JB)
PS=PS(JB)
FMAS=FMAS(JB)
INSET=1
FMAS(JB)=0

GO TO 903

C

903 JT=JT+1
J=JT
IF(JT=JTEST)98,98,110
424  J=J-1
JB=JH-1
GO TO 1104
C
C 421 MOVE THE JT-TH LINE TO THE BLANK LINE (JBL)
421 XP(JBL)=XP(JT)
YP(JBL)=YP(JT)
ZP(JBL)=ZP(JT)
TP(JBL)=TP(JT)
T'S(JBL)=T'S(JT)
FNAS(JBL)=FNAS(JT)
422 JBL=JT
423 I<On-1
1104 IF(JB-JTest)1104,1105,1106
1105 JBL=JB
C 110 IS THE TEMPORARY STORAGE LOADED, YES TO 1101
110 IF(JRST=1111,1122,1101
C
C 1101 REPLACE THE TEMPORARILY STORED PARTICLE IN THE BLANK LINE (JBL)
1101 XP(JBL)=XP
YP(JBL)=YP
ZP(JBL)=ZP
TP(JBL)=TP
T'S(JBL)=T'S
FNAS(JBL)=FNAS
1102 CONTINUE
C
C RESIZE KEYS OF PARTICLES BEFORE DUMPING OR PRINTING OR
C DUMPING THE
GO 191 JBL=JTest
191 IF(FNAS(J))1101,111,111
101 FNAS(J)=FNAS(J)
111 IF(1P(J))121,131,131
121 TP(J)=1P(J)
131 CONTINUE
C
C DO NOT DUMP THE SELECTED DESCRIPTIONS
C 50 IF THE OUTPUT FILE IS TO BE WRITTEN, FIRST SELECT AND
C WRITE AN APPROPRIATE TITLE
50 IF(NAP=ISUUT)92,92,92
C
C IF THE PRINTING OF LAST PARTICLE DESCRIPTIONS IS TO BE SUPPRESSED,
GO TO 94
51 IF(N=IO)99,99,99
WRITE (ISUUT,JTest)
WRITE (ISUUT,J)
GO TO 101,101,101,101,101,101
211 WRITE (ISUUT,9)
GO TO 516
513 WRITE (ISUUT,6)
GO TO 516
514 WRITE (ISUUT,7)
GO TO 516
515 WRITE (19,118)
516 WRITE (19,111) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
52 WRITE (INTAP) JTEST
53 GO TO 54
55 WRITE (INTAP) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
56 GO TO 54
57 WRITE (INTAP) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
58 IF (IC(6)-1) 54,521,521
59 WRITE (INTAP) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
60 GO TO 54
61 WRITE (INTAP) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
62 GO TO 54
63 WRITE (INTAP) (XP(J),YP(J),ZP(J),TP(J),PS(J),FMAS(J),J=1,JTEST)
64 C
65 C 54 ADD THE NUMBER OF LINES JUST DUMPED TO THE NUMBER OF LINES
66 C PREVIOUSLY AND THEN ZERO OUT THE TOP OF THE LINES JUST DUMPED TO
67 C AVOID DOUBLE COUNTING
68 NFREE=NFREE-JTEST
69 DU = JTEST
70 FMAS(J)=0
71 C
72 IF (NFREE-N) 151,152,152
73 C
74 C 152 ARE THERE NOT ENOUGH CONTIGUOUS BLANK LINES IN THE TOP OF THE
75 C PARTICLE ARRAY TO RELIEVE THE N PARTICLES THAT ARE WAITING TO BE
76 C KEPT IN THE YES TO 60
77 152 IF (FIN-JTEST) 60,60,154
78 C
79 C 154 CONSOLIDATE N BLANK LINES INTO THE TOP OF THE PARTICLE ARRAY
80 JFR=JFR+1
81 XP=XP+1
82 IF (FMAS(J)) 57,56,57
83 C
84 C 57 A PARTICLE MUST BE MOVED DOWN
85 57 JFR=JFR-1
86 IF (FMAS(J)) 58,58,58
87 IF (JP(XFR)=JTEST) 90,90,121
88 C
89 C MOVE THE PARTICLE
90 XP(JFR)=XP(J)
91 YP(JFR)=YP(J)
92 ZP(JFR)=ZP(J)
93 TP(JFR)=TP(J)
94 PS(JFR)=PS(J)
95 FMAS(JFR)=FMAS(J)
96 FMAS(J)=0
97 CONTINUE
98 RETURN
99 END
100 C
101 C
102 C
103 C
104 C
105 C
106 C
107 C
108 C
109 C
110 C
111 C
112 C
113 C
114 C
115 C
116 C
117 C
118 C
119 C
120 C
121 C
122 C
123 C
124 C
125 C
126 C
127 C
128 C
129 C
130 C
131 C
132 C
133 C
134 C
135 C
136 C
137 C
138 C
139 C
140 C
141 C
142 C
143 C
144 C
145 C
146 C
147 C
148 C
149 C
150 C
151 C
152 C
153 C
154 C
155 C
156 C
157 C
158 C
159 C
160 C
161 C
162 C
163 C
164 C
165 C
166 C
167 C
168 C
169 C
170 C
171 C
172 C
173 C
174 C
175 C
176 C
177 C
178 C
179 C
180 C
181 C
182 C
183 C
184 C
185 C
186 C
187 C
188 C
189 C
190 C
191 C
192 C
193 C
194 C
195 C
196 C
197 C
198 C
199 C
200 C
201 C
202 C
203 C
204 C
205 C
206 C
207 C
208 C
209 C
210 C
211 C
212 C
213 C
214 C
215 C
216 C
217 C
218 C
219 C
220 C
221 C
222 C
223 C
224 C
225 C
226 C
227 C
228 C
229 C
230 C
231 C
232 C
233 C
234 C
235 C
236 C
237 C
238 C
239 C
240 C
241 C
242 C
243 C
244 C
245 C
246 C
247 C
248 C
249 C
250 C
251 C
252 C
253 C
254 C
255 C
256 C
257 C
258 C
259 C
260 C
261 C
262 C
263 C
264 C
265 C
266 C
267 C
268 C
269 C
270 C
271 C
272 C
273 C
274 C
275 C
276 C
277 C
278 C
279 C
280 C
281 C
282 C
283 C
284 C
285 C
286 C
287 C
288 C
289 C
290 C
291 C
292 C
293 C
294 C
295 C
296 C
297 C
298 C
299 C
300 C
301 C
302 C
303 C
304 C
305 C
306 C
307 C
308 C
309 C
310 C
311 C
312 C
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
349 C
350 C
351 C
352 C
353 C
354 C
355 C
356 C
357 C
358 C
359 C
360 C
361 C
362 C
363 C
364 C
365 C
366 C
367 C
368 C
369 C
370 C
371 C
372 C
373 C
374 C
375 C
376 C
377 C
378 C
379 C
380 C
381 C
382 C
383 C
384 C
385 C
386 C
387 C
388 C
389 C
390 C
391 C
392 C
393 C
394 C
395 C
396 C
397 C
398 C
399 C
400 C
401 C
402 C
403 C
404 C
405 C
406 C
407 C
408 C
409 C
410 C
411 C
412 C
413 C
414 C
415 C
416 C
417 C
418 C
419 C
420 C
421 C
422 C
423 C
424 C
425 C
426 C
427 C
428 C
429 C
430 C
431 C
432 C
433 C
434 C
435 C
436 C
437 C
438 C
439 C
440 C
441 C
442 C
443 C
444 C
445 C
446 C
447 C
448 C
449 C
450 C
451 C
452 C
453 C
454 C
454 *
**IBM IC**

**C**

11 OCT 69

**C**

<table>
<thead>
<tr>
<th>COMMON /SET1/</th>
<th>KUTO 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>U1AM, U2TU, I4ISE, IEXC, ISIN, ISOUT</td>
</tr>
<tr>
<td>2</td>
<td>SU, S2PN, S3AM, TNE, TIP1, TIP2</td>
</tr>
<tr>
<td>3</td>
<td>I6PN</td>
</tr>
<tr>
<td>4</td>
<td>WAY, RMIN, IDISTK, SPARK1, SPARK2, SPARK</td>
</tr>
<tr>
<td>5</td>
<td>SPARK4, SPARK6, SPARK7, SPARK8, SPARK9</td>
</tr>
<tr>
<td>DIMENSION DEUTL(12),MAY(40)</td>
<td>KUTO 16</td>
</tr>
</tbody>
</table>

**C**

<table>
<thead>
<tr>
<th>COMMON /SET2/</th>
<th>KUTO 18</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>S2, S3UL, S4UT, S5LL, S6UL, S7LL, SBLL, SBYLL</td>
</tr>
<tr>
<td>2</td>
<td>SBUL, S7LU, TALL, TAYL, TRYLL, TSYLL, TXG</td>
</tr>
<tr>
<td>3</td>
<td>TBUL, TBULU, TBULP, TBULP1, TBULP2, TBLIM, JBLIM</td>
</tr>
<tr>
<td>4</td>
<td>TLIM, IL, JL, KA, XP, YP</td>
</tr>
<tr>
<td>5</td>
<td>ZP, FMAT, IP, FS, VX, YV</td>
</tr>
<tr>
<td>6</td>
<td>YC, IL, JL, IBAU, WGSTN, INTRA1</td>
</tr>
<tr>
<td>7</td>
<td>IALL, IALLU, IALLP, IALLP1, IPHU1, IPAR1, JPLU1</td>
</tr>
<tr>
<td>8</td>
<td>JUPU1, JUPU2, JUPU3, LENIM, ENDIM, IC, IEMAX</td>
</tr>
<tr>
<td>9</td>
<td>JUPU4, JUPU5, JUPU6, NQ</td>
</tr>
<tr>
<td>10</td>
<td>NAIOP, JINCL, CRMAXX, INFACE, NCL</td>
</tr>
<tr>
<td>11</td>
<td>CRMAX, CROUT, NCR1YP, DZ</td>
</tr>
<tr>
<td>12</td>
<td>UV, ON, CO, NUCIR, DULC, ATEMP</td>
</tr>
<tr>
<td>13</td>
<td>MHU, NA, IN, UTMAIL, FIKU</td>
</tr>
<tr>
<td></td>
<td>KUTO 35</td>
</tr>
<tr>
<td></td>
<td>RUPAI</td>
</tr>
</tbody>
</table>

**C**

| DIMENSION TPPLM(44,4), TNTAR(4), ITPLM(3,4) | KUTO 30 |
| DIMENSION ST1(120) | SUBS1(400), TIC(16) | KUTO 37 |
| DIMENSION AP(2,00) | TyP(2,00) | ZP(2,00) | PMAT(2,00) | KUTO 28 |
| DIMENSION TP(2,00) | HAP(2,00) | ATTEM(2,00) | KUTO 29 |
| DIMENSION XAT1(1000) | V1(1,500) | V2(2,500) | L1(70) | KUTO 41 |
| DIMENSION JL(1,00) | IBAU(1,00) | WGSTN(1,00) | KUTO 40 |
| DIMENSION WGTNN(1,00) | QSTH(1,00) | NQ(1,00) | C1(6) | KUTO 43 |
| DIMENSION CRMAX(6) | CRMAX(6) | CRMAX(6) | CRMAX(6) | KUTO 44 |
| DIMENSION CRMAX(6) | CRMAX(6) | CRMAX(6) | KUTO 43 |

**C**

| FORMATT(33NU, TPULU OR TUPU LARGU FOR PROGRAM) | KUTO 47 |
| FORMATT(33NU, INCORRECT TUPU TABLE OF CONTIN) | KUTO 50 |
| LUPU FORMATT(1,10U10) | KUTO 52 |

**C**

| 11I=1TOPNL1*Lb) | KUTO 54 |
| J=1TOPNL1*Lb) | KUTO 55 |
| K=1TOPNL1*Lb) | KUTO 56 |
| ITPUA=TOPNL1(4*Lb) | KUTO 60 |
DATA T, L
DOUT
ROU
D1
}

READ (1, 2, 3, 4, 5, 6, 7, 8, 9)
D1
GO TO 10
RETURN
END
$IBFTC  LNK5   LISTDECK*M942
SUBROUTINE  LNK5   LNK5
              LNK5
              LNK5
C               LNK5
C               LNK5
C               LNK5
C               LNK5
C               LNK5
C               LNK5
C **************************** TAPE IDENTIFICATIONS AND ASSIGNMENTS *******
C NAME   CONTENT
C IMUPU   TOPOGRAPHIC HEIGHT DATA TAPE
C IUPU    PARTICLES ALOFT BUT BEYOND IN CORE TOPOGRAPHY
C IUPU2   PARTICLES ALOFT BUT BEYOND IN CORE WIND FIELD
C IPARIN  PARTICLES ALOFT AND TO BE PROCESSED
C IPAROU  PARTICLES ALOFT WAITING NEAT TIME PERIOD WINDS
C IPUO    SYSTEM (BCD) OUTPUT TAPE
C IUPU2   SYSTEM INPUT TAPE
C
C **************************** PROGRAM GLOSSARY *******
C ATMPH(J)  ATMOSPHERIC VISCOUSITY IN THE J-TH STRATUM
C AXZAXZ2  OUTPUT ARGUMENTS OF LOCAL CIRCULATION SYSTEM
C CUSD  XYZ AND Z COMPONENTS OF WIND AT THE
C POSITION OF THE J-TH PARTICLE
C BLANK     BLANK LITERAL
C UOUTHIT(K) ALTITUDE OF BOTTOM OF KTH WIND DATA BLOCK
C BZ      LENGTH OF THE SIDE OF THE SQUARE CLOUD SUBDIVISION
C JUNS AT THE TIME OF THEIR DEFINITION
C CORE21   SUBROUTINE SEA BREEZE CIRCULATION MODEL
C CIMIN    TIME UNTIL INTERSECTION WITH THE FIRST LOCAL
C WIND SYSTEM
C CIMIP(J)  THE LOCAL CIRCULATION TYPE OF THE JTH SYSTEM
C 1 MOUNTAIN WIND
C 2 RIDGE WIND
C 3 SEA BREEZE 1
C 4 SEA BREEZE 2
C 5 NOT ASSIGNED
C CROT(J)   CLOUD RISE IDENTIFICATION
C CRMINX(J) SMALLEST X COORDINATE OF THE JTH LOCAL SYSTEM
C CRMAXX(J) LARGEST X COORDINATE OF THE JTH LOCAL SYSTEM
C CRMINY(J) SMALLEST Y COORDINATE OF THE JTH LOCAL SYSTEM
C CRMAXY(J) LARGEST Y COORDINATE OF THE JTH LOCAL SYSTEM
C CRMAXZ   HEIGHT OF TOP SURFACE OF THE JTH LOCAL CIRCULATION SYSTEM
C
C CRMINX  CRMAXX  CRMINY  CRMAXY  CRMAXZ
C 1 2 3 4 5
C SYSTEM CELLS
C UOXJ(J)  INITIAL CONDITIONS (INPUT) IDENTIFICATION
C UENTI  TOPOGRAPHIC TAPE IDENTIFICATION LITERAL (IMUPU)
C As READ FROM TAPE
C UENTI  PARTICLES ALOFT INPUT TAPE IDENTIFICATION
C LITERAL AS READ FROM TAPE
C UINC  TIME INCREMENT FOR USE IN TRANSPORT WITHIN LOCALS
C CIRCULATION SYSTEM CELLS
C UTCMAC  TIME INCREMENT FOR USE IN TRANSPORT WITHIN THE
C MACRO FIELD BUT BELOW MAXIMUM TOPOGRAPHIC HEIGHTS
C CUTF   CANNED COPY OF PARTICLES ALOFT INPUT TAPE
C IDENTIFICATION LITERAL (IPARIN)
C DUMP   SUBROUTINE SELECTS AND DUMPS ONE OR MORE
C SETS OF PARTICLE DESCRIPTIONS UNTIL TEMPORARY, OR

155
INTERMEDIATE OUTPUT TAPE** LNK5 81
TIME UP TO WHICH THE CURRENT WIND FIELD DESCRIPTION IS ASSUMED TO BE VALID LNK5 82
A SMALL NUMBER LNK5 83
SUBROUTINE ERRORS AND/OR STOP LNK5 84
SUBROUTINE PARTICLE SETTLING RATE COMPUTATION LNK5 85
MASS PER UNIT AREA (INK) ASSOCIATED WITH THE LNK5 86
J-TH CLOUD SUBDIVISION (PARTICLE) AT THE TIME OF LNK5 87
ITS DEFINITION LNK5 88
A PRECOMPUTED CONSTANT INPUT FOR SUBROUTINE LNK5 89
FALKAT LNK5 90
OUTPUT ARGUMENT OF FALKAT (INK) A POSITIVE LNK5 91
QUANTITY LNK5 92
FISSION YIELD LNK5 93
SUBROUTINE: GETS HALFWIND VECTOR AT PARTICLE. LNK5 94
POSITION LNK5 95
OUTPUT ARGUMENT OF SUBROUTINE HEIGHT LNK5 96
SUBROUTINE RETRIEVES TOPU HEIGHT AT XY LNK5 97
HEIGHT OF BOUND RELATIVE TO SURFACE HEIGHT LNK5 98
TEMPORARY STORAGE FOR STANDARD ATMOSPHERIC LNK5 99
STRATUM BUTTON HEIGHT LNK5 100
HEIGHT TOPU HEIGHT ANYWHERE ON THE TOPU TAPE LNK5 101
CANNED COPY OF TOPOGRAPHY TAPE IDENTIFICATION LNK5 102
LITERAL (HTOPU) LNK5 103
BASE ADDRESS OF KTH WIND DATA BLOCK IN ARRAYS LNK5 104
VX, VY, AND VZ LNK5 105
INITIALIZE BYPASS FLAG LNK5 106
CONTROL VARIABLES INTERPRETATIONS AS FOLLOWS LNK5 107
SUPPRESS TOPU TAPE USAGE IF = 1 LNK5 108
SUPPRESS OFF TOPU TAPE USAGE IF = 1 LNK5 109
SUPPRESS OUT OF WIND FIELD TAPE USAGE IF = 1 LNK5 110
SUPPRESS PARTICLES ATUGHT SECONDARY MEMORY IF = 1 LNK5 111
SUPPRESS ALL TAPE SECONDARY MEMORY LNK5 112
TRANSPORT TRACE CONTROL. 0 = NO TRACES LNK5 113
1 = A PARTIAL TRACE, 2 = A MORE COMPLETE TRACELNK5 114
PRINT WIND FIELD IF = 1 LNK5 115
SUPPRESS PRINT AND WRITE OF LOST PARTICLES IF = 1 LNK5 116
EXECUTIVE CONTROL WORD TO CONTROL BRANCHING LNK5 117
BETWEEN CHAIN LINK SUBROUTINES LNK5 118
INDEX OF THE LAST LINE IN THE PARTICLE ARRAYS LNK5 119
INDEX FOR THE TOPU DATA BLOCK CURRENTLY LNK5 120
CORE LNK5 121
LIMITS ON INDICES OF WIND BLOCK DATA SET K LNK5 122
ASSIGNED GO TO VARIABLE FOR USE IN DEALING WITH LNK5 123
EXCESSIVELY SMALL PARTICLE MOVEMENTS ARISING LNK5 124
DURING TRANSION LNK5 125
INDICATOR OF TYPE OF BOUNDING PLANE ENCOUNTERED LNK5 126
DURING TRANSION. 1=X-BOUNDARY, 2=Y-BOUNDARY, LNK5 127
3=Z-BOUNDARY, 4=TIME BOUNDARY, 5=LOCAL CIRC- LNK5 128
ULATION CELL BOUNDARY LNK5 129
NUMBER OF THE SOURCE STATEMENT NEAREST TO WHERE LNK5 130
AN ERROR CONDITION WAS SENSED LNK5 131
THAT IS TO BE PROCESSED BY THE TRANSPORT LOOP LNK5 132
ASSIGNED GO TO VARIABLE FOR USE IN CODE SELECT- LNK5 133
ION REGARDING USE OF PLANAR OR PIECEWISE PLANAR LNK5 119
C
C TOPOGRAPHIC DESCRIPTION.
C LNK5 120
C
C ITOPLM(J,J)
C NUMBER OF CELLS IN THE X DIRECTION FOR THE
C REGULAR GRID SECTION OF THE JTH TOPU DATA BLOCK.
C LNK5 121
C
C ITOPLM(2,J)
C SEE ITOPLM(1,J) BUT FOR Y DIRECTION
C LNK5 122
C
C ITOPLM(3,J)
C NUMBER OF ENTRIES IN THE ONE-DIMENSIONAL SUB-
C ARRAY TOPOGRAPHIC DATA ARRAY FOR THE JTH
C LNK5 123
C
C DATA BLOCK.
C LNK5 124
C
C ITT
C SEE IT.
C LNK5 125
C
C JDONE
C IF 1, INDICATES THAT THE ONLY PARTICLES THAT RE-LNK5 126
C MAIN TO HAVE THEIR TRANSPORT COMPLETED ARE
C CURRENTLY IN CORE MEMORY. ALL ARE ON THE TIME
C BOUNDARY TAPE.
C LNK5 127
C
C JFTOPU
C INDICATOR OF TOPU TAPE FILE POSITION.
C LNK5 128
C
C THIS WORD RECORDS THE NUMBER OF THE FILE
C WHICH A READ COMMAND WOULD DRING IN NEXT
C LNK5 129
C
C JJ
C ITOPLM(2,J) FOR THE TOPU DATA BLOCK CURRENTLY IN
C CORE.
C LNK5 130
C
C JL(K)
C LNK5 131
C
C JLIM
C LNK5 132
C
C JTEST
C TEMPORARY STORAGE.
C LNK5 133
C
C JTEST1
C TEMPORARY STORAGE.
C LNK5 134
C
C +1
C THE TIME BOUNDARY TAPE IS IN USE.
C LNK5 135
C
C 0
C NO TIME BOUNDARY PARTICLES ARE ON TAPE.
C LNK5 136
C
C -1
C LOOP WITHOUT FIRST READING MORE PARTICLES FROM
C THE MAIN TRANSPORTABLE PARTICLES.
C LNK5 137
C
C REMAINING IN CORE ARE TRANSPORTED JUST AFTER A
C NEW WIND FIELD HAS BEEN COMPUTED.
C LNK5 138
C
C JTEST2
C TEMPORARY STORAGE.
C LNK5 139
C
C JTIME1
C INDICATES THE PRESENCE OF PARTICLES ON THE TIME
C BOUNDARY TAPE.
C LNK5 140
C
C +1
C INDICATES THAT THE TIME BOUNDARY TAPE IS IN USE.
C LNK5 141
C
C 0
C Particles in core, -1 is used to cause an
C entrance to the main loop without first
C loop without first reading more particles from
C the main transportable particles.
C LNK5 142
C
C READING MORE PARTICLES IN SO AS TO TRANSPORT
C ALL TRANSPORTABLE PARTICLES THAT REMAIN IN
C MEMORY JUST AFTER A NEW WIND FIELD IS COMPUTED.
C LNK5 143
C
C JTOPJ
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK5 144
C LNK6 145
C
C JTOP1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 146
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 147
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 148
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 149
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 150
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 151
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 152
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 153
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 154
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 155
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 156
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 157
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 158
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 159
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 160
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 161
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 162
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 163
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 164
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 165
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 166
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 167
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 168
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 169
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 170
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 171
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 172
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 173
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 174
C
C JN
C USED BY LINK7 AND SUBROUTINES AS AN INDEX OF THE
C WIND STRATUM CONTAINING A PARTICLE.
C LNK6 175
C
C JN1
C INDEX OF MIDDLE WIND LAYER, SET BY MAIN OR
C LNK6 176
NUMBER OF PARTICLE DESCRIPTIONS ABOUT TO BE READ. LNK5 177
FROM THE CURRENT BLOCK OF THE PARTICLES INPUT LNK5 178
TAPE LNK5 179

NUMBER OF ATMOSPHERIC LEVELS DESCRIBED IN THE LNK5 180
TABLES OF STANDARD ATMOSPHERIC DEGREE AND LNK5 181
VISCOITY LNK5 182

SIZE OF ARRAY FOR PARTICLES ALWFT LNK5 183
NLBUCK LNK5 184
NUMBER OF BLOCKS OF TOPU DATA ON TOPU TAPE LNK5 185
NBMAX LNK5 186
SIZE LIMIT ON DUMP BLOCK LNK5 187
NCIR LNK5 188
TEMPORARY STORAGE FOR NCRTYP(I) LNK5 189
NCL SPARE LNK5 189
NCRTYP(I) LNK5 190
LOCAL CIRCULATION TYPE IDENTIFIER OF THE J-TH LNK5 191
LOCAL SYSTEM, 1 IS MANHATTAN RFD LNK5 192
NCRTYP(I) OTHERS ARE UNASSIGNED. LNK5 193
NFREE LNK5 194
NUMBER OF EMPTY LINES IN PARTICLES ALWFT LIST LNK5 195
NO LNK5 196
NUMBER OF PARTICLES UNBOUND. (IN CORE) LNK5 197
NINTAR(I) LNK5 198
NUMBER OF PARTICLES OVER TOPU AREA J AND IN LNK5 199
THE OUT-OF-TOPU BUFFER LNK5 200
NLUCIN LNK5 201
THE NUMBER OF LOCAL CIRCULATION SYSTEMS IN USE LNK5 202
NINTAR(I) LNK5 203
NUMBER OF PARTICLES THAT DRIFTED BEYOND LIMITS LNK5 204
NPS LNK5 205
NUMBER OF PARTICLE SIZE RANGES DESCRIBED IN THE LNK5 206
PARTICLE SIZE DISTRIBUTION ARRAYS LNK5 207
NSP SPARE LNK5 208
NSSTRAT LNK5 209
MAXIMUM NUMBER OF ATMOSPHERIC STRATUM DESCRIPT- LNK5 210
ION THAT THE PROGRAM CAN ACCOMMODATE LNK5 211
NTI LNK5 212
NUMBER OF PARTICLES AT TIME BARRIER. (IN CORE) LNK5 213
NTU LNK5 214
NUMBER OF PARTICLES AT TOPU BARRIER (IN CORE) LNK5 215
NUL INTEGER ZERO LNK5 216
NW LNK5 217
NUMBER OF PARTICLES AT WIND BARRIER (IN CORE) LNK5 218
NWTAB ZERED OUT TO VARIABLES FOR USE IN SPEEDING UP LNK5 219
THE COMPUTATION OF TIME OF FLIGHT TO LOCAL CIRC-LINKS LNK5 220
JPAT LNK5 221
PARTICLE SYSTEM CELL BOUNDARIES. LNK5 222
KUUT LNK5 223
CAVITY LUMP OF UNBOUND PARTICLES TAPE IDENT- LNK5 224
IFIER LITERAL (IPUT) LNK5 225
KRGMN LNK5 226
CONTAINS PROGRAM NAME IN BCD LNK5 227
PS(I) LNK5 228
MICRO-RANGE PARTICLE SIZE (MICRONS) OF THE J-TH LNK5 229
PARTICLE SIZE RANGE DESCRIBED IN THE PARTICLE LNK5 230
SIZE DISTRIBUTION ARRAYS LNK5 231
KSEID LNK5 232
PARTICLE SET EXPANSION PROGRAM OF INTERFACE LNK5 233
LPSIZE LNK5 234
PROGRAM RUN IDENTIFICATION LNK5 235
PARTICLE SIZE OF CURRENT LNK5 236
PARTICLE LNK5 237
XADMAX LNK5 238
MAXIMUM CLOUD RADIUS DURING CLOUD RISE LNK5 239
KDTOPU LNK5 240
SUBROUTINE READS IN A TOPU DATA BLOCK LNK5 241
KGNMAX LNK5 242
SUBROUTINE KIDGE WIND CIRCULATION MODEL LNK5 243
KHNUT(I) LNK5 244
ATMOSPHERIC DENSITY IN THE J-TH TABULATED LNK5 245
PARTICLE LNK5 246
STRATUM LNK5 247
KOPART LNK5 248
FALLOUT PARTICLE DENSITY (MKG) LNK5 249
RTST LNK5 250
TEMPORARY STORAGE LNK5 251
SUMA LNK5 252
STANDARD DEVIATION OF PARTICLE SIZE DISTRIBUTION LNK5 253
SLDTAP LNK5 254
PARTICLE SOLIDIFICATION TEMPERATURE (K) LNK5 255
SSAM LNK5 256
MSS OF CONDENSED PHASE MATERIAL AT SPECIFICAT- LNK5 257
ION TIME LNK5 258
TC LNK5 259
TEMPORARY STORAGE LNK5 260
TGZ LNK5 261
TIME OF DETONATION LNK5 262
TIDI LNK5 263
TRANSFER IDENTITY LNK5 264
TIMES OF FLIGHT TO THE FIRST X+Y+Z MACRO LNK5 265
TIA, TIB, TIC LNK5 266
LKP3
CELL BOUNDING PLANES
TIME SPECIFIED FOR THE TERMINATION OF TRANSPORT
TEMPORARY STORAGE FOR LARGEST OF (TA1,TA2) AND
TEMPORARY STORAGE FOR SMALLEST OF (TA1,TA2) AND
TIME OF SOLIDIFICATION
TOPOGRAPHY IDENTIFIER
FLOATING POINT TOPOGRAPHIC DATA ARRAY
MIN. X-COORDINATE FOR J-TH TOPU DATA BLOCK
MIN. Y-COORDINATE FOR J-TH TOPU DATA BLOCK
INITIAL (CELLULAR) UNIG INTERVAL FOR J-TH TOPU
DATA BLOCK
HIGHEST TOPU NUMBER IN THE J-TH TOPU DATA BLOCK
PARTICLE TIME STATUS
TIME OF FLIGHT TO POINT OF ENTRANCE TO CURRENT
LOCAL WIND CELL
HIGHEST TOPU ELEVATION IN THE IN CORE DATA
TOTAL YIELD
TIME OF FLIGHT TO TIME BOUNDARY
LOWER AND UPPER X AND Y COORDINATE LIMITS FOR
FING ES OF FLIGHT TO THE TWO X-PLANES BOUNDING THE
CURRENT LOCAL CIRCULATION CELL
THE AREA ACCOUNTED FOR ON THE TOPOGRAPHY TAP
SEE TA1 BUT FOR Y-PLANES
TEMPORARY STORAGE FOR X AND Y PARTICLE VELOCITY
VERTICAL VELOCITY OF PARTICLE
TEMPORARY STORAGE FOR PRECEDING VALUE OF VPZ
WIND VELOCITY COMPONENTS
GRID INTERVAL OF KTH WIND DATA BLOCK
WIND FILL DESCRIPTION IDENTIFICATION
HORIZONTAL LIMITS OF KTH WIND DATA BLOCK
HORIZONTAL LIMITS OF KTH WIND DATA BLOCK
TEMPORARY STORAGE FOR PARTICLE COORDINATES
LOWER AND UPPER X COORDINATES OF MAUCH CELL CON
TAINING PARTICLE
COORDINATES OF GROUND ZERO
AXIS AND Z INCREMENTAL DISTANCES FOR USE IN
CONSTANT TIME INTERVAL STEPPING BELOW MAXIMUM
TOPU HEIGHT
PARTICLE POSITION COORDINATES
TEMPORARY STORAGE FOR X+Y+AND Z PARTICLE COORDINATIONS
STES
SEE XBL FOR Y COORDINATE

COMON /SET1/

1 DIAM DETAIL(12), IRIE, ISEL, JOIN, IOOT LINKS 281
2 SD SPARK SPARK IVE TMPI TIME 282
3 TEM ALU VRK X LINKS 283
4 YIY(40), KHA IN D JOIN SPARK SPARK2 JUUNE LINKS 284
5 SPARK4 SPARK5 SPARK6 SPARK7 SPARK8 SPARK9 LINKS 285

COMON /SET2/

1 S SUBSIB GRINT WXLW BXLW BYLL BYLL LINKS 290
2 AYLU XLL TALU TYY TYY XGZ LINKS 292

159
3  YGZ  NBGCK  HTPU  ITPO  ITPO  ILIM  JLIM  LNK 240
4  KLIN  JJ  KK  XP  YP  LNK 294
5  ZP  PHAS  TP  PO  VA  VY  LNK 492
6  VZ  IL  JL  IDADU  WGRNT  NOTMAT  LNK 590
7  WLX  WLY  WJR  WUR  DJHMT  IPHRN  LNK 297
8  JUPO  JIWIND  ITPO  IPST  IPARU  JJPC  LNK 249
9  WVIND  WLRK  WUTL  WUTL  WCML  WCML  LNK 697
1  JTPO  KLVST  NS  NY  NNI  NW  LNK 300
2  NALUFT  JTTX1  NMMAX  NFRE  N  NCL  LNK 301
3  CHMAXX  CRWMT  NRTYP  NTP  CRINAX  CRMNP  LNK 302
4  UT  SR  CR  CLUCIR  DILUC  ATMFR  LNK 303
5  RNU  YXG  TUL  DTULM  PR0U  CRMXX  LNK 304
6  RUPART  

**DIMENSION** TUMLM(44), INTLM(4), TUPLM(34)
**DIMENSION** SUTU(10), SUTU(40), IEC(10)
**DIMENSION** XPS(40), YPS(200), PPS(200), AETAP(200), RNU(200)
**DIMENSION** XPS(170), YPS(170), ZPST(70), XMUX(170)
**DIMENSION** WGRNT(70), WGRNT(70), WMUX(70), WLY(70)
**DIMENSION** CRWMT(5), CRWMT(9), CRWMT(9), CRWMT(9)
**DIMENSION** CRWMT(5), CRWMT(9), CRWMT(9), CRWMT(9)

**FORMAT** (25X El3.5, 5X El3.5, bXE1.5) LNK 25

**FORMAT** (12A6) LNK 201

//ZDX16HHIGHT
2X, 12A6, //25A, 49H***1AK

**F0R** FORMAT (Fl3X, 35HID~cNIFICATloN FROM FORMATT)
**FORMAT** (2L12.5) LNK 45
**FORMAT** (LS) LNK 5

**PROCEDURE** PROTECT THE REEL ON TAPE AND CORRECT TAPE AND PRESS START
**PROCEDURE** TPAGHAPY TAPE 18X12A6
**PROCEDURE** TPAGHAPY TAPE 18X12A6

**DATA** TPAGHAPY TAPE 18X12A6
**DATA** TPAGHAPY TAPE 18X12A6

**DATA** TPAGHAPY TAPE 18X12A6
**DATA** TPAGHAPY TAPE 18X12A6
**SUMMARY OF INPUT IDENTIFIERS AND INITIAL CONDITIONS**

- **TOPOGRAPHIC DATA**: Various topographical data inputs are defined, including grid dimensions, topographical features, and possibly geographic coordinates.
- **PARTICLE DATA**: Information on particles, such as their density and fallout concentration, is specified.
- **WIND DATA**: Wind data, including wind speed and direction, are provided.
- **OTHER PARAMETERS**: Additional parameters such as time steps, initial conditions, and system settings are outlined.

**INITIAL CONDITIONS**

- **JTOP1**: Set to 0, indicating the initial condition for topographical data.
- **NSTRAT**: Set to 70, probably indicating the number of stratification levels for the simulation.
- **NW**: Set to 0, possibly representing a counter or status flag.
- **NTO**: Same as **NW**, also set to 0.
- **NG**: Set to 0, indicating a status or counter.

**ANALYSIS AND MODIFICATIONS**

- **BYPASSING INITIALIZATION CODING**: A conditional statement (IF) is used to bypass initialization coding after the first pass, allowing for a more streamlined processing flow.

**DIMENSION LIMITS**

- **ILIM**, **JLIM**, and **KLIM** are set to limits on topographical arrays, ensuring that the data handling within the simulation is within manageable bounds.

**DEPARTMENTAL DEPARTMENT**

- **M E N T O F D E F E N S E F A L L O U T P R E D I C T I O N**
- **TRANSPORT**
- **TECHNICAL OPERATIONS RESEARCH**

**PREPARED BY**

- **4NC**
- **52X**
- **BURLINGTON, MASS.**

**FURTHER DETAILS**

- **Kg/M**
- **Particle Density of Fallout Particles**
- **Wind Data**
- **Topographical Data**

**CATEGORIES**

- **TOPOGRAPHIC DATA**
- **PARTICLE DATA**
- **WIND DATA**
- **OTHER PARAMETERS**

- **INITIAL CONDITIONS**
- **ANALYSIS AND MODIFICATIONS**
- **DIMENSION LIMITS**
- **DEPARTMENTAL DEPARTMENT**

**End of Document**
DO 2011 J=1,NALOFT
2011 FMAS(J)=0.0
C READ IDENTIFICATION FOR TRANSPORT
READ (ISIN*11,TID(J)+J=1,12)
C READ CONTROL DATA FOR TRANSPORT
C THESE CONTROL PARAMETERS ARE FOR USE AS SIMPLIFYING SWITCHES
READ (ISIN*21,TIC(J)+J=1,18)
READ (ISIN+31,TILMIN)
C READ IDENTIFICATION
READ IDENTIFICATION
C READ INPUT TAPES
IF (IC(1)-1)*150+151
150 REWIND IHTOPO
151 IF (IC(2)-1)*152+153
152 REWIND IHTOPO
153 IF (IC(3)-1)*154+155
154 REWIND IOWIND
155 IF (IC(4)-1)*156+157+158
156 CONTINUE
157 CONTINUE
158 CONTINUE
REIWIND IPARIN
REWIND IPARIN
C CHECK IDENTIFICATIONS ON IHTOPO AND PARTICLE INPUT TAPES
206 IF (IC(1)-1)*158+203+203
207 READ (IHTOPO)DENTT
RTST=AND(DENTT,CAPL(DTST))
IF (RTST) 207,204,204
C 208 WRONG TAPE AS IHTOPO
208 PRINT 14,IHTOPO
WRITE (LSOUT,14)IHTOPO
PRINT 15
REWIND IHTOPO
PAUSE
REWIND IHTOPO
GO TO 206
C 209 CONTINUE
209 CONTINUE
2010 PRINT 14,IPARIN
WRITE (LSOUT,14)IPARIN
PRINT 15
REWIND IPARIN
PAUSE
REWIND IPARIN
GO TO 207
2031 READ(IHTOPO)TXLL,IXLU,ITYLL,ITYLU,NBLCK
C 203 CONTINUE
204 CONTINUE
204 READ (IPARIN)DENTT
RTST=AND(DENTT,CAPL(DTST))
IF (RTST) 204,208,208
C 208 READ ARBITRARY 72 CHARACTER FIREBALL,CLAYD-RISE,AND PARTICLE
C ACTIVITY IDENTIFICATIONS FROM IPARIN
208 READ (IPARIN) FW,SSAM,SLDMP,TMOD,SIGMA,TW,HOB,NSP,XGZ,YGZ,TGZ,BZ
1 NCL,RADMAX
READ (IPARIN)(PSEID(J),J=1,12)
READ (IPARIN)(GRID(J),J=1,12)
READ (IPARIN)(DETID(J),J=1,12)

READ DENSITY OF fallout particles

READ PARTICLE SIZE MASS AND ACTIVITY DISTRIBUTIONS
READ (IPARIN)ROPART

COMPUTE constant FOR fall RATE calculations
FRUG=I.3 Joan7E-17*KUPART

READ ARBITRARY TUPU IDENTIFICATION
IF (ICN1-1.)L5991600180
1 DO 170 IP=1,12
170 READ (1,IP)TUPU

READ TUPU TABLE OF CONTENTS
READ (1,ITUPU)ITUPU
READ (1,ITUPU)ITUPU

FIND HIGHEST TUPU HEIGHT
HTOP=0.0
DO 170 IP=1,NUPBK
170 HTOP=MAX(HTOP,ITUPU(4,IP))/170
CONTINUE

READ FIRST TUPU DATA BLOCK
CALL ROPTUPU ()

205 PUT AN IDENTIFICATION ON THE TRANSPORT INTERMEDIATE OUTPUT TAPE
205 READ (1,IP)TUPU
WRITE (IP,TP)
1 RUPART=KG367G67K3475AAX
WRITE (IP,TP) ()TUPU(J)J=1,12)
WRITE (IP,TP) ()TUPU(J)J=1,12)
WRITE (IP,TP)
WRITE (IP,TP) ()G(J)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IP,TP)
WRITE (IPOUT) (BLANK,J=1,12)
GO TO 206
2054 WRITE (IPOUT) (IPOTID(J,J=1,12))
C C
C PRINT TRANSPORT OUTPUT HEADING
2056 WRITE (IPOUT,23)
WRITE (IPOUT,21) (DET1(J),J=1,12) (CKL1(J),J=1,12) (POL1(J),J=1,12)
WRITE (IPOUT,7)
WRITE (IPOUT,1) (NATK5(J),J=1,10)
WRITE (IPOUT,9) (ELIM)
WRITE (IPOUT,27) (UPART)
WRITE (IPOUT,29) (DEN1(RMP),RMP=K24:Y26:Z26:B26:N26)
WRITE (IPOUT,24)
IF (IC1-1) WRITE (IPOUT)
GO TO 2058
C 2058 WRITE (IPOUT,11) (IPORT)
GO TO 2058
2051 WRITE (IPOUT,16) (UPOT1(J),J=1,12)
WRITE (IPOUT,30) (DEN1(ALL),DEN1=ALL, NLEN=1)
WRITE (IPOUT,28)
WRITE (IPOUT,3) (WID(J),J=1,12)
WRITE (IPOUT,6)
HS=-111
DO 2057 J=1,10
WRITE (IPOUT) (HOR1TEM(J),RNUM(J))
2057 HS=HS+2010
C C
** C
C C
C 200 ANY MORE TIME INTERVALS TO BE PRINTED ALL NO TO 500
C 200 IF (T1<ENDT) DO 2000 20000
C C C
C 500 MAKE FINAL TRANSPORT PROGRAM OUTPUT AND COMMENTS
C 500 NNAKOF = TO CALL DUMP TO CLEAR OUT THE ENTIRE PARTICLE ARENA
C 500 C
500 NNAKOF
C CALL DUMP
C C
C * ARE ANY PARTICLES ON THE TIME BOUNDARY TAPE? YES TO 700
C * BY TEMPERATURE TIME INTERVAL
C 700 WRITE (IPOUT) (NPAR1, NPAR1=1,12)
C 700 WRITE (IPOUT) (IPORT1)
C 700 REWIND IPORT
WRITE (IPOUT,11)
702 READ (IPORT)
IF (VS, VS=171)
701 READ (IPORT) (APAR1,APAR1(1,1),APAR1(1,2),APAR1(1,3), APAR1(1,4),APAR1(1,5),APAR1(1,6),APAR1(1,7),APAR1(1,8),APAR1(1,9),APAR1(1,10),APAR1(1,11),APAR1(1,12))
WRITE (IPOUT,12) (APAR1(1,1),APAR1(1,2),APAR1(1,3), APAR1(1,4),APAR1(1,5),APAR1(1,6),APAR1(1,7),APAR1(1,8),APAR1(1,9),APAR1(1,10),APAR1(1,11),APAR1(1,12))
GO TO 702
C 501 WRITE A FINAL ZERO BLOCK COUNT AND EOF ON IPOUT
501 WRITE (IPOUT) (NPAR1)
C END FILE IPOUT
C REWIND IPOUT

164
WRITE (18,10)IPOUT
PRINT 18,IPOUT
PRINT 19,IPOUT
C
C 5010 SKIP OVER ANY UNUSED WIND DATA
C 5020 A CARD CONTAINING 'END OF WIND FIELD DATA' MUST MARK THE END OF
C 5030 THE WIND FIELD DATA DECK
C 5040 READ(ISIN,LISTST)
C 5050 RTST=AND(END#FD,COMPLRTST))
C 5060 IF(RTST)5010D0055010
C 800 PREPARE TO CALL OUTPUT PROCESSOR PROGRAM
C 801 İEXEC=2
C 802 RETURN
C
C 400 GET OR OTHERWISE PRODUCE THE REAL TIME INTERVAL'S WIND FIELD.
C 401 İTIL=0
C 402 İEXEC = 1
C 403 RETURN
C 404 END
SUBROUTINE LNK6
CALL MKWIND
RETURN
END
SUBROUTINE ROCIRS
COMMON /SET1/
CC
CC COMMON /SET2/
CC
COMMON /ROPMARKER
CC
DATA PROGRAM /S90D3IN9/
C *** ...........................................RDCI 61
C *** ...........................................RDCI 62
C
C READ DEFINING DATA FOR LOCAL CIRCULATION SYSTEMS
C
K=0
120  K=K+1
      READ (10,45) CKNAXX(K),CKMAYX(K),CKMAXX(K),CKMAYX(K),
      NCIR=NCRTYP(K)
      IF (NCIR) 122,100,125
122   IFERROR=122
         GO TO 7734
125   IF (NCIR-3) 120,120,124
124   IFERROR=124
         WRITE (100,2) NCIR
9734  CALL ERROR(PRGRM,IFERROR,ISOUT)
C 100  THIS IS THE NORMAL EXIT
C
C  NLUCIR  THE NUMBER OF LOCAL CIRCULATION SYSTEMS DEFINED FOR USE
C  IN TRANSPORTING PARTICLES
C
C  RETURN
100  NLUCIR = K-1
END
The subroutine MKWIN2 is described in detail in the technical operations research. It computes a horizontally and vertically variant wind. The description is based on the input from the system input tape. The variables used are as follows:

1. Control variables: E, T, which defines the time at which
2. The following data cease to be valid: ALPHA, which is a weighting factor to be applied to vertical distances, BETA, which is a weighting factor to be applied to horizontal distances, C, which specifies the number of nearest vectors to be used in calculating the wind vector at a point.

Two wind grid specifications are included: the X component and the Y component. Each vector is specified by its magnitude and direction.

The subroutine terminates when the subroutine begins to be used again. The vector interpolation function is used to calculate the wind vector at a point.

The subroutine Mkwin2 is used in the technical operations research.
**NADT**
INDEX OF THE NAD THAT CONTAINS THE ADDRESS OF THE V2
WHERE IS THE LARGEST OF NEAREST IN DATA POINTS
**NCODE**
IDENTIFICATION NUMBER FOR THE METHOD OF COMPUTATION TO BE USED IN TRANSLATING THE WIND DATA INTO THE WIND ARRAYS
**NN**
The number of nearest data vectors that the wind is based on
**T1**
TEMPORARY STORAGE
**WG2**
HALF OF GRID INTERVAL / X100.3
**XG**
X COORDINATE OF GRID POINT
**XLI**
AN X-DIRECTION LIMIT FOR TESTING THE COMPLETION OF A BOX IN THE WIND FIELD ARRAY
**YG**
Y COORDINATE OF GRID POINT
**YLIM**
SEE XLI, FOR THE Y-DIRECTION
**ZG**
Z COORDINATE OF GRID POINT

**PARAMETER SET**

<table>
<thead>
<tr>
<th>1</th>
<th>DIAM</th>
<th>DETID</th>
<th>HRI-E</th>
<th>IEJC</th>
<th>IEXEC</th>
<th>ILIN</th>
<th>ILOAD</th>
<th>AMAX</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>SD</td>
<td>SPAR</td>
<td>SSKT</td>
<td>TMC</td>
<td>TMP1</td>
<td>TMP2</td>
<td>AMAX</td>
<td>65</td>
<td>70</td>
</tr>
<tr>
<td>3</td>
<td>T2V</td>
<td>U</td>
<td>VPR</td>
<td>X</td>
<td>Z</td>
<td>AMAX</td>
<td>61</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>HY</td>
<td>X1IN</td>
<td>T11TR</td>
<td>SPAR1</td>
<td>SPAR2</td>
<td>SPAR3</td>
<td>AMAX</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>SPAR4</td>
<td>SPARS</td>
<td>SPAR6</td>
<td>SPAR7</td>
<td>SPAR8</td>
<td>SPAR9</td>
<td>AMAX</td>
<td>63</td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION DETID(12)**

**PARAMETER SET**

<table>
<thead>
<tr>
<th>1</th>
<th>J</th>
<th>JGRID</th>
<th>JHIT</th>
<th>J1X</th>
<th>JY</th>
<th>JZ</th>
<th>AMAX</th>
<th>67</th>
<th>70</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>BYL</td>
<td>TALL</td>
<td>TALL</td>
<td>TALL</td>
<td>TALL</td>
<td>TALL</td>
<td>AMAX</td>
<td>67</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>YG3</td>
<td>NOLCI</td>
<td>NOLC</td>
<td>NOLC</td>
<td>NOLC</td>
<td>NOLC</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>ALIN</td>
<td>II</td>
<td>II</td>
<td>II</td>
<td>II</td>
<td>II</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>CP</td>
<td>FIAX</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>TP</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>VL</td>
<td>IL</td>
<td>JL</td>
<td>JL</td>
<td>JL</td>
<td>JL</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>LL</td>
<td>LLLY</td>
<td>LLY</td>
<td>LLY</td>
<td>LLY</td>
<td>LLY</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>I11PZ</td>
<td>IN1ND</td>
<td>IN1ND</td>
<td>IN1ND</td>
<td>IN1ND</td>
<td>IN1ND</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>I1INC</td>
<td>I1NCR</td>
<td>I1NCR</td>
<td>I1NCR</td>
<td>I1NCR</td>
<td>I1NCR</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>I1P2</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>I1P3</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>X1OUT</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>NOLCF</td>
<td>JTIME1</td>
<td>JTIME1</td>
<td>JTIME1</td>
<td>JTIME1</td>
<td>JTIME1</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>CNAXY</td>
<td>CRHIT</td>
<td>CNV87</td>
<td>CNV87</td>
<td>CNV87</td>
<td>CNV87</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>C3</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>NA</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>K0</td>
<td>X1</td>
<td>X1</td>
<td>X1</td>
<td>X1</td>
<td>X1</td>
<td>AMAX</td>
<td>66</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>R2P</td>
<td>R2PART</td>
<td>AMAX</td>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION TOP(L1)(4X4)**

<table>
<thead>
<tr>
<th>1</th>
<th>MINTAN(4)</th>
<th>IVTOPA(3X4)</th>
<th>AMAX</th>
<th>101</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION XPZ(200)**

<table>
<thead>
<tr>
<th>1</th>
<th>XPZ(200)</th>
<th>AMAX</th>
<th>102</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION TPZ(200)**

<table>
<thead>
<tr>
<th>1</th>
<th>TPZ(200)</th>
<th>AMAX</th>
<th>103</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION VX1(150)**

<table>
<thead>
<tr>
<th>1</th>
<th>VX1(150)</th>
<th>AMAX</th>
<th>104</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION JL1(70)**

<table>
<thead>
<tr>
<th>1</th>
<th>JL1(70)</th>
<th>AMAX</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION GRAND(70)**

<table>
<thead>
<tr>
<th>1</th>
<th>GRAND(70)</th>
<th>AMAX</th>
<th>106</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION GY1(70)**

<table>
<thead>
<tr>
<th>1</th>
<th>GY1(70)</th>
<th>AMAX</th>
<th>107</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION CRHIT(6)**

<table>
<thead>
<tr>
<th>1</th>
<th>CRHIT(6)</th>
<th>AMAX</th>
<th>108</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PARAMETERS**

<table>
<thead>
<tr>
<th>1</th>
<th>TOP</th>
<th>AMAX</th>
<th>111</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION AT1(30)**

<table>
<thead>
<tr>
<th>1</th>
<th>AT1(30)</th>
<th>AMAX</th>
<th>112</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION AT2(30)**

<table>
<thead>
<tr>
<th>1</th>
<th>AT2(30)</th>
<th>AMAX</th>
<th>113</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**DIMENSION AT3(30)**

<table>
<thead>
<tr>
<th>1</th>
<th>AT3(30)</th>
<th>AMAX</th>
<th>114</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PARAMETERS**

<table>
<thead>
<tr>
<th>1</th>
<th>AT1</th>
<th>AMAX</th>
<th>115</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FORMAT**

<table>
<thead>
<tr>
<th>1</th>
<th>FORMAT(6X10,3)</th>
<th>AMAX</th>
<th>116</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>FORMAT(7X60)</td>
<td>AMAX</td>
<td>117</td>
</tr>
</tbody>
</table>

**IN ATMOSPHERE UP TO TIME FL/G1**
3 FORMAT(7X6HVECT0R7X22HMM12ONTAL COORDINATES 8X17HVECTOR COMPONENTS MKW1 120
1S'7X6HHEIGHT10X2HMX10X2HY10X2HVX10X2MVZ) MKW1 121
4 FORMAT(4X6F12.3) MKW1 122
5 FORMAT(62H INADEQUATE CONTROL DATA. COMPUTATION METHOD 1 WILL BE USED) MKW1 123
6 FORMAT(64H ENCOUNTERED TWO WIND GRID LAYER REQUEST FOR THE SAME ANGLE) MKW1 125
7 1LTITUDE) MKW1 126
8 FORMAT(514) MKW1 127
9 FORMAT(15HCOMPUTATION METHOD 14+17H IS NOT AVAILABLE) MKW1 128
10 FORMAT('10X26HREQUES TED GRID ARRANGEMENT/7X6H1IGHT7XBINTH INTERVAL22X3) MKW1 167
11 16HLIMITS/35X6HLLLX4HLLY8X4HJRX8X4HURY/Y/14X6F12.3) MKW1 130
12 FORMAT(16H WIND COMPONENTS) MKW1 131
13 FORMAT(1X*13EAST REST 0X16) MKW1 132
14 FORMAT(1X10F12.5) MKW1 133
15 FORMAT(7F6P LEVEL13 G18X6HMEAN AT 8F12 37H METERS) MKW1 134
16 FORMAT(15H AN HAS BEEN REDUCED TO 15) MKW1 135
17 FORMAT(6F12.3) MKW1 136
22 FORMAT(5E6P6.6) MKW1 137
23 FORMAT() 115H AN EXCESSIVE NUMBER OF SIGNIFICANT FIGURES ARE NOT AVAILABLE) MKW1 140
24 IN THE LEAST SQUARES CALCULATIONS. THE DATA POINTS APPROXIMATE A LINE IN) MKW1 141
25 AN E39M OR A PLANE. THE ALIGNED VECTORS METHOD IS USED FOR GRID POINT) MKW1 142
26 FORMAT() 10A ALL VECTORS LIE WITHIN THE SPECIFIED ALIGNMENT REGION) MKW1 144
27 R A RANDOM SELECTION OF 14+ 35 VECTORS ARE ALIGNED ALIGNED) MKW1 145
28 25 5X 15H FOR GRID POINT) MKW1 146
29 25 5X 5H(XY2)12X F12+3112X12X123113X) MKW1 147
30 25 FORMAT() 10A 1X6BALPHA = 1X455 1X11X7MULTA = 1X455 1X7METERS) MKW1 149
31 25 DATA PROGRAM/15+4WINXMTST615 7XMM12IND1+1 3OE+30 1350*101 3OE-30) MKW1 153
32 READ TISIN1)ENXTI&ALPHA&ETA) MKW1 154
33 ALPHANALPHA&ALPHA) MKW1 155
34 ETTA&ETA&ETAT) MKW1 156
35 READ SPECIFICATION OF DESIRED WIND ARRAYS MKW1 157
36 READ TISIN5)SINXKOE) MKW1 158
37 IF(NN)1294+204=2041 MKW1 159
38 CALL ERR22(PPROGR1RNK1JOUT) MKW1 160
39 CALL ERR22(PPROGR1RNK1JOUT) MKW1 161
40 CALL ERR22(PPROGR1RNK1JOUT) MKW1 162
41 CALL ERR22(PPROGR1RNK1JOUT) MKW1 163
42 CALL ERR22(PPROGR1RNK1JOUT) MKW1 164
43 CALL ERR22(PPROGR1RNK1JOUT) MKW1 165
44 CALL ERR22(PPROGR1RNK1JOUT) MKW1 166
45 CALL ERR22(PPROGR1RNK1JOUT) MKW1 167
46 CALL ERR22(PPROGR1RNK1JOUT) MKW1 168
47 CALL ERR22(PPROGR1RNK1JOUT) MKW1 169
48 CALL ERR22(PPROGR1RNK1JOUT) MKW1 170
49 CALL ERR22(PPROGR1RNK1JOUT) MKW1 171
50 CALL ERR22(PPROGR1RNK1JOUT) MKW1 172
51 CALL ERR22(PPROGR1RNK1JOUT) MKW1 173
52 CALL ERR22(PPROGR1RNK1JOUT) MKW1 174
53 CALL ERR22(PPROGR1RNK1JOUT) MKW1 175
54 CALL ERR22(PPROGR1RNK1JOUT) MKW1 176
55 CALL ERR22(PPROGR1RNK1JOUT) MKW1 177
56 CALL ERR22(PPROGR1RNK1JOUT) MKW1 178
57 CALL ERR22(PPROGR1RNK1JOUT) MKW1 179
1153 KS=1
HTST=OTHIT(J-1)
VXT=WGRINT(J-1)
VTY=WLLX(J-1)
VTZ=WLLY(J-1)
XST=WURX(J-1)
YST=WURY(J-1)
OTHIT(J-1)=OTHIT(J)
WGRINT(J-1)=WGRINT(J)
WURX(J-1)=WURX(J)
WURY(J-1)=WURY(J)
WGRINT(J)=VXT
WURX(J)=XST
WURY(J)=YST
1051 CONTINUE
1052 WRITE (ISOUT*6)
1053 IRROR=1052
G0 TO 7734
C
C 1055 COT OF THE REQUESTED LAYERS IS COMPLETE
C MAKE SURE THAT THERE IS SUFFICIENT SPACE FOR THE LAYERS.
1055 DO 1056 J=1,JTOPV
K1=(WURX(J)-WLLX(J))/WGRINT(J)+1.0
K2=(WURY(J)-WLLY(J))/WGRINT(J)+1.0
1056 NATST=NATST+K1*K2
C
C IS AVAILABLE MEMORY EXCEEDED
1057 IF(NATST*GT. NATST) GO TO 1057
1058 IRROR=1058
G0 TO 7734
C
C 1059 DO 1060 J=1,300
KPAD = (ISIN*17)*ZS(J)*UX(J)*UY(J)+SX(J)*SX(J)+SY(J)*SY(J)
1061 JTOPV=J-1
1062 IF(NNN-JTOPV) G0 TO 1056
C
C 2061 NN=JT0PV
WRITE (ISOUT*16)JT0PV
G0 TO 106
C
C 100 CONTINUE
100 IRROR=100
G0 TO 7734
C
C VECTOR DATA ARE IN R1A0 ARRAYS ON INDEXES J=1,JTOPV
C FIRST USE NC0DE AS A TEST CONTROL VARIABLE. BRANCH ON NC0DE VALUE.
C A COMPUTED GO TO TO THE DESIRED COMPUTATION METHOD CODE.
106 IF(NC0DE)110*110+112
112 IF(NC0DE=6)113*113+110
C
C NC0DE IS INCORRECT
110 WRITE (ISOUT*5)
NC0DE=1
113 G0 TO (115*116*117*118*119*120)*NC0DE
C
C 115 METHOD 1 USES THE NN NEAREST DATA POINTS. METHODS 2, 3 AND 4 ALSO
C USE THIS CODE BUT FOR METHOD 2, NVAL AND FOR METHOD 3, NNJTOPV.
C FOR METHOD 4, THE NN SPECIFIED BY THE JSLR IS USED IN THE
C LEAST SQUARES METHOD (NN MUST BE GREATER THAN THREL).
115 IBAUD(1)=1
X=0
JX=1
C
C NEXT FILL IN THE GRID SIZE VALUES
1151 IJJ(X)I=(XJ*XJ)+LLX(JX)I/6-GRINT(JX)+4999999
XJX=(XJ*XJ)+LLX(JX)+4999999
C
C NEXT INITIALLY FOR FILLING IN THE GRID
C2#GRINT(JX)/2+1
LX=1
LY=1
XG=ALLX(JX)+0.52
YG=ALLY(JX)+0.52
IF(JX=-JTOPV)1153+1155+1156
1154 IRR=1157
G. T# 7754
1155 IF(JX=1)1118+1126+1157
1156 2Q=(LETTH1(JX)-LETTH(JX-1)
G. T# 1128
1157 2Q=2Q+LETTH1(JX)-LETTH(JX-1)
G. T# 1128
1158 2Q=(LETTH1(JX)+LETTH(JX+1))/2+1
C
C SET ALL RADIX EQUAL TO J TO PROVIDE INDICES FOR THE FULL SET OF NNJ
C DATA POINTS AND TO PROVIDE AN INITIAL SET OF NEAREST- DATA POINTS.
C SET NADJ=1 TO BEGIN THE SEARCH PROCEDURE THAT SELECTS THE NEXT
C REPEAT OF THE SET OF NEAREST- DATA POINTS. NOTE THAT FOR THE LEAST
C FAST ALL THE NN NEAREST- POINTS ARE EQUALLY LIKELY TO BE THE NEXT.
C REPEAT OF THE SET.
1158 D=203 J=1+JTOPV
2:3 XNY(J)=J
XAUT=J
C
C COMPUTE DISTANCES BETWEEN THE CURRENT GRID POINT (X(J),Y(J)) AND
C EACH OF THE DATA VECTOR LOCATIONS
C
C COMPUTE SQUARES Z DELTAS
C LY) J=1+JTOPV
T2=(Z(J)-LY)
T1=1+T2
T2=(ALPHA-ALPHA)
IF(T1>T2)12=0
144 1DZ(J)=T2/MLPMe+T1
C
C COMPUTE LESS.Unmarshal 2 Z DELTAS
C (X(J)-X(J))=Y(J)
2:2 \( X(J)-Y(J)
C
C COMPUTE SQUARES INTANCES
C
C COMPUTE SQUARES INTANCES
2:7 \( X(J)-X(J)
T1=(X(J)-X(J))\#(X(J)-X(J))\#Y2(J)
T2=(EFT2-T1)/1+ET2+T1111*Z2(J)
IF(T1>T2)22=0
227 Z(J)=1

173
Go to 202

202 D2(J)=1.0/T2

202 CONTINUE

C FIND THE ADDRESS OF AND DISTANCE TO THE MOST REMOTE POINT OF THE
C NN -NEAREST- POINTS (THE POINTS WHOSE ADDRESSES ARE GIVEN BY
C NAD(I) THAT STORE THAT MAXIMUM DISTANCE IN THE WORD DK AND
C SET NADT SUCH THAT DM=DZ(NADT(NADT)).
D NAD(NADT)
D DM=D2(KL)
Dv 2C7 J=1,NN

KL=NAD(J)

IF(CM-D2(KL))2C8*2C7*2C6

Dv=D2(KL)

NADTR

207 CONTINUE

C AT THIS POINT, DK IS THE LARGEST D2(J) FOR D=NAD(J) NADT(NADT)
C IF (NN=JTYPV)2C72*2C72*2C73

C

C2072 NN=SELECT BEST NN POINTS
C SCAN THE SET D2(J)=NAD(J) JTYPV UNTIL A D2(J) LESS THAN ON
C IS FOUND, IF ONE IS FOUND, STUDY INHARM(NADT) WITH THE SELECTED NN
C THEN RESET DM AND NADT TO INDICATE THE MOST REMOTE OF THE NEAREST
C NN POINTS, WHEN THE FULL SET D2(J)=NAD(J)JTYPV HAS BEEN
C SCANNED, THE SET OF NEAREST NN POINTS HAS BEEN SELECTED, ONLY
C ONE SCAN IS REQUIRED

C2072 Dv 2C72 J=1,JTYPV

NAD=NAD(J)

IF(CM-D2(KL))2C8*2C6*2C1

NADTR=NAD(J)

NAD=NAD(NADT)

NADT=JTYP

C IN A SCAN TO AND WANT TO THE KEY MOST REMOTE POINT

Dv=D2(KL)

Dv 2C72 KKN=1*NN

KL=NAD(KKN)

IF(CM-D2(KL))2C8*2C6*2C6

Dv=D2(KL)

NADTR=KKN

C

C208 NN AND NADT ARE SET WITH THE PARAMETERS OF THE MOST REMOTE OF
C THE NEAREST NN POINTS

C2082 CONTINUE

C208 CONTINUE

C2073 CONTINUE

C THE NEAREST NN HAVE BEEN FOUND
C **********SOME DAY INSERT HERE A BRANCH IN HEARING METHOD HERE*********

C INCREMENT INDEX FOR STORING VECTOR COMPUTED FOR POINT (A, Y, Z, X)

C IS THE LEAST SQUARES METHOD TO BE USED, YES TO 2081

IF(INCOME=41208C*2081*2080

C C2081 THIS IS THE LEAST SQUARES METHOD
C INITIALIZE FOR LEAST SQUARES METHOD

2081 SNN=NN
`SDX=0.0
SDY=0.0
SDZ=0.0
SDX2=0.0
SDY2=0.0
SDZ2=0.0
SDXY=0.0
SDXZ=0.0
SDYZ=0.0
SAI=0.0
SIV=0.0
SUX=0.0
SVY=0.0
SUX=0.0
SVX=0.0
SVX=0.0
SVY=0.0
SUX=0.0
SVZ=0.0
SVZ=0.0
SUX=0.0
SVX=0.0
`
B21=SDX*SAUG1-SDX*SAUG2+SDZ*SAU3
B31=SDX*SAUG2-SDX*SDZ*SAU4+SDX*SAUG4
B41=SDX*SAUG3-SDX*SDZ*SAU5+SDX*SDZ*SDY-SDX*SDY
C   TEST TO SEE IF A ROW OR COLUMN IS APPEARANTLY ZERO
PP=MAX1(ABS(SNN*P11)+ABS(SDX*P21)+ABS(SDY*P31)+ABS(SDZ*P41))
1IF(PP-1.E-20)SAL*3BL*37C
C   COMPUTE DETERMINANT OF A
37C0  BDD=SNK*P11-SDX*P21+SDY*P31-SDZ*P41
C   TEST FOR LOSS OF PRECISION
37C1  IF(ABS(BDD/BL)=1.EC10)3BC*38C*34C
C   C350E  TOO MANY SIGNIFICANT FIGURES ARE LOST IN THE LEAST SQUARES
C   CALCULATION* THE DATA POINTS APPROACH A LINE* OR A CURVE.
C   PLAN*  USE THE WEIGHTED VECTOR METHOD
3800  WRITE (15261+23)XG*YG*26
G0  TO 2060
C   COMPUTE WIND VECTOR.
3900  VX(K)=B11*SAU-21*SAK+B31*SUY-341*SU4+BD
VX(K)=B11*SAU-21*SAK+B31*SUY-341*SU4+BD
VX(K)=B11*SAU-21*SAK+B31*SUY-341*SU4+BD
G0  TO 2060
C   C2CE;  COMPUTE AND SUM THE WEIGHTING FACTORS
2060  SUX=C00
D6  214  J=1*NN
L=N*1UJ
2142  D21(L)=1.0/D21(L)
214  SUX=SUX+D21(L)
1IF(SUX<1.0.E-8)1649  WRITE (15261+24)XG*YG*26
C   XH. COMPUTE VECTOR ESTIMATE AT GRID POINT
C   COMPUTE STORAGE INDEX
C   COMPUTE AND STORE WIND ESTIMATE AT GRID POINT
VX(K)=0.0
VY(K)=0.0
VZ(K)=0.0
D6  216  J=1*NN
L=N*1UJ
VX(K)=VX(K)+5X11*E2L
VX(K)=VX(K)+5X11*E2L
VX(K)=VX(K)+5X11*E2L
216  VZ(K)=VZ(K)+5Z11*E2L
VX(K)=VX(K)+SUX
VY(K)=VY(K)+SUX
VZ(K)=VZ(K)+SUX
209  XG*AG+XGRINT(J,J)
LX=ILA+1
IF(LX=ILJ(J,J))Z1211*2011*2012
2712  XG=HLX(3J1)+xG2
LY=LY+1
LX=1
YG=YG+XGRINT(J,J)
1IF(LY=11J(J,J))Z1211*2011*2012
1152  J=J+C
1IF(J=J(J,J))1066*1166*1166
116  J=J+C
1BASE(J,J)=1OAD15(JJ)+(IL(JJ))*1O0+10T

176
C 116 METHOD 2 NEAREST VECTOR
116 M=1
C 117 METHOD 3 ALL VECTORS WEIGHTED
117 N=U*TP
C 118 CONTINUE
C 119 METHOD 4 LEAST SQUARES
C USE BRANCH 4 NORM=2 TO BRANCH TO LEAST SQUARES IN CASE
C N MUST BE GREATER THAN 2 FOR THE LEAST SQUARES METHOD
119 IF(N-4,1E1,115,116)
119 N=N+1
C 120 WRITE (J=23,T=472)
C 121 WRITE (1,J=23,T=472)
C 122 WRITE (L=24,T=773)
C 123 WRITE (L=24,T=773)
C 124 WRITE (L=24,T=773)
C 125 WRITE (1,J=24,T=773)
C 126 WRITE (1,J=24,T=773)
C 127 WRITE (1,J=24,T=773)
C 128 WRITE (1,J=24,T=773)
C 129 WRITE (1,J=24,T=773)
C 130 WRITE (1,J=24,T=773)
C 131 WRITE (1,J=24,T=773)
C 132 WRITE (1,J=24,T=773)
C 133 WRITE (1,J=24,T=773)
C 134 WRITE (1,J=24,T=773)
C 135 WRITE (1,J=24,T=773)
C 136 WRITE (1,J=24,T=773)
C 137 WRITE (1,J=24,T=773)
C 138 WRITE (1,J=24,T=773)
C 139 WRITE (1,J=24,T=773)
C 140 WRITE (1,J=24,T=773)
C 141 WRITE (1,J=24,T=773)
C 142 WRITE (1,J=24,T=773)
C 143 WRITE (1,J=24,T=773)
C 144 WRITE (1,J=24,T=773)
C 145 WRITE (1,J=24,T=773)
C 146 WRITE (1,J=24,T=773)
C 147 WRITE (1,J=24,T=773)
C 148 WRITE (1,J=24,T=773)
C 149 WRITE (1,J=24,T=773)
C 150 WRITE (1,J=24,T=773)
C 151 WRITE (1,J=24,T=773)
C 152 WRITE (1,J=24,T=773)
C 153 WRITE (1,J=24,T=773)
C 154 WRITE (1,J=24,T=773)
C 155 WRITE (1,J=24,T=773)
C 156 WRITE (1,J=24,T=773)
C 157 WRITE (1,J=24,T=773)
C 158 WRITE (1,J=24,T=773)
C 159 WRITE (1,J=24,T=773)
C 160 WRITE (1,J=24,T=773)
C 161 WRITE (1,J=24,T=773)
C 162 WRITE (1,J=24,T=773)
C 163 WRITE (1,J=24,T=773)
C 164 WRITE (1,J=24,T=773)
C 165 WRITE (1,J=24,T=773)
C 166 WRITE (1,J=24,T=773)
C 167 WRITE (1,J=24,T=773)
C 168 WRITE (1,J=24,T=773)
C 169 WRITE (1,J=24,T=773)
C 170 WRITE (1,J=24,T=773)
C 171 WRITE (1,J=24,T=773)
C 172 WRITE (1,J=24,T=773)
C 173 WRITE (1,J=24,T=773)
C 174 WRITE (1,J=24,T=773)
C 175 WRITE (1,J=24,T=773)
C 176 WRITE (1,J=24,T=773)
C 177 WRITE (1,J=24,T=773)
C 178 WRITE (1,J=24,T=773)
C 179 WRITE (1,J=24,T=773)
C 180 WRITE (1,J=24,T=773)
C 181 WRITE (1,J=24,T=773)
C 182 WRITE (1,J=24,T=773)
C 183 WRITE (1,J=24,T=773)
C 184 WRITE (1,J=24,T=773)
C 185 WRITE (1,J=24,T=773)
C 186 WRITE (1,J=24,T=773)
C 187 WRITE (1,J=24,T=773)
C 188 WRITE (1,J=24,T=773)
C 189 WRITE (1,J=24,T=773)
C 190 WRITE (1,J=24,T=773)
C 191 WRITE (1,J=24,T=773)
C 192 WRITE (1,J=24,T=773)
C 193 WRITE (1,J=24,T=773)
C 194 WRITE (1,J=24,T=773)
C 195 WRITE (1,J=24,T=773)
C 196 WRITE (1,J=24,T=773)
C 197 WRITE (1,J=24,T=773)
C 198 WRITE (1,J=24,T=773)
C 199 WRITE (1,J=24,T=773)
C 200 WRITE (1,J=24,T=773)
**COMMON SET1**

1. S
2. BYLO
3. YUL
4. ALL
5. ZP
6. XL
7. RLAX
8. ILUPO
9. JNUL
10. JUPO
11. NLU
12. NRU
13. CRMA
14. UU
15. KRU
16. RUPAR

**COMMON SET2**

1. M
2. TFIN
3. RT
4. RDP
5. RUPL
6. TUR
7. RL
8. RIC
9. RUO
10. RLN
11. RIK
12. RUM

**COMMON SET3**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET4**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET5**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET6**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET7**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET8**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET9**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE

**COMMON SET10**

1. P
2. PUC
3. PUS
4. PUD
5. PUP
6. PUE
7. PUN
8. PUP
9. PUE
10. PUN
11. PUP
12. PUE
C

JW=0
NUL=0
EPSI=1.0

C

ARE TRANSPORT TRACES TO BE WRITTEN... YES TO 5500
IF (IC(6)=1) THEN 5500*5500
5500 MPNT=2
GO TO 5520
5510 MPNT =1
C

BRANCH TO READ ADDITIONAL DATA
C

READ DATA SPECIFIED TO EACH LOCAL MIND SYSTEM
5520 IF (NINLOC=1) THEN 5510*5511
C

C 511 SETTING A NEGATIVE WILL CAUSE THE LOCAL CIRCULATION SYSTEM CODES
C TO READ THE DATA THAT THEY NEED WHEN THEY ARE FIRST ENTERED
511 JS=1
J0 S0 I=1*NLOC I
K=1
NC=1*NKTYP(I)
GO TO (S0U, S0U+200+S0U+S0U+S0U, S0U)
C

501 CALL MFAND1(JN, AY, AT, AY, AY, AY, AY)
GO TO 502
502 CALL RGAND1(JN, AY, AY, AY, AY, AY, AY)
GO TO 503
503 CALL CREAL1(JN, AY, AY, AY, AY, AY, AY)
GO TO 504
C

C ********** CODE INSERTION POINTS **********
C

504 CONTINUE
C

C ********** CODE INSERTION POINTS **********
C

505 INKRE=0.0
GO TO 506
506 CONTINUE
C

C

507 IF (IL=1) THEN 5800*5800
48 IF (IL=1) THEN 5700
49 IF (IL=1) THEN 5900
50 ABB=10 I=1
ABB 100 I=1
I=100
I=100
I=100
I=100
I=100
I=100
10.0 IF (ABB=1) THEN 5900
10.0 IF (ABB=1) THEN 5900
10.0 IF (ABB=1) THEN 5900
1113 JS=1
GO TO 1114
C

C ALREADY TO READ IN BLOCK OF PARTICLES NEAR BOUND RECORDS
C FIRST READ BLOCK SIZE
1112 READ (IPTIN)
C

C IF BLOCK SIZE IS NEGATIVE OR ZERO, BLOCK DISPLAYED
IF (NINLOC=1) THEN 1113
C

C CHECK TO SEE IF BLOCK WILL FIT IN ARRAY
1011 IF (NINLOC=1) THEN 21*1021*103
C
C 103  ERROR - ALQIPT LIST TOO LARGE, SHOULD NEVER HAPPEN, GO TO EXIT.
C 103  IRKOK=103
C
C 300  GENERALIZED ERROR STOP
C 300  CALL ERKOK(FLUGMA,IRKOK,ISWP)
C
C 1021 CALL DUMP
C
C
C 102  NOW READ A BLOCK OF PARTICLE ALQIPT DESCRIPTIONS
C 102  IF=1
C 102  READ (IPAKIN)XPI,J),Y(J),50(J),F(J),P(J),F(J),T(J),JU(J),JU(J)
C C ARE TRANSPORT TABLES TO BE WRITTEN, YET TO TRY
C IF(I)=0:0 1992159317961799
C 552 NO ALQIPT LIST FOUND
C 5521 NFREL-NFRERE=N
C
C **************************************************************
C 103  BEGINNING OF PARTICLE ALQIPT LIST LOOP
C 1031  DO 100 J=1,IF
C 1031  ASSIGN 3.52 TO IPAS
C 1031  GO TO (5540,5550,5560,5570)
C 5540 ALQIPT LISTSfrared=TWRED (IPAS,IPAS+1,IPAS+2,IPAS+3,IPAS+4,IPAS+5)
C
C 115 SELECT PARTICLE TO BE TRANSPORTED -- VIP +IPAS
C 194 IF(IPAS(J))=0 GO TO 192
C 192 IF(IPAS(J))=0 GO TO 194
C 194 IF(IPAS(J))=0 GO TO 197
C 197 IF(IPAS(J))=0 GO TO 199
C
C 199 A PARTICLE HAS BEEN SELECTED
C 199  IN THE CURRENT PARTICLE WITHIN A LOCAL CIRCULATION SYSTEM
C 1991  IF(CIRC(J))=0 GO TO 197
C 1992  GO TO 199
C C **************************************************************
C 1993  ALQIPT LISTS 2,3,4,5,6,7
C 19931 IF(CIRC(J))=2 GO TO 199
C 19932 IF(CIRC(J))=3 GO TO 199
C 19933 IF(CIRC(J))=4 GO TO 199
C 19934 IF(CIRC(J))=5 GO TO 199
C 19935 IF(CIRC(J))=6 GO TO 199
C 19936 IF(CIRC(J))=7 GO TO 199
C
C 1996 THE JTH PARTICLE IS WITHIN THE JTH LOCAL CIRCULATION SYSTEM
C 19961 IN A LOCAL MINI SYSTEM
C 19962 CALL LPRINT(J)
C C
C 1997 IF IN ARE A NEWLY DISCOVERED PARTICLE ARE BEING ADDED
C IF(ZP(J))=0 GO TO 197
C 19971 IF(ZP(J))=1 GO TO 197
C 19972 IF(ZP(J))=2 GO TO 197
C 19973 IF(ZP(J))=3 GO TO 197
C 19974 IF(ZP(J))=4 GO TO 197
C 19975 IF(ZP(J))=5 GO TO 197
C 19976 IF(ZP(J))=6 GO TO 197
C 19977 IF(ZP(J))=7 GO TO 197
C
C 105 CONTINUE
C
C

1956 \texttt{XX=XP(J)}
\texttt{YY=YP(J)}
\texttt{ZZ=ZP(J)}
\texttt{PSIZE=PS(J)}

1957 \texttt{CALL GETWAV(XX,YY,ZZ,WA,JW)}
\texttt{IF(WA=1) \texttt{Y=100+100}}
\texttt{GU TO 160}
\texttt{IF(WA=2) \texttt{Y=200+200}}
\texttt{GU TO 166}

1961 \texttt{THE NEEDED WIND FIELD IS NOT IN CORE}

1959 \texttt{FMAS(J)=-FMAS(J)}
\texttt{TP(J)=TP(J)}
\texttt{NL=NL+1}
\texttt{GU TO 160}
\texttt{INPUT=-100}
\texttt{GU TO 300}

C

190 \texttt{USE PARTICLE SCALE NO. \texttt{NO} FOR IT IN \texttt{FM} \texttt{WIND SCALE POSITIVe.}}

190 \texttt{CALL PARTICLE\_SIZE\_PARAM(\texttt{PARAM},\texttt{NO})}

C

\texttt{COMPUTE VERTICAL PARTICLE MOTION COMPONENT}

\texttt{IN POSITIVE \texttt{VM} DENOTES \texttt{UPWARD POINTING VECTOR}}

\texttt{VZ=Z(J-1)+K}

C

\texttt{COMPUTE \texttt{TIME} TO NEXT MAGNETIC WIND FIELD BOUNDARY}

\texttt{COMPUTE \texttt{T1} \texttt{-- TRANSIT TIME TO A BOUNDARY}}

\texttt{\texttt{AX}=\texttt{AX}(J-1)+\texttt{AX}(J)/\texttt{AX}(J)}

\texttt{\texttt{AY}=\texttt{AY}(J-1)+\texttt{AY}(J)/\texttt{AY}(J)}

\texttt{\texttt{T1}=\texttt{AX}+\texttt{AY}+\texttt{AX}(J)/\texttt{AX}(J)+\texttt{AY}(J)/\texttt{AY}(J)}

\texttt{GU TO 165}

\texttt{ASSIGN \texttt{T1} TO \texttt{NI}}

\texttt{IF(A1=AX\_AU) \texttt{TI} \texttt{TO}\texttt{102+102}}

191 \texttt{T1=(XAXL-AP(J))/10A(JXAU)}
\texttt{GU TO 160}

192 \texttt{T1=100+100}
\texttt{ASSIGN \texttt{T1} TO \texttt{NI}}
\texttt{GU TO 164}

193 \texttt{T1=100+100}
\texttt{ASSIGN \texttt{T1} TO \texttt{NI}}
\texttt{GU TO 164}

C

\texttt{COMPUTE \texttt{T2} \texttt{-- TRANSIT TIME TO Y BOUNDARY}}

194 \texttt{ASSIGN \texttt{T2} TO \texttt{NC}}
\texttt{IF(VY(J-1)=10+10+10)}

195 \texttt{T2=(VY-TP(J))/100+100}
\texttt{GU TO 166}

196 \texttt{T2=100+100}
\texttt{ASSIGN \texttt{T2} TO \texttt{NC}}
\texttt{GU TO 166}

197 \texttt{T2=(YOU-TP(J))/100(JW)}

C

\texttt{COMPUTE \texttt{T3} \texttt{-- TRANSIT TIME TO Z BOUNDARY}}

C 198 \texttt{IS PARTICLE MOVING UP OR DOWN. UP TO 171}

198 \texttt{IF(VP2)=10+10+10}
\texttt{GU TO 171}

C 199 \texttt{IS PARTICLE BELOW MAX TOMP HEIGHT? YES TO 1091}

199 \texttt{IF ((X(J))=10+10+10)}
\texttt{GU TO 175}

C

\texttt{IS MAX TOMP HEIGHT ABOVE SLICE BOTTOM? YES TO 1092}

198 \texttt{IF ((TOPS=10+10+10))=10+10+10}
\texttt{GU TO 177}

198 \texttt{T3=10+10+10}
\texttt{GU TO 177}

181
C 1711 FIND THE EALRIEST INTERSECTION WITH A LOCAL CIRCULATION SYSTEM
C 1711 CIRCULATION
C ARE THERE ANY LOCAL CIRCULATION SYSTEMS? YES TO 1714
IF (NCIRC>1) 172, 172, 1714
C
C 1712 COMPUTE TIME OF FLIGHT TO EACH OF THE FOUR VERTICAL PLANES THAT
C SURROUND THE JTH LOCAL CIRCULATION CELL.
1712 GO 1713 LJI=1, NCIRC
GO TO 1917*10, 171
918 TX1=1000.0000
TX2=1000.0000
GO TO 921
917 TX1=(CXMXX(LJ)-XF(LJ))/VX(LJ)AD
TX2=(CXMXX(LJ)-XFI(LJ))/VX(LJ)AD
921 GO TO (919, 920), N2
920 TY1=1000.0000
TY2=1000.0000
GO TO 922
919 TY1=(CXMNY(LJ)-YF(LJ))/VY(LJ)AD
TY2=(CXMNY(LJ)-YFI(LJ))/VY(LJ)AD
C
C TEST X INTERCEPTS
C IS THE FIRST X DIRECTION INTERCEPT IN THE PAST? YES TO 1714
922 IF(TX1)1714, 1714, 1716
C
C 1714 IS THE SECOND X DIRECTION INTERCEPT ALSO IN THE PAST? IF YES,
C BOTH X DIRECTION INTERCEPTS ARE IN THE PAST AND THE PARTICLE WILL
C NOT INTERSECT THIS CELL. GO TO 1713 TO CONSIDER THE NEXT CELL.
1714 IF(TX2)1713, 1713, 1715
1715 TMINA=TX1
TMAXA=TX2
GO TO 1717
1716 IF(TX1)1716, 1719, 1717
1718 TMINA=TX1
TMAXA=TX2
GO TO 1717
C
C 1719 BOTH X INTERCEPTS ARE IN THE FUTURE
1719 IF(TX1-TX2)1715, 1715, 1716
C
C 1717 NOW TEST FOR Y INTERCEPTS
1717 IF(TY1)1720, 1720, 1721
1720 IF(TY2)1713, 1713, 1723
1723 TMINY=TY1
TMAXY=TY2
GO TO 1724
1721 IF(TY2)1725, 1725, 1726
1725 TMINY=TY2
TMAXY=TY1
GO TO 1724
C
C 1726 BOTH Y INTERCEPTS ARE IN THE FUTURE
1726 IF(TY1-TY2)+1723*1723+1765
C 1724 NOW SELECT FIRST INTERCEPT
C
C SELECT THE SECOND PLANE PIERCE (LAST OF FIRSTx-FIRSTY)
1724 IF(TH1+x-THINY)1727*1728*1728
1728 TS=TMNX
TC=TMAXY
GO TO 1729
1727 TS=TMINY
TC=TMAXX
1729 IF(TS-TC)1730*1732*1732
C
C 1730 KEEP TIME OF EARLIEST INTERCEPT
1730 IF(TS-CEIRMIN)1731*1732*1732
1731 CEIRMIN=TS
1733 CONTINUE
C*********** * * * * * TEMP * * * * * **********
WRITE (200,111)CEIRMIN
C AT THIS POINT CEIRMIN CONTAINS THE TIME OF THE FIRST INTERCEPT
C BETWEEN PARTICLE AND A LOCAL CIRCULATION SYSTEM
172 C 172 NOW SELECT EARLIEST BOUNDARY
172 IF(TS-TS1)1723*1723*1723
173 IF(TS-TS2)1730*1730*1730
176 TS=TSX
IR1=1
GO TO 179
C
C 178 STORE SMALLEST TIME OF FLIGHT STORAGE AT THE TIME OF FLIGHT TO THE
C Z BOUNDARY
178 TS=TSX
IR2=3
GO TO 179
179 IF((TY-TY1)+1731*178+178
180 TS=TSX
IR2=2
C
C 179 TO INTERSECTION WITH LOCAL SYSTEM PRIOR TO EARLIEST BOUNDARY
179 IF(THS-CEIRMIN)1791*1792*1792
1792 TS=CEIRMIN
1793 IR2=3
C
C 1791 DOES TIME LIMIT COME BEFORE EARLIEST OTHER BOUNDARY, YES TO 182
C 1791 IF(TS-THS)ENUTIM1791*1791*1791
C
C 182 TRANSPORT PARTICLE UNTIL ENUTIM
182 TS=ENUTIM-THS
IR4=4
GO TO 501
C
C 181 TEST FOR EXCESSIVELY SMALL MOVEMENT
181 GO TO(5570,5580)5570
5570 WRITE ([J]0011).TS+VPL1+VPL2+ILY+FY
5580 IF(THS=EPOL)3.50*3.00*3.067
C
C SPECIAL TRANSPORT IN THE EVENT OF EXCESSIVELY SMALL LOM
C
LINT 293
LINT 294
LINT 295
LINT 296
LINT 297
LINT 298
LINT 299
LINT 300
LINT 301
LINT 302
LINT 303
LINT 304
LINT 305
LINT 306
LINT 307
LINT 308
LINT 309
LINT 310
LINT 311
LINT 312
LINT 313
LINT 314
LINT 315
LINT 316
LINT 317
LINT 318
LINT 319
LINT 320
LINT 321
LINT 322
LINT 323
LINT 324
LINT 325
LINT 326
LINT 327
LINT 328
LINT 329
LINT 330
LINT 331
LINT 332
LINT 333
LINT 334
LINT 335
LINT 336
LINT 337
LINT 338
LINT 339
LINT 340
LINT 341
LINT 342
LINT 343
LINT 344
LINT 345
LINT 346
LINT 347
LINT 348
LINT 349
LINT 350

183
GO TO IPAS(13052+3053)

C

ASSIGN 3052 TO IPAS

JWAD=JWAD

VPZT=VPZ

TSMEPSIL

GO TO 1811

IF (VPZT>VPZ) 3053:3053:3052:3052

VPZ=VPZ

TIL=TILIMIT

GO TO 3055

VPZ=(VPZ+VPZ)/2.0

C

IF VECTORS ARE OF OPPOSITE SIGN, USE A ZERO

IF (VPZ*VPZ) 3056:3056

VPZ=0.0

TIL=TILIMIT

GO TO 3055

VPZ=(VPZ+VPZ)/2.0

C

IF (VPZ*VPZ) 3059:3059

VPZ=VPZ

TIL=TILIMIT

GO TO 3055

C

VPZ=(VPZ+VPZ)/2.0

C

if (VPZ*VPZ) 3062:3062

VPZ=VPZ

TIL=TILIMIT

GO TO 3060

C

TP(U)=TP(U)+TP(H)

C

TP(U)=TP(U)+TP(H)

C

TP(U)=TP(U)+TP(H)

GO TO 3063

C

ASSIGN 3052 TO IPAS

C

IF (TP(U)=TP(U)) 1801:1801:1802:1802

1802 GO TO 1803:1803

C

1803 TRANSPORT PARTICLE FOR TSM BY STEPS OF UTMAC CHECKING TOPO AS WE

C

GO

C

COMPUTE MOVEMENT INCREMENTS FOR AXIYX AND Z DIRECTIONS

1813 XIN=UTMAC*VYJWAD)

C

1814 ZIN=UTMAC*VZP

C

1815 TP(U)=TP(U)+TP(U)+TP(U)

C

1816 TP(U)=TP(U)+TP(U)+TP(U)

C

1817 ASSIGN TP(U)=TP(U)

C

1818 TRANSPORT PARTICLE FOR TSM BY STEPS OF UTMAC CHECKING TOPO AS WE

C

GO

C

TEST FOR PARTICLE IMPACT ON TOPOGRAPHY

C

184
X=XP(J)  
Y=YP(J)  
CALL HEIGHT(X,Y,H)  
IF(J+2*JADJ+1)EQ160 CONTINUE  
1813 IF(J+2*JADJ+1)EQ160 CONTINUE  
1815 IF(J+2*JADJ+1)EQ160 CONTINUE  
1816 IF(J+2*JADJ+1)EQ160 CONTINUE  
1817 IF(J+2*JADJ+1)EQ160 CONTINUE  
C  
C 1811 TRANSPORT PARTICLE FOR TSM  
C  FIRST INCREASE TSM TO ASSURE THAT THE PARTICLE WILL ACHIEVE ITS  
C BOUNDARY  
1811 TSM=TSM*1.0+0.01  
XP(J)=XP(J)+X(JAD)TSM  
YP(J)=YP(J)+Y(JAD)TSM  
ZP(J)=ZP(J)+Z(JAD)TSM  
TP(J)=TP(J)+TSM  
3U63 CONTINUE  
C  
C ARE TRANSPORT TRACKS TO BE WRITTEN? YES TO 2200  
GO TO (5550,5550,5550)  
5550 WRITE(1,12)XP(J),YP(J),ZP(J),TP(J),I1R  
11R  
C  
C TEST FOR PARTICLE IMPACT ON TOPOGRAPHY  
5556 IF(II(J)-I2(J))LGE1 I5 CONTINUE  
C  
C 189 PARTICLE IS NOT U-V-O-C  ADJUST INDICES IN NEW GAME AND EXIT  
C RECYCLE  
189 GO TO 120525202575172+11R  
190 NTI=NTI+1  
TP(J)=TP(J)+I1R  
GO TO 160  
250 JW=ZW  
GO TO 1450  
252 IF(VZ(JAD))LGE230  
254 JW=W-JW  
GO TO 1961  
256 JW=(JW+1)  
272 GO TO 1961  
187 GO TO I1R(I1+I171)  
1871 X=XP(J)  
Y=YP(J)  
CALL HEIGHT(X,Y,H)  
1874 IF(J+2*JADJ+1)EQ160 CONTINUE  
C  
C 1872 =-i180=J+ PARTICLE BEYOND SPECIFIED TOPO  
1872 FMAS(J)=FMAS(J)  
TP(J)=TP(J)+1  
NLST=NLST+1  
GO TO 160  
C  
C 1875 M+1=I100=U PARTICLE BEYOND IN-CORE TOPO  
1875 TP(J)=TP(J)+1  
IF(JTUP1)EQ107718D  
1877 JTP1=1  
GO TO 160  
185
1876 IF(IN=IPIJ))169*169
C
C 188 TAKE CARE OF GROUND PARTICLES
188 FIX(A(J)-FIX(A(J))
TPI(J)=TPI(J)
NG=NG1
169 CONTINUE
C
C *****************************************************************
C
C END OF TRANSPORT LOOP
C
C *****************************************************************
C
190 IF(J(J)=1)1890
C
C ARE ANY PARTICLES ON THE ON-TAPE?
191 IF(J(J)=1)1890
C
C ARE ANY PARTICLES ON THE OFF-TAPE?
192 IF(J(J)=1)1890
C
C ARE ANY PARTICLES ON THE OUT-OF-WIND-FIELD TAPE?
193 IF(J(J)=1)1890
C
C 2.21 ARE PARTICLES ALONG THE BOUNDARY TAPE IN USE 03?
194 IF(J(J)=1)1890
C
195 A NEGATIVE JAINU INDICATES SOME PARTICLES ARE IN THE OUT OF THE
196 WIND FIELD BUFFER BUT NOT ON TAPE
197 A JAINU GREATER THAN ZERO INDICATES PARTICLES ARE ON THE OUT OF
198 THE WIND FIELD TAPE
199 180 SET THE REQUIRED TOPU DATA FROM TAPE
200 181 TO 187 SCAN BUFFER PARTICLE COUNTERS TO DETERMINE NEXT NEED
201 TOPU DATA CLOUD AND CHOOSES THE NEAREST ONE FOR READING.
202 "BLOCK 0" IS SET BY THE Initializing PROGRAM WHICH READS THE TOPU
203 TAPES IDENTIFICATION RECORDS
204 NINTAKIJ IS SET BY THE TRANSPORT LOOP WHEN PARTICLES LEAVE TOPU
205 AND RESET WHEN OFF-TOPU BUFFER IS EMPTIED.
206
207
C 105 JTEST=1000
  DU 107 J=1,16BLCK
  JTEST=NINT(J)
  IF J=1,16BLCK THEN GOTO 108
  108 JTEST= JTOPJU+1
  IF JTEST=JTOPJU THEN GOTO 112
  JERRNO=JU

C SEeks THE FILE IN CORE, SHoudL NEVER HAPPen, Go TO A STOP CAll.
GO To 3:
  111 IF JTEST=JTEST1 THEN GOTO 107
  JTEST=JTEST2
  GO TO 140
  110 IF JTEST=JTEST2 THEN GOTO 119
  JTEST=JTEST1
  JF=J
  117 CONTINUE

C At THIS PoINT 0P AND THE NUMbER OR THE DEsIRED FILE
C NuM MuVE TaMe To SELECTEd FILE
  IF JF TIME=l THEN JF TIME=12
  IF JF TIME=1 THEN JF TIME=12
  113 JTEST=JTEST1
  JF=J

C 1/71 PREPARE T0 MOVE FORWARD ON TiMEs, Complete NUMbER OR BuCKLES T0
C READ IN
  117 JF=JF-JTOPJU+1
  GO To 147

C 1/72 DEsIRED FILE IS DEsIRED FILE, BACK UP OR GET L1E
  117 READ TIMEO

C READ SKIP OVER (iINITAL RECORD
  READ (1,TIME,15)
  READ (1,TIME,15)
  READ (1,TIME,15)
  READ (1,TIME,15)
  JF=JF

C 1/74 READ OR THROUGH THE DEsIRED BLOCK
  117 GO JF=J, JHJ=J
  117 CALL HJUfJU(J)

C 1/16 READ All OFF-TOPU PARtICLeS
  116 GO J=1, J=1,16LOfF
  IF F(J)=1 THEN 117
  117 IF F(J)=1 THEN 117
  117 IF F(J)=1 THEN 117
  117 IF F(J)=1 THEN 117
  117 CONTINUE
  IF F(J)=1 THEN 117
  117 JTIME=J
  GOTO 117

C 1/15 READ OFF-TOPu AND PARtICLeS ALOfF TAPES AND SWap NAMEs
  115 WRITE(JTIME)JUL

187
RE\eind I\u00f8 ТоPУ
RE\eind I\u00f8арин
I\u00f8епп=I\u00f8 ТоPУ
I\u00f8 ТоPУ=I\u00f8арин
I\u00f8арин=I\u00f8епп
I\u00f8оПИ=0
Gо TO 1201

C 120 GеТ the required and filled data from tape
13\eй Gо TO 124
C Insert code here
C
C 124 Rеаd all in-core out-of-core-filled particle keys
124 0U ICs J=1,VALUFT
124 IF F=10(J) J=1241*126*126
124 IF (J\u00f8 JUIN) J=1241*126*126*2+124
124 IF (J\u00f8 JUIN)-1 J=1241*126*126*2+124
124 F\u00eawas(J)-F\u00eawas(J)
125 Cонtinue
125 F= VALUFT
125 IF(J\\u00f8 JUIN) J=1241,125,126
125 J\u00f8UIN=0
125 J1,451=1
125 Gо TO 1241

C 125 Rеаd all out-of-core and filled particle klout tapes and output names
125 Writе(J\u00f8UIN=1) Gо
125 J\u00f8UIN=0
125 RE\eind 1GAIND
125 RE\eind I\u00f8арин
I\u00f8епп=I\u00f8AIND
125 IF(J\u00f8 JUIN) J\u00f8 I\u00f8арин
125 I\u00f8арин=I\u00f8епп
125 I\u00f8оПИ=I\u00f8епп
13\eй Gо TO 1201
C
END

188
**IBFTC GETAN LIST DECK**

**SUBROUTINE GETWU**

28 NOVEMBER 1966

**** SCHWENKE TECHNICAL OPERATIONS RESEARCH

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

C

THIS SUBROUTINE RETRIEVES THE MACRO WIND FIELD VECTOR WHICH

APPLIES AT THE POINT WHOSE COORDINATES ARE IN THE ARGUMENT ARGUMENTS GETA

XY, YY, AND Z WIND VECTOR COMPONENTS GETA

ARE IN THE COMMON VARIABLES XXY(XJWAD), YY(JWAD), AND ZZ(JWAD) WHEN GETA

THE SUBROUTINE RETURNS. IF AN ENTRANCE IS MADE WITH ARGUMENT JN GETA

SET NEGATIVE, JN IS SET POSITIVE AND ITS VALUE IS USED RATHER THAN GETA

RECOMPUTED. UPON EXIT JWAD IS SET NEGATIVE IN THE EVENT THAT THE GETA

MACRO WIND FIELD REMAINING TO THE ENTRANCE POINT IS NOT MINIMIZED GETA

IN CURVE.

GETA

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

**Glossary**

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
</table>
| JW          | INDEX OF THE WIND ARRAY STRAIN IN THE MACRO WIND FIELD GETA
| JWAD        | AN INDEX FOR THE ALLOCATION ARRAY TWO DIMENSIONAL FOR GETA
| JJ          | INDEX OF THE WIND LAYER USED IN A SEARCH FOR THE LAYER CONTAINED IN GETA
| JJUP        | INDEX OF THE TOP LAYER IN THE MACRO WIND FIELD DESCRIPTION GETA
| JJUP        | TOTAL TEMPORARY STORAGE GETA
| XX          | X COORDINATE OF THE POINT FOR WHICH A MACRO WIND FIELD IS GETA
| YY          | Y COORDINATE OF THE POINT FOR WHICH A MACRO WIND FIELD IS GETA
| JJR        | NUMBER OF THE STATEMENT NEAR THE POINT WHERE AN ERROR IS DISCOVERED GETA
| ALLAT1      | LOWER LIMIT FOR X COORDINATE IN THE MACRO WIND FIELD GETA
| ALLAT1      | UPPER LIMIT FOR X COORDINATE IN THE MACRO WIND FIELD GETA
| ALLAT1      | LOWER LIMIT FOR Y COORDINATE IN THE MACRO WIND FIELD GETA
| ALLAT1      | UPPER LIMIT FOR Y COORDINATE IN THE MACRO WIND FIELD GETA
| SJM         | A DIRECTION WIND FIELD RETRIEVAL INDEX GETA
| SJM         | Y DIRECTION WIND FIELD RETRIEVAL INDEX GETA
| SJM         | A DIRECTION WIND FIELD RETRIEVAL INDEX GETA
| SJM         | Y DIRECTION WIND FIELD RETRIEVAL INDEX GETA
| SJM         | A DIRECTION WIND FIELD RETRIEVAL INDEX GETA
| SJM         | Y DIRECTION WIND FIELD RETRIEVAL INDEX GETA

**Common 1/2**

1 JOHN - JELLO - LANE - LANE - JUIN - IOOT - GETA
2 JU - JUH - JAP - JAH - JRE - JRE - GETA
3 JUH - JUH - JRE - JRE - JRE - JRE - GETA
4 JAP - JAP - JAP - JAP - JAP - JAP - GETA
5 JAH - JAH - JAH - JAH - JAH - JAH - GETA

**DIMENSION**

GETA 20
UL: 41200)
UPL: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
ULP: 41200)
C  FIRST OUT:  FUTURE COMPUTE TIME TO COMING A BOUNDARY OF LOCAL CELL
1205  IF(VX1JWAD))1205+1206
1206  GOTO 1207
C  COMPUTE TIME TO COMING A BOUNDARY INTERCEPT
1207  IF(VY1JWAD))1207+1208
1208  GOTO 1209
C  COMPUTE TIMES TO COMING A BOUNDARY (HORIZONTAL) OF LOCAL WIND CELL
1209  IF(VX1JWAD))1209+1210
1210  GOTO 1211
C  COMPUTE TIME UNTIL TIME FOR UPDATING THE WIND FIELD
1211  TIM=TIME-TIM(J)
C  SELECT TIME UNTIL THE FIRST VALID BOUNDARY INTERCEPT
1212  IF(IN1JWAD))1212+1213
1213  GOTO 1214
C  THE LOCAL CELL (IN X1JWAD)
1214  TIM=TIME+TIME(J)
C  TRANSPORT THE PARTICLE FOR THAT PERIOD OF TIME
1215  GOTO 1216
C  PARTICLE IS WITHIN LOCAL CELL. DRAWER TO CALL APPROPRIATE WIND
C  PROGRAM
C  GRAND TO CALL A LOCAL WIND PROGRAM TO TEST PARTICLE FOR IMPACT OR NOT
C  TOPOGRAPHY, IF IMPACTED ADDING A LARGE QUANTITY WIND COMPONENT
C  PARENTS, IF NOT IMPACTED IT COMPUTES CORRECT WIND COMPONENTS AT
C  THE PARTICLE POSITION
1216  GOTO 1217
1217  CALL 'WIND1(u,v,x1jwad)
1218  GO TO 130
1219  CALL 'WIND2(u,v,x1jwad)
1220  GO TO 130
1221  CALL 'WIND0(u,v,x1jwad)
C  CODE INSERTION POINTS **************************************************
C  CONTINUE
C  CONTINUE
C  CODE INSERTION POINTS **************************************************
C  CALL 'ERROR(1)
C  CALL 'ERROR(2)
C  CALL 'ERROR(3)
C  CALL 'ERROR(4)
C  TRANSPORT THE PARTICLE FOR ONE TIME INCREMENT (UTLCN)
130 XP(J)=XP(J)+DLUC*(AZ-FV)
XP(J)=XP(J)+DLUC+AX
YP(J)=YP(J)+DLUC+AY
TP(J)=TP(J)+DLUC

C 131 TEST FOR BOUNDARY CROSSING
C IS PARTICLE AT OR BEYOND THE LOCAL CIRCULATION BOUNDARIES
C YES TO 132
131 IF(XP(J)-CRH(NY(J)))/132
132 IF(YP(J)-CRH(NY(J)))/132
132 IF(CRH(NY(J)-TP(J))/132
137 IF(CRH(NY(J)-TP(J))/132

C 137 IF IT IS NOT IN LOCAL SYSTEM, REMOVE IMPACTED PARTICLES
C AND THE PARTICLES BEYOND THE LOCAL CIRCULATION LIMITS
C YES TO 132
137 IF(XP(J)-CRH(NY(J)))/132
132 IF(YP(J)-CRH(NY(J)))/132
132 IF(CRH(NY(J)-TP(J))/132

C 132 PARTICLE IS STILL WITHIN THE LOCAL SYSTEM, NO CHECK TIME BOUNDARIES
C IF THE PARTICLE HAS TRANSPORTED OR TO OR BEYOND THE TIME FOR
C UPDATING THE WIND FIELD
136 IF(TP(J)-END)/136
136 IF(TP(J)-END)/136
132 IF(PARTICLE CANNOT BE MOVED FURTHER BY LOCAL SYSTEM TRANSPORT)
C
132 RETURN
END
SUBROUTINE VTWN1(J, K, X, Y, A, L)

COMMON /SET1/

DIMENSION DE1(21,40)

COMMON /SET2/

PARAMETER REQUIRING TO SUBROUTINE VTWN1

DIMENSION AN(12), X(12), Y(12), A(12), L(12)

DIMENSION MOUNTAIN X(1), Y(1)

C(J) HALF-MOUNTAIN OR THE J-Th MOUNTAIN

AM WIND VECTOR EAST

AT WIND VECTOR NORTH

AZ VERTICAL WIND VECTOR
C 1UU THIS IS THE DATA READING ROUTE
100 J=0
103 J=J+1
READ (13IN1) XM(J),YM(J),M(J)*AI(J)
A2(J)=A(J)*AI(J)
A+1(J)=A2(J)*A(J)*H(J)
IF(J-12)103110321032
1032 IF(RRR=1032 GO TO 734
1031 IF(M(J))103910391039
C1099 COMPUTE THE KTH LOCAL CELL HEIGHT1
1099 DZ=ABS13*U*H(J))
IF (DZ-CRUMT(K))11011001100
1100 CRUMT(K)=DZ
C 110 CHECK TO SEE THAT THE MOUNTAIN JUST READ IS WITHIN THE LIMITS OF
C THE KTH LOCAL WIND SYSTEM1
111 IF(XM(J)-CRMINX(K))1141116111
111 IF(XM(J)-CRMAXX(K))1121112114
112 IF(YM(J)-CRMINY(K))114111112
113 IF(YM(J)-CRMAXY(K))110111114
C
C 114 THE MOUNTAIN IS NOT WITHIN THE LIMITS OF THE KTH LOCAL WIND SYSTEM1
114 IF(RRR=114 GO TO 734
C 115 DZ = HI(J)/AI(J)
IF (DZ=10.6116116
C 116 THE MOUNTAIN RATIO HI(J)/AI(J) IS LESS THAN 0.6
116 IF(RRR=116 GO TO 734
C
C 110 NMT=J-1
C 111 THE NUMBER OF MOUNTAINS REPRESENTED IN THIS
C MOUNTAIN WIND SYSTEM
111 NMT=J-1
C 112 COMPUTE UNPERTURBED VECTOR HERE
C THE FOLLOWING THREE CARDS CONSTITUTE THE LOCATION COORDINATES OF
C THE UNPERTURBED WIND VECTOR
YY=(CRMAXY(K)+CRMINY(K))/2.0
XX=(CRMAXX(K)+CRMINX(K))/2.0
ZZ=CRUMT(K)/2.0
CALL GETWN0(XXYYZZJWAD)*JW)
IF(JWAD=14311451145
1143 IF(RRR=1143 GO TO 734
1144 IF(RRR=1144 GO TO 734
C THE FOLLOWING THREE CARDS CONSTITUTE THE MAGNITUDE AND DIRECTION
C OF THE UNPERTURBED WIND VECTOR
1145 UD(J)=SQRT(VXJWAD)*VXJWAD+VY(JWAD)*VY(JWAD))
SN(J)=VY(JWAD)/UD(J)
CS(J)=VXJWAD/UD(J)
DO 1049 J=1NMT
1049 AZH(J)=A2(J)*HI(J)/UD(J)
WRITE (ISOUT,2)K
WRITE (ISOUT,3)(J+1,J)+A(J),XM(J),YM(J),J=1NMT
MTWN 1779
WRITE (SOUT,4)CRAXY(K),CRAXY(N),CRAXX(K),CRAXX(N),CRAWX(K),CRAWX(N)
WRITE (SOUT,5)J0(K),J0(N),K3(K),K3(N)
105 RETURN
C
C 102 this is the testing and computing route
C 103 compute the topographic increment at position of the jth particle
C 104 compute the perturbed wind components, sum them, and add them
C 105 to the unperturbed wind vector.
C
102 AX=0.0
AZ=0.0
DO 166 I=1,NMT
C 107 the following two cards translate the particle into the mountain
C 108 coordinate system.
C
DELX*PIJ-JX(I)
DELY*PYJ-JY(I)
C
C 109 the following two cards rotate the particle into the mountain
C 110 coordinate system.
C
USX = DELX * CSSK) + VELY * SNSK)
USY = DELX * CSSK) + VELY * COSK)
Y2 = USY
X2 = USX
R2=X2+Y2
C 111 now compute top height increment resulting from the jth mountain
C 112 and add it to sum.
Z2 = Z2 + R2-111/111(A211)+R2**SRT(A2-I1)+R2)
C
C 113 compute perturbation wind increments
C
AMBD0 = ZF(J)*A(I)
AMBD2=AMBD0**AMBD1
VU%=R%A0**AMBD2
4=SRT(AMBD2)**SRT(AMBD2)
&=AEM1)*S%USX*GENUM
C
C 1061 AX; the perturbed wind component in the direction of the
C 1062 unperturbed wind.
C
1061 AX=AX+Z2-111/(Y<AMBD2-2.0-A0)/GENUM
C 1063 AX, AZ; the perturbed wind components perpendicular to the
C 1064 direction of the unperturbed wind.
C
AY=AY-W*DSY
AZ=AZ-W*AMBD0
C
C 106 NOA TEST FOR IMPACTED PARTICLE
C
IF(Z2-Z.PJ.J1100.1UB
C
118 PARTICLE HAS IMPACTED
118 AZ=AZ-1.0+UB
AX=AX+UB
AY=AY+UB
GO TO 105
C
C 109 THE PARTICLE IS ALREADY HAVING THE UNPERTURBED WIND VECTOR TO
C 109 the perturbed component in the same direction.
C
109 DELX=AX+UB
C
109 THE FOLLOWING TWO CARDS DECOMPONITE THE WIND VECTOR INTO THE
C
MAGY SYSTEM.
C
AEX=DELX+CSSK)-AY+SNK)
AY=DELY+CSSK)+AY-COSK)
GO TO 105
END
$IFTC\ RGAN1\ LIST=\ DECK=\ HIVXY2$
SUBROUTINE\ RGWIND(J,\ AX,\ AXA,\ AXA2)
C
11\ OCT\ 66
C
ITихиЛСЕРГ, ТЕХНОМЕНИЕ ТЕХНИЧЕСКИЕ ЭКСПЕРИМЕНТЫ ИНСТИТУТ ИССЛЕДОВАНИЙ INC.
C
C
THIS\ SUBROUTINE\ SERVES\ THE\ PURPOSE\ OF\ READING\ LIDGE\ WIND
C
DATA\ WHEN\ THE\ SIGN\ OF\ ARGUMENT\ J\ IS\ MINUS\ AND\ COMPUTING\ THE
C
RIGHT\ WIND\ FOR\ THE\ JTH\ PARTICLE\ AFTER\ FIRST\ CHECKING\ FOR\ IMPACT
C
ON\ THE\ ANALYTICAL\ GROUND.\ IF\ IMPACT\ IS\ FOUND\ THE\ PARTICLE\ IS
C
ASSIGNED\ A\ LARGE\ COMMAND\ VELOCITY\ COMPONENT.
C
C
This\ must\ not\ be\ the\ end\ of\ the\ main\ program.
C
C
******************************************************************************
C
C
COMMON\ /SET1/
1
C
D\ PRINT\ E\ PRINT\ E\ PRINT\ D\ PRINT
C
C
COMMON\ /SET2/
1
C
******************************************************************************
C
C
DIMENSION\ LET(12),\ ANT(12)
C
C
DIMENSION\ (...)
C
C
COMMON\ /SET1/
1
C
D\ PRINT\ E\ PRINT\ E\ PRINT\ D\ PRINT
C
C
COMMON\ /SET2/
1
C
******************************************************************************
C
C
PARAMETERS\ REGULAR\ TO\ SUBROUTINE\ RGWIND
C
C
DIMENSION\ AX(12),\ AX(12),\ AXA(12),\ AXA(12),\ AXA2(12),\ AX(12),\ AX(12),\ AXA(12),\ AXA2(12)
C
******************************************************************************
C
C
AX)\ THE\ HALF\ FLUX\ OF\ THE\ KTH\ LIDGE
C
AX\ WIND\ VECTOR\ EAST
C
C
******************************************************************************
C
C
201
IF(J1).EQ.1*U1*V1*U2
101 IRK2=101
7734 CALL ERRRTA(NOPRINT,NOPRINT
C 1.40 THIS IS THE DATA READING ROUTINE
103 DATA (I, J1) X(I,J) Y(I,J) A(I,J) B(I,J)
C(J) = COS(B(J))
J(J) = SIN(B(J))
A2(J) = A(J)*A(J)
A2H(J) = 2*A(J)*H(J)
IF(J-121).LT.1*U2*V1*U2
1036 INRR=1*U2
GO TO 7736
1031 IF(IRK2.LT.1*U2*V1*U2)
C 1.41 COMPLETE THE KIN LOCAL WIND COORD.
1.99 D2=ABS(A2H(J)1/2)
IF (D2-C0RTK(J)).LT.1*U2*V1*U2
1.99 C0RTK(J)=D2
1.44 CHECK TO SEE IF THE KIN LOCAL KINU IS WITHIN THE LIMITS OF
C THE KIN LOCAL WIND SYSTEM.
1.45 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.46 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.47 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.48 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.49 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.50 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.51 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.52 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.53 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.54 IF(D2-C0RTK(J)).LT.1*U2*V1*U2
1.55 C1.44 THE KIN LOCAL KINU IS NOT WITHIN THE LIMITS OF THE KIN LOCAL WIND SYSTEM.
1.56 IRK2=116
GO TO 7736
1.57 C 1.45 THE KIN LOCAL KINU IS WITHIN THE LIMITS OF THE KIN LOCAL WIND SYSTEM.
1.58 INRR=1*U2
GO TO 7736
1.59 C 1.50 CHECK TO SEE IF THE KIN LOCAL KINU IS LESS THAN U2
1.60 D2=X(J1,J)+Y(J1,J)
IF (D2-V0).LT.1*U2*V1*U2
1.61 IF(D2-V0).LT.1*U2*V1*U2
1.62 IF(D2-V0).LT.1*U2*V1*U2
1.63 IF(D2-V0).LT.1*U2*V1*U2
1.64 IF(D2-V0).LT.1*U2*V1*U2
1.65 IF(D2-V0).LT.1*U2*V1*U2
1.66 IF(D2-V0).LT.1*U2*V1*U2
1.67 IF(D2-V0).LT.1*U2*V1*U2
1.68 IF(D2-V0).LT.1*U2*V1*U2
1.69 IF(D2-V0).LT.1*U2*V1*U2
1.70 IF(D2-V0).LT.1*U2*V1*U2
1.71 IF(D2-V0).LT.1*U2*V1*U2
1.72 IF(D2-V0).LT.1*U2*V1*U2
1.73 IF(D2-V0).LT.1*U2*V1*U2
1.74 IF(D2-V0).LT.1*U2*V1*U2
1.75 IF(D2-V0).LT.1*U2*V1*U2
1.76 IF(D2-V0).LT.1*U2*V1*U2
1.77 IF(D2-V0).LT.1*U2*V1*U2
1.78 IF(D2-V0).LT.1*U2*V1*U2
1.79 IF(D2-V0).LT.1*U2*V1*U2
1.80 IF(D2-V0).LT.1*U2*V1*U2
1.81 IF(D2-V0).LT.1*U2*V1*U2
1.82 IF(D2-V0).LT.1*U2*V1*U2
1.83 IF(D2-V0).LT.1*U2*V1*U2
1.84 IF(D2-V0).LT.1*U2*V1*U2
1.85 IF(D2-V0).LT.1*U2*V1*U2
1.86 IF(D2-V0).LT.1*U2*V1*U2
1.87 IF(D2-V0).LT.1*U2*V1*U2
1.88 IF(D2-V0).LT.1*U2*V1*U2
1.89 IF(D2-V0).LT.1*U2*V1*U2
1.90 IF(D2-V0).LT.1*U2*V1*U2
1.91 IF(D2-V0).LT.1*U2*V1*U2
1.92 IF(D2-V0).LT.1*U2*V1*U2
1.93 IF(D2-V0).LT.1*U2*V1*U2
1.94 IF(D2-V0).LT.1*U2*V1*U2
1.95 IF(D2-V0).LT.1*U2*V1*U2
1.96 IF(D2-V0).LT.1*U2*V1*U2
1.97 IF(D2-V0).LT.1*U2*V1*U2
1.98 IF(D2-V0).LT.1*U2*V1*U2
1.99 IF(D2-V0).LT.1*U2*V1*U2
203
DO 1, 49 J = 1, NRC
SSG = CS(K) * R(J) + G(K) * C(J)
CGG = CS(K) * C(J) - G(K) * S(J)
AH(J) = -A(J) * H(J) * UU(NVCC) * CGG
SG(J) = SSG / CGG
1349 C(J) = 2.0 / CGG
C
WRITE (10,OUAT.A) 10
WRITE (10,OUAT.A) 10
WRITE (10,OUAT.A) 10
WRITE (10,OUAT.A) 10
WRITE (10,OUAT.A) 10
145 RETURN
C
C 1.4 Inserting the Turbulent Wind Correlation Code
C
C Compute the Turbulent Increment in Position of the Particle
C
C Compute the Perturbation Wind Component and Add it and Modify
C
C the Unperturbed Wind Vector
C
122 AX = ...
AY = ...
AZ = ...
J = J + 1
C
C This code transforms one particle into the other
C
C with respect to the point without distance of the wind
C
C PARTICLE FROM THE UNPERTURBED
C
127 AU = VX(J) - VX(1)
AV = VX(J) - VX(1)
AW = VX(J) - VX(1)
C
C Now compute the new increment resulting from the Turbulence
C
C and add it to wind
C
C UX = UX + PARTICLE(1) + AU
C
C The following code computes the perturbation wind increments
C
C PARTICLE - UNPERTURBED
C
C THE COMPONENT IN THE DIRECTION OF THE UNPERTURBED WIND
C
C AX = CX(J) + AX + AX
C
C AZ = CZ(J) + AZ + AZ
C
C Use test for impacted particle
C
C IF(AZ < 2.0) THEN
C
C PARTICLE is impacted
C
C AZ = AZ + 4.0
C
C AX = ...
C
C AY = ...
C
C GO TO 1,6
C
C THE PARTICLE IS ADDED, NOW MODIFY THE UNPERTURBED WIND VECTOR TO
C
C THE PERTURBED COMPONENT IN THE SAME DIRECTION
C
C THE FOLLOWING two steps combine the wind system into macro field
C
C COORDINATES
C
C AX = AX * CRK - AR * SIN(K)
C
C AY = AX * SIN(K) + AY * COS(K)
C
C GO TO 105
C
END
C1 = ACUT4 + ACUSM
C1 = ASIT4 + ASIM2
C2 = ACUT4 + ACUSM
C2 = ASIT4 + ASIM2
IF(C1 < 0) GOTO 1000

1000 GO TO 5050

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050 C1 = C1 + 1

1000 GO TO 1000

200 GO TO 5050

5050
**SUBROUTINE RETRIEVE TOPO HEIGHT AT HORIZONTAL COORDINATES**

**DIMENSION DELT(G),XXY(Y)***
C SPECIFICATION AREA: m 10 set 2000.0010
9 HP=1000000
GO TO 101
C PARThCLE IS BEYOND TOP SPECIFICATION AREA: m 10 set 2000000
10 HP=2000000
GO TO 111111
11 WHILE(ISW=13)ALN
GO TO 28
C FOR PARTICLE OVER TOP SPECIFICATION IN CONSTANT Cbounce of number
C TO RELOCATE PARTICLE TO CENTER OR DIVIDED AREA
C BLOCK IN CODE
12 IF(MAX-DELAY=1)ALN
JUMP TO ACTIVITY
C TO TOP RELOCATION AREA OR SUBDIVIDED TO OBTAIN IN ORIGIN
C NOT SUBDIVIDED NEED NO IF JUMP
MAX(1,J)
IF(HA12=28,28)
C ORIGIN OF PARTICLE COORD TO CENTER OR SUBDIVIDED AREA
C SEEK PARTICLE
13 CJ=1
CJ
X=AX+BALL-(aI-1)+0.5
X=AY+BALL-(aI-1)+0.5
GO TO 28
IF CY+CAH INTO 4 QUADRANTS
SINGLE ORIGIN TO CORNER OR DIVIDED AREA
C ORIGIN-ORIGIN
CY=FLX-ORIGIN
CX=FAY-ORIGIN
C SEEK NEXT HERE
C ORIGIN NEED TO GIVE INDEX FOR SUBDIVIDED
14 KS=HA+1
C ORIGIN QUADRANT TO PARTICLE OVER
IF (CCX)(11,10,11) KS
15 IF(CCLY)=10,11 KS
16 CCX=ORIGIN
C CY OF ORIGIN
CX(=ORIGIN)
GO TO 20
17 KS=K+1
N=4
GO TO 20
18 KS=K+2
N=3
GO TO 20
19 N=1
GO TO 20
C IF TOP HI IF QUADRANT NOT SUBDIV IN SUBDIVIDED NEED NO IF OLY
20 IF(SUBSIDK)
IF(M)212828
C SUBDIVIDE QUADRANT
21 ORIGIN=ORIGIN/20
C ORIGIN OF PARTICLE COORD TO CENTER OF SUBDIVIDED QUADRANT
C PARTICLE IS OVER
GO TO 20
C CY=CCX+ORIGIN
GO TO 13
C CCY=CCY-ORIGIN
GO TO 13
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.
TRANSPORT MODULE

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

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

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

**** 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, ICIJL, HAS BEEN GIVEN THE FOLLOWING VALUES.
1 1 1 0 0 0 1 1 0 0 0 0 0 0 0

THE TRANSPORT TIME LIMIT IS 86400.000

PARTICLE DATA

DENSITY OF Fallout Particles 2600.000 KG/****
IPARIV 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 RUAD B STATIONS, 12/20/66
<table>
<thead>
<tr>
<th>HEIGHT OF BOTTOM OF STRATUM</th>
<th>VISCOSITY (MKS)</th>
<th>DENSITY (MKS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.110000E+04</td>
<td>0.182000E+04</td>
<td>0.134000E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.161440E+04</td>
<td>0.132190E+01</td>
</tr>
<tr>
<td>0.700000E+03</td>
<td>0.180800E+04</td>
<td>0.129720E+01</td>
</tr>
<tr>
<td>0.500000E+03</td>
<td>0.190190E+04</td>
<td>0.127200E+01</td>
</tr>
<tr>
<td>0.300000E+03</td>
<td>0.170870E+04</td>
<td>0.124970E+01</td>
</tr>
<tr>
<td>0.100000E+03</td>
<td>0.199140E+04</td>
<td>0.122700E+01</td>
</tr>
<tr>
<td>0.300000E+03</td>
<td>0.183430E+04</td>
<td>0.120460E+01</td>
</tr>
<tr>
<td>0.600000E+03</td>
<td>0.189010E+04</td>
<td>0.118200E+01</td>
</tr>
<tr>
<td>0.700000E+03</td>
<td>0.192300E+04</td>
<td>0.116040E+01</td>
</tr>
<tr>
<td>0.400000E+03</td>
<td>0.196240E+04</td>
<td>0.113920E+01</td>
</tr>
<tr>
<td>0.110000E+04</td>
<td>0.174520E+04</td>
<td>0.112600E+01</td>
</tr>
<tr>
<td>0.130000E+04</td>
<td>0.193130E+04</td>
<td>0.110320E+01</td>
</tr>
<tr>
<td>0.110000E+04</td>
<td>0.122630E+04</td>
<td>0.108000E+01</td>
</tr>
<tr>
<td>0.110000E+04</td>
<td>0.141000E+04</td>
<td>0.105760E+01</td>
</tr>
<tr>
<td>0.190000E+04</td>
<td>0.130400E+04</td>
<td>0.103520E+01</td>
</tr>
<tr>
<td>0.200000E+04</td>
<td>0.160000E+04</td>
<td>0.101240E+01</td>
</tr>
<tr>
<td>0.220000E+04</td>
<td>0.170500E+04</td>
<td>0.098960E+01</td>
</tr>
<tr>
<td>0.220000E+04</td>
<td>0.170940E+04</td>
<td>0.096680E+01</td>
</tr>
<tr>
<td>0.230000E+04</td>
<td>0.170160E+04</td>
<td>0.094400E+01</td>
</tr>
<tr>
<td>0.130000E+04</td>
<td>0.175670E+04</td>
<td>0.092120E+01</td>
</tr>
<tr>
<td>0.130000E+04</td>
<td>0.174980E+04</td>
<td>0.090840E+01</td>
</tr>
<tr>
<td>0.140000E+04</td>
<td>0.174310E+04</td>
<td>0.089560E+01</td>
</tr>
<tr>
<td>0.400000E+03</td>
<td>0.192520E+04</td>
<td>0.087280E+01</td>
</tr>
<tr>
<td>0.400000E+03</td>
<td>0.191840E+04</td>
<td>0.085000E+01</td>
</tr>
<tr>
<td>0.400000E+03</td>
<td>0.191160E+04</td>
<td>0.082720E+01</td>
</tr>
<tr>
<td>0.400000E+03</td>
<td>0.190480E+04</td>
<td>0.080440E+01</td>
</tr>
<tr>
<td>0.500000E+03</td>
<td>0.187960E+04</td>
<td>0.078160E+01</td>
</tr>
<tr>
<td>0.500000E+03</td>
<td>0.187280E+04</td>
<td>0.075880E+01</td>
</tr>
<tr>
<td>0.500000E+03</td>
<td>0.186600E+04</td>
<td>0.073600E+01</td>
</tr>
<tr>
<td>0.600000E+03</td>
<td>0.183240E+04</td>
<td>0.071320E+01</td>
</tr>
<tr>
<td>0.600000E+03</td>
<td>0.182560E+04</td>
<td>0.069040E+01</td>
</tr>
<tr>
<td>0.600000E+03</td>
<td>0.181880E+04</td>
<td>0.066760E+01</td>
</tr>
<tr>
<td>0.700000E+03</td>
<td>0.178470E+04</td>
<td>0.064480E+01</td>
</tr>
<tr>
<td>0.700000E+03</td>
<td>0.177790E+04</td>
<td>0.062200E+01</td>
</tr>
<tr>
<td>0.700000E+03</td>
<td>0.177110E+04</td>
<td>0.060020E+01</td>
</tr>
<tr>
<td>0.800000E+03</td>
<td>0.173700E+04</td>
<td>0.057740E+01</td>
</tr>
<tr>
<td>0.800000E+03</td>
<td>0.173020E+04</td>
<td>0.055460E+01</td>
</tr>
<tr>
<td>0.800000E+03</td>
<td>0.172340E+04</td>
<td>0.053180E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.168930E+04</td>
<td>0.050900E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.168250E+04</td>
<td>0.048620E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.167570E+04</td>
<td>0.046340E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.166900E+04</td>
<td>0.044060E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.166220E+04</td>
<td>0.041780E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.165540E+04</td>
<td>0.039500E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.164870E+04</td>
<td>0.037220E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.164200E+04</td>
<td>0.034940E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.163530E+04</td>
<td>0.032660E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.162860E+04</td>
<td>0.030380E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.162190E+04</td>
<td>0.028100E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.161520E+04</td>
<td>0.025820E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.160850E+04</td>
<td>0.023540E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.160180E+04</td>
<td>0.021260E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.159510E+04</td>
<td>0.018980E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.158840E+04</td>
<td>0.016700E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.158170E+04</td>
<td>0.014420E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.157500E+04</td>
<td>0.012140E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.156830E+04</td>
<td>0.009860E+01</td>
</tr>
<tr>
<td>0.900000E+03</td>
<td>0.156160E+04</td>
<td>0.007580E+01</td>
</tr>
</tbody>
</table>
### Wind Components

**Level 1**

<table>
<thead>
<tr>
<th>Level</th>
<th>Base Alt</th>
<th>Min. 400 Meters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Level 2**

<table>
<thead>
<tr>
<th>Level</th>
<th>Base Alt</th>
<th>1000.400 Meters</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

224
<table>
<thead>
<tr>
<th>LEVEL 35</th>
<th>BASE AT 009.200 Meters</th>
</tr>
</thead>
<tbody>
<tr>
<td>EAST-WEST</td>
<td>1</td>
</tr>
<tr>
<td>2.97510</td>
<td>2.54793</td>
</tr>
<tr>
<td>0.00060</td>
<td>0.00076</td>
</tr>
<tr>
<td>EAST-WEST</td>
<td>2</td>
</tr>
<tr>
<td>2.36341</td>
<td>2.65541</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>EAST-WEST</td>
<td>3</td>
</tr>
<tr>
<td>2.60449</td>
<td>2.71120</td>
</tr>
<tr>
<td>0.06195</td>
<td>0.00000</td>
</tr>
<tr>
<td>EAST-WEST</td>
<td>4</td>
</tr>
<tr>
<td>2.39411</td>
<td>2.13847</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>EAST-WEST</td>
<td>5</td>
</tr>
<tr>
<td>2.77441</td>
<td>1.51753</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
</tr>
<tr>
<td>EAST-WEST</td>
<td>6</td>
</tr>
<tr>
<td>0.00000</td>
<td>0.00000</td>
</tr>
</tbody>
</table>

DAVIES LUTHER AS YADDAUKEAR FOR 1.2473.701 MELKINS AT 941.047 METERS

DAVIES LUTHER AS YADDAUKEAR FOR 1.2473.701 MELKINS AT 942.099 METERS

DAVIES LUTHER AS YADDAUKEAR FOR 1.2473.701 MELKINS AT 941.020 METERS

DAVIES LUTHER AS YADDAUKEAR FOR 1.2473.701 MELKINS AT 944.709 METERS
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>150</td>
<td>2</td>
<td>50</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**TRANSPORT IS COMPLETE. INTERMEDIATE RESULTS ARE ON TAPE 9**

**ENTERING LINK 5**
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, $\mathbf{v}_0$, which would otherwise exist in the absence of the ground disturbance. The relationship between the change in the wind field $\Delta \mathbf{v}(x, y, z)$ and the curvature of the terrain is deduced by first deriving the dispersion relationship for the system (which connects the vertical attenuation constant of the velocity field to the periodicity of the ground structure) and subsequently applying the boundary condition that the surface wind trajectory be parallel to the terrain. The resulting expressions become greatly simplified for short wavelengths and when the Coriolis effect is neglected. However, for most practical cases involving tropospheric fallout, the foregoing restrictions are not severe since sounding stations are presumed to exist at reasonable distances from each other.
From a theoretical point of view, this investigation is a modified extension of the earlier work of Queney, A.1, A.2, but there are differences which render somewhat different results. Although both models utilize perturbation theory to include the effects of variable terrain, there is a distinct conceptual difference between them arising from the choice of the dependent variables. Queney deals with the displaced trajectories of the streamlines as the fundamental physical quantities of interest (which seems to introduce extra degrees of complexity into the problem), while we treat the changes in the velocity field. Our method of attack permits more refined criteria for establishing the validity of the calculation and leads quite naturally to a generalization to three-dimensional systems, which are more frequently encountered than the two-dimensional idealizations of Queney. Moreover, we show that a perturbation theory model for the hydrodynamics does not necessarily imply the applicability of superposition of ground disturbances, a result which does not seem to have been recognized earlier. This is a distinct problem. However, we are able to demonstrate that the superposition hypothesis can serve as the basis of an iterative scheme for computing the velocity field to an arbitrary degree of accuracy consistent with the initial premises of the perturbation method. In certain two-dimensional cases, there does not appear to be much difference between Queney's results and ours.

The overall validity of the model is based upon the applicability of the 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 $\epsilon$ denotes the angle between the local west-east direction and the unperturbed wind velocity, then the components of $\Omega$ are given by

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

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

Theory of Airflow

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

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

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

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

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

229
where \( k \) is the Boltzmann constant \((k = 1.38 \times 10^{-16} \text{ erg deg}^{-1})\), and \( m \) is the mass of the air molecule. The two equations which describe the aerodynamics are the continuity equation and the momentum equation:

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0
\]  
and

\[
\frac{\partial \mathbf{v}}{\partial t} = \frac{\rho}{\rho e} \mathbf{g} - \nabla \psi + \mathbf{Q},
\]

where \( \mathbf{Q} \) is the gravity force and is equal to \(-G\xi\), and \( \psi \) is a potential obtained by combining the \((1/\rho)\nabla P\) term with Eq. (A.1).

\[
\psi = \left(\frac{P_e}{\rho e}\right)^\gamma [\gamma/(\gamma - 1)] \rho^{(\gamma - 1)}.
\]

We assume that a steady state exists in which there is only one uniform (spatially homogeneous) component of velocity, \( u_0 \), which, by construction, is parallel to the \( x \) direction. The system of equations then reduces to

\[
0 = -\psi_x, \quad 0 = -2u_0 \Omega_z - \psi_y, \quad 0 = 2u_0 \Omega_y - \psi_z - G.
\]  

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

\[
\psi = Ay + Bz + \psi_{or},
\]

which when substituted into the foregoing equations gives

\[
A = -2u_0 \Omega_z, \quad B = -G + 2u_0 \Omega_y \approx -G,
\]

\[
\psi_{or} = \gamma/(\gamma - 1) \quad \frac{P_e}{\rho e} = \gamma/(\gamma - 1) \quad \left(\frac{kT_e}{m}\right) = c_s^2/\gamma \quad (A.9)
\]

where \( c_s \) is the speed of sound and equals \(3.4 \times 10^4\) cm sec\(^{-1}\) under STP conditions.
A measure of the distances over which changes in $\psi$ are important in the equilibrium case can be determined by examining the ratios $\psi_{or}/|A| = y_c$ and $\psi_{or}/|B| = z_c$, which are respectively the distances over which the independent changes in $\psi$ equal the value at the origin. We have

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

and

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

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

$$\chi = iu_o \quad (A.12)$$

and density

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

where

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

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

$$\rho_o(z) \approx \rho_e \left\{1 - \left[\frac{z}{(\alpha/(\gamma - 1))}\right]\right\} = \rho_e (1 - \beta z) \approx \rho_e e^{-\beta z}, \quad (A.15)$$
where β is defined as the tropospheric density attenuation constant,

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

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

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

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

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

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

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

and

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

For mathematical convenience, it is desirable to deal with a function, \( \eta \), related to the initial density \( \rho_o \) by the formula

\[ \eta = \bar{\rho} / \rho_o . \quad (A.22) \]
In terms of $\eta$, $\tilde{\psi}$ is given by

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

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

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

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

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

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

and

$$\frac{\partial \tilde{\psi}}{\partial z} = \eta \left( \gamma - 1 \right) - \left( \frac{\gamma k T_e}{m} \right) (1 - \alpha z) \frac{\partial \eta}{\partial z} . \quad (A.27)$$

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 $(1 - \alpha z) (1 - z/z_c) \approx 1$. Also, within this range, $1n\rho_o = 1n\rho_e - \beta z$, thereby yielding

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

$$u \frac{\partial \tilde{u}}{\partial x} = 2 \left( \tilde{\omega} \Omega_z - \tilde{\omega} \Omega_y \right) - \frac{\gamma k T_e}{m} \frac{\partial \eta}{\partial x} , \quad (A.29)$$

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

\[ u = \int A(k) e^{ik \cdot \xi} \, d^3k, \quad \bar{v} = \int B(k) e^{ik \cdot \xi} \, d^3k, \]

and

\[ w = \int C(k) e^{ik \cdot \xi} \, d^3k, \quad \eta = \int D(k) e^{ik \cdot \xi} \, d^3k, \]

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

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

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 \( k \) can be preassigned. It is at this point that the pertinent physical factors are introduced into the problem. Since the topography can be resolved into periodic components of \( x \) and \( y \), we must necessarily regard \( k_x \) and \( k_y \) as real numbers. The dispersion relationship is then interpreted as

\[ k_z = k_z' \left( k_x', k_y' \right). \quad (A.34)\]
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, \(\tilde{w}\) now becomes

\[
\tilde{w}(x, y, z) = \sum_{\mu=1,2} \int \int C_{\mu}(k_x, k_y) e^{ik_x x} e^{ik_y y} e^{ik_z z} dk_y dk_x,
\]

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

\[
\alpha k_z^2 + b(ik_z) + c = 0,
\]

where

\[
a = \sigma \left( k_x^2 u_x^2 - \Omega^2 \omega_x^2 \right),
\]

\[
b = \sigma \left[ \gamma k_x^2 u_x^2 + 2\Omega k_x k_u \omega_x + \Omega^2 \left( 2ik_x \omega_z + 2ik_y \omega_y + \gamma \beta \omega_z^2 \right) \right],
\]

and

\[
c = k_x^2 u_x^2 \left[ \sigma \left( k_x^2 + k_y^2 \right) - u_o^2 k_x^2 + (1 - \gamma) \sigma \beta^2 \right] + \Omega \beta \sigma \left[ 2k_x^2 \omega_u u_o - \gamma \left( \omega_y u_o k_x^2 - \omega_x u_o k_y \right) \right] + \Omega^2 \left[ 4k_x^2 u_x^2 - \sigma (k_x \omega_x + k_y \omega_y)^2 - \omega_z^2 (1 - \gamma) \sigma \beta^2 - \beta \sigma \gamma (ik_x \omega_y + ik_y \omega_x) \right],
\]

in which

\[
\omega_x = 2\Omega_x /\Omega, \quad \omega_y = 2\Omega_y /\Omega, \quad \omega_z = 2\Omega_z /\Omega, \quad \sigma = (G/\beta).
\]
Since Eq. A.33, is homogeneous, the absolute magnitudes of the functions $A(k), B(k), C(k)$, and $D(k)$ cannot be determined; only their relationship to each other, as deduced from the boundary conditions, can. For mathematical convenience it is desirable to eliminate $D(k)$ and deal only with the Fourier transforms of the velocity components of the wind velocity. Thus, we obtain the equation

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

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

$$
\begin{align*}
  b_{11} &= i k_x \xi + i u_0 x \Omega_y, \\
  b_{12} &= i k_y \xi - i u_0 x \Omega_x, \\
  b_{13} &= (i k_z - \beta) \xi + i u_0 k_x, \\
  b_{21} &= (i k_u y_0) \xi + i k_x \sigma \Omega_y, \\
  b_{22} &= -\Omega_z \xi - (i k_y \sigma) \Omega_y, \\
  b_{23} &= \Omega_y \xi - (i k_x \sigma) i k_x y_0, \\
  b_{31} &= \Omega_x \xi + (i k_y \sigma) \Omega_x, \\
  b_{32} &= i k_y \xi - (i k_x \sigma) \Omega_x, \\
  b_{33} &= -\Omega_x \xi + k_y k_x y_0.
\end{align*}
$$

in which $\xi = (1 - \gamma) G + (ik_x \sigma)$ 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:

$$
A = \frac{b_{12} b_{33} - b_{13} b_{32}}{b_{12} b_{31} - b_{11} b_{32}} C \equiv T_{\{k_x, k_y\}, \{k_x, k_y\}},
$$

and

$$
B = \frac{b_{13} b_{31} - b_{11} b_{33}}{b_{12} b_{31} - b_{11} b_{32}} C \equiv U_{\{k_x, k_y\}, \{k_x, k_y\}}.
$$

236
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

$$\sigma k_z^2 + (ik_z) \sigma \gamma \beta + \sigma \left( k_x^2 + k_y^2 \right) - u_0^2 k_x^2 + (1 - \gamma) \sigma \beta^2 = 0 \ . \quad (A.42)$$

We now let

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

anticipating an exponential decay with altitude. Since $\sigma = G/\beta = 1.2 \times 10^9 \text{cm}^2\text{sec}^{-2} >> u_0^2$, we can neglect $u_0^2 k_x^2$ in Eq. (A.42) and thus obtain

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

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

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

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

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

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

$$T(k_x, k_y) = -ik_x \left( k_x^2 + k_y^2 \right)^{1/2} \ , \quad (A.47)$$
and
\[ U(k_x, k_y) = -\frac{ik}{(k_x^2 + k_y^2)^{1/2}}. \] (A.48)

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

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

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

and

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

The function \( C(k_x, k_y) \) is determined by application of the perturbed boundary condition. Let
\[ \phi(x, y, z) = 0 = z - f(x, y) \] (A.52)
be the equation of the earth's surface. The normal to this surface is \( \nabla \phi \):
\[ \nabla \phi = -\frac{\partial f}{\partial x} - \frac{\partial f}{\partial y} + k. \] (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
\[ [ (\mathbf{v} \cdot \nabla \phi) ] = 0 \] along \( z = f(x, y) \) , (A.54)
where
\[ \mathbf{v} = \mathbf{j}u + \mathbf{j}v + \mathbf{k}w = \mathbf{j}u_0 + (\mathbf{j}\mathbf{u} + \mathbf{j}\mathbf{v} + \mathbf{k}\mathbf{w}) . \]  
(A.55)

Inserting Eq. (A.55) into Eq. (A.54) gives
\[ \mathbf{w}(x, y, z = f) = u_0 \left( \frac{\partial f}{\partial x} \right) + u(x, y, z = f) \left( \frac{\partial f}{\partial x} \right) + \mathbf{v}(x, y, z = f) \left( \frac{\partial f}{\partial y} \right) . \]  
(A.56)

Using Eqs. (A.47)-(A.51) in Eq. (A.56) yields the following integral equation for \( C(k_x, k_y) \):
\[
\int \int \left[ 1 - T(k_x, k_y) \left( \frac{\partial f}{\partial x} \right) - U(k_x, k_y) \left( \frac{\partial f}{\partial y} \right) \right]
\]
\[ C(k_x, k_y) e^{ik_x x} e^{ik_y y} \left[ \left( k_x^2 + k_y^2 \right)^{1/2} f(x, y) \right] dk_x dk_y \]
\[ = u_0 \int \int (ik_x) F(k_x, k_y) e^{ik_x x} e^{ik_y y} dk_x dk_y , \]  
(A.57)

where
\[ F(k_x, k_y) = \left( \frac{1}{2\pi} \right)^2 \int \int f(x, y) e^{-ik_x x} e^{-ik_y y} dx dy , \]  
(A.58)

and
\[ f(x, y) = \int \int F(k_x, k_y) e^{ik_x x} e^{ik_y y} dk_x dk_y . \]  
(A.59)

The solution for \( C(k_x, k_y) \) is impossible to achieve by direct means because of the dependence of the integration of the left-hand side of Eq. (A.57) on \( f(x, y) \) and on
the derivatives of \( f(x,y) \). However, a systematic perturbation method for computing \( C(k_x, k_y) \) can be deduced. The approximation to \( C(k_x, k_y) \), achieved by setting 
\[
\exp \left[ - \left( \frac{k_x^2 + k_y^2}{2} \right)^{1/2} f(x,y) \right]
\]
equal to unity, is equivalent to assuming that the maximum elevation, \( f_{\text{max}} \), 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 \( \tilde{u} (\delta f/\delta x) \) as compared to \( u_o (\delta f/\delta x) \) 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 \( \tilde{u} \) to \( u_o \). The neglect of \( \tilde{v} (\delta f/\delta y) \) as compared to \( u_o (\delta f/\delta x) \) is somewhat difficult to justify under all cases. Although \( \tilde{v} \) is assumed small compared to \( u_o \), we must also be sure that \( (\delta f/\delta y) \) is not substantially greater than \( (\delta f/\delta x) \).

The apriori assumption

\[
\tilde{u} \left( \frac{\partial f}{\partial x} \right) + \tilde{v} \left( \frac{\partial f}{\partial y} \right) < u_o \left( \frac{\partial f}{\partial x} \right) \quad (A.59),
\]
is equivalent to neglecting \(-T(\delta f/\delta x) - U(\delta f/\delta y)\) 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

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

Introduction of the functions

\[
\xi = 1 - \Gamma = - \sum_{n=1}^{\infty} \frac{\phi^n}{n!}, \quad (A.61)
\]

where

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

240
and

\[ \tau = T(k_x, k_y) \left( \frac{\partial f}{\partial x} \right) + U(k_x, k_y) \left( \frac{\partial f}{\partial y} \right) \]  

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

\[ \int C(k) e^{ik \cdot r} dk = u_o \int H(k) e^{ik \cdot r} dk + \int \Delta C(k) e^{ik \cdot r} dk. \]  

where

\[ \Delta = (\tau + \xi - \xi \tau) = \Delta(k, x, y), \quad H(k) = ik_x F(k), \quad \xi = ik_x + ik_y, \quad dk = dk_x dk_y. \]

Multiplying Eq. (A.64) by \( e^{-ik' \cdot r} \), and then integrating over \( r \), gives

\[ C(k') = u_o H(k') + \left( \frac{1}{2\pi} \right)^2 \int \int \Delta(k, \xi) C(k) e^{i(k - k') \cdot \xi} dk dk'. \]  

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

\[ C^{(1)}(k') = u_o H(k') = u_o (ik') F(k') \]  

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

\[ C^{(2)}(k') = u_o H(k') + \left( \frac{1}{2\pi} \right)^2 \int \int \Delta(k, \xi) \left[ u_o H(k) \right] e^{i(k - k') \cdot \xi} dk dk'. \]  

Since 241
It follows by inspection that the nth approximation to \( C(k) \) is given by

\[
C^{(n)}(k') = u_o H(k') + \left( \frac{1}{2\pi} \right)^2 \int \int \Delta(k_k, \xi) C^{(n-1)}(k) e^{i(k_k-k') \cdot \xi} \, d\xi \, dk,
\]

(A.68)

with

\[
C^{(0)} = 0.
\]

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

\[
-\frac{u^2}{2} + \frac{v^2}{2} + \frac{w^2}{2} < \frac{u_o^2}{2}.
\]

(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 \( \bar{u}(\xi), \bar{v}(\xi), \) and \( \bar{w}(\xi) \).

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

Let

$$f(x) = h e^{i k_0 x},$$

(A.70)

from which we have:

$$F(k) = \frac{h}{2\pi} \int e^{i k_0 x} e^{-i k x} \, dx = h\delta(k - k_0)$$

(A.71)

and

$$C^{(1)}(k) = i k_0 h\delta(k - k_0) = i u_o(k h) \delta(k - k_0),$$

(A.72)

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

$$\bar{w}(x, z) = \int i k_0 h\delta(k - k_0) e^{i k x} e^{-|k| z} \, dk = i k_0 h u_0 e^{i k_0 x} e^{-|k_0| z}.$$  

(A.73)

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

$$\bar{u}(x, z) = + u_0(\sqrt{|k_0| h}) e^{i k_0 x} e^{-|k_0| z}.$$  

(A.74)

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

$$(2)^{1/2}(|k_0| h) \ll 1,$$

(A.75)

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

$$C^{(2)}(k') = i u_o(k h) \delta(k' - k_0) + \frac{1}{2\pi} \int \int \Delta(k, x) \left[ i u_o(k h) \delta(k - k_0) \right] e^{i(k - k')} \, dk \, dx,$$

(A.76)

243
where
\[
\Delta(k, x) = T(k) \left( \frac{\partial f}{\partial x} \right) - \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k| f)^n + T(k) \left( \frac{\partial f}{\partial x} \right) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k| f)^n.
\]

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

\[
C^{(2)}(k') = iu_o(k_0 h) \delta(k' - k_o) + iu_o(k_0 h)^2 \delta(k' - 2k_o) \text{sgn}(k_o)
\]
\[-iu_o(k_0 h) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k_o h|^n \delta(k' - (n+1)k_o)) \]
\[+ iu_o(k_0 h)^2 \text{sgn}(k_o) \sum_{n=1}^{\infty} \frac{(-1)^n}{n!} (|k_o h|^n \delta(k' - (n+2)k_o)).
\]

where
\[
\text{sgn}(k_o) = \frac{|k_o|}{k_o}.
\]

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

\[
\bar{w}(x, z) = iu_o(k_0 h) \exp(ik_o x - |k_o| z) \left[ 2 - \exp \left( -|k_o| h \exp(ik_o x - |k_o| z) \right) \right]
\]
\[+ \text{sgn}(k_o) iu_o(k_0 h)^2 \exp \left( 2(ik_o x - |k_o| z) \exp(ik_o x - |k_o| z) \right). \]

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 \((k_0 h)^2\) 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 \( \gamma \) with respect to the direction of flow, as shown in Figure A.1.
A suitable mathematical representation of a mountain ridge when viewed along the y' axis has been deduced by Queney\textsuperscript{A.2} who showed that the topography could be represented by the equation

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

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

\[ x = x' \cos \gamma - y' \sin \gamma, \quad y = x' \sin \gamma + y' \cos \gamma \quad (A.81) \]

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

\[ f(x, y) = f'[x'(x, y), y'(x, y)] \quad , \quad (A.82) \]

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

\[ F(k_x, k_y) = \left( \frac{1}{2\pi} \right)^2 \int \int f(x, y) e^{-i k_x x - i k_y y} \, dx \, dy \quad . \quad (A.83) \]

However, since \( k \cdot x \) is invariant under an orthogonal transformation, we have

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

where

\[ k'_x = k_x \cos \gamma - k'_x \sin \gamma, \quad k'_y = k_x \sin \gamma + k'_y \cos \gamma \quad . \quad (A.85) \]

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

\[ F'(k'_x, k'_y) = \left( \frac{ah}{2} \right)^2 e^{-|k'_x|a} \delta(k'_y) \quad . \quad (A.86) \]
Using the inversion formula with \( C(k_x, k_y) = i k_x u_0 F(k_x, k_y) \) yields the following expression for \( \bar{w}(x, y, z) \):

\[
\bar{w}(x, y, z) = u_0 \int \int (i k_x) e^{i k_x x} e^{i k_y y} F(k_x, k_y) e^{-\sqrt{k_x^2 + k_y^2}} z \left( k_x^2 + k_y^2 \right)^{1/2} dk_x dk_y
\]

(A.87)

\[
= u_0 \frac{\partial}{\partial x} \int \int e^{i (k'_x x' + k'_y y')} F'(k'_x, k'_y) e^{-\sqrt{k'_x^2 + k'_y^2}} z \left( k'_x^2 + k'_y^2 \right)^{1/2} dk'_x dk'_y
\]

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

\[
\bar{w}(x, y, z) = -2u_0 \frac{\partial}{\partial x} \varepsilon \cos \gamma \left( \frac{x'}{x'^2 + \lambda^2} \right)^2 ,
\]

(A.88)

where

\[
x' = x \cos \gamma + y \sin \gamma,
\]

and

\[
\lambda = z + a .
\]

(A.89)

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

\[
\bar{u}(x, y, z) = -u_0 \frac{\partial}{\partial x} \varepsilon \cos \gamma \left[ \left( x'^2 - \lambda^2 \right)/\left( x'^2 + \lambda^2 \right)^2 \right] ,
\]

(A.90)

and

\[
\bar{v}(x, y, z) = -u_0 \frac{\partial}{\partial y} \varepsilon \sin \gamma \left[ \left( x'^2 - \lambda^2 \right)/\left( x'^2 + \lambda^2 \right)^2 \right] .
\]

(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

\[
\left( \frac{|\Delta y|}{u_0} \right) = \left( \tilde{\nu}^2 + \tilde{w}^2 \right)^{1/2} / u_0 = r = ah / \left[ x^2 + (z + a)^2 \right], \quad (A.92)
\]

where the altitude \( z \) is defined in the range \( z > f(x) = h/(1 + (x/a)^2) \). 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

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

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

\[
r_b = a|h| / (a - |h|)^2 , \quad (A.94)
\]

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

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

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

\[
F(k_x, k_y) = \frac{A}{(2\pi)^2} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} e^{-ik \cdot \xi} \frac{1}{\left(\frac{a^2 + r^2}{a^2 + r^2}\right)^{3/2}} \, dx \, dy ,
\]

\[
F(k_x, k_y) = F(k) = \frac{A}{(2\pi k)^{1/2}} \int_{0}^{\infty} \frac{r^{1/2}}{\left(\frac{a^2 + r^2}{a^2 + r^2}\right)^{3/2}} J_0(kr)(kr)^{1/2} \, dr , \quad (A.96)
\]

where

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

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

\[
F(k) = \frac{A}{2\pi a} e^{-ak} . \quad (A.97)
\]
If \( u_0 \) is the unperturbed velocity, it follows that the first-order correction to the vertical component of the wind is given by

\[
\bar{w}(x, y, z) = u_0 \int \int (ik_x) F(k) e^{i(k_x x + k_y y)} e^{-kz} dk_x dk_y,
\]

\[
\bar{w}(x, y, z) = \frac{u_0 A}{a} \frac{\partial}{\partial x} \int_0^\infty e^{-(a + z)k} J_0(kr) dk,
\]

\[
\bar{w}(x, y, z) = \frac{-3\lambda A u_0}{a} \frac{x}{(x^2 + y^2 + \lambda^2)^{5/2}} = \frac{-3\lambda a^2 hx u_0}{(r^2 + \lambda^2)^{5/2}}, \quad (A.98)
\]

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:

\[
\tilde{u}(x, y, z) = -u_0 \int \int \frac{(ik_x)(ik_y)}{k} F(k) e^{i(k_x x + k_y y)} e^{-kz} dk_x dk_y
\]

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

\[
\tilde{u}(x, y, z) = u_0 (a^2 h) \frac{(y^2 + \lambda^2 - 2x^2)}{(r^2 + \lambda^2)^{5/2}}; \quad (A.99)
\]
Wind Streamlines and Fallout Particle Streamlines in TwoDimensions

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 \( u \) and \( w \) denote the horizontal and vertical components of the terrain effects, \( \mathbf{v} \) 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

\[
\mathbf{v}(x, y, z) = -u_0 \int \int \left( \frac{ik_y}{k} \right) \left( \frac{ik_x}{k} \right) F(k) e^{i(k_x x + k_y y)} e^{-kz} \, dk_x \, dk_y ,
\]

\[
\mathbf{v}(x, y, z) = -u_0 \frac{\partial^2}{\partial x \partial y} \int \int_0^\infty F(k) e^{ikr \cos \phi} e^{-kz} \, dk \, d\phi .
\]

\[
\mathbf{v}(x, y, z) = -\frac{3u_0 (\alpha \phi)^{3/2}}{(r^2 + \lambda^2)^{5/2}} .
\]

(A.100)

The flow lines, or trajectories, of fallout particles can be determined from a quantity called the stream function, \( \Phi(x, z) \), defined by the partial differential equations

\[
u_p = u_0 + \bar{u} ,
\]

\[
w_p = -V_F + \bar{w} ,
\]

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, \( \Phi(x, z) \), defined by the partial differential equations

\[
w_p = -V_F + \bar{w} = -\frac{\partial \Phi}{\partial x} (x, z) ,
\]

(A.103)
Figure A.2. Fall Velocity vs Particle Size

and

$$u_p = u_o + \frac{\partial \Phi}{\partial z}(x,z).$$  \hspace{1cm} (A.104)

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

$$\frac{dx}{u_p} = \frac{dz}{w_p} \quad \text{or} \quad -w_p dx + u_p dz = 0, \hspace{1cm} (A.105)$$

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

$$\frac{\partial \Phi}{\partial x} dx + \frac{\partial \Phi}{\partial z} dz = 0 \quad \text{or} \quad d\Phi = 0. \hspace{1cm} (A.106)$$
We thus see that the curves for which

$$\Phi = \text{constant}$$  \hspace{1cm} \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

$$\bar{u} = \frac{\partial \Psi}{\partial z},$$ \hspace{1cm} \text{(A.108)}

and

$$\bar{w} = -\frac{\partial \Psi}{\partial x},$$ \hspace{1cm} \text{(A.109)}

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

$$\Psi = \frac{-u_o (ah)(z + a)}{(z + a)^2 + x^2}.$$ \hspace{1cm} \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

$$\Phi = u_o z + \Psi + V_F x.$$ \hspace{1cm} \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

$$\text{constant} = C = \bar{z} + r \bar{x} - \alpha \frac{(1 + \bar{z})}{(1 + \bar{z})^2 + \bar{x}^2},$$ \hspace{1cm} \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
\[
\frac{\tilde{z}}{a} = \frac{h}{a} \left( 1 + \frac{1}{1 + (x/a)^2} \right) = \frac{\alpha}{1 + (x/a)^2}.
\] (A.113)

which is the equation of the straight-line descent of a fallout particle with slope 
\(-\frac{V_F}{u_0}\). 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

$$C^{(1)}(k) = \frac{u_0(ah)}{2} (ik) e^{-|k|a}.$$  

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, $\Delta$, between the height of the mountain ridge and the maximum elevation increases with a corresponding increase in $(h/a)$. We define $\Delta$ 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 $\Delta$ 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 $\alpha$ (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 $\alpha$. (Mathematically, this discrepancy is the difference between the $45^\circ$ ideal
Figure A.6. Differences Between Surface Trajectory and Mountain Ridge
lift line and the actual curve of \( \bar{Z}_{om} \) vs \( \alpha \). 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 \( \alpha \) 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 \( \alpha \). Since it is beyond the scope of this report to perform a parametric analysis of the stream function, we present the results for a fall-to-wind velocity ratio, \( V_F/u_o \), equal to 0.1; the curves differ only in the choice of \( \alpha \),

![Figure A.7. Fallout Particle Trajectories (Note vertical scale is amplified 10 times; dimensions are in units of \( \alpha \), the half width of the mountain range.)](image-url)
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 \( \gamma \) and \( \alpha \) and have applied the code for the cases \( \gamma = 0.1; \alpha = 0.25, 0.50, \) and \( 0.75 \) (between the limits \(-5 \leq x \leq 5\)). The results show no additional time delay due to the effect of the mountain ridge. On the contrary, we found a relative speedup of between 1 to 2%. Apparently, the increase in path length is slightly more than counterbalanced by the increase in velocity.

The fact that the perturbed trajectories which fail to intercept the mountain ridge always end up on the same unperturbed streamline is a general result, as can be seen by examination of the stream function,

\[
C = \bar{z} + r\bar{x} - \left[ \frac{\alpha(1 + \bar{z})}{(1 + \bar{z})^2 + \bar{x}^2} \right].
\]

The function in brackets is the contribution to the stream function attributable to the mountain ridge; for either large negative or positive values of \( \bar{x} \) this goes to zero.

The middle set of curves in Figure A.9 can be used to obtain some estimate of the enhanced fallout range caused by the mountain ridge for \( (V_F/u_o) = 0.1 \) with \( (\bar{Z}_{om}/a) = 0.5 \). From Figure A.2 we see that this would correspond to a 100-\( \mu \) particle with \( u_o = 30 \) mph, 150-\( \mu \) particle with \( u_o = 50 \) mph, or a 230-\( \mu \) particle with \( u_o = 70 \) mph, all of which are quite possible situations. If the actual mountain ridge peak were 1 mi, the intercept of the alluded-to trajectory would hit the horizontal axis at \( \bar{x} = 5 \) or \( x = 10 \) mi.
For heavy fallout particles, terrain effects are less important since the vertical lift decreases. At the other extreme, extremely light fallout particles will follow the wind streamlines.

**Conditions for Neglecting the Coriolis Effect**

We shall now examine the coefficients in the dispersion relationship — namely $a$, $b$, and $c$ (see Eqs. (A.35) and (A.36)) — to determine the conditions on the velocity and wavelength for which the Coriolis effect can be neglected. Since we are primarily interested in short range effects, we shall freely make use of the assumed inequality

\[
\left( k_x^2 + k_y^2 \right)^{1/2} > 0.75 \times 10^{-6} \text{ cm}^{-1}, \tag{A.115}
\]

which signifies that the wavelength of the horizontal variations in the terrain is less than the $2\pi\beta^{-1}$, or approximately 50 mi.

The Coriolis effect can be neglected in Eq. (A.36a) if

\[
|k_x|u_0 > \Omega, \tag{A.116}
\]

or equivalently

\[
(2\pi) \frac{|u_0|}{\Omega} > L_x, \tag{A.117}
\]

where $L_x$ is a representative wavelength in the $x$ direction. The distance $(u_0/\Omega)$ equals

\[
\frac{u_0}{\Omega} = d = \left( 3.8 u_m \right) \text{ mi}, \tag{A.118}
\]

where $u_m$ is the wind velocity expressed in miles per hour. This restricts $L_x$ to less than $24 u_m$ mi. We have assumed the inequality of Eq. (A.117) to hold in our analysis. Equation (A.117) will not be satisfied when the unperturbed velocity is
too large or when both a small $u_0$ 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 $u_0$ 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 $\Omega$ 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 $\Omega$ and $\Omega^2$.) The ratio of the first-to-second terms in Eq. (A.36b) is

$$\frac{\gamma \beta k_x^2 u_0^2}{2\Omega |k_x||k_y| u_0 \omega x} = \frac{\gamma \beta L_y u_0}{4\Omega L_x \cos \theta \sin \epsilon}, \quad (A.119)$$

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

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

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)

$$\left( \frac{k_x^2 u_0^2}{\Omega^2} \right) \frac{\beta}{|k_x|} = r_1. \quad (A.121)$$
Using Eq. (A.118) we find that the requirement that the foregoing expression is much greater than unity is given by

\[ 2\pi (1.8) u_m^2 \gg L_x \text{ (miles)} \]  

(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)

\[ \frac{k_x^2 u_o^2}{\Omega^2} \left( \frac{\beta}{|k_y|} \right) = r_1 \left( \frac{L_y}{L_x} \right) = \left[ 4\pi (1.8) u_m \right] \left( 0.5 \frac{L_y}{L_x} \right) u_m \]  

(A.123)

Since \( \left( 0.5 \frac{L_y u_m}{L_x} \right) \) is assumed greater than unity, the foregoing expression will also be greater than unity.

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

\[ c \approx k_x^2 u_o^2 \frac{\left( k_x^2 + k_y^2 \right)}{\Omega^2} + \Omega\sigma u_o k_x^2 + \Omega^2 \sigma k_y^2 \]  

(A.124)

By using the previously mentioned inequalities it is easy to show that \( \Omega \) and \( \Omega^2 \) 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


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\textsuperscript{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, 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 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 \( \Omega = \frac{2\pi}{(\text{sidereal day})} = 7.3 \times 10^{-5} \text{ sec} \). 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, 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 and Haurwitz, 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\textsuperscript{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.\textsuperscript{B.8} As pointed out by Fisher, the most important feature of the numerical method lies in the fact that the nonlinear advective terms in the equations may be retained and thus allow the feedback effect of the wind field itself on the sea breeze to be studied. Fisher's model is conceptually identical to the linear model of Haurwitz and may be considered the most definitive work in the field inasmuch as it includes not only nonlinear horizontal advection but also the spatial variation of the transport parameters. This solution shows the sea breeze in the stages of development and decay and succeeds in reproducing the gross features of the wind system and many of its small details as well.

The main drawback in applying Fisher's model to the fallout problem is its sheer complexity, particularly in view of the fact that, as pointed out by Fisher himself, its principal contribution is its ability to describe the fine structure in the sea-breeze development. Although we can justify the incorporation of the sea breeze in fallout models, we are hard-pressed to justify the inclusion of its subleties. Other effects such as the irregularity of the coastline, the presence of a prevailing wind, and uncertainties in the transport coefficients would completely overshadow any improvement attributed to incorporation of the sea-breeze fine structure. (Recently, an attempt was made by Travelers Insurance Research Laboratory\textsuperscript{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. 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. The dynamics of Defant's model are governed by the continuity equation, the three equations of motion, the equation of state, and the heat-diffusion equation. By neglecting variations in density except in so far as they modify the action of gravity, it becomes possible to construct a stream function which is used to describe the motion in the plane perpendicular to the coast. The mathematical equations are based on the assumption of an infinitely long coastline which we designate as the y axis. Variations of the meteorological equations in this direction are ignored. The x axis is perpendicular to the coast, and positive inland, while the z axis denotes the vertical. The equations which describe the system are:

\[
\frac{\partial u}{\partial x} + \frac{\partial w}{\partial z} = 0 \quad , \tag{B. 1}
\]

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + w \frac{\partial u}{\partial z} - fv = -\frac{1}{\rho} \frac{\partial p}{\partial x} - \sigma u \quad , \tag{B. 2}
\]

\[
\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + w \frac{\partial v}{\partial z} + fu = -\sigma v \quad , \tag{B. 3}
\]

\[
\frac{\partial w}{\partial t} + u \frac{\partial w}{\partial x} + w \frac{\partial w}{\partial z} = -\frac{1}{\rho} \frac{\partial p}{\partial z} - g - \sigma w \quad . \tag{B. 4}
\]

\[
p = \rho RT \quad , \tag{B. 5}
\]

and

\[
\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + w \frac{\partial T}{\partial z} = K \frac{\partial^2 T}{\partial z^2} \quad . \tag{B. 6}
\]
where \( u, v, \) and \( w \) are the velocity components along the \( x, y, \) and \( z \) axes, respectively; \( p \) denotes the pressure; \( T \) is the temperature; \( \rho \) 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 into account through the Guldberg-Mohn friction parameter, \( \sigma \). 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 \( (dp/dt) = 0 \), 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:

\[
\begin{align*}
w(z = 0) &= 0, \\
w(z \to \infty) &= 0, \\
T(z = 0) &= T_0 + T(x, t).
\end{align*}
\] (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, \( \sigma \), 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 $K_0^2 \frac{\partial u}{\partial z}^2$ and $K_0^2 \frac{\partial v}{\partial z}^2$, which are then used with the boundary conditions $u(z = 0) = v(z = 0) = 0$, we arrive at a sea-breeze model in which a boundary layer (in the sense of Schlicting and Prandtl) is built into the theory. In such a situation, the horizontal components of velocity increase with altitude from a minimum value of zero at the land and water surface. According to Haurwitz's model, the distance over which this buildup occurs is of the order of the characteristic height of the sea breeze. This seems to be somewhat inconsistent with the everyday experiences at the ocean front where strong horizontal winds are evident a few feet from the ground. Strictly speaking, when boundary layer theory is used, the temperature of the moving fluid at the boundary is the same as the surface temperature. Thus, if the theory of the boundary layer were rigorously applied on a clear sunny day in the summertime, we would necessarily have to use a land temperature of about $90^\circ - 100^\circ$ F and a water temperature between $60^\circ - 70^\circ$ F. This corresponds to a temperature differential of about $20^\circ$ 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 $\psi$ and the vorticity $\eta$, which are related to the x and z component of the velocity by

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

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

\[
\frac{\partial \eta}{\partial t} + \left( u \frac{\partial}{\partial x} + w \frac{\partial}{\partial z} \right) \eta - f \frac{\partial v}{\partial z} = -\frac{1}{\rho} \frac{\partial \rho}{\partial z} \left( \frac{\partial T}{\partial x} \frac{\partial \rho}{\partial x} - \frac{\partial \rho}{\partial x} \frac{\partial \rho}{\partial z} \right) - \sigma \eta .
\]  

(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 \(p = \rho RT\).

\[
S = -\frac{1}{\rho} \frac{\partial \rho}{\partial x} \left( \frac{\partial T}{\partial x} \frac{\partial \rho}{\partial x} - \frac{\partial \rho}{\partial x} \frac{\partial \rho}{\partial z} \right) = \frac{R}{\rho} \left( \frac{\partial T}{\partial x} \frac{\partial \rho}{\partial x} - \frac{\partial T}{\partial z} \frac{\partial \rho}{\partial z} \right) .
\]  

(B.11)

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

\[
S = \frac{R}{\rho} \left( \frac{\partial T}{\partial x} \right) = -\frac{\rho}{T} \left( \frac{\partial T}{\partial x} \right) .
\]  

(B.12)

As is usual in dynamic meteorology, we now replace \((1/T) (\partial T/\partial x)\) in Eq. (B.12) by \((1/\bar{\theta}) (\partial \bar{\theta}/\partial x)\), where \(\bar{\theta}\) is the so-called potential temperature. In Eq. (B.6) we replace \(T\) by \(\bar{\theta}\); thus,

\[
\frac{\partial \bar{\theta}}{\partial t} + u \frac{\partial \bar{\theta}}{\partial x} + w \frac{\partial \bar{\theta}}{\partial z} = \frac{\partial^2 \bar{\theta}}{\partial z^2} .
\]  

(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
\[ \tilde{\theta} = \theta_0 (z) + \theta(x, z, t) \quad , \] 

(B.14)

where \( \theta_0 \) and \( \theta \) are the perturbed and unperturbed parts, respectively. We then arrive at a set of linearized equations:

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

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

and

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

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 \( D_u = u(\partial \psi/\partial x) + w(\partial \psi/\partial z) \) with \( \sigma \), the Guldberg-Mohn parameter in Eq. (B.2). Roughly speaking, \( D_u \) can be assigned a value approximately equal to:

\[ D_u \approx \frac{\bar{u}}{L_x} \approx \frac{\bar{w}}{L_z} , \quad (B.18)\]

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. 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>
<thead>
<tr>
<th>Range (km)</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>16-32</td>
<td>New England</td>
</tr>
<tr>
<td>15</td>
<td>Flemish Coast</td>
</tr>
<tr>
<td>20-30</td>
<td>Baltic Sea</td>
</tr>
<tr>
<td>30-40</td>
<td>Holland</td>
</tr>
<tr>
<td>40-50</td>
<td>Sweden</td>
</tr>
<tr>
<td>up to 50</td>
<td>Jutland</td>
</tr>
<tr>
<td>40</td>
<td>Albania</td>
</tr>
<tr>
<td>&gt;50</td>
<td>Northern Coast of Java</td>
</tr>
</tbody>
</table>

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 $\bar{w} = 2 \text{ cm sec}^{-1} \text{°C}^{-1}$. 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 $5 \text{°C}$ is assumed, we obtain the following value of $D_u$

$$D_u \approx \frac{10}{20 \times 10^3} + \frac{0.1}{500} \sim 7 \times 10^{-4} \text{ sec}^{-1} \quad (B.19)$$
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 \( \theta_0 \) can be considered constant in Eq. (B.15), and its derivative at equilibrium, \( \Gamma \), 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

\[
\begin{align*}
\theta &= A(z, t) \sin \lambda x \quad \text{(B.20)} \\
\psi &= B(z, t) \cos \lambda x \\
v &= C(z, t) \cos \lambda x
\end{align*}
\]

Since the surface temperature differential can in general be represented by a Fourier series in multiples of the sidereal day frequency \( \Omega \), it follows from the principle of linear superposition that \( A, B, \) and \( C \) will be given by

\[
\begin{align*}
A(z, t) &= \sum_{n=1}^{\infty} A_n(z) e^{i\Omega t} \quad \text{(B.23)} \\
B(z, t) &= \sum_{n=1}^{\infty} B_n(z) e^{i\Omega t} \quad \text{(B.24)}
\end{align*}
\]

and

\[
\sum_{n=1}^{\infty} C_n(z) e^{i\Omega t} \quad \text{(B.25)}
\]
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 \( C_n \):

\[
(\sigma + \imath \Omega) \left( B_n'' - \lambda^2 B_n \right) + f C_n' = \alpha \lambda A_n \ , \tag{B.26}
\]

and

\[
(\sigma + \imath \Omega) C_n = f B_n' \ , \tag{B.27}
\]

\[
\imath \Omega A_n - \lambda \Gamma B_n = K A_n'' \ , \tag{B.28}
\]

where

\[
\alpha = g/\theta_0 \ .
\]

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

\[
W_n = -\lambda B_n \ , \tag{B.29}
\]

in terms of which the velocity components are given by

\[
w(x, z, t) = \sin \lambda x \sum_{n=1}^{\infty} W_n(z) e^{\imath \Omega t} \ , \tag{B.30}
\]

\[
u(x, z, t) = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} W_n'(z) e^{\imath \Omega t} \ , \tag{B.31}
\]

and

\[
v(x, z, t) = \sum_{n=1}^{\infty} \left( f/q_n \right) U_n(x, z) e^{\imath \Omega t} \ , \tag{B.32}
\]

where

\[
q_n = \sigma + \imath \Omega \ . \tag{B.33}
\]
Eliminating Eq. (B.27) and using (B.29) gives

$$W'' = a_n W_n - b_n A_n ,$$  \hspace{1cm} (B.34)

and

$$A'' = c_n W_n + d_n A_n ,$$  \hspace{1cm} (B.35)

where

$$a_n = \frac{q_n^2 \lambda^2}{\left(q_n^2 + f^2\right)} , \quad c_n = \frac{\Gamma K}{\left(q_n^2 + f^2\right)} ,$$  \hspace{1cm} (B.36)

$$b_n = \frac{q_n \alpha \lambda^2}{\left(q_n^2 + f^2\right)} , \quad d_n = \frac{\text{in} \Omega}{K} .$$  \hspace{1cm}

If we now let

$$A_n(z) = \hat{A}_n e^{\alpha_n z} ,$$  \hspace{1cm} (B.37a)

and

$$W_n(z) = \hat{W}_n e^{\alpha_n z} ,$$  \hspace{1cm} (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:

$$\begin{pmatrix} b_n & \mu_n - a_n \\ \mu_n - d_n & -c_n \end{pmatrix} \begin{pmatrix} \hat{A}_n \\ \hat{W}_n \end{pmatrix} = 0 ,$$  \hspace{1cm} (B.38)

where

$$\mu_n = \frac{q_n^2}{\alpha_n} .$$  \hspace{1cm} (B.39)
The only nontrivial solutions to Eq. (B.38) are those for which the determinant vanishes. This gives the two "allowed" values for \( \mu_n \):

\[
\mu_{n1} = \frac{(a_n + d_n) + (a_n + d_n)^2 - 4(a_n d_n + c_n b_n)^{1/2}}{2} = E_{n1} e^{i\gamma_{n1}},
\]

\[
\mu_{n2} = \frac{(a_n + d_n) - (a_n + d_n)^2 - 4(a_n d_n + c_n b_n)^{1/2}}{2} = E_{n2} e^{i\gamma_{n2}}.
\] (B.40)

The roots of the dispersion relationship correspond to four values of \( \alpha_n \) which are given by

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

and

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

where

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

Of the four possible roots for \( \alpha_n \) 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 \( \alpha_n \) must be negative so as to insure exponential damping of the sea breeze. We define the two roots for \( \alpha_n \) by the equations

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

and

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

\[ \epsilon_{n1} = +1 \text{ if } \cos \left( \eta_{n1} \right) < 0, \]

\[ \epsilon_{n1} = -1 \text{ if } \cos \left( \eta_{n1} \right) > 0, \]

(B.43)

\[ \epsilon_{n2} = +1 \text{ if } \cos \left( \eta_{n2} \right) < 0, \]

\[ \epsilon_{n2} = -1 \text{ if } \cos \left( \eta_{n2} \right) > 0. \]

In terms of the \( \alpha_n \), the solution to the problem is given by

\[
\theta = \sin \lambda x \sum_{n=1}^{\infty} \left( \hat{A}_{n1} e^{\alpha_{n1} z} + \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{i\omega t}, \quad (B.44)
\]

\[
w = \sin x \sum_{n=1}^{\infty} \left( r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{i\omega t}, \quad (B.45)
\]

\[
u = \lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} \left( \alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{i\omega t}, \quad (B.46)
\]

\[
v = \lambda^{-1} \cos \lambda z \sum_{n=1}^{\infty} \left( \alpha_{n1} r_{n1} \hat{A}_{n1} e^{\alpha_{n1} z} + \alpha_{n2} r_{n2} \hat{A}_{n2} e^{\alpha_{n2} z} \right) e^{i\omega t}, \quad (B.47)
\]
where

\[ g_n = - \left( \frac{1}{q_n} \right), \quad \text{(B.48)} \]

and

\[ r_{n1} = - \frac{b_n}{(\mu_{n1} - a_n)}, \quad r_{n2} = - \frac{b_n}{(\mu_{n2} - a_n)}. \quad \text{(B.49)} \]

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

\[ \theta(x, z=0, t) = \sin \lambda x \cdot T(t), \]

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

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

(B.50)

where

$$
T_n = \Omega (2\pi)^{-1} \int_{0}^{2\pi} T(t) e^{-in\Omega t} dt = \Omega (2\pi)^{-1} \left[ \int_{0}^{2\pi} T(t) \cos(n\Omega t) dt + i \int_{0}^{2\pi} T(t) \sin(n\Omega t) dt \right]
$$

(B.51)

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

$T_n^*$ is the magnitude of $T_n$ and $\tau_n$ is its phase as computed from Eq. (B.51). On the other hand, from Eq. (B.44) we must have

$$
T_n = \hat{A}_{n1} + \hat{A}_{n2}
$$

(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,

$$
r_{n1} \hat{A}_{n1} + r_{n2} \hat{A}_{n2} = 0
$$

(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

\[ \theta = \sum_{n=1}^{\infty} \theta_n , \quad (B.54) \]

\[ w = \sum_{n=1}^{\infty} w_n , \quad (B.55) \]

\[ u = \sum_{n=1}^{\infty} u_n , \quad (B.56) \]

\[ v = \sum_{n=1}^{\infty} v_n , \quad (B.57) \]
where

\[ \theta_n = \sin \lambda x T_n^* \left[ S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) + S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) \right], \quad (B.58) \]

\[ w_n = \sin \lambda x T_n^* \left[ S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) + S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) \right], \quad (B.59) \]

\[ u_n = \lambda^{-1} \cos \lambda x T_n^* \left[ \epsilon_n S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) + \epsilon_n S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) \right], \quad (B.60) \]

\[ v_n = \lambda^{-1} \cos \lambda x T_n^* \left[ \epsilon_n S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) + \epsilon_n S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) \right] \]

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

\[ \psi = -\lambda^{-1} \cos \lambda x \sum_{n=1}^{\infty} T_n^* \left[ S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) + S_n e^{k_n z} \cos \left( n\Omega t + \ell_n z + e^{k_n z} \right) \right], \quad (B.62) \]
References


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

<table>
<thead>
<tr>
<th>Card Number</th>
<th>Content</th>
<th>Variable Names and Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Limiting coordinates of the area to be covered by this topographic data tape: lower X, upper X, lower Y, upper Y (m)</td>
<td>TXLL, TXLU, TYLL, TYLU (4E13.6)</td>
</tr>
<tr>
<td>2</td>
<td>Topography identification card</td>
<td>(TOPID(J), J=1, 12) (12A6)</td>
</tr>
<tr>
<td>3</td>
<td>Control integer to indicate which data checking program is to be used. 0 indicates DATERR, other values are unassigned</td>
<td>ISUBR (I2)</td>
</tr>
<tr>
<td>4</td>
<td>Print control integer. 0 causes all inputs to be printed. 1 suppresses printing.</td>
<td>IPRINT (I2)</td>
</tr>
<tr>
<td>5</td>
<td>Grid interval and limiting coordinates for the first block of topo data (m)</td>
<td>GRINT, BXLL, BXLU, BYLL, BYLU (5E13.6)</td>
</tr>
<tr>
<td>6</td>
<td>Number of grid squares in the X direction and in the Y direction, respectively, in the regular data array $S(i, j)$</td>
<td>I, JJ (2I12)</td>
</tr>
<tr>
<td>7</td>
<td>Regular grid data and address array of the current data block</td>
<td>$(S(i, j), I=1, II), J=1, JJ$ (5E13.6)</td>
</tr>
<tr>
<td>8</td>
<td>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.</td>
<td>SUBSID(K) to SUBSID(K+5) (5E13.6)</td>
</tr>
<tr>
<td>9</td>
<td>Same as card set 5 but for the second data block</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Same as card set 6 but for the second data block</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Same as card 7 but for the second block</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Same as card 8 but for the second data block</td>
<td></td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>Last Card</td>
<td>blank</td>
<td></td>
</tr>
</tbody>
</table>

286
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 DATE"R 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 when it is entered after all topographic data have been processed.

Upon return from DATERR or any other data reading and checking program, TOPIN checks parameter GRINT for the termination condition (GRINT = 0 0). If termination is indicated, a transfer is made to statement number 11 (see the program listing) for final processing; if otherwise, parameter ITAPE is tested to see if a valid topo tape is still possible (ITAPE = 0) or if only a check of the remaining input deck can be made (ITAPE != 0). If ITAPE equals zero, a block count and the
arrays S(I, J) and SUBSID(K) are put temporarily onto tape ITEMP0 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 ≠ 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 ≠ 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
READ GRID INTERVAL AND LIMITING COORDINATES FOR CURRENT BLOCK OF TOPO DATA

IBLOCK = IBLOCK + 1

READ THE REGULAR 2D ARRAY OF TOPO HEIGHTS FOR THE CURRENT BLOCK OF TOPO DATA

KL = 1
KH = 6
KHH = 1

READ FIVE ELEMENTS OF THE ARRAY SUBSID, (SUBSID(K), K = KL, KH)

K = KL

SUBSID(K) + . . .

SUBSID(K) = 0

K = K + 1

K > KH

YES

KL = KL + 6
KH = KH + 6

NO

K = K - 1

END OF DATA FOR ARRAY SUBSID HAS BEEN REACHED

(a)

FC-C.2. Flow Charts of Subroutine DATERR

293
READ ALL NEGATIVE NO. S IN ARRAY [S, J] INTO ARRAY IRRAY AND COUNT THEM IN NN

READ ALL NEGATIVE NO. S IN ARRAY SUBSID(K) INTO ARRAY IRRAY AND CONTINUE COUNT INTO NN

ARE INPUTS TO BE PRINTED?

PRINT ALL INPUTS TO THIS PASS OF DATERR

(b)

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

FC-C.2. (Continued) Flow Charts of Subroutine DATERR
**Glossary for Topin and Datek**

<table>
<thead>
<tr>
<th>Topin</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BAIL</td>
<td>Lower a boundary of topographic data block currently in core.</td>
</tr>
<tr>
<td>BAILU</td>
<td>Upper a boundary of topographic data block currently in core.</td>
</tr>
<tr>
<td>BYLL</td>
<td>Lower y boundary of topographic data block currently in core.</td>
</tr>
<tr>
<td>BYLU</td>
<td>Upper y boundary of topographic data block currently in core.</td>
</tr>
<tr>
<td>GRINT</td>
<td>Length of the standard unit interval for 3(i,j).</td>
</tr>
<tr>
<td>IHTOPU</td>
<td>Final prepared topography tape number.</td>
</tr>
<tr>
<td>JJ</td>
<td>Upper limit of j dimension of 3(i,j) array.</td>
</tr>
<tr>
<td>JJUJ</td>
<td>Two-dimensional array containing either topographic data.</td>
</tr>
<tr>
<td>IPRINT</td>
<td>Indicates whether input (topo array) boundaries, limits etc for each block is to be printed.</td>
</tr>
<tr>
<td>ITAPE</td>
<td>Indicates that errors have been found in data set and that a valid topo tape cannot be written.</td>
</tr>
<tr>
<td>ISIN</td>
<td>System input tape number.</td>
</tr>
<tr>
<td>ISSN</td>
<td>System output tape number.</td>
</tr>
<tr>
<td>IUSER</td>
<td>Indicates if = 0 that subroutine DATEK is to be used.</td>
</tr>
<tr>
<td>ITEMPU</td>
<td>Temporary topo tape number.</td>
</tr>
<tr>
<td>JJUJ</td>
<td>Two-dimensional array containing either topographic data.</td>
</tr>
<tr>
<td>SUBJ(k)</td>
<td>One-dimensional array containing either a topographic data.</td>
</tr>
<tr>
<td>TALL</td>
<td>Lower a boundary of the complete topography area</td>
</tr>
<tr>
<td>TALLU</td>
<td>Upper a boundary of the complete topography area</td>
</tr>
<tr>
<td>TAYL</td>
<td>Lower y boundary of the complete topography area</td>
</tr>
<tr>
<td>TAYLU</td>
<td>Upper y boundary of the complete topography area</td>
</tr>
</tbody>
</table>

**Dimensional Arrays**

- **SUBJ(k)**

**Common Variables**

- **GRINT, BAIL, BAILU, BAYL, BYLU, JJ**
- **SUBJ, ITAPE, IPRINT, JJUJ**
- **TALL, TALLU, TAYL, TAYLU**

**Format**

- (12H1) IHTOPU
- TXLL = f6.18H
- TXLU = f6.18H
- TYYLL = f6.18H
GO TO 16
20 WRITE (IHTOPU) DENT
  WRITE (IHTOPU)ALL+ALOS+LL+TYLUS+NOT
  WRITE (IHTOPU)
  WRITE (IHTOPU)IHTOPU
  WRITE (IHTOPU)IHTOPU
  REWIND ITEMPU
18 READ (ITEMPU)IBLOCK
  I=ITOPU*1*IBLOCK
  J=ITOPU*2*IBLOCK
  K=ITOPU*3*IBLOCK
  READ (ITEMPU)(A)J
  READ (ITEMPU)A=1
  WRIT (IHTOPU)(A)J
  WRITE (IHTOPU)A=1
  IF (IBLOCK=NOT)10
19 END FILE IHTOPU
  WRITE (1501*8)
16 STOP
END
**SUBROUTINE DATER**

**SUBROUTINE TESTING FOR LOOPS AND INCOMPLETE RECORDS (IN X010)**

**COMMON BLOCK DATA**

**DIMENSION**

**FORMAT**

**DATA**
1

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

16 READ(I,J,K,L,U,V,W)

IF (MOD(I,K)) = 1

17 ISLACK = ISLACK + 1

READ (I,J,K,L,U,V,W)

IF (GRIND) = 1

18 A = A + 1

IF (A) = 2

19 K = K + 1

GO TO 20

20 CONTINUE

K = K + 1

GO TO 21

21 CONTINUE

K = K + 1

GO TO 22

22 A = A + 1

K = K + 1

GO TO 23

23 CONTINUE

I = I + 1

GO TO 24

24 CONTINUE

I = I + 1

IF (I) = 1

25 N = N + 1

IF (N) = 1

26 CONTINUE

N = N + 1

IF (N) = 1

27 M = M + 1

IF (M) = 1

28 M = M + 1

IF (M) = 1

29 M = M + 1

IF (M) = 1

303
COMPARISON OF TOTAL AREA OF TOPO REGION WITH THE SUM OF THE AREA UNIT 177
OF THE TOPO SUBDIVISIONS

42 K=1,
AREA=0.0
NO 1 NO 1=1*NBLCK
<IN#1=UPUL(1*1)
ALL=1.LPL(1*1)
NOJ=1.TPUL(1*1)
STO(1*1)=1*UPUL(1*1)
STO(1*1)=1*UPUL(1*1)
STO(41)=1*UPUL(2*1)+XJUP+GRINT
AREA=AREA+(STO(3*11)+STO(1*11)+(STO(4*1)+STO(2*1))
1.0 CONTINUE
THEA=1.TALL*+(TYL-TYLL)
IF(AREA-TREA)E=4+m*2+1
2.0 WRITE (150D1-226)
ITAPE=1
G0 TO 41
2.1 WRITE (150D1-221)

CHECK FOR THE ENTRAPMENT OF ONE TOPO SUBDIVISION BY
ANOTHER

2.3 K=1
217 NJ N0 1=1*NBLCK
IF(1.K1)=1+1+2+1
219 IF(STO(1*11)=STO(1*1112+*12+11
221 IF(STO(3*1)=STO(3*11)+12+12+12
221 IF(STO(4*1)=STO(4*11)+14+14+14
210 CONTINUE
IF(11-NBLCK)I=15+10+10+1
215 K1=K1+1
G0 TO 217
214 WRITE (150D1-221)
ITAPE=1
G0 TO 41

CHECK FOR GAPS BETWEEN TOPO SUBDIVISIONS

101 TS=0.0
EPSLN=1.0*5
XTEST=1.TALL+EPSLN
YTEST=TYL-EPSLN
105 DO 105 K=1*NBLCK
IF(XTEST-STO(1*11)+15+15+15
102 IF(YTEST=YTEST)+10+10+10+10+10
104 IF(YTEST=YTEST)+10+10+10
105 CONTINUE
G0 TO 202

DATE 177
DATE 178
DATE 179
DATE 180
DATE 181
DATE 182
DATE 183
DATE 184
DATE 185
DATE 186
DATE 187
DATE 188
DATE 189
DATE 190
DATE 191
DATE 192
DATE 193
DATE 194
DATE 195
DATE 196
DATE 197
DATE 198
DATE 199
DATE 200
DATE 201
DATE 202
DATE 203
DATE 204
DATE 205
DATE 206
DATE 207
DATE 208
DATE 209
DATE 210
DATE 211
DATE 212
DATE 213
DATE 214
DATE 215
DATE 216
DATE 217
DATE 218
DATE 219
DATE 220
DATE 221
DATE 222
DATE 223
DATE 224
DATE 225
DATE 226
DATE 227
DATE 228
DATE 229
DATE 230
DATE 231
DATE 232
DATE 233
DATE 234

305
106 IN=1
  K=K+1
  X[N(K)]=IN
  KMAX=K
  IF(JO(2*IN)-JO)100,100,107
107 IS=JOIN(2*IN)
108 XTEST=STOP(3*IN)+EPOLN
  IF(XTEST)-TALK1)109*20*110
110 YTEST=TS-EPOLN
  IF(YTEST)-TALK1)203*20*111
202 WRITE (1SOUT*222)XTEST,YTEST
  ITAPE=1
  GO TO 41
111 XTEST=TALK1+EPOLN
  TS=0.0
  GO TO 149

C
C   CHECK TO ENSURE THAT ALL VISO SUBLISTINGS HAVE BEEN LECTURED
C
C

203 XBLCK=NBLCK
204 DO 205 K=1,KMAX
  IF(X[N(K)]=XBLCK)GO TO 207,207
205 CONTINUE
206 WRITE (1SOUT*229)
  ITAPE=1
  GO TO 41
207 XBLCK=XBLCK+1
  IF(XBLCK)-L4*41,204
41 RETURN
ENU