MESOILT2, a Lagrangian Trajectory Climatological Dispersion Model

J. V. Ramsdell, Jr.
K. W. Burk

March 1991

Prepared for the Technical Steering Panel
DISCLAIMER

This report was prepared under the direction of the HANFORD ENVIRONMENTAL DOSE RECONSTRUCTION PROJECT Technical Steering Panel by Battelle Memorial Institute's Pacific Northwest Laboratories operating the Pacific Northwest Laboratory for the U.S. Department of Energy (DOE). While funding for the work was provided by DOE, the work is not under DOE direction or control. The views and opinions of the authors expressed in this document do not necessarily reflect those of the United States Government or any agency thereof. Reference herein to any specific commercial product, process or service by trade name, trademark, manufacturer or otherwise does not necessarily constitute or imply its endorsement, recommendation or favoring by the U.S. Government or any agency thereof, nor by Battelle Memorial Institute.

Printed in the United States of America

Available to DOE and DOE contractors from the
Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831;
prices available from (615) 576-8401. FTS 626-8401.

Available to the public from the National Technical Information Service,
MESOILT2, A LAGRANGIAN TRAJECTORY CLIMATOLOGICAL DISPERSION MODEL

Hanford Environmental Dose Reconstruction Project

J. V. Ramsdell, Jr.
K. W. Burk

March 1991

Prepared for
the Technical Steering Panel

Pacific Northwest Laboratory
Richland, Washington 99352
MESOILT2, A LAGRANGIAN TRAJECTORY CLIMATOLOGICAL DISPERSION MODEL

March 1991

This document has been reviewed and approved by the Technical Steering Panel.

John E. Till, Chairman
Technical Steering Panel
Hanford Environmental Dose Reconstruction Project

March 15, 1991
FOREWORD

Appendix D is a record of TSP comments and Battelle's responses to those comments; the TSP has reviewed and approved Battelle's responses. The comment numbers appear in this document in the left margin next to the paragraphs in which the corresponding comments are addressed. Any text that has been changed is shown in italics. In addition to changes to address TSP comments, some text has been changed for correction of errors and for further clarification.
The objective of the Hanford Environmental Dose Reconstruction (HEDR) Project is to estimate the radiation dose that individuals could have received as a result of emissions from nuclear operations at the Hanford Site since their inception in 1944. An independent Technical Steering Panel (TSP) directs the project, which is conducted by the Pacific Northwest Laboratory (PNL). The TSP directed PNL to demonstrate that its recommended approach for dose reconstruction is technically feasible and practical. This demonstration was Phase I of the project.

The dose reconstruction approach recommended by PNL involves:

- estimating the amount of radioactive material released and the uncertainty in these estimates
- modeling the transport, diffusion, and deposition of the material entering the atmospheric and water environmental pathways and the uncertainty and variability in the model predictions
- modeling the transfer of the material through the food chain and the uncertainty and variability of this transfer
- compiling available environmental monitoring data, determining the variability of the measurements and estimating their uncertainty
- determining the life styles of groups of individuals potentially exposed to the material
- computing ranges of dose estimates for individuals, considering the predicted and measured radionuclide concentrations in the environment and the diversity of the habits of individuals.

This report is specifically concerned with the approach that PNL recommends for dealing with the atmospheric pathway.

The TSP established a model domain for the atmospheric pathway for Phase I that includes 10 counties in Washington and Oregon and covers several thousand square miles. It is unrealistic to assume that atmospheric models which estimate transport and diffusion based on the meteorological conditions near the point of release of material at the time of release are adequate for a region this large. As a result, PNL recommended use of a Lagrangian trajectory, puff dispersion model for the Phase I study.
The dispersion model is implemented in the MESOILT2 computer code. MESOILT2 was prepared from existing codes specifically for the Phase I study. It will also serve as a base for further model development. Transport and diffusion calculations, which are based on hourly meteorological data from as many as 40 locations, account for spatial and temporal variations in the atmospheric conditions between the release point and receptors in the environment. The code produces estimates of time-integrated air concentrations, average deposition rates, and surface contamination on a monthly basis for radionuclides at nodes of a Cartesian grid. A post-processor program, PHASEII, uses the MESOILT2 results to determine average values and a measure of the variability/uncertainty in the averages for use in later stages of the dose reconstruction process.

This report describes the MESOILT2 computer code and the atmospheric transport, diffusion, deposition, and depletion models used in Phase I. The contents of the report include a technical description of the models, a user's guide for the codes, and descriptions of the individual code elements. Complete listings of the codes are contained in Appendixes A and B, respectively. Appendix C contains modifications to MESOILT2 made since the Phase I calculations were completed. These modifications do not affect the Phase I results.
## CONTENTS

**FOREWORD** ................................................................. iii
**SUMMARY** ................................................................. v

### 1.0 INTRODUCTION ......................................................... 1.1

- **1.1 MODELING REQUIREMENTS** .................................... 1.2
- **1.2 MODELING APPROACH** ......................................... 1.3
- **1.3 QUALITY ASSURANCE** .......................................... 1.5
- **1.4 REPORT ORGANIZATION** ....................................... 1.6

### 2.0 TECHNICAL DESCRIPTION .............................................. 2.1

- **2.1 REPRESENTATION OF THE ENVIRONMENT** ...................... 2.2
  - **2.1.1 Topography** .................................................. 2.2
  - **2.1.2 Atmosphere** .................................................. 2.3
- **2.2 CALCULATION OF PUFF TRANSPORT** ........................... 2.4
  - **2.2.1 Wind Field Definition** .................................. 2.5
  - **2.2.2 Puff Movement** .............................................. 2.10
- **2.3 CALCULATIONS OF DIFFUSION AND DEPOSITION AND TREATMENT OF DEPLETION** ........................................ 2.11
  - **2.3.1 Concentration Estimates** .................................. 2.12
  - **2.3.2 Diffusion** .................................................... 2.17
  - **2.3.3 Deposition** .................................................. 2.20
  - **2.3.4 Depletion** .................................................... 2.24
- **2.4 CENSUS DIVISION ESTIMATES** ................................ 2.26
  - **2.4.1 Predicted Values** ......................................... 2.26
  - **2.4.2 Uncertainty** ................................................ 2.29
- **2.5 MODEL EVALUATION** ............................................. 2.32

### 3.0 MESOILT2 USER'S GUIDE .............................................. 3.1
3.1 PROGRAM EXECUTION

3.2 INPUT DATA FILES

3.2.1 Meteorological Data File
3.2.2 Meteorological Stations File
3.2.3 Wind Speed Units File
3.2.4 Topographical Data File
3.2.5 Wind Adjustment Data File
3.2.6 Source-Term Data File
3.2.7 Residual Puff Data File

3.3 OUTPUT FILES

3.3.1 MESOILT2 Log File
3.3.2 Computational Results Files
3.3.3 Residual Puff Data File

3.4 PHASEII POST-PROCESSOR

3.4.1 Input Files
3.4.2 Output Files

4.0 THE MESOILT2 COMPUTER CODE

4.1 THE MAIN PROGRAM

4.1.1 Model Initialization
4.1.2 Dispersion Calculations
4.1.3 Model Output

4.2 INITIALIZATION PROGRAM ELEMENTS

4.2.1 Subroutine GRIDINLT
4.2.2 Subroutine REARNG
4.2.3 Subroutine STRAY
4.2.4 Subroutine ASCND
4.2.5 Subroutine TOPFIL .................................. 4.8
4.2.6 Subroutine INITLT .................................. 4.8
4.2.7 Subroutine RELEASLT ................................ 4.9

4.3 WIND PROCESSING PROGRAM ELEMENTS ................. 4.9
4.3.1 Subroutine WINDLT .................................. 4.10
4.3.2 Subroutine TERRALT ................................ 4.10
4.3.3 Subroutine DIRSPD .................................. 4.11

4.4 TRANSPORT AND DIFFUSION PROGRAM ELEMENTS ....... 4.11
4.4.1 Subroutine PUFFR .................................. 4.11
4.4.2 Subroutine PUFFMLT ................................ 4.12
4.4.3 Subroutine DTOPO .................................. 4.13
4.4.4 Subroutine DIFDEPLT ................................ 4.13
4.4.5 Subroutine SIGMA .................................. 4.15

4.5 INPUT AND OUTPUT PROGRAM ELEMENTS .................. 4.16
4.5.1 Subroutine METOPN .................................. 4.16
4.5.2 Subroutine DATRDLT ................................ 4.16
4.5.3 Subroutine DATWR .................................. 4.17
4.5.4 Subroutine READQ .................................. 4.17
4.5.5 Subroutine TESTM .................................. 4.17
4.5.6 Subroutine PRINTELT ................................ 4.18
4.5.7 Subroutine PUFFOUT ................................ 4.19

4.6 TIME-KEEPING PROGRAM ELEMENTS ....................... 4.19
4.6.1 Subroutines DATE, TIME, and SECOND ................. 4.19
4.6.2 Subroutines DTHOUR and HOURDT ..................... 4.19
4.6.3 Subroutine JULIAN .................................. 4.20

4.7 HOUSEKEEPING PROGRAM ELEMENTS ....................... 4.20
4.7.1 Subroutine CLEANLT .......................... 4.20
4.7.2 Subroutine SHIFT ............................. 4.21

5.0 THE PHASE II COMPUTER CODE ................... 5.1

6.0 REFERENCES ...................................... 6.1

APPENDIX A - MESOILT2 CODE LISTING ................. A.1
APPENDIX B - PHASE II CODE LISTING .................. B.1
APPENDIX C - MESOILT2 CODE MODIFICATIONS .......... C.1
APPENDIX D - SUMMARY OF TSP COMMENTS AND BATTELLE RESPONSES ........ D.1
## FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>HEDR Conceptual Model</td>
<td>1.1</td>
</tr>
<tr>
<td>1.2</td>
<td>Atmospheric Transport and Diffusion Modeling Approach</td>
<td>1.4</td>
</tr>
<tr>
<td>1.3</td>
<td>HEDR Phase I Data Summary Process</td>
<td>1.5</td>
</tr>
<tr>
<td>2.1</td>
<td>Geometry of Transport Component Adjustment to Account for Terrain Effects</td>
<td>2.9</td>
</tr>
<tr>
<td>2.2</td>
<td>Error Band for Numerical Process Used in Estimation of Time-Integrated Values</td>
<td>2.16</td>
</tr>
<tr>
<td>2.3</td>
<td>Horizontal Diffusion Coefficients</td>
<td>2.19</td>
</tr>
<tr>
<td>2.4</td>
<td>Vertical Diffusion Coefficients</td>
<td>2.19</td>
</tr>
<tr>
<td>2.5</td>
<td>HEDR Census Divisions</td>
<td>2.27</td>
</tr>
<tr>
<td>2.6</td>
<td>Phase I Atmospheric Modeling Domain</td>
<td>2.28</td>
</tr>
<tr>
<td>2.7</td>
<td>Comparison of Predictions of Monthly Average Krypton-85 Concentrations with Measured Values</td>
<td>2.31</td>
</tr>
<tr>
<td>2.8</td>
<td>Phase I Process for Estimating Iodine-131 Concentrations in Sage</td>
<td>2.33</td>
</tr>
<tr>
<td>2.9</td>
<td>Comparison of Preliminary Phase I Predictions of Iodine-131 Concentrations in Sage with Average Measured Values for Census Districts</td>
<td>2.34</td>
</tr>
<tr>
<td>2.10</td>
<td>Comparison of Preliminary Phase I Predictions of Iodine-131 Concentrations in Sage with Average Measured Values for Census Districts Using Concentrations Normalized to the Estimated Monthly-Total Release of Iodine-131</td>
<td>2.35</td>
</tr>
<tr>
<td>2.11</td>
<td>Temporal Variation of Measured and Predicted Concentrations of Iodine-131 in Sage in Four Census Divisions</td>
<td>2.36</td>
</tr>
<tr>
<td>3.1</td>
<td>An Example of an Output File Containing the Results of MESOILT2 Calculations</td>
<td>3.14</td>
</tr>
<tr>
<td>3.2</td>
<td>An Example of an Output File Data for Census Division</td>
<td>3.18</td>
</tr>
<tr>
<td>4.1</td>
<td>MESOILT2 Code Organization</td>
<td>4.2</td>
</tr>
</tbody>
</table>
### TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Parameter Values for Calculating Diffusion Coefficients Using the NRC Parameterization</td>
<td>2.18</td>
</tr>
<tr>
<td>2.2</td>
<td>MESOILT2 Washout Coefficients</td>
<td>2.23</td>
</tr>
<tr>
<td>3.1</td>
<td>MESOILT2 Command File for January 1984</td>
<td>3.2</td>
</tr>
<tr>
<td>3.2</td>
<td>MESOILT2 Input File Summary</td>
<td>3.6</td>
</tr>
<tr>
<td>3.3</td>
<td>Units and Ranges for Meteorological Variables</td>
<td>3.8</td>
</tr>
<tr>
<td>COMPUTER CODE LISTINGS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MESOILT2</td>
<td>A.1</td>
<td></td>
</tr>
<tr>
<td>ASCND</td>
<td>A.8</td>
<td></td>
</tr>
<tr>
<td>CLEANLT</td>
<td>A.10</td>
<td></td>
</tr>
<tr>
<td>DATRDLT</td>
<td>A.13</td>
<td></td>
</tr>
<tr>
<td>DATWR</td>
<td>A.15</td>
<td></td>
</tr>
<tr>
<td>DIFDEPLT</td>
<td>A.16</td>
<td></td>
</tr>
<tr>
<td>DIRSPD</td>
<td>A.23</td>
<td></td>
</tr>
<tr>
<td>DTHOUR</td>
<td>A.25</td>
<td></td>
</tr>
<tr>
<td>DTOPO</td>
<td>A.26</td>
<td></td>
</tr>
<tr>
<td>GRIDINLT</td>
<td>A.29</td>
<td></td>
</tr>
<tr>
<td>HOURDT</td>
<td>A.32</td>
<td></td>
</tr>
<tr>
<td>INITLT</td>
<td>A.34</td>
<td></td>
</tr>
<tr>
<td>JULIAN</td>
<td>A.38</td>
<td></td>
</tr>
<tr>
<td>METOPN</td>
<td>A.39</td>
<td></td>
</tr>
<tr>
<td>PRINTELT</td>
<td>A.41</td>
<td></td>
</tr>
<tr>
<td>PUFFIN</td>
<td>A.44</td>
<td></td>
</tr>
<tr>
<td>PUFFMLT</td>
<td>A.46</td>
<td></td>
</tr>
<tr>
<td>PUFFOUT</td>
<td>A.48</td>
<td></td>
</tr>
<tr>
<td>PUFFR</td>
<td>A.50</td>
<td></td>
</tr>
<tr>
<td>READQ</td>
<td>A.52</td>
<td></td>
</tr>
<tr>
<td>REARNG</td>
<td>A.53</td>
<td></td>
</tr>
<tr>
<td>RELEASLT</td>
<td>A.54</td>
<td></td>
</tr>
<tr>
<td>SHIFT</td>
<td>A.57</td>
<td></td>
</tr>
<tr>
<td>SIGMA</td>
<td>A.58</td>
<td></td>
</tr>
<tr>
<td>STRAY</td>
<td>A.60</td>
<td></td>
</tr>
<tr>
<td>Column 1</td>
<td>Column 2</td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>----------</td>
<td></td>
</tr>
<tr>
<td>TERRALT</td>
<td>A.62</td>
<td></td>
</tr>
<tr>
<td>TESTM</td>
<td>A.66</td>
<td></td>
</tr>
<tr>
<td>TOPFIL</td>
<td>A.68</td>
<td></td>
</tr>
<tr>
<td>WINDLT</td>
<td>A.70</td>
<td></td>
</tr>
<tr>
<td>CHAR. INC</td>
<td>A.74</td>
<td></td>
</tr>
<tr>
<td>CONST. INC</td>
<td>A.74</td>
<td></td>
</tr>
<tr>
<td>DATIM. INC</td>
<td>A.75</td>
<td></td>
</tr>
<tr>
<td>DECAY. INC</td>
<td>A.76</td>
<td></td>
</tr>
<tr>
<td>MATRIX. INC</td>
<td>A.76</td>
<td></td>
</tr>
<tr>
<td>PUFFS. INC</td>
<td>A.77</td>
<td></td>
</tr>
<tr>
<td>REL. INC</td>
<td>A.77</td>
<td></td>
</tr>
<tr>
<td>STATN. INC</td>
<td>A.78</td>
<td></td>
</tr>
<tr>
<td>TOPOGR. INC</td>
<td>A.78</td>
<td></td>
</tr>
<tr>
<td>UNITS. INC</td>
<td>A.79</td>
<td></td>
</tr>
<tr>
<td>WINDS. INC</td>
<td>A.79</td>
<td></td>
</tr>
<tr>
<td>PHASEII.FOR</td>
<td>B.1</td>
<td></td>
</tr>
</tbody>
</table>
1.0 INTRODUCTION

The Hanford Environmental Dose Reconstruction (HEDR) Project has been established to estimate the radiation doses that people may have received from emissions to the environment from operations that began at the Hanford Site in 1944. The project is being conducted by the Pacific Northwest Laboratory (PNL) under the direction of an independent Technical Steering Panel (TSP). Possible technical approaches to the project were discussed by both organizations at a series of public TSP meetings during 1988. During these discussions, PNL recommended a comprehensive approach that involves simultaneous estimation of the radiation doses and the uncertainties in the dose estimates. This approach is shown in Figure 1.1 and is described in detail by Napier (1990).

The conceptual approach begins with identification of the radionuclides released and estimation of the magnitude and timing of the releases to each of the environmental pathways. This information is collectively referred to as the source term. The atmosphere, surface water, and ground water provide the environmental pathways that could potentially result in radiation doses. This report deals specifically with the atmospheric pathway. Another task in the project deals with the other two pathways.

Once in the atmosphere, material carries downwind and deposits on the earth's surface. Radionuclides in the atmosphere result in exposure to humans.

FIGURE 1.1. HEDR Conceptual Model
and animals through direct radiation and inhalation. Radionuclides deposited on the earth’s surface result in exposure to humans and animals through direct radiation and through ingestion of contamination. Animals and plants exposed to airborne and deposited radionuclides provide an indirect, but important, pathway for exposure of humans.

In January 1989, the TSP directed PNL to demonstrate that the conceptual approach is both technically feasible and practical (PNL 1989). This demonstration is referred to as Phase I of the HEDR Project. The TSP directed that the atmospheric pathway approach be demonstrated for the period from December 1944 through 1947 because it was the period of the highest level of emissions of iodine-131 to the atmosphere. Subsequently, the Phase I domain for the atmospheric pathway was defined as Benton County, Washington, the seven adjacent counties in Washington, and the two adjacent counties in Oregon. The results of the Phase I study will be considered in establishing domains for later phases of the HEDR Project.

Doses to humans from operations at the Hanford Site cannot be estimated until the distribution of radionuclides in the environment is known. In theory, this distribution could be determined directly from environmental monitoring data. However, environmental monitoring data for the early period of operations at the Hanford Site are not adequate to estimate doses within the entire area of interest in the HEDR Project. The alternative approach is to estimate the environmental distribution of radionuclides by estimating the magnitude and timing of releases to the environment and using pathway models to distribute the radionuclides within the environment. Monitoring data, where available, can be used for dose estimates and to evaluate the environmental pathway models. Models for estimating transport, diffusion, and deposition of radionuclides released to the atmospheric pathway are described in this report.

1.1 MODELING REQUIREMENTS

After considering various approaches to modeling the atmospheric pathway, the TSP determined that temporal and spatial variations in atmospheric conditions must be considered in the atmospheric dispersion models used in the HEDR Project. This decision was implicit in the TSP’s approval of the PNL work.
plan for 1989 and use of the MESOI/MESORAD suite of dispersion models as a starting point for development of dispersion models specifically for the HEDR Project (PNL 1989). Rationale leading to these decisions is discussed in a separate report (Ramsdell 1989).

Based on the TSP directives and discussions with the TSP, the following modeling requirements were established:

- use existing models/computer programs/program components to the extent possible
- incorporate temporal and spatial variations in atmospheric conditions in models
- estimate the monthly average, time-integrated atmospheric concentrations of iodine-131 for census divisions in the 10-county, Phase I domain and the uncertainty in these averages
- estimate the monthly average iodine-131 deposition rates for census divisions in the 10-county, Phase I domain and the uncertainty in these averages
- estimate average month-end iodine-131 surface contamination for census divisions in the 10-county, Phase I domain and the uncertainty in these averages.

1.2 MODELING APPROACH

The modeling approach selected was to modify an existing computer code for the Phase I atmospheric pathway calculations. The modified code is an adaptation of Version 2.0 of the MESOI code (Ramsdell et al. 1983) designed to test the feasibility of using Lagrangian trajectory, puff models in climatological applications at Hanford. Mesoilt2 is the name given to this version of the code. Mesoilt2 computes the monthly time-integrated air concentration, monthly average deposition rate, and month-end surface contamination at nodes of a 31-by-31 Cartesian grid. A post-processing program called PhaseII computes averages of these values and estimates of their uncertainty/variability for census divisions from Mesoilt2 output. Mesoilt2 may undergo major revisions or changes after Phase I of the HEDR Project.

Meteorological data for 1944 through 1947 were ordered from the National Climatic Data Center following receipt of the TSP directive. However, by May 1989 it was clear that the data would not be received in time to permit
Phase I calculations to be completed on schedule. Therefore, a surrogate meteorological data set, 1983 through 1987, was selected rather than delay the Phase I calculations.

Use of the surrogate data set has two effects on the Phase I calculations. It provides for better definition of typical dispersion patterns in the vicinity of the Hanford Site because data are available for more meteorological stations. However, use of the 1983 to 1987 data increases the uncertainty in the atmospheric pathway calculations. This increase comes from the assumption that the 1983 to 1987 data are representative of the data for the earlier period and inability to match releases with the meteorological conditions during the releases.

Figures 1.2 and 1.3 illustrate the modeling approach used in the Phase I calculations. The remainder of this report discusses details of that approach.

Figure 1.2 shows the primary input and output of MESOILT2. Following directions supplied in the Command File, MESOILT2 reads data files containing meteorological data, source-term data, and topographic data. It performs atmospheric transport and diffusion calculations, and at the end of each month it creates output files that contain the time-integrated air concentration, the average deposition rate, and the month-end surface contamination for each node. MESOILT2 also creates a file that contains information about the puffs existing at the end of the month. Data in this file are carried forward to the following month.

**FIGURE 1.2.** Atmospheric Transport and Diffusion Modeling Approach
When MESOILT2 calculations are complete, there are 120 files containing time-integrated air concentrations, 60 files with average deposition rates and 60 files with month-end surface contamination. The PHASEll program takes the files for each calendar month (e.g., the five January time-integrated air concentrations) and computes an average value (geometric mean) and an uncertainty estimate for each census district. These values become input to later stages of the overall dose estimation process.

1.3 QUALITY ASSURANCE

The work described in this report was conducted in accordance with the requirements of ANSI/AMSE WQA-1 1986 Edition (ASME 1986) as interpreted by the PNL Quality Assurance (QA) Program. Development of the MESOILT2 and PHASEll computer codes was guided by the PNL QA program requirements. All procedures used to support the report were appropriately written and controlled in accordance with PNL QA program requirements. Records that support the data in this report were created and stored in accordance with applicable HEDR record control procedures. Following completion of development, MESOILT2 was placed under configuration control and subjected to computational tests outlined in the MESOILT2 Software Testing Plan. The result of the tests were examined by the programmer and an independent reviewer and were found to be correct.
1.4 REPORT ORGANIZATION

This report is divided into five major sections. These sections are the introduction, a technical description of the atmospheric models, a user's guide to the MESOILT2 code that implements the models, descriptions of the program elements in the MESOILT2 code, and a description of the PHASEII computer code. Completed listings of the computer codes are presented in Appendixes A and B. Appendix C lists modifications to MESOILT2 made since the Phase I calculations were completed. These modifications do not affect the Phase I results.
2.0 TECHNICAL DESCRIPTION

MESOILT2 is a Lagrangian trajectory, puff dispersion model in which puff movement is treated independently of diffusion within the puffs. It is an interim code derived from existing programs with limited modifications. Early workers in the field of atmospheric dispersion, (e.g., Sutton 1932; Frenkiel and Katz 1956; Gifford 1957; Gifford 1959) approached the study of diffusion in plumes by first considering puffs. Equations describing plumes were developed by integration of the equations derived for puffs. Start and Wendell (1974) developed a computer code to perform this integration numerically. Their code is the MESODIF model, which is the starting point for development of the MESOI/MESORAD suite of models at PNL.

MESOI Version 1.0 (Ramsdell and Athey 1981) was basically a rework of MESODIF. All releases and receptors were assumed to be at ground level, and the atmosphere was modeled as a single layer having winds that were not a function of height. The model did not include deposition or radioactive decay information. Following publication of MESOI Version 1.0, the U.S. Department of Energy and the U.S. Nuclear Regulatory Commission combined resources in the development of MESOI Version 2.0 (Ramsdell et al. 1983). Version 2.0 extended MESOI to include treatment of elevated releases and the treatment of deposition and decay. It also extended the description of atmosphere to include a three-layer wind model and permit adjustment of wind fields to account for anticipated effects of terrain.

A number of dispersion models have been developed from MESOI Version 2.0. These models include the MESORAD model developed for emergency response applications for the U.S. Department of Energy and U.S. Nuclear Regulatory Commission (Ramsdell et al. 1988; Scherpelz et al. 1986), and the MESOSEA model developed for the U.S. Environmental Protection Agency (Droppo et al. 1987) to evaluate potential environmental impacts of ocean incineration. During this same time period, Glantz developed a modified version of MESOI to explore the use of puff models in climatological dispersion problems. MESOILT2 is a modification of Glantz's code. Following completion of the modifications, the MESOILT2 code was verified for use in the Phase I demonstration of the conceptual model.
The following sections describe the representation of the environment in MESOILT2, the calculation of puff transport, the calculation of the diffusion within puffs, and the treatment of deposition and decay. It does not include detailed derivations. These derivations, in varying levels of detail, may be found in textbooks on atmospheric dispersion, for example Csanady (1973), Hanna et al. (1982), Pasquill and Smith (1983), Randerson (1984), and Seinfeld (1986).

2.1 REPRESENTATION OF THE ENVIRONMENT

The MESOILT2 model domain is a three-dimensional space. Horizontal positions are defined by a Cartesian coordinate system. Wind fields and puff positions are specified in a grid with 16 nodes on a side; and concentrations and deposition computations are made at ground level on a second grid with twice the resolution. In the vertical, positions are represented for the most part as heights above ground in meters. The exception to this rule is in calculation of concentrations at locations that are not directly beneath the center of a puff. In this case, the vertical positions of the puff center and the receptor are heights above a reference plane, typically mean sea level.

2.1.1 Topography

Topography may be used in MESOILT2 in two ways—it may be used in the diffusion calculations, and wind fields may be modified to account for the expected effects of topography. In both cases, the required topographic information must be provided by the model user in data files. If the topographic data files are not provided or are not read successfully by the program, the model will execute in a "flat earth" mode.

Wind fields are modified only at nodes of the wind field grid where terrain effects are expected. Typically, these nodes are at locations where the slope is steep and the wind is likely to be diverted. Data required for modification of wind fields are location, orientation of the slope, and a reduction factor for the upslope component of the wind vector. For HEDR Phase I model runs, the wind field modification data were computed from terrain elevations digitized at 30" latitude and longitude intervals (Hittelman et al. 1989). Data processing involved smoothing the topography using a low-pass
filter and computing slope gradients using the smoothed data. The slope orien-
tation is a horizontal vector perpendicular to the gradient with a direction
from north clockwise through south. The reduction factor for the upslope wind
component is a function of the magnitude of the slope gradient. Adjustment of
the wind field is discussed further when the wind field representation is
described.

Topographic data used in the diffusion calculations are entered for all
nodes of the concentration grid. Typically, they are elevations above mean
sea level, although any reference plane may be used. Terrain elevations for
the Phase I domain were extracted from the smoothed topographic data used in
preparation of the wind field modification file.

Data used in modifying the wind field and the topographic data are input
to Mesoilt2 using sequential, formatted data files. These files are described
in Sections 3.2.4 and 3.2.5 of this report.

2.1.2 Atmosphere

The atmospheric variables used in Mesoilt2 are wind speed and direction,
atmospheric stability class, and mixing layer thickness. Provision has been
made for entry of ambient air temperature and precipitation class. However,
neither of these variables were used in the Phase I calculations. All vari-
ables are permitted to vary as with time, based on input meteorological data.
The wind speed and direction vary in both time and space.

Mesoilt2 is set to accept meteorological data observed at hourly or
3-hour intervals. Two sets of meteorological data are available to the model
at all times. These sets are the current or most recent set of observations
and the set made at the next observation time. If data are entered at hourly
intervals, the set of current meteorological conditions is used for all calcu-
lations. The data are assumed to persist for the full hour in which the
observations were made. If data are entered at 3-hour intervals, calcula-
tions are made using the current observations for the first hour. Calcula-
tions made during the second and third hours are based on meteorological
conditions estimated by interpolation between the most recent set of observa-
tions and the next set observations. During the second hour, the weight given
to the most recent data set is two-thirds, and the weight given to the coming

2.3
MESOILT2 uses a three-layer wind field. The bottom (surface) layer, which extends from the earth's surface to 10 m, is defined from hourly observations of winds at up to 40 locations in and near the model domain. Procedures used to define the surface wind field are discussed later. Wind direction and speed are independent of height in this layer but do vary horizontally. The top layer starts at the top of the mixing layer and extends upward. In this layer the wind is assumed to be independent of height and uniform over the entire model domain. Wind direction and speed for this layer are entered as part of the meteorological data file. Winds in the layer between 10 m and the top of the mixing layer vary with both height and horizontal position. They are not entered via the meteorological data file. They are computed as needed by linear interpolation between the surface-layer and top-layer winds.

Atmospheric stability, the mixing-layer thickness, precipitation, and ambient air temperature are assumed to be constant throughout the MESOILT2 domain. They are entered via the meteorological data file. In general, it is anticipated that observations most nearly representative of the vicinity in which the releases take place will be used to determine values for these variables. For the Phase I calculations, data from the Hanford Meteorology Station were used to estimate atmospheric stability and mixing-layer thickness. Neither precipitation nor air temperature were used in the Phase I calculations. They will be used when calculations for 1944 through 1947 are repeated using meteorological data for the period rather than the surrogate meteorological data set.

2.2 Calculation of Puff Transport

The fundamental assumptions in all puff models are that plumes can be represented by a sequence of puffs and that puff movement may be separated from puff diffusion. Representation of plumes by a sequence of puffs was
discussed earlier as a step in derivation of plume models. We now consider the question of separation of transport and diffusion.

Winds are, in many respects, a random process that involves eddies that have a wide range of sizes. On the small end of the scale, eddies associated with dissipation of energy have dimensions of millimeters and less, and on the large-scale, motions associated with weather systems have dimensions of thousands of kilometers. Between these extremes are eddies of varying sizes. Turbulence energy spectra described by Panofsky and Dutton (1984) indicate that a local energy maximum is associated with eddies with periods on the order of a few (10 to 20) minutes. Spectra also indicate an energy minimum associated with eddies with periods on the order of an hour. Thus, there tends to be a natural division of eddy sizes in the atmosphere that roughly coincides with the observation frequency for meteorological data. Large eddies associated with day-to-day variations in meteorological conditions are treated in transport. Growth of puffs and plumes caused by small eddies are treated as diffusion.

Puff and plume dimensions also act to separate transport from diffusion. Eddies that are large compared to the crosswind or vertical dimensions of plumes tend to move the plume rather than expand it. Physically, the effects of eddies on plumes can be imagined by considering forces on the plume. The plume’s response to large eddies is basically response to uniform forcing across the plume. These forces result in displacement not distortion. On the other hand, if the dimensions of eddies and plumes are about the same, the forces on one side of the plume may be opposite to those on the other side. In this case, the effect is distortion, which results in plume growth. If eddies are small relative to plume dimensions, then the forces tend to cancel rather than cause displacement or distortion of the plume. As a result, eddies that are much smaller than plumes are not significant factors in either transport or diffusion.

2.2.1 Wind Field Definition

Wind direction and speed data enter MESOILT2 via the meteorological data file. This file may contain surface-level wind directions and speeds for as many as 40 locations defined in a meteorological stations file. These files
are described in Sections 3.2.1 and 3.2.2. Before use in the transport calculations, the meteorological station wind speeds are converted to meters per second, and the direction and speed are converted to Cartesian coordinates of the wind transport vector. The wind transport components for the stations are then used to estimate components at nodes throughout the model domain. Finally, the transport components are adjusted to account for the expected effect of terrain. This section describes these data manipulations.

Wind speeds are commonly measured and reported in three sets of units—miles per hour, knots (nautical miles per hour), and meters per second. A wind speed conversion routine has been incorporated in MESOILT2 to permit wind speeds for each station to be entered in the units in which they are reported. The wind speed units are specified for each station in a file read during program initialization. Conversion of the speeds in MESOILT2 rather than before the program minimizes the loss in precision of the data that would result from conversion outside of the program.

Wind directions are reported as the direction from which the wind blows. Thus, a north wind comes from the north. North winds transport material to the south not the north. MESOILT2 requires wind direction input to the nearest 10°. During the 1940s and 1950s, meteorological stations recorded directions by compass points: north, north-northeast, east, etc. MESOILT2 will be modified to convert wind directions from compass points to directions in degrees before using any of these data. Appendix C lists code changes for this modification.

MESOILT2 uses transport components in its calculations rather than wind speed and direction. The transport components are the components of the movement vector, which is the direction toward which the wind blows. Conversion of wind speed and direction to transport components involves rotation of the wind direction by 180° and projection of the resulting vector on north-south and east-west axes.

Following wind data input and wind speed conversion, the surface wind data may be considered as a pair of vectors $g(k)$, which contains the directions at each of the meteorological stations, and $S(k)$, which contains the speeds. These vectors are used to compute the two equivalent transport
component vectors $U(k)$ and $V(k)$, where $U(k)$ contains the east-west component of the transport vector and $V(k)$ contains the north-south component. By convention, transport to the east and north are considered to be positive. This transformation is accomplished by

$$U(k) = S(k) \cos(270^\circ - \theta(k))$$

and

$$V(k) = S(k) \sin(270^\circ - \theta(k)).$$

After the wind data at each station have been converted to transport components, an initial wind field is estimated by interpolation and extrapolation of the meteorological station transport components to nodes of a 16 by 16 Cartesian grid. If a meteorological station is not at a node, interpolation and extrapolation to the nodes is done using $1/r^2$ weighted averages of the transport components from the meteorological stations closest to each node. Goodin et al. (1979) considered various methods of interpolation and extrapolation of wind data and concluded that $1/r^2$ weighted averaging is a good compromise between computational costs and accuracy.

Using $1/r^2$ weighted averaging, the $U$ component at node $i,j$ is computed using

$$U(i,j) = \sum w(k) U(k) / \sum w(k)$$

where $w(k)$ is the weight and is defined by

$$w(k) = 1/r(k)^2$$

and

$$r(k)^2 = [i - x(k)]^2 + [j - y(k)]^2$$

2.7
with the position of the meteorological stations $x(k)$ and $y(k)$ given in grid coordinates. A similar equation with $V(i,j)$ and $V(k)$ replacing $U(i,j)$ and $U(k)$, respectively, is used to compute the north-south transport component.

The summations in this equation incorporate data from the five closest meteorological stations if there are five or more stations within five grid units of the node. If there are not five stations within this distance, the summation will proceed until it includes data from three stations or until data from all stations are included. The model will run with wind data from a single station.

If data have been provided to modify the transport winds for expected terrain effects, the modification is made following computation of the initial gridded wind field. Wind field modifications are made only at nodes for which the adjustment data are provided. The adjustment data consist of an angle clockwise from north to a line tangent to the gradient vector and two coefficients used in adjusting the upslope component of the transport vector. The first of these coefficients is used if the transport vector is to the right of the tangent line, and the second is used if the transport vector is to the left.

Figure 2.1 shows an example of how the adjustment is made at a node. The node at point 0 is on the southeast side of a large hill, and the solid arrow represents the initial transport vector, which is almost directly upslope. Assuming a slope orientation ($\phi_t$) of $047^\circ$ and a coefficient of 0.5 for transport to the left, the adjustment proceeds as follows. The normal MESOILT2 coordinate system is indicated by the x and y axes. A second coordinate system is established with axes tangent and normal to the slope. This system is represented by the t and n axes. The initial transport vector is projected on the t and n axes as shown by the dotted line. The n axis component is then multiplied by the coefficient 0.5, and the t axis component is increased to preserve the original transport speed. The result gives the t and n components of the modified transport vector shown by the dashed lines and the vector shown by the double-line arrow. In effect, the wind direction has been rotated clockwise $42^\circ$. The final step is to compute the x and y components of the modified transport vector for use in computing puff transport.
FIGURE 2.1. Geometry of Transport Component Adjustment to Account for Terrain Effects

After the wind field has been adjusted for anticipated terrain effects, MESOILT2 is ready to begin transport calculations. Puff movement is based on the transport vector at the position of the nominal center of the puff. This position is given by the puff's grid coordinates and the effective release height. In general it is not at a node or within the surface layer.

Transport vectors at the puff centers are computed as they are needed. If the effective release height is above the top of the mixing layer, transport vector components computed from the top-level wind are used to determine puff movement. The components are not a function of position. If the effective release height is 10 m or less, puff movement is based only on the surface-level winds. Otherwise, puff movement is based on the surface wind directly below the puff center and the top-layer wind. The surface-level transport vector at the puff center is calculated from the gridded transport component fields.

Bilinear interpolation is used to compute the components of the transport vector if the puff center is within the domain. Given a puff position
x, y with x between nodes i and i + 1 and y between j and j + 1, then the east-west component of the transport vector in the surface layer, \( U(x,y,10) \), is computed using

\[
U(x,y,10) = (1 - \alpha)(1 - \beta)U(i,j) + \alpha(1 - \beta)U(i + 1,j) + \beta(1 - \alpha)U(i,j + 1) + \alpha\beta U(i + 1,j + 1)
\]

where

\[
\alpha = \frac{(x - i)}{(i + 1) - i}
\]

\[
\beta = \frac{(y - j)}{(j + 1) - j}.
\]

A similar equation with \( V(\ldots) \) replacing the \( U(\ldots) \) is used to compute the north-south component. If the puff center is on the edge of the domain, linear interpolation is used to compute the transport vector components at the puff center. Finally, if the puff center is outside the domain, but the puff is still affecting the nodes, the transport vector components at the puff center are set equal to the components at the closest node.

When the components of the surface-layer transport vector beneath the puff center have been computed, the components at the height of the puff center are computed by linear interpolation. Given \( U(x,y,10) \) and the transport component \( U(L) \) at the top of the mixing layer, the transport component, \( U(x,y,z) \), at the puff center is

\[
U(x,y,z) = U(x,y,10) + \frac{(z - 10)}{(L - 10)} [U(L) - U(x,y,10)]
\]

where \( L \) is the mixing layer thickness. A similar expression gives the \( V \) component at the puff center.

2.2.2 Puff Movement

Puff movement is computed in a two-step process. An initial estimate of the movement is made using the transport vector at the puff's starting position. Transport vector components are estimated at the new position of the puff. The puff's original position and these components are used to obtain a second estimate of the puff movement. Finally, the two estimates are averaged, and the puff is moved.
For a puff initially at x,y,z this process may be mathematically represented by the following steps:

1. estimate initial movement

\[ \Delta x = U(x,y,z) \Delta t \]
\[ \Delta y = V(x,y,z) \Delta t \]

2. make a first estimate of the coordinates of the new horizontal position

\[ x' = x + \Delta x \]
\[ y' = y + \Delta y \]

3. make a second estimate of the puff movements

\[ \Delta x' = U(x',y',z) \Delta t \]
\[ \Delta y' = V(x',y',z) \Delta t \]

4. compute the coordinates of the new horizontal position

\[ x'' = x + (\Delta x + \Delta x')/2 \]
\[ y'' = y + (\Delta y + \Delta y')/2. \]

In MESOILT2, \( \Delta t \) is 15 minutes. This is the same interval at which puffs are released when a source is active.

Puffs are moved from position x,y to x'',y'' in a single step when twice the horizontal diffusion coefficient (\( \sigma_y \)) is equal to or larger than the distance to be moved. Until they reach this size, the puffs are moved in three equal steps to permit more accurate estimation of time-integrated concentrations and deposition rates. This point is discussed later under diffusion.

2.3 CALCULATIONS OF DIFFUSION AND DEPOSITION AND TREATMENT OF DEPLETION

Once material is released to the atmosphere, it disperses with the freedom of the winds. Large-scale motions move plumes about, and small-scale atmospheric motions distribute material within plumes. The preceding discussion of the transport described how MESOILT2 accounts for the effects of large-scale motions. This section describes how MESOILT2 accounts for the
effects of the small-scale motions. It also covers the deposition of material on surfaces and depletion to account for deposition and radioactive decay.

2.3.1 Concentration Estimates

Development of the puff diffusion model used in MESOILT2 starts with a differential equation called the diffusion equation. This equation and its solutions under very specific boundary and initial conditions are described in advanced mathematics texts. The conditions for which the equation has been solved are rarely found in the real atmosphere. However, a number of approximate solutions, including the Gaussian diffusion models, have been applied to the atmospheric diffusion problems with reasonable success. Of the eight preferred air quality models listed by the U.S. Environmental Protection Agency (EPA 1986), seven are Gaussian plume models. Similarly, of the 27 alternative air quality models listed, 23 are Gaussian models or include components that are Gaussian models. The derivation and application of atmospheric dispersion models is covered in specialized texts, for example Csanady (1973), Gifford (1968), Neiuwstadt and van Dop (1982), Pasquill and Smith (1983), and Seinfeld (1986).

The best known and most widely applied atmospheric dispersion models are the straight-line Gaussian plume models. When development of the MESOI/MESORAD suite dispersion model was begun, a Gaussian puff model was chosen over the straight-line model and other models because 1) it is easier to represent spatial and temporal variations in atmospheric conditions realistically in puff models than it is in straight-line models, and 2) Gaussian puff models retain much of the computational simplicity of straight-line Gaussian models.

One of the basic assumptions in puff models is that a continuous plume can be approximated by a finite number of puffs released in succession. The concentration at a receptor is assumed to be equal to the sum of the concentrations from all of the puffs, that is

\[
x(x,y,z,t) = \sum_{i=1}^{N} x_i(x,y,z,t)
\]
where \( x \) = concentration \\
\( x, y, z \) = position of the receptor in Cartesian coordinates \\
\( t \) = time of the concentration estimate \\
\( i \) = puff number \\
\( N \) = total number of puffs in the model domain.

In practice, computational rules based on puff dimensions have been established to limit the number of terms included in the summation.

In the absence of external influences such as the ground, the concentration distribution in each of the puffs in MESOILT2 is assumed to be Gaussian. Diffusion in the along-wind and cross-wind directions is assumed to be equal; that is, horizontal cross sections through puffs are circular. A corollary of this assumption is that concentrations in a horizontal plane decrease as a function of increasing distance from the puff center and are independent of the direction in which the distance is increased. Therefore, the \( x \) axis of the coordinate system can be assumed to point toward the east with the \( y \) axis pointing to the north and the vertical axis pointing upward.

With these assumptions, the concentration at \( x, y, z \) at time \( t \) due to puff \( i \) is given by

\[
x_i(x, y, z, t) = Q(t) F(y) G(z) / [(2\pi)^{3/2} \sigma_y \sigma_z]
\]

where 
\( Q(t) = \) mass of material (radionuclide) in the puff at \( t \)
\( F(y) = \) exponential function that describes the horizontal concentration distribution
\( G(z) = \) set of terms describing the vertical concentration distribution
\( \sigma_y, \sigma_z = \) diffusion coefficients that describe the spread of the puff in the horizontal and vertical, respectively (referred to as sigma \( y \) and sigma \( z \)).

\( F(y) \) is defined by

\[
F(y) = \exp[-r^2/(2\sigma^2)]
\]

2.13
where

\[ r^2 = (x - x_0)^2 + (y - y_0)^2 \]

and \( x_0, y_0 \) represent the position of the puff center.

Definition of \( G(z) \) requires further discussion of the modeling assumptions. The height of the puff center above ground has been assumed to be constant. This height is referred to as the effective release height. In the current version of MESOILT2, it is the stack or vent height. Plume rise is not included in this version of the code. The elevation of the puff center is the sum of the elevation of the terrain immediately beneath the center of the puff and the effective release height. As the puff is transported downwind, it rises and falls, conforming to the changes in terrain elevation. Real plumes do not always behave in this manner. Under appropriate meteorological conditions, plumes may intersect rising terrain or may not descend with falling terrain.

In MESOILT2, the ground and the top of mixing layer are assumed to be totally reflecting surfaces for material within the mixing layer. The top of the mixing layer is not a reflecting surface for material above the mixing layer. Consequently, the top of the mixing layer is similar to a semi-permeable membrane.

\( G(z) \) describes both the vertical diffusion of material and the effects of the reflection. It is an infinite sum that involves superposition of contributions from virtual sources located below the ground and above the top of the mixing layer. This approach follows from the discussion in Csanady (1973) and is described in detail in Ramsdell et al. (1983). When the receptor is at ground level, as the receptors are in MESOILT2, \( G(z) \) is given by

\[
G(z) = 2 \sum_{n=-\infty}^{\infty} \exp[-0.5 \frac{(2nH - h_e)^2}{\sigma_z^2}]
\]

where \( H \) is the mixing layer height and \( h_e \) is the effective release height. The infinite sum of exponential terms rapidly converges to a limit. Only the
terms with \( n = -1, 0, \) and 1 are used in MESOILT2. When the vertical diffusion coefficient becomes sufficiently large (\( \sigma_z \approx 0.8 \, H \) or \( \sigma_z \approx 0.8 \, h_e \), whichever is larger), material may be assumed to be uniformly distributed in the vertical. In this case, \( G(z) \) is given by

\[
G(z) = \begin{cases} 
(2\pi)^{1/2} \sigma_z/2H & \text{if } h_e \leq H \\
(2\pi)^{1/2} \sigma_z/2h_e & \text{if } h_e > H
\end{cases}
\]

and the concentration in the puff is given by

\[
x_i(x,y,z,t) = Q(t) \frac{F(y)}{[(2\pi)\sigma_y \, H]}
\]

or

\[
x_i(x,y,z,t) = Q(t) \frac{F(y)}{[(2\pi)\sigma_y \, h_e]}
\]

MESOILT2 uses the difference in elevation between the center of the puffs and the elevation of each grid point in diffusion computations. The elevation of the ground beneath the center of the puff at each time step is determined using bilinear interpolation from elevations at the surrounding grid points. If a topographic data file is not available, the model will run assuming that the model domain is flat.

Time-integrated concentrations are primary products of MESOILT2. These values are computed at each node in the model domain from the puff concentrations using the approximation

\[
T/\Delta t \sum_{j=1}^{N_j} \sum_{i=1}^{N_i} x_{i,j}(r) \Delta t
\]

where \( \text{TIC}(1,m) = \text{time-integrated concentration at node } l,m \)

\( T = \text{total time period that is being modeled} \)

\( \Delta t = \text{duration of the time interval} \)

\( j = \text{model interval within } T \)

2.15
\[ N_j = \text{number of puffs at time interval } j \]
\[ i = \text{puff number} \]
\[ X_{i,j} = \text{concentration at } l,m \text{ due to puff } i \text{ at time interval } j \]
\[ r = \text{distance between } l,m \text{ and the center of puff } i. \]

The accuracy of this approximation depends upon the ratio of puff dimensions to the separation between puffs. Decreasing the time step used in the calculation increases the accuracy of the approximation. However, it also increases computational time. Figure 2.2 shows the range of potential errors in time-integrated concentrations as a function of the ratio between distance moved and \( \sigma_y \), assuming a Gaussian concentration distribution within the puffs. When the distance moved is less than \( 2\sigma_y \), the maximum error in time-integrated concentrations and average deposition rates are less than 5%.

MESOILT2 adjusts the time step used in diffusion calculations based on \( \sigma_y \) and the puff transport speed. If the ratio between \( \sigma_y \) for a puff and the distance moved during 15 minutes is greater than 2, the time step used for calculation of the puff's contribution to time-integrated concentrations,
average deposition rates, and month-end surface contamination is reduced to 5 minutes. Considering the distances involved in the HEDR Phase I calculations, further reduction of the time step used in accumulation of time-integrated concentrations would not be accompanied by a corresponding improvement in estimates of the integrated values. The dashed lines show that the ratio between distance moved and $2\sigma_y$ could be relaxed to almost four before the range of potential errors in the integrated values would increase to plus or minus a factor of 2.

2.3.2 Diffusion

Many experiments have been conducted to measure atmospheric diffusion coefficients. Results from various tests have been combined to develop diffusion coefficient parameterizations. In general the parameterizations are a function of atmospheric stability and travel distance or time. Four parameterizations are included in the MESOI/MESORAD suite of models to provide users with alternatives. One of those alternatives, the U.S. Nuclear Regulatory Commission (NRC) parameterization used in the MESODIF-II, XOQDOQ, and PAVAN models (Powell et al. 1979; Sagendorf et al. 1982; and Bander 1982, respectively), was selected for use in MESOILT2 because it has been used for environmental dose calculations for the last few years (e.g., Price 1986, et seq.).

The NRC diffusion coefficient parameterizations have been attributed to Eimutis and Konicek (1972), but in fact the parameterization for $\sigma_y$ is properly attributed to Tadmor and Gur (1969), and the $\sigma_z$ parameterization is properly attributed to Martin and Tikvart (1968). These parameterizations have the general form

$$\sigma = ax^b + c$$

where x is the distance downwind and a, b, and c are model parameters. For $\sigma_y$, a is a function of stability, and b and c are constants having values of 0.9031 and 0.0, respectively. For $\sigma_z$, a, b, and c depend on stability. In addition, they have different values for three different distance ranges. Table 2.1 gives the parameter values used in calculating $\sigma_y$, and the
TABLE 2.1. Parameter Values for Calculating Diffusion Coefficients Using the NRC Parameterization

<table>
<thead>
<tr>
<th>Stability Class</th>
<th>$\sigma_y$ (a)</th>
<th>$\sigma_z$ (a, b, c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 (A)</td>
<td>0.3658</td>
<td>0.00024, 2.094, -9.6</td>
</tr>
<tr>
<td>2 (B)</td>
<td>0.2751</td>
<td>0.055, 1.098, 2.0</td>
</tr>
<tr>
<td>3 (C)</td>
<td>0.2089</td>
<td>0.113, 0.911, 0.0</td>
</tr>
<tr>
<td>4 (D)</td>
<td>0.1471</td>
<td>1.26, 0.516, -13.0</td>
</tr>
<tr>
<td>5 (E)</td>
<td>0.1046</td>
<td>6.73, 0.305, -34.0</td>
</tr>
<tr>
<td>6 (F)</td>
<td>0.0722</td>
<td>18.05, 0.18, -48.6</td>
</tr>
<tr>
<td>7 (G)</td>
<td>0.0481</td>
<td>10.83, 0.18, -29.2</td>
</tr>
</tbody>
</table>

parameters used in calculating $\sigma_z$ for distances of 1 km or more. Figures 2.3 and 2.4 show the relationships between the diffusion coefficients and distance represented in the NRC parameterization.

Two constraints are applied to the diffusion coefficients. The first is that neither the horizontal nor the vertical diffusion coefficient is permitted to decrease. Allowing the diffusion coefficients to decrease corresponds to compressing the puffs, which is contrary to the third law of thermodynamics.

The second constraint is applied only to the vertical diffusion coefficient. Vertical diffusion is limited to the layer of the atmosphere next to the earth's surface. This layer is referred to as the mixing layer. The earth's surface and the top of the mixing layer are assumed to be reflecting boundaries. For material released within the mixing layer, these assumptions lead to concentrations that are uniform in the vertical when the magnitude of the vertical diffusion coefficient equals 80% of the mixing-layer thickness. As a result, the magnitude of the vertical diffusion coefficient is not permitted to exceed 80% of the thickness of the mixing layer, unless the mixing-layer thickness decreases with time. When the vertical diffusion coefficient is greater than 80% of the mixing-layer thickness because the mixing-layer thickness has decreased as a function of time, no further increase in the vertical diffusion coefficient is permitted. If material is released above the
top of the mixing layer, it is permitted to diffuse downward into the mixing layer, where it becomes trapped.

In puff models such as MESOILT2, diffusion coefficients cannot be computed directly from stability and distance or time. If they were, there would be discontinuities in puff dimensions when the stability changed. In effect the plume being simulated would expand instantaneously when stability decreased and contract when stability increased.

To avoid this problem, MESOILT2 uses virtual travel distances, rather than actual distances, to more realistically simulate the effects of changes in stability. At the beginning of each time interval, the stability class for the interval and the diffusion coefficients from the end of the previous interval are used to compute distances to a virtual point source (virtual distances) for the horizontal and vertical coefficients. These virtual distances and the distance moved by the puff during the interval are then used to
compute the diffusion coefficients used in the interval. In this scheme, the diffusion rates change when the stability changes but there are no discontinuities in the diffusion coefficients.

Atmospheric diffusion results from turbulence in the wind, which is only indirectly related to measures of atmospheric stability. The diffusion coefficient parameterization included with MESOILT2 uses seven stability classes to represent the effects of turbulence because turbulence measurements are not routinely available. There are several methods for estimating stability classes from readily available data. Among the more familiar methods are the Pasquill-Gifford-Turner methods (Gifford 1961 and 1976; Turner 1964), and the NRC definitions based on vertical temperature gradient and wind direction fluctuations found in Regulatory Guide 1.23 (USNRC 1974). Stability classes for Phase I were estimated from vertical temperature gradients using the NRC class definitions.

Methods for estimating stability classes have been compared in many papers, for example, Lague et al. (1980), Lalas et al. (1979), Luna and Church (1972), Mitchell (1982), Sedefian and Bennett (1980), Skaggs and Robinson (1976), and Weil (1979). The consensus is that the various methods of determining stability classes give inconsistent results when applied to the common data sets. However, when used for climatological purposes, the various methods tend to give similar results.

2.3.3 Deposition

Once released to the atmosphere, material undergoes a number of processes that reduce the amount which remains suspended in the atmosphere. These processes include dry deposition of material that comes in contact with surfaces, washout of material by precipitation, and radioactive decay. This section briefly describes the treatment of decay, dry deposition, and washout of iodine-131 in MESOILT2.

Basic diffusion computations are performed assuming a standard release that corresponds to a unit release rate for plume models and a nondepleting species. Each puff has associated with it both a nondecaying, nondepositing mass and an amount of iodine-131. As the puff is advected and material deposits, the amount of iodine-131 in the puff is reduced to account for the lost
mass. In addition, at the end of each hour, the amount of iodine in the puff is reduced to account for radioactive decay.

MESOILT2 incorporates a source-depletion dry deposition model. Source-depletion dry deposition models are described in many places in the literature, for example, Hanna et al. (1982), Pasquill and Smith (1983), and Van der Hoven (1968). The essence of the source-depletion model is that the flux of material to the surface is proportional to the concentration near the surface. The constant of proportionality is called the deposition velocity because it has units of length divided by time.

The basic formulation of the model is

\[ \omega_d = v_d \cdot \chi(x, y, 0) \]

where \( \omega_d \) is the dry deposition rate at position \((x, y)\), \( v_d \) is the deposition velocity, and \( \chi(x, y, 0) \) is the concentration near the surface. Sehmel (1980) discusses the measurement and uncertainties associated with deposition velocities and summarizes much of the material on them. In light of Sehmel's discussion, a deposition velocity of 0.01 m/s is assumed for iodine-131.

Deposition rates are used in computing the monthly average deposition rate and month-end surface contamination at each node in the MESOILT2 domain. The average deposition rate at node \( l, m \) is computed as

\[ DR_d(l, m) = \frac{T}{\Delta t} \sum_{j=1}^{N_j} \sum_{i=1}^{N_j} v_d \cdot \chi_{i,j}(r) \cdot \Delta t \]

where

- \( DR_d(l, m) \) = average dry deposition rate at node \( l, m \)
- \( T \) = total time period that is being modeled
- \( \Delta t \) = length of the time interval
- \( j \) = model interval within \( T \)
- \( N_j \) = number of puffs at time interval \( j \)
- \( i \) = puff number
- \( v_d \) = deposition velocity
\[ x_{i,j} = \text{concentration at } l,m \text{ due to puff } i \text{ at time interval } j \]
\[ r = \text{distance between } l,m \text{ and the center of puff } i. \]

The month-end surface contamination resulting from dry deposition at the node is

\[
\frac{T}{\Delta t} \sum_{j=1}^{N_j} \sum_{i=1}^{I} v_d x_{i,j}(r) \Delta t \exp(-\lambda t_r).
\]

The exponential term on the right side of the equation is a radioactive decay term that accounts for decay from the time of deposition to the end of the period. This term is evaluated at beginning of each hour as

\[ t_r = T - j\Delta t. \]

Dry deposition is a surface layer phenomenon; washout, on the other hand is an integral phenomenon. In washout, the amount of material deposited on the surface and removed from the puffs depends on the concentration of material integrated vertically through the puff, and the precipitation type and rate. Hanna et al. (1982) and Slinn (1978) discuss the assumptions and limitations of simple washout models of this sort.

In MESOILT2, the flux to surface at a distance \( r \) from the center of a puff resulting from wet deposition is given by

\[
\omega_w(r) = w_c Q F(r)/(2\pi\sigma_y^2)
\]

where \( w_c \) is the washout coefficient. Washout coefficients used in MESOILT2 are listed in Table 2.2 and are based on data shown by Engelmann (1968).

According to the Federal Meteorological Handbook (DOC 1982), light rain corresponds to rainfall of less than 2.5 mm/hr and rainfall greater than 7.6 mm/hr is heavy. Intensity of snowfall is generally determined from visibility.
TABLE 2.2. Mesoilt2 Washout Coefficients (hr\(^{-1}\))

<table>
<thead>
<tr>
<th>Type</th>
<th>Precipitation Intensity</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Light</td>
<td>Moderate</td>
<td>Heavy</td>
</tr>
<tr>
<td>Liquid</td>
<td>0.79</td>
<td>2.2</td>
<td>4.0</td>
</tr>
<tr>
<td>Frozen</td>
<td>0.36</td>
<td>1.2</td>
<td>2.3</td>
</tr>
</tbody>
</table>

The contributions of wet deposition to the average deposition rate and month-end surface contamination at grid nodes are computed by

\[
DR_{w}(l,m) = \frac{T/\Delta t}{N_j} \sum_{j=1}^{N_j} \sum_{i=1}^{\omega_{dl,j}(r)\Delta t}
\]

and

\[
SC_{w}(l,m) = \frac{T/\Delta t}{N_j} \sum_{j=1}^{N_j} \sum_{i=1}^{\omega_{dl,j}(r)\Delta t \exp(-\lambda t_r)}
\]

where all terms are as previously defined.

The average deposition rates and month-end surface contamination at the grid nodes are the sum of the dry and wet deposition components.

Although the wet deposition process is included in Mesoilt2, wet deposition was not computed during the Phase I calculations because surrogate meteorological data were used in place of the actual data. Hanford Meteorology Station precipitation records (Stone et al. 1983) contain the following statistics, which indicate that this omission is unlikely to be significant in achieving the Phase I objectives.

- The frequency with which precipitation is observed ranges from a low of 1.4% in July to a high of 13.3% in January.
- Daily total rainfall of 2.5 mm or more occurs an average of 23 days per year.
A 2.5 mm/hr rainfall rate will persist for 4 hours an average of every other year.

Heavy rainfall will persist for an hour an average of about once in 3 years.

### 2.3.4 Depletion

MESDILT2 maintains a mass balance in the puffs by accounting for decay and depletion resulting from dry and wet deposition. The amount of material in a puff that is deposited on the surface during each time interval is the integral of the deposition rate over the period and horizontal area of the puff. In MESDILT2, the dry deposition rate is constant during each interval with the result that the decrease in the mass of the puff is

\[
\Delta Q = \Delta t \int_{\theta=0}^{2\pi} \int_{r=0}^{\infty} \omega_d r dr d\theta.
\]

Substituting for \(\omega_d\) and performing the integration, the decrease in material becomes

\[
\Delta Q_d = 2v_d Q \bar{G}(z) \Delta t / [(2\pi)^{1/2} \sigma_z].
\]

In MESDILT2, it is assumed that this loss of material is distributed throughout the entire puff. With this assumption, the mass in each puff is updated at the end of each time step using

\[
Q_{i,j+1} = Q_{i,j} \{1 - 2v_d \bar{G}(z) \Delta t / [(2\pi)^{1/2} \sigma_z]\}.
\]

This is called a source depletion model.

Horst (1977) suggests an alternative to the source depletion model called a surface depletion model. However, the surface depletion model requires a large amount of computational time. Results similar to those obtained with the surface depletion model can be obtained with a source depletion model and a deposition velocity that is computed as function of wind speed, atmospheric
stability, and surface roughness. Incorporation of this model, which is described by Seinfeld (1986), will be considered during later phases of the HEDR Project.

During periods of precipitation, the rate of mass loss from a puff by precipitation can be determined by integrating the wet deposition rate over the area covered by the puff.

\[
\Delta Q_W = \Delta t \int_{\theta=0}^{2\pi} \int_{r=0}^{\infty} \omega_w(r) \, r \, dr \, d\theta
\]

When this integration is carried out, the rate of loss is found to be equal to the product of the mass in the puff and the washout coefficient. Therefore, the change in mass puffs resulting from precipitation is computed as

\[
Q_{i,j+1} = Q_{i,j} \exp(-w_c \Delta t).
\]

Again, the loss of mass is assumed to be distributed throughout the puff. However, in the case of wet deposition, this is a reasonable assumption.

MESOILT2 computes depletion at the end of each time step. If there is no precipitation, the mass remaining in the puff is computed assuming only dry deposition as described above. When there is precipitation, the depletion calculation is

\[
Q_{i,j+1} = Q_{i,j} \left[1 - 2 v_d Q G(z) \frac{\Delta t}{[\sqrt{2\pi} \sigma_z]} \exp(-w_c \Delta t)\right].
\]

The final correction made to the mass in a puff is to account for the decay during transit. At the end of each hour, the mass is reduced using

\[
Q_{i,j+1} = Q_{i,j} \exp(-\lambda A \Delta t)
\]

where \(\Delta t\) is 15 minutes.
2.4 CENSUS DIVISION ESTIMATES

MESOILT2 computations are made at nodes on a 31-by-31 Cartesian grid, but Phase I dose estimates are to be made for the HEDR census divisions shown in Figure 2.5. A post-processor computer program, PHASEII was developed to estimate census division values of monthly time-integrated atmospheric concentrations, deposition rates, and month-end surface contaminations from the MESOILT2 output. This process involves averaging the values at nodes in and near the census divisions.

The transport modeling domain, 160 mi (256 km) on a side is centered at 46°37'30"N latitude, 119°00'W longitude. It extends from the Cascade Mountains eastward to approximately the Washington-Idaho border. Spokane, Washington, is near the northeast corner of the domain, and The Dalles, Oregon, is near the southwest corner. The 200 Areas of the Hanford Site, where most of the atmospheric releases of iodine-131 occurred, are about 25 mi (40 km) east of the center of the domain. Figure 2.6 shows the relationship between the atmospheric modeling domain and the 10-county area. County names are shown in the figure in light print, and selected population centers are shown in bold print.

The MESOILT2 domain for Phase I was offset from the 10-county region for three reasons. The first reason is that the prevailing wind directions at Hanford have westerly components, which carry material released at Hanford to the east. The second reason is that winds with easterly components, which carry material to the west, are infrequent, generally light, and do not persist. The third reason is that the terrain to the west of the MESOILT2 domain is mountainous. Concentration and deposition estimates for the western boundary of the MESOILT2 domain were assumed to be conservative estimates of the concentrations and deposition in the portion of the 10-county region to the west of the model domain.

2.4.1 Predicted Values

MESOILT2 computed monthly time-integrated atmospheric concentrations, monthly deposition rates, and month-end surface contamination at nodes within the model domain for each month from January 1983 through December 1987.
FIGURE 2.5. HEDR Census Divisions
These values cannot be used directly to estimate doses for the census districts, but they can be used to estimate diffusion factors that represent typical atmospheric transport and diffusion conditions. The PHASEII program computes the typical diffusion factors from the MESOILT2 output.

MESOILT2 results indicate that monthly time-integrated concentrations at nodes within a small area have an approximately log-normal distribution. As a result, the PHASEII program computes averages of the logarithms (base 10) of the time-integrated concentrations, deposition rates and surface
contaminations for each census division. Antilogarithms of these averages are referred to as geometric means. In log-normal distributions, the geometric mean is also the median of the distribution.

Dimensions of several of the census divisions are smaller than can be resolved on the MESOILT2 grid. Computations for these divisions are made using nodes surrounding the division. Similarly, there are two instances where it is not possible to distinguish between two divisions (Ellensburg and Kittitas Division 9, and Walla Walla Division 9 and College Place). In each case, a single set of computations is used to characterize the atmospheric transport and deposition for both points. There are also several census divisions that are entirely off the MESOILT2 grid. In each of these cases, the census division values should be lower than in nearby divisions for which computations were made. Values for the nearby divisions should be used in these cases.

A number of nodes in the MESOILT2 domain did not experience any airborne concentrations or deposition during some months. This poses a computational problem because the logarithm of zero is undefined. Background values are assigned to these nodes in PHASEII, but the nodes are not used in computations for the census divisions. Neglecting the background values results in overestimation of the geometric mean and underestimation of the uncertainty.

2.4.2 Uncertainty

PHASEII provides estimates of the uncertainty associated with the averages of the logarithms as well as the averages. These estimates include three components. The first component is the variability of values at the nodes within a census division; the second is related to the year-to-year variations in atmospheric conditions; and the third is related to the basic uncertainty associated with the MESOILT2 model.

Variances of the monthly values that are above background at the nodes associated with each census division are computed by PHASEII as well as the average values. These variances are the measure of the uncertainty of the census division values resulting from spatial variations in the monthly output from MESOILT2 for each year and from the year-to-year variations in the
meteorological conditions. This variance, which covers the first two components of the uncertainty, is combined with an estimate of the uncertainty of the basic model.

No measures of the uncertainty of MESOILT2 existed at the time the Phase I calculations were made. However, the earlier climatological model had been evaluated using monthly average concentrations of krypton-85 measured by the Hanford Environmental Program in the vicinity of Hanford. There are 59 individual samples of 14 to 38 days duration collected at six locations between 19 and 44 km from the release point in the data set. Of the 59 samples, 51 were between 27 and 30 days in duration. Figure 2.7 shows a comparison of the predicted and measured concentrations for these 59 periods.

The model is seen to give slightly biased estimates of the concentration. On the average, the predicted concentrations are a factor of about 2 higher (2.17) than the measured values. If the actual relationship between logarithms of the predicted and measured concentrations is linear as it appears in Figure 2.7, the square of the correlation coefficient, which is commonly denoted as $r^2$, is the fraction of the variation in the measured concentrations that is accounted for by the model. For this data set $r^2$ is about 0.58, which indicates that the model generally accounts for about 58% of the variability in the observed monthly mean concentrations. Analysis of these data is discussed in more detail by Ramsdell (1989, Appendix C). The performance of this model is better than the weighted average performance of all models used to estimate monthly-mean concentrations and is about the same as the performance of the best model in the Savannah River Laboratory model evaluation (Weber et al. 1982).

The variance of the differences between the logarithms of predicted and measured concentrations is a measure of the errors, or uncertainty, in the model. For these data, the variance is 0.13. However, it is based on a prediction of monthly average concentrations of a noble gas. Errors in predicted concentrations of and surface contamination by a depositing material are likely to be larger than for a nondepositing material. Therefore, the
MESOILT2 is arbitrarily assumed to have a variance of differences between the logarithms of predicted and measured concentrations that is twice the variance computed for the earlier model.

With this assumption, the variance for estimates of the census district monthly time-integrated air concentrations is computed as

\[ \sigma_c^2 = \sigma_n^2 + 0.26 \]

where \( \sigma_n^2 \) is the variance of concentrations at all nodes in the census district during the 5-year period. Similar equations are used to estimate variances of the average deposition rate and surface contamination. Probability
statements may be made given these variances and the log-normal assumption. For example, assuming that there are no spatial variations within the census district or temporal variations among the 5 years (i.e., $\sigma_n^2 = 0$), the geometric standard deviation of errors in model predictions would be $10$ raised to the square root of $0.26$ power, or $3.24$. This value indicates that the minimum range of values required to be $90\%$ certain of containing the true value is

$$g.m./5.32 < x < 5.32 \, g.m.$$

where $g.m.$ is the geometric mean, $x$ is the true value, and $5.32$ is $3.24$ times $1.645$. Determination of the significance and confidence intervals for the correlation coefficients and predicted values is covered in statistical texts (e.g., Brownlee 1965 and Snedecor and Cochran 1980).

MESOILT2 calculations were made assuming $1000 \, \text{Ci}$ were released each month. The PHASEII program normalizes all monthly atmospheric concentrations and surface contamination values to a $1\,-\text{Ci}$ release. This permits later stages of the dose estimation process to adjust the predicted time-integrated concentrations, deposition rates and surface contamination to a specific month in the study period by multiplying the census district averages by the actual monthly iodine-131 release.

2.5 MODEL EVALUATION

In the overall dose estimation process described by Napier (1990), the output of the PHASEII model is used to estimate iodine-131 concentrations in vegetation. This next stage in the process combines the deposition rate estimates with information describing the interception and uptake of material by vegetation, and information on vegetation density. It uses a Monte Carlo procedure to estimate concentrations in vegetation. This stage is illustrated in Figure 2.8.

The MESOILT2 and PHASEII calculations for Phase I were completed in July 1989. Since then the census district deposition rate estimates have been used to estimate iodine-131 concentrations in sage brush. These estimates are important to the atmospheric transport diffusion modeling effort because they may be compared with environmental monitoring data to make an initial
evaluation of the MESOILT2 model. If the concentrations in vegetation estimated by the models compare favorably with the monitoring data, the model may be assume to be reasonable.

Figure 2.9 compares predicted and measured iodine-131 concentrations in sage brush in six census districts in Benton and Franklin counties for months with at least three weekly samples. The measured data were obtained from September 1945 through December 1947, and the predicted data are based on an assumed iodine-131 release at an elevation of 61 m from a point midway between the 200-East and 200-West Areas. The figure shows that there is reasonable agreement between the predicted and measured concentrations. The value of $r^2$ between the logarithms of these predicted and measured concentrations is 0.61, which is about the same as for earlier model. However, the iodine-131 releases decreased during this period from a high of about 47,000 Ci in October 1945 to a low of about 250 Ci in November 1947. This decrease is responsible, in part, for the high correlation. A better evaluation of the atmospheric model performance can be made if both predicted and measured concentrations are normalized by the monthly releases. Figure 2.10 compares the normalized concentrations. In this case, $r^2$ is 0.52, which is still significant but not as good as the $r^2$ between the unnormalized concentrations. In either case, the predicted concentrations tend to be a factor of about 1.5 higher than the measured values. Of the 68 monthly mean census district concentrations predicted, the largest under prediction was less than a factor of 4, and the largest over prediction was less than a factor of 10.
FIGURE 2.9. Comparison of Preliminary Phase I Predictions of Iodine-131 Concentrations in Sage with Average Measured Values for Census Districts

Measured and predicted concentrations in sage are shown in Figure 2.11 for four census divisions. The Benton 2 division contains the Hanford Site, and the Benton 7, Benton 3, and Franklin 3 divisions contain Richland, Kennewick, and Pasco, respectively. Collectively, these plots show the tendency of the model predictions to track the measured concentrations as they change in time.
Collectively, Figures 2.9, 2.10, and 2.11 indicate that the procedures and models used in Phase I are reasonable. However, these figures should not be interpreted as a complete validation of the models and procedures. The census districts in which the monitoring data were collected are all close to the Hanford Site and generally downwind in the prevailing transport direction. As a result, the models and procedures are being evaluated in the most favorable light. It is likely that their performance in other areas is not as good.
FIGURE 2.11. Temporal Variation of Measured and Predicted Concentrations of Iodine-131 in Sage in Four Census Divisions
3.0 MESOILT2 USER'S GUIDE

MESOILT2 is a climatological dispersion model that uses a Lagrangian trajectory, Gaussian puff approach to estimate dispersion. The computer code implementing the model is written in FORTRAN for execution on VAX® 11/780 computers using the VMS™ operating system.

In general, the code conforms to the FORTRAN 77 standard (ANSI 1978). However, it makes use of several extensions to the standard including

- INCLUDE (.INC) files for TYPE, DIMENSION, and COMMON statements
- long variable names
- DO and ENDDO statements.

These extensions are part of many FORTRAN implementations. If the code is to be compiled using a compiler that doesn’t support these extensions, the non-standard statements are easily replaced with standard statements. For example, the entire INCLUDE file should be inserted wherever there is an INCLUDE statement. Long variable names may be shortened as long as they remain unique, or they may be changed, and the DO - ENDDO construct may be changed to a DO xxx, xxx CONTINUE construct where xxx is a numerical statement label.

MESOILT2 also contains calls to several VAX 11/780 system subroutines. The specific subroutines called are SECOND, DATE, and TIME. SECOND is used to determine elapsed time used in various portions of the MESOILT2 code. Calls to SECOND may be deleted if the elapsed time information isn’t needed, or they may be replaced with corresponding system routines for other computers. DATE and TIME are used to obtain the date and time that MESOILT2 execution begins. These values are entered on the MESOILT2 log and are added to names of the programs primary output files. Calls to DATE and TIME may be replaced by corresponding systems calls for other routines; however, a date and time must be available for use by the program.

*VAX and VMS are trademarks of Digital Equipment Corporation, Maynard, Massachusetts.*
3.1 PROGRAM EXECUTION

MESOILT2 is written as an interactive program. That is, when the program is run in its normal mode, the code requests information directly from users. This mode of operation is a remnant of the codes from which MESOILT2 was developed. For HEDR Phase I, MESOILT2 was run in a batch mode using the VMS Command Language SUBMIT command. The SUBMIT command has the form SUBMIT NAME where NAME is the name of a command file that initiates MESOILT2 execution and contains the responses to the MESOILT2 user prompts.

Table 3.1 shows a typical command file that could be used for atmospheric transport and diffusion calculations for January 1984. It contains 18 records. The first of these records is the command to start MESOILT2 execution, and the remaining 17 records contain the information that controls

<table>
<thead>
<tr>
<th>No.</th>
<th>Input</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$ RUN MESOILT2</td>
<td>! Start MESOILT2 execution</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>! Computation option; 0 or 1</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>! Minimum time step; 1, 3, 5 or 15</td>
</tr>
<tr>
<td>4</td>
<td>MET84JAN.DAT</td>
<td>! Meteorological data file name</td>
</tr>
<tr>
<td>5</td>
<td>HORTEL</td>
<td>! Meteorological station file name</td>
</tr>
<tr>
<td>6</td>
<td>HEDRTOP</td>
<td>! Topographic data file name</td>
</tr>
<tr>
<td>7</td>
<td>JANUARY 1984</td>
<td>! Run identification</td>
</tr>
<tr>
<td>8</td>
<td>N</td>
<td>! Hourly meteorological data; Y or N</td>
</tr>
<tr>
<td>9</td>
<td>010184</td>
<td>! Date to start model run</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>! Hour of the day to start model run</td>
</tr>
<tr>
<td>11</td>
<td>WINDU37.DAT</td>
<td>! Wind speed units conversion file name</td>
</tr>
<tr>
<td>12</td>
<td>Y</td>
<td>! Adjust wind field for terrain; Y or N</td>
</tr>
<tr>
<td>13</td>
<td>-44.21,-6.89,61.0</td>
<td>! Position of source</td>
</tr>
<tr>
<td>14</td>
<td>010184</td>
<td>! Date of initial release</td>
</tr>
<tr>
<td>15</td>
<td>00,00</td>
<td>! Hour and minute of initial release</td>
</tr>
<tr>
<td>16</td>
<td>013184</td>
<td>! Date of end of model run</td>
</tr>
<tr>
<td>17</td>
<td>24</td>
<td>! Hour of end of model run</td>
</tr>
<tr>
<td>18</td>
<td>PUFFS_DEC83.BIN</td>
<td>! File name for puffs from previous month</td>
</tr>
</tbody>
</table>

TABLE 3.1. MESOILT2 Command File for January 1984
the program. All of the records shown are mandatory. In each record, an apostrophe separates the command or model input from a brief description of the record content. The record numbers at the beginning of each record are not part of the record. More complete descriptions of the records are provided in the following paragraphs.

The $ in the first record is the VAX system prompt. It indicates that the record contains a VAX system command. In this case, it is the RUN command, and MESOILT2 is the name of the program to be executed.

MESOILT2 has two computational options. It computes time-integrated concentrations for a nondepositing, nondecaying gas in both options. This gas might be a long-lived noble gas such as krypton-85, or some other inert tracer. A one should be entered in the second record if this is the only computational product desired. The alternative is to enter a zero in the second record. This will cause MESOILT2 to compute the time-integrated air concentration, average deposition rate, and month-end surface contamination for iodine-131 in addition to the time-integrated air concentration for an inert gas. The iodine-131 half-life is a model parameter. Modeling deposition, depletion, and decay of other radionuclides requires changing and recompiling of the code.

The third record controls the minimum time step used in the model. Puff movement is calculated from wind data every 15 minutes, but when the puffs are small they are moved in smaller time steps to improve the accuracy of the summations used to compute the time-integrated air concentration, average deposition rate, and surface contamination. This record controls the size of smaller time step. The number entered must be an integer factor of 15.

The fourth and fifth records specify the names of files that contain the meteorological data and the locations of the meteorological stations to be used in the model run. Both records must contain names of files that exist. If either file is missing or unreadable, program execution will abort. The contents and format of these files are described later.

The sixth record specifies the name of a file containing the topographic data for the model run. These data are the terrain elevations used in diffusion calculations. They are not the data used to adjust wind fields. This 3.3
record is required, but it may be blank. If the record is blank or the file named cannot be read by the MESOILT2, the program will execute assuming flat terrain.

The seventh record supplies a character string that is used to identify the model run. This string may contain as many as 80 characters. If the first character in the string is an asterisk, a program option will be activated that generates additional model output which is used for code verification. This output includes position, distance moved, diffusion coefficients, and mass in each puff at the end of each 15-minute period.

The eighth record defines the meteorological data frequency. Hourly meteorological data is the default assumption in MESOILT2. It is specified by entering a Y in the eighth record. The alternative is data taken every 3 hours. This is selected by entering an N. MESOILT2 does not check the times in the meteorological data file to confirm the option selected here is correct. Meteorological data will be read at the correct times but the interpolation will be by error if the wrong option is selected.

The ninth and tenth records specify the date and time that the model simulation begins. These values are used in searching the meteorological data file for the initial meteorological data record. The date must be entered in the ninth record as a six-digit integer in the form mmddyy where mm is the month, dd is the day, and yy is the year. Leading zeros must be entered. The year must be 82 (1982) or later. If a date before January 1, 1982, is entered, program execution will be aborted. Similarly, program execution will be aborted if either the dd or mm is less than 1, if dd is greater than 31, or if mm is greater than 12. The tenth record contains the hour that the simulation begins. For the purpose of this record, the day begins at hour 00 (midnight), and the last hour of the day is 23 (11 pm). Leading zeros may be dropped for the first 9 hours of the day. Entries less than 00 or greater than 23 will cause program execution to abort.

Depending on the data source, wind speeds are reported in miles per hour, meters per second, or knots (nautical miles per hour). The eleventh record contains the name for a file designating how wind speeds are reported
for each station. These data are used in MESOILT2 to select the appropriate factors to convert speeds to meters per second.

The twelfth record controls adjustment of the wind fields for terrain effects. This option is selected by entering a Y and bypassed by entering an N. If the option is selected, the program expects to find a file named TERRAN.DAT, which contains the information needed to make the adjustments. If it does not find or cannot open the file, the program will continue without adjusting the wind fields. If the file is found and read successfully, messages will be included in the MESOILT2 log file stating that wind fields will be adjusted and giving the number of nodes at which the wind will be modified. If an error occurs reading the file, program execution will stop.

The position of the release point is entered in the thirteenth record. Three entries are expected. The first two entries give the horizontal position of the release point relative to the center of the model domain. These positions are entered as distances east and north of the center in kilometers. Positions west and south of the domain center are entered as negative numbers. Release points must be within the model domain. The third entry is the height of the release point above the ground. It is entered in meters. Release heights are constrained to be between zero and 300. If the position of the release point is out of range, the program will attempt to reread the position and should ultimately abort the run.

The fourteenth and fifteenth records specify the date and time for the beginning of the initial release. The date given in the fourteenth record as mmddyy is compared with the date given for the beginning of the simulation. If it is before the start of the simulation, the program will attempt to reread the position and should ultimately abort the run. The fifteenth record contains the time of day of the start of the initial release in hours and minutes in the form hh,mm. Hours are entered as 00 to 23, and minutes as 00 to 59. An improper time will cause the program to abort.

The sixteenth and seventeenth records give the date and time for the end of the simulation. The date is contained in the sixteenth record in the same form as the dates for starting the simulation and the date of the initial
release. Record 17 gives the hour of the end of the simulation. This number is the last hour of the day to be completed and should be between 01 and 24.

The final record in the command file is used to specify the name of file containing data on puffs in the model domain at the beginning of the simulation. This is a mandatory record, but a blank record may be used to indicate that no puff data from a previous model run are to be read. If a file name is specified and the file is not found or cannot be opened, program execution will abort.

3.2 INPUT DATA FILES

Data required to run Mesoilt2 are entered via input data files. There are seven data input files in addition to the command file used to control the model execution. These files are listed by contents in Table 3.2. Names for five of the files are passed through entries in the Command File. The other two file names are set within Mesoilt2. Required names for these files are given in the table. The Access and Form columns in the table describe the file structure and form. Direct access files are used for the time-dependent meteorological and source-term data. These files must be created with a program using a formatted write statement. Once created, the files cannot be edited using a text editor. The residual puff data file is a sequential access, binary file created by Mesoilt2 using an unformatted write statement. It is read by a read statement that matches the write statement. There should

<table>
<thead>
<tr>
<th>Contents</th>
<th>Name</th>
<th>Access</th>
<th>Form</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Meteorological data</td>
<td>Optional</td>
<td>Direct</td>
<td>Formatted</td>
<td>Required</td>
</tr>
<tr>
<td>Meteorological stations</td>
<td>Optional</td>
<td>Sequential</td>
<td>Formatted</td>
<td>Required</td>
</tr>
<tr>
<td>Wind speed units</td>
<td>Optional</td>
<td>Sequential</td>
<td>Formatted</td>
<td>Required</td>
</tr>
<tr>
<td>Topographical data</td>
<td>Optional</td>
<td>Sequential</td>
<td>Formatted</td>
<td>Optional</td>
</tr>
<tr>
<td>Wind adjustment data</td>
<td>TERRAN.DAT</td>
<td>Sequential</td>
<td>Formatted</td>
<td>Optional</td>
</tr>
<tr>
<td>Source-term data</td>
<td>QFILE.DAT</td>
<td>Direct</td>
<td>Formatted</td>
<td>Required</td>
</tr>
<tr>
<td>Residual puff data</td>
<td>Optional</td>
<td>Sequential</td>
<td>Unformatted</td>
<td>Optional</td>
</tr>
</tbody>
</table>

TABLE 3.2. Mesoilt2 Input File Summary
not be any need to modify the residual puff data files. The remaining input
files are sequential access, formatted files. These files may be created and
modified using a text editor.

The following paragraphs describe the input files in more detail.

3.2.1 Meteorological Data File

The meteorological data file is a direct access, formatted file with a
record length of 234 characters. It is read in subroutine DATRDLT. Each time
DATRDLT is called, the file is accessed twice. The first time the file is
accessed, the subroutine reads the full data record to obtain data used in
transport and diffusion calculations. The second time the file is accessed,
the subroutine obtains the data and time of the next set of meteorological
data.

Data are stored in the record in the following order: year, day of the
year, hour, minute, stability class, mixing-layer thickness, precipitation
class, top-layer wind, surface air temperature, release height air tempera­
ture, surface-layer winds. Wind data are read as four-digit integers. The
first two digits in each integer give the wind direction rounded to the near­
est 10°, and the last two digits give the speed. Surface wind data may be
entered for 40 locations. The record format is

\[(1X,I2,I3,I2,I2,1X,I1,1X,I3,1X,1X,1X,I4,1X,I3,1X,I3,40(1X,I4)).\]

The units and ranges for meteorological variables are listed in Table 3.3.

3.2.2 Meteorological Stations File

 Meteorological station locations are entered via the meteorological sta­
tions file, which is read in subroutine GRIDINLT. This file is a formatted,
sequential access file that may be created and edited with a text editor.
There must be a station entry for each location for which wind data are to be
provided, and the station order must correspond to the order of the wind data.
### TABLE 3.3. Units and Ranges for Meteorological Variables

<table>
<thead>
<tr>
<th>Variable</th>
<th>Units</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Year</td>
<td>none</td>
<td>82 through 99</td>
</tr>
<tr>
<td>Day</td>
<td>none</td>
<td>001 through 366</td>
</tr>
<tr>
<td>Hour</td>
<td>none</td>
<td>00 through 23</td>
</tr>
<tr>
<td>Minute</td>
<td>none</td>
<td>00 through 59</td>
</tr>
<tr>
<td>Stability class</td>
<td>none</td>
<td>1 through 7</td>
</tr>
<tr>
<td>Mixing layer thickness</td>
<td>tens of meters</td>
<td>1 through 999</td>
</tr>
<tr>
<td>Precipitation class</td>
<td>none</td>
<td>0 through 6</td>
</tr>
<tr>
<td>Top-Layer wind</td>
<td>degrees/as listed in file for 1st surface wind station</td>
<td>0000 through 3675, 8888 and 9999</td>
</tr>
<tr>
<td>Surface temperature</td>
<td>°F</td>
<td>none</td>
</tr>
<tr>
<td>Release ht. temp.</td>
<td>°F</td>
<td>none</td>
</tr>
<tr>
<td>Surface winds</td>
<td>degrees/as listed in file</td>
<td>0000 through 3699, 8888 and 9999</td>
</tr>
</tbody>
</table>

Records in the file contain the station name, the position of the station relative to the center of the domain, the station elevation, and a status flag. The format for the file is

\[(1X,A4,2(2X,6F6.2),4X,I4,4X,I1)].\]

Station names consist of four alpha-numeric characters selected by the user. They are only used for identification in the MESOILT2 log and are not required. Station position is required and is specified by a pair of numbers that are the distances east and north of the center of the domain in kilometers. Positions west and south of the center are indicated by negative numbers. The stations are not required to be in the model domain. A station elevation is an optional entry. If entered, it is the elevation of the station in meters above sea level. The status flag is a switch that may be used to eliminate specific stations from consideration in calculating wind fields. The station status must be one if data from the station are to be considered, otherwise the data for the station will be ignored.
3.2.3 Wind Speed Units File

The wind speed units file consists of a record for each meteorological station that has a single integer to indicate the units in which the wind speed are reported. In Mesoillt2, this integer determines the factor used to convert the reported wind speed to meters per second. The wind speed code is 1 = meter per second, 2 = miles per hour, and 3 = knots.

This file is a sequential access, formatted file that may be created and modified with a text editor, which is read in subroutine INITLT. Records in the file have a (IX,II) format. If the file doesn't exist or Mesoillt2 is unable to read the data in the file, program execution will be aborted.

3.2.4 Topographical Data File

The topographic data file is used to enter terrain elevations for the diffusion calculations. It is an optional sequential access, formatted file that may be created or modified by a text editor. If the command file does not contain a name for the file or the Mesoillt2 is unable to open the file named in the command file, the program will assume flat terrain and continue. A message to this effect will be written to the Mesoillt2 log.

The topographic data file is read in subroutine TOPFIL. It consists of up to 31 records, with each record containing elevation data for 31 nodes in the Mesoillt2 domain. When the file is read, elevation data are expected to begin in the northwest corner of the model domain. The order of the records is from north to south, and within the records elevations are entered from west to east. If an end-of-file is encountered while the topographic data are being read, the remaining elevations will be assumed to be zero.

All elevations are in meters above sea level and are entered as integers. The record format is (IX,3I14).

3.2.5 Wind Adjustment Data File

Adjustment of winds for expected terrain is optional in Mesoillt2. This option is selected in the command file. If selected, Mesoillt2 will attempt to read a file named TERRAN.DAT in subroutine TERRALT to obtain data to be used.
in these adjustments. If the file is not found or cannot be opened, the pro-
gram will write a message to the log file and will continue as if the option
was not selected.

TERRAN.DAT is a sequential access, formatted file that may be created and
modified with a text editor. It contains a record for each node where the
wind is to be modified. Thus, the number of records in the file depends on
the site; records are read until an end-of-file is reached. MESOILT2 will
abort if an error occurs while reading TERRAN.DAT.

Each record in the file contains the position of node in the model
domain, the slope orientation angle, and two scaling coefficients. The
position of the node is entered in grid coordinates as a pair of integers.
The first integer gives the east-west coordinate, and the second gives the
north-south coordinate. For this file, the coordinate for the southwest
corner of the domain is 0,0, and the coordinate of the northeast corner is
15,15. Subroutine TERRALT adds one to each of the coordinates for use in the
program. The slope orientation angle, as defined in Section 2.0 related to
wind fields, is the direction of the vector perpendicular to the gradient of
the slope. It must be between 0° and 179°. The scaling coefficients are the
fractions by which the upslope component of transport vector will be reduced.
If transport is from the left when facing in the direction of the slope ori-
tentation, the first scaling coefficient is used; the other scaling coefficient
is used when transport is from the right. The TERRAN.DAT record format is
(IX,I2,1X,I2,F4.0,1X,F4.0,1X,F4.0).

3.2.6 Source-Term Data File

Source terms are entered via the QFILE.DAT file. QFILE.DAT is a direct
access, formatted file that must be created using a program. It cannot be
created or modified using a text editor, but it can be viewed with an editor.
The file is read by subroutine READQ, which is called from the main program.
If QFILE.DAT does not exist, cannot be opened, or cannot be read, MESOILT2
will abort.

QFILE.DAT must have a record for the initial source term and an addi-
tional record for each change of source term during the simulation. Each
record contains a year, day, and hour followed by a source term. The year,
day, and hour define the initial time to which the source term applies. The
source term is the amount of material to be released each hour. It defines a
release rate that remains constant until another source term is entered. Each
time subroutine READQ reads a source term, it checks the next record to deter­
mine the time of the next change. Thus, source terms may be changed hourly
during a simulation if appropriate. The QFILE.DAT record format is
$(1X,I2,I3,I2,3X,F12.5)$.

3.2.7 Residual Puff Data File

The residual puff data file provides the means for carrying material
remaining airborne at the end of a simulation forward to a subsequent simula­
tion. A residual puff data file is a sequential access, unformatted file that
is created automatically at the end of every MESOILT2 run. Users have the
option of specification of a residual puff data file name in the command file.
If a name is entered and the file exists, subroutine PUFFIN will read the
file, thereby initializing the simulation with material released during a
previous simulation. If a file name is entered and the file cannot be read,
the program will abort.

Data in the residual puff data file include the title of the MESOILT2 run
that created the file, the date and time of the run creating the file, the
number of puffs for which data are recorded, the position and height of the
center, the original mass, the last movement, the diffusion coefficients, the
depleted mass, a flag, and the source of each puff. The data in the file are
in binary form. As a result, the file cannot be examined or modified using a
text editor.

3.3 OUTPUT FILES

MESOILT2 creates three types of output files: log files, results files,
and residual puff data files. The number of files created depends on the
computational mode option selected in the second record of the command file.
A log file and a residual puff data file are always created. If the computa­
tional mode option is set to one, two results files are created, otherwise
MESOILT2 creates eight results files. The following paragraphs describe the
contents of the output files.

3.11
3.3.1 MESOILT2 Log File

The MESOILT2 log file is the record of the model run. Its heading lists the program name and version, and the date and time of the run. It then lists the results of each stage of the initialization process. Prompts that would be sent to the user’s terminal if the model were being run in an interactive mode are sent to the log file. MESOILT2 then reads the response from the command file and writes it to the log. Interspersed with the prompts and responses are status reports, data from input files, and reports of model parameters settings.

After the model initialization is complete, MESOILT2 logs each access to the QFILE.DAT and to the meteorological data file. The log entry for accessing the source-term file includes the date, time, and new source term. The entry for accessing meteorological data gives the subroutine name, the date, and the time.

If a model run terminates in a normal mode, the message

*** SIMULATION TERMINATED ***

is entered in the log. However, this message does not ensure that the model executed as intended. It may have been unable to read one of the optional data files and continued in a default mode. If the program aborts while trying to read a required file, the terminal message in the log will indicate the file access that caused the program to terminate prematurely.

The last part of the MESOILT2 log is a listing of the names of the output files, other than the log, that were created in the model run. These names indicate the contents and type of file. For example, files containing the average deposition rate for iodine-131 computed using February 1983 meteorological data would have the names I_DR_FEBB3.OUT and I_DR_FEBB3.BIN. The first of these is a file that can be printed or examined at a computer terminal, while the second is an unformatted file for input to post-processing programs.

3.12
3.3.2 **Computational Results Files**

As indicated in the example above, MESOILT2 produces two classes of files that contain the computational results. The first of these is a printable file, and the other is a file for use by post-processing programs. Other than file structure, the primary difference between these classes of files is that the printable files contain headings to assist in identifying the data. The post-processing file contains only the numerical results of the computations. Both classes of file are created by subroutine PRINTELT.

Each time MESOILT2 is run it will create a pair of files that contain the results of transport and diffusion calculations for a nondepositing, nondecaying material. These files begin with the letters NTIC, which stand for nondecaying, nondepositing, time-integrated concentration. The remainder of the name is determined from the calendar month and year. A file extension of .OUT indicates that the file is a print file, while an extension of .BIN indicates that is an unformatted file for input to post-processing programs. Figure 3.1 shows NTIC_MAY87.OUT as an example of these files.

If the computational mode is not set to one, an additional six files are created. Three of the files are printable files, and the other three are binary files for post-processing. Again, the file names indicate the contents of the files. Files that begin ITIC contain the monthly time-integrated air concentrations assuming an iodine-131 release. The material in the puffs has been decreased assuming an 8.05-day half-life, and the puffs have been depleted to account for dry deposition computed assuming a deposition velocity of 0.01 m/s and for wet deposition during precipitation. The month and year for the meteorological data are added to the ITIC as before, as is the proper file extension. Files beginning I_DR contain the monthly average deposition rates, and files beginning I_SC contain the month-end surface contamination. In both cases, material in the puffs is decreased to account for decay and depleted to account for deposition. No further adjustments are made in the deposition rate calculations, but an additional decay factor is applied in calculating the month-end surface concentrations. In these calculations, the material depositing is reduced to account for decay from the time of deposition to the end of the month. This decay is computed assuming an 8.05-day half-life.
### CUMULATIVE EXPOSURE (UNDEPLETED)

**Simulation Hour** 744  **Advection Step** 4  **Results to** **Jun 1, 1987**

#### Output Points

<table>
<thead>
<tr>
<th>X Coordinate of (X)</th>
<th>Y Coordinate of (Y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 2 3 4 5 6 7 8 9 10 11</td>
<td>1 2 3 4 5 6 7 8 9 10 11</td>
</tr>
<tr>
<td>31 9.04E-11 7.4E-12 4.8E-09 1.5E-08 4.8E-08 7.0E-08 9.5E-08 8.5E-08 2.7E-08 1.0E-08 1.4E-08</td>
<td></td>
</tr>
<tr>
<td>38 1.79E-11 2.9E-12 1.8E-09 4.8E-08 7.0E-08 1.5E-07 1.6E-07 4.7E-07 2.3E-06 1.7E-06</td>
<td></td>
</tr>
<tr>
<td>29 6.0E-08 4.3E-11 5.2E-08 5.0E-08 6.0E-08 1.5E-07 1.4E-07 3.6E-07 2.3E-06 2.3E-06</td>
<td></td>
</tr>
<tr>
<td>28 6.0E-08 3.2E-12 1.8E-09 4.8E-08 6.0E-08 1.5E-07 2.2E-07 2.3E-07 8.7E-07 8.9E-07</td>
<td></td>
</tr>
<tr>
<td>27 6.0E-08 9.9E-09 7.4E-12 2.9E-10 3.0E-09 5.1E-08 1.7E-07 4.8E-07 4.2E-07 2.1E-06 1.4E-06</td>
<td></td>
</tr>
<tr>
<td>20 3.9E-08 9.0E-08 9.0E-09 3.8E-11 2.3E-10 5.6E-09 2.5E-08 6.1E-08 8.3E-07 6.8E-07</td>
<td></td>
</tr>
<tr>
<td>25 4.0E-08 9.0E-08 9.0E-09 0.8E-08 1.3E-11 7.9E-08 2.8E-07 7.2E-07 1.1E-06 9.3E-07 3.8E-07</td>
<td></td>
</tr>
<tr>
<td>24 4.0E-08 9.0E-08 9.0E-09 0.8E-08 1.3E-11 7.9E-08 2.8E-07 7.2E-07 1.1E-06 9.3E-07 3.8E-07</td>
<td></td>
</tr>
<tr>
<td>20 4.0E-08 9.0E-08 9.0E-09 0.8E-08 1.3E-11 7.9E-08 2.8E-07 7.2E-07 1.1E-06 9.3E-07 3.8E-07</td>
<td></td>
</tr>
<tr>
<td>26 4.0E-08 9.0E-08 9.0E-09 0.8E-08 1.3E-11 7.9E-08 2.8E-07 7.2E-07 1.1E-06 9.3E-07 3.8E-07</td>
<td></td>
</tr>
<tr>
<td>29 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>11 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>10 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>9 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>8 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>7 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>6 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>5 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>4 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>3 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>2 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
<tr>
<td>1 2.5E-10 8.29E-11 8.94E-10 5.71E-19 2.24E-09 5.39E-08 1.14E-10 7.63E-10 2.45E-08 2.22E-06 1.44E-06</td>
<td></td>
</tr>
</tbody>
</table>

### FIGURE 3.1. An Example of an Output File Containing the Results of MESOILT2 Calculations (May 1987)

3.14
<table>
<thead>
<tr>
<th>POINTS</th>
<th>23</th>
<th>24</th>
<th>25</th>
<th>26</th>
<th>27</th>
<th>28</th>
<th>29</th>
<th>30</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y COORDINATE OF THE GRID POINTS</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>23</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>31</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>25</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>26</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>27</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>28</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>29</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>30</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
<tr>
<td>31</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
<td>2.44E-07</td>
</tr>
</tbody>
</table>

**FIGURE 3.1. (contd)**

3.15
3.3.3 Residual Puff Data File

The last type of output file from MESOILT2 is the residual puff data file. This unformatted file contains information for use in subsequent simulations. These files, which have been discussed already, are the way in which continuity between months is maintained. A residual puff data file is created by subroutine PUFFOUT each time the code is run. Names for the residual puff data files are created by concatenation of PUFFS with the month and year of the meteorological data used when the puffs were released and with the extension .BIN.

3.4 PHASEII POST-PROCESSOR

The MESOILT2 post-processor program PHASEII is used to combine monthly computational results to obtain the typical values and their uncertainty for census divisions. It takes the MESOILT2 files and generates typical time-integrated air concentrations, average deposition rates, and month-end surface contamination for each calendar month. It also computes uncertainty considering the basic uncertainty of the MESOILT2 model, the spatial variation of computational results within the census division each year, and the year-to-year variation in the results.

3.4.1 Input Files

PHASEII expects to find the nodes assigned to each census division in a file named CENDIST.LST. The nodes are read in one census division at a time. There may be as many as 31 nodes for any division. The data for each census division are contained in four records. The first record contains a number giving the position of the division in the file, the division identification, and the first nine MESOILT2 nodes assigned to the division. Each node is specified by a pair of integers giving the x and y coordinates of the node in that order. The format for first record is (IX,I2,2X,A4,1X,9(2X,I2,1X,I2)). The last three records only contain nodes assigned to the division. The format for these records is (IOX,9(2X,I2,1X,I2)). All four records must be present even if the number of nodes assigned to the census division does not require them.
At the completion of the MESOILT2 model runs for HEDR Phase I, 240 unformatted files contain results of the model calculations. For each output product (NTIC, ITIC, I_DR, and I_SC) there are five sets of results for January, five for February, etc. These unformatted MESOILT2 output files are the input to the PHASEII post-processor program. PHASEII reads in 20 MESOILT2 output files at a time (5 years x 4 products per year) for processing. Data in the files are expected to be in the order in which they were written in the unformatted write in MESOILT2. In addition, the type and dimensions of the variables receiving the data in PHASEII must be identical to the type and dimensions of the variables in MESOILT2.

3.4.2 Output Files

The PHASEII post-processor program produces two sets of output files. As before, one set contains results that can be printed or viewed on a computer terminal, and the other contains unformatted data for use as input in the next stage of the dose calculations. A pair of files is created for each calendar month. These files are given the names PHASEII_xxx.OUT and PHASEII_xxx.BIN where xxx is the three-letter abbreviation for the month.

Both files contain the census division identification, the number of values possible and the number of values above the threshold, followed by four pairs of numbers summarizing the data. The first number in each pair is the mean logarithm for the census division, and the second is the measure of uncertainty. The order of the pairs of numbers is time-integrated air concentration of a nondepositing, nondecaying gas; time-integrated air concentration of iodine-131, average iodine-131 deposition rate; and month-end iodine-131 surface contamination. Figure 3.2 shows PHASEII_FEB.OUT as an example of these files.
<table>
<thead>
<tr>
<th>Division (February)</th>
<th>1.43</th>
<th>1.44</th>
<th>1.56</th>
<th>1.45</th>
<th>1.42</th>
<th>1.82</th>
<th>1.47</th>
<th>1.41</th>
<th>1.15</th>
<th>3.18</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.43</td>
<td>1.31</td>
<td>1.34</td>
<td>1.23</td>
<td>1.28</td>
<td>1.32</td>
<td>1.25</td>
<td>1.33</td>
<td>1.28</td>
<td>1.22</td>
<td>1.38</td>
</tr>
<tr>
<td>1.31</td>
<td>1.36</td>
<td>1.37</td>
<td>1.29</td>
<td>1.30</td>
<td>1.33</td>
<td>1.26</td>
<td>1.34</td>
<td>1.30</td>
<td>1.24</td>
<td>1.40</td>
</tr>
<tr>
<td>1.34</td>
<td>1.38</td>
<td>1.39</td>
<td>1.30</td>
<td>1.31</td>
<td>1.33</td>
<td>1.26</td>
<td>1.35</td>
<td>1.31</td>
<td>1.25</td>
<td>1.41</td>
</tr>
<tr>
<td>1.23</td>
<td>1.27</td>
<td>1.28</td>
<td>1.29</td>
<td>1.30</td>
<td>1.32</td>
<td>1.25</td>
<td>1.34</td>
<td>1.30</td>
<td>1.24</td>
<td>1.39</td>
</tr>
<tr>
<td>1.28</td>
<td>1.29</td>
<td>1.30</td>
<td>1.31</td>
<td>1.32</td>
<td>1.33</td>
<td>1.26</td>
<td>1.35</td>
<td>1.31</td>
<td>1.25</td>
<td>1.40</td>
</tr>
<tr>
<td>1.45</td>
<td>1.46</td>
<td>1.47</td>
<td>1.48</td>
<td>1.49</td>
<td>1.50</td>
<td>1.43</td>
<td>1.52</td>
<td>1.48</td>
<td>1.42</td>
<td>1.54</td>
</tr>
<tr>
<td>1.42</td>
<td>1.43</td>
<td>1.44</td>
<td>1.45</td>
<td>1.46</td>
<td>1.47</td>
<td>1.40</td>
<td>1.49</td>
<td>1.45</td>
<td>1.39</td>
<td>1.51</td>
</tr>
<tr>
<td>1.82</td>
<td>1.83</td>
<td>1.84</td>
<td>1.85</td>
<td>1.86</td>
<td>1.87</td>
<td>1.79</td>
<td>1.88</td>
<td>1.84</td>
<td>1.78</td>
<td>1.90</td>
</tr>
<tr>
<td>1.47</td>
<td>1.48</td>
<td>1.49</td>
<td>1.50</td>
<td>1.51</td>
<td>1.52</td>
<td>1.44</td>
<td>1.53</td>
<td>1.49</td>
<td>1.43</td>
<td>1.55</td>
</tr>
<tr>
<td>1.41</td>
<td>1.42</td>
<td>1.43</td>
<td>1.44</td>
<td>1.45</td>
<td>1.46</td>
<td>1.39</td>
<td>1.48</td>
<td>1.44</td>
<td>1.38</td>
<td>1.50</td>
</tr>
</tbody>
</table>

**FIGURE 3.2.** An Example of an Output File Data for Census Division (February)
| Y10  | 48  | 48  | -0.21728 | 0.48758 | -0.459096 | 0.57473 | -17.94871 | 0.57083 | -11.86668 | 0.59127 |
| Y10  | 38  | 38  | -0.00435 | 0.38490 | -0.187183 | 0.53296 | -17.56358 | 0.53328 | -11.58158 | 0.53679 |
| Y10  | 28  | 28  | -3.16927 | 0.29949 | -3.384763 | 0.55483 | -17.75142 | 0.55822 | -11.75013 | 0.56819 |
| Y13  | 28  | 7   | -11.80064 | 1.89192 | -14.33702 | 1.31872 | -22.84683 | 0.77579 | -18.49463 | 2.07185 |
| Y16  | 58  | 68  | -0.995498 | 1.21823 | -0.883825 | 1.19845 | -18.25824 | 1.16766 | -12.37794 | 1.31918 |
| Y19  | 60  | 70  | -0.95608 | 1.41500 | -10.78753 | 1.58772 | -19.85641 | 1.37114 | -13.49862 | 1.78968 |
| Y20  | 35  | 35  | -0.88210 | 0.71464 | -10.35600 | 0.81790 | -18.82473 | 0.61846 | -12.77089 | 0.92926 |
| Y21  | 50  | 55  | -0.88684 | 1.17733 | -11.11190 | 1.52414 | -19.42184 | 1.44024 | -13.74844 | 1.71386 |
| Y22  | 50  | 55  | -7.99953 | 0.46872 | -9.14413 | 0.53713 | -17.55865 | 0.53865 | -11.61296 | 0.54905 |
| Y23  | 45  | 45  | -8.88755 | 0.39187 | -9.27936 | 0.53099 | -17.85688 | 0.53959 | -11.78499 | 0.54585 |
| Y24  | 30  | 30  | -8.24233 | 0.84293 | -9.28177 | 0.87790 | -17.84439 | 0.87756 | -11.81362 | 0.73325 |
| Y25  | 50  | 50  | -7.92063 | 0.50956 | -8.92319 | 0.56353 | -17.59991 | 0.56330 | -11.43568 | 0.56864 |
| Y26  | 45  | 45  | -8.88275 | 0.39181 | -9.29477 | 0.53834 | -17.87019 | 0.53837 | -11.73622 | 0.54208 |
| Y27  | 45  | 45  | -8.88431 | 0.52492 | -9.80273 | 0.56944 | -17.46946 | 0.56576 | -11.50255 | 0.57417 |
| Y28  | 90  | 95  | -8.52777 | 0.84465 | -9.65365 | 0.89036 | -18.84387 | 0.58110 | -12.19206 | 0.76980 |
| Y29  | 60  | 60  | -8.87853 | 0.45879 | -9.22996 | 0.56225 | -17.01567 | 0.56170 | -11.73523 | 0.56866 |
| Y30  | 30  | 30  | -8.49631 | 0.87590 | -9.57597 | 0.64253 | -17.99051 | 0.58232 | -12.15430 | 0.76108 |
| Y31  | 50  | 50  | -8.23468 | 0.73715 | -9.50684 | 0.56096 | -17.82207 | 0.55690 | -11.93586 | 0.56206 |
| Y32  | 40  | 40  | -8.19297 | 0.59990 | -9.18407 | 0.53420 | -17.57130 | 0.53451 | -11.08911 | 0.54800 |
| Y33  | 40  | 40  | -8.25491 | 0.85433 | -9.28748 | 0.80644 | -17.65142 | 0.56664 | -11.65191 | 0.71350 |
| Y34  | 45  | 45  | -8.23537 | 0.39703 | -9.56334 | 0.54120 | -17.08267 | 0.56192 | -11.72333 | 0.57331 |
| Y35  | 35  | 35  | -8.28977 | 0.44994 | -9.47973 | 0.57650 | -17.09645 | 0.57845 | -11.30078 | 0.59011 |

**FIGURE 3.2. (contd)**
4.0 THE MESOILT2 COMPUTER CODE

Previous sections of this report have described the technical basis for MESOILT2 and provided a user's guide to the code. This section describes the program elements of code. Appendix A contains a listing of the entire code. In the discussion that follows, the program elements are discussed in groups. The first of these discussions covers the main program. It is followed by a discussion of program elements that are used primarily for initialization of the model. Program elements used to process the wind data and prepare transport component fields are discussed next. Then the transport and diffusion program elements are discussed. The last three groups of program elements are those elements dealing with model input and output, time-keeping elements, and house-keeping elements. Appendix C contains code modifications made following the Phase I calculations.

4.1 THE MAIN PROGRAM

The main program of the MESOILT2 code is MESOILT2. Its primary purpose is to control model execution. This is done by calling model subroutines at the appropriate times. MESOILT2 calls 24 of the code's 31 subroutines directly. It also performs several housekeeping functions. Among these functions are computing factors to correct for decay of iodine-131, maintaining the simulation clock, and turning releases on and off at the appropriate times.

The general organization of the main program is shown in the flowchart in Figure 4.1. MESOILT2's major functions are represented by boxes in the flowchart. Following model initialization, MESOILT2 operates in a loop until the end of the simulation is reached. That loop involves meteorological characterization, release characterization, transport and diffusion calculations, and housekeeping. When the end of the simulation is reached, the model results are written to files. The functions are described in the following sections, and the individual subroutines are discussed later.
4.1.1 Model Initialization

As soon as code execution begins, the main program obtains the date and time from the computer system clock. It also obtains a reference time for use in computing elapsed central processor unit time for the simulation. The date
and time become part of all MESOILT2 output file names. Thus, the file names identify both the type of output and the specific MESOILT2 simulation that produced the output.

Immediately after obtaining the date and time, MESOILT2 opens a file named MESOUT.DAT. This is the MESOILT2 log file. At the end of program execution it will contain a complete record of the model run. It includes a listing of the files read and the data used for the simulation. Once opened, the log remains open throughout the simulation. It is closed at the end of the program. The logical unit used to designate the log is defined in a data statement in the UNITS include block and is passed to other program elements via the UNITS common block.

After the log is opened, MESOILT2 continues the initialization by calling six subroutines (GRIDINLT, TOPFIL, INITLT, RELEASLT, READQ, and PUFFIN) in addition to the time-keeping subroutines. These subroutines open various files and read in the data used to configure the model and control its execution. The files used for data input are opened when needed and closed when the input is complete.

The model identification, option selections, and data are passed among program elements via common blocks.

The time of the beginning of the simulation is entered in the common month, day, year, and hour formats. Subroutine DTHOUR converts the date and time to the time elapsed in hours since midnight, local time on January 1, 1982. All times used by MESOILT2 are referenced to this time, which is stored in the variable TOTHR_RUN. As the simulation progresses, TOTHR_RUN and SHR, which is time since the beginning of the simulation, are advanced in 1-hour increments.

Subroutine GRIDINLT establishes the size of the MESOILT2 computational domain in definition of the variable DELXY. For Phase I of the HEDR Project, DELXY = 16,000 m. GRIDINLT reads in the meteorological station data and initiates the process of determining those stations closest to each node in the model domain. The interrelationships between the stations and the nodes are determined by subroutines REARNG and STRAY.
Following the call to GRIDINLT, subroutine TOPFIL is called to obtain the terrain elevations data, if any.

Subroutine INITLT is called to perform the initialization of the variables used in diffusion computations. It is where the frequency of puff releases, the deposition velocity, and the washout coefficients are set and the output arrays are initialized. In addition, INITLT calls subroutine METOPN to determine the number of the meteorological record to be used at the beginning of the simulation and the time of the observation.

Subroutine RELEASLT establishes coordinates of the release point and sets the maximum number of puffs that can be active at any one time. The call to RELEASLT is followed by a call to subroutine READQ to determine the starting time for the release. READQ is called again later in MESOILT2 to determine the release rate.

The final event in the model initialization is to read in information about residual puffs that may be in the domain as a result of a previous release. This is accomplished by a call to subroutine PUFFIN.

4.1.2 Dispersion Calculations

The end of MESOILT2 initialization and the beginning of the simulation are marked in the code by the 150 CONTINUE statement. MESOILT2 returns to the 150 CONTINUE statement at the end of each hour. Immediately following this statement, MESOILT2 increments the simulation time.

The program then enters the computational loop that includes meteorological and release characterization and puff releases. This loop, which begins with the statement DO 300 IADV = 1,NPH, increments in 15-minute intervals. When IADV is one, MESOILT2 checks meteorological and release characterization to determine if they need revision. It checks for a puff release every 15 minutes.

In the meteorological characterization stage, MESOILT2 checks to see if a meteorological data record should be read. If this comparison indicates that new data should be read, MESOILT2 calls subroutine SHIFT to move meteorological data from variables in which future data are stored into the variables for current data. It then calls subroutine DATRDLT to obtain the next set of
observations. Subroutine WINDLT is called after meteorological data have been read. WINDLT converts the reported wind directions and speeds to transport components and interpolates the data to the nodes of the wind grid. The wind field is adjusted to account for terrain effects by subroutine TERRALT, if the necessary information has been provided.

The release characterization process is next. If TOTHR_RUN is equal to the time that a change in release occurs (TOTHR_NEXTQ), subroutine READQ is called to obtain the new release rate and the time of the next change. When a new release rate is read, it is converted to mass per puff for use later in the program. A puff is released every 15 minutes when the mass per puff is greater than zero.

Transport, diffusion, deposition, and depletion calculations are made in the DO 300 loop. These calculations are made separately for each puff within the loop beginning DO 250 M = 1,TPUFFS. Excluding time keeping, this section of the code is only 10 lines long. However, it contains the essence of the model. The computations that are made by the subroutines called in this section account for about 90% of the total execution time of the code. The two major subroutines called in this section of code (PUFFMLT and DIFDEPLT) are discussed at length in later sections.

Starting with the oldest puff, MESOILT2 determines if each puff is active (MF(M) .NE. 0). If it is not active, MESOILT2 moves to consideration of the next puff. Otherwise, subroutine PUFFMLT is called to determine movement of the puff during the current period. If topographic data are available, subroutine DTOPO is called to estimate the surface elevation beneath the center of the puff at the beginning and end of the advection period. Subroutine DIFDEPLT moves the puffs, controls and contributes to the diffusion calculations, and makes the deposition calculations. After the transport, diffusion, deposition, and depletion calculations are completed, MESOILT2 considers the next puff or, if there are no more puffs, moves to the next stage of the procedure.

After all calculations have been made for a 15-minute period, MESOILT2 determines if the extended output for model verification has been requested in
the command file. If it has, the program calls subroutine TESTM to write information related to puff position, movement, and mass to the MESOILT2 log.

At this point the program has reached the 300 CONTINUE statement. If the end of an hour has been reached, MESOILT2 performs a number of functions that loosely fit into the category of housekeeping in preparation for the next hour. These functions include decreasing the mass in the puffs to account for decay and calling subroutine CLEANLT if the total number of puffs being tracked exceeds eight (2*NPH). CLEANLT deletes any puffs that have been marked inactive. It also combines puffs that are close together. The reduction in the number of puffs by CLEANLT reduces the likelihood that the total number of puffs will exceed the maximum. It also reduces model execution time. When these functions are complete, MESOILT2 is ready to return to the 150 CONTINUE statement or enter the section of the code that records the final results of the simulation.

4.1.3 Model Output

The output section at the end of MESOILT2 is short. It consists of calls to subroutines PRINTELT, CLEANLT, and PUFFOUT. Subroutine PRINTELT produces the primary output files. When PRINTELT is done, CLEANLT makes one last check of the puff data, then MESOILT2 calls PUFFOUT to create a residual puff data file for use in subsequent model runs. After this output is complete, MESOILT2 writes *** SIMULATION TERMINATED *** on the log and closes the log file.

4.2 INITIALIZATION PROGRAM ELEMENTS

Seven subroutines are used for model initialization. These subroutines are GRIDINLT, REARNG, STRAY, ASCND, TOPFIL, INITLT, and RELEASLT. These subroutines are written to be run in an interactive mode in which users provide model input in response to prompts. In the current version of MESOILT2, the command file takes the place of the user. An eighth subroutine, TERRALT, is also used in initialization, but its primary use is in adjustment of wind transport components.
4.2.1 Subroutine GRIDINLT

Subroutine GRIDINLT establishes the dimensions of grids to be used by MESOILT2. The domain for the wind field is a square with 16 equally spaced nodes on a side. The spacing between nodes (in meters) is set by the variable DELXY. The variable DELXY is used in converting distances from wind grid units to meters. The diffusion and deposition computations are made on a grid that has double the node density (31 by 31).

GRIDINLT has a second function. It initializes the meteorological model. The first step in this initialization is to determine the locations of the meteorological stations. When the meteorological stations have been defined, the subroutine deletes any stations that are marked as inactive. It then calls subroutine STRAY, which builds a look-up table for each node that gives the distance from the node to each meteorological station. STRAY, in turn, calls subroutine ASCND, which arranges the entries in the look-up table in order of increasing distance.

When these computations are complete, GRIDINLT writes the grid spacing and meteorological station locations to the MESOILT2 log.

4.2.2 Subroutine REARNG

Subroutine REARNG is used to establish the number and locations of stations that will provide meteorological data for a simulation. It is called if the status variable for any of the meteorological stations indicates that a station is to be considered inactive for the simulation. The subroutine starts at the beginning of meteorological station list with a station counter set to zero. If the first station is active, the station counter is incremented and the station information is copied onto itself. If the station is inactive, the subroutine checks each station until the first active station is found. When it is found, the station counter is incremented, and the station information is copied to the first station position in the data arrays. Thus the data on the original first station are deleted. This process is repeated, until the data on all inactive stations are deleted and the remaining station information occupies contiguous locations in the arrays.
4.2.3 Subroutine STRAY

Subroutine STRAY sets up two three-dimensional arrays that relate the meteorological stations to the nodes on the wind grid. One of these arrays contains the distance from each node to each meteorological station, and the other array contains identification numbers assigned to the meteorological stations. The distances in the first array correspond to the station identification numbers in the second.

The data for each node are initially entered into two vectors. STRAY calls subroutine ASCND to arrange the station data in the two vectors in order of increasing distance. After the order of the data has been established, the data are copied into the final three-dimensional distance and station number and arrays.

4.2.4 Subroutine ASCND

Subroutine ASCND is a bubble-sort routine. It arranges the distances in the distance vector in order of increasing magnitude. The order of the station numbers is adjusted as the order of the distances is adjusted to maintain the correspondence between station numbers and distances.

4.2.5 Subroutine TOPFIL

Subroutine TOPFIL is used to read the topography data file. If the data file name provided in the command file is blank, the topography is assumed to be flat. A message to this effect is written to the MESOILT2 log, and a normal exit is made from TOPFIL.

4.2.6 Subroutine INITLT

Subroutine INITLT is the primary initialization routine for diffusion and deposition calculations. It sets the frequency with which puffs are released (NPH), the deposition velocity (DV), the washout coefficients (WC), the minimum concentration of interest (CHIMIN), and the coefficients used for vertical interpolation of transport wind components. INITLT also initializes puff and release characteristics and arrays in which the primary results are accumulated.

After the initialization of the diffusion and deposition variables is complete, it triggers the meteorological initialization. Meteorological
initialization includes, in order, determining the data frequency, calling subroutine METOPN to find the initial meteorological data record for the simulation, calling subroutine DATRDLT to read the initial meteorological data, reading the wind speed conversion data file, calling subroutine DATWR to copy the initial meteorological data record to the MESOILT2 log, calling subroutine WINDLT to generate the initial gridded transport wind fields, and calling subroutine TERRALT, if appropriate, to adjust the wind transport components for effects of terrain.

When the meteorological data initialization is complete, INITLT records the run title, date, and time on the log. If the additional output for use in code verification has been requested, a note to that effect is written. Finally INITLT records the time that the simulation is to start and the duration of the simulation on the MESOILT2 log.

4.2.7 Subroutine RELEASLT

Subroutine RELEASLT defines the position of the sources. In general, MESOILT2 has the ability to treat simultaneous releases from four sources. However, the present version of the code is limited to releases from a single source. This limitation is imposed in subroutine RELEASLT by setting the variable NSOURC = 1. The release may be from any location within the model domain and from any release height from the ground to 300 meters. The position of the source is recorded on the MESOILT2 log.

The release schedule is established using the time information in QFILE.DAT. Source terms may be changed hourly, if appropriate.

4.3 WIND PROCESSING PROGRAM ELEMENTS

Three subroutines have processing wind data as their primary purpose. These subroutines are WINDLT, TERRALT, and DIRSPD. WINDLT is called after a meteorological data record is read to convert the reported wind data to transport winds and prepare initial fields of transport components, and TERRALT adjusts these fields to account for the expected effects of terrain. Subroutine DIRSPD converts transport components to a conventional wind direction and speed.
4.3.1 Subroutine WINDLT

Subroutine WINDLT performs several functions. It decodes wind data from the meteorological stations, converts wind directions into transport directions and speeds to meters per second, and generates transport vector component fields from the transport direction and speed.

Wind directions are recorded as the direction from which the wind blows. As a result, material transported by the wind moves in the direction opposite to the wind direction. WINDLT converts wind directions to transport directions, which are expressed in radians. The variable ANG is the transport direction. When the transport direction has been determined, it is used along with the speed to obtain transport vector components, u and v. A positive u indicates transport to the east, and a positive v indicates transport to the north.

Transport components at nodes on the wind grid are computed by weighted interpolation. The interpolation weights are inversely proportional to the square of the distance between the meteorological station and the node. Only the data from the closest meteorological stations are used for the interpolation. The maximum number of locations considered in the interpolation is five. If data are available from fewer than three locations, all data are used in the interpolation regardless of distance. If meteorological data are available from three or more locations, the data from the closest two stations are used regardless of distance, but the data from the next three stations are used only if the stations are within 80 kilometers of the node.

4.3.2 Subroutine TERRALT

Subroutine TERRALT has two functions. It reads the data used to modify wind fields for anticipated terrain effects, and it performs the modifications.

If the option to modify transport wind fields is selected in the command file, TERRALT checks for the file TERRAN.DAT the first time that the subroutine is called. If the file does not exist, TERRALT records the condition in the MESOILT2 log, turns off the flag that enables wind field modifications, and returns to the main program. If the file name is present and accessible, TERRALT reads the modification data. If TERRAN.DAT is read successfully,
TERRALT writes a message to the MESOILT2 log indicating that wind fields will be adjusted for terrain effects. The message will also give the number of locations at which the adjustments will occur.

TERRALT modifies wind transport components each time it is called. These modifications are made only at nodes specified in TERRAN.DAT. The order of operations in the modification process is 1) determine wind direction at the node, 2) convert it to a transport direction, 3) project the transport vector on axes perpendicular and parallel to the slope gradient vector, 4) reduce the upslope component of transport vector, 5) increase the other component to conserve the transport speed, 6) compute the new transport direction in the original coordinate system, and 7) compute the u and v components of the new transport vector.

4.3.3 Subroutine DIRSPD

Subroutine DIRSPD transforms a wind transport vector from u and v components in Cartesian coordinates to a true wind direction and speed. If both of the transport vector components are zero, the wind speed and direction are set to zero. Otherwise, the signs of the u and v components are used to determine the proper quadrant for the wind direction, and the direction within the quadrant is computed using the arctangent function. The wind speed is computed as the vector sum of the u and v component speeds.

4.4 TRANSPORT AND DIFFUSION PROGRAM ELEMENTS

Five subroutines are included in the group of program elements that are primarily related to transport and diffusion. These elements include PUFFR and PUFFMLT, which generate and compute the movement of puffs; DTOPO, which computes the terrain elevation beneath the center of puffs as they move; DIFDEPLT, which computes the time-integrated air concentrations, deposition rates, month-end surface contamination, and puff depletion; and SIGMA, which computes the diffusion coefficients.

4.4.1 Subroutine PUFFR

Subroutine PUFFR assigns initial characteristics to each puff. When PUFFR is called, the first actions taken are to increment the total number of puffs, to increment the number of puffs released from a specific source, and
to set the flag that indicates the puff is active to .TRUE. The puff location is then initialized. It should be noted that the capability to treat sources that have different horizontal positions is retained in PUFFR even though RELEASLT limits the number of release locations to one. The puff is also assigned a source identification.

Each puff is assigned two masses. The first, QP, is its initial mass. The second mass is a depleted mass, which changes as a result of deposition. Finally, each puff is assigned initial diffusion coefficients. They are set to give concentrations at the center of a puff at the time of release that are equal to the concentration in a stack with a 20,000-cfm flow. For Phase I of the HEDR Project, the initial diffusion coefficients are $\sigma_z = 3.1$ meters, and $\sigma_y = 7.3$ meters. Other values may be used, but the values must be greater than zero.

4.4.2 Subroutine PUFFMLT

Subroutine PUFFMLT is used to determine the total puff movement during an 15-minute period. The movement is computed in wind-grid units. If a puff release height is at or above the top of the mixing layer, the puff movement is computed in a single step from the Cartesian components of the upper-level transport wind. If the puff is below the top of the mixing layer, a two-step procedure is followed.

The two-step procedure involves estimating the puff movement based on the winds at the puff's initial location, using that movement to determine an approximate ending location, using the winds at that location to get a second estimate of the movement, and averaging the two estimates. If the puff is above 10 meters, the final puff movement is determined by interpolation in the vertical.

The method of estimating the winds at the puff center depends on the location of the puff. If the puff is within the computational domain (Cartesian grid), the wind is estimated from the four surrounding nodes using bilinear interpolation. If the puff center is on a boundary of the computational domain, the wind is determined by linear interpolation between two adjacent nodes. Finally, if the puff is outside of the computational domain, the wind at the closest node is used.
When the puff no longer affects the nodes in the model domain, the puff is marked inactive by setting MF( ) to .FALSE. Once marked inactive, puffs are not permitted to return to the domain.

4.4.3 Subroutine DTOPO

Subroutine DTOPO is used to determine the terrain elevation under the center of a puff at the beginning and end of an advection step. The arguments passed to DTOPO in the call list are the puff position at the beginning of the advection period (XP,YP), in wind grid coordinates, and the distance to be moved during the advection period (DXS,DYS), also in wind grid coordinates. From this information, DTOPO computes the beginning and ending positions in exposure grid coordinates and the terrain elevation at the beginning and ending positions.

Bilinear interpolation is used to determine elevations for puff positions that are within the exposure computational domain. Elevations for puff positions outside of the computational domain are determined by linear interpolation if the position is directly north, south, east, or west of the domain. If the position is off one of the corners, the elevation at the puff center is assumed to be the same as the elevation at the corner node.

The terrain elevation under the puff center at the beginning of the advection step is stored in the variable TPMI, and the elevation at the end of the advection step is stored in TPMF. These elevations are passed via the TOPOGR common block to DIFDEPLT, where they are used in the diffusion calculations.

4.4.4 Subroutine DIFDEPLT

Subroutine DIFDEPLT is the primary program element in computation of diffusion, deposition, and depletion of the material in puffs. It performs the transport computations for the puffs within advection periods.

DIFDEPLT is called once for each puff in each advection period. When DIFDEPLT is entered, the first function that is performed is the selection of the number of sampling intervals needed in the approximate integration of the concentrations and deposition at grid nodes. This determination is made on the basis of the ratio between puff movement during a sampling interval and
the horizontal diffusion coefficient. The maximum number of intervals is established in the command file. If the puff is large, only one sampling interval is used in a 15-minute period. After the number of sampling intervals and the sampling interval duration have been selected, DIFDEPLT computes the distance traveled in each interval.

Having completed these preliminary steps, DIFDEPLT enters a computational loop that processes the puff for each sampling interval. DIFDEPLT determines the diffusion coefficients for diffusion calculations by calling subroutine SIGMA. Four diffusion coefficient parameterizations were included in earlier programs in the MESOI/MESORAD Suite. MESOILT2 includes only one, the U.S. Nuclear Regulatory Commission parameterization.

Once the diffusion computations have been made, DIFDEPLT determines the location of the puff in wind grid coordinates and selects those computations that are appropriate for the location. If the puff is completely out of the model domain, the logical flag that indicates that the puff is active is set to .FALSE., and the program returns to MESOILT2.

While the puff is contributing to nodes within the model domain, DIFDEPLT checks to determine if the concentration at ground level is below a threshold value, CHINDT. If it is, DIFDEPLT jumps to the section that computes wet deposition. If the concentration at ground level is greater than CHIMDT, DIFDEPLT continues with the accumulation of the time-integrated concentrations and deposition computations. The specific area in which the exposures, dry deposition, and external doses are accumulated is determined by the position of the center of the puff and the puff radius at ground level. This area is defined in terms of the nodes on the Cartesian grid.

The accumulation of time-integrated air concentrations, average deposition rates, and month-end surface contamination is made in the double DO loops DO 310 and DO 300. The range of indices in these loops covers the accumulation from south to north and from west to east, respectively. Within these loops, the order of computations is 1) time-integrated concentration of a undepleted, nondepositing gas, 2) time-integrated air concentration of
iodine-131, 3) average deposition rate of iodine-131, and 4) month-end surface contamination from iodine-131. This computational cycle is completed for each node within the affected area.

At this point, the computations are complete except for wet deposition. If there is no precipitation, the wet deposition computations are skipped. If there is precipitation, the initial set of computations calculates the amount of material deposited under the center of the puff as a result of wet deposition and defines the area affected by the wet deposition. The accumulation of material deposited by precipitation is treated in the same manner as is the accumulation resulting from dry deposition. This occurs in the DO 420 and DO 410 loops.

When all diffusion and deposition computations are complete for the sampling interval, the mass in the puff is depleted to account for deposition. Depletion from dry deposition is calculated first, then depletion from wet deposition. After depleting the puff, the pass through the sampling interval loop is complete. If there are more sampling intervals in the advection period, the process is repeated for the next interval.

4.4.5 Subroutine SIGMA

The diffusion coefficient parameterization used in MESOILT2 is the U.S. Nuclear Regulatory Commission parameterization (Bander 1982). The diffusion coefficients are computed as functions of atmospheric stability and distance traveled. However, if the computations were made using actual distances, changing atmospheric stability could result in a decrease in the diffusion coefficients with increasing distance. This decrease would be physically unreal. Therefore, the diffusion coefficients are computed using virtual distances.

Subroutine SIGMA computes the horizontal diffusion coefficient first. The horizontal diffusion coefficient from the previous sampling interval and the current atmospheric stability are used to compute the distance to a virtual point source. The distance traveled in the sampling interval is added to the virtual distance to get a new virtual distance, and the new virtual distance is used to compute the new horizontal diffusion coefficient.
After the horizontal diffusion coefficient is computed, the subroutine checks the magnitude of the old vertical diffusion coefficient. If it is greater than or equal to 80% of the mixing-layer thickness, the vertical diffusion coefficient is not changed. If it is less than 80% of the mixing-layer thickness, a new vertical diffusion coefficient is computed following the same procedure used to compute horizontal diffusion coefficients. The new vertical coefficient is then compared with 80% of the mixing-layer thickness. If it exceeds that value, the diffusion coefficient is set equal to the value.

4.5 INPUT AND OUTPUT PROGRAM ELEMENTS

Seven program elements are used for data input and output. These elements are subroutines METOPN, DATRDLT, DATWR, READQ, TESTM, PRINTELT, and PUFFOUT. Subroutines DATRDLT and READQ provide the meteorological data and source-term data input, respectively. Subroutine PRINTELT provides the primary model output.

4.5.1 Subroutine METOPN

Subroutine METOPN is used to open the meteorological data file and set the meteorological record index to the proper observations for the start of the simulation. If the character variable METFIL does not contain the name of a meteorological data file, if there is an error in opening the file, or if an error is encountered in reading the date/time group in the first record, METOPN will write a message to the MESOILT2 log that identifies the problem. It will then abort execution of the program.

After the date/time group in the first record has been read successfully, an error in reading a subsequent date/time group will only result in writing an error message to the log. The simulation will continue. Data will be used as appropriate until the record in which the error occurred is encountered. The remainder of the simulation will use the data in the last good record.

4.5.2 Subroutine DATRDLT

Subroutine DATRDLT is used to read formatted records from the direct access meteorological data file. METOPN reads date/time groups until it identifies the record containing the meteorological data to be used at the
beginning of a simulation. The variable MINDEX contains the number of that record. When DATRDLT is called, it reads the complete record identified in MINDEX.

If an error occurs in reading the data, the record number and error condition are written to the MESDILT2 log. The simulation will continue using the last meteorological data that were read without error. If the data record is read successfully, the data are copied from the temporary variables to the variables used in the simulation, and the data are written to the log.

DATRDLT then attempts to read the date/time group of the next record. If the attempt is successful, the date and time are converted to a time that is used to trigger the next meteorological data read, and MINDEX is incremented by 1. If an error occurs in reading the date/time group, the error is noted in the log, and an end-of-file flag is set for the meteorological data. In either case, the simulation will continue with the last set of meteorological observations that were read successfully.

4.5.3 Subroutine DATWR

Subroutine DATWR writes meteorological data records to the MESOILT2 log. It is called from subroutine INITLT twice, after subroutine DATRDLT reads the first and second meteorological records to be used in the simulation.

4.5.4 Subroutine READQ

Subroutine READQ reads source-term data from a formatted, direct access file. MESOILT2 calls READQ directly. Each time it is called, READQ opens the source-term file and reads two records. The first record is read to obtain the new source term, and the second is read to get the date and time of the next source-term change. READQ calls subroutine DTHOUR to convert the date and time of the next change to a time relative to the model reference. The time of the next change is stored in the variable TOTHR_NEXTQ. The record number for the data is stored in the variable QINDEX.

4.5.5 Subroutine TESTM

MESOILT2 calls subroutine TESTM at the end of each 15-minute period if the first character of the run title entered in the command file is an asterisk (*). When called, the subroutine writes the status of each puff to the
MESOILT2 log. The information written includes the puff number and its source, the puff position on the wind grid (grid units) and its release height (m), the distance moved in the x and y directions in the last 15 minutes (m), the diffusion coefficients (m) at the end of the period, and undepleted and depleted activity (or mass) in the puff at the end of the period (ci).

4.5.6 Subroutine PRINTELT

Subroutine PRINTELT creates the files that contain the primary results from MESOILT2. MESOILT2 calls PRINTELT after writing the end-of-simulation message on the log. When PRINTELT is called, it converts the relative time at the end of simulation (TOTHR_RUN) to a date and time and then enters a loop in which the results are written to files.

The output loop in PRINTELT steps through the types of computations performed in MESOILT2, starting with the time-integrated air concentration for nondepositing, nondecaying gas, then proceeding with the time-integrated air concentration for iodine-131, the average deposition rate for iodine-131, and the month-end surface contamination by iodine-131. The first action in the loop is to create names for the output files.

The names are created by concatenation of a prefix designating contents of the file, the month and year for which the simulation was made, and an extension indicating the type of file. For example, for the average iodine-131 deposition rate for March 1985, PRINTELT would create the names "I_DR_MAR85.OUT" and "I_DR_MAR85.BIN". I_DR_MAR85.OUT would be opened as a sequential access, formatted file, and I_DR_MAR85.BIN would be opened as a sequential access, unformatted file.

PRINTELT opens both files. When PRINTELT opens the output files it writes a message to the MESOILT2 log that gives the file name and indicates whether the file was opened successfully or not. It then writes the results to the formatted file. The run title, run date, run time, and headings are written to the file in this process. When the formatted file is complete, the unformatted file is written. The unformatted file contains the run title, run date, and run time, but does not contain headings. After both files are complete, they are closed, and the program returns to the beginning of the loop to process the next set of results.
4.5.7 Subroutine PUFFOUT

Subroutine PUFFOUT creates the residual puff data file at the end of each MESOILT2 simulation. It is called by the main program.

PUFFOUT starts by creating the name for the residual puff data file. This name is formed by concatenation of the prefix PUFFS_ with the month and year of the simulation and the extension .BIN. For the example above, the file name would be PUFFS_MAR85.BIN. This file is opened as an unformatted, sequential file. PUFFOUT then writes the file name and status to the MESOILT2 log. If the file was opened successfully, the run title, run date, and run time are written to the file. This information is followed by the number of puffs on the grid at the end of the simulation and the information for each puff. Finally, PUFFOUT closes the file and notes the results on the log.

4.6 TIME-KEEPING PROGRAM ELEMENTS

MESOILT2 uses six program elements for time keeping. Three of these elements, DATE, TIME, and SECOND are system routines that provide data for the program. The other three elements, DTHOUR, HOURDT, and JULIAN are used for date and time conversions.

4.6.1 Subroutines DATE, TIME, and SECOND

MESOILT2 makes use of three VAX system subroutines. When called, subroutine DATE returns the current date as a nine-character string. The form of the date is dd-mmm-yy, where dd is the day, mmm is a three-letter abbreviation for the month, and yy is the last two digits of the year. Subroutine TIME returns the current time as an eight-character string. The form of the string is hh:mm:ss, where hh is the hour (24-hour clock), mm is the minute, and ss is the second. Subroutine SECOND returns the elapsed run time for the program in hundredths of seconds. It is used to time various portions of the code during code development.

4.6.2 Subroutines DTHOUR and HOURDT

In MESOILT2, various actions, such as starting and stopping releases, are controlled by elapsed time in minutes from a predetermined reference. The current reference time is midnight beginning January 1, 1982. In contrast,
the times provided to the model and in model output are in terms of year, month, day, hour, and minute. Subroutines DTHOUR and HOURDT are used to convert from one time representation to the other. DTHOUR converts dates and times to elapsed time, while HOURDT converts elapsed times to date and time. The conversion takes leap years into account up to the year 2000. All times are assumed to be standard times. The conversion process does not account for changes to and from daylight savings time.

4.6.3 Subroutine JULIAN

MESOILT2 requests dates for the beginning and end of the simulation and release. For convenience, these dates are requested in the standard month, day, year form. Subroutine JULIAN takes this information and returns a day of the year ranging from 1 to 366.

4.7 HOUSEKEEPING PROGRAM ELEMENTS

MESOILT2's last two program elements are used for housekeeping. Subroutine CLEANLT eliminates unnecessary puffs, and subroutine SHIFT moves meteorological data from one set of variables to another.

4.7.1 Subroutine CLEANLT

Subroutine CLEANLT is used to reduce the number of puffs that are being tracked by MESOILT2. It is called at the end of each hour if the number of active puffs exceeds two times the number of puffs released each hour. Puffs are eliminated because they have been marked inactive. Puffs may also be eliminated by combining with other puffs.

Puffs are marked inactive by changing the value of MF( ) from true to false. This change is made when a puff leaves the model domain or when the concentration at the center of the puff falls below a threshold concentration.

When the separation between consecutive puffs from the same source becomes small compared with the size of the puff, the information in the two puffs is not significantly greater than the information in a single puff. If the separation between two puffs becomes smaller than the sum of their horizontal diffusion coefficients, the masses in the puffs are combined into a
single puff, the position of the combined puff is a weighted average of the positions of the individual puffs, and the larger diffusion coefficients are assigned to the new puff.

4.7.2 Subroutine SHIFT

MESOILT2 maintains two sets of meteorological observations and transport vector components in memory at all times. One of these sets contains data from the most recent observations, and the other contains data from the next set of observations. As the simulation progresses, the next set of observations becomes the current observations. At this time, subroutine SHIFT copies the data from variables storing the next observations to variables storing the most recent observations. When this is complete, the program reads the meteorological data from the next meteorological data record.
5.0 THE PHASEII COMPUTER CODE

The PHASEII computer code summarizes Mesoilt2 results based on meteorological data for 1983 through 1987 for use in subsequent stages of the dose estimation process. It is a simple code that has no subroutines. The code is listed in Appendix 8.

PHASEII has four parts. The first part of the code reads a file that assigns nodes in the Mesoilt2 model domain to HEDR census divisions. This part of the code is executed once. The remaining parts of the code read in Mesoilt2 results, compute the census division averages and an uncertainty in the averages, and write output files. These parts of the code are in a loop that is executed once for each calendar month.

PHASEII expects to find the nodes assigned to each census division in a file named CENDIST.LST. The nodes are read in one census division at a time. There may be as many as 31 nodes for any division. The data for the census division are contained in four records. After all four records have been read, PHASEII counts and stores the number of nodes. It then reads the data for the next division.

When the data for all census divisions have been read, the program begins the process of summarizing the Mesoilt2 results by division and month. It does this in a loop beginning with the I11 CONTINUE statement. When the loop is entered, the variable MONTH equals 1. As a result, PHASEII reads in the 20 Mesoilt2 output files for January (5 years x 4 products per year). Data from these files are placed in the GRIDS array. This completes the second part of the PHASEII program.

The third part of the program is the summarization of the results. In this part of the program, PHASEII checks the magnitudes of the values at each node. If a value is above a threshold, the program takes the logarithm to the base 10 of the value, subtracts three, and accumulates statistics for use in computing the mean and variance of the logarithms. If the value at a node is not above the threshold, the program assumes that it is equal to the threshold, but does not use the value in computing the mean and variance. Threshold values were introduced to avoid the mathematically undefined operation of
taking the logarithm of zero. The three is subtracted to normalize the values to a 1-Ci/mo release rate. A 1000-Ci/mo release rate was assumed in MESOILT2.

When all node values have been processed, the mean and variance of the logarithms are computed for census divisions using standard numerical procedures if there is more than one node with a value above the threshold level in the division. At this point, the variance describes the spatial and year-to-year variations in the values at nodes within the census division. It does not include an estimate of the basic model uncertainty. The array C2 contains estimates of the variance in model estimates of monthly time-integrated air concentrations, average deposition rates, and month-end surface contamination. The overall uncertainty associated with the census division estimates is characterized by a variance that is the sum of the variances in model predictions and the spatial and temporal variation of model predictions for the census division. The square root of the sum is stored as the measure of the uncertainty. If there is only one node with a value above the threshold, that value is stored as the mean, and the measure of uncertainty is not estimated. Finally, if no values are above the threshold, then zeros are stored for the both the mean and measure of uncertainty. This concludes the third part of the PHASEII program.

The last part of PHASEII opens two files and writes the results of the monthly summary to the files. The files are named PHASEII_JAN.OUT and PHASEII_JAN.BIN. Both files contain the census division identification, the number of values possible, the number of values above the threshold, and four pairs of numbers summarizing the data. The first number in each pair is the mean logarithm for the census division, and the second is the measure of uncertainty. In order, the pairs of numbers are for a time-integrated air concentration of a nondepositing, nondecaying gas; the time-integrated air concentration of iodine-131; the average iodine-131 deposition rate; and the month-end iodine-131 surface contamination. PHASEII_JAN.OUT may be printed or viewed on a computer terminal. PHASEII_JAN.BIN is for transmittal to the next stage in the dose computations.

5.2
When the PHASEII_JAN.OUT and PHASEII_JAN.BIN files have been written and closed, the fourth part of the PHASEII program is complete. The program then increments MONTH by 1 and returns to 111 CONTINUE to process the MESOILT2 results for February. This cycle continues until data for all 12 months have been processed.
6.0 REFERENCES


APPENDIX A

MESOILT2 CODE LISTING
APPENDIX A

MESOILT2 CODE LISTING

PROGRAM MESOILT2
C*******************************************************************
C MESOILT2 MESOILT2 VERSION 1.1
C Interactive Puff Diffusion Model (LONG TERM VERSION)
C
J.V. Ramsdell, C.S. Glantz, K. W. Burk
Pacific Northwest Laboratory
P.O. Box 999
Richland, Washington 99352

Created: 9/15/85
Updated: 3/06/90 Last Modifications

Description:
This is the main controlling program for the MESOILT2 model.
The programs have been written primarily in standard FORTRAN-77.
Documentation describes the non-standard usages. Code development
was done on a VAX 11/780.

Relationship to other modules:
Makes calls to: CLEANLT, DATRLT, DIFEPLT, DTHOUR, DTOPD,
GRIDINLT, INITLT, JULIAN, PRINTLT, PUFFIN,
PUFFMLT, PUFFOUT, PUFFR, READQ, RELEASLT,
SHIFT, TERRALT, TESTM, TOPFIL, WINDLT

Called from: NONE

C*******************************************************************
C INCLUDE 'CHAR.INC'
C INCLUDE 'CONST.INC'
C INCLUDE 'DATIM.INC'
C INCLUDE 'DECAY.INC'
C INCLUDE 'MATRIX.INC'
C INCLUDE 'PUFFS.INC'
C INCLUDE 'REL.INC'
C INCLUDE 'STATN.INC'
C INCLUDE 'TOPOGR.INC'
C INCLUDE 'UNITS.INC'
C INCLUDE 'WINDS.INC'

A.1
INTEGER SHR
REAL*4 Q
LOGICAL QERR

C ** USE SYSTEM CALLS TO OBTAIN THE CURRENT DATE AND TIME

CALL SECOND (TINITIAL)
CALL DATE (RDATE)
CALL TIME (RTIME)

WRITE (LUN(2),110) RTIME, RDATE
110 FORMAT (//////, 15X, 'MESOILT2 - PUFF DIFFUSION MODEL',
+ /26X, 'VERSION 1.1  , 1990', //26X, 'TIME = ', A8,
+ //26X, 'DATE = ', A9///)

WRITE(6,*)' ENTER A 1 IF THIS IS AN EXPOSURE ONLY RUN'
READ(5,*) IFLAGS
WRITE(6,*)' ENTER THE # OF MINUTES PER INTERVAL (1,3,5,OR 15)'
READ(5,*) IOPDTA

OPEN(UNIT=LUN(5),FILE='MESOUT.DAT',STATUS='NEW')
WRITE (LUN(5),110) RTIME, RDATE

WRITE( 6,* ) ' Enter the Meteorological Data File Name'
READ( 5,'(aGO)' ) METFILE

C ** DEFINE AND INITIALIZE GRID, TOPOGRAPHY, MODEL PARAMETERS,
C ** CHECKPOINTS AND RELEASE

120 CALL GRIDINLT
CALL TOPFIL (DELXY)
MINDEX = 1
EOFMET = .FALSE.
130 CALL INITLT
CALL RELEASLT (DDAY, DHR)

PRINT *, ' ENTER MO DAY YR FOR END OF simulation (010185)'
READ(5,111) IENDMO, IENDDY, IENDYR
111 FORMAT(3I2)

PRINT *, ' ENTER HOUR FOR END OF simulation (I2)'
READ(5,112) IENDHR
112 FORMAT(I2)

WRITE(6,113) IENDMO, IENDDY, IENDYR, IENDHR
113 FORMAT(1X,I2,'/','I2,'/','I2,5X,'HR: ','I2)

CALL JULIAN( IENDYR, IENDMO, IENDDY, JENDDY )
WRITE(6,* ) 'JULIAN DAY: ', JENDDY
C CHECK TO MAKE SURE THE Q (SOURCE) FILE AND THE MET FILE ARE VALID
C BY MAKING SURE THE FIRST RECORD FITS INTO START-STOP INTERVAL

OPEN(UNIT=22,FILE='QFILE.DAT',
*     STATUS='OLD',ACCESS='DIRECT', RECL=23,FORM='FORMATTED')

READ(22,'(1x,I2,I3,I2)',REC=1)QYR,QJDAY,QHR
CALL DTHOUR(QYR,QJDAY,QHR,TOTHR_Q)       !TOTAL HOUR Q
TOTHR_NEXTQ = TOTHR_Q                      ! FIRST TIME THRU, NEXT=FIRST REC
CLOSE(22)
QERR = .FALSE.

CALL DTHOUR(IYR,STDAY,STHR,TOTHR_START)    !TOTAL HOUR START
CALL DTHOUR(IENDYR,JENDDY,IENDHR,TOTHR_STOP) !TOTAL HOUR STOP

IF(TOTHR_Q .LT. TOTHR_START .OR. TOTHR_Q .GT. TOTHR_STOP)QERR=.TRUE.

IF(QERR) THEN
  WRITE(6,237)
237 FORMAT(I,3X,'ERROR IN Q SOURCE FILE, CANNOT CONTINUE',/)
  GOTO 410
ENDIF

CALL PUFFIN

TOTHRS = TOTHR_STOP - TOTHR_START
TOTSEC = 3600 * TOTHRS
DAYSLEFT = FLOAT(TOTHRS) / 24.0

NHOURS = 0
SHR = 0

145 CONTINUE

LINDEX = 3

C INITIALIZE CONSTANTS FOR IODINE-131

HLIFEI = 8.05
DECAYI = EXP(-0.69315 * (1.0 / 24.0) / HLIFEI)

QINDEX = 1            ! INITIALIZE TO FIRST Q RECORD
TOTHR_RUN = TOTHR_START ! INITIALIZE THE RUNNING MODEL TIME
QSOURC(1) = 0.0       ! INITIALIZE THE SOURCE TERM TO ZERO
TINC = 0.0

IF( INTFLG ) THEN ! SET THE MET DATA INTERVAL
  INTRVL = 1
ELSE
  INTRVL = 3
ENDIF

CALL SECOND( TENDINIT )

C BEGINNING OF HOURLY LOOP

150 CONTINUE

SHR = SHR + 1

C TRDKI ACCOUNTS FOR 1-131 DECAY TO END OF THE ACCUMULATION PERIOD
TRDKI = EXP( -0.69315 * DAYSLEFT / HLIFE1)

C CI AND C2 ARE WEIGHTS FOR LINEAR INTERPOLATION IN TIME

IF(INTFLG) THEN
  C2 = 0.0
ELSE
  C2 = TINC / 3.0
ENDIF

CI = 1.0 - C2

C MAIN COMPUTATIONAL LOOP

DO 300 IADV = 1, NPH

   IF (IADV .EQ. 1) THEN

   C CHECK TO SEE IF IT IS TIME TO READ NEXT MET RECORD

   IF(TOTHR_RUN+INTRVL .GE. TOTHR_NEXTMET .AND. .NOT.(EOFMET)) THEN
      CALL SHIFT
      CALL DATRDLT (LUINDX, SHR, IADV)
      CALL WINDLT
      IF(TERFLG) CALL TERRALT
      LLD = LLD * TERRALT
      TT = ((TEMPBB/10.) - 32.) * (5./9.)
   ENDIF

300 CONTINUE
C CHECK TO SEE IF IT IS TIME TO READ NEXT SOURCE RECORD

IF(TOTHR_RUN .GE. TOTHR_NEXTQ) THEN
   CALL READQ(Q)
ENDIF

QSORC(I) = Q / NPH
NHOURS = NHOURS + 1

LDEPTH = C1 * LD + C2 * LLD
STAB = C1 * IST1 + C2 * IST2
AIRT = C1 * T + C2 * TT
IF(TINC .LE. 1) THEN
   PRECIP = PRCP
ELSE
   PRECIP = PRCPP
ENDIF

ENDIF

DO 230 NS = 1, NSOURC
   210 IF (TPUFFS .LT. MAXPUF(NS)) THEN
        PRISE = 0.0
        IF(QSORC(I) .GT. 0.0) THEN
C ADD CALL TO PLUME RISE SUBROUTINE HERE
            CALL PUFFR(NS, PRISE, LDEPTH)
C ADD CALL TO SECTOR AVERAGE MODEL HERE
        ENDIF
        ENDIF
   ENDIF
   230 CONTINUE

DO 250 M = 1, TPUFFS
   IF (MF(M) .EQ. 0) GOTO 250
   IF (XP(M) .LE. 0.0 .OR. XP(M) .GT. 15.0) GOTO 240
   IF (YP(M) .LE. 0.0 .OR. YP(M) .GT. 15.0) GOTO 240
   CALL PUFFMLT(M)
   240 IF (TOPFLG) THEN
       CALL DTOPO (XP(M), YP(M), DXS(M), DYS(M))
   ENDIF
   CALL SECOND(TDIFDEPB)
   CALL DIFDEPLT(M, IFLAGS)

A.5
CALL SECOND( TDIFDEPE )
TDIFDEP = TDIFDEP + ( TDIFDEPE - TDIFDEPB )

250 CONTINUE

   IF( TITLE(1) .EQ. '*' .AND. TPUFFS .GT. 0 )
      CALL TESTM( IADV, SHR, DELXY )

300 CONTINUE

C DECAY PUFFS FOR PUFF RELEASE INTERVAL
C
 DO 305 M = 1,TPUFFS
    QPI(M) = QPI(M) * DECAYI
305 CONTINUE

C ** DETERMINE FLAG STATUS AND EXPOSURE AT SUPPLEMENTAL EXPOSURE SITES

 IF ( TPUFFS .GT. 2*NPH ) CALL CLEANLT
 TOTHR_RUN = TOTHR_RUN + 1 ! INCREMENT RUNNING TIME

380 IF( TOTHR_RUN .GE. TOTHR_STOP ) GOTO 410 ! END OF HOURLY LOOP
    TINC = TINC + 1.0
    IF( TINC .GT. 2.0 ) TINC = 0.0
    DAYSLEFT = DAYSLEFT - (1.0 / 24.0)
    GOTO 150

2323 WRITE(6,'(/,A60)')'** MET FILE DOES NOT INCLUDE THE STARTING DATE **'

410 WRITE (LUN(2),420)
 WRITE (LUN(5),420)
420 FORMAT (/5X, '*** SIMULATION TERMINATED ***'/)

CALL SECOND( TPRINT )
EI03 = EI03 + (TPRINT-TENDINIT)
CALL PRINTELT( SHR,IADV-1,IFLAGS )
CALL CLEANLT
CALL PUFFOUT

CALL SECOND( TENDEND )
EI01 = EI01 + (TENDEND - TINITIAL)
EI02 = EI02 + (TENDINIT - TINITIAL)

A.6
WRITE( 6,2930 ) SHR,TINITIAL,TENDEND

2930 FORMAT( 6x,' TOTAL SIMULATION HOURS: ',IS,5X,
+ ' INITIAL CPU TIME: ',F10.3,5X,' FINAL CPU TIME: ',F10.3 )

WRITE( LUN(2),2933 ) E101
WRITE( LUN(5),2933 ) E101
WRITE( LUN(2),2934 ) E102
WRITE( LUN(5),2934 ) E102
WRITE( LUN(2),2935 ) E103
WRITE( LUN(5),2935 ) E103
WRITE( LUN(5),2937 ) TDIFDEP
WRITE( LUN(2),2937 ) TDIFDEP

2933 FORMAT(15X,F10.3,' IS TOTAL ELAPSED CPU TIME')
2934 FORMAT(15X,F10.3,' IS PROGRAM PREP TOTAL ELAPSED CPU TIME')
2935 FORMAT(15X,F10.3,' IS MAIN COMPUTATION TOTAL ELAPSED CPU TIME')
2937 FORMAT(15X,F10.3,' IS CPLUME ELAPSED CPU TIME')

999 CONTINUE
CLOSE( LUN(5) )
STOP ' MESOI1T2 EXECUTION -- NORMAL TERMINATION'
END
ASCND
Order stations
Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352
Created: 6/1/83
Updated: 3/08/90
Description: For a given grid point, stations are arranged in ascending order by distance from the point.
RT -- distance (X**2 + Y**2)
NSB -- station number
ACTSTA -- total number of active stations
Relationship to other modules:
Makes calls to: NONE
Called from: STRAY

SUBROUTINE ASCND (RT, NSB, ACTSTA)
REAL RT(40), TMP
INTEGER NSB(40), NTMP
DO 110 I = 1, ACTSTA-1
   J = 0
   DO 100 K = 1, ACTSTA-1
      IF (RT(K+1) .GT. RT(K)) GOTO 100
      J = K + 1
      TMP = RT(K)
      NSBTMP = NSB(K)
      RT(K) = RT(J)
      NSB(K) = NSB(J)
      RT(J) = TMP
      NSB(J) = NSBTMP
100 CONTINUE
IF (J .EQ. 0) GOTO 120
110 CONTINUE

120 RETURN
END
SUBROUTINE CLEANLT

include 'char.inc'
INCLUDE 'CONST.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'UNITS.INC'

** MERGE LOOP; CHECK PUFFS STARTING WITH YOUNGEST

I = TPUFFS
J = TPUFFS - 1

** IF PUFF IS INACTIVE IT IS NOT CONSIDERED

100 IF (MF(I) .EQ. 0) GOTO 120

IF (SP(I) .EQ. SP(J) .AND. MF(J) .EQ. 1) THEN

** CALCULATE SEPARATION; COMBINE PUFFS IF SEPARATION IS LESS THAN THE SUM OF THE SIGMA Y'S.

SEP = DELXY * SQRT((XP(I) - XP(J)) ** 2 + (YP(I) - YP(J)) ** 2 )
IF (SEP / (SIGMAY(I) + SIGMAY(J)) .LT. 1.0) THEN

MF(J) = 0
QPI(I) = QPI(I) + QPI(J)

C ** DETERMINE WEIGHTS FOR AVERAGING POSITIONS

QT = QP(I) + QP(J)
IF( QP(I) .LE. QP(J) ) THEN
  IS = I
  IL = J
ELSE
  IS = J
  IL = I
ENDIF
WTS = QP(IS) / QT
WTL = 1.0 - WTS

C ** MERGE PUFFS

XP(I) = WTS * XP(I) + WTL * XP(J)
YP(I) = WTS * YP(I) + WTL * YP(J)
ZP(I) = WTS * ZP(I) + WTL * ZP(J)
DXS(I) = WTS * DXS(I) + WTL * DXS(J)
DYS(I) = WTS * DYS(I) + WTL * DYS(J)
QP(I) = QP(I) + QP(J)
SIGMAY(I) = AMAX1( SIGMAY(I),SIGMAY(J) )
SIGMAZ(I) = AMAX1( SIGMAZ(I),SIGMAZ(J) )

ENDIF
ENDIF

120  I = I - 1
     J = I - 1
IF (I .GT. I) GOTO 100

C ** DROP ALL THE INACTIVE PUFFS BY EDITING THEIR ATTRIBUTES OUT OF
C ** THE VARIOUS ARRAYS.

I = 1
J = 1

200  IF (MF(I) .EQ. 0) GOTO 220
MF(J) = MF(I)
QP(J) = QP(I)
SIGMAY(J) = SIGMAY(I)
SIGMAZ(J) = SIGMAZ(I)
SP(J) = SP(I)
XP(J) = XP(I)
YP(J) = YP(I)
ZP(J) = ZP(I)

A.11
DXS(J) = DXS(I)
DYS(J) = DYS(I)
QPI(J) = QPI(I)
J = J + 1

220  I = I + 1
     IF (I .LE. TPUFFS) GOTO 200
     IF (TITLE(I) .EQ. '*') WRITE( LUN(5),240 ) TPUFFS - (J-I), J-I

240  FORMAT (5X, 'CLEAN CALLED --> ', I4, ' PUFFS DROPPED',
     + 4X, I4, ' PUFFS REMAIN ON THE GRID',/)

TPUFFS = J - 1

RETURN
END
DATRDLT
Meteorological data input module

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/06/90

Description: Subroutine to read the meteorological data. The
  calling program passes the unit number on which the observation
  or forecast information is located. The meteorological data is
  in direct access form.

Relationship to other modules:

  Makes calls to: DTHOUR
  Called from: MESOILT2, INITLT

********************************************************************************
SUBROUTINE DATRDLT (LUINDX, SHR, IADV)
PARAMETER(IMAX=40) !NUMBER OF STATIONS
INTEGER SHR

INCLUDE 'DATIM.INC'
INCLUDE 'UNITS.INC'

OPEN(UNIT=22,FILE=metfile,
  STATUS='old',ACCESS='direct',RECL = 234, FORM='formatted')

READ(22,110,rec=mindex,ERR=900) DYR,ODAY,DHR,OMIN,IST2,LLD,PRCPP,
  UPWIND,TEMPAA,TEMPBB,(DATA(I),I=1,IMAX)
110 FORMAT(IX,12,13,12,12,IX,II,IX,14,1X,I3,1X,I3,40(1X,I4))

TYPE*,'DATRDL**',DYR,DDAY,DHR

TOTHR MET = TOTHR_NEXTMET
MIND EX = MIND EX + 1

READ(22,'(1X,I2,I3,I2)'),REC=MINDEX,ERR=900)NYR,NDAY,NHR
CALL DTHOUR(NYR,NDAY,NHR,TOTHR_NEXTMET)

CLOSE(22)
RETURN

900 PRINT *, 'DATRD >> EOF AT RECORD ', MINDEX, ' UNIT ', LUN(LUINOX)
PRINT *, ' SIMULATION HOUR ', SHR, ' ADVECTION STEP = ', IADV
STOP

END
C*******************************************************************
C
C
DATWR
C
MESOILT2 VERSION 1.1
C
Meteorological data output module
C
Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
C
Pacific Northwest Laboratory
C
PO Box 999
C
Richland, Washington 99352
C
Created: 6/1/83
C
Updated: 3/08/90
C
Description: Subroutine to list the basic meteorological input
data to the specified output file. Called after 'DATRD'
has been used.
C
Relationship to other modules:
C
Makes calls to: NONE
C
Called from: INITLT
C
C*******************************************************************
SUBROUTINE DATWR
INCLUDE 'DATIM.INC'
INCLUDE 'UNITS.INC'

WRITE (LUN(2),100) DYR, DDAY, DHR, DMIN
100 FORMAT (/5X, 'DATA FOR JULIAN DAY: ',I2,1X,I3,' AT ',I2,':',I2/)

WRITE (LUN(2),120) IST2, LLD, PRCPP, TEMAA, TEMBB
120 FORMAT (8X, 'STABILITY CLASS = ',I1, 3X, 'MIXING DEPTH = ',I4,
+            3X, 'PRECIP CODE = ',I1, 3X,
+            'AIR TEMPS(F) = ', I3, ' AND ', I3/)

WRITE (LUN(2),140) UPWIND, (DATA(I), I=1, 10)
140 FORMAT (8X, 'WINDS', 3X, I4, 1X, 10(1X,I4))

WRITE (LUN(2),160) (DATA(I), I=11, 20)
WRITE (LUN(2),160) (DATA(I), I=21, 30)
WRITE (LUN(2),160) (DATA(I), I=31, 40)
160 FORMAT (21X, 10(1X,I4))

RETURN
END
**Diffusion, transport and deposition module**


Battelle - Pacific Northwest Laboratories

PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/07/90

Description: Diffusion, deposition and decay computations and time integration during the advection step.

Relationship to other modules:

Makes calls to: SIGMA

 Called from: MESOILT2

---

**SUBROUTINE DIFDEPLT (M, IFLAGS)**

INCLUDE 'CHAR.INC'
INCLUDE 'CONST.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'DECAY.INC'
INCLUDE 'MATRIX.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'REL.INC'
INCLUDE 'TOPOGR.INC'

INTEGER DT, NSI, DTA(12), NSIA(12), EAST, WEST, NORTH, SOUTH

DATA DTA / 15, 5, 3, 1, 8*0 /
DATA NSIA / 1, 3, 5, 15, 8*0 /

DS = SQRT(DXS(M)**2 + DYS(M)**2)
DSMTR = DS * DELXY

**DETERMINE DT AND NSI FOR PUFF THIS ADVECTION STEP**

DO 150 I = 1, 12
DSMTRI = DSMTR / NSIA(I)
IF (DTA(I) .EQ. IPODTA .OR. (DSMTRI / SIGMAY(M)) .LT. 2) THEN
   DT = DTA(I)
   NSI = NSIA(I)
   GOTO 160
ENDIF

A.16
CONTINUE

CONTINUE

DTPM = (TPMF - TPMI) / NSI
CHIMDT = CHIMIN * DT

DXMI = DXS(M) / NSI
DYMI = DYS(M) / NSI

DO 500 JN = 1, NSI

C ** SUBROUTINE SIGMA COMPUTES SIGMA Y AND Z

CALL SIGMA (DSMTRI, STAB, LDEPTH, SIGMAZ(M), SIGMAY(M))

SIGYSQ = SIGMAY(M) * SIGMAY(M)
SIGZSQ = SIGMAZ(M) * SIGMAZ(M)
HSGSQ = (-0.5 * (DELXY / 2.0) ** 2) / SIGYSQ

C ** COMPUTE THE TERMS ASSOCIATED WITH THE VERTICAL DISTRIBUTION

GCPCHI = 0.0
UDEPTH = 0.0
IF (ZP(M) GT LDEPTH) THEN

C ** THE PUFF CENTER IS ABOVE THE MIXING LAYER

IF (SIGMAZ(M) GE 0.8 * ZP(M)) THEN

C ** UNIFORM PUFF DISTRIBUTION IN THE VERTICAL

UDEPTH = SIGMAZ(M)/0.8
PUFCHI = QP(M) * (DT*60.0) / (TWOPI * SIGYSQ * UDEPTH)

IF (PUFCHI LE CHIMDT) THEN
GO TO 200
ELSE
CHIR = ALOG(PUFCHI/CHIMDT)
RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
VEXP = 1.0
ENDIF
ELSE

C ** REFLECTED GAUSSIAN DISTRIBUTION IN THE VERTICAL

PUFCHI = 2.0*QP(M)*(DT*60.0) / ((TWOPI**1.5)*SIGYSQ*SIGMAZ(M))

IF (PUFCHI LE CHIMDT) THEN
GOTO 200
ELSE

A.17
CHIR = ALOG(PUFCHI/CHIMDT)
RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
VEXP = EXP(-0.5 * ZP(M) * ZP(M) / SIGZSQ)
ENDIF

ENDIF

ELSE

C  ** THE PUFF CENTER IS WITHIN OR AT THE TOP OF THE MIXING LAYER

IF (SIGMAZ(M) .GE. 0.8 * LDEPTH) THEN

C  ** WITH UNIFORM DISTRIBUTION IN THE VERTICAL

UDEPTH = SIGMAZ(M)/0.8
PUFCHI = QP(M) *(DT*60.0) / (TWOPI * SIGYSQ * UDEPTH)

IF (PUFCHI .LE. CHIMDT) THEN
    GOTO 200
ELSE
    CHIR = ALOG(PUFCHI/CHIMDT)
    RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
    VEXP = 1.0
ENDIF

ELSE

C  ** REFLECTED GAUSSIAN DISTRIBUTION IN THE VERTICAL

PUFCHI = 2.0*QP(M)*(DT*60.0) / ((TWOPI**1.5)*SIGYSQ*SIGMAZ(M))

IF (PUFCHI .LE. CHIMDT ) THEN
    GOTO 200
ELSE
    CHIR = ALOG(PUFCHI/CHIMDT)
    RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
    VEXP = EXP(-0.5*ZP(M)*ZP(M)/SIGZSQ)
    EXPARG = 0.5*ZP(M)*ZP(M)/SIGZSQ

    IF(EXPARG .LT. 15) THEN
        VEXP = VEXP + EXP(-0.5*(2*LDEPTH-ZP(M))**2/SIGZSQ)
        IF(ABS(EXPTERM) .LT. 15)
            VEXP = VEXP + EXP(-0.5*(2*LDEPTH+ZP(M))**2/SIGZSQ)
        ENDIF
    ENDIF

ENDIF

ENDIF

ENDIF

A.18
**COMPUTE PUFF CONCENTRATION AT THE GROUND IF IT IS NOT NEGLIGIBLE**

IF ( EXP(-CHIR) .LT. VEXP ) THEN
   GCPCHI = PUFCHI * VEXP
ELSE
   IF ((PRECIP .EQ. 0) .OR. (IFLAGS .EQ. 1)) THEN
      TPMI = TPMI + DTPM
      XP(M) = XP(M) + DXMI
      YP(M) = YP(M) + DYMI
      IF (XP(M) .LT. 0.0 .OR. XP(M) .GT. 15.0) GOTO 200
      IF (YP(M) .LT. 0.0 .OR. YP(M) .GT. 15.0) GOTO 200
      GOTO 500
   ENDIF
ENDIF

**COMPUTE THE PER INTERVAL MOVEMENT OF THE PUFF CENTER**

TPMI = TPMI + DTPM
XP(M) = XP(M) + DXMI
YP(M) = YP(M) + DYMI

IF (XP(M)+RPEW .LT. 0.0) GOTO 200
IF (XP(M)-RPEW .GT. 15.0) GOTO 200
IF (YP(M)+RPEW .LT. 0.0) GOTO 200
IF (YP(M)-RPEW .LE. 15.0) GOTO 210

200 MF(M) = 0
RETURN

**CALCULATE THE EFFECTIVE GROUND RADIUS OF THE PUFF**

210 IF (ZP(M) .LE. 10.0) THEN
   RPGW = RPEW
ELSE
   IF (GCPCHI .GT. CHIMDT) THEN
      RPGW = (SIGMAY(M) * SQRT(2. * ALOG(GCPCHI / CHIMDT)) / DELXY)
   ELSE
      GOTO 400
   ENDIF
ENDIF

RPGE = RPGW * 2.0
** PUFCX AND PUFCY ARE THE COORDINATES OF THE PUFF ON THE FINE GRID
EAST, WEST, NORTH AND SOUTH ARE EDGE COORDINATES
RPGE IS THE PUFF RADIUS AT GROUND LEVEL IN FINE GRID UNITS

PUFCX = 2 * XP(M) + 1
WEST = PUFCX - RPGE
EAST = PUFCX + RPGE

PUFCY = 2 * YP(M) + 1
SOUTH = PUFCY - RPGE
NORTH = PUFCY + RPGE

IF (WEST .LT. 1) WEST = 1
IF (EAST .GT. 31) EAST = 31
IF (SOUTH .LT. 1) SOUTH = 1
IF (NORTH .GT. 31) NORTH = 31

** CALCULATE NEW CONCENTRATIONS FOR EACH CHI GRID POINT
** INFLUENCED BY THE CURRENT PUFF

DO 300 J = SOUTH, NORTH
DO 310 I = WEST, EAST

RSQ = (((XP(M)*2.0) - (I-1)**2) +
+ (((YP(M)*2.0) - (J-1)**2)
IF (RSQ .GT. (RPGE*RPGE)) GOTO 310

IF (UDEPTH .EQ. 0.0) THEN
HEFF = ZP(M) - (TOPO(I,J) - TPMI)
VEXP = EXP(HEFF ** 2 / (-2.0 * SIGZSQ))
EXPTERM = (2.0 * LDEPTH + HEFF) ** 2 / (-2.0 * SIGZSQ)
IF(ABS(EXPTERM) .LT. 15)
+ VEXP = VEXP + EXP((2.0 * LDEPTH + HEFF) ** 2 / (-2.0 * SIGZSQ))
IF (ZP(M) .LE. LDEPTH) THEN
IF(ABS(EXPTERM) .LT. 15)
+ VEXP = VEXP + EXP((-2.0 * LDEPTH + HEFF) ** 2 / (-2.0 * SIGZSQ))
ENDIF
ENDIF
HEXP = EXP(HSGSQ*RSQ)
PCHI = PUFCHI * HEXP * VEXP
EXPCUM(I,J) = EXPCUM(I,J) + PCHI

** END OF NORMAL DIFFUSION - START CORRECTIONS FOR DECAY,
DEPOSITION AND WASHOUT
C ** COMPUTE THE DRY DEPOSITION

IF(IFLAGS .NE. 1) THEN
    DPCHII = PCHI * QPI(M)/QP(M)
C ** COMPUTE EXPOSURE FROM DEPLETED PUFFS
    TICI(I,J) = TICI(I,J) + DPCHII
C ** COMPUTE DRY DEPOSITION
    SCI(I,J) = SCI(I,J) + DVI * DPCHII * TRDKI
    DRI(I,J) = DRI(I,J) + DVI * DPCHII / TOTSEC
ENDIF
310  CONTINUE
300  CONTINUE
C ** COMPUTE THE WET DEPOSITION
400  CONTINUE
C ** IF WE HAVE WET DEPOSITION OF A SIGNIFICANT PUFF

IF ((PRECIP.GT. 0) .AND. (IFLAGS .NE. 1)) THEN
    WCDT = WC(PRECIP) * DT / 60.0
    WSHOI = WCDT * QPI(M) / (TWOPI * SIGYSQ)
    RPEE = RPEW * 2.0
    WEST = PUFCX - RPEE
    EAST = PUFCX + RPEE
    SOUTH = PUFCY - RPEE
    NORTH = PUFCY + RPEE

    IF (WEST .LT. 1) WEST = 1
    IF (EAST .GT. 31) EAST = 31
    IF (SOUTH .LT. 1) SOUTH = 1
    IF (NORTH .GT. 31) NORTH = 31

C ** COMPUTE WASHOUT FROM THE PUFF USING THE ELEVATED RADIUS

DO 420 I = WEST, EAST
    DO 410 J = SOUTH, NORTH
        RSQ = (((XP(M) * 2.0) - (I-1)) ** 2) +
              (((YP(M) * 2.0) - (J-1)) ** 2)
        IF (RSQ.GT. (RPEE*RPEE)) GOTO 410
    410  CONTINUE
420  CONTINUE
A.21
HEXP = EXP(HSGSQ * RSQ)

SCI(I,J) = SCI(I,J) + WSHOI * HEXP * TRDKI

DRI(I,J) = DRI(I,J) + WSHOI * HEXP / TOTSEC

CONTINUE

C ** RADIOACTIVE DECAY OF THE PUFF AT THE END OF EACH INTERVAL

C ** DEPLETE BY DRY DEPOSITION

1499 IF (IFLAGS .NE. 1) THEN

   IF (GCPCHI .GT. 0.0) THEN

      QPI(M) = QPI(M) * (1.0 - 2.0 * DVI *
      * (DT * 60.0) * VEXP / (SQRT(TWOP) * SIGMAZ(M)))

   ENDIF

C ** DEPLETE BY WET DEPOSITION

   IF (PRECIP .GT. 0) THEN

      QPI(M) = QPI(M) * EXP(-WCDT)

   ENDIF

   QPI(M) = AMAX1(QPI(M),0.0)

ENDIF

CONTINUE

RETURN

END
SUBROUTINE DIRSPD (U, V, DIR, SPEED)

REAL U, V, DIR, SPEED

IF (U .EQ. 0.0 .AND. V .EQ. 0.0) THEN
   DIR = 0
   SPEED = 0
   RETURN
ENDIF

IF (U .GT. 0.0) THEN
   IF (V .GT. 0.0) THEN
      DIR = 180 + ATAN(ABS(U/V)) * 57.2958
   ELSE
      DIR = 270 + ATAN(ABS(V/U)) * 57.2958
   ENDIF
   GOTO 100
ENDIF

IF (U .LT. 0) THEN
   IF (V .GE. 0.0) THEN
      DIR = 90 + ATAN(ABS(V/U)) * 57.2958
   ELSE
      DIR = ATAN(ABS(U/V)) * 57.2958
   ENDIF
GOTO 100
ELSE

   IF (V .GT. 0.0) THEN
      DIR = 180.
   ELSE
      DIR = 0.0
   ENDIF

GOTO 100
ENDIF

100  SPEED = (U*U + V*V) ** 0.5
RETURN
END
SUBROUTINE DTHOUR(yr,jday,hour,tothr)

INTEGER yr,jday,hour,tothr
INTEGER dpy(4)

DATA dpy /366,365,365,365/

tothr = 0
idays = 0

index = yr - 82

DO i = 1,index
  j = JMOD(81+i,4) + 1
  idays = idays + dpy(j)
  enddo

idays = idays + (jday-1)

FIND TOTAL HOURS

tothr = (idays * 24) + hour

RETURN
END
SUBROUTINE DTOPO (XP, YP, DXS, DYS)

INCLUDE 'TOPOGR.INC'

REAL XPFG(2), YPFG(2), THT(2), WT(4)

** CONVERT PUFF POSITIONS TO COORDINATES ON EXPOSURE FINE GRID

XPFG(1) = (XP * 2.0) + 1.0
YPFG(1) = (YP * 2.0) + 1.0

XPFG(2) = (DXS * 2.0) + XPFG(1)
YPFG(2) = (DYS * 2.0) + YPFG(1)

DO 100 IT = 1, 2

** DETERMINE SUBSCRIPTS DEFINING THE SURROUNDING GRID POINT

IX1 = INT(XPFG(IT))
IX2 = IX1 + 1

IY1 = INT(YPFG(IT))
IY2 = IY1 + 1

IF(IX1 .GE. 1 .AND. IX2 .LE. 31 .AND. IY1 .GE. 1 .AND.
+ IY2 .LE. 31) THEN
C ** PUFF IS ON GRID; COMPUTE INTERPOLATION WEIGHTS

\[
\begin{align*}
X_1 &= X_{PFG}(IT) - I\times 1 \\
X_2 &= 1.0 - X_1 \\
Y_1 &= Y_{PFG}(IT) - I\times 1 \\
Y_2 &= 1.0 - Y_1
\end{align*}
\]

IF (X_1 < 0.001 .AND. Y_1 < 0.001) THEN
  THT(IT) = TOPO(I\times 1, I\times 1)
  GO TO 100
ENDIF

\[
\begin{align*}
W_T(1) &= 1.0 / \sqrt{X_1^2 + Y_1^2} \\
W_T(2) &= 1.0 / \sqrt{X_2^2 + Y_1^2} \\
W_T(3) &= 1.0 / \sqrt{X_1^2 + Y_2^2} \\
W_T(4) &= 1.0 / \sqrt{X_2^2 + Y_2^2}
\end{align*}
\]

SUM = 0.0
DO 80 I = 1, 4
  SUM = SUM + W_T(I)
  CONTINUE

DO 90 I = 1, 4
  W_T(I) = W_T(I) / SUM
  CONTINUE

C ** COMPUTE TERRAIN HEIGHT UNDER PUFF CENTER

\[
\begin{align*}
THT(IT) &= W_T(1) \times TOPO(I\times 1, I\times 1) + \\
        & \quad W_T(2) \times TOPO(I\times 2, I\times 1) + \\
        & \quad W_T(3) \times TOPO(I\times 1, I\times 2) + \\
        & \quad W_T(4) \times TOPO(I\times 2, I\times 2)
\end{align*}
\]

ELSE

C ** PUFF IS OFF THE GRID

IF (I\times 1 < 1) THEN

C ** PUFF IS WEST OF THE GRID **

IF (I\times 1 < 1) THEN
  THT(IT) = TOPO(I, 1)
ELSE
  IF (I\times 2 < 31) THEN
    THT(IT) = TOPO(I, 31)
  ELSE
    W_T(2) = Y_{PFG}(IT) - I\times 1
    W_T(1) = 1.0 - W_T(2)
    THT(IT) = W_T(1) \times TOPO(I, I\times 1) + W_T(2) \times TOPO(I, I\times 2)
  ENDIF
ENDIF

ENDIF
ELSE
   IF (IX2 .GT. 31) THEN
   " ** PUFF IS EAST OF GRID ** 
   IF (IY1 .LT. 1) THEN
      THT(IT) = TOPO(31,1)
   ELSE
      IF (IY2 .GT. 31) THEN
         THT(IT) = TOPO(31,31)
      ELSE
         WT(2) = YPFG(IT) - IY1
         WT(1) = 1.0 - WT(2)
         THT(IT) = WT(1)*TOPO(31,IY1) + WT(2)*TOPO(31,IY2)
      ENDIF
   ENDIF
   ELSE
      WT(2) = XPFG(IT) - IX1
      WT(1) = 1.0 - WT(2)
      IF (IY1 .LT. 1) THEN
      " ** PUFF IS NORTH OF GRID **
      THT(IT) = WT(1) * TOPO(IX1,1) + WT(2) * TOPO(IX2,1)
      ELSE
      " ** PUFF IS SOUTH OF GRID **
      THT(IT) = WT(1) * TOPO(IX1,31) + WT(2) * TOPO(IX2,31)
      ENDIF
   ENDIF
ENDIF
"100 CONTINUE

TPMI = THT(1)
TPMF = THT(2)

RETURN
END
GRIDINLT
Grid initialization module

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 9/15/85
Updated: 3/06/90

Description: Initialization of the grid and set up of
meteorological stations. Reads data from a user specified
file and allows editing of the station parameters and the
addition of new stations.

Relationship to other modules:

Makes calls to: REARNG, STRAY
Called from: MESOILT2

** DEFINE THE METEOROLOGICAL STATIONS:
** NAMST = 4 CHARACTER NAME
** XDIST = STATION DISTANCE FROM GRID CENTER IN KILOMETERS ALONG X AXIS
** YDIST = STATION DISTANCE FROM GRID CENTER IN KILOMETERS ALONG Y AXIS
** STATUS = 0 IF ACTIVE; 1 IF DISABLED
** ELEV = ELEVATION IN METERS ABOVE SEA LEVEL

DELXY = 16000.0

WRITE (LUN(2),100)
WRITE (LUN(5),100)
FORMAT ('/5X, 'MESO ---> GRID INITIALIZATION',/)
WRITE (LUN(2),110) DELXY
FORMAT ('/8X, 'THE CURRENT WIND GRID IS: 16 ROWS X 16 COLUMNS', + 8X, 'DELXY = ', F10.2, ' METERS')
CONTINUE
WRITE (LUN(2),180)
FORMAT ('/5X, 'MESDILT2 ---> STATION INITIALIZATION',/)
NUMSTA = 0
PRINT *, 'ENTER FILENAME FOR READING STATION INFORMATION > '
READ (LUN(1),200) FILE1
FORMAT (A80)
OPEN (UNIT=LUN(15), FORM='FORMATTED', STATUS='OLD', FILE=FILE1, + IOSTAT = IER)
IF ( IER .NE. 0) THEN
  PRINT *, 'GRIDIN >> ERROR ', IER, ' OPENING SPECIFIED FILE'
  GOTO 190
ENDIF
READ (LUN(15),220,END=230) NAMST(NUMSTA+1), XDIST(NUMSTA+1), + YDIST(NUMSTA+1), ELEV(NUMSTA+1), STATUS(NUMSTA+1)
FORMAT (1X, A4, 2(2X,F6.2), 4X, 4X, II)
NUMSTA = NUMSTA + 1
GOTO 210
CLOSE (LUN(15))
NSTAT = 0
DO 260 I=1, NUMSTA
  XSTA(I) = (XDIST(I)*1000. / DELXY) + 7.5
  YSTA(I) = (YDIST(I)*1000. / DELXY) + 7.5
  IF (STATUS(I).EQ.1) NSTAT = NSTAT + 1
CONTINUE
WRITE (LUN(2),270) NUMSTA,NSTAT
WRITE (LUN(2),280) NUMSTA,NAMST(I),XSTA(I),YSTA(I),ELEV(I),STATUS(I)
DO 300 I = 1, NUMSTA
  WRITE (LUN(2),290) I,NAMST(I),XSTA(I),YSTA(I),ELEV(I),STATUS(I)
  FORMAT (6X,I2, 2X,A4, 3X,F8.2, 2X,F8.2, 4X,I4, 6X,II)
  IF (MOD(I,15) .EQ. 0) PAUSE 'TYPE C TO CONTINUE'
CONTINUE
610 CONTINUE
ACTSTA = NUMSTA - NSTAT
IF (NSTAT .NE. 0) CALL REARNG(NUMSTA)
CALL STRAY

WRITE (LUN(2),620)
620 FORMAT (/,, 5X, '** END GRID INITIALIZATION **',//)

C ** WRITE COMPLETED GRID INFORMATION TO OUTPUT FILE

WRITE (LUN(5),110) DELXV
WRITE (LUN(5),270) NUMSTA, NSTAT
WRITE (LUN(5),280)
DO 630 I = 1, NUMSTA
   WRITE (LUN(5),290) I, NAMST(I), XSTA(I), YSTA(I), ELEV(I), +
   STATUS(I)
630 CONTINUE
WRITE (LUN(5),620)

RETURN

999 STOP 'ERROR IN READ OR END OF FILE IN GRIDINLT - PROGRAM TERMINATED'

END
**HOURDT**

K.W. Burk, J.V. Ramsdell
Pacific Northwest Laboratory
P.O. Box 999
Richland, Washington 99352

Created: 6/01/89
Updated: 3/06/90

Description: Accepts the total number of hours that have elapsed since midnight (12:01 a.m.) January 1, 1982, then returns the corresponding month, day and year.

Relationship to other modules:

- Makes calls to: **NONE**
- Called from: **PRINTELT**

**SUBROUTINE HOURDT(TOTHR, LVR, LMO, LDAY, LHR)**

```
INTEGER TOTHR, LVR, LMO, LDAY, LHR,
       * TOTDAYS, TOTYR

DATA DPM / 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 /
DATA DPY / 365, 365, 366, 365/

TOTDAYS = TOTHR / 24

I = 1
ND = 0
LVR = 82

100 CONTINUE

IF(ND + DPY(I) .GT. TOTDAYS) THEN
   NDAYS = TOTDAYS - ND
   GOTO 200
ENDIF
ND = ND + DPY(I)
LVR = LVR + I
I = I + 1
IF(I .GT. 4) I = 1
GOTO 100
```
200 CONTINUE
IF(DPY(I) .EQ. 366) DPM(2) = 29

JDAY = 0
DO J = 1, 12
  LMO = J
  IF(JDAY + DPM(J) .GT. NDAYS) THEN
    LDAY = NDAYS - JDAY + 1
    GOTO 300
  ELSE
    JDAY = JDAY + DPM(J)
  ENDIF
ENDDO

300 CONTINUE

LHR = TOTHR - TOTDAYS * 24

RETURN
END
Primary initialization module

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 9/15/85
Updated: 3/06/90

Description: Primary initializations of the model, including:
Identification title, current date and time, and basic
simulation parameters.

Relationship to other modules:

Makes calls to: DATRDLT, DATWR, JULIAN, METOPN,
SHIFT, TERRALT, WINDLT

Called from: MESOILT2

SUBROUTINE INITLT

INCLUDE 'CHAR.INC'
INCLUDE 'CONST.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'MATRIX.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'REL.INC'
INCLUDE 'STATN.INC'
INCLUDE 'UNITS.INC'
INCLUDE 'WINDS.INC'

CHARACTER*50 WINDFILE
CHARACTER*1 SELECT

** DEFINE WASHOUT COEFFICIENTS

DATA WC/ 0.79, 2.2, 4.0, 0.36, 1.2, 2.3 /

WRITE (LUN(2),100)
WRITE (LUN(5),100)
100 FORMAT (/5X, 'MESO -> PRIMARY INITIALIZATION',/)

110 WRITE (LUN(2),120)
120 FORMAT (5X, 'ENTER RUN IDENTIFICATION TITLE OF UP TO',
+ '80 CHARACTERS')
READ (LUN(1),130,END=999,ERR=999) TITLE
130 FORMAT (80A1)

C ** SET MODEL PARAMETERS:
C ** NPH ----- NUMBER OF PUFFS PER HOUR
C ** CHIMIN -- INITIAL MINIMUM SIGNIFICANT CONCENTRATION

NPH = 4
CHIMIN = 1.0E-15 * 60.

IF ((MOD(60,NPH)) .NE. 0) GOTO 800

DO 150 I = 1, 500
   MF(I) = 0
150 CONTINUE

DO 160 I = 1, 4
   NPUFFS(I) = 0
160 CONTINUE

TPUFFS = 0

DO 180 J = 1, 31
   DO 170 I = 1, 31
      TICI(I,J) = 0.0
      SCI(I,J) = 0.0
      DRI(I,J) = 0.0
170 CONTINUE
180 CONTINUE

C ** DEPOSITION VELOCITY = 0.01 M/SEC OR 0.001 M/S

DVI = 0.01 ! M/SEC
INTFLG = .FALSE.
WRITE (LUN(2),190)
190 FORMAT (/5X, 'METEOROLOGICAL DATA HOURLY ', + 'INTERVALS? Y OR N', /7X, 'DEFAULT IS THREE HOUR DATA'/)
READ (LUN(1), 200) SELECT
200 FORMAT (AI)

IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y') INTFLG = .TRUE.

WRITE (LUN(2),220)
220 FORMAT (/5X, 'ENTER DATE FOR SIMULATION START -- MMDDYY')
READ (LUN(1),230,END=999,ERR=999) IMO, IDAY, IYR
230 FORMAT (1012)

IF (IYR .LT. 82 .OR. IYR .GT. 99) GOTO 210
CALL JULIAN (IYR, IMO, IDAY, STDAY)

IF (IMO.LT.1 .OR. IMO.GT.12 .OR. IDAY.LT.1 .OR. IDAY.GT.31) THEN
   WRITE (LUN(2),240) IMO, IDAY, IYR

A.35
240 FORMAT (5X, 'A DATE OF ', I2,'/',I2,'/',I2, ' IS IMPOSSIBLE ',
     + GOTO 210
   ELSE
     WRITE (LUN(2),250) STDAY, IYR
250 FORMAT (/5X, 'JULIAN DATE = ', I3, 4X, '19',I2,/) ENDIF
260 WRITE (LUN(2),270)
270 FORMAT (/5X, 'ENTER HOUR FOR START OF SIMULATION',/) READ (LUN(1),112,END=999,ERR=999) STHR
112 FORMAT (I2)
   IF (STHR .LT. 0 .OR. STHR .GT. 23) THEN
     WRITE (LUN(2),280) STHR
280 FORMAT (5X, 'A STARTING HOUR OF ',I3, ' IS IMPOSSIBLE ',
     + GOTO 260
   ENDIF
   CALL METOPN
   LUINDX = 3
   CALL DATRDLT (LUINDX, 0, 0)
   ALPHAU = 1.0
   ALPHAV = 1.0

C READ THE FILE THAT HAS THE WIND SPEED UNITS VECTOR.
C EACH ELEMENT OF THAT VECTOR CONTAINS A CODE FOR THE TYPE
C OF UNITS USED FOR THE WIND SPEED AT THAT STATION. 1=M/S,2=MPH,3=KTS.

WRITE(LUN(2),231)
231 FORMAT(/,5X,'ENTER THE FILENAME THAT HAS THE WIND SPEED',/,
     * 5X,'CONVERSION VECTOR (WINDU.DAT)'
READ(LUN(1),'(A60)')WINDFILE
   OPEN(UNIT=31,FILE=WINDFILE,STATUS='OLD',IOSTAT=IER)
   IF(IER .NE. 0 )GOTO 991
   DO I = 1,NUMSTA
     READ(31,'(1X,I1)',END=991)WINDU(I)
   ENDDO
   WRITE(LUN(2),232)WINDFILE,(I,I=1,NUMSTA)
   WRITE(LUN(2),233)(WINDU(I),I=1,NUMSTA)
232 FORMAT(/,IX,'WIND SPEED UNIT FILE ',A20,' HAS BEEN READ',5X,
     * '1=M/SEC, 2=MPH, 3=KTS',/,
     * 1X,'STATION :',40(1X,I2))
FORMAT( IX, 'SPD UNIT:', 40(IX, I2), //)
CLOSE(31)
PRINT *
PRINT *, 'ALL FILES OPENED AND FIRST WIND RECORDS PROCESSED'
CALL DATWR
CALL WINDLT
TERFLG = .FALSE.
CALL TERRALT
TT = ((TEMPBB / 10.) - 32.) * (5./9.)
LLD= LLD * 10 + 1
CALL SHIFT

IF (.NOT. EOFMET) THEN ! GET ANOTHER MET RECORD
  CALL DATRDLT
  CALL DATWR
  CALL WINDLT
  IF(TERFLG) CALL TERRALT
  TT = ((TEMPBB / 10.) - 32.) * (5./9.)
  LLD= LLD * 10 + 1
ENDIF

WRITE (LUN(2),300) TITLE, RDATE, RTIME
300 FORMAT (/6X, 'RUN TITLE --> ', 80A1, 5X, A8, 2X, A8)

IF (TITLE(1) .EQ. '*') THEN
  WRITE (LUN(2),310)
310 FORMAT (/8X, 'TESTING MODE SELECTED'/)
ENDIF

WRITE (LUN(2),320) STHR, IMO, IDAY, IYR
320 FORMAT (/5X, 'SIMULATION STARTS AT HOUR ', I2, ' ON DAY ', +
     I2, '-', I2, '-', I2, '/)

RETURN

991 TYPE*, ' ERROR READING THE WIND UNITS CONVERSION FILE ', WINDFILE
999 STOP ' READ ERROR OR END IN INIT'
800 STOP ' ERROR -- UNACCEPTABLE SPECIFICATION IN INIT'

END
SUBROUTINE JULIAN (IYR, IMO, IDAY, JDATE)

REAL LYFLG

INTEGER IYR, IMO, IDAY, JDATE, DYBFOR(12)

DATA DYBFOR / 0,31,59,90,120,151,181,212,243,273,304,334 /

JDATE = DYBFOR(IMO) + IDAY

LYFLG = FLOAT(IYR)/4 - IYR/4

IF (LYFLG .EQ. 0 .AND. IMO .GE. 3) JDATE = JDATE + 1

RETURN
END
SUBROUTINE METOPN

INCLUDE 'DATIM.INC'

OPEN(UNIT=22,FILE=METFILE,
* ACCESS='DIRECT',STATUS='OLD',FORM='FORMATTED',RECL=234,IOSTAT=IER)

IF( IER .NE. 0) THEN
  WRITE(6,'(/,3X,A,A)')'ERROR opening met data file: ',METFILE
  WRITE(6,'(3X,A, 16)')'ERROR # = ', IER
  WRITE(6,'(/3X,A)')'ABORTING SIMULATION DUE TO LACK OF MET DATA ..'
  STOP
ENDIF

MINDEX = 1

READ(22,'(1X,I2,I3,I2)',REC=MINDEX,IOSTAT=IER)DYR,DDAY,DHR

IF( IER .NE. 0) THEN
  WRITE(6,'(3X,A)')'EOF or Error reading 1st record of met data file.'
  WRITE(6,'(3X,A)')'ERROR # = ', IER
  WRITE(6,'(/3X,A)')'ABORTING SIMULATION DUE TO LACK OF MET DATA ..'
  STOP
ENDIF

CALL DTHOUR(DYR,DDAY,DHR,TOTHR_MET)
100 IF(TOTHR_MET.LT. TOTHR_START) THEN
   MINDEX = MINDEX + 1
   READ(22,'(1X,I2,13,12)',REC=MINDEX,IOSTAT=IER)DYR,DDAY,DHR
   IF(IER .NE. 0) THEN
      WRITE(6,'(1X,A,I3)') * 'ERROR READING DATE/TIME GROUP IN MET RECORD ',MINDEX
      WRITE(6,'(1X,A)') * CONTINUE WITH LAST GOOD DATA.'
      MINDEX = MINDEX - 1
      EOFMET = .TRUE.
      RETURN
   ENDIF
   CALL DTHOUR(DYR,DDAY,DHR,TOTHR_MET)
   GOTO 100
ELSE
   MINDEX = MINDEX - 2
   IF(MINDEX .LT. 1) MINDEX = 1
   EOFMET = .FALSE.
   RETURN
ENDIF
END
SUBROUTINE PRINTELT (SHR, IADV, IFLAGS)

INCLUDE 'CHAR.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'MATRIX.INC'
INCLUDE 'UNITS.INC'

INTEGER SHR, istart(3), iend(3)
REAL MATR(31, 31, 4)

DATA istart/1,12,23/  
DATA iend /11,22,31/  

EQUIVALENCE (MATR(1,1,1), EXPCUM)  
EQUIVALENCE (MATR(1,1,2), TICI)  
EQUIVALENCE (MATR(1,1,3), SCI)  
EQUIVALENCE (MATR(1,1,4), DRI)

CHARACTER*50 MATRID(4)  
CHARACTER*20 MATUNT(4)  
CHARACTER*3 monthn(12)  
CHARACTER*15 FNAME1, FNAME2  
CHARACTER*2 CYR

A.41
DATA monthn /'JAN','FEB','MAR','APR','MAY','JUN','JUL',
   + 'AUG','SEP','OCT','NOV','DEC'/

DATA MATRID /
  + 'CUMULATIVE EXPOSURE (UNDEPLETED) ',
  + 'TIME-INTEGRATED AIR CONCENTRATION -- I-131 ',
  + 'MONTH-END SURFACE CONTAMINATION -- I-131 ',
  + 'AVERAGE DEPOSITION RATE -- I-131 '/

DATA MATUNT /
  + 2*'CI - SECONDS / M**3 ',
  + 'CI / M**2 ',
  + 'CI / (M**2 - SECOND)'/

CHARACTER*5 PREFIX(4)

DATA PREFIX / 'N_TIC', '!_TIC', '!_SC', '!_DR'/

CALL HOURDT(TOTHR_RUN, LYR, LMO, LDAY, LHR)

LYEAR = LYR
LMON = LMO
IF((LHR .EQ. 0) .AND. (LDAY .EQ. 1)) THEN
   LMON = LMO - 1
   IF(LMON .LT. 1) THEN
      LMON = 12
      LYEAR = LYR - 1
   ENDIF
ENDIF

IF(IFLAGS .EQ. 1) THEN
   NEND = 1
ELSE
   NEND = 4
ENDIF

DO 200 N = I, NEND

WRITE(CYR,'(I2.2)')LYEAR
FNAME1 = PREFIX(N)//'_//MONTHN(LMON)//CYR//'_OUT'
FNAME2 = PREFIX(N)//'_//MONTHN(LMON)//CYR//'_BIN'

OPEN(UNIT=30,FILE=FNAME1,STATUS='NEW',IOSTAT=IER)
IF(IER .NE. 0) THEN
   WRITE(LUN(2),50) IER,FNAME,30
   50 FORMAT(/,SX,'ERROR NUMBER ',I4,' OPENING FILE ',AIS,
        * ',ON UNIT ',I2)
ELSE
   WRITE(LUN(2),60)FNAME1,30

A.42
OPEN(UNIT=31,FILE=FNAME2,STATUS='NEW',FORM='UNFORMATTED',IOSTAT=IER)
IF(IER .NE. 0) THEN
  WRITE(LUN(2),50) IER,FNAME2,31
ELSE
  WRITE(LUN(2),60)FNAME2,31
ENDIF
WRITE (30,100) TITLE, RDATE, RTIME
WRITE (30,110) MATRID(N), MATUNT(N)
WRITE (30,120) SHR,IADV,LHR,MONTHN(LMO),LDAY,LYR
DO 160 L = I, 3
  WRITE (30,130) (I,I=istart(l),iend(l))
  FORMAT (1HO,'Y COORDINATE OF',25X,'X COORDINATE OF',/2X,'THE GRID POINTS', I4, 11110/)
  DO 150 J = I, 31
    WRITE (30,140) 32-J, (MATR(I,32-J,N), I=istart(l),iend(l))
    FORMAT (111, 5X, 1P11E10.2)
  CONTINUE
CONTINUE
WRITE(31) TITLE, RDATE, RTIME
WRITE(31) MATRID(N), MATUNT(N)
WRITE(31) ((MATR(I,J,N),I=1,31),J=1,31)
CLOSE(30)
CLOSE(31)
CONTINUE
CONTINUE
RETURN
END
SUBROUTINE PUFFIN

INCLUDE 'PUFFS.INC'
INCLUDE 'UNITS.INC'

CHARACTER*8 PRUNTIME
CHARACTER*9 PRUNDATE
CHARACTER*15 FNAME
CHARACTER*80 PTITLE

WRITE( LUN(2), '(/,A)') 'ENTER PUFF FILE NAME '
READ( LUN(1), '(A15)') FNAME

IF( FNAME .EQ. ' ') THEN

WRITE( LUN(2), '( A )' ) 'NO PUFF FILE SPECIFIED'
GOTO 999

ELSE

OPEN( UNIT = 40, FILE = FNAME,
+ FORM = 'UNFORMATTED', STATUS = 'OLD', IOSTAT = IER )

IF( IER .NE. 0 ) THEN

WRITE( LUN(2), 50 ) IER, FNAME, 40

A.44
50 FORMAT( / 5X, 'ERROR NUMBER ', I4, ' OPENING FILE ', A15, + ' ON UNIT ', I2 ' UNABLE TO READ PUFF DATA ')
    STOP

ELSE
    WRITE( LUN(2), 60 ) FNAME, 40

50 FORMAT( / 5X, 'FILE ', A15, ' OPENED ON UNIT ', I2 )
ENDIF

READ( 40 ) PTITLE, PRUNDATE, PRUNTIME
READ( 40 ) TPUFFS, XP, YP, ZP, QP, DXS, DYS, SIGMAY, SIGMAZ, QPI, MF, SP

CLOSE( 40 )

WRITE( LUN(2), 100 ) TPUFFS, PTITLE, PRUNDATE, PRUNTIME

ENDIF

999 CONTINUE
RETURN
END
SUBROUTINE PUFFMLT( M )

INCLUDE 'CONST.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'REL.INC'
include 'statn.inc'
INCLUDE 'WINDS.INC'

REAL XG(2), YG(2), DXSFC(2), DYSFC(2),
+ RX, RY, RX1, RY1, ZFU, ZFV

FAC = 3600.0 / (NPH * DELXY)

** RELEASE IS ABOVE THE MIXED LAYER; USE UPPER LEVEL WINDS ONLY

IF (ZP(M) .GT. 10.0) THEN

   DUX = (C1 * UG + C2 * UUG) * FAC
   DUY = (C1 * VG + C2 * VVG) * FAC
   IF(ZP(M) .GE. LDEPTH) THEN
       DXS(M) = DUX
       DYS(M) = DUY
       RETURN
   ENDF I
ENDIF

END
XG(1) = XP(M)
YG(1) = YP(M)

DO 200 K = 1,2

   I = XG(K) + 1.0
   IF (I .LT. 1 .OR. I .GE. 16) RETURN
   J = YG(K) + 1.0
   IF (J .LT. 1 .OR. J .GE. 16) RETURN
   RX = XG(K) - (I-1)
   RY = YG(K) - (J-1)
   RXI = 1.0 - RX
   RYI = 1.0 - RY

   UCOMP = (RXI*RYI) * (C1*U(I,J) + C2*UU(I,J))
   UCOMP = UCOMP + (RY*RXI) * (C1*U(I,J+1) + C2*UU(I,J+1))
   UCOMP = UCOMP + (RX*RY) * (C1*U(I+1,J) + C2*UU(I+1,J))
   RXSFC(K) = UCOMP * FAC

   VCOMP = (RXI*RYI) * (C1*V(I,J) + C2*VV(I,J))
   VCOMP = VCOMP + (RY*RXI) * (C1*V(I,J+1) + C2*VV(I,J+1))
   VCOMP = VCOMP + (RX*RY) * (C1*V(I+1,J) + C2*VV(I+1,J))
   Dysfc(K) = VCOMP * FAC

   IF (K.EQ.2) GOTO 200
   XG(K+1) = XG(K) + DXSFC(K)
   YG(K+1) = YG(K) + Dysfc(K)

200  CONTINUE

DXS(M) = (DXSFC(1) + DXSFC(2)) / 2.0
DYS(M) = (DYSFC(1) + DYSFC(2)) / 2.0

C ** INTERPOLATE BETWEEN GROUND LEVEL AND UPPER LEVEL WINDS

IF (ZP(M) .GT. 10.0) THEN

C **** ADDED AND MODIFIED FOR CURRENT STATE OF UPPER LEVEL WIND
C DATA FROM HMS, ASSUMES USE OF 400 FT WIND FOR UPWIND ****

   ZFU = ((ZP(M) -10.0) / (ldepth - 10.0)) ** ALPHAU
   ZFV = ((ZP(M) -10.0) / (ldepth - 10.0)) ** ALPHAV

C *****************************************************************************

201  DXS(M) = DXS(M) + (DUX - DXS(M)) * ZFU
       DYS(M) = DYS(M) + (DUY - DYS(M)) * ZFV
ENDIF

RETURN
END
**PUFFOUT**

MESOILT2 VERSION 1.1

J. V. Ramsdell, K. W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/28/89
Updated: 3/07/90

Description: PUFFOUT writes out the data for puffs on the grid at the end of the simulation period. This information can be read in at the beginning of the next period using PUFFIN.

Relationship to Other Modules:
- Makes Calls to: None
- Called From: MESOILT2

```fortran
SUBROUTINE PUFFOUT

INCLUDE 'CHAR.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'UNITS.INC'

CHARACTER*2 CYR
CHARACTER*3 MONTHN(12)
CHARACTER*6 PREFIX
CHARACTER*15 FNAME

DATA MONTHN / 'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN', 'JUL',
+ 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' /

DATA PREFIX / 'PUFFS_' /

WRITE( CYR, '(I2.2)' ) LYEAR

FNAME = PREFIX // MONTHN( LMON ) // CYR // '.BIN'

OPEN( UNIT = 40, FILE = FNAME,
+ FORM = 'UNFORMATTED', STATUS = 'NEW', IOSTAT = IER )

IF( IER .NE. 0 ) THEN
  WRITE( LUN(2), 50 ) IER, FNAME, 40

A.48
```
50    FORMAT( / 5X, 'ERROR NUMBER ', I4, ' OPENING FILE ', A15,
    + ' ON UNIT ', I2 )
    RETURN
ELSE
    WRITE( LUN(2), 60 ) FNAME, 40
60    FORMAT( / 5X, 'FILE ', A15, ' OPENED ON UNIT ', I2 )
ENDIF

WRITE( 40 ) TITLE, RDATE, RTIME
WRITE( 40 ) TPUFFS, XP, YP, ZP, QP, DXS, DYS, SIGMAY, SIGMAZ,
    + QPI, MF, SP
CLOSE( 40 )

WRITE( LUN(2), 100 ) TPUFFS, FNAME
100   FORMAT( / 5X, 'DATA ON ', I4, ' PUFFS WAS WRITTEN TO FILE ', A15 )

RETURN
END
**PUFFR release module**

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/07/90

Description: Assigns initial attributes to puffs at release time.

Relationship to other modules:

Makes calls to: NONE

Called from: MESOILT2

**NO PENETRATION OF LID DUE TO PLUME RISE**

IF (ZP(TPUFFS), GT. LDEPTH .AND. zsourc(ns).lt. ldepth) + + ZP(TPUFFS) = LDEPTH
QP(TPUFFS) = QSOURCE(NS)

IF(.NOT. IFLAGS) THEN
    QPI(TPUFFS) = QSOURCE(NS)
ENDIF
SIGMAZ(TPUFFS) = 3.1
SIGMAY(TPUFFS) = 7.3

RETURN
END
SUBROUTINE readq(q)

INCLUDE 'datim.inc'

OPEN(UNIT=22,FILE='qfile.dat',
    STATUS='old',ACCESS='direct',FORM='formatted',RECL=23)

READ(22,'(1x,i2,i3,i2,3x,f12.5)',REC=qindex)QYR,QJDAY,QHR,Q

WRITE(6,300)qyr,qjday,qhr,q
300 FORMAT(/,2x, 'Change in Source Term. -- Date: ',i2,i4,i3,
    '5X, ' New Q = ',f12.5,/) 

TOTHR Q = TOTHR_NEXTQ
QINDEX = QINDEX + 1

GET DATE FOR NEXT Q

READ(22,'(1x,i2,i3,i2)',REC=qindex)QYR,QJDAY,QHR

CALL DTHOUR(QYR,QJDAY,QHR,TOTHR_NEXTQ)

CLOSE(22)

RETURN
END
SUBROUTINE REARNG

INCLUDE 'STATN.INC'

I = 0
DO 100 K = 1, NUMSTA
   IF (STATUS(K) .EQ. I) GOTO 100
   I = I + 1
   XSTA(I) = XSTA(K)
   YSTA(I) = YSTA(K)
   NAMST(I) = NAMST(K)
   ELEV(I) = ELEV(K)
100 CONTINUE

RETURN
END
SUBROUTINE RELEASLT (DDAY, DHR)

INCLUDE 'CONST.INC'
INCLUDE 'DECAY.INC'
INCLUDE 'REL.INC'
INCLUDE 'UNITS.INC'

INTEGER IYR(4), IMO(4), IDY(4), DDAY, DHR, DMIN,
+       DURHR, DURMIN

CHARACTER SELECT*1

WRITE (LUN(2),110)
WRITE (LUN(5),110)
110 FORMAT (/5X, 'MESOI ---> RELEASE INITIALIZATION',://)

NSOURC = 1       ! ALWAYS

DO 440 NS = 1, NSOURC
440 WRITE (LUN(2),160) NS
160 FORMAT (/5X, 'SPECIFY COORDINATES (X,Y,Z) OF SOURCE', I3, /5X,
+ ' IN KILOMETERS FROM GRID CENTER AND METERS ABOVE GROUND',://)

READ (LUN(1),*,END=999,ERR=999) XDIST, YDIST, ZSOURC(NS)
XSOURC(NS) = XDIST * 1000. / DELXY + 7.5
YSOURC(NS) = YDIST * 1000. / DELXY + 7.5
IF (XSOURC(NS) .LT. 0.0 .OR. XSOURC(NS) .GT. 15.0) GOTO 450
IF (YSOURC(NS) .LT. 0.0 .OR. YSOURC(NS) .GT. 15.0) GOTO 450
IF (ZSOURC(NS) .LT. 0.0 .OR. ZSOURC(NS) .GT. 300.) GOTO 470

170 WRITE (LUN(2), 180)
180 FORMAT (/5X, 'ENTER DATE OF RELEASE --> MMDDYY')
READ (LUN(1), 190, END=999, ERR=999) IMO(NS), IDY(NS), IYR(NS)
190 FORMAT (1012)

IF (IYR(NS) .LT. 82 .OR. IYR(NS) .GT. 99) GOTO 490
IF (IMO(NS) .LT. 1 .OR. IMO(NS) .GT. 12) GOTO 490
IF (IDY(NS) .LT. 1 .OR. IDY(NS) .GT. 31) GOTO 490

CALL JULIAN (IYR(NS), IMO(NS), IDY(NS), RELDAY(NS))

IF (RELDAY(NS) .GT. (DDAY+2)) GOTO 510
IF (RELDAY(NS) .LT. DDAV .AND. RELDAY(NS) .GT. 2) GOTO 510

210 WRITE (LUN(2), 220)
220 FORMAT (/5X, 'TIME OF RELEASE? HOURS, MINUTES')
READ (LUN(1), *, END=999, ERR=999) RELHR(NS), RELMIN(NS)

IF (RELHR(NS) .LT. 0 .OR. RELHR(NS) .GT. 23) GOTO 530

PARTHR(NS) = RELMIN(NS) / (60/NPH) + 1

MAXPUF(NS) = 500

280 DURHR = 0
290 DURMIN = 0

WRITE (LUN(2), 290) NPH

WRITE (LUN(2), 290) NPH

440 CONTINUE
GOTO 600

** ERROR MESSAGE SECTION **

450 WRITE (LUN(2), 460) XSOURC(NS), YSOURC(NS)
460 FORMAT (/5X, 'SOURCE IS OFF GRID -- X = ', F10.2, 5X,
          'Y = ', F10.2//)
GOTO 150

470 WRITE (LUN(2), 480) ZSOURC(NS)
480 FORMAT (/5X, 'SOURCE HEIGHT OF ', F6.1, ' METERS IS UNREALISTIC')
GOTO 150

490 WRITE (LUN(2), 500) IMO(NS), IDY(NS)
500 FORMAT (/5X, 'RELEASE >> A DATE OF ', 212, ' IS NOT VALID')
GOTO 170

A.55
510 WRITE (LUN(2),520) DDAY, DHR, RELDAY(NS), RELHR(NS)
520 FORMAT (/5X, 'RELEASE >> DATA STARTS ON DAY ', I3, ' AT HOUR ', I2, /5X, 'A RELEASE ON DAY ', I3, ' AT HOUR ', I2, ' CANNOT BE SIMULATED')
   GOTO 170
530 WRITE (LUN(2),540) RELHR(NS)
540 FORMAT (/5X, 'RELEASE >> SPECIFIED HOUR OF RELEASE IS NOT', ' POSSIBLE --> ', I5)
   GOTO 210
550 WRITE (LUN(2),560) RELMIN(NS)
560 FORMAT (/5X, 'RELEASE >> SPECIFIED MINUTES OF RELEASE NOT', ' POSSIBLE --> ', I5)
   GOTO 210
600 PRINT *, ' SOURCE DEFINITION COMPLETE'
610 WRITE (LUN(5),610) NSOURC
610 FORMAT (8X, 'NUMBER OF SOURCES = ', I2/)
   DO 680 NS = 1, NSOURC
      WRITE (LUN(5),620) NS, XSOURC(NS), YSOURC(NS), ZSOURC(NS)
620 FORMAT (/5X, 'SOURCE ', I2, ': LOCATED AT WIND GRID ', 2(F5.2,1X), 'AND ', F6.1, ' METERS ABOVE GROUND/')
      WRITE (LUN(5),630) RELHR(NS), RELMIN(NS), IMO(NS), IDY(NS), IYR(NS), MAXPUF(NS), QSOURC(NS)
630 FORMAT (/17X, 'RELEASE WILL OCCUR AT ', I2, ':', I2, ' ON ', I2, '-', I2, '-', I2, '/', F8.3, ' PUFFS WITH ', F8.3, ' MASS EACH')
680 CONTINUE
   RETURN
999 STOP 'ERROR IN READ OR END OF READ IN RELEASE'
END
**SHIFT**

Save wind field, times and temperature

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/07/90

Description: Module to save parameters from the previous hour. Called at the end of each simulation hour.

Relationship to other modules:

Makes calls to: NONE
Called from: MESOILT2, INITLT

---

SUBROUTINE SHIFT

INCLUDE 'DATIM.INC'
INCLUDE 'WINDS.INC'

DO 100 J = 1, 16
   DO 200 I = 1, 16
       U(I,J) = UU(I,J)
       V(I,J) = VV(I,J)
   200 CONTINUE
100 CONTINUE

UG = UUG
VG = VVG
T = TT

LD = LLD
PRCP = PRCPP
IST1 = IST2
TEMPA = TEMPA
TEMPB = TEMPB

RETURN
END
SIGMA

SIGMA Mesoilt2 Version 1.1

Diffusion Curves Used In XQDOQ and PAVAN

Athey, G.F., J.V. Ramsdell, C.S. Glantz
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/10/90

Description: Computes new diffusion coefficients given the
last values, atmospheric stability, mixing layer thickness
and distance moved.

Relationship to other modules:

Makes calls to: NONE

Called from: DIFDEPLT

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

SUBROUTINE SIGMA (DSMTRI, STAB, LDEPTH, SIGMAZ, SIGMAY)

REAL AY(7), AZ(7,3), BZ(7,3), CZ(7,3)

INTEGER STAB

DATA AY/ 0.3658, 0.2751, 0.2089, 0.1471, 0.1046, 0.0722, 0.0481/
DATA AZ/ 0.192, 0.156, 0.116, 0.079, 0.063, 0.053, 0.032,
+ 0.0066, 0.0382, 0.113, 0.222, 0.211, 0.086, 0.052,
+ 0.0024, 0.055, 0.113, 1.26, 6.73, 18.05, 10.83 /
DATA BZ/ 0.936, 0.922, 0.905, 0.881, 0.871, 0.814, 0.814,
+ 1.941, 1.149, 0.911, 0.725, 0.678, 0.74, 0.74,
+ 2.094, 1.098, 0.911, 0.516, 0.305, 0.18, 0.18 /
DATA CZ/ 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
+ 9.27, 3.3, 0.0, -1.7, -1.3, -0.35, -0.21,
+ -9.6, 2.0, 0.0, -13., -34.0, -48.6, -29.2 /

XVY = ( SIGMAY/AY(STAB) )**(1.0/0.9031)
XEW = XVY + DSMTRI
SIGMAY = AY(STAB) * XEW**0.9031

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

C ** SIGMA Z COMPUTATIONS

C ** CHECK INITIAL SIGMA Z SIZE AGAINST MAXIMUM

SZLIM = 0.8 * LDEPTH
IF ( SIGMAZ .GE. SZLIM) RETURN

A.58
C ** COMPUTE VIRTUAL DISTANCE

\[ XVZ = ( \sigma_{\text{AZ}} / AZ(\text{STAB},1) )^* (1.0 / BZ(\text{STAB},1)) \]

IF ( SIGMAZ.GT.1.0 ) THEN
  IF((XVZ+DSMTRI).GT.100.0.AND.SIGMAZ.GT.CZ(\text{STAB},2)) THEN
    XVZ= ( ( SIGMAZ - CZ(\text{STAB},2) ) / AZ(\text{STAB},2) )**
      ( 1.0 / BZ(\text{STAB},2) )
  IF((XVZ+DSMTRI).GT.1000.0.AND.SIGMAZ.GT.CZ(\text{STAB},3)) THEN
    XVZ= ( ( SIGMAZ - CZ(\text{STAB},3) ) / AZ(\text{STAB},3) )**
      ( 1.0 / BZ(\text{STAB},3) )
  ENDIF
ENDIF
ENDIF

XEZ = XVZ + DSMTRI

IF ( XEZ .LE. 100.0 ) THEN
  SIGMAZ = AZ(\text{STAB},1) * XEZ ** BZ(\text{STAB},1)
ELSE IF ( XEZ .LE. 1000.0 ) THEN
  SIGMAZ = AZ(\text{STAB},2) * XEZ ** BZ(\text{STAB},2) + CZ(\text{STAB},2)
ELSE IF ( XEZ .GT. 1000.0 ) THEN
  SIGMAZ = AZ(\text{STAB},3) * XEZ ** BZ(\text{STAB},3) + CZ(\text{STAB},3)
ENDIF

IF (SIGMAZ .GT. SZLIM) SIGMAZ = SZLIM

RETURN
END
SUBROUTINE STRAY
REAL RT(40)
INTEGER NSB(40)
INCLUDE 'CHAR.INC'
INCLUDE 'STATN.INC'
INCLUDE 'WINDS.INC'
YG = 0.0
DO 200 J = 1, 16
  XG = 0.0
  DO 300 I = 1, 16
    DO 100 L = 1, ACTSTA
      NSB(L) = L
      RSQR = (XSTA(L)-XG)**2 + (YSTA(L)-YG)**2
      IF (RSQR.NE.0.0) THEN
        RT(L) = SQRT(RSQR)
      ELSE
        RT(L) = RSQR
      END IF
  END DO
  A.60
CONTINUE

IF (ACTSTA.GT.1) CALL ASCND (RT, NSB, ACTSTA)

DO 150 L = 1, ACTSTA
    STDIST(L,I,J) = RT(L)
    STNUM(L,I,J) = NSB(L)
150 CONTINUE

XG = XG + 1.0

CONTINUE

YG = YG + 1.0

CONTINUE

RETURN

END
**DESCRIPTION:**

This module is called after the wind subroutine has decoded the meteorological data into u and v components. It modifies the components for each grid point according to the terrain. Revisions have been made to accommodate variable grid spacings.

**Relationship to other modules:**

- Makes calls to: DIRSPD
- Called from: MESOILT2, INITLT

---

**SUBROUTINE TERRALT**

```c
SUBROUTINE TERRALT

INCLUDE 'CONST.INC'
INCLUDE 'UNITS.INC'
INCLUDE 'WINDS.INC'

INTEGER NCR
CHARACTER*1 SELECT
CHARACTER*40 LINE
REAL SCOEFF(16,16,2), STANGL(16,16)

** SET ALL TERRAIN THAT MAY NOT BE PROCESSED TO FLAT TERRAIN

DATA SCOEFF / 512 * 1.0 /
DATA STANGL / 256 * -1.0 /

TERRAIN DATA IS STORED IN FILE CALLED 'TERRAN.DAT'
Each record consists of:

- COL 1-2: X GRID DESIGNATOR 0-15
- COL 4-5: Y GRID DESIGNATOR 0-15
- COL 7-10: ANGLE IN DEGREE FROM NORTH 0-179.999
- COL 12-15: 1ST COEFFICIENT; FOR TRANSPORT TO THE RIGHT
- COL 17-20: 2ND COEFFICIENT; FOR TRANSPORT TO THE LEFT
```

A.62
IF (TERFLG) GOTO 150

WRITE(6,295)
295 FORMAT(IX,'DOES THIS SITE HAVE TERRAIN DATA? Y OR N: ') READ(LUN(1),'(A)') SELECT

IF(SELECT.EQ.'Y'.OR.SELECT.EQ.'y') THEN
   OPEN (UNIT=LUN(15),STATUS='OLD',FILE='TERRAN.DAT',I STAT=IER)
   IF (IER .NE. 0) THEN
      PRINT *, 'ERROR OPENING WIND FIELD MODIFICATION FILE; ',
      'USE FLAT UNALTERED WIND FIELDS'
      GOTO 140
   ELSE
      PRINT *, 'WIND FIELD WILL BE MODIFIED FOR TERRAIN EFFECTS'
   END IF
ELSE
   GO TO 140
END IF

END

C ** IN CREATING TERRAIN FILES, COEFFICIENTS ARE CHOSEN SO THAT
C ** THE GREATER THE SLOPE, THE LARGER THE COEFFICIENT. USER
C ** INPUTTED LEVEL SFCS ARE 0.0. AFTER PROCESSING BELOW, THIS
C ** IS REVERSED, SO THAT THE GREATER THE SLOPE, THE SMALLER THE
C ** COEFFICIENT. ALL LEVEL SURFACES ARE THEN HAVE COEFFICIENTS OF
C ** 1.0. TERRAIN COEFFICIENTS ARE USED TO DETERMINE THE AMOUNT
C ** THE WIND IS TURNED PARALLEL TO TERRAIN FEATURES.

IREAD = 0

100 READ (LUN(15),110,END=130,ERR=120) I,J,STANGL(I+1,J+1),
    + SCOEFF(I+1,J+1,1), SCOEFF(I+1,J+1,2)
110 FORMAT (IX,J2,IX,JX,I2,FX,F4.0,F4.0,F4.0,F4.0)

SCOEFF(I+1,J+1,1) = 1.0 - SCOEFF(I+1,J+1,1)
SCOEFF(I+1,J+1,2) = 1.0 - SCOEFF(I+1,J+1,2)
IREAD = IREAD + 1

GOTO 100

120 STOP 'TERSCA RD ERR'
130 CONTINUE

WRITE(6,135) IREAD
135 FORMAT (/5X, I4, ' TERRAIN POINTS DEFINED IN TERRA'/)

CLOSE (LUN(15))
TERFLG=.TRUE.

C -- LOOPS TO CALCULATE EFFECT OF TERRAIN ON THE WINDS AT EACH GRID POINT

150 DO 200 J = 1,16
   DO 180 I = 1,16

C --------------------- MODIFY ONLY GRID POINTS WITH TERRAIN ---------------------

   IF( STANGL(I,J) .LT. 0.0 ) GOTO 180

C ADJUST TRANSPORT VECTOR FOR TERRAIN

   CALL DIRSPD(UU(I,J),VV(I,J),WDIR,WSPD)

C CONVERT DIR FROM WIND TO TRANSPORT DIRECTION

   TDIR = WDIR + 180.0
   IF( TDIR.GT.360.0) TDIR = TDIR - 360.0

C COMPUTE ANGLE BETWEEN TRANSPORT AND TERRAIN VECTORS ( -180 TO 180 DEGREES)

   RANGLI=TDIR-STANGL(I,J)
   IF (RANGLI.GE.0.0.AND.RANGLI.LT.180.0) THEN
      A=SIN(RANGLI*DTR)
      B=-COS(RANGLI*DTR)
      TURN=90.0
      K=1
   ELSE
      A=-SIN(RANGLI*DTR)
      B=COS(RANGLI*DTR)
      TURN=270.0
      K=2
   ENDIF

C COMPUTE AND ADJUST TRANSPORT COMPONENTS
C (V=PERPENDICULAR TO TERRAIN VECTOR)

   IF (SCOEFF(I,J,K).GE.1.0) GOTO 180

   U1 = WSPD * B
   V1 = WSPD * A
   V2 = V1*SCOEFF(I,J,K)

C EXAM UPPER LEVEL WIND

   IF (ABS(U1) .LT. 0.005) THEN
      CALL DIRSPD(UUG,VVG,UD,US)
      UTDIR = UD + 180.0
      IF( UTDIR.GT.360.0) UTDIR = UTDIR - 360.0
URANGL1=UTDIR-STANGL(I,J)
IF (URANGL1.GE.0.0.AND.URANGL1.LT.180.0) THEN
   B=-COS(URANGL1*DTR)
ELSE
   B=COS(URANGL1*DTR)
END IF
U1 = WSPD * B
IF (ABS(U1) .LT. 0.005) U1=-1
ENDIF

U2=SQR(T(WSPD**2-V2**2))
U2=SIGN(U2,U1)

C COMPUTE NEW RELATIVE WIND DIRECTION, CONVERT TO RELATIVE THEN
C ABSOLUTE TRANSPORT DIRECTIONS

IF (U2.GT.0.0) RDIR2=ATAN(V2/U2)
IF (U2.LT.0.0) RDIR2=ATAN(V2/U2)+(180.0 * DTR)
IF (U2.EQ.0.0 .AND. V2.GT.0.0) RDIR2=90.0
IF (U2.EQ.0.0 .AND. V2.LT.0.0) RDIR2=270.0*DTR
TDIR2=RDIR2 - ( STANGL(I,J) + TURN ) * DTR

C COMPUTE UU AND VV TRANSPORT COMPONENTS

UU(I,J) = WSPD*COS(TDIR2)
VV(I,J) = WSPD*SIN(TDIR2)

180 CONTINUE
200 CONTINUE
140 RETURN
END
**TESTM**

Testing mode output module

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/08/90

Description: Subroutine to print additional puff information when the model is run in the testing mode.

Relationship to other modules:

Makes calls to: NONE

Called from: MESOILT2

---

**SUBROUTINE TESTM (IADV, SHR, DELXY)**

```
INCLUDE 'datim.INC'
INCLUDE 'PUFFS.INC'
INCLUDE 'UNITS.INC'
INCLUDE 'winds.INC'

INTEGER SHR

WRITE(LUN(5),7007)DDAY, DYR, DHR, DMIN
WRITE(LUN(5),7004)
WRITE(LUN(5),7005)I, I=0,15
DO 7010 J=16,1,-1
    WRITE(LUN(5),7000)J-1,(UU(I,J),I=1,16)
7010 CONTINUE

7000 FORMAT(1X,I2,':',4X,16(IX,F6.3))
7005 FORMAT(1X,2X,5X,16(3X,I2,2X))
7004 FORMAT(20X,':*************** UU (M/S) ***************')
7007 FORMAT(1H1,20X,'JDATE: ',13,' 19',12,' TIME: ',12,':',12)///)
7003 FORMAT(///20X,':*************** VV (M/S) ***************')

WRITE(LUN(5),7003)
WRITE(LUN(5),7005)I, I=0,15

DO 7011 J=16,1,-1
    WRITE(LUN(5),7000)J-1,(VV(I,J),I=1,16)
```

A.66
7011 CONTINUE

WRITE(LUN(5),7012)

7012 FORMAT(1X///)

IF (IAADV.EQ.1 .OR. IFIRST.NE.1) THEN
  WRITE (LUN(2),100) SHR
  FORMAT (///,10X,'TESTING MODE -- PUFF DESCRIPTORS',
  ' FOR SIMULATION HOUR ', I4)
  WRITE (LUN(2),110)
  FORMAT (///,1X,'ADVC PUFF SRC GRID GRID Z',
  ' MOVE X', 4X,'MOVE Y', 5X,'SIGMA Y SIGMA Z', 6X,'QP',
  ' QPI')
  WRITE (LUN(2),120)
  II0 FORMAT (1X,'STEP NO', 17X,'X', 6X,'Y')
ENDIF

WRITE (LUN(2),130)

130 FORMAT (1X)

DO 200 M = 1, TPUFFS

  XMOVE = DXS(M) * DELXY
  YMOVE = DYS(M) * DELXY

  WRITE (LUN(2),150) IADV, M, SP(M), XP(M), YP(M),
  ZP(M), XMOVE, YMOVE, SIGMAY(M), SIGMAZ(M), QP(M),
  QPI(M)
150 FORMAT (1X,I2, 3X,I3, 4X,I1, 10X,F5.2, 2X,F5.2,
  2X,F4.0, 2(3X,F7.0), 2X, 2(2X,F7.1), 2X, 2(1X,1PE10.3))

200 CONTINUE

IFIRST = 1

RETURN
END
TOPFIL
Topography data input module

Athey, G.F., J.V. Ramsdell, C.S. Glantz, K.W. Burk
Pacific Northwest Laboratory
PO Box 999
Richland, Washington 99352

Created: 6/1/83
Updated: 3/07/90

Description: Module to input ground level elevations to describe
topography for the 31 x 31 fine grid.

Relationship to other modules:

Makes calls to: NONE
Called from: MESOILT2

SUBROUTINE TOPFIL

INCLUDE 'CONST.INC'
INCLUDE 'REL.INC'
INCLUDE 'TOPOGR.INC'
INCLUDE 'UNITS.INC'

REAL ALPHA(2), BETA(2)

CHARACTER*80 FILE1

TOPFLG = .FALSE.
DO 110 J = 1, 31
   DO 100 I = 1, 31
      TOPO(I,J) = 0.0
  100 CONTINUE
  110 CONTINUE

WRITE (LUN(2),120) DELXY/2.0
120 FORMAT (/2X, 'TOPFIL -- THE FINE GRID HAS BEEN INITIALIZED ', +
      'WITH A SPACING OF ', F7.0, ' METERS.', /1X,
      'SPACING OF THE TOPOGRAPHY DATA POINTS MUST BE THE SAME')

130 PRINT *, 'ENTER NAME OF FILE CONTAINING TOPOGRAPHY DATA'
140 FORMAT ( A80 )
IF (FILE1 .EQ. 'NONE'. OR. FILE1 .EQ. 'none') THEN
  PRINT *, 'TOPO ARRAY SET TO ZEROES; UNIFORM, FLAT TERRAIN'
  GOTO 900
ELSE
  OPEN (UNIT=LUN(I5), FORM='FORMATTED', STATUS='OLD',
   FILE=FILE1, IOSTAT=IER)
  IF (IER .NE. 0) THEN PRINT*, 'TOPFIL >> ERROR ',IER, ' OPENING TOPOGRAPHY FILE'
    GOTO 130
  END IF
END IF
TOPFLG = .TRUE.
DO 200 J = 31, 1, -1
  READ (LUN(I5), I50, IOSTAT=IER) (TOPO(I,J),I=1,31)
150 FORMAT (IX, 31I4)
  IF (IER .GT. 0) THEN
    PRINT *, 'TOPFIL >> ERROR ', IER, ' READING FILE'
    STOP
  ENDIF
  IF (IER .LT. 0) THEN
    PRINT *, 'EOF DURING TOPOGRAPHY FILE READ'
    PRINT *, J-1, ' ROWS ENTERED; REMAINDER WILL BE ZEROES'
    GOTO 900
  ENDIF
200 CONTINUE
900 CLOSE (LUN(I5))
RETURN
END
SUBROUTINE WINDLT

INCLUDE 'CONST.INC'
INCLUDE 'DATIM.INC'
INCLUDE 'STATN.INC'
INCLUDE 'UNITS.INC'
INCLUDE 'WINDS.INC'

REAL USTA(40), VSTA(40), USPD

INTEGER LFLAG(40)

C ** DECODE WIND DIRECTION AND SPEED FOR EACH ACTIVE STATION
C ** CHECK FOR MISSING DATA (9999)

RCH = 5.0
N = 0

DO 110 J = 1, NUMSTA
   IF (STATUS(J) .EQ. 1) GOTO 110
   N = N + 1
   DIR(N) = DATA(J) / 100
   SPD(N) = (DATA(J) - DIR(N) * 100)
   IF (DIR(N) .EQ. 88 .OR. SPD(N) .EQ. 88) GOTO 100
   IF (DIR(N) .EQ. 99 .OR. SPD(N) .EQ. 99) GOTO 100
   DIR(N) = DIR(N) * 10
   IF (DIR(N) .EQ. 0 .AND. SPD(N) .EQ. 0) GOTO 100

A.70
LFLAG(N) = 0

C ** CALCULATE U AND V COMPONENTS AND CONVERT TO METERS/SEC

ANG = (270. - DIR(N)) * DTR
USTA(N) = (SPD(N) * COS(ANG)) * CONV(WINDU(J))
VSTA(N) = (SPD(N) * SIN(ANG)) * CONV(WINDU(J))
GOTO 110

100  LFLAG(N) = 1
     IF (N .GE. ACTSTA) GOTO 135
110  CONTINUE

120  FORMAT (5X, 'WARNING -- STATION NUMBER ', I2, /7X,
       +    'DIRECTION OUT OF NORMAL RANGE --> ', F6.1,
       +    /7X, 'JDAY = ', I3, ' AT ', I2, ':', I2//)

C ** UPPER LEVEL WIND

135  udirl=udirl
        UDIR = UPWIND / 100
        USPD = (UPWIND - (UPWIND/100) * 100)
        UDIR = UDIR * 10
        IF (UDIR .LT. 0 .OR. UDIR .GT. 360) THEN
            UDIR=udirl
        ENDIF
        IF (USPD .Le. 0 .OR. USPD .GT. 75) THEN
            WRITE (LUN(2),150) USPD, DDAY, DHR, DMIN
150  FORMAT (/5X, 'WARNING --> UPPER LEVEL WIND SPEED IS OUT ',
       +    'OF NORMAL RANGE', /17X, F5.0, ' MPH ON JDAY ',
       +    I3, ' AT ', I2, ':', I2//)
        USPD=1
        ENDIF
        UANG = (270. - UDIR) * DTR
        UUG = USPD * COS(UANG) * CONV(WINDU(1))
        VVG = USPD * SIN(UANG) * CONV(WINDU(1))

C ** WIND INTERPOLATION SECTION; USES UP TO 10 STATIONS TO CALCULATE
C ** U AND V FOR GRID POINTS(I,J)

200  DO 280 J = 1, 16
     DO 270 I = 1, 16
        SNU = 0.0
        SNV = 0.0
        SND = 0.0
        NS  = 0
        LEND = MIN(ACTSTA,40)
    270  CONTINUE
    280  CONTINUE

A.71
DO 220 L = 1, LEND
LS = STNUM(L,I,J)
IF (LFLAG(LS) .EQ. 1) GOTO 220
C ** STATION IS AT THE GRID POINT, NO INTERPOLATION REQUIRED
   IF (STDIST(L,I,J) .LE. 1.0E-7) GOTO 250
   IF (NS .LT. 3) GOTO 210
   IF (STDIST(L,I,J) .GT. RCH) GOTO 260
210   SNU = SNU + USTA(LS) / STDIST(L,I,J)**2
   SNV = SNV + VSTA(LS) / STDIST(L,I,J)**2
   SND = SND + (1.0 / STDIST(L,I,J))**2
   NS = NS + 1
   IF (NS.GE.5) GO TO 260
220   CONTINUE
   IF (NS.EQ.0) THEN
      WRITE (6,230) I, J, DDAY, DHR, DMIN
      WRITE (60,230) I, J, DDAY, DHR, DMIN
      GO TO 300
230   FORMAT (/5X, 'ERROR -- NO DATA AVAILABLE TO ',
       + 'CALCULATE WINDS AT GRID POINT ', I2, ',', I2,
       + '/7X, 'JDAY = ', I3, 5X, 'AT ', I2, ':', I2//)
   END IF
250   UU(I,J) = USTA(LS)
   VV(I,J) = VSTA(LS)
   GOTO 270
260   UU(I,J) = SNU / SND
   VV(I,J) = SNV / SND
270   CONTINUE
280   CONTINUE
   GO TO 500
300   CONTINUE
   WRITE(LUN(2),305)DDAY,DHR,DMIN
305   FORMAT (/5X, 'NO SURFACE WIND DATA FOR DAY ',I4,' HOUR ',I3,
       + ' MINUTE ',I3,/5X,' PERSISTENCE ASSUMED ')
   DO 315 J = 1,16
      DO 310 I = 1,16
         UU(I,J)=U(I,J)
         VV(I,J)=V(I,J)
310    CONTINUE
315    CONTINUE
500 CONTINUE
RETURN
END
CHAR. INC

Created: 6/1/83
Updated: 3/12/90
Description: Character variable include file

Included in modules: CLEANLT, DIFDEPLT, INITLT, MESOILT2,
PRINTLT, PUFFOUT, STRAY

CHARACTER*1 TITLE(80)
CHARACTER*8 RTIME
CHARACTER*9 RDATE

COMMON /CHAR/ TITLE, RDATE, RTIME

CONST. INC

Created: 6/1/83
Updated: 3/12/90
Description: Constant include file

Included in modules: CLEANLT, DIFDEPLT, GRININLT, INITLT,
MESOILT2, PUFFMLT, PUFFER, RELEASELT,
TERRALT, TOPFIL, WINDLT

REAL CHIMIN, ADT, DELXY, TWOPI, WC(6), DVR, DVL,
+ XS(31), YS(31), RCH, ALPHAU, ALPHAV, C1, C2

LOGICAL INTFLG

INTEGER NPH, IOPDTA

DATA PI, TWOPI, DTR /3.141593, 6.283185, 0.01745329/

COMMON /CONST/ CHIMIN, ADT, DELXY, PI, TWOPI, C1, C2,
+ WC, XS, YS, RCH, ALPHAU, ALPHAV, NPH, IOPDTA,
+ INTFLG, DTR, DVI

A.74
DATIM.INC

Created: 6/1/83
Updated: 3/12/90

Description: Data, date and time include file

Included in modules: DATRDLT, DATWR, DIFDEPLT, INITLT, MESOILT2, METOPN, PRINTELT, PUFFMLT, PUFFOUT, READQ, SHIFT, TESTM, WINDLT

INTEGER Dyr, Dday, Dh, Dmm, Drec, stab, Ldepth, Precip,  
+ Upwind, Data(40), Tmpt, Tpt,  
+ Qyr, Qjday, Qhr, iyr, stdy, sthr,  
+ ld, lld, Tmpta, Tmpbb, prcp, prcpp, ist1, ist2

INTEGER TOThr_start, TOThr_stop, TOThr_q, TOThr_nextq,  
* TOThr_met, TOThr_nextmet, TOThr_run, qindex, mindex,  
* TOThrs, TOTsec, lmon, lyear

CHARACTER*20 Metfile
LOGICAL EOFmet

COMMON /METFIL/ Metfile, EOFMET

COMMON /DIRECT/ TOThr_start, TOThr_stop, TOThr_q, TOThr_nextq,  
* TOThr_met, TOThr_nextmet, TOThr_run, qindex, mindex,  
* TOThrs, TOTsec, lmon, lyear

COMMON /DATIM/ Dyr, Dday, Dh, Dmm, Drec, stab,  
+ Ldepth, Precip, Upwind, Data, Tmpt, Tmpbb,  
+ Ttp, imonthc,  
+ Qyr, Qjday, Qhr, iyr, stdy, sthr,  
+ ld, lld, Tmpta, Tmpbb, prcp, prcpp, ist1, ist2
C******************************************************************************
C DECAY.INC
C Created: 6/1/83
C Updated: 3/12/90
C Description: Radioactive decay variable include file
C Included in modules: DIFDEPLT, MESOILT2, PUFFR, RELEASLT
C******************************************************************************

REAL HLIFEI, DECAYI, TRDKI
COMMON /DECAY/HLIFEI, DECAYI, TRDKI

C******************************************************************************
C MATRIX.INC
C Created: 6/1/83
C Updated: 3/12/90
C Description: Grid matrix array include file
C Included in modules: DIFDEPLT, INITLT, MESOILT2, PRINTELT
C******************************************************************************

REAL EXPCUM(31,31),
  * TICI(31,31),
  * SCI(31,31),
  * DRI(31,31)

COMMON / MATRIX / EXPCUM, TICI, SCI, DRI
C*******************************************************************
C C C C C C C C C C C C
PUFFS.INC
C Created: 6/1/83
C Updated: 3/12/90
C Description: Puff parameter include file
C Included in modules: CLEANLT, DIFDEPLT, INITLT, MESOILT2,
C PUFFIN, PUFFMLT, PUFFOUT, PUFFR,
C TESTM
C*******************************************************************

REAL XP(500), YP(500), ZP(500), QP(500), DXS(500), DYS(500),
+ SIGMAY(500), SIGMAZ(500),
+ QPI(500)

INTEGER TPUFFS, MF(500), SP(500)

COMMON /PUFFS/ XP, YP, ZP, QP, DXS, DYS, SIGMAY,
+ SIGMAZ, TPUFFS, MF, SP,
+ QPI

C*******************************************************************
C C C C C C C C C C C C
REL.INC
C Created: 6/1/83
C Updated: 3/12/90
C Description: Release parameter include file
C Included in modules: DIFDEPLT, INITLT, MESOILT2, PUFFMLT,
C PUFFR, RELEASET, TOPFIL
C*******************************************************************

REAL XSOURC(4), YSOURC(4), ZSOURC(4), QSOURC(4), QH(4),
+ STEMPC, RSTACK

INTEGER NSOURC, NPUFFS(4), RELDAY(4), RELHR(4), RELMIN(4),
+ PARTHR(4), MAXPUF(4)

COMMON /REL/ XSOURC, YSOURC, ZSOURC, QSOURC, QH, STEMPC, EFFLUX,
+ NSOURC, NPUFFS, RELDAY, RELHR, RELMIN, PARTHR,
+ MAXPUF, RSTACK

A.77
STATN.INC

Created: 6/1/83
Updated: 3/12/90

Description: Station parameter include file

Included in modules: GRINMLT, INITLT, MesoILT2, PUFFMLT, REARNG, STRAY, WINDLT

C*******************************************************************

CHARACTER*4 NAMST(40)
REAL XSTA(40), YSTA(40)
INTEGER ACTSTA, STATUS(40), ELEV(40)
COMMON /STATN/ NUMSTA, XSTA, YSTA, ACTSTA, STATUS, ELEV

C*******************************************************************

TOPOGR.INC

Created: 6/1/83
Updated: 3/12/90

Description: Topography data include file

Included in modules: DIFDEPLT, DTOPO, MesoILT2, TOPFIL

C*******************************************************************

INTEGER TOPO(31,31)
LOGICAL TOPFLG

COMMON /TOPOGR/ TOPO, TOPFLG, TPMI, TPMF
C*******************************************************************************
C
C UNITS.INC
C
Created: 6/1/83
Updated: 3/12/90

Description: Logical unit assignment include file

Included in modules: CLEANLT, DATRAILT, DATWR, GRIDINLT
INITLT, MESOILT2, PRINTLT, PUFFIN,
PUFFOUT, RELEASELT, TERRALT, TESTM,
TOPFIL, WINDLT

C*******************************************************************************

INTEGER LUN(20)

COMMON / UNITS / LUN

DATA LUN / 5, 6, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
+ 21, 22, 23, 28, 30, 31, 32/

C*******************************************************************************
C
C WINDS.INC
C
Created: 6/1/83
Updated: 3/13/90

Description: Wind grid include file

Included in modules: INITLT, MESOILT2, PUFFMLT, SHIFT,
STRA, TERRALT, TESTM, WINDLT

C*******************************************************************************

REAL STDIST(40,16,16), U(16,16), V(16,16), UG, VG, UU(16,16),
+ VV(16,16), UUG, VVG

REAL CONV(3)

INTEGER WINDU(40)

INTEGER STNUM(40,16,16), DIR(40), SPD(40)

LOGICAL TERFLG

COMMON /WINDS/ STDIST, U, V, UG, VG, UU, VV, UUG, VVG, STNUM,
+ SPD, DIR, TERFLG, CONV, WINDU

DATA CONV/ 1.0, .447, .514/

A.79
APPENDIX B

PHASE II CODE LISTING
APPENDIX B

PHASEII CODE LISTING

******************************************************************************
* PHASEII.FOR
* Mesoilt2 Post-Processor
* K. W. Burk, J.V. Ramsdell
* Pacific Northwest Laboratory
* P.O. Box 999
* Richland, Washington 99352
* Created: 8/01/89
*
* Description: Reads in the monthly MESOILT2 output and
* produces Atmospheric Transport Calculation
* Results, by Census District, by month, for
* a five year period. Total Exposure, and all
* Iodine output is used.
******************************************************************************

REAL GRIDS(31,31,5,4), Y(40,5)
REAL OUTMEAN(83,4), OUTSTDV(83,4)
REAL C2(4), VMIN(4)

CHARACTER*80 TITLE
CHARACTER*50 MATRID
CHARACTER*20 MATUNT
CHARACTER*9 RDATE
CHARACTER*8 RTIME
CHARACTER*6 FILE(4)
CHARACTER*4 DISTID(83)
CHARACTER*3 CMONTH(12)
CHARACTER*2 YEAR(5), CID

INTEGER YR, GPTS(83,40,2), JCOUNT(83), DISTNUM, NPTSOUT(83)

DATA FILE /'N_TIC_', 'I_TIC_', 'I_DR_', 'I_SC_'/
DATA CMONTH /'JAN', 'FEB', 'MAR', 'APR', 'MAY', 'JUN',
  'JUL', 'AUG', 'SEP', 'OCT', 'NOV', 'DEC' /
DATA YEAR /'83', '84', '85', '86', '87' /
READ IN THE CENSUS DISTRICT NODE ASSIGNMENTS AND PUT THE
OUTPUT INTO ARRAYS FOR EACH TYPE, DISTRICT, AND YEAR. (4, 83, 5)
DONE ONLY ONCE

TYPE*, 'OPENING CENSUS DISTRICT FILE ...'

DO I = 1, 83

READ(3, 100) DISTNUM, DISTID(I), ((GPTS(I, J, 1), GPTS(I, J, 2)), J = 1, 9)
100 FORMAT(IX, I2, 2X, A4, 1X, 9(2X, I2, I2))
READ(3, 101) (GPTS(I, J, 1), GPTS(I, J, 2), J = 10, 18)
READ(3, 101) (GPTS(I, J, 1), GPTS(I, J, 2), J = 19, 27)
READ(3, 101) (GPTS(I, J, 1), GPTS(I, J, 2), J = 28, 31)
101 FORMAT(10X, 9(2X, I2, 1X, I2))

DO J = 1, 31
  IF(GPTS(I, J, 1) .EQ. 0) GOTO 888
END DO
888 JCOUNT(I) = J - 1  ! NUMBER OF GRID POINTS IN DISTRICT

TYPE*, 'OPENING AND READING INPUT FILES ...'

DO I = 1, 4
  DO J = 1, 5
    OPEN(UNIT=7, FILE=FILE(I)//CMONTH(MONTH)//YEAR(J)//'.BIN',
         STATUS='OLD', FORM='UNFORMATTED')
    READ(7) TITLE, RDATE, RTIME
    READ(7) MATID, MATUNT
    READ(7) ((GRIDS(L, M, J, I), L = 1, 31), M = 1, 31)
    CLOSE(7)

CONTINUE
BEGIN COMPUTATIONS

TYPE*, 'BEGINNING COMPUTATIONS ... ', CMONTH(MONTH)
TYPE*

DO I = 1, 83
  TYPE*, 'DISTRICT ', I
  IF(I.EQ.69) GOTO 69
  DO K = 1, 4
    NPTS = 0
    SUM = 0.0
    SUMSQ = 0.0
    DO J = 1, 5
      DO L = 1, JCOUNT(I)
        IF(GRIDS(GPTS(I, L, 1), GPTS(I, L, 2), J, K) .GE. VMIN(K)) THEN
          Y(L, J) = LOG10(GRIDS(GPTS(I, L, 1), GPTS(I, L, 2), J, K)) - 3.0
        ENDIF
      ENDDO
      NPTS = NPTS + 1
      SUM = SUM + Y(L, J)
      SUMSQ = SUMSQ + Y(L, J)**2
    ELSE
      Y(L, J) = LOG10(VMIN(K)) - 3.0  ! ARBITRARY ZERO
    ENDDO
  END DO
  OUTMEAN(I, K) = SUM / NPTS
  OUTSTDV(I, K) = 0.0
  IF(NPTS .GT. 0) THEN
    OUTMEAN(I, K) = SUM / NPTS
    IF(NPTS .GT. 1) THEN
      VAR = SUMSQ / (NPTS - 1) - SUM**2 / (NPTS * (NPTS - 1))
      OUTSTDV(I, K) = SQRT(VAR + C2(K))
    ENDDO
  ELSE
    OUTSTDV(I, K) = 0.0
  ENDIF
  IF(K .EQ. 1) THEN
    NPTSOUT(I) = NPTS
  ENDIF
END DO

CONTINUE

ENDDO

C OUTPUT TO A BINARY FILE AND A FORMATTED файл

B.3
OPEN(UNIT=2,FILE='PHASEII'//CMONTH(MONTH)//'.BIN',
   STATUS='NEW',FORM='UNFORMATTED')
OPEN(UNIT=3,FILE='PHASEII'//CMONTH(MONTH)//'.OUT',
   STATUS='NEW',FORM='FORMATTED')

DO I = 1,83
   IF(I .EQ. 69) GOTO 6969
   WRITE(2)DISTID(I),JCOUNT(I)*5,NPTSOUT(I),
      (OUTMEAN(I,J),OUTSTDV(I,J),J=1,4)
   WRITE(3,500)DISTID(I),JCOUNT(I)*5,NPTSOUT(I),
      (OUTMEAN(I,J),OUTSTDV(I,J),J=1,4)
500   FORMAT(IX,A4,3X,I3,15,2X,4(2F10.5,5X))
6969   CONTINUE
ENDDO

CLOSE(2)
CLOSE(3)
CLOSE(9)

MONTH = MONTH + 1
IF(MONTH .LE. 12) GOTO 111
TYPE*
TYPE*, ' END OF PROGRAM - PHASEII'
TYPE*
STOP'
END
APPENDIX C

MESOILT2 CODE MODIFICATIONS
APPENDIX C

MESOILT2 CODE MODIFICATIONS

Appendix A is a list of the version of MESOILT2 used in Phase I calculations. That version of the code requires that wind directions be specified to the closest 10 degrees. Prior to 1965, wind directions were recorded by compass points [for example, north (N), north-northeast (NNE)]. This appendix 1) describes changes to MESOILT2 that will allow users to specify wind directions in numerals corresponding to the compass points used to report wind directions in the early period of Hanford operations and 2) corrects two minor problems in the code listed in Appendix A that were identified when the code was converted to run on a computer other than a VAX computer using the VMS operating system.

Wind Direction Unit Option

Extension of MESOILT2 to allow entry of wind directions as compass points involves adding a variable to indicate how the directions are being entered. This new variable is WINDDU. WINDDU is a vector that contains a wind direction unit indicator for each meteorological station. If the wind directions at the station are reported to the nearest 10 degrees, the direction unit indicator is 1; if the directions are reported by a 16-point compass, the indicator is 2. Values for WINDDU are entered via a file read in subroutine INITLT. The file contains one record for each meteorological station. The name of the file is entered by the user.

Subroutine INITLT, as listed in APPENDIX A, contains a section that reads wind speed conversion factors. That section of the code ends with the lines

233  FORMAT( 1X,'SPD UNIT:',40(1X,I2),//)
CLOSE(31)

To modify INITLT to permit use of wind directions reported by compass points, add the following code segment following the CLOSE statement.

C.1
WRITE(LUN(2),1231)
1231 FORMAT(/,5X,'ENTER THE FILENAME THAT HAS THE WIND DIRECTION',/,
   5X,'CONVERSION VECTOR (WINDDU.DAT)')
READ(LUN(1),'(A20)')WINDFILE

OPEN(UNIT=31,FILE=WINDFILE,STATUS='OLD',IOSTAT=IER)
IF(IER .NE. 0 )GOTO 991

DO I = 1,NUMSTA
   READ(31,'(1X,I1)',END=991)WINDU(I)
ENDDO

WRITE(LUN(2),1232)WINDFILE,(I,I=1,NUMSTA)
WRITE(LUN(2),1233)(WINDDU(I),I=1,NUMSTA)
1232 FORMAT(/,IX,'WIND DIRECTION UNIT FILE ',A20,' HAS BEEN READ',5X,
   1X,'1=TENS OF DEGREES, 2=16-POINT DIRECTION',/,
   1X,'STATION : ',40(1X,I2))
1233 FORMAT(1X,'DIR UNIT: ',40(Ix,I2)//)
CLOSE(31)

The variable WINDDU is passed from subroutine INITLT to subroutine WINDLT, where it is used via the WINDS common block. To add WINDDU to the common block, make the following changes to WINDS. INC: Change the statement

INTEGER WINDU(40)

and change

COMMON /WINDS/ STDIST, U, V, UG, VG, UU, VV, UUG, VVG, STNUM,
   + SPD, DIR, TERFLG, CONV, WINDU

and change

COMMON /WINDS/ STDIST, U, V, UG, VG, UU, VV, UUG, VVG, STNUM,
   + SPD, DIR, TERFLG, CONV, WINDU, WINDDU

If wind directions recorded as compass points are to be used in MESOILT2, the compass points must be converted to numerical values. The conversion convention established for MESOILT2 is NNE = 1, NE = 2, ... NNW = 15, and N = 16. Calm winds are represented by zeros for both the wind direction and the wind speed. A wind direction greater than 16 will cause the wind data for the station to be treated as missing data.

C.2
Wind direction data are read in the DATRDLT subroutine. In the Mesoilt2 code listed in Appendix A, the variable DATA, which is used for temporary storage of the wind data reported by the meteorological stations, is a one-dimensional vector. The elements of DATA are read as I4 variables. The first half of each element of DATA is the wind direction, and the second half is the speed. When the code was extended to permit wind directions to be entered by compass points, DATA was changed to a two-dimensional matrix. In the matrix DATA(I,J), I is the meteorological station index, and J is the data index. The elements DATA(I,1) are the wind directions, and the DATA(I,2) elements are the wind speeds.

The change in the variable DATA requires modification to the DATIM include block and to subroutines DATRDLT, DATWR, and WINDLT. In the DATIM include block, change the INTEGER type statement from

```
INTEGER DYR, DDAY, DHR, DMIN, DREC, STAB, LDEPTH, PRECIP,
 + UPWIND, DATA(40), TEMPA, TEMPB, T, TT,
 + QYR, QJDAY, QHR, iyr, stday, sthr,
 + LD, LLD, TEMPAA, TEMPBB, PRCP, PRCPP, IST1, IST2
```

to

```
INTEGER DYR, DDAY, DHR, DMIN, DREC, STAB, LDEPTH, PRECIP,
 + UPWIND, DATA(40,2), TEMPA, TEMPB, T, TT,
 + QYR, QJDAY, QHR, iyr, stday, sthr,
 + LD, LLD, TEMPAA, TEMPBB, PRCP, PRCPP, IST1, IST2
```

No other changes are required in DATIM.

Meteorological data are read by subroutine DATRDLT. The statement that reads the data in the code listed in Appendix A and the accompanying format are

```
READ(22,110,rec=mindex,ERR=900) DYR,DDAY,DHR,DMIN,IST2,LLD,PRCPP,
 * UPWIND,TEMPAA,TEMPBB,(DATA(I),I=1,IMAX)
110 FORMAT(IX,I2,I3,I2,I2,IX,II,IX,I3,IX,II,IX,
 * I4,IX,I3,IX,13,40(IX,14))
```

Change these statements to

```
READ(22,110,rec=mindex,ERR=900) DYR,DDAY,DHR,DMIN,IST2,LLD,PRCPP,
 * UPWIND,TEMPAA,TEMPBB,
   (DATA(I,1),DATA(I,2),I=1,IMAX)
```

C.3
Subroutine DATWR writes the meteorological data to the log file before conversion or use of the data. Change the following WRITE and FORMAT statements from

```fortran
WRITE (LUN(2), 140) UPWIND, (DATA(I), I=1, 10)
140 FORMAT (8X, 'WINDS', 3X, 14, IX, 10(I4, I4))
WRITE (LUN(2), 160) (DATA(I), I=11, 20)
WRITE (LUN(2), 160) (DATA(I), I=21, 30)
WRITE (LUN(2), 160) (DATA(I), I=31, 40)
160 FORMAT (21X, 10(I4, I4))
```

to

```fortran
WRITE (LUN(2), 140) UPWIND, (DATA(I,1), DATA(I,2), I=1, 10)
140 FORMAT (8X, 'WINDS', 3X, 14, IX, 10(I4, I4))
WRITE (LUN(2), 160) (DATA(I,1), DATA(I,2), I=11, 20)
WRITE (LUN(2), 160) (DATA(I,1), DATA(I,2), I=21, 30)
WRITE (LUN(2), 160) (DATA(I,1), DATA(I,2), I=31, 40)
160 FORMAT (21X, 10(I4, I4))
```

The remaining changes made in modifying MESOILT2 to permit use of wind directions reported in compass points are in subroutine WINDLT. These changes are found in the following code segment.

```fortran
DIR(N) = DATA(J,1)
SPD(N) = DATA(J,2)
IF(WINDDU(J) .EQ. 1) THEN
  IF((DIR(N) .GT. 36) .OR. (SPD(N) .GT. 80)) GOTO 100
ELSE IF(WINDDU(J) .EQ. 2) THEN
  IF((DIR(N) .GT. 16) .OR. (SPD(N) .GT. 80)) GOTO 100
ENDIF
IF(WINDDU(J) .EQ. 2) THEN
  DIR(N) = IFIX(22.5 * FLOAT(DIR(N)))
ELSE
  DIR(N) = DIR(N) * 10
ENDIF
LFLAG(N) = 0
IF(DIR(N) .EQ. 0 .AND. SPD(N) .EQ. 0) THEN
C.4
```
The code segment should replace the existing code in the listing of WINDLT in Appendix A that is between:

\[ N = N + 1 \]

and

C ** CALCULATE U AND V COMPONENTS AND CONVERT TO METERS/SEC

in the DO 110 loop.

Correction to Subroutine INITLT

Subroutine INITLT contains two calls to subroutine DATRDLT. The first call includes a list of arguments; the second call does not. The arguments in the list are used only in a statement at the end of DATRDLT that prints a message if the end of the meteorological data file is reached. They do not affect operation of the code.

There are two reasonable fixes. One fix is to add the argument list to the second call. This is the easier of the fixes.

The second fix is to delete the argument list from the first call. If the argument list is deleted from the call, it should also be deleted from the SUBROUTINE statement in DATRDLT. In addition, three statements in the subroutine should be deleted, and one statement should be added. The INTEGER SHR statement and the PRINT statements at the end of DATRDLT should be deleted. A 900 CONTINUE statement should be added to replace the 900 PRINT .... statement.

Correction to Subroutine PUFFR

The logical variable IFLAGS is used in subroutine PUFFR but is not defined in or passed to the subroutine. IFLAGS is defined in the main program, where it is used to bypass a number of calculations during code testing. A similar use of the variable was intended in PUFFR. However, the time savings in PUFFR would not be significant. Consequently, the fix proposed is
to delete the IF(.NOT. IFLAGS) THEN and ENDIF statements. This change will cause the code to execute as it did during the Phase I calculations.

The alternative fix to the code is to add IFLAGS to the list of arguments in the call to PUFFR from MESOILT2 and in the argument list in the SUBROUTINE statement in PUFFR. This fix also requires the addition of a type statement to PUFFR that defines IFLAGS as a logical variable.
APPENDIX D

SUMMARY OF TSP COMMENTS AND BATTelle RESPONSES
### APPENDIX D
### SUMMARY OF TSP COMMENTS AND BATTELLE RESPONSES

<table>
<thead>
<tr>
<th>Comment Number</th>
<th>Commenter</th>
<th>Page, Paragraph</th>
<th>Comment Summary</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>John E. Till</td>
<td>Page 2.3, para. 2</td>
<td>An example of how topographic data are entered would be useful along with a map showing elevations and coordinates for a selected area.</td>
<td>Added a paragraph on p. 2.3 - a map of this information does not exist.</td>
</tr>
<tr>
<td>2.</td>
<td>JET</td>
<td>Page 2.6, para. 3</td>
<td>Last sentence: This routine should be added before the final version of the report is completed.</td>
<td>Reworded the sentence. Modifications of the code were made, but not included in document because they were not in the code used for Phase I calculations. Added Appendix C on Phase II code modifications.</td>
</tr>
<tr>
<td>3.</td>
<td>JET</td>
<td>Page 2.9</td>
<td>Could a 3-D version of the figure showing coordinates be developed?</td>
<td>NA. Figure is adequate for this report and may not be needed for later reports because of probable changes in code.</td>
</tr>
<tr>
<td>4.</td>
<td>JET</td>
<td>Page 2.21</td>
<td>I recommend numbering equations and identifying the parameters of an equation by stating &quot;where&quot;, followed by symbol =, definition, a comma after each definition, with the word &quot;and&quot; before the final definition.</td>
<td>NA. Equations are in Battelle style. Numbering of equations not needed because equations are not referenced in text.</td>
</tr>
<tr>
<td>5.</td>
<td>JET</td>
<td>Page 2.28, Fig. 2.6</td>
<td>I still do not understand why the grid is shifted to the east. How does this affect our calculating doses in the western regions of Yakima County, for example?</td>
<td>Added a paragraph on p. 2.26.</td>
</tr>
</tbody>
</table>

NA = No action.
## DISTRIBUTION

<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>No. of Copies</th>
</tr>
</thead>
<tbody>
<tr>
<td>OFFSITE</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
</tr>
</tbody>
</table>

**Technical Steering Panel**

- **D. S. Barth**
  University of Nevada
  4505 Maryland Parkway
  Las Vegas, NV  89154

- **W. A. Bishop**
  2503 Wedgewood Court SE
  Olympia, WA  98501

- **M. L. Blazek**
  Oregon Department of Energy
  625 Marion Street N.E.
  Salem, OR  97310

- **G. G. Caldwell**
  Director
  Tulsa City-County Health Dept
  4616 East 15th Street
  Tulsa, OK  74112

- **S. N. Davis**
  Dept. of Hydrology & Water Resources
  Bldg. 11
  University of Arizona
  Tucson, AZ  85721

- **N. J. Germond**
  224 Iron Mountain Blvd.
  Lake Oswego, OR  97034

- **P. C. Klingeman**
  Civil Engineering Dept.
  Oregon State University
  Corvallis, OR  97331-2302

**K. J. Kopecky**
Fred Hutchinson Cancer Research Center
1124 Columbia Street
Seattle, WA  98104

**R. L. Morrill**
Dept. of Geography
DP-10
University of Washington
Seattle, WA  98195

**A. H. Murphy**
c/o Climate Analyses Center
National Weather Service, NOAA
W/NMC51, WWB, Room 604
Washington, DC  20233

**V. Nguyen**
EWA, Inc.
133 1st Ave. N.
Minneapolis, MN  55401

**D. W. Price**
Agricultural Economics
Hulbert Hall Room 211
Washington State University
Pullman, WA  99164-6210

**M. A. Robkin**
Radiological Sciences SB-75
University of Washington
Seattle, WA  98195

**G. S. Roessler**
Rt. 1, Box 139H
Elysian, MN  56028
No. of Copies

B. Shleien
2421 Homestead Drive
Silver Springs, MD 20902

A. P. Slickpoo, Sr.
P.O. Box 331
809 Nez Perce Lane
Kamiah, ID 83536

J. E. Till
Rt. 2 Box 122
Neeses, SC 29107

D. E. Walker, Jr.
P.O. Box 4147
Boulder, CO 80306

2 DOE Office of Scientific and Technical Information
Technical Information Center
P.O. Box 62
Oak Ridge, TN 37830

M. W. Carter
Chairman
4621 Ellisbury Drive NE
Atlanta, GA 30338

S. R. Hanna
Sigma Research Corporation
234 Littleton Road Suite 2E
Westford, MA 01886

C. W. Miller
Illinois Dept. of Nuclear Safety
1035 Outer Park Drive
Springfield, IL 62704

10 J. S. Stohr
Office of Nuclear Waste Mgmt.
Department of Ecology
99 South Sound Center
Mail Stop PV-11
Olympia, WA 98504

No. of Copies

ONSITE

DOE Richland Operations

R. F. Brich, SED A5-55

44 Pacific Northwest Laboratory

K. J. Allwine K6-11
C. M. Berkowitz K6-08
J. C. Doran K6-11
J. G. Droppo K6-96
C. E. Elderkin K6-03
J. W. Falco K6-78
C. S. Glantz K6-03
W. A. Glass K1-40
S. P. Gydesen P8-55
J. M. Hales K6-04
P. C. Hays K6-86
D. J. Hales K6-07
L. A. Mahoney K6-03
A. H. McMakin K6-86
P. C. Owczarski K6-03
W. T. Pennell K6-98
J. V. Ramsdell (5) K6-03
R. L. Skaggs K6-77
E. D. Skyllingstad K6-03
W. L. Templeton K1-30
L. L. Wendell K6-07
S. B. Yabusaki K6-77
Project Office Files (5) K6-89
Public Reading Room (5) A1-65
Publishing
Coordination (2) K1-11
Records Center K3-70
Technical Report Files (5) P8-55