PACIFIC NORTHWEST LABORATORY
RECEIVED
DEC 12 1983
BATTELLE TECH. INFO.

PNL-4753

NUREG/CR-3344
PNL-4753

MESOI Version 2.0: An Interactive Mesoscale Lagrangian Puff Dispersion Model With Deposition and Decay

Prepared by J. V. Ramsdell, G. F. Athey, C. S. Glantz

Pacific Northwest Laboratory
Operated by
Battelle Memorial Institute

Prepared for
U.S. Nuclear Regulatory
Commission

REFERENCE COPY

NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, or any of their employees, makes any warranty, expressed or implied, or assumes any legal liability of responsibility for any third party's use, or the results of such use, of any information, apparatus, product or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 1717 H Street, N.W.
   Washington, DC 20555

2. The NRC/GPO Sales Program, U.S. Nuclear Regulatory Commission,
   Washington, DC 20555

3. The National Technical Information Service, Springfield, VA 22161

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda; NRC Office of Inspection and Enforcement bulletins, circulars, information notices, inspection and investigation notices; Licensee Event Reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the NRC/GPO Sales Program: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, and NRC booklets and brochures. Also available are Regulatory Guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances.*

Documents available from the National Technical Information Service include NUREG series reports and technical reports prepared by other federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal and periodical articles, and transactions. *Federal Register* notices, federal and state legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free upon written request to the Division of Technical Information and Document Control, U.S. Nuclear Regulatory Commission, Washington, DC 20555.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, 7920 Norfolk Avenue, Bethesda, Maryland, and are available there for reference use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018.

GPO Printed copy price: $6.50

NUREG/CR-3344
PNL-4753

MESOI Version 2.0: An Interactive Mesoscale Lagrangian Puff Dispersion Model With Deposition and Decay

Manuscript Completed: September 1983
Date Published: November 1983

Prepared by
J. V. Ramsdell, G. F. Athey, C. S. Glantz

Pacific Northwest Laboratory
Richland, WA 99352

Prepared for
Division of Emergency Preparedness and Engineering Response
Office of Inspection and Enforcement
U.S. Nuclear Regulatory Commission
Washington, D.C. 20555
NRC FIN B1282

ABSTRACT

MESOI Version 2.0 is an interactive Lagrangian puff model for estimating the transport, diffusion, deposition and decay of effluents released to the atmosphere. The model is capable of treating simultaneous releases from as many as four release points, which may be elevated or at ground-level. The puffs are advected by a horizontal wind field that is defined in three dimensions. The wind field may be adjusted for expected topographic effects. The concentration distribution within the puffs is initially assumed to be Gaussian in the horizontal and vertical. However, the vertical concentration distribution is modified by assuming reflection at the ground and the top of the atmospheric mixing layer. Material is deposited on the surface using a source depletion, dry deposition model and a washout coefficient model. The model also treats the decay of a primary effluent species and the ingrowth and decay of a single daughter species using a first order decay process.

This report is divided into two parts. The first part discusses the theoretical and mathematical bases upon which MESOI Version 2.0 is based. The second part contains the MESOI computer code. The programs were written in the ANSI standard FORTRAN 77 and were developed on a VAX 11/780 computer. Two appendices complete the report. The first contains the definitions of and other information related to the MESOI global variables, and the second contains partial model output, which can be used to check the numerical operation of the model after it has been installed on different computer systems.

ACKNOWLEDGEMENTS

The continuing support of MESOI by Dana Hoitink, PNL Geosciences Research and Engineering Department, D.R. Elle, Safety and Quality Assurance Division, Richland Operations Office of the US Department of Energy, and Barry Zalcman, Division of Emergency Preparedness and Engineering Response, US Nuclear Regulatory Commission have made Version 2.0 of MESOI possible. The forecast staff of the Hanford Meteorology Station and the staff of the Hanford Unified Dose Assessment Center have contributed to the current version of MESOI by providing feedback on their experiences with previous versions of MESOI. And, others have contributed to the development of MESOI Version 2.0 by suggesting features that should be added or changed and by reviewing various aspects of the model as it has been developed. The contributions of all are appreciated.

ACKNOWLEDGEMENTS

The continuing support of MESOI by Dana Hoitink, PNL Geosciences Research and Engineering Department, D.R. Elle, Safety and Quality Assurance Division, Richland Operations Office of the US Department of Energy, and Barry Zalcman, Division of Emergency Preparedness and Engineering Response, US Nuclear Regulatory Commission have made Version 2.0 of MESOI possible. The forecast staff of the Hanford Meteorology Station and the staff of the Hanford Unified Dose Assessment Center have contributed to the current version of MESOI by providing feedback on their experiences with previous versions of MESOI. And, others have contributed to the development of MESOI Version 2.0 by suggesting features that should be added or changed and by reviewing various aspects of the model as it has been developed. The contributions of all are appreciated.

Contents

TABLES

FIGURES

PROGRAM LISTINGS

Model Initialization

Data Input/Output

Transport, Diffusion and Depletion

General Utility

MESOI VERSION 2.0: AN INTERACTIVE MESOSCALE LAGRANGAIN PUFF DISPERSION MODEL WITH DEPOSITION AND DECAY

INTRODUCTION

MESOI Version 2.0 is a third generation Lagrangian puff transport and diffusion model based on the MESODIF model prepared initially by Start and Wendell for the National Reactor Testing Station (now the Idaho National Engineering Laboratory) (Start and Wendell 1974), and MESOI Versions 1.0 and 1.1 (Ramsdell and Athey 1981, and Athey and Ramsdell 1982). It has been developed for use by the US Department of Energy, Richland Operations Office and the US Nuclear Regulatory Commission, Office of Inspection and Enforcement in responding to emergencies at nuclear facilities.

This version of MESOI explicitly treats physical processes affecting material released to the atmosphere that were not treated in previous versions. Additions to the model include: depletion of puffs by dry deposition at surfaces and by precipitation scavenging, decay of radionuclides and ingrowth of daughter products, specification of the vertical variation of the horizontal wind field, adjustment of the wind field to conform to major terrain features, incorporation of terrain height in diffusion computations, and estimation of plume rise. MESOI Version 2.0 also permits specification of multiple release points (up to a maximum of 4); they may be either ground-level or elevated locations. Finally, three additional parameterizations have been provided for estimating the diffusion coefficients.

The main reasons for the revision of MESOI Versions 1.0 and 1.1 were incorporation of the processes that alter and deplete effluents and more realistic treatment of the wind field that transports the effluents. Radioactive decay and daughter ingrowth are treated using the differential equations that describe these processes. Depletion of the puffs is treated using a source depletion model and constant deposition velocity for dry deposition, and wet deposition is treated using a washout model with a coefficient that is a function of precipitation type and rate. The models are not state-of-the-art, but they should provide a reasonable indication of areas where surface contamination is likely to be high. The addition of an upper-level wind and vertical interpolation of the horizontal winds between the surface wind field and the upper-level wind provides a means for estimating some of the effects of wind shear on effluent transport.

Changes have been made in the data required for model execution, but the increases are modest. It is now necessary to specify an upper-level wind, such as the geostrophic wind, a precipitation type and intensity, and an air temperature. Topographic data must be provided if modification of the wind field is desired, or if terrain is to be considered in diffusion computations. The data are contained in two data files, one containing information for use in adjusting the wind field and the other for use in the diffusion computations. If the required topographic data are not available, MESOI Version 2.0 can be run as a "flat earth" model, just as it predecessors were run.

Model output has also been modified. If desired, plots of the initial surface wind data and the interpolated wind field, adjusted for terrain, are available. Time-integrated air concentrations (exposures) for each grid point in the model domain are available for the entire period of release. If the effluent decays, time-integrated air concentrations, assuming effluent depletion, are available for both the parent and daughter nuclides. Short-term exposures, computed assuming effluent depletion, are available for both the parent and daughter nuclides. They may be obtained for each grid point. Both short-term and long-term surface contamination from the primary effluent and a daughter, if appropriate, may also be obtained. Finally, MESOI Version 2.0 signals the time at which the exposure at each of up to 35 points of interest exceeds two predefined levels.

In upgrading the earlier versions of MESOI, five criteria were established for use in evaluating potential model modifications. These criteria were: 1) representation of as many physically real processes as possible, 2) flexibility of the model to accept as much meteorological data as becomes available, 3) easily understood input and output, 4) minimal program size, and 5) minimum program execution time. In this revision of the model, the same criteria are used, although the relative weight given to each individual criterion is changed.

The controlling criterion is the time required to run the model. No change or group of changes that caused the model execution time to exceed 15 minutes would be retained. In practice, model execution time for simulation of a 12 hour release from a single source remains less than 5 minutes, including the time required for model initialization and operator interactions. In Versions 1.0 and 1.1 of MESOI, the size of the program was given a relatively high weight. Those versions of the program were targetted for computers with rather limited memories; this version is targetted for minicomputers with relatively large memories, such as the Digital Equipment Corporation VAX computers and the Data General MV 8000 computers. However, MESOI is a highly modular program so there should not be too much difficulty in fitting it into smaller minicomputers or one of the larger microcomputers.

This description of MESOI Version 2.0 is divided into two parts. The first part includes this introduction and the following sections that describe the equations that form the theoretical basis for the model. The second part contains the MESOI Version 2.0 code. The code for each program element is presented separately along with a description of the purpose of the element and a discussion of the specific manipulations being performed. Interrelationships between the program elements are defined in this section, and a list of model variable names and their associated meanings is presented. The last part contains several data sets that can be used to test MESOI program operation along with the resulting output.

Two appendices are included. Appendix A contains a listing of the major variables used in MESOI, i.e. those variables used in more than one program element. The variables are listed in alphabetical order along with their definitions, listings of the program elements in which they are defined and used, and specification of the method by which they are passed between program elements. Appendix B defines scenarios that may be used to check MESOI operation following implementation on other computer systems. Scenarios are provided for evaluating puff transport, the depletion processes, and overall model operation. Selected output is included in the Appendix for each scenario for comparison purposes.

Two appendices are included. Appendix A contains a listing of the major variables used in MESOI, i.e. those variables used in more than one program element. The variables are listed in alphabetical order along with their definitions, listings of the program elements in which they are defined and used, and specification of the method by which they are passed between program elements. Appendix B defines scenarios that may be used to check MESOI operation following implementation on other computer systems. Scenarios are provided for evaluating puff transport, the depletion processes, and overall model operation. Selected output is included in the Appendix for each scenario for comparison purposes.

PART I. MESOI THEORETICAL BASES

DIFFUSION WITHIN PUFFS

MESOI uses a Gaussian puff approach to modeling diffusion. Material released to the atmosphere is assumed to be contained in one or more puffs. In the absence of reflection at surfaces, the distribution of material within the puffs is assumed to be Gaussian. With this assumption, the concentration at a point in space (x, y, z) is given by:

$$\chi(x,y,z) = \frac{Q}{(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z} \exp\left\{-\frac{1}{2}\left[\left(\frac{x-x_o}{\sigma_x}\right)^2 + \left(\frac{y-y_o}{\sigma_y}\right)^2 + \left(\frac{z-z_o}{\sigma_z}\right)^2\right]\right\} \quad (1)$$

where Q is the mass of material contained in the puff, (x_o, y_o, z_o) represents the position of the center of the puff, and the diffusion coefficients sigma x, sigma y, and sigma z (σ_x, σ_y, and σ_z) are characteristic puff dimensions in the x, y, and z directions. The derivation of the equation is discussed in many places including: Pasquill (1974), Gifford (1968) and Csanady (1973).

The right-hand side of (1) may be treated as the product of three factors: the source strength, Q, the relative concentration that accounts for the decrease in concentration at the center of the puff, $1/[(2\pi)^{3/2}\sigma_x\sigma_y\sigma_z]$, and the exponential term to account for the decrease in concentration with increasing distance from the puff center. This section is concerned with estimating the diffusion coefficients and modifications to the exponential term to account for the effects of topography and initial release characteristics. Following sections cover modifications to the source strength to account for deposition on surfaces and radioactive decay, and movement of the puffs.

Equation (1) allows for different puff dimensions in the along wind (x), crosswind (y) and vertical (z) directions. Only the crosswind and vertical directions are of major importance, because MESOI integrates the concentrations at a point as puffs pass. Therefore, in MESOI it is assumed that horizontal puff cross sections are circular, i.e. that sigma x is equal to sigma y. In the absence of terrain effects, the concentration profile in the horizontal is Gaussian, and concentrations are a function of only the distance from the puff center. The characteristic radial dimension is referred to as sigma y.

At the time a puff is released, the center is assigned coordinates (x_o, y_o, z_o) where x_o and y_o are the horizontal coordinates of the release point and z_o is the effective release height. The effective release height, also denoted by h_e, is the height of the actual release point above ground plus any puff rise due to buoyancy or momentum. The values of x_o and y_o change as puffs are advected by the wind, but the value of h_e remains constant. Puffs rise and fall along with the terrain elevation directly beneath their centers. The terrain elevation at off center locations is assumed to affect height, therefore, the effective release height is corrected for the difference in elevation between a position directly beneath the puff center and the point for which concentrations are being estimated. The correction is accomplished by defining a position dependent or local effective release height h_e',

$$h_e' = h_e - (T_r - T_c) \qquad (2)$$

where T_r and T_c are the terrain elevations at the receptor and puff center, respectively. The local effective release height may be either positive or negative.

Diffusion of puffs, as defined by (1), assumes that material is free to spread to infinity in all directions. That assumption is unrealistic, particularly for diffusion in the vertical. As an alternative, the vertical diffusion can be assumed to be limited by the ground and the top of the atmospheric mixing layer. In Version 1 of MESOI, the ground was assumed to be a reflecting surface, but not the top of the mixing layer. The top of the mixing layer only limited diffusion. In Version 2.0, both the ground and top of the mixing layer are assumed to be reflecting surfaces.

Mixing layer thickness is one of the meteorological variables that is needed for model execution. A single atmospheric mixing layer thickness is defined for the model domain. The top of the mixing layer is assumed to rise and fall to conform to changes in the elevation of the terrain. While the mixing layer thickness is uniform in space, it is not constant in time. The mixing layer thickness varies from hour-to-hour according to the meteorological conditions as defined in the model input.

Figures 1 and 2 show the physical relationships between the local effective release height, the terrain and the top of the mixing layer. Figure 1 shows the behavior of the center of the puffs as they move over terrain. By contrast with Figure 1, Figure 2 shows the physical relationships as a function of position at a specific instant. In particular, it should be noted that the effective release height directly beneath the center of the puff does not change, but that away from the puff center the local effective release height is altered by variations in the height of the terrain.

Reflection is incorporated in the model by adding another set of exponential terms. This set will be discussed in detail shortly, however two limiting cases are sufficiently important to be singled out here. If the effective release height and sigma z are small compared to the mixing layer thickness and the concentration at ground-level is to be estimated, the reflection terms simply result in increasing the concentration estimated with (1) by a factor of two. At the other extreme, as sigma z approaches the

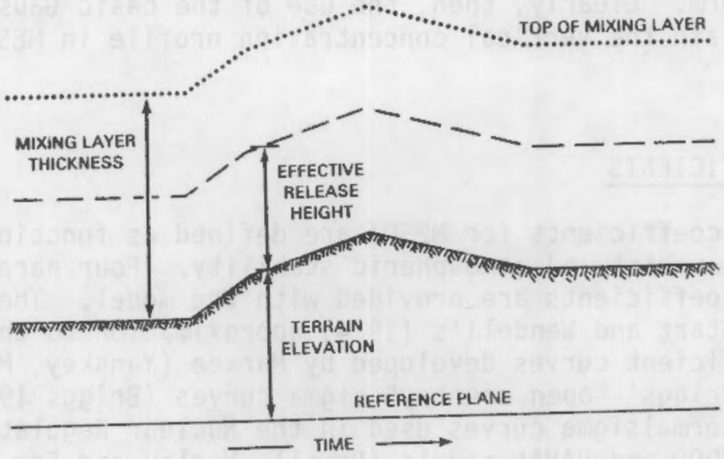


FIGURE 1. Variation of the Effective Release Height and Mixing Layer Thickness at the Puff Center as Puffs Move Over Terrain Features

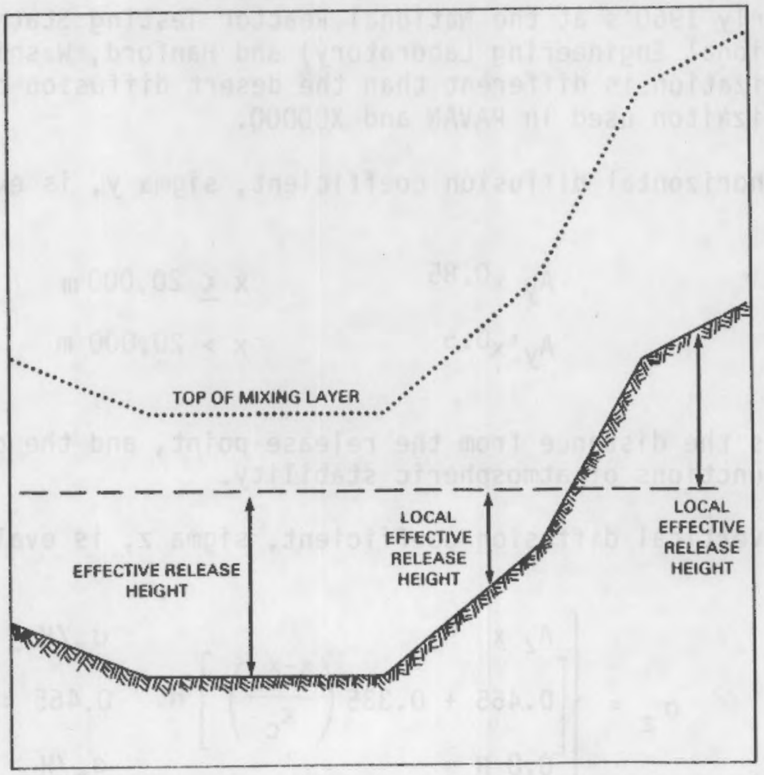

TOP OF MIXING LAYER
EFFECTIVE RELEASE HEIGHT
LOCAL EFFECTIVE RELEASE HEIGHT
LOCAL EFFECTIVE RELEASE HEIGHT

FIGURE 2. Variation of the Local Effective Release Height and Mixing Layer Thickness with Terrain Elevation Away from Puff Centers

mixing layer thickness in magnitude, the vertical concentration profile tends to become uniform. Clearly, then, the use of the basic Gaussian puff model does not constrain the vertical concentration profile in MESOI to be Gaussian.

DIFFUSION COEFFICIENTS

Diffusion coefficients for MESOI are defined as functions of distance from the release point and atmospheric stability. Four parameterizations for the diffusion coefficients are provided with the model. These parameterizations are: 1) Start and Wendell's (1974) approximation to the desert diffusion coefficient curves developed by Markee (Yanskey, Markee and Richter 1966), 2) the Briggs' "open country" sigma curves (Briggs 1973, and Gifford 1976), 3) the normalsigma curves used in the Nuclear Regulatory Commission MESODIF-II, XOQDOQ and PAVAN models (Powell, Wegley and Fox 1979, Sagendorf, Goll and Sandusky 1982, and Bander 1982),and 4) a parameterization recommended for use by the US Army (Ramsdell, Hanna and Cramer 1982).

Parameterizations

The desert diffusion coefficient parameterization is used in the MESODIF and MESOI Version 1 models. It was initially formulated by Start and Wendell (1974) to approximate a set of curves derived from diffusion data collected in the early 1960's at the National Reactor Testing Station (now called the Idaho National Engineering Laboratory) and Hanford, Washington. This parameterization is different than the desert diffusion coefficient parameterizaiton used in PAVAN and XOQDOQ.

The horizontal diffusion coefficient, sigma y, is evaluated from

$$\begin{array}{ll} A_y\, x^{0.85} & x \le 20{,}000 \text{ m} \\ A_y'\, x^{0.5} & x > 20{,}000 \text{ m} \end{array} \qquad (3)$$

where x is the distance from the release point, and the coefficients A_y and A_y' are functions of atmospheric stability.

The vertical diffusion coefficient, sigma z, is evaluated from

$$\sigma_z = \begin{cases} A_z\, x^{B} & \sigma_z/H \le 0.465 \\ \left[0.465 + 0.335\left(\frac{x-x_c}{x_c}\right)\right] H & 0.465 < \sigma_z/H \le 0.8 \\ 0.8\ H & \sigma_z/H > 0.8 \end{cases} \qquad (4)$$

where A_z and B are functions of atmospheric stability, H is the mixing layer depth, and x_c is the distance at which sigma z equals 0.465 H.

Table 1 gives values for A_y, A_y', A_z and B_z as functions of stability class. The content Ay, Ay' and Az are dimensional, having dimensions of $m^{0.15}$, $m^{0.5}$ and m^{1-B}, respectively. Stability classes are defined using a scheme involving season, time of day, wind speed, and cloud cover that is similar to the Pasquill scheme (Pasquill 1961). The NRTS scheme is shown in Table 2. It should be noted that stability class G has been added to the scheme, but that parameter values for Class G have been assumed to be the same as those for Class F.

TABLE 1. Parameter Values for the Desert Diffusion Coefficient Curves

Stability Class	Ay	Ay'	Az	Bz
A	0.718	23.0	0.100	1.033
B	0.425	13.6	0.105	0.975
C	0.349	11.2	0.128	0.891
D	0.267	8.55	0.146	0.824
E	0.299	9.57	0.331	0.567
F	0.401	12.8	0.812	0.307
G	0.401	12.8	0.812	0.307

TABLE 2. Stability Estimates Using Time of Day, Windspeed, and Cloud Cover for the NRTS

Time of Day by Season				Cloud Cover				
November through January	February through April August through October	May through July	Windspeed at 8 m (m/sec)	0 through 5/10	6/10 through 10/10	Heavy Overcast	Insolation	Solar Altitude (degrees)
10 through 12	08 through 11	06 through 09	< 6	B	C	D	Slight	15 through 35
13 through 16	14 through 18	17 through 20	06 through 09	C	C through D	D		
			> 9	D	D	D		
12 through 13	11 through 14	09 through 11	< 6	A	B	C	Moderate	35 through 60
		14 through 17	06 through 09	B	B through C	C		
			09 through 11	C	C	D		
			> 11	D	D	D		
		11 through 14	< 9	A	A through B	B	Strong	> 60
			09 through 11	B	B through C	C		
			> 11	C	C	C		
16 through 10	18 through 08	20 through 06	< 3	G	F	E	None (night)	< 15
			03 through 09	F	E	E		
			09 through 11	E	E	D		
			> 11	D	D	D		

A = Extremely Unstable E = Slightly Stable
B = Moderately Unstable F = Moderately Stable
C = Slightly Unstable G = Extremely Stable
D = Neutral

The "open country" diffusion curves (Briggs 1973) were developed using data from a variety of sources. These are discussed in several locations including Gifford (1976) and Hanna, Briggs and Hosker (1982). The parameterizations are all of the form

$$\sigma = A\, x\, (1 + B\, x)^{C} \tag{5}$$

where A, B and C are functions of stability for sigma z and A is a function of stability for sigma y. For sigma y, B and C have constant values of 0.0001 and -0.5 respectively. Values for A_y, A_z, B_z and C_z are given in Table 3. The values of B have dimensions of m^{-1}.

TABLE 3. Parameter Values for the Briggs' "Open Country" Diffusion Coefficient Curves

Stability Class	Ay	Az	Bz	Cz
A	0.22	0.20	0.0	0.0
B	0.16	0.12	D.0	0.0
C	0.11	0.08	0.0002	-0.5
D	0.08	0.06	0.0015	-0.5
E	0.06	0.03	0.0003	-1.0
F	0.04	0.016	0.0003	-1.0
G	0.027	0.011	0.0003	-1.0

Briggs did not include curves for stability class G. The parameter values given for stability class G in Table 3 were derived assuming that sigma y for G stability is 2/3 of sigma y for F stability. Similarly, sigma z for G stability is assumed to have a value equal to 3/5 of the value of sigma z for F stability. The ratios between the sigmas of F and G stability are those suggested by Regulatory Guide 1.145 (USNRC 1982).

The diffusion coefficient parameterizations used in the NRC MESODIF-II, XOQDOQ and PAVAN models have been attributed to Eimutis and Konicek (1972). In reality, the sigma y parameterization is more properly attributed to Tadmor and Gur (1969) and the sigma z parameterization should be attributed to Martin and Tikvart (1968).

These parametcrizations have the general form

$$\sigma = A\ x^{B} + C \tag{6}$$

where A is a function of stability for both sigma y and sigma z, and B and C are functions of stability for sigma z. For sigma y, B and C are constants having the values of 0.9031 and 0.0, respectively. For sigma z, the values of A, B and C are functions of distance from the source, as well as stability, having different values for three distance ranges. The three distance ranges are: 1)less than 100m, 2)100 to 1000m, and 3) greater than 1000m. Parameter values for A,B and C for the NRC model parameterizations are given in Table 4. The A's have dimensions of m^{1-B} and the C's have dimensions of meters.

TABLE 4. Parameter Values for the NRC Diffusion Curve Formulations

Stability Class	Ay	Az			Bz			Cz		
		Distance Range			Distance Range			Distance Range		
		1	2	3	1	2	3	1	2	3
A	.3658	.192	.00066	.00024	.936	1.941	2.094	0	9.27	-9.6
B	.2751	.156	.0382	.055	.922	1.149	1.098	0	3.3	2
C	.2089	.116	.113	.113	.905	.911	.911	0	0	0
D	.1471	.079	.222	1.26	.881	.725	.516	0	-1.7	-13
E	.1046	.063	.211	6.73	.871	.678	.305	0	-1.3	-34
F	.0722	.053	.086	18.05	.814	.74	.18	0	-.35	-48.6
G	.0481	.032	.052	10.53	.814	.74	.18	0	-.21	-29.2

The final diffusion coefficient parameterization in MESOI is a slight modification of a parameterization recommended to the US Army by Ramsdell, Hanna and Cramer(1982). It is based on the premise that diffusion coefficients should be estimated using wind turbulence data to the extent possible. The basic formulation of the parameterization is

$$\sigma = I \quad F(x) \quad x \tag{7}$$

where I is an intensity of turbulence, F(x) is a function that accounts for the efficiency of the turbulence in diffusing material, and x is the distance from the source. This approach has also been suggested by Hanna et al. (1977), Pasquill (1979), Irwin (1983) and others.

For sigma y, I_y is the ratio of the standard deviation of the cross wind component of turbulence to the mean wind speed. It is also approximately equal to the standard deviation of the wind direction fluctuations when expressed in radians. The intensity of turbulence for sigma z is the ratio of the standard deviation of the vertical wind speed fluctuations to the mean wind speed. It can be approximated by the standard deviation of the wind elevation angle expressed in radians. Table 5 lists turbulence intensities recommended for each stability class in Ramsdell, Hanna and Cramer (1982).

TABLE 5. Turbulence Intensities for Use in Estimating Diffusion Coefficients

Stability Class	I_y	I_z
A	0.37	0.22
B	0.30	0.20
C	0.20	0.15
D	0.15	0.10
E	0.10	0.05
F	0.10	0.05
G	0.10	0.05

The function F(x) that represents the efficiency of the turbulence is non-dimensional. For horizontal diffusion

$$F(x) = \begin{cases} (x/x_r)^{-1/10} & x < 10{,}000 \text{ m} \\ 1 & x \geq 10{,}000 \text{ m} \end{cases} \tag{8}$$

The purpose of x_r is to non-dimensionalize the distance. It is convenient to choose x_r to be a unit distance, i.e. 1 m if x is in meters, 1 km if x is in kilometers, etc. For vertical diffusion, the form chosen for F(x) is

$$F(x) = (1 + b\ x)^c \tag{9}$$

This form is taken from Briggs' parameterization. As a result, the b and c are functions of stability and take on the values given for B_z and C_z in Table 3. It should be noted that b is a dimensional number; specifically, it has the dimensions m^{-1}.

The four diffusion coefficient parameterizations are compared for unstable, neutral and stable atmospheric conditions in Figure 3. The parameterizations give horizontal diffusion coefficient estimates that are about the same for unstable and neutral conditions. This is to be expected because they are based on about the same sets of diffusion data. The departure of the stable desert diffusion coefficient curve from the others is a reflection of the meander of the wind under the low speeds that frequently accompany stable conditions in the desert. Correction factors have been introduced to account for meander during low wind speed conditions when the other parameterizations are used, e.g. in Regulatory Guide 1.145 (USNRC 1982). The data base for horizontal diffusion coefficients extends to distances of a few kilometers with little data for distances greater than 10 kilometers.

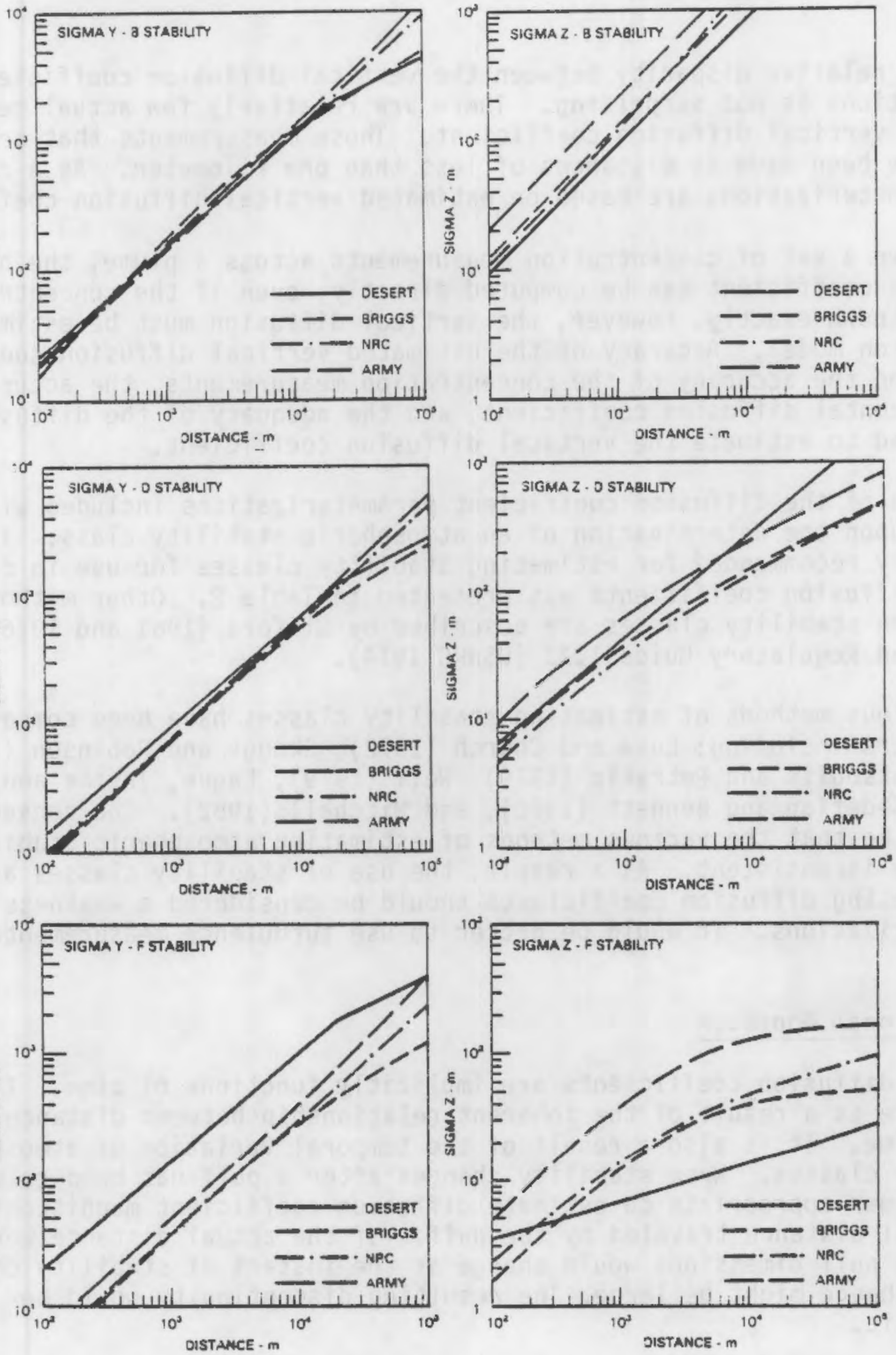


FIGURE 3. Comparison of the MESOI Version 2.0 Diffusion Coefficient Parameterizations

The relative disparity between the vertical diffusion coefficient parameterizations is not surprising. There are relatively few actual measurements of vertical diffusion coefficient. Those measurements that are available have been made at distances of less than one kilometer. As a result, the parameterizations are based on estimated vertical diffusion coefficients.

Given a set of concentration measurements across a plume, the horizontal diffusion coefficient can be computed directly, even if the concentrations are not known exactly. However, the vertical diffusion must be estimated from a diffusion model. Accuracy of the estimated vertical diffusion coefficient depends on the accuracy of the concentration measurements, the accuracy of the horizontal diffusion coefficient, and the adequacy of the diffusion model being used to estimate the vertical diffusion coefficient.

Each of the diffusion coefficient parameterizations included with MESOI depends upon the determination of an atmospheric stability class. The method originally recommended for estimating stability classes for use in computing desert diffusion coefficients was presented in Table 2. Other methods of estimating stability classes are described by Gifford (1961 and 1976), Turner (1964) and Regulatory Guide 1.23 (USNRC 1974).

Various methods of estimating stability classes have been compared in many papers including: Luna and Church (1972), Skaggs and Robinson (1976), Lalas, Catsoulis and Petrakis (1979), Weil (1979), Lague, Irvine and Lavery (1980), Sedefian and Bennett (1980), and Mitchell (1982). The consensus appears to be that the various methods of estimating atmospheric stability are generally inconsistent. As a result, the use of stability classes as means of estimating diffusion coefficients should be considered a weakness in these parameterizations. It would be better to use turbulence measurements directly.

Computational Approach

The diffusion coefficients are implicitly functions of time. Time dependence is a result of the inherent relationship between distance and travel time. It is also a result of the temporal variation of atmospheric stability classes. When stability changes after a puff has been released, it is no longer appropriate to estimate diffusion coefficient magnitudes from the actual distance traveled by the puff. If the actual distance were to be used, the puff dimensions would change at the instant of stability change, and the change might be large. The resulting discontinuity would be unrealistic.

An alternative procedure for computing diffusion coefficient magnitudes in MESOI Version 2.0 makes use of virtual distances. The computational procedure involves three steps, which are the same for each of the parameterizations. The steps are:

1) At the beginning of each sampling interval (the time interval in MESOI time integrations) the distances to virtual point sources, x_v and y_v, are computed from the values of sigma y and sigma z that existed at the end of the previous sampling interval and the atmospheric conditions (stability and mixing layer thickness) that exist in the current advection period (the time interval over which meteorological conditions are assumed to be constant). All of the diffusion coefficient parameterizations included with MESOI can be rearranged to give virtual distances.

2) Effective puff travel distances at the end of the sampling interval, x_e, are computed by addina the distance traveled during the sampling interval to the virtual distances.

3) Diffusion coefficient values (sigma y and sigma z) are computed for the end of the sampling interval using the chosen parameterization and the effective travel distances.

By following these steps, continuity of diffusion coefficients is insured, and accumulation of numerical errors due to round off is minimized.

MESOI applies constraints to the magnitude of sigma z. During normal growth, sigma z is not permitted to exceed 8/10 of the mixing layer thickness' because when sigma z =0.8 H, the puff is uniformly mixed in the vertical. If, as the result of a decreasing thickness, sigma z becomes larger than 8/10 of the thickness, it is not allowed to increase further until the mixing layer thickness increases. The magnitude of sigma z is never decreased.

The procedure outlined above is a departure from the procedure used in earlier versions of MESOI. In those versions, the diffusion coefficients were evaluated using the last value of the diffusion coefficients, the derivatives of the parameterizations evaluated at the virtual distances plus one half of the distance moved in the sampling interval, and the distance moved during the sampling interval. That procedure involved more computations and was conducive to the accumulation of numerical errors.

Figure 4 shows how the procedure appears when superimposed on a standard set of diffusion coefficient curves. The heavy lines show the growth of the sigma value, while the horizontal dashed lines show the changes from the last effective distance, x_e, to the new virtual distance, x_v, are associated with changes in stability. The continuity of the diffusion coefficient is shown in Figure 5. Figure 5 also shows that the rate of change in diffusion coefficients is changed when the stability changes. The rate of growth of sigma is discontinuous, but that discontinuity is acceptable whereas a discontinuity in the sigma vlaue is not.

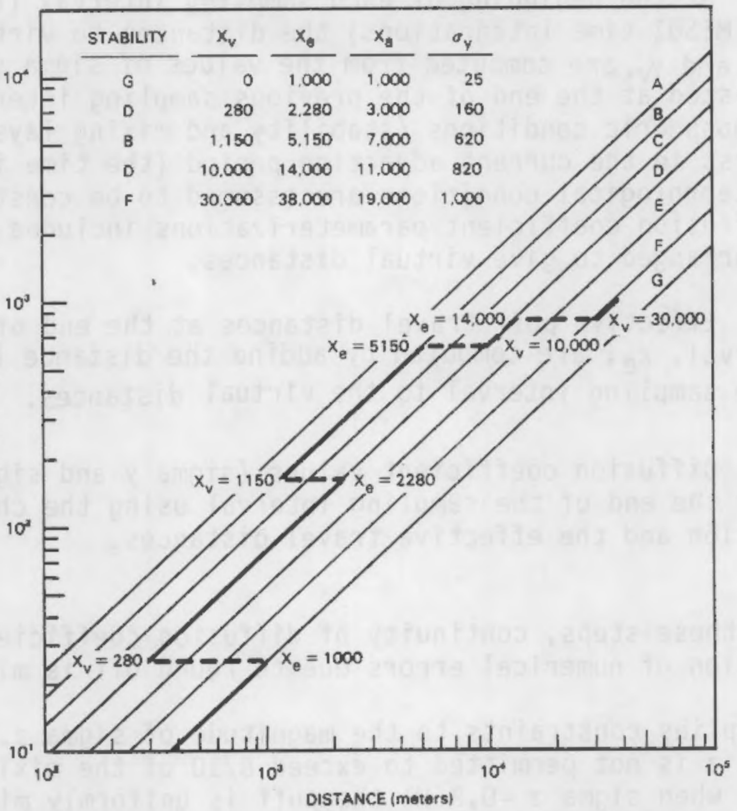


FIGURE 4. Horizontal Diffusion Coefficient Curves Illustrating the MESOI Procedure for Testing Changes in Atmospheric Stability

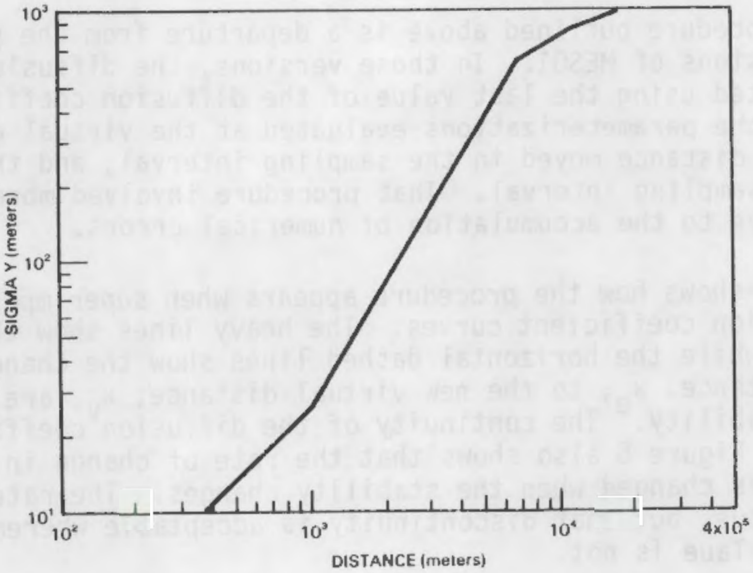


FIGURE 5. Horizontal Diffusion Coefficient Growth Under Conditions of Changing Atmospheric Stability

When the virtual distances are computed in step 1 above, it is unlikely that the same virtual distance will be obtained for both sigma y and sigma z. If the atmospheric stability is decreasing with time, as it generally does in the morning, the virtual distances will be less than the actual travel distance. On the other hand, if the stability is increasing with time, the virtual distances will be greater than the actual distance.

PLUME RISE

MESOI Version 2.0 has been extended to include elevated as well as ground-level releases. The release height used in computations is an effective release height that is the sum of the actual release height above the terrain and puff (plume) rise. If a release is through a stack that has a known effluent flow and temperature, MESOI will compute the effective release height from the stack height, effluent flow and temperature, and the ambient air temperature:

$$h_e = h_s + h_r \qquad (10)$$

where h_s is the stack height and h_r is the puff rise due to bouyancy.

If the release is expected to act as an elevated release, but it is not through a stack, or if stack effluent flow or temperature is unknown, the plume rise estimates in MESOI should not be used. The effective release height should be estimated independently and the estimtes entred directly during model initialization. Conditions that could require separate evaluation of plume rise include venting through safety valves.

Plume rise is estimated in MESOI using the procedures developed by Briggs (Briggs 1969, and Hanna, Briggs and Hosker 1982). Only the final plume rise is estimated, and buoyancy is assumed to be the dominant factor in determining the final rise. Plume rise is not estimated unless the effluent temperature exceeds the ambient air temperature by at least 10°C. The momentum contribution to plume rise is not estimated.

During stable atmospheric conditions the final plume rise is determined by three parameters -- the buoyancy flux, a stability parameter, and the wind speed at release height. The buoyancy flux, F_o is defined as

$$F_o = \frac{g\ V\ (T_s - T_a)}{T_a} \qquad (11)$$

where g is the gravitational acceleration, V is the stack flow, T_s is the effluent temperature, and T_a is the ambient air temperature. Both temperatures must be on an absolute scale. The stability parameter, S, is defined as

$$S = \frac{g}{T_a}\left(\frac{\partial\theta}{\partial z}\right) \simeq \frac{g}{T_a}\left(\frac{\partial T_a}{\partial z} + 0.01\right) \tag{12}$$

where $\partial\theta/\partial z$ is the potential temperature lapse rate in °K/m.

When the wind is calm, the final plume rise is given by

$$h_r = 5.3\, F_o^{1/4}\, S^{-3/8} - 6R_o \tag{13}$$

where R_o is the stack radius. During windy conditions final plume rise is given by

$$h_r = 2.6\, [F_o/(US)]^{1/3} \tag{14}$$

The calm wind equation basically assumes that the puff emanates from a point slightly below the top of the stack. The $-6R_o$ in the equation is a correction factor to account for this lower release point.

The equations that are given for the final rise of plumes in neutral and unstable atmospheric conditions in Hanna, Briggs and Hosker (1982) involve an estimate of the friction velocity U*, which is a function of surface roughness. Rather than attempt to estimate U* from the data that is likely to be available during execution of MESOI, the following approach to estimating plume rise in neutral and unstable conditions has been selected. Near the stack, plume rise is proportional to distance to the 2/3 power and is independent of stability. If final plume rise is assumed to occur at a predetermined distance, the initial plume rise equation can be used to estimate final rise.

The initial rise of buoyancy dominated plumes (Briggs 1969 and Hanna, Briggs and Hosker 1982) is given by

$$h_r\,(x) = 1.6\, F_o^{1/3}\, U^{-1}\, x^{2/3} \tag{15}$$

Briggs (1969) suggests two alternatives for the distance to the final rise. For stacks with a thermal emission of 20 Mw or more, he suggests that the distance be set equal to 10 times the stack height. For stacks with smaller thermal emissions, he suggests estimating the distance using

$$x = 6.49\, F_o^{2/5}\, h_s^{3/5} \tag{16}$$

where 6.49 is a dimensional constant with units of $(\text{sec/m})^{6/5}$.

During light winds, a minimum wind speed of 0.5 m/s is assumed for use with (15).

MESOI places one final constraint on plume rise. Plume rise is not permitted to carry material above the top of the mixing layer at the time of release. If the effective release height including plume rise is greater than the mixing layer thickness, the effective release height is reduced to the mixing layer thickness. The only ways in which the effective release height can exceed the mixing layer thickness are: 1) a stack that extends above the top of the mixing layer, and 2) a decrease in mixing layer thickness with time following release.

REFLECTION TERMS

MESOI Version 2.0 assumes that material is reflected by the surface at the ground unless explicitly deposited, and by the top of the mixing layer once material is within the mixing layer. The method used to incorporate reflection follows directly from the discussion in Sections 2.7 and 6.6 of the text by Csanady (1973).

The initial step in adding the reflection terms is to separate the exponential term in (1) into parts describing the horizontal and vertical decrease in concentration:

$$\exp\left[-\frac{1}{2}\left[\left(\frac{x-x_o}{\sigma_x}\right)^2+\left(\frac{y-y_o}{\sigma_y}\right)^2+\left(\frac{z-z_o}{\sigma_z}\right)^2\right]\right]=\exp\left[-\frac{1}{2}\left(\frac{|\underline{r}|}{\sigma_y}\right)^2\right]\exp\left[-\frac{1}{2}\left(\frac{z-z_o}{\sigma_z}\right)^2\right] \tag{17}$$

where

$$|\underline{r}| = \left[(x-x_o)^2 + (y-y_o)^2\right]^{1/2}$$

The vertical exponential term is then replaced with a sum of exponentials

$$\exp\left[-\frac{1}{2}\left(\frac{z-z_o}{\sigma_z}\right)\right]^2 \approx \sum_{n=-\infty}^{\infty}\left\{\exp\left[-\frac{1}{2}\left(\frac{2nH - h_e' - z}{\sigma_z}\right)^2\right]+\exp\left[\frac{-1}{2}\left(\frac{2nH + h_e' - z}{\sigma_z}\right)^2\right]\right\} \tag{18}$$

where z is the height of the receptor above the ground and H is the mixing layer thickness. As a practical matter, the summation can be truncated after a few terms with n near zero. In MESOI, n has a maximum range of -4 to +4.

Figure 6 shows the general diffusion scenario: a release takes place at a local effective release height h_e' during a period when the mixing layer thickness is H. As vertical diffusion proceeds, material reaches the ground and is reflected. The contribution of the reflected material to the concentration at and above the ground is as if there were a below ground, virtual source at $-h_e$. Similarly, material reaching the top of the mixing layer is reflected and contributes to the concentration within the mixing layer as if it had been released from an elevated virtual source. As diffusion continues, more terms in the sum contribute to the concentration in the mixing layer. It should be noted that H, h_e' and z must be expressed relative to the same reference plane.

Table 6 lists the sources and corresponding numerators for the first ten exponentials in the sum. The pattern in the terms is clearly evident.

TABLE 6. Initial Terms in the Sum of Exponentials Related to the Vertical Concentration Distribution

Source	Numerator
Real	$h_e' - z$
Below ground (b.g.) virtual	$- h_e' - z$
Elevated virtual	$2H - h_e' - z$
1st ground reflection of elevated virtual	$-2H + h_e' - z$
1st elevated reflection of b.g. virtual	$2H + h_e' - z$
1st ground reflection of b.g. virtual	$-2H - h_e' - z$
1st elevated reflection of elevated virtual	$4H - h_e' - z$
2nd ground reflection of elevated virtual	$-4H + h_e' - z$
2nd elevated reflection of b.g. virtual	$4h + h_e' - z$
2nd ground reflection of b.g. virtual	$-4H - h_e' - z$

If the receptor is at ground level, the sum of exponentials becomes:

$$\sum_{n=-\infty}^{\infty}\left\{\exp\ -\frac{1}{2}\left[\left(\frac{2nH - h_e' - z}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{2nH + h_e' - z}{\sigma_z}\right)^2\right]\right\} = 2\sum_{n=-\infty}^{\infty}\exp\left[-\frac{1}{2}\left(\frac{2nH - h_e'}{\sigma_z}\right)^2\right] \qquad (19)$$

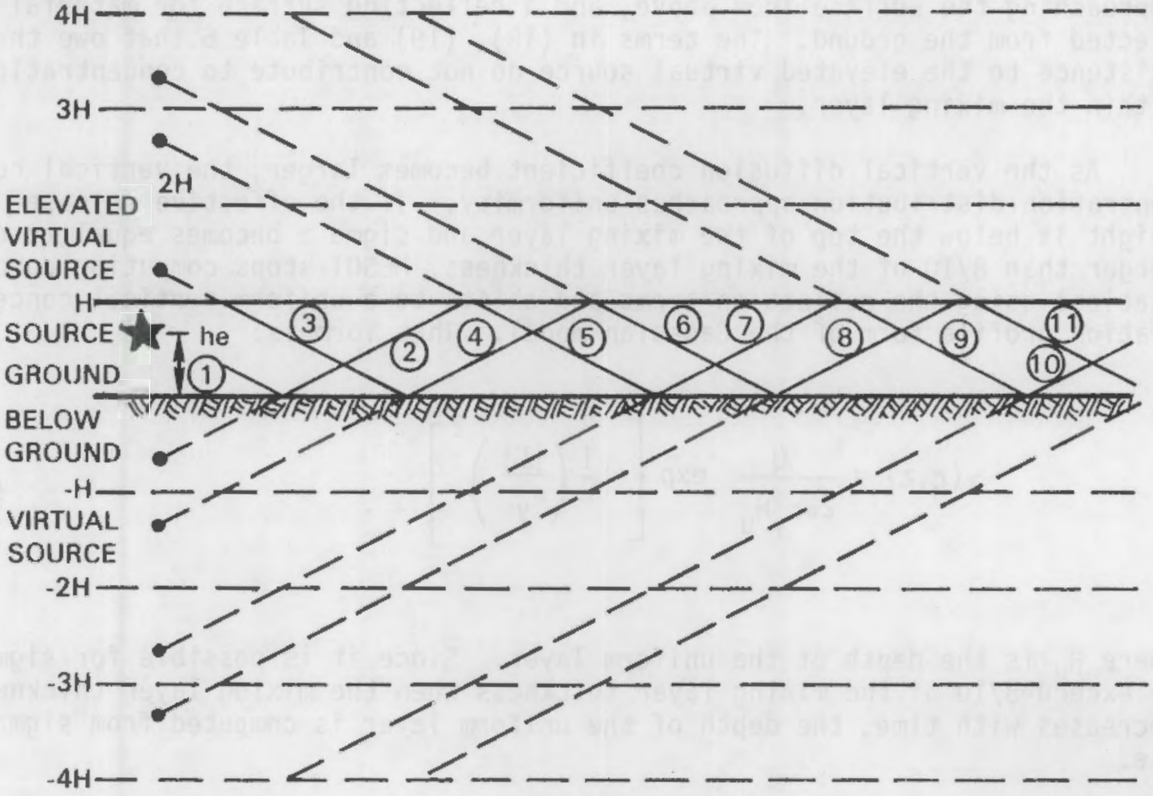


FIGURE 6. Locations of Virtual Sources Assumed by the Reflection Terms in MESOI

When the effective release height is above the atmospheric mixing layer, the elevated virtual source shown in Figure 6 no longer exists. In this case, the top of the mixing layer serves as a permeable surface for material approaching the surface from above, and a reflecting surface for material reflected from the ground. The terms in (18), (19) and Table 6 that owe their existence to the elevated virtual source do not contribute to concentrations within the mixing layer.

As the vertical diffusion coefficient becomes larger, the vertical concentration distribution approaches uniformity. If the effective release height is below the top of the mixing layer and sigma z becomes equal to or larger than 8/10 of the mixing layer thickness, MESOI stops computing concentrations using the reflection terms and shifts to a uniform vertical concentration profile form of the Gaussian model. That form is:

$$\chi(\underline{r},z) = \frac{Q}{2\pi\sigma_y^2 H_u} \exp\left[-\frac{1}{2}\left(\frac{|\underline{r}|}{\sigma_y}\right)^2\right] \qquad (20)$$

where H_u is the depth of the uniform layer. Since it is possible for sigma z to exceed 8/10 of the mixing layer thickness when the mixing layer thickness decreases with time, the depth of the uniform layer is computed from sigma z, i.e.

$$H_u = 1.25\,\sigma_z \qquad (21)$$

This approach was chosen to avoid compressing the puffs vertically when the thickness decreases.

The vertical concentration profile in puffs may also be uniform if the effective release height is above the top of the mixing layer. In this case the test used to determine if a uniform profile should be assumed compares sigma z with the effective release height. If sigma z equals or exceeds 8/10 the effective release height, the uniform profile model is used and the uniform layer depth is computed using (21). Otherwise, the model is used with proper reflection terms.

TERRAIN REPRESENTATION

The difference in the elevation of the terrain directly beneath the center of a puff and at the receptor is used to estimate the local effective release height. In MESOI, terrain elevations are entered explicitly for each point on the receptor grid. These elevations are maintained in a data file that is established when the model is customized for a specific location. The elevations directly beneath puff centers are computed from the elevations of surrounding grid points at the beginning and end of each advection period using a weighted interpolation scheme. In the interpolation, the weight given to the elevation is proportional to the reciprocal of the distance between the grid point and the puff center. The elevations of the puff center at the sampling times within the advection period are estimated by linearly interpolating between elevations at the beginning and times.

A terrain flag has been included in MESOI Version 2 to allow the model to be run without a terrain file. If this option is selected, the model will assume flat terrain.

PUFF DEPLETION

Once released to the atmosphere, material undergoes a number of processes that reduce the amount of the original material that remains suspended in the air. These processes include the deposition of material on surfaces, the washout of material by precipitation, chemical transformation of the material and radioactive decay. MESOI Version 2.0 includes rudimentary treatment of the deposition of material by dry and wet processes to assist in the identification of areas where relatively high levels of surface contamination might be expected to occur following a release. It is expected that actual surface concentration measurements would be made to provide quantitative data as needed. MESOI Version 2.0 also provides for decay of a single released species into a second species and for decay of the second species. The decay/ingrowth/decay scheme is specifically intended for radioactive releases, but it may also be adapted for chemical decay that follows the same decay relationships.

DRY DEPOSITION

A source depletion, dry deposition model is incorporated in MESOI Version 2.0. Source depletion dry deposition models are described in many places in the literature, including: Van der Hoven (1968), Pasquill (1974), and Hanna, Briggs and Hosker (1982). The essence of the dry deposition model is that the flux of material to the surface (mass per unit time per unit area) is assumed to be proportional to the concentration of the material in the air just above the surface. The proportionality constant has the units of speed (length/ time) and is therefore called the deposition velocity. Sehmel (1980) discusses the measurement and uncertainties associated with deposition velocities and summarizes most of the available data on them.

Horst (1977) suggests an alternative to source depletion dry deposition models. His alternative is a surface depletion model that more realistically distributes the depleted mass through the vertical extent of plumes. However, the surface depletion model requires significantly more computational time than the source depletion model. As a result of the uncertainties in dry deposition due to the known but unmodeled variations in deposition velocity that would still exist if the surface depletion model were to be used in MESOI, a source depletion model is used.

In MESOI, the flux of material to the surface is given by

$$\omega_d(\underline{r}) = V_d\chi(r,o) = \frac{2V_dQ'}{(2\pi)^{3/2}\sigma_y^2\sigma_x} \exp\left[-\frac{1}{2}\left(\frac{r}{\sigma_y}\right)^2\right] \sum_{n=-\infty}^{\infty} \exp\left[-\frac{1}{2}\left(\frac{2nH - h_e'}{\sigma_z}\right)^2\right] \qquad (22)$$

where V_d is the deposition velocity and Q' is mass in the puff and centered at x_o, y_o. If mass continuity is to be maintained, the mass remaining in the puff must be decreased by the amount depositing. The rate of change of mass in the puff is

$$\frac{dQ'}{dt} = - \int_0^{2\pi} \int_0^{\infty} |\underline{r}| \omega_d(\underline{r}) d|\underline{r}| d\theta \qquad (23)$$

Evaluation of these integrals must be done numerically when the concentration at ground-level varies in a complex way because the reflection terms are a function of position.

As an alternative to numerical integration, (23) can be simplified by assuming that h_e' can be replaced by h_e, which is not a function of position. In effect, this is equivalent to assuming flat terrain for the puff depletion calculation. The assumption is considered to be an acceptable compromise between accuracy and computational efficiency. When it is made, (23) becomes

$$\frac{dQ'}{dt} = -2\pi\sigma_y^2 V_d \chi(x_o, y_o, 0) \qquad (24)$$

where $\chi(x_o, y_o, 0)$ is the ground-level concentration beneath the center of the puff.

In (23) and (24), it has been assumed implicitly that the deposition velocity is not a function of position. That is not likely to be the case. Sehmel (1980) contains a thorough discussion of the factors that affect the deposition velocity. These factors include: micrometeorological variables such as wind speed, turbulence and relative humidity, surface variables such as type and roughness, and effluent characteristics - such as chemical activity of gases and size and density of particulates. Sehmel's summaries of reported deposition velocities for iodine and sulfur dioxide show ranges exceeding two orders of magnitude. The median deposition velocity for both species is about 0.01 m/s. This value also appears to be a reasonable estimate for many other species in the absence of detailed information on the micrometeorology and surface characteristics. As a result, MESOI uses a constant deposition velocity of 0.01 m/s.

WET DEPOSITION

Dry deposition is a surface layer phenomenon; it is related to air concentration near the surface. Wet deposition, on the other hand, is an integral phenomenon. It is a function of the precipitation as it falls through a puff or plume. In theory (e.g., Engelmann 1968, and Hanna, Briggs and Hosker 1982) there are two fundamentally different processes that lead to wet deposition: in-cloud scavenging, and below-cloud scavenging. MESOI Version 2.0 is not sufficiently sophisticated to be able to distinguish between these processes. Rather it treats them as a single process called washout. Hanna, Briggs and Hosker (1982) discuss some of the assumptions and limitations of this approach. Slinn (1978) also discusses the assumptions and limitations.

The basic assumption in the wet deposition model included in MESOI is that precipitation removes material from puffs in proportion to the rate of precipitation and the local concentration of material within the puff. The wet deposition flux is given by the integral

$$\omega_w(\underline{r},0) = -\lambda_w \int_0^\infty \chi(\underline{r},z)dz = + \frac{\lambda_w Q'}{2\pi\sigma_y^2} \exp\left[-\frac{1}{2}\left(\frac{|\underline{r}|}{\sigma_y}\right)^2\right] \qquad (25)$$

where λ_w is a washout coefficient that is a function of precipitation type and rate.

Using the relationship between qualitative description of precipitation rates (light, moderate and heavy) and quantitative descriptions in millimeters/hour shown in Table 7 and the data in Figures 5.10 and 5.12 of Engelmann (1968), washout coefficients were selected for use in MESOI as a function of precipitation type and rate. The washout coefficients used in MESOI are listed in Table 7.

TABLE 7. Washout Coefficients Used in MESOI Version 2.0 (hr^{-1})

	Precipitation Rate		
Type	Light (< 1 mm/hr)	Moderate	Heavy (>1 cm/hr)
Liquid	0.79	2.2	4.0
Frozen	0.36	1.2	2.3

These values are consistent with the comment in Hanna, Briggs and Hosker (1982) that a typical value of a generic washout coefficient is about 1.0 x 10^{-4} sec $^{-1}$. The values of washout coefficients in Table 7 are also consistent with values that might be derived from the review by Slinn (1978).

Integrating (25) over the horizontal extent of the puffs yields the total change in the airborne mass in the puff

$$\frac{dQ'}{dt} = - \int_{o}^{2\pi} \int_{o}^{\infty} \omega_{w}(\underrightarrow{r},o)|\underrightarrow{r}|d|\underrightarrow{r}|d\theta = -\lambda_{w}Q' \qquad (26)$$

Since the integrations in (26) cover the entire volume occupied by the puff, evaluation of (26) is independent of the actual vertical distribution of concentration within the puff and is therefore exact to the extent that the basic formulation is correct.

The surface contamination rate represented by the flux to the surface in the sum of (22) and (25) is only approximate. It should provide an indication of regions likely to experience high deposition, but it is not likely to give an accurate estimate of the actual surface contamination rate.

DECAY

MESOI Version 2.0 permits three decay scenarios: a stable effluent of species A that doesn't decay, an effluent of species A that decays to stable species B, and an effluent of species A that decays to species B that in turn decays to species C that is of no interest. The decay may be chemical or radioactive as long as the rate of decay of each decaying species is proportional to the current concentration of the species. The scenario in which species A does not decay is trivial. Species A is deposited,if appropriate, otherwise it is treated as a non-depositing species and the book-keeping related to species is bypassed.

If A is a decaying species, it is assumed to decay into species B at a rate which is proportional to the concentration of A. This is the normal decay situation that is described by

$$\frac{dN_A}{dt} = - \lambda_A N_A \qquad (27)$$

where N_A is the amount of species A present and λ_A is a decay constant for A related to A's half life. At a time T following release of an effluent of pure species A, the amount of A remaining is

$$N_A(T) = N_{A_o} \exp(-\lambda_A T). \tag{28}$$

If B is a stable species, the amount of B present at time T is

$$N_A(T) = N_{A_o} \left[1 - \exp(-\lambda_A T)\right]. \tag{29}$$

assuming that there was no B present at T = 0. These relationships follow directly from (27) and are derived in physics and elementary differential equations texts.

When species A decays to B and B decays to C, the rate of change in B is given by

$$\frac{dN_B}{dt} = \lambda_A N_A - \lambda_B N_B \quad . \tag{30}$$

This equation has two different solutions, one for the case in which the decay constants for A and B are the same and the other for the case in which they are different. When the decay constants are equal, the amount of B at time T is given by

$$N_B(T) = N_{A_o} \exp(-\lambda T). \tag{31}$$

when the decay constants are unequal, the solution is

$$N_B(T) = N_{A_o} \left(\frac{\lambda_A}{\lambda_A - \lambda_B}\right) [\exp(-\lambda_A T) - \exp(-\lambda_B T)]. \tag{32}$$

MESOI computes the decay constants for both the parent and daughter species from half lives supplied during model initialization.

PUFF TRANSPORT

In MESOI Version 2.0, puffs of diffusing material are transported with the wind at the effective release height of the puff. This wind is determined from a three dimensional field of the horizontal components of the wind. The wind field consists of three layers that are defined as function of time. The three layers are: a surface layer, a geostrophic or gradient wind layer, and an intermediate interpolated layer. The representation of the wind in each of the layers is described below.

SURFACE LAYER WINDS

Surface layer winds are defined at the nodes of a 16 x 16 grid that covers the spatial domain of the model by interpolation and extrapolation of available surface wind data. MESOI requires wind data from at least one source, and it can accept data from as many as 30 measurement locations. The position of each measurement location must be given in kilometers north or south and east or west of the center of the model domain.

Interpolation and extrapolation of wind data to the nodes is done using weighted averages of the winds at the measurement locations closest to each node. Winds from as many as ten measurement locations may be included in the average for a node. The weight given to each wind included in an average is proportional to the inverse of the square of the distance between the node and the measurement location. Wendell (1972) describes the interpolation scheme in detail. Goodin, McRae and Seinfeld (1979) indicate that inverse square weighting provides a good compromise between accuracy and computational costs.

In earlier versions of MESOI, the wind data were assumed to reflect the effects of terrain. In this way the effects of terrain were implicitly represented in the wind field. This approach may be adequate if wind data are available for a large number of locations, as is the case at Hanford. On the other hand, if the wind data are limited or the terrain is complex, the approach is likely to be inadequate. A module has been incorporated in MESOI version 2.0 to permit the user to modify the wind field for terrain effects that may not be represented in the available wind data. The module uses a simple, deterministic procedure to automatically adjust the winds at selected locations rather than requiring manual adjustment of each wind field by the user.

To make use of the adjustment feature, it is necessary to define the orientation of the face of the terrain at each node where an adjustment is to be made and to provide two adjustment factors for winds at the node. The two adjustment factors permit different adjustments to be for winds approaching the terrain feature from different directions e.g., downslope and upslope flow. This information is entered into a data file when MESOI is customized for a specific site.

Selection of adjustment factors is a subjective matter. Intuitively, they ought to be related to the slope of the face. However, no quantitative relationship has been developed. It is likely that initial estimates of adjustment factor values will be made on the basis of terrain slope, and that refinements will be made in the estimates to achieve a desired modification of the wind field. There is no substitute for the exercise of sound meteorological judgement in the selection of adjustment factors.

In determining which terrain features should be represented in the surface wind field adjustment module, both horizontal and vertical dimensions of the features should be considered. If a feature has typical horizontal dimensions that are small compared to the model grid spacing, the feature will be difficult to represent in the model. Similarly, features that do not extend to likely effective release heights may not significantly affect puff transport.

The adjustment of the wind at a node is done in the following manner. The wind at the node following the initial interpolation is resolved into components perpendicular and parallel to the face of the feature. The magnitude of the perpendicular component is multiplied by the appropriate adjustment factor, and the parallel component is adjusted to preserve the magnitude of the initial wind vector. Finally, the adjusted wind is resolved into the north-south and east-west components used for computing puff transport.

TOP LAYER WINDS

The top layer in the wind field represents the wind above the surface boundary layer and is applied to puffs with effective release heights above the top of the mixing layer. A single wind is used. Therefore, it is essential to select the wind that is most representative over the entire model domain. Two candidates for the top layer wind are the geostrophic and gradient winds. It is not necessary to measure these winds directly. They can be estimated from isobars on surface pressure charts or they can be computed from sea level pressures. If upper level wind measurements are made in the area where MESOI is to be applied, they, too, could be used.

No adjustments are made to the wind in the top layer.

INTERMEDIATE LAYER WINDS

In MESOI, the surface layer winds represent the winds at 10 m and below, the top layer winds represent the winds above the top of the mixing layer. The winds in the layer between 10 m and the top of the mixing layer are the intermediate layer winds. They are determined by interpolation in the vertical between the surface and top layer winds. As a result, they are computed only when needed and are not stored by MESOI.

A power law interpolation scheme is used to estimate the wind at the effective release height from the surface layer wind beneath the center of a puff and the top layer wind. The interpolation is performed independently for each wind component using the formula:

$$U(h_e) = U(10m) + \left[U(H) - U(10m)\right]\left(\frac{h_e - 10m}{H - 10m}\right)^{\alpha} \tag{33}$$

where U is one of the component speeds and α is an interpolation parameter that may be set equal to 1.0 for linear interpolation, or it may be treated as a function of stability. In addition, the interpolation parameter may be allowed to take on a different value for each of the components.

Analysis of wind profile data from Hanford, Washington indicated that the interpolation parameter does have some stability dependence and that the parameters are different for the two wind components under neutral and stable atmospheric conditions. However, the range of parameter values determined during any given set of conditions was relatively large and median values tended to be between 0.5 and 1.5. As a result, use of an interpolation parameter value of 1.0 for both components is reasonable in the absence of data on which to base a different choice. MESOI currently sets $\alpha = 1.0$ regardless ot atmospheric conditions or model options selected.

PUFF MOVEMENT

Two wind fields are retained in MESOI at all times. These are: the wind field for the beginning of the current simulation period, and the wind field for the beginning of the next simulation period. In general simulation periods are an hour long, but if wind data are updated more frequently, the periods may be shorter. The wind used for puff advection is an average based on the position of the center of the puff, vertical interpolation to the effective release height of the puff, and linear interpolation in time between the two wind fields. Within an advection period, the puff is assumed to move with a constant speed and direction.

MESOI OUTPUT

Following the interactive exchanges during the initialization of MESOI Version 2.0, model output occurs in three categories: meteorological data, puff positions, and time-integrated air concentrations and surface contamination.

METEROLOGICAL DATA

The meteorological data used for each hour of the simulation are written to a disk file as part of the simulation record.

If desired, two wind plots may be obtained for each hour. The first is a plot of the observed surface wind vectors at each of the wind stations, and the other is a plot of the interpolated surface winds at each of the grid points. The interpolated wind field that is output is the wind field after adjustment for terrain effects. In both cases, the vectors have been rotated by 180 degrees so they indicate the direction of effluent transport.

PUFF POSITION

MESOI Version 2.0 contains a routine to generate a plot file each hour that shows the position and size of each puff. When the puffs are plotted on a plotter or graphics video display terminal, they are shown as circles with centers at the proper grid positions and radii that indicate the distance to a predetermined, normalized air concentration, set at 10^{-15} hr/m^3. Plotting puffs using radii based on a low concentration limit results in puff dimensions that are larger than might be expected near the source, particularly if the expectation is that puff radii represent ± 3 sigma y. In the MESOI puff plotting routine, initial puff radii are many sigma y when concentrations at puff centers are high. As a puff moves, its radius will increase in magnitude, but the increase in radius will not be as rapid as the increase in sigma y. The apparent rate of growth in puff radius will decrease as sigma y approaches the radius in magnitude. Eventually, the decreasing concentration in the puff may cause the puff radius used in plotting to decrease, even though sigma y continues to increase.

A second set of routines for indicating puff positions is available in MESOI Version 2.0. This set of routines is related to specific receptors of interest. As many as 35 receptors of specific interest can be defined. Each receptor is called a checkpoint and is assigned to the closest grid point for which time-integrated concentrations are computed. Two time-integrated concentration thresholds are set. When the time-integrated concentration at a checkpoint rises above the lower threshold, MESOI prints a message giving the checkpoint name, the time, and threshold that has been exceeded. A similar message is printed when the upper threshold is exceeded. At the end of the simulation, a summary containing the status of each checkpoint is given.

TIME-INTEGRATED CONCENTRATIONS AND SURFACE CONTAMINATION

The primary output of MESOI is a set of nine time-integrated concentration and surface contamination matrices. Five of the matrices contain time-integrated concentrations at nodes of a 31 x 31 grid, and four of the matrices contain surface contamination at the nodes. The spacing between nodes on these matrices is one half of the spacing on the wind field grid.

Ground level, time-integrated air concentrations, E(x, y, 0), frequently referred to in meteorological literature as exposures, are initially computed for a non-depositing, non-decaying effluent using

$$E(x,\ y,\ 0) = \sum_{i} \sum_{j} X_{ij}(x,y,o)\Delta T_{i} \tag{34}$$

where the concentration is defined by (19) or (20), as appropriate, and ΔT_i is the sampling interval. The summation over time (index i) in (34) starts at the beginning of the simulation and continues until the simulation is terminated or no active puffs remain on the computational grid. The other summation in (34) (index j) is over all puffs affecting the node. In Version 2.0 of MESOI, time-integrated concentrations are expressed in units of mass-hours/meters^3.

The normalized time-integrated air concentrations computed for a non-depositing, non-decaying effluent are larger than those computed for an effluent that either deposits or decays. As a result, these air concentrations are used for determination of puff arrival times at checkpoints.

For decaying and depositing effluents, MESOI Version 2.0 maintains more than one source term for each puff. The primary source term is the initial mass contained in the puff. If the mass is depleted by deposition or decay, a depleted source term is maintained for the primary species. It is initially determined by applying corrections from (24), (26) and (27) to the undepleted source term. After the first sampling interval, the corrections are applied to the depleted source term, i.e.

$$Q'_A(t) = Q'_A(t-1) + \left\{ \left.\frac{dQ'_A(t-1)}{dt}\right|_{dry} + \left.\frac{dQ'_A(t-1)}{dt}\right|_{wet} + \left.\frac{dQ'_A(t-1)}{dt}\right|_{decay} \right. \Delta T \tag{35}$$

where the A subscript refers to the initial species, $Q_A'(t=0)$ equals the initial undepleted source strength.

When species A decays to a second species B, it is assumed that B does not exist at the release time and the ingrowth of B into a third source term is governed by (30). If B decays, the magnitude of $Q_B(t)$ is controlled by (31) or (32), which ever is appropriate.

Time-integrated, ground-level air concentrations of A and B, E_A and E_B respectively, are computed from the undepleted air concentrations by multiplying them by the ratios between the depleted and undepleted source terms and summing over all puffs.

$$E_A(x, y, 0) = \sum_i \sum_j \left[\left(\frac{Q'_{A_{ij}}}{Q} \right) \chi_{ij}(x, y, 0)\ \Delta T_i \right] \tag{36}$$

and

$$E_B(x, y, 0) = \sum_i \sum_j \left[\left(\frac{Q'_{B_{ij}}}{Q} \right) \chi_{ij}(x, y, 0)\ \Delta T_i \right] \tag{37}$$

Two sets of depleted ground-level air exposures are computed. The first set contains the exposures accumulated during the advection period. For this set of exposures the time summations in (36) and (37) go from 1 to the number of sampling intervals in an advection period for the specific puff. These exposures may be used to estimate short-term average concentrations. The second set of exposures contains the cumulative exposures from the beginning of the simulation. The short-term exposures are computed directly; the second set contains the running total of the short-term exposures. Once the contribution of the exposures during an advection period has been added to the long-term exposures, the short-term exposures are reset to zero.

Surface contamination, Sc(x, y), at a grid node, is computed directly from the dry and wet deposition fluxes defined in (22) and (25)

$$Sc(x, y) = \sum_i \sum_j \left[\left(\omega_{d_{ij}} + \omega_{w_{ij}} \right) \Delta T_i \right] \quad (38)$$

Again, MESOI Version 2.0 maintains two sets of surface concentrations. The first is for the contamination accumulated during the advection period, and the second is for the long-term accumulation. The method of accumulation of surface concentration is the same as for the time-integrated air concentrations. The units for the normalized surface contamination are mass per square meter.

Both sets of surface concentrations are depleted to account for decay, if appropriate. The decay of species A results in an increase in B, which may also decay.

The sequence of computation in MESOI is as follows:

1) compute the contribution to the undepleted time-integrated ground-level air concentration,

2) compute the contribution to the depleted time-integrated ground-level air concentrations,

3) compute the contribution of dry deposition to the surface contamination,

4) compute the contribution of wet deposition to the surface contamination,

5) compute decay and ingrowth of the material in the puffs,

6) deplete the puffs due to dry deposition, and

7) deplete the puffs due to wet deposition.

At the end of each advection period, the long-term cumulative normalized, ground-level time-integrated air and surface concentrations are corrected for decay and the contributions of the most recently completed advection period are added to the long-term time-integrated air and surface concentrations.

MESOI LIMITATIONS

MESOI Version 2.0 is a tool for use in evaluating the transport, diffusion and deposition of material released to the atmosphere. As with other tools, it has limitations. MESOI's limitations arise from three general areas: the mathematical models used to approximate complex physical processes, the data used by the models, and numerical compromises made to keep MESOI's execution time and memory requirements within acceptable limits. Proper use of MESOI depends on a clear understanding of its limitations and their implications.

MATHEMATICAL MODEL LIMITATIONS

The atmospheric processes represented in MESOI are complex. In some cases, such as decay, it is possible to represent processes exactly. But in most cases, the complete representation of the processes is not possible or would require more information than is likely to be available for use by the model. As a result, most of the processes are represented in MESOI by simple mathematical analogs. Among the more important of these analogs are: the Gaussian puff approximation of plume behavior, the empirical adjustment of the surface wind field to account for the effects of terrain, the vertical interpolation of the wind, and the estimation of plume rise.

The limitations imposed by these analogs are fundamental. If they are not acceptable, the only alternative is to replace the analog. An attempt has been made to structure the MESOI computer code to facilitate changes in the manner in which each of the processes is modeled. The modular structure of MESOI also facilitates the addition of features to represent atmospheric processes not currently depicted in the model. Features that might be added include the representation of the effects of wind shear on diffusion and better resolution near sources.

DATA LIMITATIONS

The limitations imposed by data fall into three categories: uncertainties related to the source term, inadequacies and uncertainties in the data used to develop the parameterizations in various parts of MESOI, and inadequacies in the meteorological data used by MESOI.

Of all of the information required for use of MESOI in an emergency response situation, the most likely to be undefined, or at least poorly defined, is the source term. Initial source term uncertainty may be orders of magnitude. In this event, MESOI may be run by assuming a unit release rate, and the output can be treated as if normalized to release rate. Normalized output only provides estimates of plume positions and relative concentrations. When source term estimates become available, the actual time-integrated air concentrations and surface concentrations at points of interest can be estimated by multiplying the normalized values by the source term.

The parameterizations for the diffusion coefficients, deposition velocity and washout coefficients limit the accuracy that can be expected in the MESOI output. Given detailed meteorological data and uncomplicated terrain, the accuracy of MESOI in estimating concentrations within a plume is likely to be no better than the accuracy of a Gaussian continuous plume model. The primary advantage of MESOI over the straight-line Gaussian plume approach is a more realistic treatement of effluent transport.

A realistic upper limit to the accuracy of short-term (a few hours or less) air concentrations for non-depositing, non-decaying effluents is, then, about plus or minus a factor of two of the actual value (American Meterological Society 1978). As the duration of the release increases, the accuracy of concentration estimates will tend to improve.

In areas of complex terrain and/or changing meteorological conditions, the accuracy will be less. The use of stability classification schemes to estimate diffusion conditions also degrades the accuracy of diffusion estimates.

Estimates of air concentrations at specific points in space, as opposed to points relative to the center of the plume or puff, are less accurate than those quoted above. This additional decrease in accuracy results from potential errors in the position of the plume or puff.

The accuracy of estimates of surface concentration is limited by the accuracy of the deposition velocity and washout coefficients as well as the estimates of air concentrations. Neither the deposition velocities nor washout coefficients are better than order of magnitude estimates (see Sehmel 1980 and Slinn 1978). As a result, surface concentrations estimated by MESOI should only be used to gain a "feeling" of the areas where surface contamination is likely, and potential variations of contamination within those areas.

In order to realize the full potential of MESOI, it is necessary to provide the model with adequate meteorological data. As the model is currently configured, wind data can be accepted from as many as 30 measurement locations. The remainder of the meteorological conditions are specified independent of position within the model domain. It is assumed that they are most representative of conditions near the release point because that is where they have the greatest effect on model output.

Given the typical availability of wind data in the vicinity of a potential release point, increasing the number of wind measurement locations is likely to be the action that can be taken by the model user that will give the greatest increase in model accuracy. If there are sufficient wind observations, particularly in the vicinity of topographic features, the observed winds will define the effects of the features, the need for empirical adjustments will be eliminated, and the errors in the spatial interpolation will be reduced.

In the current version of MESOI wind directions are entered to the nearest ten degrees and wind speeds are entered to the nearest meter per second or mile per hour. As a result, the direction and rate of movement of a low level puff located near a wind station may be off by as much as five degrees and one half meter per second or mile per hour respectively. When a puff is not located near a wind station, the errors in its movement are likely to be larger because of the errors inherent in the horizontal interpolation scheme. When a puff is elevated, the errors in estimates of puff movement will be larger than those for low level puffs. Similarly, the winds used in estimating puff movement at an observation time are less likely to be in error than are the winds that result from interpolation between observation times. Individually these errors may be small, and they may be sufficiently random that they tend to average out. However, their net effect is to produce uncertainty in MESOI puff positions and concentrations.

If the effective release height is not well defined, the uncertainties in puff position become even larger. In this case, it may be useful to estimate the range in uncertainty of puff positions by running MESOI two or three times, varying the release height between runs. Effective release heights used might be the lowest likely, most likely and highest likely heights.

NUMERICAL LIMITATIONS

Numerically, MESOI is limited by: the resolution of the grids on which the time-integrated air concentrations and surface concentrations are accumulated, the number of puffs released per hour, and the minimum interval used in performing the time-integrations.

The resolution of the grid used for the wind fields is currently set at 5 km, and the resolution of the grids on which the concentrations are accumulated is 2.5 km. These can be changed, but the concentration grid will always be twice that of the wind grid.

The practical limit to the resolution obtainable in MESOI is determined by the minimum interval used in the time-integration. The interval used in the integration is referred to as the sampling interval. It varies as a function of the product of wind speed and puff size, but has a minimum value of one minute. When the puff is small and the wind speed is large, the puff movement during one minute will be large in comparison with the puff dimension. As a result, the number of samples used to estimate the time-integrated concentrations at a point in space affected by the puff will be small, and the concentrations will depend on the positions relative to the puff center used in the estimates.

Figure 7 illustrates the dependence of the accuracy of time-integration on the ratio between the incremental distance moved and sigma y for a Gaussian puff. The distance moved is the product of the wind speed and sampling interval. The accuracy of the estimate of the time integral is indicated in the figure by the ratio between the estimates based increasingly large incremental distances and the true value of the integral. As the incremental distance becomes larger than sigma y, a range of estimates is evident. This range is indicated by the two lines in the figure. Some of the estimates are smaller than the true value, while others are larger.

Applying Figure 7 to MESOI, the maximum resolution of the model depends upon the accuracy range acceptable for the concentration estimates. If time-integrated concentration estimates must be accurate to within about 20%, the maximum incremental distance moved in a sampling interval must be somewhat less than three sigma y. However, because of the minimum sampling interval of ore minute, the limiting factor is the size of sigma y. In a 10 m/s wind, sigma y must be greater than 200 m before the desired accuracy is achieved. Assuming that high wind speeds are generally associated with neutral stability, Figure 3 indicates that sigma y does not reach 200 m until the distance from the source is about 3 km.

The fact that the MESOI time integration is acceptably accurate at a distance does not mean that MESOI will provide a usable definition of the ground-level concentration patterns at that distance, particularly for short duration releases. Useful definition of concentration patterns occurs at those distances where sigma y approaches the distance between nodes. In the example above, a sigma y of 200 m is required for acceptable accuracy in integration, but at a grid spacing of 3 km it is likely that the high concentration regions of a puff with a 200 m sigma y will pass between nodes rather than impinge on one. It may be reasonable to assume that once sigma y reaches one half the grid spacing that the concentration patterns can be defined. In general, this occurs at distances of 10 km or more from the source.

Both the minimum sampling interval and grid spacing can be reduced to improve model resolution, however altering these values can have a significant impact on the computer memory required by the model and the time required for model execution. The time required for model execution is inversely proportional to the sampling interval. While it is not likely that decreasing the minimum sampling interval by a factor of two would double the execution time of the model, it could substantially increase the run time for short duration releases where much of the execution time is associated with puffs near the source.

The number of puffs released per hour, or more specifically, the time between puff releases, determines how well the puff model can simulate a plume when the wind direction changes rapidly. The initial distance between puffs is equal to the product of the wind speed and the time between releases, which is referred to as the advection period. Until the magnitude of sigma y approaches the separation between puffs, the concentration gra-

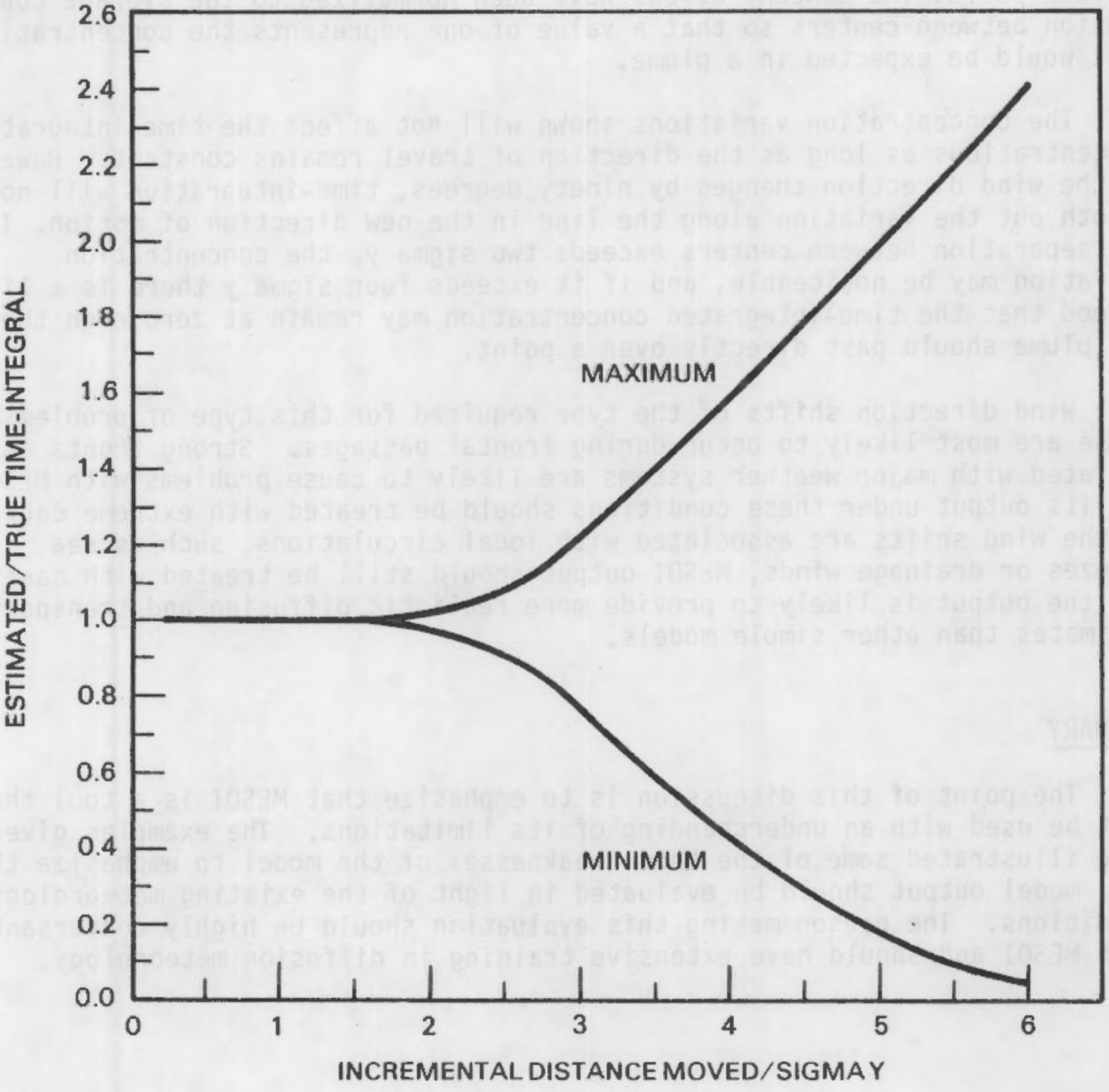


FIGURE 7. Sensitivity of the Accuracy of Time-Integrated Concentration Estimates to Variations in the Ratio Between the Distance Moved in a Sampling Interval and a Puff Sigma y

dient in the direction of puff travel will show an unrealistic oscillation. This oscillation is illustrated in Figure 8. In Figure 8, the variation of concentration between two puffs of equal magnitude is shown as a ratio between the distance from the center of one of the puffs and the separation between puffs. The concentrations have been normalized to the average concentration between centers so that a value of one represents the concentration that would be expected in a plume.

The concentration variations shown will not affect the time-integrated concentrations as long as the direction of travel remains constant. However, if the wind direction changes by ninety degrees, time-integration will not smooth out the variation along the line in the new direction of motion. If the separation between centers exceeds two sigma y, the concentration variation may be noticeable, and if it exceeds four sigma y there is a likelihood that the time-integrated concentration may remain at zero even though the plume should pass directly over a point.

Wind direction shifts of the type required for this type of problem to arise are most likely to occur during frontal passages. Strong fronts associated with major weather systems are likely to cause problems with MESOI, and its output under these conditions should be treated with extreme caution. If the wind shifts are associated with local circulations, such as sea breezes or drainage winds, MESOI output should still be treated with caution, but the output is likely to provide more realistic diffusion and transport estimates than other simple models.

SUMMARY

The point of this discussion is to emphasize that MESOI is a tool that must be used with an understanding of its limitations. The examples given have illustrated some of the known weaknesses of the model to emphasize that the model output should be evaluated in light of the existing meteorological conditions. The person making this evaluation should be highly conversant with MESOI and should have extensive training in diffusion meteorology.

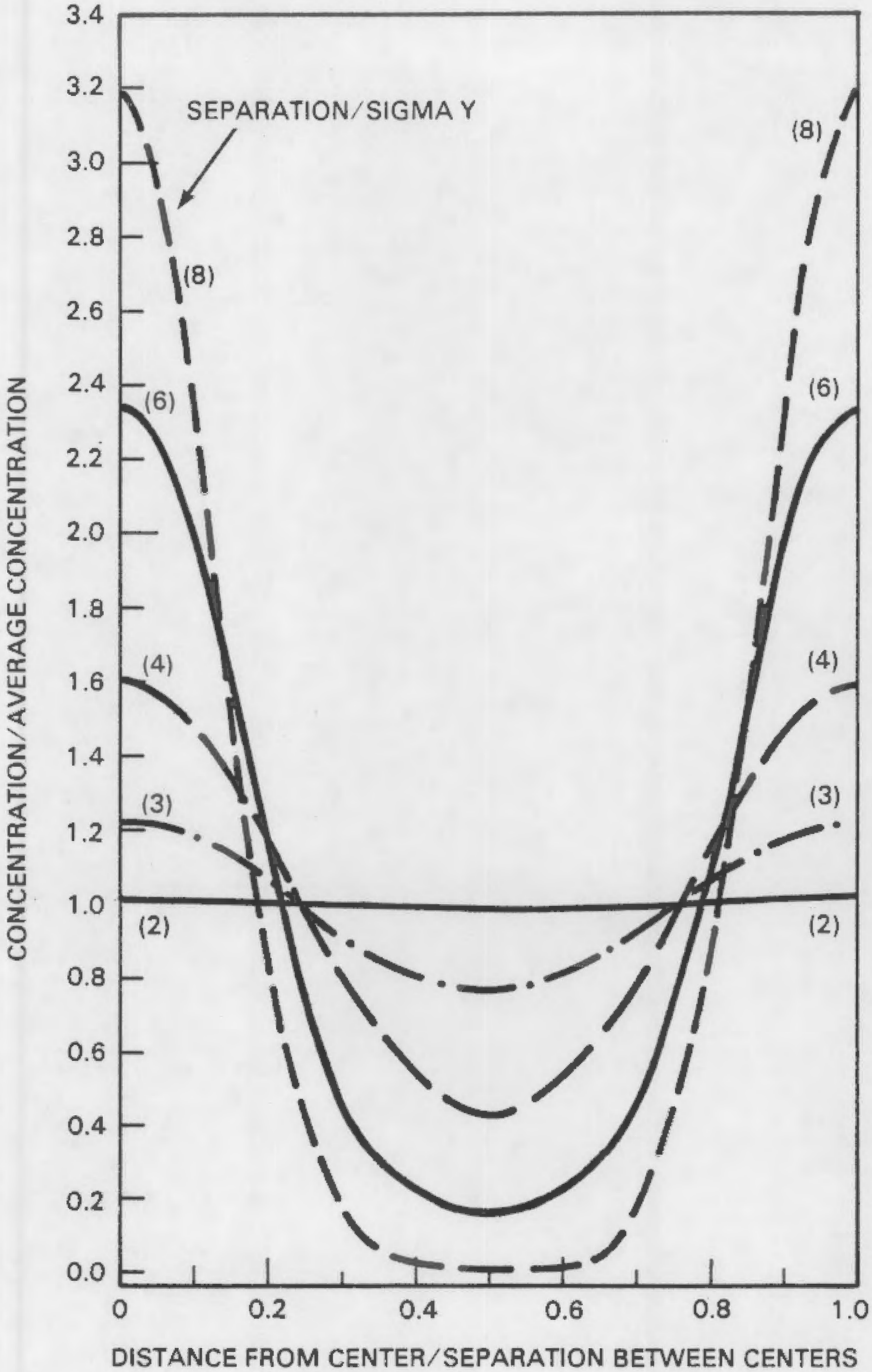


FIGURE 8. Variation in Concentration between Puffs as a Function of the Ratio of the Distance from the Center of a Puff to the Distance Between Adjacent Puff Centers

PART II. THE MESOI VERSION 2.0 COMPUTER PROGRAM

PROGRAM ORGANIZATION

MESOI Version 2.0 is a highly modularized, interactive, Lagrangian puff, atmospheric transport diffusion and deposition model. It consists of a main program, called MESOI, and 41 subroutines. The 41 subroutines can be divided into four groups: 1) model initialization, 2) data input and output other than initialization, 3) transport, diffusion and deposition, and 4) general utility. Data transfer between program elements is generally done through 12 labeled common blocks that contain related variables and constants.

User interaction, through a video display terminal, provides data for model initialization and controls the output of MESOI graphics products at the end of each hour of a simulation. Meteorological data required by the model are input through data files that are generated prior to running the model. Primary model output is directed to disk files for later disposition to printers and plotters, as appropriate. Limited alpha-numeric and graphical output is provided for alpha-numeric terminals, and all of the graphics products prepared for a plotter can be viewed on a graphics display terminal.

MESOI and its subroutines are written in standard FORTRAN 77 with a limited number of extensions. The most common extension is the use of INCLUDE files (files having names xxx.INC) to incorporate the COMMON block and the other specification statements associated with the variables in the COMMON block into the program elements. This extension may be eliminated by substituting the proper COMMON and associated type-statements in each element where the INCLUDE statement occurs. The other non-standard aspects of MESOI are generally related to communication between the computer and terminals. The program elements in MESOI that generate graphics products make use of CALCOMP commands. If other plotting devices are to be used with MESOI, it may be necessary to change the commands.

The inter-relationships among the program elements and the COMMON blocks are shown in Table 8. The subroutines are listed under program elements in the order in which they are called. Indentation shows the layering of the subroutines.

The total length of the MESOI Version 2.0 computer code is about 155,000 bytes. The actual memory requirement depends upon which of the diffusion coefficient parameterizations is used. Distribution of the memory allocation is approximately: 36,000 bytes for executable code, 24,000 bytes for local data storage, and 95,000 bytes for storage of data in COMMON blocks.

The next six sections in Part II describe MESOI, the subroutines in each of the four groups, and the COMMON blocks. The major variables used in MESOI are defined in the Appendix. The last section in this part discusses customization of the MESOI code.

TABLE 8. Inter-Relationships Between Program Elements and Common Blocks in MESOI Version 2.0

PROGRAM	COMMON BLOCKS											
ELEMENT	ARRIV	CHAR	CONST	DATIM	DECAY	MATRIX	PUFFS	REL	STATN	TOPOGR	UNITS	WINDS
MESOI	XXX	XXX	XXX	XXX	XXX	XXX	XXX	XXX		XXX	XXX	XXX
FILOPN											XXX	
GRIDIN		XXX							XXX		XXX	
REARNG		XXX							XXX			
STRAY		XXX							XXX			XXX
ASCND												
TOPFIL										XXX	XXX	
INIT		XXX	XXX	XXX		XXX	XXX	XXX			XXX	XXX
JULIAN												
LOCATE											XXX	
DATRD				XXX							XXX	
DATWR				XXX							XXX	
WIND		XXX		XXX					XXX		XXX	XXX
SHIFT				XXX								XXX
TERRA		XXX									XXX	XXX
DIRSPD												
ARRIVN	XXX										XXX	
RELEAS			XXX		XXX			XXX			XXX	
JULIAN												
RDKIN					XXX						XXX	
LOCATE											XXX	
DATRD				XXX							XXX	
DATWR				XXX							XXX	
WIND		XXX		XXX					XXX		XXX	XXX
TERRA		XXX									XXX	XXX
DIRSPD												
ALPHA												
WINSPD			XXX					XXX				XXX
PLMRIZ		XXX		XXX				XXX			XXX	
PUFFR			XXX		XXX		XXX	XXX				
PUFFM			XXX	XXX			XXX					XXX
DTOPO										XXX		
DIFDEP		XXX	XXX	XXX	XXX	XXX	XXX	XXX		XXX		
RDK					XXX						XXX	
SIGMA												
ARRTIM	XXX		XXX	XXX		XXX					XXX	
PRINTE		XXX				XXX					XXX	
TESTM							XXX				XXX	
SHIFT				XXX								XXX
CLEAN							XXX				XXX	
SCREEN		XXX		XXX		XXX		XXX			XXX	
PLOTZ		XXX		XXX			XXX	XXX			XXX	
DTCHAR												
WNDFLD		XXX		XXX					XXX		XXX	XXX
DIRPSD												
DTCHAR												
WNDPLT		XXX		XXX					XXX		XXX	XXX
DTCHAR												
EXPSUM	XXX					XXX					XXX	
MODEND											XXX	

MESOI, THE MAIN PROGRAM

MESOI is the main program. It controls model execution by calling the subroutines in the proper sequence. As can be seen in Table 8, MESOI calls 29 of the model's subroutines directly. It also has direct access to 11 of the model's common blocks.

MESOI execution starts by printing a sign-on message on the user's interactive terminal. When instructed to continue by the operator, it automatically opens the files needed for data input and the output of model results.

Once the necessary files are open, the program enters the outer loop of four nested loops. This loop starts with model initialization. When the initialization is complete, a bell is rung at the terminal and the model begins the simulation. The simulation is done in the inner three loops. The outer of these loops proceeds in one hour time steps until the user elects to stop the simulation or until 48 hours of simulated time have passed. The next loop proceeds in steps that correspond to the time between puff releases (advection period). The puffs are moved individually during an advection period. The movement of puff during an advection period takes place within the innermost loop. That loop contains the DIFOEP subroutine called by MESOI.

Figure 9 shows the structure of the MESOI main program. The three outer loops are indicated by the left branches from the decision points marked: More Advection Steps this Hour? More Hours? and New Simulation? The innermost loop is the left branch from the decision point marked: More Puffs?

The model initialization portion of MESOI includes the direct calls to subroutines GRIDIN, TOPFIL, INIT, ARRIVN, and RELEAS. These subroutines call other subroutines as indicated in Table 8. All of the subroutines that are used in model initialization are described in the next section. However, it should be noted that the left branch at "New Simulation?" is actually two branches. One permits total reinitialization of the model, and the other retains the previous grid and topographical initialization. In the latter case, the reinitialization is more concerned with release points, sources, effluents and critical receptors. It also includes positioning the model at the appropriate position in the data files. Subroutines LOCATE and DATRD are used to position the files. All meteorological data used in the simulation are written to disk files using subroutine DATWR.

Once the simulation starts, the model immediately moves to the first advection period in the first hour. The appropriate meteorological data are read, and the surface wind field is prepared using subroutines WIND and TERRA. When the wind field is prepared, subroutines ALPHA, SPEED and PLMRIZ are used to compute the effective release height for a puff, and the proper attributes are assigned to the puff by subroutine PUFFR. Subroutine PUFFM

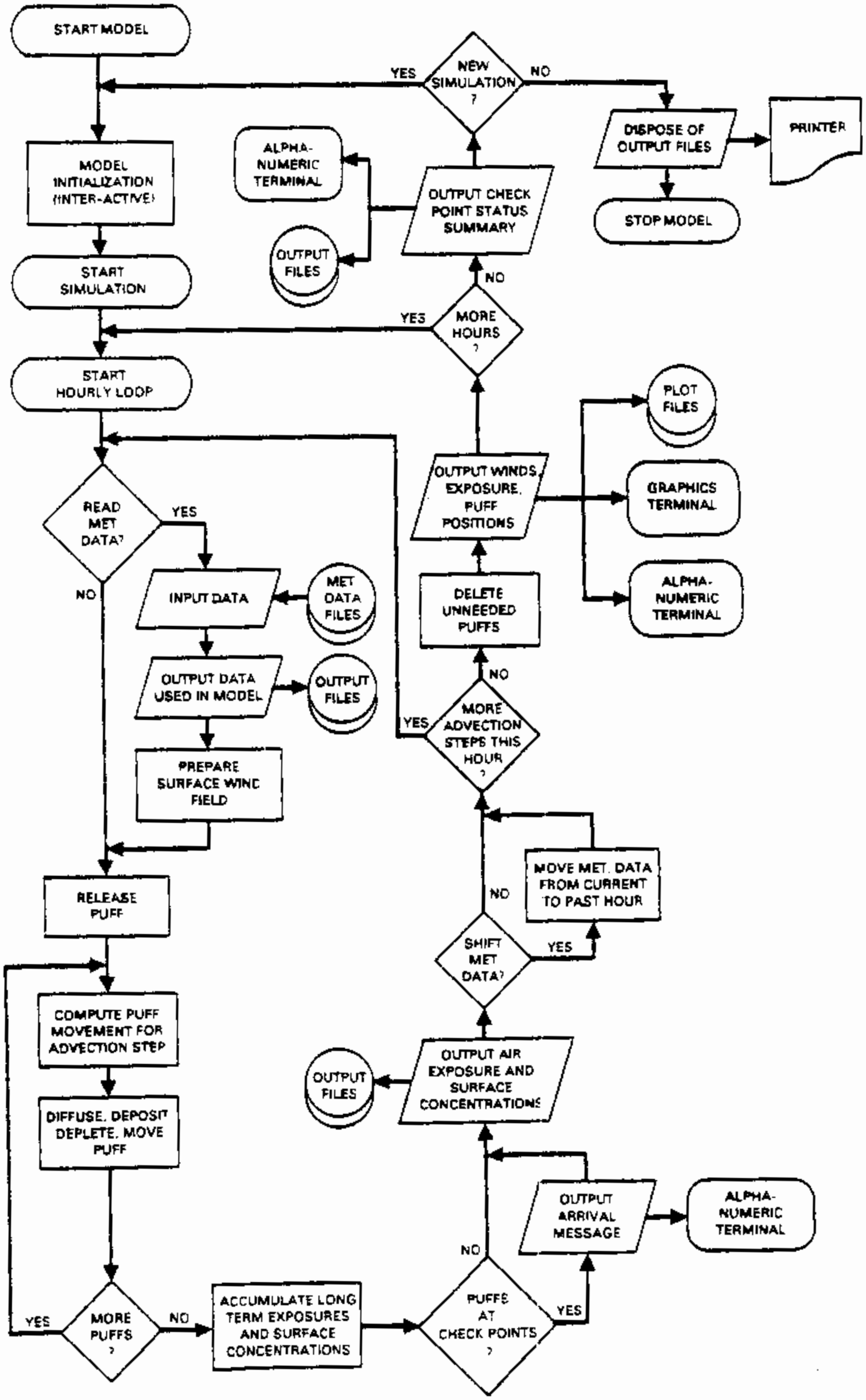


FIGURE 9. Program Logic for MESOI Version 2.0

determines the puff movements during the advection period. Subroutines DTOPO, DIFDEP, RDK and SIGMA are then used in the actual transport, diffusion, deposition and depletion computations. At the completion of these computations, MESOI moves to the next puff. Puffs are moved in inverse chronological order, i.e. the most recently released puff is moved first.

Time-integrated air concentrations and surface concentrations are accumulated in DIFDEP as the puffs are moved during the advection periods. Only after all puffs have been moved in an advection period do these accumulations represent an estimate of the changes made during the period. At this time the the time-integrated air and surface concentrations accumulated during an advection period are added to the long term accumulations. The addition of the advection period time-integrated air concentrations is a simple arithmetic operation, but the addition of the surface concentrations may involve decay and ingrowth. Figure 10 shows how decay and ingrowth are treated in the accumulation of the long term time-integrated concentrations.

MESOI automatically writes both the advection period and long term time-integrated air surface concentrations to disk files at the end of each advection period. These files may be used to estimate average concentrations. The user is also provided with a warning message if the time-integrated air concentration at any of the checkpoints has risen above a level of concern during the period. These features are provided by subroutines PRINTE and ARRTIM, respectively.

If detailed output on each puff is desired at the end of the advection periods, it may be obtained using subroutine TESTM.

When all puffs have been moved, MESOI goes to the next advection step. The meteorological data are updated if new data are required, a new puff is prepared for release, and the puff movement-computational cycle is repeated. Updating the meteorological data involves moving wind and temperature data from the coming data interval to the past interval using subroutine SHIFT, and then reading new data for the coming periods using subroutine DATRD.

The computational cycle and model output are described in detail in later sections.

At the completion of each simulated hour, MESOI provides the user with an opportunity to view the intermediate results of the simulation. These results may be directed to an alpha-numeric or graphics terminal or to files to be plotted on a hard-copy plotter. At this point MESOI also evaluates each of the puffs that are being followed. Puffs that are close together are combined, and those with center concentrations that have fallen below a minimum concentration of interest are deleted.

Subroutines involved in the output options at the end of each hour include SCREEN, PLOTZ, WNDFLD and WNDPLT. Subroutine CLEAN does the house-keeping chores associated with deleting and combining puffs.

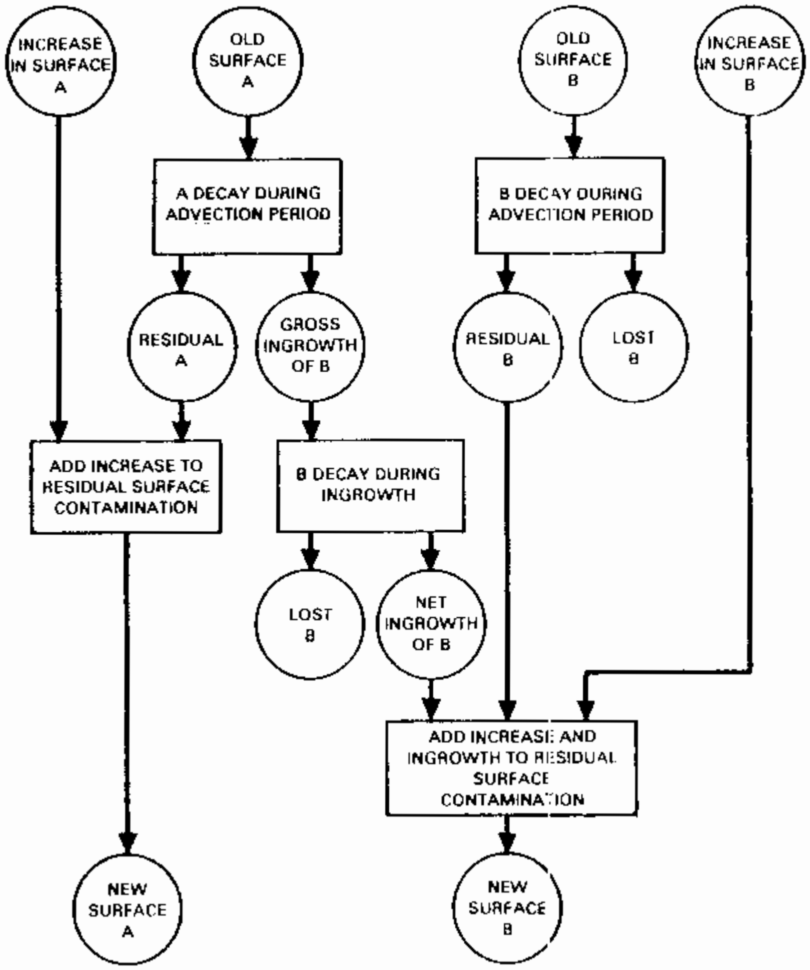


FIGURE 10. Decay and Ingrowth Scheme for Long-Term Time-Integrated Surface Concentrations During Advection Periods

When 48 simulation hours have been completed, or the user elects to bring the simulation to a close, MESOI returns a summary of the conditions at predefined checkpoints. The summary gives the times at which each of two threshold time-integrated air concentrations have been estimated to have been exceeded.

MESOI then requests instructions with respect to further simulations. If further simulations are to follow, instructions are requested about reinitialization. If there are no further simulations, the MESOI requests instructions regarding the disposition of the output files generated by the model. Disposition options include saving, printing, printing then deleting and deleting the files. The options available at the end of simulations are provided by subroutine MODEND.

Program Listing 1 provides the MESOI computer code. The main program and its associated common blocks require almost 98,000 bytes of memory. Of this total, about 2100 bytes are used for storage of the executable code, about 1000 bytes are used for local data, and about 94,600 bytes are used by the common blocks.

```
      PROGRAM MESOI

C*****************************************************************
C
C     MESOI
C     Interactive Puff Diffusion Model, Version 2.0, June 1983
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description:
C       This is the main controlling progam for the MESOI model.
C
C       The programs have been written primarily in standard FORTRAN-77.
C       Documentation describes the non-standard usages. Code development
C       was done on a VAX 11/780.
C
C     Relationship to other modules:
C
C         Makes calls to:  ALPHA,  ARRIVN, ARRTIM, CLEAN,  DATRD,
C                          DATWR,  DIFDEP, DTOPO,  EXPSUM, FILOPN,
C                          GRIDIN, INIT,   LOCATE, MODEND, PLMRIZ,
C                          PLOTZ,  PRINTE, PUFFM,  PUFFR,  RELEAS,
C                          SCREEN, SHIFT,  WINSPD,  TERRA,  TESTM,
C                          TOPFIL, WIND,   WNDFLD, WNDPLT
C
C         Called from:     NONE
C
C*****************************************************************

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

      INTEGER SHR

      CHARACTER*1  IOPT(6)

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

      CALL DATE (RDATE)
      CALL TIME (RTIME)

      WRITE (LUN(2),110) RTIME, RDATE
  110 FORMAT (//////, 15X, 'MESOI -- INTERACTIVE PUFF DIFFUSION MODEL',
     +  /26X, 'VERSION 2.0     JUNE 1983', //26X, 'TIME = ', A8,
     +  //26X, 'DATE = ', A9////)
      PAUSE 'TYPE C TO CONTINUE'

      CALL FILOPN
      WRITE (LUN(5),110) RTIME, RDATE
```

Program Listing 1 MESOI

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

  120 CALL GRIDIN (DELXY, NUMSTA)
      CALL TOPFIL (DELXY)
      NXTDAY=0
  130 CALL INIT (NUMSTA)
      CALL ARRIVN (DELXY)
      CALL RELEAS (DDAY, DHR)

C   **  SEND BELL TO TERMINAL AND INDICATE START

      WRITE(LUN(1),140) "7, "7
  140 FORMAT('+', 5A1///)
      PRINT *, ' *** START SIMULATION ***'

C   ** HOURLY LOOP FOR SIMULATION

      LUINDX = 3

      DO 400 SHR = 1, 48

        DO 300 IADV = 1, NPH

          IF (IADV .EQ. 1 .OR. INTFLG .EQ. 1)  THEN
            IF (LUINDX .EQ. 3)  THEN
              IF (NXTDAY .EQ. 888)  THEN
                LUINDX = 4
                CALL LOCATE (DYR,DDAY,DHR,DMIN,LERROR,LUINDX,INTFLG)
              ENDIF
            ENDIF

            CALL DATRD (LUINDX, SHR)

            LDEPTH = LDEPTH * 10
            TT = TEMPB
            CALL DATWR

            CALL WIND (NUMSTA)
            IF (TITLE(1) .NE. '*')  CALL TERRA

            IF (LDEPTH .GT. 800 .AND. ALPHFL .EQ. 1)  THEN
              CALL ALPHA (STAB, ALPHAU, ALPHAV)
            ENDIF
          ENDIF

          TINC = (IADV-1) * ADT
          DO 230 NS = 1, NSOURC
            IF (PUFLG(NS) .EQ. 1) GOTO 210
            IF (RELDAY(NS) .NE. PREVDY) GOTO 230

            IF (RELHR(NS) .EQ. PREVHR .AND. PARTHR(NS) .EQ. IADV)  THEN
              PUFLG(NS) = 1
              WRITE (LUN(2),200) NS, PREVHR, IADV
              WRITE (LUN(5),200) NS, PREVHR, IADV
  200         FORMAT (/5X, 'SOURCE ', I1, '  PUFF RELEASE FLAG SET ',
     +          'AT HOUR ', I2, '  ADVECTION STEP ', I1/)
            ELSE
              GOTO 230
            ENDIF

  210       IF (NPUFFS(NS) .LT. MAXPUF(NS))  THEN
              PRISE = 0.0
              IF (PLMRFL(NS) .EQ. 1)  THEN

                AIRT = T + ((IADV - 1) * (T - TT) / NPH)
```

MESOI (cont'd)

```
               IF (STEMPC .GE. AIRT + 10.)  THEN
                 CALL WINSPD (NS, TINC, SPD)
                 CALL PLMRIZ (NS, AIRT, SPD, PRISE)
               ENDIF
             ENDIF
             CALL PUFFR (NS, PRISE, LDEPTH)
           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 PUFFM (M, TINC)

 240       DS = SQRT(DXS(M)**2 + DYS(M)**2)

           IF (TOPFLG .EQ. 1)  THEN
             CALL DTOPO (XP(M), YP(M), TPMI, DXS(M), DYS(M), TPMF)
           ENDIF

           DSMTR = DS * DELXY

           CALL DIFDEP (M, DSMTR, TPMI, TPMF)

 250     CONTINUE

         DO 270 J = 1, 31
           DO 260 I = 1, 31
             AXPDPL(I,J) = AXPDPL(I,J) + AXPADV(I,J)
             BXPDPL(I,J) = BXPDPL(I,J) + BXPADV(I,J)
             BDEPOS(I,J) = BDEPOS(I,J) * BDECA1 + ADEPOS(I,J) *
    +                      BINAS + BDPADV(I,J)
             ADEPOS(I,J) = ADEPOS(I,J) * ADECA1 + ADPADV(I,J)
 260       CONTINUE
 270     CONTINUE

         IF (ARRFL .EQ. 1) CALL ARRTIM (TINC)
         IF (TPUFFS .GT. 0) THEN

           IPSUM = 0
           DO 310 M = 1, TPUFFS
             IPSUM = IPSUM + MF(M)
 310       CONTINUE

           CALL PRINTE (SHR, IADV, DDAY, DHR, DMIN, IPSUM, DPPLG)
         ENDIF

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

C   ** CLEAR THE 15 MINUTE MATRIX ARRAYS

         DO 290 J = 1, 31
           DO 280 I = 1, 31
             AXPADV(I,J) = 0.0
             BXPADV(I,J) = 0.0
             ADPADV(I,J) = 0.0
             BDPADV(I,J) = 0.0
 280       CONTINUE
 290     CONTINUE
         IF (IADV .EQ. NPH .OR. INTFLG .EQ. 1)  CALL SHIFT
 300    CONTINUE
```

MESOI (cont'd)

```
        IF (TPUFFS .GT. 2*NPH)  CALL CLEAN

C   **    CLEAR THE OUTPUT OPTION ARRAY

        DO 320 K = 1, 6
          IOPT(K) = ' '
  320   CONTINUE

        WRITE (LUN(2),330)  SHR, DDAY, DHR, DMIN, TPUFFS
        WRITE (LUN(5),330)  SHR, DDAY, DHR, DMIN, TPUFFS
  330   FORMAT (/5X, 'END OF SIMULATION HOUR ', I3, 5X, 'DATA = DAY ',
     +    I3, ' TIME =  ', I2, ':', I2, /7X, I3, ' PUFFS ACTIVE'/)

  340   WRITE (LUN(2),350)
  350   FORMAT (/5X, 'SELECT OUTPUT OPTIONS:  S = SCREEN',
     +         /29X, 'P = PUFF PLOT',
     +         /29X, 'W = WIND FIELD PLOT',
     +         /29X, 'V = WIND VECTORS AT STATIONS',
     +        //29X, 'T = TERMINATE')

        READ (LUN(1),360,END=340,ERR=340) (IOPT(K),K=1,6)
  360   FORMAT (6A1)

        DO 370 K=1, 6
          IF (IOPT(K) .NE. ' ')  THEN
            IF (IOPT(K) .EQ. 'S' .OR. IOPT(K) .EQ. 's')  THEN
              CALL SCREEN (SHR, IPSUM)
              GOTO 370
            ENDIF

            IF (IOPT(K) .EQ. 'P' .OR. IOPT(K) .EQ. 'p')  THEN
              CALL PLOTZ
              GOTO 370
            ENDIF

            IF (IOPT(K) .EQ. 'W' .OR. IOPT(K) .EQ. 'w')  THEN
              CALL WNDFLD
              GOTO 370
            ENDIF

            IF (IOPT(K) .EQ. 'V' .OR. IOPT(K) .EQ. 'v')  THEN
              CALL WNDPLT
              GOTO 370
            ENDIF

            IF (IOPT(K) .EQ. 'T' .OR. IOPT(K) .EQ. 't') GOTO 410

          ENDIF
  370   CONTINUE

        PRINT *, ' SIMULATION ACTIVE'
  400 CONTINUE

  410 WRITE (LUN(2),420)
      WRITE (LUN(5),420)
  420 FORMAT (/5X, '*** SIMULATION  TERMINATED ***'///)
      IF (ARRFL .EQ. 1) CALL EXPSUM (TITLE)
      CALL MODEND (*120, *130, DPFLG)

      END
```

MESOI (cont'd)

MODEL INITIALIZATION SUBROUTINES

There are nine model initialization subroutines included in MESOI Version 2.0. The initialization subroutines are listed in Table 9, along with their memory requirements and purpose. The memory requirements are distributed among executable code (CODE), local data (LOCAL DATA) and data in common blocks (COMMON DATA). A more detailed description and the code for each of these subroutines is presented in this section. The discussion follows the order in which the subroutines are called.

GRIDIN

The GRIDIN module initializes the wind grid and establishes the stations for which wind data are available. It is called from MESOI and requires user interaction and a file defining the stations.

The module begins by specifying the default grid spacing (5000 m) and allowing the user to change it within limits (2.5 to 15 km). The user is then asked to enter the name of the file containing the station data. MESOI is configured to handle up to 30 stations. The data file is opened and read. Each record defines a station by:

4 character station name
x distance in kilometers from grid center
y distance in kilometers from grid center
station elevation in meters
station status (0 if active; 1 if disabled)

After the file has been read, the coordinates are converted into grid units. Active stations which are off the grid are disabled. The user display then shows the list of stations which will be used in the simulation.

The user is then given the opportunity to edit the station list. The status may be changed and stations can be added. The process can be repeated until the user is statisfied with the station list. When the list is complete, GRIDIN calls REARNG to delete the inactive stations from the list.

Program listing 2 contains the GRIDIN code.

REARNG

The REARNG module is called from GRIDIN if some of the station flags have been set to inactive (STATUS = 1). It deletes those wind stations from the list used in the wind field interpolation.

Program listing 3 contains the REARNG code

TABLE 9. Model Initialization Subroutines

	SIZE (BYTES)			
		LOCAL	COMMON	
NAME	CODE	DATA	DATA	USE
GRIDIN	2778	2191	735	Initialization of wind station parameters, set up wind stations for use in interpolation of surface winds, initialization of time integrated concentration, surface contamination and wind grids.
REARNG	133	12	671	Deletes inoperative wind stations from list.
STRAY	222	284	25503	Selects 10 wind stations closest to each grid node.
ASCND	175	96		Places 1D wind stations closest to a grid node in order by distance, closest first.
TOPFIL	548	636	3912	Initializes topography file for diffusion computations.
INIT	1327	1020	88419	Primary initializations of the model, e.g. puff release rate, deposition velocity, washout coefficients, set constants used in diffusion and deposition computations, enter title, date and time for start of simulation.
ARRIVN	13D6	9D7	1624	Initialization of selected check points for arrival of puffs.
RELEAS	3036	2D38	1372	Input of information on releases including: positions, times, durations, quantities, characteristics.
RDKIN	527	412	824	Input of half-lives of effluent and daughter, compute constants for use in decay and ingrowth calculations.

STRAY

This module sets up the station array for each grid point and is called from GRIDIN.

First, the station number and a distance from the station to the grid point are saved. This list is then arranged in order of increasing distance from the point by subroutine ASCND. The final list of the stations to be used in interpolation is then placed into arrays for later reference. This sequence is repeated for each grid point.

Program listing 4 contains the STRAY code.

ASCND

The ASCND module takes the list of station numbers and distances for a given grid point and arranges them in order of increasing distance. This module is called from the STRAY module for each grid point.

Program Listing 5 contains the ASCND code.

TOPFIL

This module reads in topographical data for the fine grid. The data are used in computing ground-level concentrations for elevated releases. They do not affect the wind field.

TOPFIL is called from MESOI as part of the initialization. It requests the user to enter the name of the file containing the data. If no data are available, the user enters "NONE" as the file name and the model will use uniform, flat terrain in it's calculations. If a file is opened, the topography flag is set (TOPFLG = 1) and a READ loop is executed. The data are expected to be integers representing elevation in meters at the nodes of the 31 by 31 grid. The data is read as 31 rows north to south with each row consisting of 31 values west to east. The loop executes 31 READ's with the format (1x, 31I4). This data structure allows the set to conform to the actual way the numbers or points would appear on a map.

TOPFIL will stop model execution if it detects an error in reading the data. However, no bounds checking is done on the data and there is no provision for missing data. If the module encounters an EOF before filling the TOPO array the remainder of the spaces are assumed to be zero. This could create a severe discontinuity and should be avoided.

Program listing 6 contains the TOPFIL code.

INIT

INIT is the primary initialization module and is called from MESOI. It begins by asking the user to input a title. This character string is used on all output to uniquely identify the run. A special feature of the title is its use in controlling the testing mode of operations. If the first character of the title is an asterisk(*), the testing mode is enabled (see TESTM description).

Next, a number of constants are defined and the major arrays and flags are set to zero. Following is an interactive input section where the user defines the data interval and the simulatior starting data and time. LOCATE is called search the data files for matching data. If the proper data is located, the starting records are read and cecoded.

Program listing 7 contains the INIT code.

ARRIVN

This module initializes the exposure checking utility. Up to 35 locations can be specified to monitor total cumulative exposure. During a simulation, messages are printed whenever the exposure at a checkpoint exceeds a preset threshold.

ARRIVN is called from MESOI during the initialization. The array ARRIVL is used to maintain all the needed information about the checkpoints. Eleven pieces of information are stored for each of the 35 points:

```
1)  x coordinate in km from grid center
2)  y coordinate in km from grid center
3)  status flag; 0 if location not on current grid
                 1 if checkpoint active
                 2 if 1st threshold has been exceeded
                 3 if 2nd threshold has been exceeded
4)  first 4 characters of location name
5)  second 4 characters of location name
6)  Julian date when first threshold was exceeded
7)  hour when first threshold was exceeded
8)  minute when first threshold was exceeded
9)  Julian date when second threshold was exceeded
10) hour when second threshold was exceeded
11) minute when second threshold was exceeded.
```

ARRIVN requests the name of a file containing the checkpoint definitions. If none is defined, the checkpoint flag (ARRFL) is set to zero and the subroutine returns. If a file is opened, data records are read to define the checkpoint names and locations. The data file consists of three elements per record: x and y coordinates anc name. The format is

(1X,F6.1,1X,F6.1,2A4). The input loop reads each record and checks to see if the location is on the grid. If it is, the first five elements of ARRIVL are defined. Input stops when an EOF is encountered or 35 checkpoints have been defined.

After the checkpoints are defined, they are listed to both the terminal and the output file. The two thresholds (THRSH1 and THRSH2) are then defined by the program. The user has the option to change them at this point.

Program Listing 9 contains the ARRIVN code.

RELEAS

The RELEAS module is used to initialize the source characteristics. It is called from MESOI and requires extensive user input.

The user first specifies the number of sources (1 to 4). Then for each source, the user specifies the following:

1) location of source; x and y from grid center in kilometers
2) release height; meters above ground level
3) release date and time
4) release duration
5) source term (if known)
6) stack parameters (if any).

The location coordinates are converted to grid units and checked to make sure the location is on the grid. The height, date and time are checked for realistic values. The date and time are then compared with the data to see if the release is after the specified simulation start. In all cases, an error message is generated if a problem is detected and the user is prompted again for input. The release duration is checked for acceptable values and is converted into the number of puffs to be released from that source. Finally, if the source term is known it can be defined in units of mass per hour. If the release is from a stack, the stack temperature and the efflux rate can be entered for later use in the plume rise calculations.

Once definition is complete for all the sources, the user specifies whether deposition and/or radioactive decay are to be enabled. There is no provision for having these options separate for each source. If decay is enabled, RELEAS calls RDKIN to initialize the process.

Program listing 9 contains the RELEAS code.

RDKIN

RDKIN is the subroutine that is used to initialize the decay/ingrowth process in MESOI. It is called from subroutine RELEAS. It requests that the user supply half-lives, in seconds, for the initial effluent species and the daughter species that results from the decay of the initial species. The half lives are converted to decay constants (ALMBDA and BLMBDA) for use with time increments in minutes.

Entering a half-life of zero for the parent species (A) will cause an early exit from RDKIN and cause the model to skip all decay related computations. A negative half-life will not be accepted for the parent species. Entering a zero or negative half-life for the daughter species (B) will indicate that the daughter species is stable, i.e. the daughter will not decay.

When the decay constants have been computed, RDKIN computes the fractions of the parent and daughter species remaining at the end of an advection period, and the fraction of the parent species that will result in net ingrowth of the daughter species during an advection period. These fractions are stored in the variables ADECA1, BDECA1 and BINAS. A coefficient, DKCOEF, is computed if the half-lives of the species are equal.

The RDKIN subroutine is contained in Listing 10.

```
C*******************************************************************
C
C     GRIDIN                                        MESOI  VERSION 2.0
C     Grid initialization module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Initialization of the grid and set up of
C       meteorological stations. Reads data from a user specified
C       file and allows editting of the station parameters and the
C       addition of new stations.
C
C     Relationship to other modules:
C
C         Makes calls to:  REARNG, STRAY
C
C         Called from:     MESOI
C
C*******************************************************************

      SUBROUTINE GRIDIN (DELXY, NUMSTA)

      INCLUDE 'CHAR.INC'
      INCLUDE 'STATN.INC'
      INCLUDE 'UNITS.INC'

      REAL XDIST(30), YDIST(30), NEW2, NEW3

      INTEGER STANUM(30)

      CHARACTER*4 NEW1
      CHARACTER*1 SELECT
      CHARACTER*80 FILE1

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

      DELXY = 5000.0
      WRITE (LUN(2),100)
      WRITE (LUN(5),100)
  100 FORMAT (//5X, 'MESOI ---> GRID INITIALIZATION',/)

      WRITE (LUN(2),110) DELXY
  110 FORMAT (/8X, 'THE CURRENT WIND GRID IS: 16 ROWS X 16 COLUMNS',
     +       //8X, 'DELXY = ', F10.2, ' METERS')
  120 WRITE (LUN(2),130)
  130 FORMAT (/5X, 'DO YOU WANT A DIFFERENT GRID SPACING?  Y OR N')

      READ (LUN(1),140,END=120,ERR=120) SELECT
  140 FORMAT (A1)
      IF (SELECT .NE. 'Y' .AND. SELECT .NE. 'y') GOTO 170

  150 WRITE (LUN(2),160)
  160 FORMAT (/5X, 'SPECIFY NEW DELTA XY (IN METERS)'/)
      READ (LUN(1),*,END=150,ERR=150) DELXY
```

Program Listing 2 Subroutine GRIDIN

```
    IF (DELXY .LT. 0)  THEN
      PRINT *, 'NEGATIVE VALUES NOT ACCEPTABLE'
      GOTO 150
    ENDIF

    IF (DELXY .LT. 2500.0 .OR. DELXY .GT. 15000.0)  THEN
      PRINT *, '      ****  WARNING    ****'
      PRINT *, 'GRID SPACING SELECTED IS OUTSIDE USUAL LIMITS'
      PRINT *, '  USE CARE IN INTERPRETING MODEL OUTPUT'
    ENDIF

170 WRITE (LUN(2),180)
180 FORMAT (/5X, 'MESOI ---> STATION INITIALIZATION',/)
    NUMSTA = 0

190 PRINT *, 'ENTER FILENAME FOR READING STATION INFORMATION > '
    READ (LUN(1),200) FILE1
200 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

210 READ (LUN(15),220,END=230) NAMST(NUMSTA-1),  XDIST(NUMSTA+1),
   +     YDIST(NUMSTA+1), ELEV(NUMSTA+1), STATUS(NUMSTA+1)
220 FORMAT (1X, A4, 2(2X,F6.2), 4X, I4, 4X, I1)
    NUMSTA = NUMSTA + 1
    GOTO 210

230 CLOSE (LUN(15))
240 NSTAT = 0
    DO 260 I=1, NUMSTA
      XSTA(I) = (XDIST(I)*1000. / DELXY) + 7.5
      YSTA(I) = (YDIST(I)*1000. / DELXY) + 7.5
      IF (XSTA(I) .LT. 0.0 .OR. XSTA(I) .GT. 15.0 .OR.
   +      YSTA(I) .LT. 0.0 .OR. YSTA(I) .GT. 15.0)  THEN
        WRITE (LUN(2),250) NAMST(I)
250     FORMAT (/5X, 'NOTE: ',A4, ' STATION IS OFF CURRENT GRID',
   +           /10X, 'STATUS SET TO 1 [INACTIVE]')
        STATUS(I) = 1
      ENDIF

      IF (STATUS(I).EQ.1) NSTAT = NSTAT + 1
260 CONTINUE

    WRITE (LUN(2),270) NUMSTA,NSTAT
270 FORMAT (/5X, 'THERE ARE CURRENTLY ',I2,' STATIONS WITH ',I2,
   +  ' DISABLED',/)
    WRITE (LUN(2),280)
280 FORMAT (6X, 'STA NAME', 6X, 'GRIDX', 5X, 'GRIDY', 4X, 'ELEV',
   +    4X, 'STATUS')

    DO 300 I = 1, NUMSTA
      WRITE (LUN(2),290) I,NAMST(I),XSTA(I),YSTA(I),ELEV(I),STATUS(I)
290   FORMAT (6X,I2, 2X,A4, 3X,F8.2, 2X,F8.2, 4X,I4, 6X,I1)
      IF (MOD(I,15) .EQ. 0)  PAUSE 'TYPE C TO CONTINUE'
300 CONTINUE

310 WRITE (LUN(2),320)
320 FORMAT (/5X, 'ANY CHANGES TO STATUS FLAGS?   Y OR N')
    READ (LUN(1), 330, END=310, ERR=310) SELECT
330 FORMAT (A1)
```

Subroutine GRIDIN (cont'd)

```
    IF (SELECT .NE. 'Y' .AND. SELECT .NE. 'y') GOTO 400

340 WRITE (LUN(2),350)
350 FORMAT (/5X, 'HOW MANY STATIONS TO BE CHANGED?')
    READ (LUN(1), *, END=340, ERR=340) ICHG
    IF (ICHG .LT. 0) GOTO 400
    IF (ICHG .LE. NUMSTA) GOTO 370
    WRITE (LUN(2),360)
360 FORMAT (/5X, 'GRIDIN >> VALUE EXCEEDS NUMBER OF CURRENT STATIONS',
   +        //)
    GOTO 340

370 WRITE (LUN(2),380)
380 FORMAT (/5X, 'ENTER STATION NUMBERS TO BE CHANGED --> N,N,N...')
    READ (LUN(1),*,END=370,ERR=370) (STANUM(I),I=1,ICHG)

    DO 390 I = 1, ICHG
      IF (STANUM(I) .LT. 0.0 .OR. STANUM(I) .GT. NUMSTA)  THEN
        PRINT *, STANUM(I),' IS NOT A CURRENT STATION; CHANGE IGNORED'
        GOTO 390
      ENDIF

      J = STANUM(I)
      IF (STATUS(J) .EQ. 0)  THEN
        STATUS(J) = 1
      ELSE
        STATUS(J) = 0
      ENDIF
390 CONTINUE
    GOTO 240

400 WRITE (LUN(2),410)
410 FORMAT (/5X, 'ANY STATIONS TO BE ADDED?  Y OR N')
    READ (LUN(1),420,END=400,ERR=400) SELECT
420 FORMAT (A1)
    IF (SELECT .NE. 'Y' .AND. SELECT .NE. 'y') GOTO 610
430 WRITE (LUN(2),440)
440 FORMAT (/5X, 'HOW MANY STATIONS TO BE ADDED?')
    READ (LUN(1),*,END=430,ERR=430) IADD
    IF (IADD .LE. 0) GOTO 610
    IF (NUMSTA+IADD .LE. 30) GOTO 460
    WRITE (LUN(2),450)
450 FORMAT (5X, 'GRIDIN >> ONLY 30 STATIONS ALLOWED')
    GOTO 430

460 WRITE (LUN(2),470)
470 FORMAT (5X, 'EACH STATION SPECIFICATION MUST INCLUDE:',/7X,
   +     'STATION ID -- 4 CHARACTERS',/7X,
   +     'X AND Y DISTANCES IN KILOMETERS FROM GRID CENTER',/7X,
   +     'ELEVATION IN METERS ABOVE SEA LEVEL',/)

    DO 600 I = 1, IADD
480   WRITE (LUN(2),490) NUMSTA+I
490   FORMAT (5X, 'SPECIFY 4 CHARACTER NAME FOR STATION ',I2,/)
      READ (LUN(1),500,END=480,ERR=480) NEW1
500   FORMAT (A4)
510   WRITE (LUN(2),520) NUMSTA+I
520   FORMAT (5X, 'SPECIFY X AND Y DISTANCES FOR STATION ',I2,/)
      READ (LUN(1),*,END=510,ERR=510) NEW2, NEW3
      IF (ABS(NEW2) .GT. (7.5*DELXY/1000.0)) GOTO 640
      IF (ABS(NEW3) .GT. (7.5*DELXY/1000.0)) GOTO 640

530   WRITE (LUN(2),540) NUMSTA+I
540   FORMAT (/5X, 'SPECIFY ELEVATION FOR STATION ',I2/)
      READ (LUN(1),*,END=530) NEW4
      IF (NEW4 .LT. 0 .OR. NEW4 .GT. 9999) THEN
```

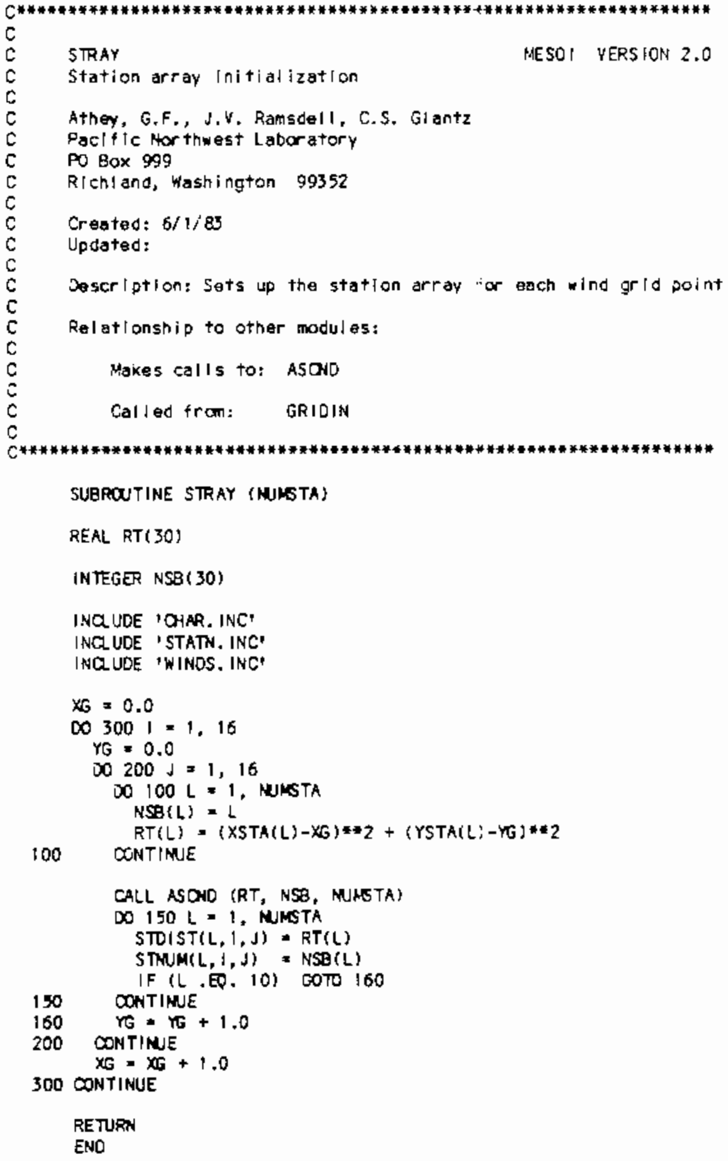
Subroutine GRIDIN (cont'd)

```
          PRINT *, 'ELEVATION OUT OF NORMAL RANGE'
          GOTO 530
        ENDIF

C   **  IF SPECIFICATIONS ARE ACCEPTABLE, ENTER INTO ARRAYS

        NAMST(NUMSTA+I) = NEW1
        XDIST(NUMSTA+I) = NEW2
        YDIST(NUMSTA+I) = NEW3
         ELEV(NUMSTA+I) = NEW4

  600 CONTINUE
      NUMSTA = NUMSTA + IADD
      GOTO 240

  610 CONTINUE
      ACTSTA = NUMSTA - NSTAT
      IF (NSTAT .NE. 0)  CALL REARNG (NUMSTA)
      CALL STRAY (ACTSTA)

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

C   **  WRITE COMPLETED GRID INFORMATION TO OUTPUT FILE

      WRITE (LUN(5),110) DELXY
      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

  640 WRITE (LUN(2),650) NUMSTA+I
  650 FORMAT (/5X, 'NEW STATION [NUMBER ',I2,'] IS OFF THE GRID',/)
      GOTO 400

      END
```

Subroutine GRIDIN (cont'd)

```
C*******************************************************************
C
C     REARNG                                        MESOI  VERSION 2.0
C     Rearrangement of station list
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Removes stations with status flag set from the
C           set of active stations to be used in wind interpolation.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     GRIDIN
C
C*******************************************************************

      SUBROUTINE REARNG (NUMSTA)

      INCLUDE 'CHAR.INC'
      INCLUDE 'STATN.INC'

      I = 0
      DO 100 K = 1, NUMSTA
        IF (STATUS(K) .EQ. 1)  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
```

Program Listing 3 Subroutine REARNG



```
C*********************************************************************
C
C     STRAY                                          MESOI  VERSION 2.0
C     Station array initialization
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/85
C     Updated:
C
C     Description: Sets up the station array for each wind grid point
C
C     Relationship to other modules:
C
C         Makes calls to:  ASCND
C
C         Called from:     GRIDIN
C
C*********************************************************************

      SUBROUTINE STRAY (NUMSTA)

      REAL RT(30)

      INTEGER NSB(30)

      INCLUDE 'CHAR.INC'
      INCLUDE 'STATN.INC'
      INCLUDE 'WINDS.INC'

      XG = 0.0
      DO 300 I = 1, 16
        YG = 0.0
        DO 200 J = 1, 16
          DO 100 L = 1, NUMSTA
            NSB(L) = L
            RT(L) = (XSTA(L)-XG)**2 + (YSTA(L)-YG)**2
 100      CONTINUE

          CALL ASCND (RT, NSB, NUMSTA)
          DO 150 L = 1, NUMSTA
            STDIST(L,I,J) = RT(L)
            STNUM(L,I,J)  = NSB(L)
            IF (L .EQ. 10)  GOTO 160
 150      CONTINUE
 160      YG = YG + 1.0
 200    CONTINUE
        XG = XG + 1.0
 300  CONTINUE

      RETURN
      END
```

Program Listing 4 Subroutine STRAY

```
C*****************************************************************
C
C     ASCND                                        MESOI  VERSION 2.0
C     Order stations
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: For a given grid point, stations are arranged in
C           ascending order by distance from the point.
C
C           RT  -- distance (X**2 + Y**2)
C           NSB -- station number
C           NUMSTA - total number of stations
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     STRAY
C
C*****************************************************************

      SUBROUTINE ASCND (RT, NSB, NUMSTA)

      REAL RT(30), TMP

      INTEGER NSB(30), NTMP

      DO 110 I = 1, NUMSTA-1
        J = 0
        DO 100 K = 1, NUMSTA-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
```

Program Listing 5 Subroutine ASCND

```
C**********************************************************************
C
C     TOPFIL                                         MESOI  VERSION 2.0
C     Topography data input module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Module to input ground level elevations to describe
C          topography for the 31 x 31 fine grid.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C**********************************************************************

      SUBROUTINE TOPFIL (DELXY)

      INCLUDE 'TOPOGR.INC'
      INCLUDE 'UNITS.INC'

      CHARACTER*80 FILE1

      TOPFLG = 0
      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.', /7X,
     +     'SPACING OF THE TOPOGRAPHY DATA POINTS MUST BE THE SAME'//)

 130  PRINT *, 'ENTER NAME OF FILE CONTAINING TOPOGRAPHY DATA'
      PRINT *, '  USE ''NONE'' WHEN TERRAIN IS NOT KNOWN >> '
      READ (LUN(1),140) FILE1
 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(15), FORM='FORMATTED', STATUS='OLD',
     +             FILE=FILE1,  IOSTAT=IER)
        IF (IER .NE. 0)  THEN
          PRINT *, 'TOPFIL >> ERROR ',IER, ' OPENING TOPOGRAPHY FILE'
          GOTO 130
        ENDIF
      ENDIF

      TOPFLG = 1
      DO 200 J = 31, 1, -1
        READ (LUN(15),150,IOSTAT=IER) (TOPO(I,J),I=1,31)
 150    FORMAT (1X, 31I4)
        IF (IER .GT. 0)  THEN
```

Program Listing 6 Subroutine TOPFIL

```
          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(15))
      RETURN
      END
```

Subroutine TOPFIL (cont'd)

```
C*******************************************************************
C
C     INIT                                         MESOI  VERSION 2.0
C     Primary initialization module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Primary initializations of the model, including:
C       identification title, current date and time, and basic
C       simulation parameters.
C
C     Relationship to other modules:
C
C         Makes calls to:  DATRD,  DATWR,  JULIAN, LOCATE,
C                          SHIFT,  TERRA,    WIND
C
C         Called from:     MESOI
C
C*******************************************************************

      SUBROUTINE INIT (NUMSTA)

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

      INTEGER STDAY, STHR

      CHARACTER*1 SELECT

C   **  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, 'MESOI ---> PRIMARY INITIALIZATION',/)

  110 WRITE (LUN(2),120)
  120 FORMAT (5X, 'ENTER RUN IDENTIFICATION TITLE OF UP TO',
     +   ' 50 CHARACTERS')
      READ (LUN(1),130,END=110,ERR=110) TITLE
  130 FORMAT (50A1)

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
```

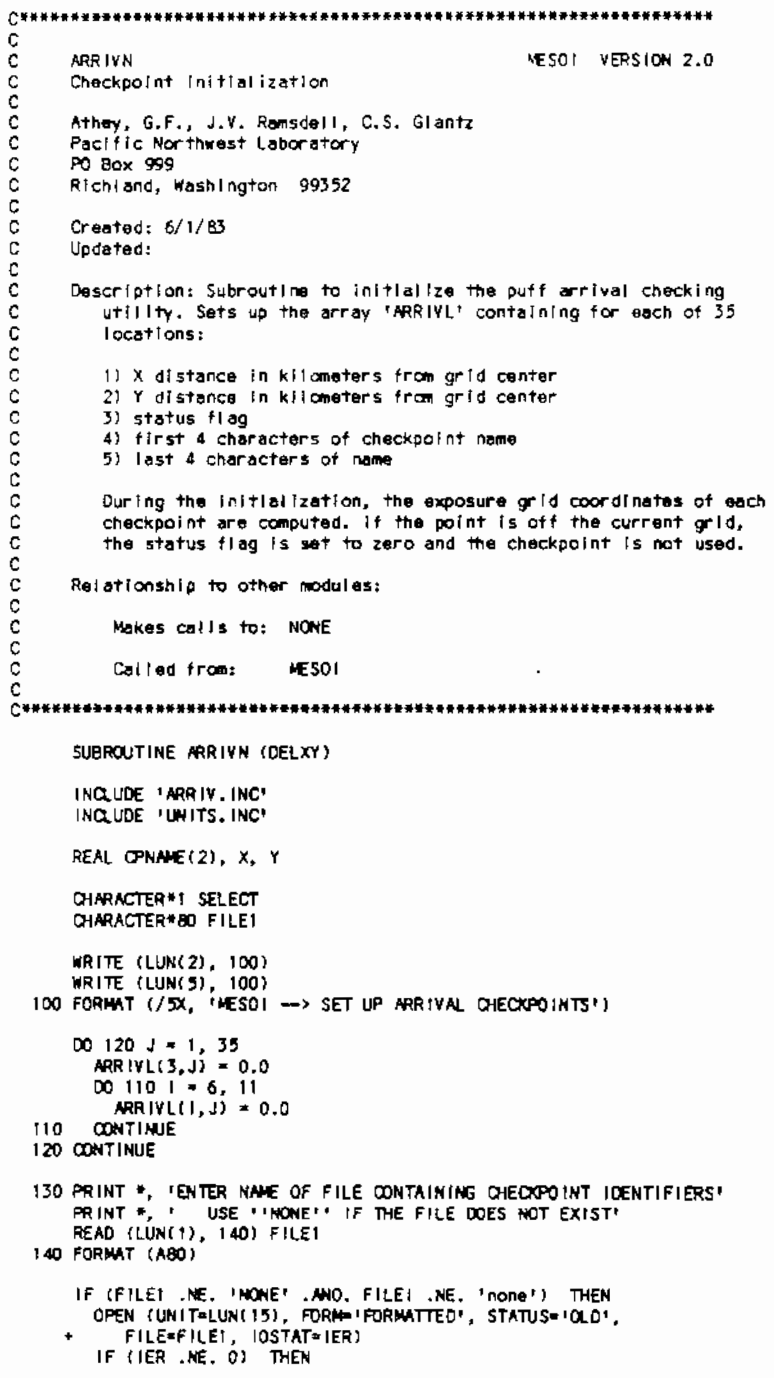
Program Listing 7 Subroutine INIT

```
        MF(I) = 0
  150 CONTINUE

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

      DO 180 J = 1, 31
        DO 170 I = 1, 31
          EXPCUM(I,J) = 0.0
          ADEPOS(I,J) = 0.0
          BDEPOS(I,J) = 0.0
          AXPDPL(I,J) = 0.0
          BXPDPL(I,J) = 0.0
          AXPADV(I,J) = 0.0
          BXPADV(I,J) = 0.0
          ADPADV(I,J) = 0.0
          BDPADV(I,J) = 0.0
  170   CONTINUE
  180 CONTINUE

C   **  DEPOSITION VELOCITY = 1 CM/SEC
C   **    OR 36 METERS/HOUR

         DV = 0.01 * 3600
      TWOPI = 6.283185

      ADT = 1.0 / NPH
      FAC = ADT * 3600 / DELXY
      ACNST = 2.0 / (60.0 * TWOPI ** 1.5)

      INTFLG = 0
      WRITE (LUN(2),190)
  190 FORMAT (/5X, 'IS METEOROLOGICAL DATA AVAILABLE AT 15 MINUTE ',
     +  'INTERVALS?  Y OR N', /7X, 'DEFAULT IS HOURLY DATA'/)
      READ (LUN(1), 200) SELECT
  200 FORMAT (A1)
      IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  INTFLG = 1

  210 WRITE (LUN(2),220)
  220 FORMAT (//5X, 'ENTER DATE FOR SIMULATION START -- MMDDYY')
      READ (LUN(1),230,END=210,ERR=210) IMO, IDAY, IYR
  230 FORMAT (10I2)
      IF (IYR .LT. 70 .OR. IYR .GT. 85)  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
  240   FORMAT (5X, 'A DATE OF ', I2,'/',I2,'/',I2, ' IS IMPOSSIBLE ',
     +    '--> TRY AGAIN',//)
        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),*,END=260,ERR=260) STHR
      IF (STHR .LT. 1 .OR. STHR .GT. 24)  THEN
        WRITE (LUN(2),280) STHR
  280   FORMAT (5X, 'A STARTING HOUR OF ',I3,' IS IMPOSSIBLE ',
     +    '--> TRY AGAIN',//)
        GOTO 260
```

Subroutine INIT (cont'd)

```
      ENDIF

      LUINDX = 3
      CALL LOCATE (IYR, STDAY, STHR, 0, LERROR, LUINDX, INTFLG)
      IF (LERROR .EQ. 1)  GOTO 210

      CALL DATRD (LUINDX, 0)
      LDEPTH = LDEPTH * 10
      TT = TEMPB

      ALPHAU = 1.0
      ALPHAV = 1.0
      ALPHFL = 0
      WRITE (LUN(2),290)
  290 FORMAT (/2X, 'DO YOU WISH TO USE THE ALPHA MODULE IN THE ',
     +  'VERTICAL INTERPOLATION OF THE WINDS?', /4X, 'Y OR N. ',
     +  /7X, 'IF NO, A LINEAR INTERPOLATION (ALPHA =1.0) WILL BE USED'/)
      READ (LUN(1),200) SELECT
      IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y') ALPHFL = 1

      PRINT *, 'ALL FILES OPENED AND FIRST WIND RECORDS PROCESSED'

      CALL DATWR

      CALL WIND (NUMSTA)

      CALL SHIFT

      TERFLG = 0
      IF (TITLE(1) .NE. '*')  CALL TERRA

      WRITE (LUN(5),300) TITLE, RDATE, RTIME
  300 FORMAT ('1', //5X, 'RUN TITLE --> ', 50A1, 5X, A8, 2X, A8)
      IF (TITLE(1) .EQ. '*')  THEN
        WRITE (LUN(5),310)
  310   FORMAT (/8X, 'TESTING MODE SELECTED'/)
      ENDIF

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

  800 STOP ' ERROR   -- UNACCEPTABLE SPECIFICATION IN INIT'
      END
```

Subroutine INIT (cont'd)



```
C*********************************************************************
C
C     ARRIVN                                        MESOI  VERSION 2.0
C     Checkpoint Initialization
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to initialize the puff arrival checking
C        utility. Sets up the array 'ARRIVL' containing for each of 35
C        locations:
C
C        1) X distance in kilometers from grid center
C        2) Y distance in kilometers from grid center
C        3) status flag
C        4) first 4 characters of checkpoint name
C        5) last 4 characters of name
C
C        During the initialization, the exposure grid coordinates of each
C        checkpoint are computed. If the point is off the current grid,
C        the status flag is set to zero and the checkpoint is not used.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*********************************************************************

      SUBROUTINE ARRIVN (DELXY)

      INCLUDE 'ARRIV.INC'
      INCLUDE 'UNITS.INC'

      REAL CPNAME(2), X, Y

      CHARACTER*1 SELECT
      CHARACTER*80 FILE1

      WRITE (LUN(2), 100)
      WRITE (LUN(5), 100)
  100 FORMAT (/5X, 'MESOI --> SET UP ARRIVAL CHECKPOINTS')

      DO 120 J = 1, 35
        ARRIVL(3,J) = 0.0
        DO 110 I = 6, 11
          ARRIVL(I,J) = 0.0
  110   CONTINUE
  120 CONTINUE

  130 PRINT *, 'ENTER NAME OF FILE CONTAINING CHECKPOINT IDENTIFIERS'
      PRINT *, '   USE ''NONE'' IF THE FILE DOES NOT EXIST'
      READ (LUN(1), 140) FILE1
  140 FORMAT (A80)

      IF (FILE1 .NE. 'NONE' .AND. FILE1 .NE. 'none')  THEN
        OPEN (UNIT=LUN(15), FORM='FORMATTED', STATUS='OLD',
     +      FILE=FILE1, IOSTAT=IER)
        IF (IER .NE. 0)  THEN
```

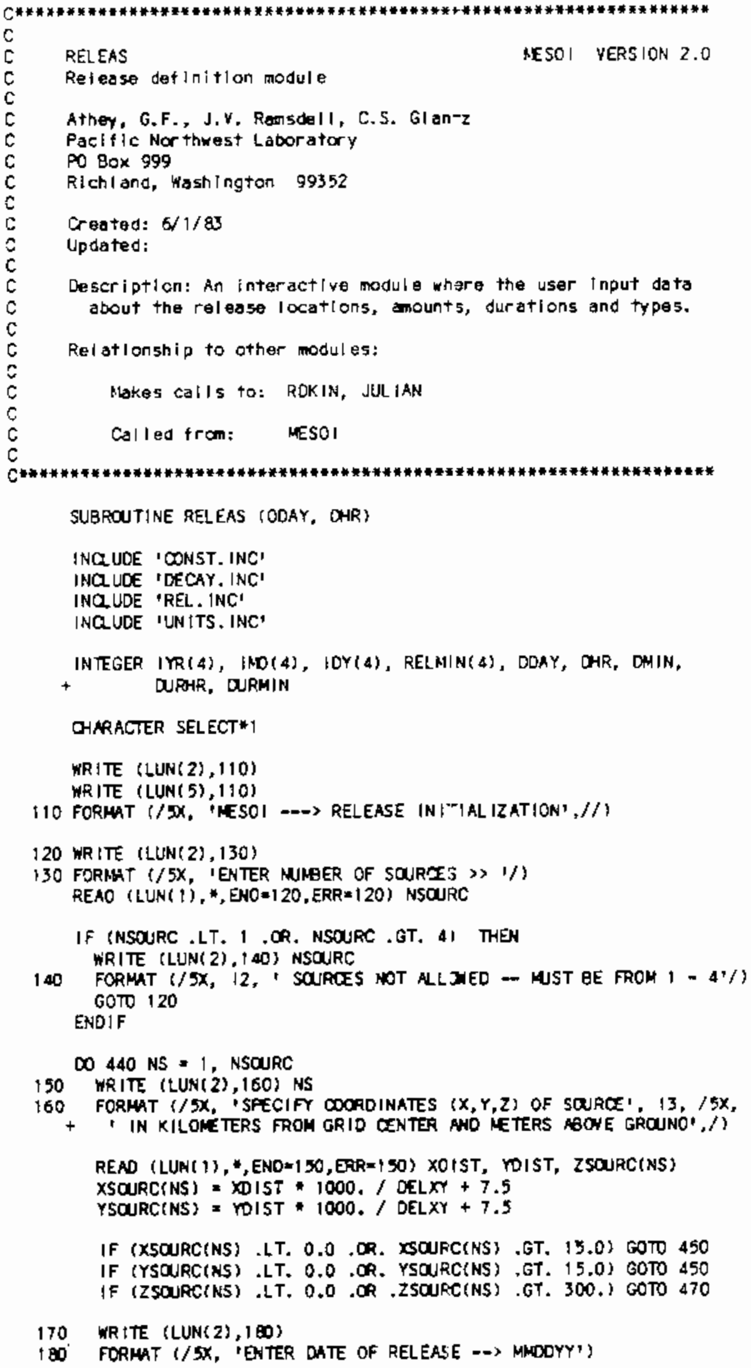
Program Listing 8 Subroutine ARRIVN

```
        PRINT *, 'ARRIVN >> ERROR ',IER,' OPENING FILE'
        GOTO 130
       ENDIF
      ELSE
       WRITE (LUN(2),150)
       WRITE (LUN(5),150)
  150  FORMAT (/7X,' NO CHECKPOINTS INITIALIZED FROM DISK FILE')
       ARRFL = 0

C   ** AN INTERACTIVE CHECKPOINT DEFINITION SECTION COULD FIT IN HERE

       RETURN
      ENDIF

      DO 220 J = 1, 35
  200  READ (LUN(15), 210, END=230) X, Y, C?NAME
  210  FORMAT (1X, F6.2, 1X, F6.2, 1X, 2A4)
       X = (X * 1000.0 / (DELXY/2.0)) + 16
       Y = (Y * 1000.0 / (DELXY/2.0)) + 16
       IF (X .LT. 1 .OR. X .GT. 31)  GOTO 200
       IF (Y .LT. 1 .OR. Y .GT. 31)  GOTO 200
       ARRIVL(1,J) = X
       ARRIVL(2,J) = Y
       ARRIVL(3,J) = 1.0
       ARRIVL(4,J) = CPNAME(1)
       ARRIVL(5,J) = CPNAME(2)
  220 CONTINUE
  230 CLOSE (LUN(15))

      NCP = 0
      DO 240 J = 1, 35
       NCP = NCP + ARRIVL(3,J)
  240 CONTINUE

      WRITE (LUN(2),250) NCP
      WRITE (LUN(5),250) NCP
  250 FORMAT (/10X, I2, ' CHECKPOINTS ACTIVE ON THE CURRENT GRID'/)
      ARRFL =1


      WRITE (LUN(5),280)
  280 FORMAT (8X, 'CHKPNT  EXP COORD', /10X, 'NO      X',
     +  4X, 'Y       NAME'/)

      DO 300 J = 1, NCP
       L = ARRIVL(1,J) + 0.5
       M = ARRIVL(2,J) + 0.5
       WRITE (LUN(5),290) J, L, M, ARRIVL(4,J), ARRIVL(5,J)
  290  FORMAT (10X, I2, 2X, 2(2X,I3), 5X, 2A4)
  300 CONTINUE

      THRSH1 = 1.0E-15
      THRSH2 = 1.0E-10

  360 WRITE (LUN(2),370) THRSH1, THRSH2
  370 FORMAT (/5X, 'DEFAULT VALUES FOR THE THRESHOLDS ARE:',
     +  /7X, 'THRSH1 = ', 1PE10.3, /7X, 'THRSH2 = ', E10.3)
      WRITE (LUN(2),380)
  380 FORMAT (/5X, 'DO YOU WISH TO CHANGE THE THRESHOLDS?  Y OR N')
      READ (LUN(1),390) SELECT
  390 FORMAT (A1)

      IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
       PRINT *, ' ENTER VALDES FOR THRESHOLDS 1 AND 2'
       READ (LUN(1),*) THRSH1, THRSH2
       IF (THRSH1 .LT. 0.0 .OR. THRSH2 .LT. 0.0)  THEN
```

Subroutine ARRIVN (cont'd)

```
        PRINT *, ' NEGATIVE THRESHOLD VALUES NOT ACCEPTABLE'
        GOTO 360
      ENDIF
    ENDIF

    WRITE (LUN(2),400) THRSH1, THRSH2
    WRITE (LUN(5),400) THRSH1, THRSH2
400 FORMAT (/10X, 'THE ARRIVAL CHECKING THRESHOLDS ARE: '/12X,
   +     1PE10.3, ' AND ', E10.3/)

    EXPFLG = 0
    RETURN
    END
```

Subroutine ARRIVN (cont'd)



```
C***********************************************************************
C
C     RELEAS                                          MESOI  VERSION 2.0
C     Release definition module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: An interactive module where the user input data
C       about the release locations, amounts, durations and types.
C
C     Relationship to other modules:
C
C         Makes calls to:  RDKIN, JULIAN
C
C         Called from:     MESOI
C
C***********************************************************************

      SUBROUTINE RELEAS (ODAY, OHR)

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

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

      CHARACTER SELECT*1

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

  120 WRITE (LUN(2),130)
  130 FORMAT (/5X, 'ENTER NUMBER OF SOURCES >> '/)
      READ (LUN(1),*,END=120,ERR=120) NSOURC

      IF (NSOURC .LT. 1 .OR. NSOURC .GT. 4)  THEN
        WRITE (LUN(2),140) NSOURC
  140   FORMAT (/5X, I2, ' SOURCES NOT ALLOWED -- MUST BE FROM 1 - 4'/)
        GOTO 120
      ENDIF

      DO 440 NS = 1, NSOURC
  150   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=150,ERR=150) 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')
```

Program Listing 9 Subroutine RELEAS

```
      READ (LUN(1),190,END=170,ERR=170) IMO(NS), IDY(NS), IYR(NS)
190   FORMAT (10I2)

      IF (IYR(NS) .LT. 70 .OR. IYR(NS) .GT. 85) 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. DDAY .AND. RELDAY(NS) .GT. 2) GOTO 510

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

      IF (RELHR(NS) .LT. 1 .OR. RELHR(NS) .GT. 24) GOTO 530
      IF (RELHR(NS) .LT. DHR .AND. RELDAY(NS) .EQ. DDAY) GOTO 510
      IF (RELMIN(NS) .LT. 0 .OR. RELMIN(NS) .GT. 59) GOTO 550
      PARTHR(NS) = RELMIN(NS) / (60/NPH) + 1

250   WRITE (LUN(2),260)
260   FORMAT (/5X, 'DURATION OF RELEASE?  HOURS,MINUTES',
     +        /7X, 'IF CONTINUOUS,  ENTER 0,0 ')
      READ (LUN(1),*,END=280,ERR=280) DURHR,DURMIN

      IF (DURHR .LT. 0 .OR. DURMIN .LT. 0) GOTO 570
      IF (DURHR .EQ. 0 .AND. DURMIN .EQ. 0) GOTO 280
      MAXPUF(NS) = (DURHR*NPH) + (DURMIN / (60/NPH))

      IF (MAXPUF(NS) .LE. 0)  THEN
        PRINT *, ' DURATION SHORTER THAN BASIC PUFF RELEASE INTERVAL'
        PRINT *, '    A SINGLE PUFF WILL BE RELEASED'
        MAXPUF(NS) = 1
      ENDIF

      WRITE (LUN(2),270) MAXPUF(NS),NS
270   FORMAT (/5X, I4, ' PUFFS WILL BE RELEASED FROM SOURCE', I3)
      GOTO 300

280   DURHR = 0
      DURMIN = 0
      MAXPUF(NS) = 125
      WRITE (LUN(2),290) NPH
290   FORMAT (5X, 'RELEASE WILL BE CONTINUOUS AT ', I3, ' PUFFS/HOUR')

300   WRITE (LUN(2),310)
310   FORMAT (/5X, 'IS THE SOURCE TERM KNOWN?  Y OR N', /7X,
     +            'DEFAULT IS A UNIT RELEASE'/)
      READ (LUN(1),320,END=300,ERR=300) SELECT
320   FORMAT (A1)

      IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
330     WRITE (LUN(2),340)
340     FORMAT (/5X, ' INPUT A RELEASE RATE IN MASS ',
     +     '[CURIE, GRAM, ETC.] PER HOUR',/)
        READ (LUN(1),*,END=330,ERR=330) QSOURC(NS)

        IF (QSOURC(NS) .LE. 0.0)  THEN
          PRINT *, ' ZERO OR NEGATIVE RELEASE RATE NOT ACCEPTABLE'
          GOTO 300
        ENDIF

        QSOURC(NS) = QSOURC(NS) / NPH
      ELSE
        QSOURC(NS) = 1.0 / NPH
```

Subroutine RELEAS (cont'd)

```
      ENDIF

 350  WRITE (LUN(2),360)
 360  FORMAT (/5X, 'IS RELEASE FROM A STACK FOR WHICH PLUME ',
     +       'RISE CAN BE COMPUTED? Y OR N')

      READ (LUN(1),320,END=350,ERR=350) SELECT
      IF(SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
 370    WRITE (LUN(2),380)
 380    FORMAT (/5X, 'ENTER STACK GAS TEMPERATURE IN DEG C > ',/)
        READ (LUN(1),*,END=370,ERR=370)  STEMPC

        IF (STEMPC .LT. 0.0 .OR. STEMPC .GT. 400.0)  THEN
          PRINT *, ' ACCEPTABLE VALUES ARE 0 TO 400 ONLY'
          GOTO 370
        ENDIF

 390    WRITE (LUN(2),400)
 400    FORMAT (/5X, 'ENTER EFFLUX RATE IN CUBIC METERS PER SECOND',/)
        READ (LUN(1),*,END=390,ERR=390)  EFFLUX

        IF (EFFLUX .LE. 0.0)  THEN
          PRINT *, ' ZERO OR NEGATIVE EFFLUX NOT ACCEPTABLE'
          GOTO 390
        ENDIF

        PLMRFL(NS) = 1
        QH(NS) = EFFLUX
      ELSE
        PLMRFL(NS) = 0
            QH(NS) = 0.0
      ENDIF

 440 CONTINUE
    GOTO 600

C   **  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, 'RELEAS >> A DATE OF ', 2I2, ' IS NOT VALID')
      GOTO 170

 510  WRITE (LUN(2),520) DDAY, DHR, RELDAY(NS), RELHR(NS)
 520  FORMAT (/5X, 'RELEAS >> 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, 'RELEAS >> SPECIFIED HOUR OF RELEASE IS NOT',
     +       ' POSSIBLE --> ', I5)
      GOTO 210

 550  WRITE (LUN(2),560) RELMIN(NS)
 560  FORMAT (/5X, 'RELEAS >> SPECIFIED MINUTES OF RELEASE NOT',
     +       ' POSSIBLE --> ', I5)
      GOTO 210
```

Subroutine RELEAS (cont'd)

```
570   WRITE (LUN(2),580) DURHR, DURMIN
580   FORMAT (/5X, 'RELEAS >> NEGATIVE VALUES OF DURATION NOT',
   +        ' ALLOWED --> ', 2I5/)
      GOTO 250

600 PRINT *, '  SOURCE DEFINITION COMPLETE '

    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, /19X, I4, ' PUFFS WITH ',F8.3,' MASS EACH')

      IF (PLMRFL(NS) .EQ. 0 )  THEN
        WRITE (LUN(5),640)
640     FORMAT (/17X, 'NO PLUME RISE')
      ELSE
        WRITE (LUN(5),650)
650     FORMAT (/17X, 'PLUME RISE WILL BE CALCULATED')
      ENDIF

680 CONTINUE

710 WRITE (LUN(2),720)
720 FORMAT (/5X, 'WILL THE MATERIAL RELEASED DEPOSIT? Y OR N')
    READ (LUN(1),320,ERR=710,END=710) SELECT

    IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
      DPFLG = 1
      WRITE (LUN(5),750)
750   FORMAT (/7X, 'MATERIAL WILL DEPOSIT')
    ELSE
      DPFLG = 0
      WRITE (LUN(5),760)
760   FORMAT (/7X, 'NO MATERIAL DEPOSITION')
    ENDIF

    RDKFL = 0

    ALMBDA = 0.0
    BLMBDA = 0.0
    ADECA1 = 1.0
    BDECA1 = 1.0

    WRITE (LUN(2),800)
800 FORMAT (/5X, 'DO YOU WISH TO ENABLE RADIOACTIVE DECAY? Y OR N'/)
    READ (LUN(1),320) SELECT

    IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
      CALL RDKIN (ADT)
      IF (ALMBDA .GT. 0) RDKFL = 1
    ENDIF

    RETURN
    END
```

Subroutine RELEAS (cont'd)

```
C*******************************************************************
C
C     RDKIN                                         MESOI  VERSION 2.0
C     Radioactive decay initialization module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999,
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Module to define the half-lives for parent and daughter
C          products and to calculate the necessary constants for use in the
C          decay calculations.
C
C     Relationship to other modules:
C
C        Makes calls to:  NONE
C
C        Called from:     RELEAS
C
C*******************************************************************

      SUBROUTINE RDKIN (ADT)

      INCLUDE 'DECAY.INC'
      INCLUDE 'UNITS.INC'

      REAL HLIFA, HLIFB

 100  WRITE (LUN(2),110)
 110  FORMAT (/5X, 'ENTER HALF-LIFE IN SECONDS FOR PARENT SPECIES',
     +        /7X, '0 ABORTS THE DEFINITION PROCESS'/)
      READ (LUN(1),*) HLIFA
      IF (HLIFA .EQ. 0)  RETURN
      IF (HLIFA .LT. 0)  THEN
        PRINT *, '  NEGATIVE HALF-LIFE NOT ACCEPTED'
        GOTO 100
      ENDIF

      IF (HLIFA .LT. (60) .OR. HLIFA .GT. 3.6E6)  THEN
        PRINT *, ' SPECIFIED HALF-LIFE NOT WORTH TRACKING'
        GOTO 100
      ENDIF

      ALMBDA = 0.69315 * 60.0 / HLIFA

 200  WRITE (LUN(2),210)
 210  FORMAT (/5X, 'ENTER HALF-LIFE IN SECONDS FOR DAUGHTER SPECIES',
     +        /7X, 'ZERO IF STABLE'/)
      READ (LUN(1),*) HLIFB
      IF (HLIFB .LT. 0)  THEN
        PRINT *, '  NEGATIVE HALF-LIFE NOT ACCEPTED'
        GOTO 200
      ENDIF

      IF (HLIFB .GT. 0)  BLMBDA = 0.69315 * 60.0 / HLIFB

      ADECA1 = EXP (-ALMBDA * 60.0 * ADT)
      BDECA1 = EXP (-BLMBDA * 60.0 * ADT)
      IF (ALMBDA .NE. BLMBDA)  THEN
        DKCOEF = ALMBDA / (BLMBDA - ALMBDA)
        BINAS  = DKCOEF * (ADECA1 - BDECA1)
```

Program Listing 10 Subroutine RDKIN

```
      ELSE
        BINAS  = ALMBDA * 60.0 * ADT * ADECA1
      ENDIF

      PRINT *, ' DECAY CONSTANTS CALCULATED'
      WRITE (LUN(2), 320) HLIFA, HLIFB
      WRITE (LUN(5),320) HLIFA, HLIFB
320   FORMAT (/5X, 'RADIOACTIVE DECAY PROCESS ENABLED: ',/7X,
     +  '  PARENT HALF-LIFE = ', F7.1, ' SECONDS', /7X,
     +  'DAUGHTER HALF-LIFE = ', F7.1, ' SECONDS'/)
      RETURN
      END
```

Subroutine RDKIN (cont'd)

DATA INPUT/OUTPUT SUBROUTINES

After model initialization, MESOI Version 2.0 uses 10 subroutines for data input and for output of the results of simulation. These subroutines are listed, along with their memory requirements and a brief description of their use, in Table 10. A description and listing of each of the subroutines is presented in this section. The subroutines are discussed in the order in which they are used.

DATRD

This is the meteorological data input module. It is called once from INIT and then from MESOI at regular intervals. The calling program passes an index (LUINDX) which indicates which of the data files (observation or forecast)is to be accessed. The subroutine consists primarily of three READ statements; one for each of the data records in the standard data block. If the index indicates that observation data is being input, a fourth READ is used to check the date of the next data block. If an 88888 code is detected, MESOI assumes that observational data is exhausted and the index is altered to make DATRD use the forecast data file. DATRD will cause the model to stop if any of the READ statements detects an error or an EOF.

The model is set up to handle either hourly or 15 minute data. This is specified in INIT and is used by the model as the interval flag (INTFLG). When the flag is set (INTFLG = 1) the data is expected to be at 15 minute intervals. Both observation and forecast data file have the same record structure. Each time period is composed of three records. A sample of the data is shown below:

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789
         1         2         3         4         5         6         7
-------------------------------------------------------------------------------
 83112 1 0 1 4  30 0 2703 2703 2703 2703 2703 2703 2703 2703 2703 2703 2703
 83112 1 0 2 20   25    2703 2703 2703 2703 2703 2703 2703 2703 2703 2703
 83112 1 0 3            2703 2703 2703 2703 2703 2703 2703 2703 2703 2703
```

The following list applies to elements contained in each of the three records.

TABLE 10. Data Input/Output Subroutines

		SIZE (BYTES)		
NAME	CODE	LOCAL DATA	COMMON DATA	USE
DATRO	778	245	252	Reads meteorological data records from data file.
DATWR	342	194	252	Writes meterological data to output files.
ARRTIM	748	78	36732	Monitors time-integrated concentrations for puff arrival, writes messages when thresholds exceeded.
PRINTE	452	955	34847	Writes time-integrated concentrations and surface concentrations to disk files.
TESTM	444	300	28068	Writes out puff information at end of advection step if requested (primarily for model performance evaluation).
SCREEN	847	1756	35259	Interactive display of time-integrated concentration and surface contamination at end each hour--for alpha-numeric terminal.
PLOTZ	6221	5668	28667	Provides puff position plots for graphics terminal and plotter.
WNDFLD	925	1360	25755	Provides terrain adjusted surface wind field plots for graphics terminal and plotter.
WNDPLT	1017	1370	25755	Provides observed surface wind plots for graphics terminal and plotter.
EXPSUM	934	353	36220	Provides summary of check point status at end of simulation.

Record	Columns	Parameter	Comments
1	14	Stability	1 - 7
1	16-18	Mixing depth	in tens of meters
1	20	Precipitation code	0 - 6
1	22-25	Upper level wind	DDFF
2	15-16	Air temperature	first level
2	19-20	Air temperature	second level

These elements are contained in only one of the records.

Columns	Parameters	Comments
2-3	YEAR	last two digits only
4-6	Julian date	
7-10	time	HHMM
12	record number	1,2 or 3 used to check continuity of records
27-30	wind data	station 1, 11 or 21; DDFF
.		
.		
72-75	wind data	station 10, 20 or 30; DDFF

Program listing 11 contains the DATRD code.

DATWR

This subroutine writes the meteorological data to the model summary file (MESOUT.DAT). It is accessed from MESOI and INIT after they have called DATRD. The purpose of the module is to record the actual data used in the simulation along with the other pertinent information.

The module is a series of formatted WRITE statements which output a heading, labels and the current data. A sample output as it would appear in the output file is shown below:

```
DATA FOR JULIAN DAY:  83 112 AT 12:30

 STABILITY CLASS = 2  MIXING DEPTH = 1000 METERS  PRECIP CODE = 0  AIR TEMPS(C) = 20 AND 25
                                                                  /stations\
 WINDS    2702  2702 2702 2702 2702 2702 2702 2702 2702 2702 2702 (  1 - 10 )
                2702 2702 2702 2702 2702 2702 2702 2702 2702 2702 ( 11 - 20 )
           ↑    2702 2702 2702 2702 2702 2702 2702 2702 2702 2702  \21 - 30/
      (upper level)
          wind
```

Winds are in the same format as the original data, DDFF. Direction is in tens of degrees and speed is in the units of choice.

Program listing 12 contains the DATWR code.

ARRTIM

This module monitors the exposure level at the checkpoints and issues a message whenever a threshold is exceeded. ARRTIM is called from MESOI at the end of each advection step and only if the checkpoint flag (ARRFL) is set. The program is primarily a loop to cycle thru the ARRIVL array checking each point. For each definition, the status is checked to see if the point is active and if the thresholds have both been exceeded. If either is true, the point is not checked further. If they are false, then the point location is used to specify a location on the exposure grid. This cumulative value is compared to the appropriate threshold. If the exposure exceeds the threshold, the date and time are entered into the ARRIVL array and a message is printed in both the output file and on the users terminal. Refer to the discussion of the ARRIVN module for a description of the various elements of the ARRIVL array.

Program listing 13 contains the ARRTIM code.

PRINTE

This subroutine handles the writing of the various exposure and deposition arrays to the output files. PRINTE is called from MESOI at the end of each advection step. To simplify the module, each of the arrays to be output is equivalenced to a section of the MATR array. Then, a simple loop can be used to write out the various sections to their respective files. The character arrays MATRID and MATUNT contain the descriptive title and units to match each section. The array LUN defines the files to which the various sections are output. If the deposition flag (DPFLG) has not be set, the output of the 4 deposition arrays (ADEPOS, BDEPOS, ADPADV and BDPADV) is skipped.

Program listing 14 contains the PRINTE code.

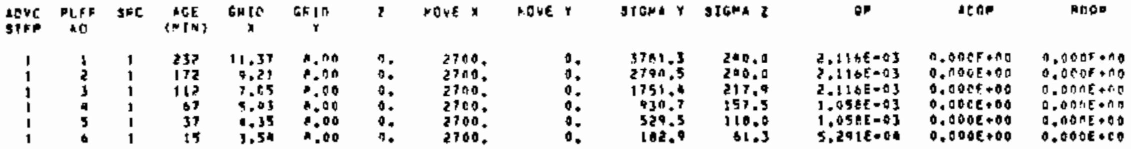
TESTM

This module is used primarily in checking the model execution. It is designed to output significant information about the puffs to enable the user to verify the various processes of transport, diffusion, decay and deposition. TESTM is called from MESOI at the end of each advection step and only if the option has been enabled and puffs have been released. The user enables the testing mode by using an asterisk (*) as the first character of the run title.

TESTM lists the following information:

1) number of the advection step
2) puff number
3) source number for that puff
4) age of the puff in minutes
5) x and y coordinates of the puff center in grid units
6) x and y movement of the puff that advection step in meters
7) sigma y and sigma z for the puff
8) undepleted puff mass (QP)
9) mass of parent and daughter (ADQP and BDQP)

A sample of TESTM output is shown below:



```
ADVC  PUFF  SRC   AGE   GRID   GRID    Z    MOVE X    MOVE Y     SIGMA Y  SIGMA Z        QP         ADQP        BDQP
STEP   NO        (MIN)   X      Y
   1     1    1   232   11.37   8.00   0.    2700.         0.    3781.3    240.0   2.116E-03  0.000E+00  0.000E+00
   1     2    1   172    9.21   8.00   0.    2700.         0.    2790.5    240.0   2.116E-03  0.000E+00  0.000E+00
   1     3    1   112    7.65   8.00   0.    2700.         0.    1751.4    217.9   2.116E-03  0.000E+00  0.000E+00
   1     4    1    67    5.43   8.00   0.    2700.         0.     930.7    157.5   1.058E-03  0.000E+00  0.000E+00
   1     5    1    37    4.35   8.00   0.    2700.         0.     529.5    110.0   1.058E-03  0.000E+00  0.000E+00
   1     6    1    15    3.54   8.00   0.    2700.         0.     182.9     61.3   5.291E-04  0.000E+00  0.000E+00
```

Program Listing 15 contains the TESTM code.

SCREEN

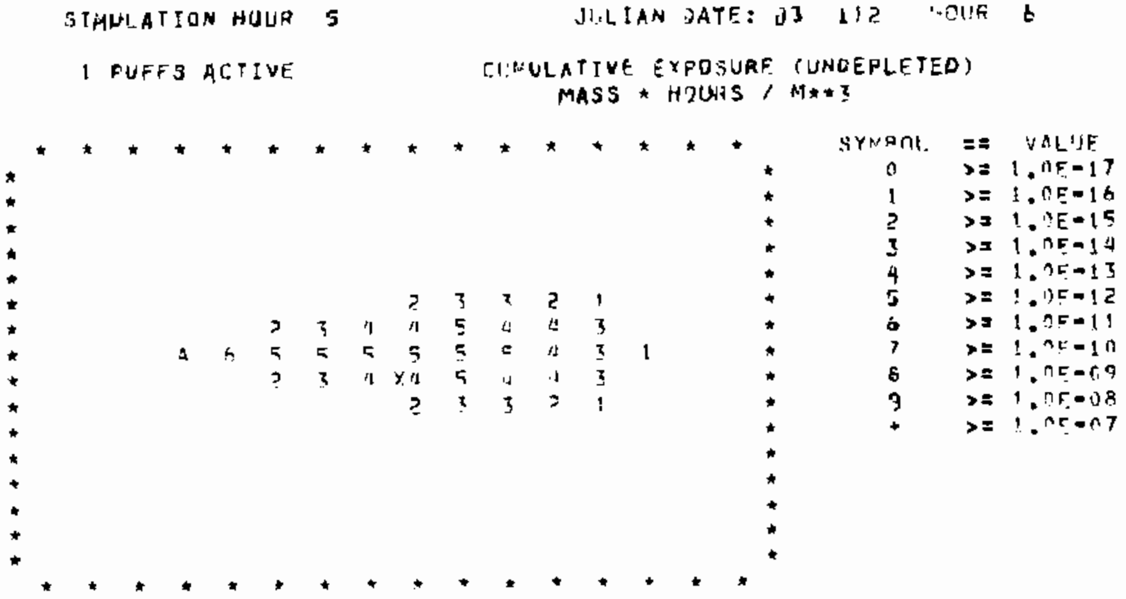
SCREEN provides the user with simple alphanumeric graphic representations of the exposure and deposition matrices. It is accessed only when requested by the user during output option selection in MESOI.

To keep the module small and general, the arrays that need to be available for display are equivalenced to portions of the larger array MATR. Thus, the proper matrix for display can be selected using an array subscript in MATR. SCREEN begins by presenting a menu on the user's display terminal:

```
WHICH CUMULATIVE MATRIX DO WISH TO DISPLAY?

          1 = TOTAL EXPOSURE
          2 = DEPLETED PARENT
          3 = DEPLETED DAUGHTER
          6 = DEPOSITION - PARENT
          7 = DEPOSITION - DAUGHTER

          0 = RETURN AND CONTINUE
```

Unrecognized responses are ignored and the menu presented again. The only exit from SCREEN is via the 0 option to 'RETURN and CONTINUE'. The module does not monitor the deposition flag and any matrix can be displayed whether it is being used or not by the model.

A sample of the output from SCREEN is shown below:



```
 SIMULATION HOUR  5                  JULIAN DATE: 83  112   HOUR  6

  1 PUFFS ACTIVE                 CUMULATIVE EXPOSURE (UNDEPLETED)
                                      MASS * HOURS / M**3

   *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *        SYMBOL  ==  VALUE
*                                                  *        0     >= 1.0E-17
*                                                  *        1     >= 1.0E-16
*                                                  *        2     >= 1.0E-15
*                                                  *        3     >= 1.0E-14
*                                                  *        4     >= 1.0E-13
*                          2  3  3  2  1           *        5     >= 1.0E-12
*                 2  3  4  4  5  4  4  3           *        6     >= 1.0E-11
*           A  6  5  5  5  5  5  5  4  3  1        *        7     >= 1.0E-10
*                 2  3  4 X4  5  4  4  3           *        8     >= 1.0E-09
*                          2  3  3  2  1           *        9     >= 1.0E-08
*                                                  *        +     >= 1.0E-07
*                                                  *
*                                                  *
*                                                  *
*                                                  *
*                                                  *
   *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *
```

The grid is 16 by 16 and represents the odd numbered nodes of the 31 by 31 matrices. Letters are used to specify location. X is the grid center and is given for visual reference only. The A - D are used to define the source locations. The letters are placed at the point closest to their actual location on the grid. The numbers represent the values in the selected matrix at the time that the module was called. The key which defines the categories and the headings used for each display are set up in DATA statements. If more than one letter or number is in a given character space, priority is given first to the letters and then to the numbers.

Program listing 16 contains the SCREEN code.

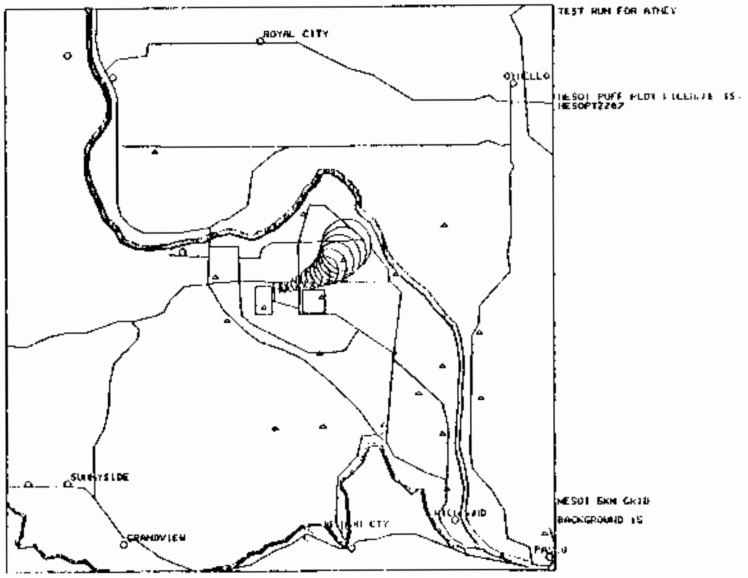
PLOTZ

PLOTZ is a graphics module designed to produce a plot representing the active puffs. It can be used to easily visualize the distribution of material on the grid, especially when overlayed with background maps.

PLOTZ uses a Pacific Northwest Laboratory (PNL) version of CALCOMP graphics software. It also has some statements which transmit octal numbers (represented in VAX FORTRAN by the prefix") to control a VISUAL 500 terminal. However, the module should be easily modified to match other software and hardware configurations.

The subroutine is called from MESOI whenever the 'P' output option is selected. It begins by drawing a square display border and a title header. Source locations are translated from grid coordinates to inches from the plot origin and a symbol is drawn at each source. The puffs are drawn inside a loop which translates their centers into inches and computes a series of 36 line segments to approximate a circle around the centers at a given radius. The radii (RADP) are calculated in the DIFDEP module and can be adjusted to show that portion of a puff which is considered significant. The plotting loop draws only those puffs which are active (i.e. MF = 1).

An example of PLOTZ output overlayed on a map background is shown below.



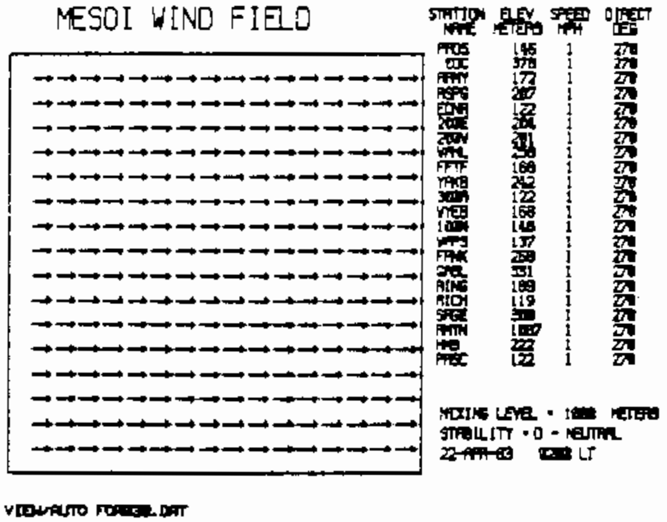
Program Listing 17 contains the PLOTZ code.

WNDFLD

This is a graphics module designed to generate a display of the model wind field. WNDFLD is called from MESOI whenever the "W" option is selected as an output option. The program uses a PNL version of the CALCOMP graphics software. It creates a file which can be displayed on graphics output devices. Opening and closing of this file is done within the graphics package. Use of this module will be system dependent and will require linking the program to the needed libraries.

WNDFLD begins by drawing a display border and headings. Then the arrays containing the station names, elevations and winds are used to list the current meteorology. Finally, the wind component arrays (UU and VV) are used to create an arrow to represent the direction and magnitude of the flow at each grid point. The base of the arrow is located at the grid point and the arrow length is directly related to wind speed.

A sample of the graphic product generatea by WNDFLD is shown below:



STATION NAME	ELEV METERS	SPEED MPH	DIRECT DEG
PROS	145	1	270
DOE	376	1	270
ARMY	172	1	270
RSPG	207	1	270
EDNA	122	1	270
200E	204	1	270
200W	201	1	270
WAHL	250	1	270
FFTF	168	1	270
YAKB	242	1	270
300A	122	1	270
WYEB	168	1	270
100N	146	1	270
WPPS	137	1	270
FRNK	250	1	270
SPRL	331	1	270
RING	189	1	270
RICH	119	1	270
SAGE	300	1	270
RMTN	1087	1	270
HMS	222	1	270
PASC	122	1	270

MIXING LEVEL = 1000 METERS
STABILITY = D - NEUTRAL
22-APR-83 [illegible] LT

VIEW/AUTO FOR[illegible].DAT

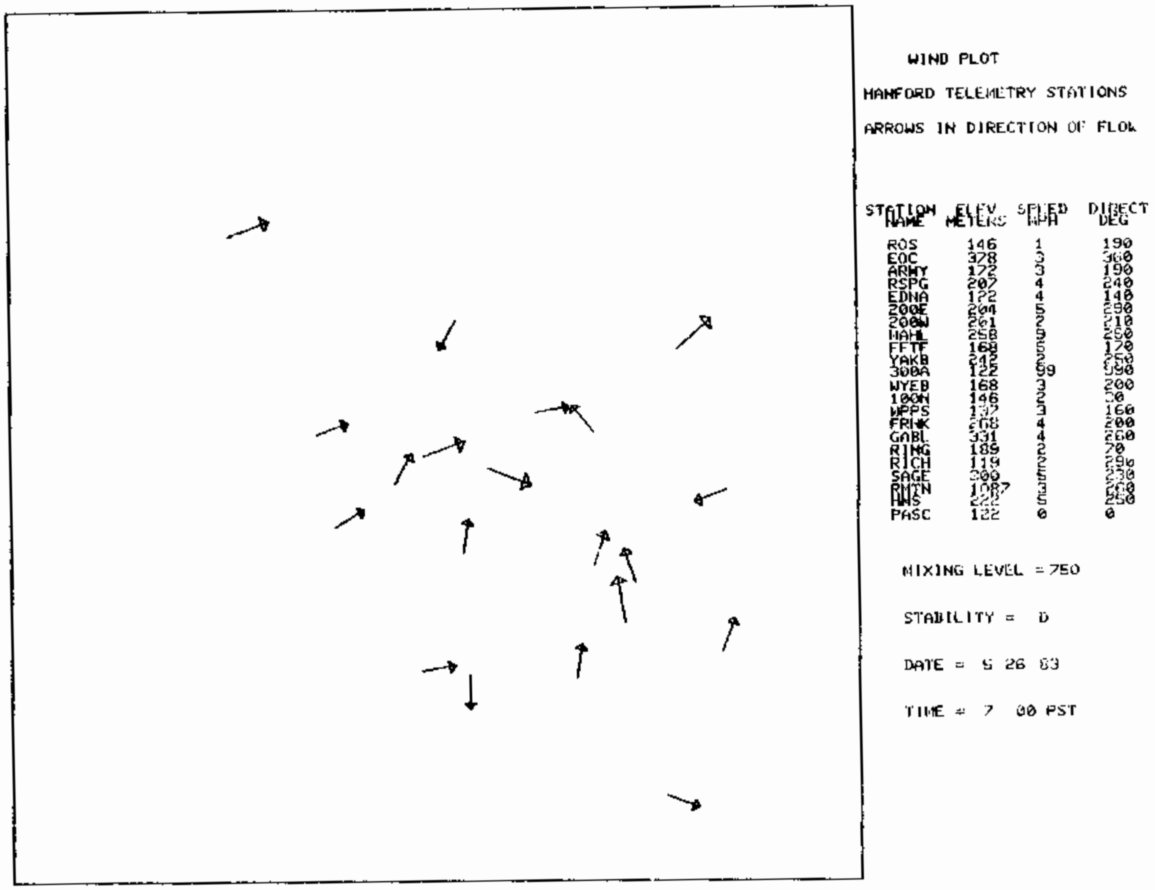
Program listing 18 contains the WNDFLD code.

WNDPLT

This is a graphics module designed to generate a display of the winds at the telemetry stations. WNDPLT is called from MESOI whenever the "V" option is selected as an output option. The program uses a PNL version of the CALCOMP graphics software. It creates a file which can be displayed on graphic output devices. Opening and closing of this file is done within the graphics package. Use of this module will be system dependent and will require linking the program to the needed libraries.

WNDPLT begins by drawing a display border and headings. Then the arrays containing the station names, elevations and winds are used to list the current meteorology. Finally, the station wind arrays (DIR and SPD) are used to create an arrow to represent the direction and magnitude of the flow at each station. The base of the arrow is located at the station location and the arrow length is directly related to wind speed.

A sample of the graphic product generated by WNDPLT is shown below.



```
     WIND PLOT

HANFORD TELEMETRY STATIONS

ARROWS IN DIRECTION OF FLOW


STATION  ELEV   SPEED  DIRECT
  NAME  METERS   MPH    DEG

  ROS     146     1      190
  EOC     378     3      360
  ARMY    172     3      190
  RSPG    207     4      240
  EDNA    172     4      140
  200E    204     5      290
  200W    261     2      210
  MAHL    258     9      250
  FFTF    168     5      170
  YAKB    242     2      250
  300A    122     99     990
  WYEB    168     3      200
  100N    146     2      50
  WPPS    137     3      160
  FRNK    268     4      200
  GABL    331     4      260
  RING    189     2      70
  RICH    119     2      290
  SAGE    300     5      230
  RMTN    1087    3      260
  HMS     222     5      250
  PASC    122     0      0

  MIXING LEVEL =750

  STABILITY =   D

  DATE =  5 26 83

  TIME =  7  00 PST
```

Program listing 19 contains the WNDPLT code.

EXPSUM

This module is called from MESOI whenever simulation is terminated and the checkpoint flag (ARRFL) has been set to indicate active checkpoints. EXPSUM is called before the MODEND subroutine and thus summarizes exposure only over a single simulation. The arrays used in checking are reset if the simulation is restarted. In the current model (Version 2.0), only the total cumulative air concentrations are monitored during the checking.

EXPSUM consists of a series of formatted WRITE statements which output headings and the contents of the ARRIVL array to both the user's terminal and model output file (MESOUT). When called, the subroutine first checks the EXPFLG to see if any checkpoints have been exceeded. If not (EXPFLG = 0) a message to that effect is printed and the subroutine returns.

If the flag is set, EXPSUM executes a loop to check the contents of the ARRIVL array and to print information for those checkpoints at which cumulative exposure exceeded at least one of the thresholds set in ARRIVN. Initially, the 3rd column of the array (ARRIVL(3,J)) is checked. If it is greater than one, the checkpoint is active. Next, the final cumulative exposure of an active checkpoint is determined. If it does not exceed the first threshold, no further reporting is done for that point. The intent is to summarize only those points which received significant exposure.

When the criterion are met for display, the contents of the ARRIVL array and the final exposure are written to the two files. A sample output is shown below:

```
EXPOSURE SUMMARY FOR MESOI --> *TEST FOR CHECKPOINTS


CHECKPOINT      EXP X,Y                 EXCEEDED                  FINAL
   NAME       COORDINATES     1.000E-15       1.000E-10     CONCENTRATION
                              DAY TIME        DAY TIME


 EAST 1          9.   17.     112   115       112   130          4.867E-09

 EAST 2         11.   17.     112   130       112   2 0          1.458E-09

 NE   0         17.   19.     112   3 0         0   0 0          4.477E-12

 200 EAST       15.   19.     112   245         0   0 0          1.438E-12

 200 WEST       12.   18.     112   2 0         0   0 0          1.035E-11

 GABLE MT       17.   19.     112   3 0         0   0 0          4.477E-12

 LANDFILL       17.   16.     112   245         0   0 0          7.241E-11

 ARMY LP        14.   16.     112   215         0   0 0          3.914E-11
```

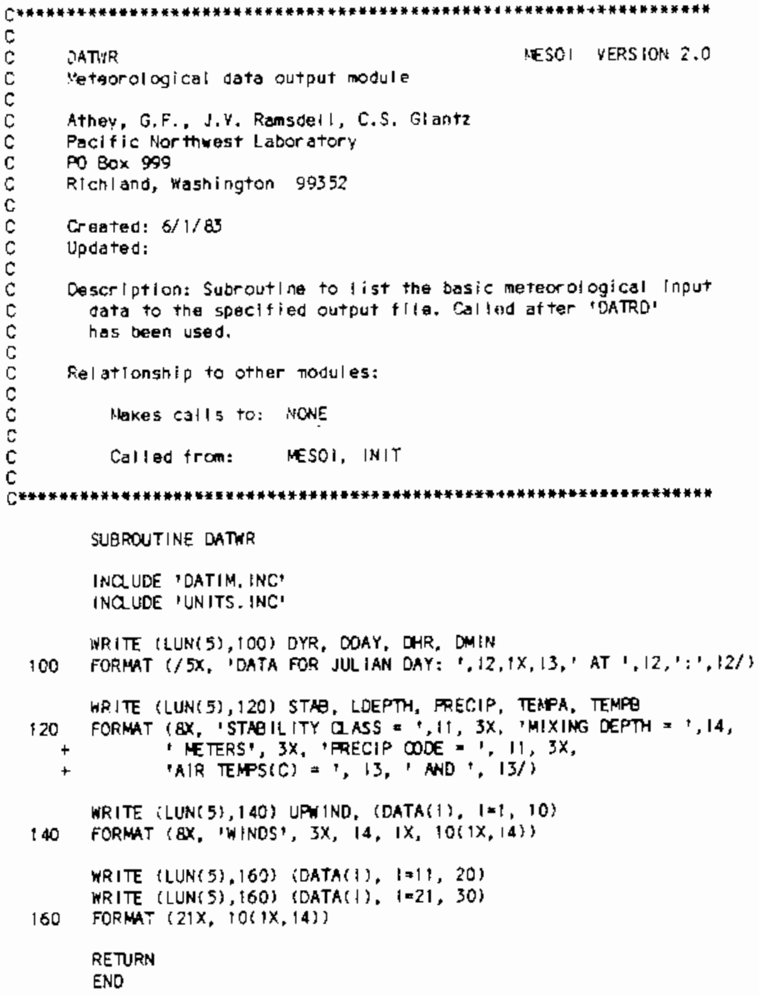
For each exposed checkpoint is listed:

checkpoint name
location in exposure grid units
date and time (to nearest advection step) when the thresholds were exceeded.
final concentration at end of simulation.

Program listing 20 contains the EXPSUM code.

```
C*****************************************************************
C
C     DATRD                                    MESOI  VERSION 2.0
C     Meteorological data input module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to read the meteorological data. The
C       calling program passes the unit number on which the observation
C       or forecast information is located.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI, INIT
C
C*****************************************************************

      SUBROUTINE DATRD (LUINDX, SHR, IADV)

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

      IREAD = 1
  100 READ (LUN(LUINDX), 200, END=900, ERR=800, IOSTAT=IER) DYR,
     +  DDAY, DHR, DMIN, DREC, STAB, LDEPTH, PRECIP, UPWIND,
     +  (DATA(I),I=1,10)
  200 FORMAT (1X,I2,I3,I2,I2,1X,I1,1X,I1,1X,I3,1X,I1,11(1X,I4))

      IREAD = 2
      READ (LUN(LUINDX), 210, END=900, ERR=800, IOSTAT=IER) TEMPA,
     +  TEMPB, (DATA(I), I=11,20)
  210 FORMAT (13X, I3, 1X, I3, 5X, 10(1X,I4))

      IREAD = 3
      READ (LUN(LUINDX), 220, END=900, ERR=800, IOSTAT=IER)
     +  (DATA(I),I=21,30)
  220 FORMAT (25X, 10(1X,I4))

      IF (LUINDX .EQ. 3)  THEN
        IREAD = 4
        READ (LUN(LUINDX), 230, ENO=900, ERR=800, IOSTAT=IER) NXTDAY
  230   FORMAT(3X,I3)
        BACKSPACE LUN(LUINOX)
      ENDIF
      RETURN

  800 PRINT *, 'DATRD >> ERROR ',IER, 'AT RECORD ',IREAD,
     +  ' ON UNIT ',LUN(LUINDX)
      PRINT *, '     SIMULATION HOUR ',SHR, '    ADVECTION STEP = ',IADV
      STOP

  900 PRINT *, 'DATRD >> EOF AT RECORD ',IREAD, '  UNIT ', LUN(LUINDX)
      PRINT *, '     SIMULATION HOUR ',SHR, '    ADVECTION STEP = ',IADV
      STOP
      END
```

Program Listing 11 Subroutine DATRD



```
C*********************************************************************
C
C     DATWR                                         MESOI  VERSION 2.0
C     Meteorological data output module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to list the basic meteorological input
C       data to the specified output file. Called after 'DATRD'
C       has been used.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI, INIT
C
C*********************************************************************

      SUBROUTINE DATWR

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

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

      WRITE (LUN(5),120) STAB, LDEPTH, PRECIP, TEMPA, TEMPB
  120 FORMAT (8X, 'STABILITY CLASS = ',I1, 3X, 'MIXING DEPTH = ',I4,
     +        ' METERS', 3X, 'PRECIP CODE = ', I1, 3X,
     +        'AIR TEMPS(C) = ', I3, ' AND ', I3/)

      WRITE (LUN(5),140) UPWIND, (DATA(I), I=1, 10)
  140 FORMAT (8X, 'WINDS', 3X, I4, 1X, 10(1X,I4))

      WRITE (LUN(5),160) (DATA(I), I=11, 20)
      WRITE (LUN(5),160) (DATA(I), I=21, 30)
  160 FORMAT (21X, 10(1X,I4))

      RETURN
      END
```

Program Listing 12 Subroutine DATWR

```
C******************************************************************
C
C     ARRTIM                                       MESOI  VERSION 2.0
C     Monitor checkpoint thresholds
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to report the arrival of transported
C       material at specified checkpoints on the exposure grid.
C       The points are checked at the end of each advection step.
C       A report is made when THRSH1 or THRSH2 is exceeded.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C******************************************************************

      SUBROUTINE ARRTIM (TINC)

      INCLUDE 'ARRIV.INC'
      INCLUDE 'CONST.INC'
      INCLUDE 'DATIM.INC'
      INCLUDE 'MATRIX.INC'
      INCLUDE 'UNITS.INC'

      INTEGER MIN

      MIN = TINC * 60
      DO 500 J = 1, 35
        IF (ARRIVL(3,J) .EQ. 0)  GOTO 500
        IF (ARRIVL(3,J) .GE. 3)  GOTO 500

        INDX = ARRIVL(1,J) + 0.5
        JNDX = ARRIVL(2,J) + 0.5

        IF (ARRIVL(3,J) .EQ .2)  GOTO 250

        IF (EXPCUM(INDX,JNDX) .LT. THRSH1)  GOTO 500

          EXPFLG = 1
          ARRIVL(3,J) = 2
          WRITE (LUN(2),100) PREVDY, PREVHR, MIN, ARRIVL(4,J),
     +                   ARRIVL(5,J), THRSH1
          WRITE (LUN(5),100) PREVDY, PREVHR, MIN, ARRIVL(4,J),
     +                   ARRIVL(5,J), THRSH1
  100     FORMAT (/5X, 'JDAY ', I3, ' AT ', I2, ':', I2,
     +      ' EXPOSURE AT ', 2A4, ' EXCEEDS ', 1PE10.3/)
          ARRIVL(6,J) = PREVDY
          ARRIVL(7,J) = PREVHR
          ARRIVL(8,J) = MIN
          GOTO 500

  250   IF (EXPCUM(INDX,JNDX) .LT. THRSH2)  GOTO 500

          ARRIVL(3,J) = 3
          WRITE (LUN(2),100) PREVDY, PREVHR, MIN, ARRIVL(4,J),
```

Program Listing 13 Subroutine ARRTIM

```
   +                 ARRIVL(5,J), THRSH2
        WRITE (LUN(5),100) PREVDY, PREVHR, MIN, ARRIVL(4,J),
   +                 ARRIVL(5,J), THRSH2
        ARRIVL(9,J)  * PREVDY
        ARRIVL(10,J) = PREVHR
        ARRIVL(11,J) = MIN

500 CONTINUE
    RETURN
    END
```

Subroutine ARRTIM (cont'd)

```
C*****************************************************************
C
C     PRINTE                                      MESOI  VERSION 2.0
C     Output of printer compatible files
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/85
C     Updated:
C
C     Description: Handles the output of the matrices to separate
C           disk files.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE PRINTE (SHR, IADV, DDAY, DHR, DMIN, IPSUM, DPFLG)

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

      REAL MATR(31, 31, 9)

      INTEGER SHR, DDAY, DHR, DMIN, IPLUS, DPFLG

      EQUIVALENCE (MATR(1,1,1), EXPCUM)
      EQUIVALENCE (MATR(1,1,2), AXPDPL)
      EQUIVALENCE (MATR(1,1,3), BXPDPL)
      EQUIVALENCE (MATR(1,1,4), AXPADV)
      EQUIVALENCE (MATR(1,1,5), BXPADV)
      EQUIVALENCE (MATR(1,1,6), ADEPOS)
      EQUIVALENCE (MATR(1,1,7); BDEPOS)
      EQUIVALENCE (MATR(1,1,8), ADPADV)
      EQUIVALENCE (MATR(1,1,9), BDPADV)

      CHARACTER*50 MATRID(9)
      CHARACTER*20 MATUNT(9)

      DATA MATRID /
     +   'CUMULATIVE EXPOSURE (UNDEPLETED)                 ',
     +   'DEPLETED PARENT -- CUMULATIVE EXPOSURE           ',
     +   'DEPLETED DAUGHTER -- CUMULATIVE EXPOSURE         ',
     +   'DEPLETED PARENT -- ADVECTION PERIOD EXPOSURE     ',
     +   'DEPLETED DAUGHTER -- ADVECTION PERIOD EXPOSURE   ',
     +   'PARENT DEPOSITION -- CUMULATIVE                  ',
     +   'DAUGHTER DEPOSITION -- CUMULATIVE                ',
     +   'PARENT DEPOSITION -- ADVECTION PERIOD            ',
     +   'DAUGHTER DEPOSITION -- ADVECTION PERIOD          '/

      DATA MATUNT /
     +   'MASS * HOURS / M**3 ',
     +   'MASS * HOURS / M**3 ',
     +   'MASS * HOURS / M**3 ',
     +   'MASS * HOURS / M**3 ',
     +   'MASS * HOURS / M**3 ',
     +   'MASS / M**2         ',
```

Program Listing 14 Subroutine PRINTE

```
     +   'MASS / M**2          ',
     +   'MASS / M**2          ',
     +   'MASS / M**2          '/

      LUINDX = 6
      DO 200 N = 1, 9
        IF (DPFLG .EQ. 0 .AND. N .EQ. 6)  RETURN

        IPLUS = 0
        WRITE (LUN(LUINDX),100) TITLE, RDATE, RTIME
100     FORMAT ('1', 4X, 50A1, /60X, A8, 2X, A8/)
        WRITE (LUN(LUINDX),110) MATRID(N), MATUNT(N)
110     FORMAT (/5X, A50, 2X, A20/)
        WRITE (LUN(LUINDX),120) SHR, IADV, DDAY, DHR, DMIN, IPSUM
120     FORMAT (5X, 'SIMULATION HOUR ',I2, '  ADVECTION STEP ', I2,
     +    5X, 'DATA FOR  DAY ', I3, 3X,' AT ', I2, ':', I2/,
     +     7X, I4, ' PUFFS ACTIVE'/)

        DO 160 L = 1, 3
          WRITE (LUN(LUINDX),130) (I+IPLUS,I=1,11)
130       FORMAT (1H0,' Y COORDINATE OF',25X,' X COORDINATE ',
     +      'OF THE GRID POINTS', /2X, ' THE GRID POINTS', I4, 11I10/)

          DO 150 J = 1, 31
            WRITE (LUN(LUINDX),140) 32-J, (MATR(I,32-J,N),
     +               I=1+IPLUS,11+IPLUS)
140         FORMAT (I11, 5X, 1P11E10.2)
150       CONTINUE

          IPLUS = IPLUS + 11
160     CONTINUE
        LUINDX = LUINDX + 1

200   CONTINUE
      RETURN
      END
```

Subroutine PRINTE (cont'd)

```
C******************************************************************
C
C     TESTM                                        MESOI  VERSION 2.0
C     Testing mode output module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to print additional puff information
C           when the model is run in the testing mode.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C******************************************************************

        SUBROUTINE TESTM (IADV, SHR, DELXY)

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

        INTEGER SHR

        IF (IADV .EQ. 1 .OR. IFIRST .NE. 1)  THEN
          WRITE (LUN(5),100) SHR
 100      FORMAT ('1', /10X, 'TESTING MODE -- PUFF DESCRIPTORS',
     +       ' FOR SIMULATION HOUR ', I2)
          WRITE (LUN(5),110)
 110      FORMAT (//1X, 'ADVC  PUFF  SRC   AGE   GRID   GRID     Z',
     +       4X,'MOVE X', 4X, 'MOVE Y', 5X, 'SIGMA Y  SIGMA Z', 9X, 'QP',
     +       9X, 'ADQP', 8X, 'BDQP')
          WRITE (LUN(5),120)
 120      FORMAT (1X, 'STEP   NO', 8X, '(MIN)', 4X, 'X', 6X, 'Y')
        ENDIF

        WRITE (LUN(5),130)
 130    FORMAT (1X)

        DO 200 M = 1, TPUFFS

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

          WRITE (LUN(5),150) IADV, M, SP(M), AGE(M), XP(M), YP(M),
     +       ZP(M), XMOVE, YMOVE, SIGMAY(M), SIGMAZ(M), QP(M),
     +       ADQP(M), BDQP(M)
 150      FORMAT (2X,I2, 3X,I3, 4X,I1, 2X,I5, 3X,F5.2, 2X,F5.2,
     +       2X,F4.0, 2(3X,F7.0), 2X, 2(2X,F7.1), 2X, 3(2X,1PE10.3))
 200    CONTINUE

        IFIRST = 1
        RETURN
        END
```

Program Listing 15 Subroutine TESTM

```
C**********************************************************************
C
C     SCREEN                                           MESOI  VERSION 2.0
C     Output module for visual display terminal
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Any of the cumulative matrices can be displayed in
C       simple graphical form using alphanumeric characters.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C**********************************************************************

      SUBROUTINE SCREEN ( SHR, IPSUM )

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

      REAL VALUE(11,9), MATR(31,31,9)

      INTEGER SHR

      CHARACTER*1 DISPL(50), SORSYM(4), SYMBOL(11)
      CHARACTER*3 BORDER(16)
      CHARACTER*35 HEADR1(9), HEADR2(9)

      EQUIVALENCE (MATR(1,1,1), EXPCUM)
      EQUIVALENCE (MATR(1,1,2), AXPDPL)
      EQUIVALENCE (MATR(1,1,3), BXPDPL)
      EQUIVALENCE (MATR(1,1,4), AXPADV)
      EQUIVALENCE (MATR(1,1,5), BXPADV)
      EQUIVALENCE (MATR(1,1,6), ADEPOS)
      EQUIVALENCE (MATR(1,1,7), BDEPOS)
      EQUIVALENCE (MATR(1,1,8), ADPADV)
      EQUIVALENCE (MATR(1,1,9), BDPADV)

      DATA BORDER/16*' * '/
      DATA SYMBOL/'0','1','2','3','4','5','6','7','8','9','+'/
      DATA SORSYM/'A','B','C','D'/

C   **  DEFINE THE BINS FOR CODING OF DISPLAY CHARACTERS FOR EACH OF
C   **     THE FIVE ARRAYS

      DATA ( VALUE(I,1),I=1,11 ) / 1.0E-17, 1.0E-16, 1.0E-15, 1.0E-14,
     +  1.0E-13, 1.0E-12, 1.0E-11, 1.0E-10, 1.0E-09, 1.0E-08, 1.0E-07/

      DATA ( VALUE(I,2),I=1,11 ) / 1.0E-17, 1.0E-16, 1.0E-15, 1.0E-14,
     +  1.0E-13, 1.0E-12, 1.0E-11, 1.0E-10, 1.0E-09, 1.0E-08, 1.0E-07/

      DATA ( VALUE(I,3),I=1,11 ) / 1.0E-17, 1.0E-16, 1.0E-15, 1.0E-14,
     +  1.0E-13, 1.0E-12, 1.0E-11, 1.0E-10, 1.0E-09, 1.0E-08, 1.0E-07/
```

Program Listing 16 Subroutine SCREEN

```
      DATA ( VALUE(I,6),I=1,11 ) / 1.0E-17, 1.0E-16, 1.0E-15, 1.0E-14,
     +  1.0E-13, 1.0E-12, 1.0E-11, 1.0E-10, 1.0E-09, 1.0E-08, 1.0E-07/

      DATA ( VALUE(I,7),I=1,11 ) / 1.0E-17, 1.0E-16, 1.0E-15, 1.0E-14,
     +  1.0E-13, 1.0E-12, 1.0E-11, 1.0E-10, 1.0E-09, 1.0E-08, 1.0E-07/


      DATA HEADR1(1) /' CUMULATIVE EXPOSURE (UNDEPLETED)  '/
      DATA HEADR1(2) /' DEPLETED PARENT - CUMUL EXPOSURE  '/
      DATA HEADR1(3) /'DEPLETED DAUGHTER - CUMUL EXPOSURE '/
      DATA HEADR1(6) /'  PARENT DEPOSITION - CUMULATIVE   '/
      DATA HEADR1(7) /'DAUGHTER DEPOSITION - CUMULATIVE   '/

      DATA HEADR2(1) /'MASS * HOURS / M**3                '/
      DATA HEADR2(2) /'MASS * HOURS / M**3                '/
      DATA HEADR2(3) /'MASS * HOURS / M**3                '/
      DATA HEADR2(6) /'MASS / M**2                        '/
      DATA HEADR2(7) /'MASS / M**2                        '/

  100 WRITE (LUN(2),110)
  110 FORMAT (/5X, 'WHICH CUMULATIVE MATRIX DO YOU WISH TO DISPLAY?',
     +       /7X, '1 = TOTAL EXPOSURE '
     +       /7X, '2 = DEPLETED - PARENT',
     +       /7X, '3 = DEPLETED - DAUGHTER',
     +       /7X, '6 = DEPOSITION - PARENT',
     +       /7X, '7 = DEPOSITION - DAUGHTER',
     +      //7X, '0 = RETURN AND CONTINUE'/)

      READ (LUN(1),120,ERR=100) ISEL
  120 FORMAT (I1)
      IF (ISEL .LT. 0 .OR. ISEL .GT. 7) GOTO 100
      IF (ISEL .EQ. 4 .OR. ISEL .EQ. 5)  THEN
        PRINT *, '  PRODUCT NOT AVAILABLE FOR SCREEN DISPLAY'
        GOTO 100
      ENDIF
      IF (ISEL .EQ. 0)  RETURN

C   **  SCREEN DISPLAY OF SELECTED HOURLY OUTPUT

      WRITE (LUN(2),130) TITLE, RDATE, RTIME
  130 FORMAT (5X,50A1,/,60X,A8,2X,A8,/)
      WRITE (LUN(2),140) SHR, DYR, DDAY, DHR
  140 FORMAT (5X,'SIMULATION HOUR ',I2,15X,'JULIAN DATE: ',I2,2X,
     +     I3,3X,'HOUR ', I2)
      WRITE (LUN(2),150) IPSUM, HEADR1(ISEL), HEADR2(ISEL)
  150 FORMAT (/3X, I4, ' PUFFS ACTIVE', 11X, A35, /37X, A35)

      WRITE (LUN(2),160) BORDER
  160 FORMAT (2X, 16A3, 5X, 'SYMBOL  ==  VALUE')
      DISPL(1) = '*'
      DISPL(50) = '*'

      DO 230 N = 1, 16
        DO 170 I = 2, 49
          DISPL(I) = ' '
  170   CONTINUE

        DO 190 I = 1, 16
          IF (MATR(I*2-1, 33-(N*2), ISEL) .LT. VALUE(1,ISEL)) GOTO 190
          DO 180 K = 2, 11
            IF (MATR(I*2-1, 33-(N*2), ISEL) .LT. VALUE(K,ISEL))  THEN
              DISPL(I*3) = SYMBOL(K-1)
              GOTO 190
            ENDIF
  180     CONTINUE
```

Subroutine SCREEN (cont'd)

```
          DISPL(I*3) = SYMBOL(11)
  190   CONTINUE
C     **  PUT CHARACTERS 'A,B,C OR D' AT RELEASE POINTS
C         PUT CHARACTER 'X' AT GRID CENTER

        DO 200 I = 1, NSOURC
          IF (IFIX(YSOURC(I)) .EQ. 16-N)  THEN
            DISPL(IFIX((XSOURC(I)+1.0)*3+0.5)) = SORSYM(I)
          ENDIF
  200   CONTINUE
        IF (N .EQ. 9) DISPL(26) = 'X'
        IF (N .LT. 12)  THEN
          WRITE (LUN(2),210) DISPL, SYMBOL(N), VALUE(N,ISEL)
  210     FORMAT (1X, 50A1, 7X, A1, 4X, '>=', 1X, 1PE7.1)
        ELSE
          WRITE (LUN(2), 220) DISPL
  220     FORMAT (1X, 50A1)
        ENDIF

  230 CONTINUE
      WRITE (LUN(2),240) BORDER
  240 FORMAT (2X,16A3)
      PAUSE 'TYPE C TO CONTINUE'
      GOTO 100
      END
```

Subroutine SCREEN (cont'd)

```
C*****************************************************************
C
C     PLOTZ                                        MESOI  VERSION 2.0
C     Puff plotting module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: This subroutine allows display of the active puffs
C       on the grid. The subroutine must be tailored to the computer
C       hardware available. This module uses a Battelle version of
C       CALCOMP and is designed to utilize a VISUAL 500 terminal.
C
C     Relationship to other modules:
C
C         Makes calls to:  DTCHAR
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE PLOTZ

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

      CHARACTER*9 DDATE, DTIME

      DTR = 0.0174532925

      CALL DTCHAR (DYR, DDAY, DHR, DMIN, DDATE, DTIME)

C   **  SET VISUAL 500 TERMINAL TO ALPHAGRAPHICS MODE

      IPLT = 0
      WRITE (LUN(1),10) "37
   10 FORMAT ('+', 5A1)


C   **  INITIALIZE PLOTFILE, SET SIZE, DRAW A BORDER AND HEADERS

      CALL PLOTS (0.0, 0.0, LUN(16))
      CALL FACTOR (1.0)

      CALL PLOT (0.0, 0.0, 3)
      CALL PLOT (8.5, 0.0, 2)
      CALL PLOT (8.5, 8.5, 2)
      CALL PLOT (0.0, 8.5, 2)
      CALL PLOT (0.0, 0.0, 2)

      CALL SYMBOL (1.0, 9.0, 0.42, 'MESOI PUFF PLOT ', 0.0, 16)
      CALL SYMBOL (9.2, 8.0, 0.21, DDATE, 0.0, 9)
      CALL SYMBOL (9.2, 7.5, 0.21, DTIME, 0.0, 9)

C   **  PUT A SYMBOL AT EACH SOURCE LOCATION

      DO 200 I = 1, NSOURC
```

Program Listing 17 Subroutine PLOTZ

```
          XSP = XSOURC(I) * 0.5667
          YSP = YSOURC(I) * 0.5667
          CALL SYMBOL (XSP, YSP, 0.28, 11, 0.0, -1)
  200   CONTINUE

C    **  DRAW A CIRCLE (36 LINE SEGMENTS) FOR EACH PUFF ACTIVE

        DO 500 I = 1, TPUFFS
          IF (MF(I) .EQ. 0)  GOTO 500
          PX = XP(I) * 0.5667
          PY = YP(I) * 0.5667
          PLEN = RADP(I) * 0.5667

          DO 400 J = 10, 370, 10
            RADJ = (J - 10) * DTR
            CIRCX = PX + (SIN(RADJ) * PLEN)
            IF (CIRCX .LT. 0.0)  CIRCX = 0.0
            IF (CIRCX .GT. 8.5)  CIRCX = 8.5

            CIRCY = PY + (COS(RADJ) * PLEN)
            IF (CIRCY .LT. 0.0)  CIRCY = 0.0
            IF (CIRCY .GT. 8.5)  CIRCY = 8.5

            IF (J .EQ. 10)  THEN
              IPEN = 3
            ELSE
              IPEN = 2
            ENDIF

            CALL PLOT (CIRCX, CIRCY, IPEN)
  400     CONTINUE
          IPLT = IPLT + 1
  500   CONTINUE

C    **  CLOSE PLOTFILE AND DISPLAY

        CALL PLOTND
        PAUSE

C    **  CLEAR ALPHAGRAPHICS DISPLAY AND RETURN TO ALPHANUMERIC MODE

        WRITE (LUN(1),10) "33, "14
        WRITE (LUN(1),10) "30
        PRINT *, IPLT, ' PUFFS PLOTTED '
        RETURN
        END
```

Subroutine PLOTZ (cont'd)

```
C******************************************************************
C
C     WNDFLD                                    MESO1  VERSION 2.0
C     Wind field plot module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Graphics generation program to display the wind field.
C       Module uses the arrays UU and VV.
C
C     Relationship to other modules:
C
C         Makes calls to:  DTCHAR, DIRSPD
C
C         Called from:     MESO1
C
C******************************************************************

      SUBROUTINE WNDFLD

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

      CHARACTER*9 DDATE, DTIME
      CHARACTER*24 STABCL(7)

      DATA STABCL / 'A - VERY UNSTABLE       ',
     +              'B - MODERATELY UNSTABLE ',
     +              'C - SLIGHTLY UNSTABLE   ',
     +              'D - NEUTRAL             ',
     +              'E - SLIGHTLY STABLE     ',
     +              'F - MODERATELY STABLE   ',
     +              'G - VERY STABLE         ' /

      ARRMIN = 0.3
      ARRINC = 0.1

      CALL DTCHAR (DYR, DDAY, DHR, DMIN, DDATE, DTIME)

C   **  SET VISUAL 500 TERMINAL TO ALPHAGRAPHICS MODE (SEND <US>)

      WRITE (LUN(1),10) "37
   10 FORMAT ('+', 5A1)

C   **  INITIALIZE PLOTFILE, SET SIZE, DRAW BORDER AND HEADERS

      CALL PLOTS (0.0, 0.0, LUN(16))
      CALL FACTOR (1.0)

      CALL PLOT (0.0, 0.0, -3)
      CALL PLOT (0.0, 0.0,  2)
      CALL PLOT (8.5, 0.0,  2)
      CALL PLOT (8.5, 8.5,  2)
      CALL PLOT (0.0, 8.5,  2)
      CALL PLOT (0.0, 0.0,  2)
```

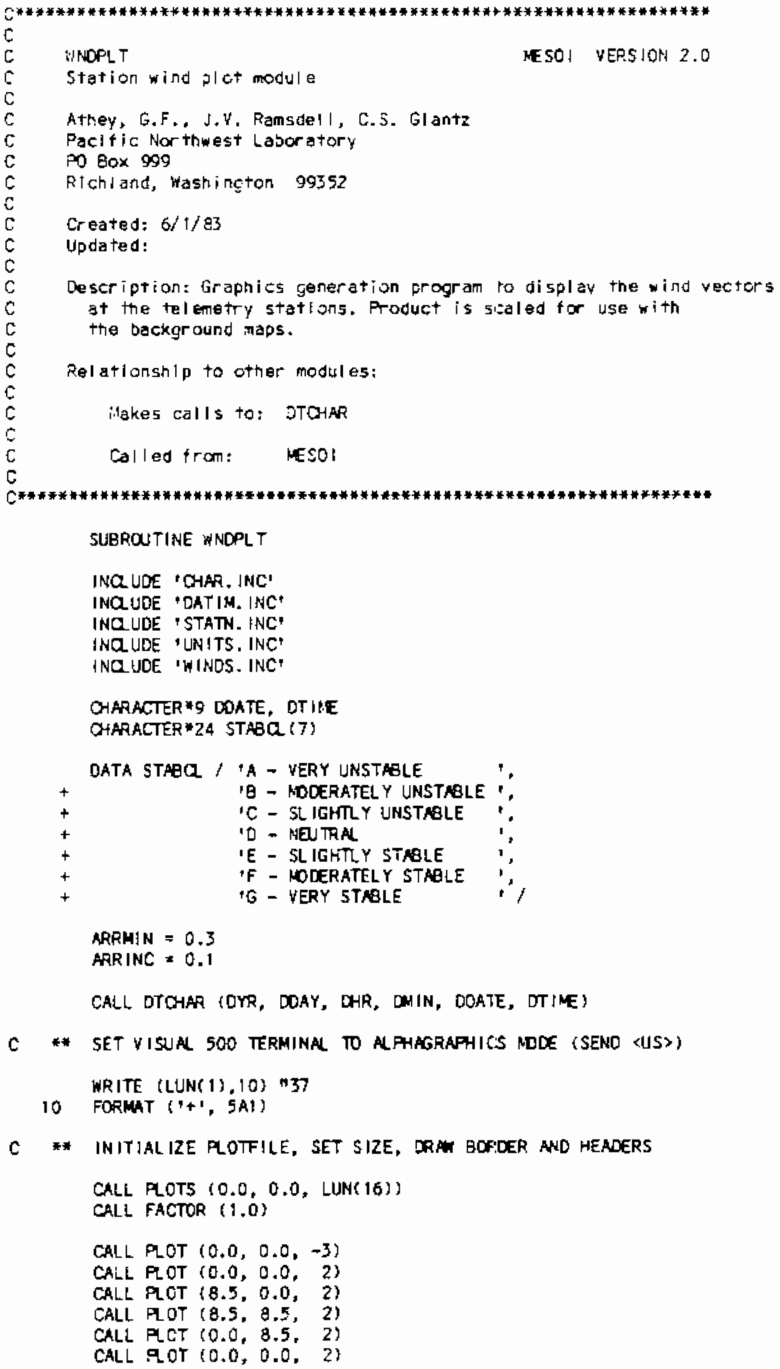
Program Listing 18 Subroutine WNDFLD

```
        CALL SYMBOL (1.0, 9.0, 0.42, 'MESOI WIND FIELD', 0.0, 16)
        CALL SYMBOL (8.6,10.0,0.21,'HANFORD TELEMETRY STATIONS',0.0,26)
        CALL SYMBOL (8.6,9.7,0.21,'ARROWS IN DIRECTION OF FLOW ',0.0,28)
        CALL SYMBOL (8.6,9.2,0.21,'STATION  ELEV  SPEED  DIRECT',0.0,28)
        CALL SYMBOL (8.6,8.9,0.21,'  NAME  METERS  MPH     DEG',0.0,26)

        XORG = 0.0
        YORG = 0.0

        DTR = 0.0174532925

        CALL SYMBOL (8.9, 0.3, 0.21, DDATE, 0.0, 9)

        CALL SYMBOL  (8.9, 0.7, 0.21, 'STABILITY = ', 0.0, 12)
        CALL SYMBOL (10.8, 0.7, 0.21, STABCL(STAB), 0.0, 24)
        CALL SYMBOL  (8.9, 1.1, 0.21, 'MIXING LEVEL =', 0.0, 14)
        DEPTH = FLOATJ(LDEPTH)
        CALL NUMBER (11.4, 1.1, 0.21, DEPTH, 0.0, -1)
        CALL SYMBOL (12.4, 1.1, 0.21, 'METERS', 0.0, 6)
        CALL SYMBOL (10.9, 0.3, 0.21, DTIME, 0.0, 9)

C   ** LIST THE STATIONS NAMES, ELEVATIONS AND WINDS

        YCOORD = 8.5

        DO 500 J = 1, ACTSTA
          CALL SYMBOL (8.8, YCOORD, 0.21, NAMST(J), 0.0, 4)
          FNUM = FLOATJ(ELEV(J))
          CALL NUMBER (10.3, YCOORD, 0.21, FNUM, 0.0, -1)
          FNUM = FLOATJ(SPD(J))
          CALL NUMBER (11.4, YCOORD, 0.21, FNUM, 0.0, -1)
          FNUM = FLOATJ(DIR(J))
          CALL NUMBER (12.4, YCOORD, 0.21, FNUM, 0.0, -1)
          YCOORD = YCOORD - 0.30
  500   CONTINUE

C   **  DRAW AN ARROW ORIGINATING AT EACH GRID POINT TO REPRESENT THE
C   **    MAGNITUDE AND DIRECTION OF THE WIND FLOW.

        DO 610 I=1,16
          DO 600 J=1,16
            PX1 = XORG + (I * 0.5)
            PY1 = YORG + (J * 0.5)

C   **  CALCULATE A SPEED AND DIRECTION FOR EACH GRID POINT BASED
C   **  ON THE ARRAYS UU AND VV.

            CALL DIRSPD ( UU(I,J), VV(I,J), DIRECT, SPEED )

            ARRLEN = ARRMIN + (ARRINC * (SPEED / 5 + 0.5))
            DIRECT = (360 - (DIRECT - 90)) + 180
            IF (DIRECT .GT. 360) DIRECT = DIRECT - 360
            POINT = DIRECT * DTR

            PX2 = COS(POINT) * ARRLEN + PX1
            PY2 = SIN(POINT) * ARRLEN + PY1

            CALL PLOT (PX1, PY1, 3)
            CALL PLOT (PX2, PY2, 2)

            ARRLEN = ARRLEN / 4
            AHD1 = DIRECT + 135
            IF (AHD1 .GT. 360) AHD1 = AHD1 - 360
            AHD2 = DIRECT - 135
            IF (AHD2 .LT. 1)  AHD2 = AHD2 + 360
```

Subroutine WNDFLD (cont'd)

```
            BPX1 = COS(AHD1*DTR) * ARRLEN + PX2
            BPY1 = SIN(AHD1*DTR) * ARRLEN + PY2
            BPX2 = COS(AHD2*DTR) * ARRLEN + PX2
            BPY2 = SIN(AHD2*DTR) * ARRLEN + PY2

            CALL PLOT (BPX1, BPY1, 2)
            CALL PLOT (BPX2, BPY2, 2)
            CALL PLOT (PX2, PY2, 2)

  600     CONTINUE
  610   CONTINUE

C   **  CLOSE THE PLOTFILE AND DISPLAY

        CALL PLOTND
        PAUSE

C   **  CLEAR ALPHAGRAPHICS SCREEN AND RETURN TERMINAL TO ALPHANUMERIC
C   **    MODE (SEND <ESC> <FF> AND <CAN>)

        WRITE (LUN(1),10) "33, "14
        WRITE (LUN(1),10) "30

        RETURN
        END
```

Subroutine WNDFLD (cont'd)



```
C*****************************************************************
C
C     WNDPLT                                      MESOI  VERSION 2.0
C     Station wind plot module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Graphics generation program to display the wind vectors
C       at the telemetry stations. Product is scaled for use with
C       the background maps.
C
C     Relationship to other modules:
C
C        Makes calls to:  DTCHAR
C
C        Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE WNDPLT

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

      CHARACTER*9 DDATE, DTIME
      CHARACTER*24 STABCL(7)

      DATA STABCL / 'A - VERY UNSTABLE       ',
     +              'B - MODERATELY UNSTABLE ',
     +              'C - SLIGHTLY UNSTABLE   ',
     +              'D - NEUTRAL             ',
     +              'E - SLIGHTLY STABLE     ',
     +              'F - MODERATELY STABLE   ',
     +              'G - VERY STABLE         ' /

      ARRMIN = 0.3
      ARRINC = 0.1

      CALL DTCHAR (DYR, DDAY, DHR, DMIN, DDATE, DTIME)

C  ** SET VISUAL 500 TERMINAL TO ALPHAGRAPHICS MODE (SEND <US>)

      WRITE (LUN(1),10) "37
   10 FORMAT ('+', 5A1)

C  ** INITIALIZE PLOTFILE, SET SIZE, DRAW BORDER AND HEADERS

      CALL PLOTS (0.0, 0.0, LUN(16))
      CALL FACTOR (1.0)

      CALL PLOT (0.0, 0.0, -3)
      CALL PLOT (0.0, 0.0,  2)
      CALL PLOT (8.5, 0.0,  2)
      CALL PLOT (8.5, 8.5,  2)
      CALL PLOT (0.0, 8.5,  2)
      CALL PLOT (0.0, 0.0,  2)
```

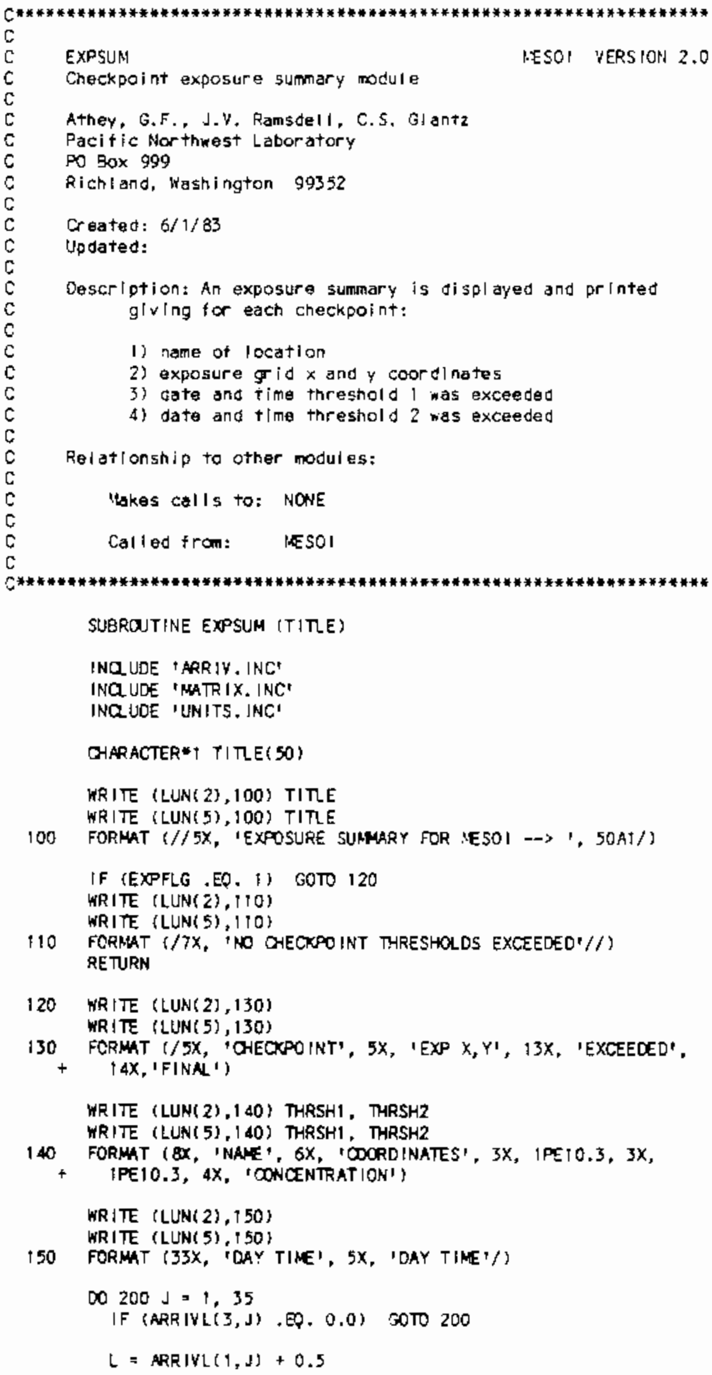
Program Listing 19 Subroutine WNDPLT

```
      CALL SYMBOL (1.0, 9.0, 0.42, 'STATION WIND PLOT ', 0.0, 18)
      CALL SYMBOL (8.6,10.0,0.21,'HANFORD TELEMETRY STATIONS',0.0,26)
      CALL SYMBOL (8.6,9.7,0.21,'ARROWS IN DIRECTION OF FLOW ',0.0,28)
      CALL SYMBOL (8.6,9.2,0.21,'STATION  ELEV  SPEED  DIRECT',0.0,28)
      CALL SYMBOL (8.6,8.9,0.21,'  NAME  METERS  MPH     DEG',0.0,26)

      XORG = 0.0
      YORG = 0.0

      DTR = 0.0174532925

      CALL SYMBOL (8.9, 0.3, 0.21, DDATE, 0.0, 9)

      CALL SYMBOL  (8.9, 0.7, 0.21, 'STABILITY = ', 0.0, 12)
      CALL SYMBOL (10.8, 0.7, 0.21, STABCL(STAB), 0.0, 24)
      CALL SYMBOL  (8.9, 1.1, 0.21, 'MIXING LEVEL =', 0.0, 14)

      DEPTH = FLOATJ(LDEPTH)
      CALL NUMBER (11.4, 1.1, 0.21, DEPTH, 0.0, -1)
      CALL SYMBOL (12.4, 1.1, 0.21, 'METERS', 0.0, 6)
      CALL SYMBOL (10.9, 0.3, 0.21, DTIME, 0.0, 9)

C   **  LIST THE STATIONS NAMES, ELEVATIONS AND WINDS

      YCOORD = 8.5

      DO 500 J = 1, ACTSTA
        CALL SYMBOL (8.8, YCOORD, 0.21, NAMST(J), 0.0, 4)
        FNUM = FLOATJ(ELEV(J))
        CALL NUMBER (10.3, YCOORD, 0.21, PNUM, 0.0, -1)
        FNUM = FLOATJ(SPD(J))
        CALL NUMBER (11.4, YCOORD, 0.21, FNUM, 0.0, -1)
        FNUM = FLOATJ(DIR(J))
        CALL NUMBER (12.4, YCOORD, 0.21, FNUM, 0.0, -1)
        YCOORD = YCOORD - 0.30
  500 CONTINUE

C   **  DRAW AN ARROW ORIGINATING AT EACH STATION LOCATION TO REPRESENT
C   **    THE MAGNITUDE AND FLOW DIRECTION OF THE WINDS.

      DO 700 J = 1, ACTSTA
        PX1 = XORG + (XSTA(J) * 0.5)
        PY1 = YORG + (YSTA(J) * 0.5)
        CALL SYMBOL (PX1, PY1, 0.14, 5, 0.0, -1)

        IF (SPD(J) .LT. 1 .OR. SPD(J) .GT. 50)  GOTO 700
        ARRLEN = ARRMIN + (ARRINC * (SPD(J) / 5))
        DIR(J) = (360 - (DIR(J) - 90)) + 180
        IF (DIR(J) .GT. 360) DIR(J) = DIR(J) - 360
        POINT = DIR(J) * DTR
        PX2 = COS(POINT) * ARRLEN + PX1
        PY2 = SIN(POINT) * ARRLEN + PY1

        CALL PLOT (PX1, PY1, 3)
        CALL PLOT (PX2, PY2, 2)

        ARRLEN = ARRLEN / 4
        AHD1 = DIR(J) + 135
        IF (AHD1 .GT. 360) AHD1 = AHD1 - 360
        AHD2 = DIR(J) - 135
        IF (AHD2 .LT. 1) AHD2 = AHD2 + 360

        BPX1 = COS(AHD1 * DTR) * ARRLEN + PX2
        BPY1 = SIN(AHD1 * DTR) * ARRLEN + PY2
        BPX2 = COS(AHD2 * DTR) * ARRLEN + PX2
```

Subroutine WNDPLT (cont'd)

```
          BPY2 = SIN(AHD2 * DTR) * ARRLEN + PY2

          CALL PLOT (BPX1, BPY1, 2)
          CALL PLOT (BPX2, BPY2, 2)
          CALL PLOT (PX2, PY2, 2)
  700   CONTINUE

C   **  CLOSE PLOTFILE AND DISPLAY

        CALL PLOTND
        PAUSE

C   **  CLEAR ALPHAGRAPHICS SCREEN AND RETURN TERMIANL TO ALPHANUMERIC
C   **    MODE (SEND <ESC> <FF> AND <CAN>).

        WRITE (LUN(1),10) "33, "14
        WRITE (LUN(1),10) "30

        RETURN
        END
```

Subroutine WNDPLT (cont'd)



```
C*****************************************************************
C
C     EXPSUM                                    MESOI  VERSION 2.0
C     Checkpoint exposure summary module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: An exposure summary is displayed and printed
C           giving for each checkpoint:
C
C           1) name of location
C           2) exposure grid x and y coordinates
C           3) date and time threshold 1 was exceeded
C           4) date and time threshold 2 was exceeded
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE EXPSUM (TITLE)

      INCLUDE 'ARRIV.INC'
      INCLUDE 'MATRIX.INC'
      INCLUDE 'UNITS.INC'

      CHARACTER*1 TITLE(50)

      WRITE (LUN(2),100) TITLE
      WRITE (LUN(5),100) TITLE
  100 FORMAT (//5X, 'EXPOSURE SUMMARY FOR MESOI --> ', 50A1/)

      IF (EXPFLG .EQ. 1)  GOTO 120
      WRITE (LUN(2),110)
      WRITE (LUN(5),110)
  110 FORMAT (/7X, 'NO CHECKPOINT THRESHOLDS EXCEEDED'//)
      RETURN

  120 WRITE (LUN(2),130)
      WRITE (LUN(5),130)
  130 FORMAT (/5X, 'CHECKPOINT', 5X, 'EXP X,Y', 13X, 'EXCEEDED',
     +   14X,'FINAL')

      WRITE (LUN(2),140) THRSH1, THRSH2
      WRITE (LUN(5),140) THRSH1, THRSH2
  140 FORMAT (8X, 'NAME', 6X, 'COORDINATES', 3X, 1PE10.3, 3X,
     +   1PE10.3, 4X, 'CONCENTRATION')

      WRITE (LUN(2),150)
      WRITE (LUN(5),150)
  150 FORMAT (33X, 'DAY TIME', 5X, 'DAY TIME'/)

      DO 200 J = 1, 35
        IF (ARRIVL(3,J) .EQ. 0.0)  GOTO 200

        L = ARRIVL(1,J) + 0.5
```

Program Listing 20 Subroutine EXPSUM

```
        IF (L .LT. 1)  L = 1
        IF (L .GT. 31) L = 31

        M = ARRIVL(2,J) + 0.5
        IF (M .LT. 1)  M = 1
        IF (M .GT. 31) M= 31
        IF (EXPCUM(L,M) .LT. THRSH1)  GOTO 200

        IDAY1 = ARRIVL(6,J)
        IHR1 =  ARRIVL(7,J)
        IMIN1 = ARRIVL(8,J)

        IDAY2 = ARRIVL(9,J)
        IHR2 = ARRIVL(10,J)
        IMIN2 = ARRIVL(11,J)

        WRITE (LUN(2),160) (ARRIVL(I,J),I=4,5), (ARRIVL(I,J),I=1,2),
   +      IDAY1, IHR1, IMIN1, IDAY2, IHR2, IMIN2, EXPCUM(L,M)
        WRITE (LUN(5),160) (ARRIVL(I,J),I=4,5), (ARRIVL(I,J),I=1,2),
   +      IDAY1, IHR1, IMIN1, IDAY2, IHR2, IMIN2, EXPCUM(L,M)
160     FORMAT (/6X, 2A4, 4X, 2(1X,F4.0), 2(5X,I3,1X,2I2), 6X,1PE10.3)
        IF (MOD(J,10) .EQ. 0)  PAUSE 'TYPE C TO CONTINUE'
200   CONTINUE
    RETURN
    END
```

Subroutine EXPSUM (cont'd)

TRANSPORT, DIFFUSION AND DEPLETION SUBROUTINES

MESOI Version 2.0 uses 11 subroutines that are primarily related to the transport, diffusion, deposition, decay and depletion of effluents. These subroutines, along with their memory requirements and a brief description of their uses, are listed in Table 11. Four alternatives are provided for use as subroutine SIGMA; only one of the alternatives should be included when MESOI is customized for a specific location. Subroutine ALPHA is listed, but only a dummy subroutine is included.

This section contains a description and listing of each of the transport, diffusion, deposition and depletion subroutines. They are treated in the order in which they are called.

WIND

This module handles the decoding of the wind data and the generation of a transport wind field over the model domain. WIND is called from MESOI or INIT after DATRD has been accessed to input the data records.

Initially, a loop through data elements of the active stations breaks out the wind speed and directions. These are then converted to U and V transport components. If the data are missing (9999) or winds are calm (0000), the decoding steps are skipped and a flag (LFLAG) is set to indicate that no data are available at that station. A warning is printed if the speed or direction exceeds a preset range. A similar process takes place for the upper level wind.

The winds are then interpolated to each grid point. At each point, the winds for up to the ten nearest stations are weighted by their distance from the grid point to obtain U and V transport components. If a station is on the point, only that data is used without any spatial interpolation.

Program listing 21 contains the WIND code.

TERRA

TERRA is designed to modify the transport wind field in response to terrain features. It uses a separate data set from the diffusion related topography. TERRA is called from MESOI and INIT after they call WIND and only when the model is not in the testing mode.

TERRA uses a very simple description of the topography at each wind grid to modify the U and V components computed in WIND. Each feature can be viewed as a line segment with a given orientation which will redirect the wind vector. For any given feature, the wind can come from either of the two sides. Coefficients are used to describe the influence that either of these

TABLE 11. Transport, Diffusion and Depletion Subroutines

	SIZE (BYTES)			
NAME	CODE	LOCAL DATA	COMMON DATA	USE
WIND	1519	1054	25755	Decodes wind data and interpolates data to grid nodes.
TERRA	552	3290	25083	Adjusts wind data to account for expected terrain effects.
ALPHA	11			Provides exponents for use in vertical interpolation of the wind.
WINSPD	559	40	25380	Provides wind speed at puff release height and time for plume rise computations.
PLMRIZ	643	218	663	Computes final plume rise for use in estimating effective release height.
PUFFR	306		29312	Assigns initial attributes to puffs at release time.
PUFFM	1158	112	53348	Calculates puff movement for advection step.
DTOPO	802	84	3848	Computes terrain elevation under center of puffs at beginning end of advection step.
DIFDEP	3346	3284	68131	Diffusion, deposition and decay computations and time integration during the advection step.
RDK	419	8	824	Computes decay and ingrowth fractions for sampling interval and remainder of advection step.
SIGMA				Computes new diffusion coefficients given the last values, atmospheric stability, mixing layer thickness and distance moved.
ASIG	383	140		
BSIG	354	148		
DSIG	424	136		
NRCSIG	399	300		

two surfaces would have on the wind perpendicular to the line. These coefficients are inversely related to the slope of the topographic feature present to the wind. A steeper slope will cause greater modification of the winds and thus the coefficient is smaller.

When TERRA is called for the first time (from INIT) the wind terrain flag (TERFLG) is passed as zero. The module then sets all values in the terrain angle array (TANGLE) to a negative one. This indicates that there is no defined terrain at the grid point to influence the wind. All values in the slope coefficient array (COEFF1) are set to 1.0. This defines the terrain at each point as level; no influence on wind values. The module then opens a data file named TERRA.DAT. If this file cannot be opened, TERRA returns and the model will not do any terrain corrections of the wind field.

If the file is opened properly, terrain data is input. Each record corresponds to a grid point at which the terrain is expected to modify the wind. The data record specifies the x and y grid coordinates, the angle of the terrain feature to an east-west line, and the coefficients defining the attenuation of the perpendicular wind vector on each side of the feature. Input continues until an EOF is encountered. The wind terrain flag is then set to one to indicate the completion of the initialization.

Whenever TERRA is called with TERFLG = 1, the code skips the initialization and executes only the wind vector modification section. This section consists of a double loop to check and modify winds at each grid point as necessary. The U and V values returned from WIND are passed to the DIRSPD module and converted into speed and direction. The direction value is checked to see which face of the terrain feature will be used in the computations. A relative angle between the direction and the terrain is used to compute components parallel and perpendicular to the terrain face. The perpendicular component is then decreased by the specified magnitude and the parallel component is increased to maintain the same final magnitude. The final components are then translated back into true wind components.

The creation of the data set for this module requires some insight into the effects of local topography on the wind field. Only major topographic features can be represented. Building a data set which looks good takes some time. Use of the WNDFLD module helps visualize the output of TERRA.

Program Listing 22 contains the TERRA code.

ALPHA

Subroutine ALPHA is included in Version 2.0 of MESOI as a means of simplifying future enhancement of the model. The purpose of ALPHA is to permit the vertical interpolation of the wind field to be a function of stability. Normal atmospheric boundary layer wind speed profiles are a function of stability. Analysis of wind profile data at Hanford, Washington indicates that the exponents in the equations used for vertical interpolation of the wind in MESOI may also be functions of stability. However, there is sufficient scatter in the Hanford profile data to make it difficult to support the choice of one set of exponents over any other set. As a result, until further analysis is completed, subroutine ALPHA returns values of one for the exponents. This results in a linear vertical interpolation.

Listing 23 contains the subroutine ALPHA code.

WINSPD

This subroutine is used by MESOI to calculate a surface wind speed at a specified release point. It uses the U and V components of the surrounding grid points to approximate the speed. This speed is used in the calculation of plume rise. WINSPD is called from MESOI only when the plume rise flag is set (PRFLG = 1).

The program first determines the grid point to the southeast of the release point. Using those coordinates, the other three grid points surrounding the release point can be defined. Weighting factors (C1 and C2) are computed for the temporal interpolation. If 15 minute wind data are being used (INTFLG = 1) C1 and C2 both equal 0.5. If hourly data are used, the weights are dependent on which advection step is being executed. Spatial weighting factors (RX, RY, RX1, RY1) are computed based on source location and represent the distances in grid units along the axis to the four grid points. U and V components of the wind at the release point are calculated by summing the contribution of the wind at each point. A speed is then determined from these wind components.

Program listing 24 contains the WINSPD code.

PLMRIZ

The PLMRIZ module is designed to calculate the final plume rise of a release whenever the user has defined a stack. PLMRIZ is called form MESOI before PUFFR is called. The rise calculated is used to modify the specified release height.

If the stack is above the mixed depth, no plume rise occurs. If the release is within the mixed layer, a series of calculations is made based on the work of Briggs (1969). The theory behind this module is discussed in an earlier part of the document. The module returns a value for PRISE. If the testing mode has been enabled, the parameters used in the calculations are reported.

Program listing 25 contains the PLMRIZ code.

PUFFR

This subroutine assigns the initial attributes to puffs when they are released. PUFFR is called from MESOI if the puff release flag (PUFLG) is set and the maximum number of puffs have not already been released.

PUFFR sets the following attributes each time it is called:

puff active flag (MF)
x, y and z coordinates
source number
puff age
source strength
diffusion coefficients.

Program listing 26 contains the PUFFR code.

PUFFM

This module computes puff displacement during each advection period. It is called from within the advection loop of MESOI.

The displacement is based on the interpolated winds at the puff center at the beginning and end of the advection period. The initial advection surface winds are computed at each time using spatial interpolation between the puff center and the surrounding grid points and temporal interpolation between the surface winds at the previous and current data interval. An initial estimate of the puff position at the end of the advection period is made using the displacement estimate made with the surface winds at the beginning of the advection period. Another estimate is made by averaging the displacements calculated for the beginning and end of the advection step. The final displacement is estimated by interpolating the winds between the surface and the top of the mixed layer.

If the puff moves off the grid, it continues to move with its last speed and direction until it no longer contributes to the exposure at that grid point.

Program listing 27 contains the PUFFM code.

DTOPO

Subroutine DTOPO computes the terrain elevation under the center of puffs as they are transported across the grid. The terrain elevation is determined by interpolation between the puff's position and the surrounding grid nodes. The weights given to the elevations at the grid nodes are based on the reciprocal of the distances between the puff's position and the nodes. The elevation under the puff center is estimated at the beginning and end of each advection period. DIFDEP determines the elevation for each sampling interval by dividing the change in elevation for the advection period by the number of sampling intervals and adding the sampling interval change to the current elevation at the beginning of each interval.

The terrain elevation under the puff center is used in the diffusion computations to provide a ground-level reference for determining if a specific receptor is above or below the effective release height.

When a puff is off of the edge of the grid, the closest two grid points are used to determine the elevation under the puff center, except when the puff is off of a corner. In this latter case, the elevation assigned to the terrain under the puff center is the elevation of the corner grid node.

The computer code for subroutine DTOPO is contained in Listing 28.

DIFDEP

DIFDEP is the subroutine that performs the actual transport, diffusion, deposition and depletion computations for MESOI. Most of the subroutine is contained within a large loop that moves the puffs. Each step in the loop corresponds to a time step of one sampling interval. Figure 11 shows the flow of logic in DIFDEP.

Prior to entering its primary loop, DIFDEP determines the size of the sampling interval and the corresponding number of sampling intervals in the advection period. The magnitude of the sampling interval is controlled by the wind speed and size of the puff. Starting with a sampling interval equal to the advection period, the ratio between the distance moved in a sampling interval and sigma y is computed. If the ratio is less than 2 (see Figure 7), the sampling period to be used for computations is set equal to the advection period. If the ratio is greater than 2, the next largest integer factor of the advection period (in minutes) is used as a potential sampling interval, and the ratio is recomputed. This process continues until an acceptable sampling interval is found or until the trial sampling interval is one minute. The possible values for the sampling interval and number of sampling intervals for a 15 minutes advection period (four puffs per hour) are found in the data statements at the beginning of DIFDEP.

As soon as the proper values are selected for these variables, several other variables, whose values depend on the number of sampling intervals and their duration, are evaluated. If the decay flag has been set during initialization, the variables evaluated at this time include the decay constants for each of the decaying species for a sampling interval and for intervals corresponding to the time remaining in the advection period at the end of each sampling interval. The evaluation takes place in subroutine RDK. The constants are used for decay and ingrowth computations within the advection period.

The primary loop in DIFDEP is then executed once for each sampling interval, unless the puff leaves the grid or the puff concentration falls below the minimum concentration of interest. In either of these cases, the active puff flag is turned off and control is returned to MESOI.

The first action within the loop is to compute the diffusion coefficients to be used in the sampling interval. This takes place in subroutine SIGMA.

As soon as the diffusion coefficients have been determined, DIFDEP: computes the concentration at the puff center, determines the puff radii for plotting and determining the potentially affected area, and evaluates the term in the diffusion equation that describes the vertical concentration distribution at the center of the puff. These computations are shown as a single box in Figure 11. However, that box contains a sizeable segment of the code because there are separate paths that may be followed through the box. The path taken depends upon the ratios between the effective release height and mixing layer thickness and between the vertical diffusion coefficient and the mixing layer thickness of effective release height. These paths are shown in Figure 12. In each path, the concentration at the puff center is compared with the minimum concentration of interest. If the puff center concentration is lower, a path leading to one of the four circles marked GO TO 200 is taken. These paths result in the active puff flag being turned off and control returning to MESOI. The puff radii are computed for the effective release height and have slightly different magnitudes.

Next DIFDEP computes the concentration at ground-level beneath the center of the puff. At this point, there is a branch in the loop. If the ground-level concentration is above the minimum concentration of interest or precipitation is occurring, the right branch in Figure 11 is taken, leading to evaluation of the impact of the puff on the time-integrated air and surface concentrations. Otherwise, the left branch is taken. In either case the puff is moved and it position in relation to the grid is determined. If it has departed from the grid, the active puff flag is turned off and control is returned to MESOI.

Accumulation of the time-integrated, ground-level air concentrations is straight forward. The accumulation of surface contamination due to dry deposition and washout is more complex because it involves keeping track of material remaining on the surface rather than just passing over the surface. Figure 13 shows the bookkeeping involved in each sampling interval. At the beginning of the advection period, the surface concentrations are set to zero; the only material on the ground at the end of the first sampling interval is the material deposited during the interval. At the end of the second and succeeding intervals, there is contamination on the ground from the current sampling interval and preceeding intervals. However, some of the material deposited during the earlier intervals will have changed species if it decays. To facilitate the bookkeeping, changes that will occur during the remainder of the advection period are taken into account when material is deposited on the surface.

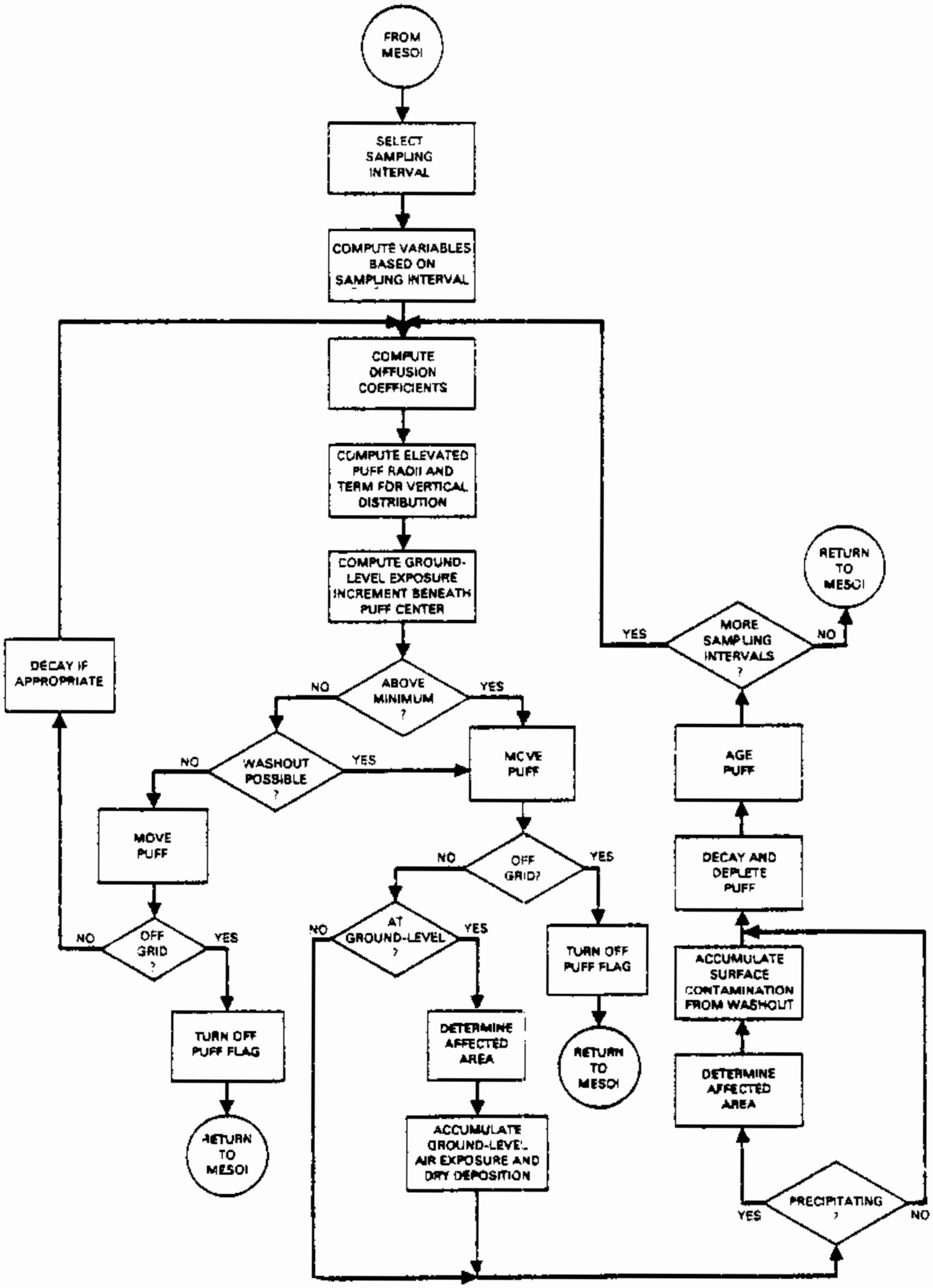


FIGURE 11. Subroutine DIFDEP Logic

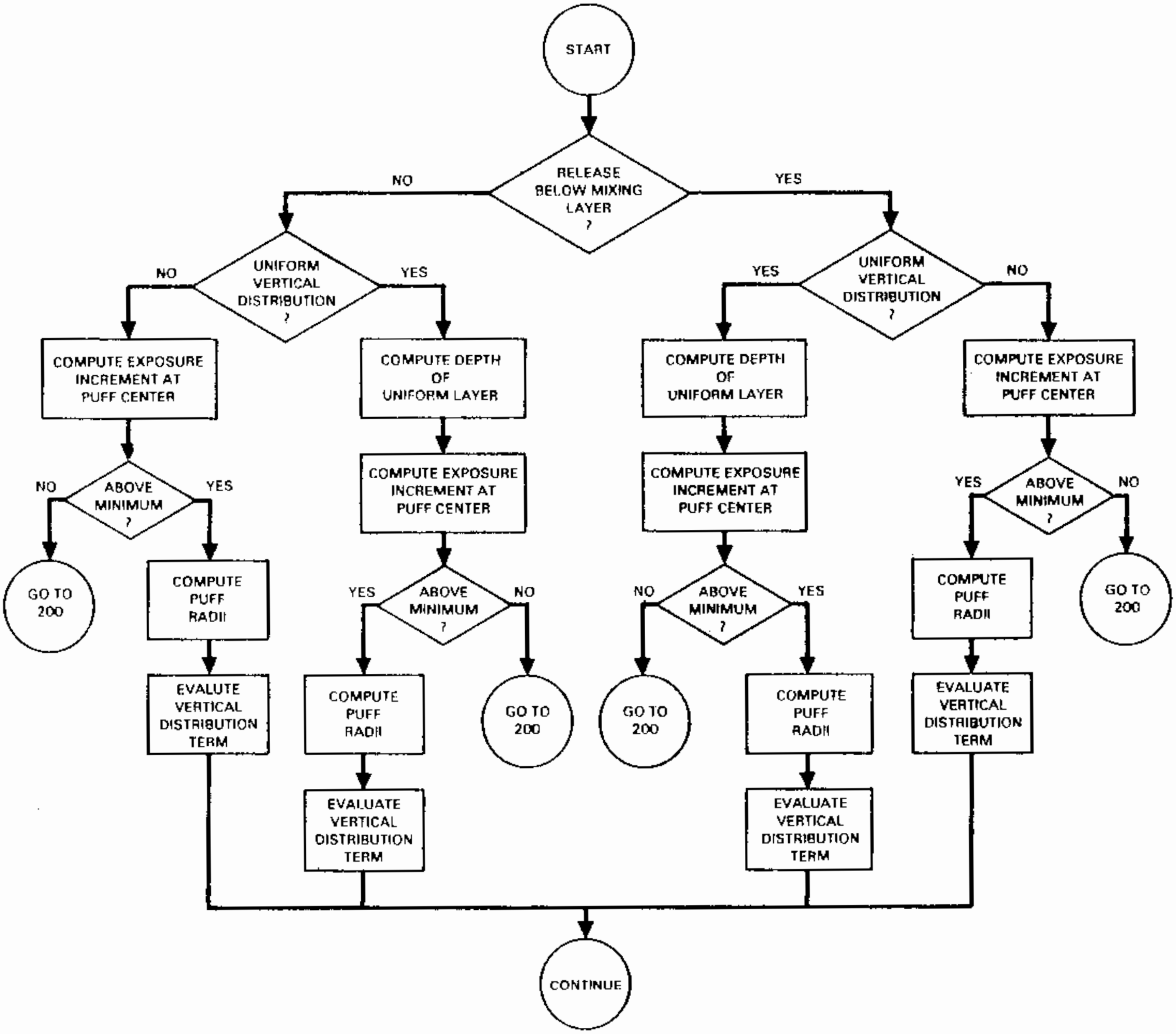

START
RELEASE BELOW MIXING LAYER ?
NO
YES
UNIFORM VERTICAL DISTRIBUTION ?
NO
YES
YES
NO
COMPUTE EXPOSURE INCREMENT AT PUFF CENTER
COMPUTE DEPTH OF UNIFORM LAYER
COMPUTE DEPTH OF UNIFORM LAYER
COMPUTE EXPOSURE INCREMENT AT PUFF CENTER
ABOVE MINIMUM ?
NO
YES
COMPUTE EXPOSURE INCREMENT AT PUFF CENTER
COMPUTE EXPOSURE INCREMENT AT PUFF CENTER
ABOVE MINIMUM ?
YES
NO
GO TO 200
COMPUTE PUFF RADII
EVALUTE VERTICAL DISTRIBUTION TERM
ABOVE MINIMUM ?
YES
NO
ABOVE MINIMUM ?
NO
YES
COMPUTE PUFF RADII
GO TO 200
GO TO 200
COMPUTE PUFF RADII
COMPUTE PUFF RADII
GO TO 200
EVALUATE VERTICAL DISTRIBUTION TERM
EVALUATE VERTICAL DISTRIBUTION TERM
EVALUATE VERTICAL DISTRIBUTION TERM
CONTINUE

FIGURE 12. Alternative Logic Paths for Computation of Exposure Increment at Puff Centers, Vertical Concentration Distributions and Puff Radii

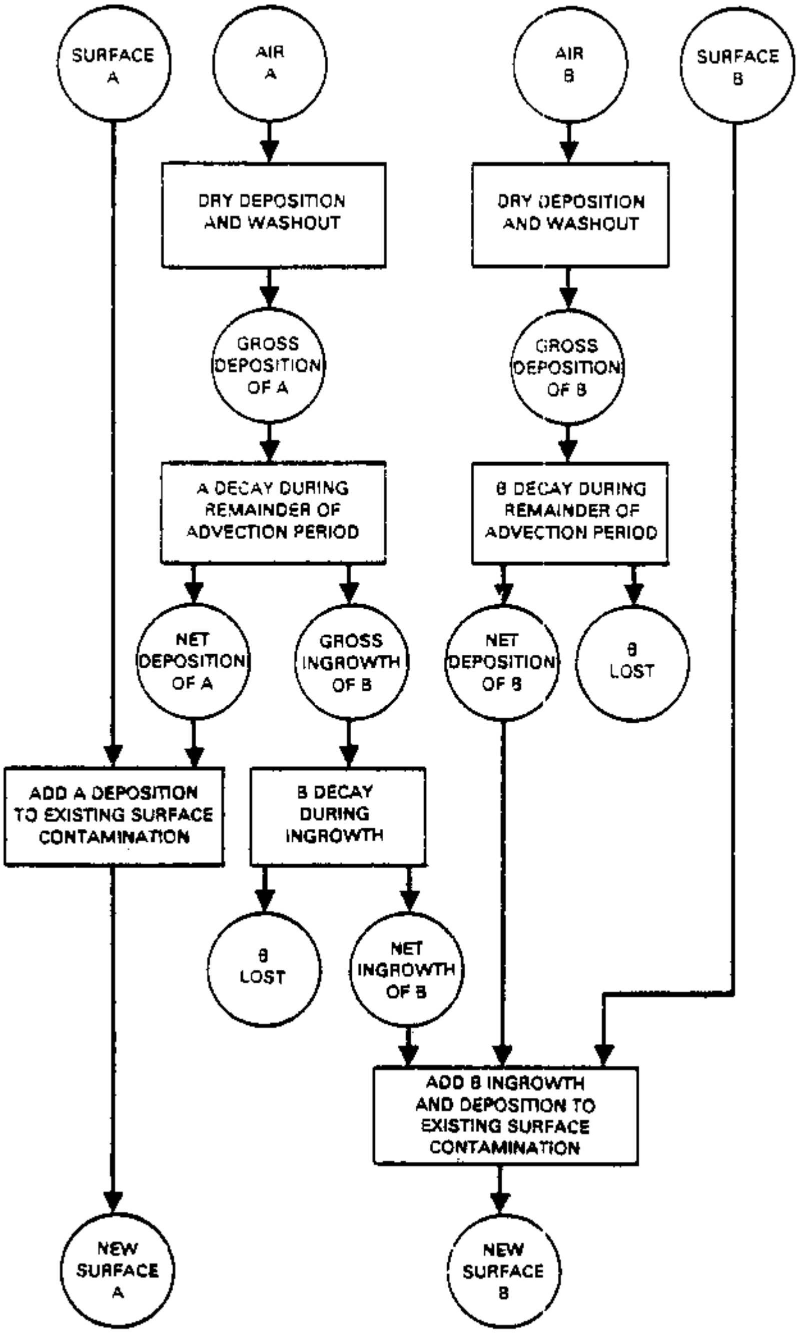

SURFACE A
AIR A
AIR B
SURFACE B
DRY DEPOSITION AND WASHOUT
DRY DEPOSITION AND WASHOUT
GROSS DEPOSITION OF A
GROSS DEPOSITION OF B
A DECAY DURING REMAINDER OF ADVECTION PERIOD
B DECAY DURING REMAINDER OF ADVECTION PERIOD
NET DEPOSITION OF A
GROSS INGROWTH OF B
NET DEPOSITION OF B
B LOST
ADD A DEPOSITION TO EXISTING SURFACE CONTAMINATION
B DECAY DURING INGROWTH
B LOST
NET INGROWTH OF B
ADD B INGROWTH AND DEPOSITION TO EXISTING SURFACE CONTAMINATION
NEW SURFACE A
NEW SURFACE B

FIGURE 13. Surface Deposition and Decay Logic for Sampling Intervals

At the end of each sampling interval, after the contributions to the time-integrated ground-level air concentration and surface contamination have been accumulated, the material in the puff is corrected to account for decay, dry deposition and washout. Figure 14 shows how the depletion of the puff is accomplished. Together, Figures 13 and 14 provide a description of the fate of the initial effluent and its daughter.

The last action in the loop, prior to moving to the next sampling interval, in minutes, is to increment the age of the puff by the duration of the sampling interval. The age of the puff is monitored for use in future extensions of MESOI. Two likely uses of the puff age are a diffusion coefficient parameterization based on travel time rather tan distance and extension of the treatment of decay to more than a single parent-daughter combination.

The DIFDEP computer code is given is Listing 29.

RDK

Subroutine RDK computes the fractions of each of the decaying species that remain at the end of the sampling interval and at the end of the advection period. The fraction of the parent species remaining at the end of the sampling interval is ADECA2, and that for the daughter is BDECA2. The fractions remaining at the end of the advection period are in ADK and BDK. These arrays contain a different fraction for each sampling interval within the advection period. Finally, the fractions of the parent decaying into the daughter that remain at the end of the sampling interval and advection period after correction for daughter decay are in BINSI and BDIN, respectively.

Program Listing 30 contains the RDK computer code.

SIGMA

The diffusion coefficients used in MESOI Version 2.0 are computed in subroutine SIGMA. Computer codes for four different diffusion coefficient parameterizations are included with MESOI Version 2.0 program package. The individual parameterizations are discussed in Part I, and the codes are contained in Program Listings 31 through 34. The code for each of the alternatives is listed under SUBROUTINE SIGMA so the user can change alternatives prior to compilation without changing any other portion of the program. The parameterizations are easily identified by examination of the headers.

The structure of each of the alternative subroutines is identical. Sigma y is computed first, then sigma a. In computing the sigma values, the first step is to use the old sigma value and the current stability to determine the distance to a virtual point source. The effective distance to the puff position at the end of the sampling interval is then determined by adding the

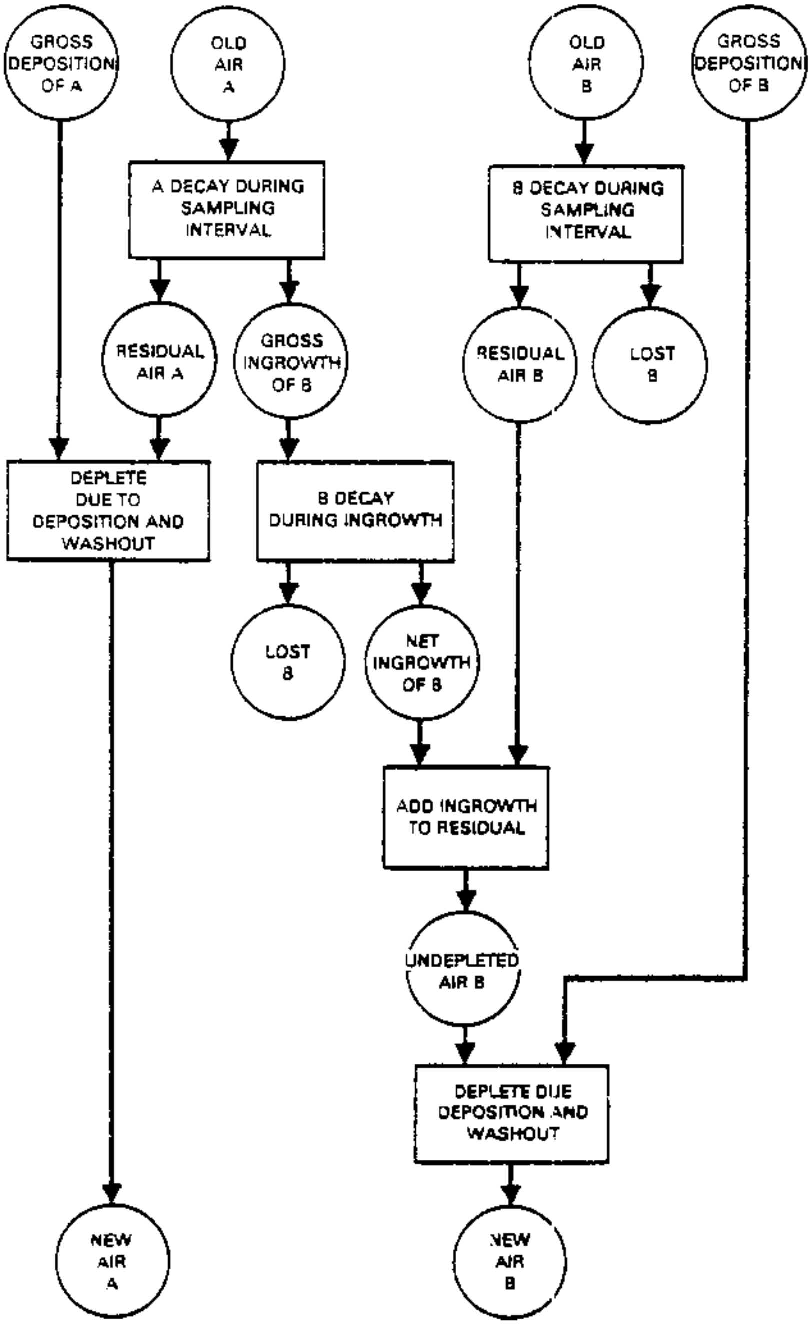


FIGURE 14. Puff Depletion Logic

position at the end of the sampling interval is then determined by adding the distance moved during the interval to the virtual distance. Finally, the sigma values are computed directly from the parameterization using the current stability and the effective distance.

The growth of sigma z is controlled by the thickness of the mixing layer (LDEPTH). If sigma z is initially equal to or greater than eight-tenths of LDEPTH, the sigma z computations are bypassed, and the old sigma z value is returned as the new value. If sigma z is initially less than eight-tenths of LDEPTH but exceeds that value at the end of the sampling interval, sigma z is set equal to eight-tenths of LDEPTH. Under other conditions it is permitted to increase as dictated by the parameterizations.

```
C*******************************************************************
C
C     WIND                                          MESOI  VERSION 2.0
C     Wind field generation module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Program to decode wind data elements into direction
C       and speeds then interpolate to each grid location
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI, INIT
C
C*******************************************************************

      SUBROUTINE WIND(NUMSTA)

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

      REAL UDIR, USPD, UANG, USTA(30), VSTA(30)

      INTEGER LFLAG(30)

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. 99 .OR. SPD(N) .EQ. 99)  GOTO 100
        DIR(N) = DIR(N) * 10.0
        IF (DIR(N) .EQ. 0 .AND. SPD(N) .EQ. 0)  GOTO 100

        LFLAG(N) = 0

        IF (DIR(N) .LT. 0.0 .OR. DIR(N) .GT. 360.0)
     +    WRITE (LUN(2),120) J, DIR(N), DDAY, DHR, DMIN
        IF (SPD(N) .LT. 0.0 .OR. SPD(N) .GT. 45.0)
     +    WRITE (LUN(2),130) J, SPD(N), DDAY, DHR, DMIN

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

        ANG = (270. - DIR(N)) * (3.1415927 / 180.)
        USTA(N) = (SPD(N) * COS(ANG)) * CONV
        VSTA(N) = (SPD(N) * SIN(ANG)) * CONV
        GOTO 110
```

Program Listing 21 Subroutine WIND

```
  100   LFLAG(N) = 1
        IF (N .GE. ACTSTA)  GOTO 200
  110 CONTINUE

  120 FORMAT (5X, 'WARNING -- STATION NUMBER ', I2, /7X,
     +  'DIRECTION OUT OF NORMAL RANGE --> ', F6.1,
     +  /7X, 'JDAY = ', I3, ' AT ', I2, ':', I2//)
  130 FORMAT (5X, 'WARNING -- STATION NUMBER ', I2, /7X,
     +  'SPEED OUT OF NORMAL RANGE --> ', F6.1, /7X,
     +  'JDAY = ', I3, 2X, ' AT ', I2, ':',  I2//)

C   **  UPPER LEVEL WIND

      UDIR = UPWIND / 100
      USPD = UPWIND - (UPWIND/100) * 100
      UDIR = UDIR * 10.0
      IF (UDIR .LT. 0 .OR. UDIR .GT. 360)  THEN
        WRITE (LUN(2),140) UDIR, DDAY, DHR, DMIN
  140   FORMAT (/5X, 'WARNING --> UPPER LEVEL WIND DIRECTION ',
     +    'IS OUT OF NORMAL RANGE ', /17X, I5, ' DEG ON JDAY ',
     +    I3, ' AT ', I2, ':', I2//)
      ENDIF

      IF (USPD .LT. 0 .OR. USPD .GT. 60)  THEN
        WRITE (LUN(2),150) USPD, DDAY, DHR, DMIN
  150   FORMAT (/5X, 'WARNING --> UPPER LEVEL WIND SPEED IS OUT ',
     +    'OF NORMAL RANGE', /17X, I5, ' MPH ON JDAY ',
     +    I3, ' AT ', I2, ':', I2//)
      ENDIF

      UANG = (270. - UDIR) * (3.1415927 / 180.)
       UUG = USPD * COS(UANG) * CONV
       VVG = USPD * SIN(UANG) * CONV

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

  200 DO 280 I = 1, 16
        DO 270 J = 1, 16
          SNU = 0.0
          SNV = 0.0
          SND = 0.0
           NS = 0
          LEND = MIN(ACTSTA,10)

          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-15)  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)
            SNV = SNV + VSTA(LS) / STDIST(L,I,J)
            SND = SND + (1.0 / STDIST(L,I,J) )
            NS = NS + 1
  220     CONTINUE

          IF (NS .EQ. 0)  THEN
            WRITE (LUN(2),230) I, J, DDAY, DHR, DMIN
  230       FORMAT (/5X, 'FATAL ERROR -- NO DATA AVAILABLE TO ',
     +        'CALCULATE WINDS AT GRID POINT ', I2, ',', I2,
```

Subroutine WIND (cont'd)

```
   +        /7X, 'JDAY = ', I3, 5X, ' AT  ', I2, ':', I2//)
          STOP
        END IF

        IF (NS .LT. 3 .AND. TITLE(1) .NE. '$') THEN
          WRITE (LUN(2),240) NS, I, J, DDAY, DHH, DMIN
240       FORMAT (5X, 'WARNING -- ONLY ', I2, ' STATIONS AT ',
   +        'WIND GRID POINT ', I2, ',', I2, /7X, 'JDAY = ',
   +         I3, 5X, ' AT ', I2, ':', I2//)
          GOTO 260
        ENDIF

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

    RETURN
    END
```

Subroutine **WIND** (cont'd)

```
C****************************************************************
C
C     TERRA                                      MESOI  VERSION 2.0
C     Terrain modification of wind field
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: This module is called after the wind subroutine has
C       decoded the meteorological data into u and v components. It
C       modifies the components for each grid point according to the
C       terrain.
C
C     Relationship to other modules:
C
C         Makes calls to:  DIRSPD
C
C         Called from:     MESOI
C
C****************************************************************

        SUBROUTINE TERRA

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

        REAL COEFF1(16,16,2), TANGLE(16,16)

        DATA COEFF1 / 512 *  1.0 /
        DATA TANGLE / 256 * -1.0 /

C   **  SET COEFFICIENTS TO DEFINE THE TERRAIN SLOPE: EACH NUMBER
C   **    REPRESENTS THE FRACTION OF THE COMPONENT PERPENDICULAR
C   **    TO THE TERRAIN FEATURE FOR EACH GRID POINT. THE GREATER THE
C   **    SLOPE, THE SMALLER THE COEFFICIENT. LEVEL SFCS = 1.0

        DATA DTR        /.017453/

C        TERRAIN DATA IS STORED IN FILE CALLED TERRA.DAT
C        EACH RECORD CONSISTS OF:
C               COL  1- 2:        X GRID DESIGNATOR 0-15, LOWER LEFT=0
C               COL  4- 5:        Y GRID DESIGNATOR 0-15, LOWER LEFT=0
C               COL  7-10:        ANGLE IN DEGREE FROM HORIZONTAL 0-179.999
C               COL 12-15:        1ST COEFFICIENT; FOR WINDS FROM EAST
C               COL 17-20:        2ND COEFFICIENT; FOR WINDS FROM WEST

        IF (TERFLG .EQ. 1)  GOTO 140

        OPEN (UNIT=LUN(15), FILE='TERRA.DAT', STATUS='OLD',
     +        IOSTAT=IER)

        IF (IER .NE. 0)  THEN
          PRINT *, ' TERRA >> ERROR OPENING FILE; USE FLAT TERRAIN'
          GOTO 140
        ENDIF

  100   READ (LUN(15), 110, END=130, ERR=120)
     +    I, J, TANGLE(I+1,J+1), COEFF1(I+1,J+1,1), COEFF1(I+1,J+1,2)
  110   FORMAT (1X,I2,1X,I2,1X,F4.0,1X,F4.0,1X,F4.0)
        GOTO 100
```

Program Listing 22 Subroutine TERRA

```
  100   READ (LUN(15), 110, END=130, ERR=120)
     +    I, J, TANGLE(I+1,J+1), COEFF1(I+1,J+1,1), COEFF1(I+1,J+1,2)
  110   FORMAT (1X,I2,1X,I2,1X,F4.0,1X,F4.0,1X,F4.0)
        GOTO 100

  120   STOP 'TERRA RD ERR'

  130   CONTINUE
        CLOSE (LUN(15))
        TERFLG = 1

C   **  DEFINE THE ANGLE OF THE TERRAIN WITH RESPECT TO THE XY COORDINATE
C   **    SYSTEM FOR EACH GRID POINT. IF THERE IS NO DEFINED TERRAIN FOR
C   **    A POINT, ANGLE IS SET TO -1.

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

  140   DO 200, I = 1,16

          DO 180, J = 1,16

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

            IF( TANGLE(I,J) .LT. 0.0 ) GOTO 160

C   **  CALCULATE THE SPEED AND DIRECTION, THEN CONVERT DIRECTION TO
C   **    ANGLE OF THE WIND VECTOR IN XY COORD SYSTEM

            CALL DIRSPD (UU(I,J), VV(I,J), ANG0, SPEED)
            ANG0 = 270 - ANG0
            IF (ANG0 .LT. 0) ANG0 = ANG0 + 360

C   **  DETERMINE IF WIND IS APPROACHING THE TERRAIN LINE FROM EAST OR WEST

            IF (ANG0 .GE. 90.0 .OR. ANG0 .LT. 270.0 ) THEN

C   **   WIND FROM WEST

              K = 2
            ELSE

C   **   WIND FROM EAST

              K = 1
            ENDIF

C -------------- DETERMINE RELATIVE ANGLE OF WIND TO TERRAIN -------------

            RANGLE = (ANG0 - TANGLE(I,J)) * DTR

C   **  CALCULATE THE COMPONENTS PARALLEL AND PERPENDICULAR TO TERRAIN;
C   **    THEN TRANSLATE TO AN ANGLE AND SPEED

            U1 = SPEED * COS( RANGLE )
            V1 = SPEED * SIN( RANGLE )
            CALL DIRSPD (U1, V1, ANG1, S)

C   **  MODIFY THE COMPONENTS: PERPENDICULAR IS MULTIPLIED BY A COEFFICIENT
C   **    TO DECREASE IT IN PROPORTION TO THE TERRAIN SLOPE, PARALLEL
C   **    COMPONENT IS INCREASED TO MAINTAIN THE TOTAL VECTOR MAGNITUDE.

            V2 = V1 * COEFF1(I,J,K)
            IF (U1 .EQ. 0.0) THEN
              U2 = 0.0
```

Subroutine TERRA (cont'd)

```
          ELSE
            U2 = U1 * (((SPEED**2) - (V2**2))**0.5 / ABS(U1))
          ENDIF

          CALL DIRSPD (U2, V2, ANG2, SPEED)

          ANG0 = ((ANG1 - ANG2) + ANG0) * DTR
          UU(I,J) = SPEED * COS( ANG0 )
          VV(I,J) = SPEED * SIN( ANG0 )

160       CONTINUE

180     CONTINUE
200   CONTINUE

      RETURN
      END
```

Subroutine TERRA (cont'd)

```
C***********************************************************************
C
C     ALPHA                                          MESOI   VERSION 2.0
C     Wind vertical interpolation exponent
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Dummy subroutine; Could be configured to compute
C       an alpha value for use in the vertical interpolation of the
C       winds.
C
C     Pelationship to other modules:
C
C         Nakes calls to:  NONE
C
C         Called from:     MESOI
C
C***********************************************************************
        SUBROUTINE ALPHA (STAB, ALPHAU, ALPHAV)

        REAL ALPHAU, ALPHAV

        INTEGER STAB

        ALPHAU = 1.0
        ALPHAV = 1.0

        RETURN
        END
```

Program Listing 23 Subroutine ALPHA

```
C*****************************************************************
C
C     WINSPD                                        MESOI  VERSION 2.0
C     Wind speed interpolation
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to calculate the wind speed at a release
C       point it interpolates in space and time. Speed is used in
C       the plume rise computation.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE WINSPD (NS, TINC, SPEED)

      INCLUDE 'CONST.INC'
      INCLUDE 'REL.INC'
      INCLUDE 'WINDS.INC'

      I = XSOURC(NS)+ 1
      J = YSOURC(NS) + 1
      C2 = TINC
      IF (INTFLG .EQ. 1)  C2 = 0.5
      C1 = 1.0 - C2

      RX = XSOURC(NS) - (I - 1)
      RY = YSOURC(NS) - (J - 1)
      RX1 = 1.0 - RX
      RY1 = 1.0 - RY

      UCOMP = (RX1*RY1) * (C1*U(I,J) + C2*UU(I,J))
      UCOMP = UCOMP + (RX1*RY) * (C1*U(I,J+1) + C2*UU(I,J+1))
      UCOMP = UCOMP + (RX*RY) * (C1*U(I+1,J+1) + C2*UU(I+1,J+1))
      UCOMP = UCOMP + (RX*RY1) * (C1*U(I+1,J) + C2*UU(I+1,J))

      VCOMP = (RX1*RY1) * (C1*V(I,J) + C2*VV(I,J))
      VCOMP = VCOMP + (RX1*RY) * (C1*V(I,J+1) + C2*VV(I,J+1))
      VCOMP = VCOMP + (RX*RY) * (C1*V(I+1,J+1) + C2*VV(I+1,J+1))
      VCOMP = VCOMP + (RX*RY1) * (C1*V(I+1,J) + C2*VV(I+1,J))

      SPEED = (UCOMP * UCOMP + VCOMP * VCOMP) ** 0.5

      RETURN
      END
```

Program Listing 24 Subroutine WINSPD

```
C*******************************************************************
C
C     PLMRIZ                                        MESOI  VERSION 2.0
C     Plume rise module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to calculate the final plume rise given
C       prescribed conditions of affluent release. Called only if the
C       release is from a defined stack.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*******************************************************************

      SUBROUTINE PLMRIZ (NS, AIRT, SPD, PRISE)

      INCLUDE 'CHAR.INC'
      INCLUDE 'DATIM.INC'
      INCLUDE 'REL.INC'
      INCLUDE 'UNITS.INC'

      ZS = ZSOURC(NS)

C   ** IF STACK HEIGHT GREATER THAN OR EQUAL TO MIXED DEPTH, NO PLUME RISE

      IF (ZS .GE. LDEPTH)  THEN
        PRISE = 0.0
        GOTO 140
      ENDIF

C   ** CALCULATE A BUOYANCY FLUX FROM THE HEAT FLUX

      FB = 3.119 * QH(NS) * (STEMPC - AIRT) / (AIRT + 273.15)

C   ** FOR STABLE CONDITIONS:

      IF (STAB .GE. 5)  THEN
        GOTO (100, 110, 120) STAB - 4

C   ** CALCULATE A RESTORING FORCE (S) BASED ON AN ASSUMED TEMPERATURE
C   **   GRADIENT IN THE STABLE LAYER

 100    S = 0.049 / ( AIRT + 273.15 )
        GOTO 130

 110    S = 0.27 / ( AIRT + 273.15 )
        GOTO 130

 120    S = 0.49 / ( AIRT + 273.15 )
 130    CONTINUE

C   ** SPEEDS LESS THAN 1.0 M/SEC = CALM

        IF (SPD .LE. 1.0)  THEN
```

Program Listing 25 Subroutine PLMRIZ

```
            PRISE = 5.3 * (FB ** 0.25) / (S ** 0.375) - 6.0
          ELSE
            PRISE = 2.6 * (FB / (SPD * S)) ** (1. / 3.)
          ENDIF

        ELSE

C   **  FOR NEUTRAL AND UNSTABLE CONDITIONS:

          IF (SPD .LT. 0.5)  SPD = 0.5
          FLUX = 68.8 * QH(NS) * (STEMPC - AIRT)
          IF (FLUX .LT. 20.0E6)  THEN
            X = 6.49 * (FB ** (2./5.)) * (ZS ** (3./5.))
          ELSE
            X = 10.0 * ZS
          ENDIF

          PRISE = 1.6 / SPD * (FB ** (1./3.)) * (X ** (2./3.))
        ENDIF

  140   IF (TITLE(1) .EQ. '*')  THEN
          WRITE (LUN(2),150) NS
          WRITE (LUN(5),150) NS
  150     FORMAT (/5X, 'PLUME RISE PARAMETERS FOR SOURCE ', I2)

          WRITE (LUN(2),160) QH(NS), AIRT, SPD, PRISE
          WRITE (LUN(5),160) QH(NS), AIRT, SPD, PRISE
  160     FORMAT (/5X, 'SENSIBLE HEAT FLUX (WATTS) = ', 1PE10.3,
     +             /5X, '   AIR TEMPERATURE (DEG C) = ', 0PF5.1,
     +             /5X, 'WIND SPEED AT SOURCE (M/S) = ', F5.1,
     +             /5X, 'CALCULATED PLUME RISE (M)  = ', F6.1)
        ENDIF

        RETURN
        END
```

Subroutine PLMRIZ (cont'd)

```
C*****************************************************************
C
C     PUFFR                                        MESOI  VERSION 2.0
C     Puff release module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glartz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Assigns initial attributes to puffs at release
C       time.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE PUFFR (NS, PRISE, LDEPTH)

      INCLUDE 'CONST.INC'
      INCLUDE 'DECAY.INC'
      INCLUDE 'PUFFS.INC'
      INCLUDE 'REL.INC'

      NPUFFS(NS) = NPUFFS(NS) + 1
      TPUFFS = TPUFFS + 1

      MF(TPUFFS) = 1
      XP(TPUFFS) = XSOURC(NS)
      YP(TPUFFS) = YSOURC(NS)
      ZP(TPUFFS) = ZSOURC(NS) + PRISE
      SP(TPUFFS) = NS
      AGE(TPUFFS) = 0

C   **  NO PENETRATION OF LIO DUE TO PLUME RISE

      IF (ZP(TPUFFS). GT. LDEPTH .AND. PRISE .NE. 0.0)
     +     ZP(TPUFFS) = LDEPTH
      QP(TPUFFS) = QSOURC(NS) * ACONST
      ADQP(TPUFFS) = 0.0
      BDQP(TPUFFS) = 0.0

      IF (DPFLG .EQ. 1 .OR. RDKFL .EQ. 1)  ADQP(TPUFFS) = QP(TPUFFS)
      SIGMAZ(TPUFFS) = 0.1
      SIGMAY(TPUFFS) = 1.0

      RETURN
      END
```

Program Listing 26 Subroutine PUFFR

```
C**********************************************************************
C
C     PUFFM                                          MESO1  VERSION 2.0
C     Puff movement module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Calculates movement of each puff for each
C       advection step.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESO1
C
C**********************************************************************

      SUBROUTINE PUFFM (M, TINC)

      INCLUDE 'CONST.INC'
      INCLUDE 'DATIM.INC'
      INCLUDE 'PUFFS.INC'
      INCLUDE 'WINDS.INC'

      REAL XG(2), YG(2), DXSFC(2), DYSFC(2), DUX(2), DUY(2),
     +     C1, C2, RX, RY, RX1, RY1, CST, ZFU, ZFV

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

      IF (ZP(M) .GE. LDEPTH)  THEN
        DO 100 K = 1, 2
          C2 = TINC + (FLOAT(K) - 1.0) * ADT
          IF (INTFLG .EQ. 1)  C2 = 0.5
          C1 = 1.0 - C2
          DUX(K) = (C1 * UG + C2 * UUG) * FAC
          DUY(K) = (C1 * VG + C2 * VVG) * FAC
  100   CONTINUE
        DXS(M) = (DUX(1) + DUX(2)) / 2.0
        DYS(M) = (DUY(1) + DUY(2)) / 2.0
        RETURN
      ENDIF

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

C   **  TWO STEP ITERATION OF THE ADVECTION

      DO 200 K = 1, 2
        C2 = TINC + (FLOAT(K) - 1.0) * ADT
        IF (INTFLG .EQ. 1)  C2 = 0.5
        C1 = 1.0 - C2
        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)
        RX1 = 1.0 - RX
```

Program Listing 27 Subroutine PUFFM

```
        RY1 = 1.0 - RY

          UCOMP = (RX1*RY1) * (C1*U(I,J) + C2*UU(I,J))
          UCOMP = UCOMP + (RY*RX1) * (C1*U(I,J+1) + C2*UU(I,J+1))
          UCOMP = UCOMP + (RX*RY) * (C1*U(I+1,J+1) + C2*UU(I+1,J+1))
          UCOMP = UCOMP + (RX*RY1) * (C1*U(I+1,J) + C2*UU(I+1,J))
        DXSFC(K) = UCOMP * FAC

          VCOMP = (RX1*RY1) * (C1*V(I,J) + C2*VV(I,J))
          VCOMP = VCOMP + (RY*RX1) * (C1 * V(I,J+1) + C2*VV(I,J+1))
          VCOMP = VCOMP + (RX*RY) * (C1*V(I+1,J+1) + C2*VV(I+1,J+1))
          VCOMP = VCOMP + (RX*RY1) * (C1*V(I+1,J) + C2*VV(I+1,J))
        DYSFC(K) = VCOMP * FAC

        IF (ZP(M) .GT. 10.0)  THEN
          DUX(K) = (C1 * UG + C2 * UUG) * FAC
          DUY(K) = (C1 * VG + C2 * VVG) * FAC
        ENDIF

        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
        DUXM = (DUX(1) + DUX(2)) / 2.0
        DUYM = (DUY(1) + DUY(2)) / 2.0

        ZFU = ((ZP(M) -10.0) / (LDEPTH - 10.0)) ** ALPHAU
        ZFV = ((ZP(M) -10.0) / (LDEPTH - 10.0)) ** ALPHAV
        DXS(M) = DXS(M) + (DUXM - DXS(M)) * ZFU
        DYS(M) = DYS(M) + (DUYM - DYS(M)) * ZFV
      ENDIF

      RETURN
      END
```

Subroutine PUFFM (cont'd)

```
C*****************************************************************
C
C   DTOPO                                        MESOI  VERSION 2.0
C   Diffusion topography module
C
C   Athey, G.F., J.V. Ramsdell, C.S. Glantz
C   Pacific Northwest Laboratory
C   PO Box 999
C   Richland, Washington  99352
C
C   Created: 6/1/83
C   Updated:
C
C   Description: Computes terrain elevation under the center of
C     the puffs at the beginning of the advection step.
C
C   Relationship to other modules:
C
C       Makes calls to:  NONE
C
C       Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE DTOPO (XP, YP, TPMI, DXS, DYS, TPMF)

      INCLUDE 'TOPOGR.INC'

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

C  ** 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

C  **  DETERMINE SUBSCRIPTS DEFINING THE SURROUNDING GRID POINT
C  **    ON THE EXPOSURE MATRIX

        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

          X1 = XPFG(IT) - IX1
          X2 = 1.0 - X1

          Y1 = YPFG(IT) - IY1
          Y2 = 1.0 - Y1

          IF ( X1 .LT. 0.001 .AND. Y1 .LT. 0.001 ) THEN
            THT(IT) = TOPO(IX1,IY1)
            GOTO 100
          ENDIF
```

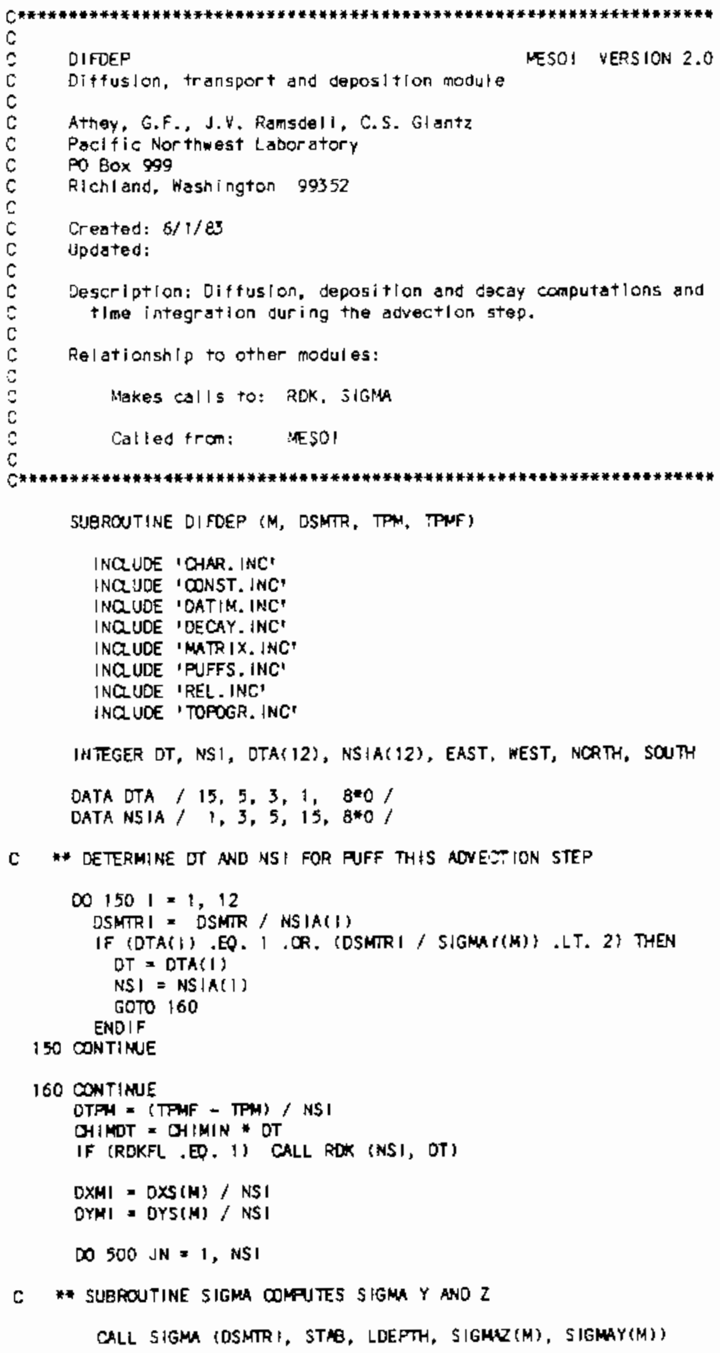
Program Listing 28 Subroutine DTOPO

```
            WT(1) = 1.0 / SQRT(X1*X1 + Y1*Y1)
            WT(2) = 1.0 / SQRT(X2*X2 + Y1*Y1)
            WT(3) = 1.0 / SQRT(X1*X1 + Y2*Y2)
            WT(4) = 1.0 / SQRT(X2*X2 + Y2*Y2)

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

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

C   **  COMPUTE TERRAIN HEIGHT UNDER PUFF CENTER

            THT(IT) = WT(1) * TOPO(IX1,IY1) +
     1                WT(2) * TOPO(IX2,IY1) +
     2                WT(3) * TOPO(IX1,IY2) +
     3                WT(4) * TOPO(IX2,IY2)
          ELSE

C   **  PUFF IS OFF THE GRID

            IF (IX1 .LT. 1)  THEN

C  **  PUFF IS WEST OF THE GRID **
              IF (IY1 .LT. 1)  THEN
                THT(IT) = TOPO(1,1)
              ELSE
                IF (IY2 .GT. 31)  THEN
                  THT(IT) = TOPO(1,31)
                ELSE
                  WT(2) = YPFG(IT) - IY1
                  WT(1) = 1.0 - WT(2)
                  THT(IT) = WT(1) * TOPO(1,IY1) + WT(2) * TOPO(1,IY2)
                ENDIF
              ENDIF
            ELSE
              IF (IX2 .GT. 31)  THEN

C   **  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

C   **  PUFF IS NORTH OF GRID  **

                  THT(IT) = WT(1) * TOPO(IX1,1) + WT(2) * TOPO(IX2,1)
                ELSE

C   **  PUFF IS SOUTH OF GRID  **
```

Subroutine DTOPO (cont'd)

```
                THT(IT) = WT(1) * TOPO(IX1,31) + WT(2) * TOPO(IX2,31)
              ENDIF
            ENDIF
          ENDIF
        ENDIF
100   CONTINUE

      TPMI = THT(1)
      TPMF = THT(2)
      RETURN
      END
```

Subroutine DTOPO (cont'd)



```
C*****************************************************************
C
C     DIFDEP                                      MESOI  VERSION 2.0
C     Diffusion, transport and deposition module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Diffusion, deposition and decay computations and
C       time integration during the advection step.
C
C     Relationship to other modules:
C
C         Makes calls to:  RDK, SIGMA
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE DIFDEP (M, DSMTR, TPM, TPMF)

        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 /

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

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

  160 CONTINUE
      DTPM = (TPMF - TPM) / NSI
      CHIMDT = CHIMIN * DT
      IF (RDKFL .EQ. 1)  CALL RDK (NSI, 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))
```

Program Listing 29 Subroutine DIFDEP

```
        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 =(SQRT(TWOPI)/2.0) * QP(M) * DT / (SIGYSQ * UDEPTH)

            IF ( PUFCHI .LE. CHIMDT ) THEN
              GO TO 200
            ELSE
              CHIR = ALOG(PUFCHI/CHIMDT)
              RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
              RADP(M) = ( SIGMAY(M) * SQRT(2 * ALOG(PUFCHI/
     +                     1.0E-14)) / DELXY )
              VEXP = 1.0
            ENDIF

          ELSE

C    **        REFLECTED GAUSSIAN DISTRIBUTION IN THE VERTICAL

            PUFCHI = QP(M) * DT / (SIGYSQ * SIGMAZ(M))

            IF (PUFCHI .LE. CHIMDT)  THEN
              GOTO 200
            ELSE
              CHIR = ALOG(PUFCHI/CHIMDT)
              RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
              RADP(M) = ( SIGMAY(M) * SQRT(2 * ALOG(PUFCHI/
     +                     1.0E-14)) / 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 = (SQRT(TWOPI)/2.0) * QP(M) * DT / (SIGYSQ*UDEPTH)

            IF (PUFCHI .LE. CHIMDT) THEN
              GOTO 200
            ELSE
              CHIR = ALOG(PUFCHI/CHIMDT)
              RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
              RADP(M) = ( SIGMAY(M) * SQRT(2 * ALOG(PUFCHI/
     +                     1.0E-14)) / DELXY )
              VEXP = 1.0
```

Subroutine DIFDEP (cont'd)

```
            ENDIF

           ELSE

C    **          REFLECTED GAUSSIAN DISTRIBUTION IN THE VERTICAL

             PUFCHI = QP(M) * DT / (SIGYSQ*SIGMAZ(M))

             IF (PUFCHI .LE. CHIMDT ) THEN
               GOTO 200
             ELSE
               CHIR = ALOG(PUFCHI/CHIMDT)
               RPEW = (SIGMAY(M) * SQRT(2 * CHIR) / DELXY)
               RADP(M) = ( SIGMAY(M) * SQRT(2 * ALOG(PUFCHI/
     +                    1.0E-14)) / DELXY )
               VEXP = EXP(-0.5*ZP(M)*ZP(M)/SIGZSQ) +
     +                EXP(-0.5*(2*LDEPTH-ZP(M))**2/SIGZSQ) +
     +                EXP(-0.5*(2*LDEPTH+ZP(M))**2/SIGZSQ) +
     +                EXP(-0.5*(4*LDEPTH-ZP(M))**2/SIGZSQ)
             ENDIF

           ENDIF

        ENDIF

C    **  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. ADQP(M) .LE. 0.0)  THEN
            TPM    = TPM + 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

C    **  RADIOACTIVE DECAY OF THE PUFF BEFORE THE NEXT INTERVAL

     IF (RDKFL .EQ. 1)  THEN
              BDQP(M) = BDQP(M) * BDECA2 + ADQP(M) * BINS1
       ADQP(M) = ADQP(M) * ADECA2
     ENDIF
            GDTO  499
          ENDIF

        ENDIF

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

        TPM    = TPM + 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

C    **  CALCULATE THE EFFECTIVE GROUND RADIUS OF THE PUFF

  210   IF (ZP(M) .LE. 10.0)  THEN
```

Subroutine DIFDEP (cont'd)

```
          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

C   **  PUFCX AND PUFCY ARE THE COORDINATES OF THE PUFF ON THE FINE GRID
C   **    EAST, WEST, NORTH AND SOUTH ARE EDGE COORDINATES
C   **    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

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

        DO 310 I = WEST, EAST
          DO 300 J = SOUTH, NORTH
            RSQ = (((XP(M)*2.0) - (I-1))**2) +
     +        (((YP(M)*2.0) - (J-1))**2)
            IF (RSQ .GT. (RPGE*RPGE))  GOTO 300

            IF (UDEPTH .EQ. 0.0) THEN
              HEFF = ZP(M) - (TOPO(I,J) - TPM)
              VEXP = EXP(HEFF ** 2 / (-2.0 * SIGZSQ))
     +             + EXP((2.0 * LDEPTH + HEFF) ** 2 / (-2.0 * SIGZSQ))
     +             + EXP((4.0 * LDEPTH + HEFF) ** 2 / (-2.0 * SIGZSQ))
              IF (ZP(M) .LE. LDEPTH)  THEN
                VEXP = VEXP
     +               + EXP((-2.0 * LDEPTH + HEFF) ** 2 / (-2. * SIGZSQ))
     +               + EXP((-4.0 * LDEPTH + HEFF) ** 2 / (-2. * SIGZSQ))
              ENDIF
            ENDIF

            HEXP = EXP(HSGSQ*RSQ)
            PCHI = PUFCHI * HEXP * VEXP
            EXPCUM(I,J) = EXPCUM(I,J) + PCHI

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

C   **      COMPUTE THE DRY DEPOSITION

            IF (DPFLG .EQ. 1)  THEN

              AQR = ADQP(M) / QP(M)
              ADPCHI = PCHI * AQR

              IF (RDKFL .EQ. 1) THEN
                BQR = BDQP(M) / QP(M)
```

Subroutine DIFDEP (cont'd)

```
              BDPCHI = PCHI * BQR
            ENDIF

C    **     COMPUTE EXPOSURE FROM DEPLETED PUFFS

            AXPADV(I,J) = AXPADV(I,J) + ADPCHI
            IF (RDKFL .EQ. 1) BXPADV(I,J) = BXPADV(I,J) + BDPCHI

C    **     COMPUTE DRY DEPOSITION

            ADEPT = DV * ADPCHI
            IF (RDKFL .EQ. 1) THEN
              BADEPT = ADEPT * BDIN(JN)
              ADPADV(I,J) = ADPADV(I,J) + ADEPT * ADK(JN)
              BDPADV(I,J) = BDPADV(I,J) + BADEPT +
     +                         ( DV * BDPCHI ) * BDK(JN)
            ELSE
              ADPADV(I,J) = ADPADV(I,J) + ADEPT
            ENDIF
          ENDIF

  300     CONTINUE
  310   CONTINUE

C    ** COMPUTE THE WET DEPOSITION

  400   CONTINUE

C    ** IF WE HAVE WET DEPOSITION OF A SIGNIFICANT PUFF

        IF (PRECIP .GT. 0 .AND. DPFLG .EQ. 1)  THEN

          WCDT = WC(PRECIP) * DT
          IF (WCDT .GT. 60.)  WCDT = 60.
          WOCOEF = (WCDT / SIGYSQ) * (SQRT TWOPI) / 2.0)
          AWSHO = WOCOEF * ADQP(M)
          IF (RDKFL .EQ. 1) BWSHO = WOCOEF * BDQP(M)

          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
              HEXP = EXP(HSGSQ * RSQ)
              ADEPT = AWSHO * HEXP

              IF (RDKFL .EQ. 1) THEN
                BADEPT = ADEPT * BDIN(JN)
                ADPADV(I,J) = ADPADV(I,J) + ADEPT * ADK(JN)
                BDPADV(I,J) = BDPADV(I,J) + BADEPT
                IF (BDQP(M) .GT. 0.0)  THEN
```

Subroutine DIFDEP (cont'd)

```
                  BADEPT = BWSHO * HEXP * BDK(JN)
                  BDPADV(I,J) = BDPADV(I,J) + BADEPT
                ENDIF
              ELSE
                ADPADV(I,J) = ADPADV(I,J) + ADEPT
              ENDIF

410         CONTINUE
420       CONTINUE

        ENDIF

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

        IF (RDKFL .EQ. 1)  THEN
          BDQP(M) = BDQP(M) * BDECA2 + ADQP(M) * BINSI
          ADQP(M) = ADQP(M) * ADECA2
        ENDIF

C   **  DEPLETE BY DRY DEPOSITION

        IF (DPFLG .EQ. 1)  THEN
          IF (GCPCHI .GT. 0.0)  THEN
            CFACT = 1 - (2 * SIGYSQ * DV * GCPCHI) / (SQRT(TWOPI) *
     +        60.0 * QP(M))
            IF (ADQP(M) .GT. 0.0) THEN
              ADQP(M) = ADQP(M) * CFACT
            ENDIF

            IF ( RDKFL .EQ. 1 )  THEN
              BDQP(M) = BDQP(M) * CFACT
            ENDIF

          ENDIF

C   **  DEPLETE BY WET DEPOSITION

          IF (PRECIP .GT. 0)  THEN
            ROUT = 1.0 - WC(PRECIP) * DT/60.
            IF (ROUT .LT. 0.0)  ROUT = 0.0
            ADQP(M) = ADQP(M) * ROUT
            IF ( RDKFL .EQ. 1 ) BDQP(M) = BDQP(M) * ROUT
          ENDIF
        ENDIF

  499     AGE(M) = AGE(M) + DT

  500 CONTINUE

      RETURN
      END
```

Subroutine DIFDEP (cont'd)

```
C*****************************************************************
C
C     RDK                                         MESOI  VERSION 2.0
C     Radioactive decay module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 5/1/83
C     Updated:
C
C     Description: Module to calculate the necessary constants for
C       use in the decay calculations.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     RELEAS
C
C*****************************************************************

      SUBROUTINE RDK (NSI, DT)

      INCLUDE 'DECAY.INC'
      INTEGER DT, NSI

      ADECA2 = EXP(-ALMBDA * DT)

      BDECA2 = EXP(-BLMBDA * DT)

      IF (ALMBDA .NE. BLMBDA)  THEN
        BINSI = DKCOEF * (ADECA2 - BDECA2)
      ELSE
        BINSI = ALMBDA * DT * ADECA2
      ENDIF

      DO 300 I = 1, NSI
        ADK(I) = EXP(-ALMBDA * ( NSI + 1 - I ) * DT)
        BDK(I) = EXP(-BLMBDA * ( NSI + 1 - I ) * DT)

        IF (ALMBDA .NE. BLMBDA)  THEN
          BDIN(I) = DKCOEF * (EXP(-ALMBDA * ( NSI + 1 - I ) * DT)
     +                     - EXP(-BLMBDA * ( NSI + 1 - I ) * DT))
        ELSE
          BDIN(I) = ALMBDA * ( NSI + 1 - I ) * DT
     +               * EXP(-ALMBDA * ( NSI + 1 - I ) * DT)
        ENDIF

  300 CONTINUE

      RETURN
      END
```

Program Listing 30 Subroutine RDK

```
C*****************************************************************
C
C     ASIG                                          MESOI  VERSION 2.0
C     Army Diffusion Curves
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Computes new diffusion coefficients given the
C       last values, atmospheric stability, mixing layer thickness
C       and distance moved.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     DIFDEP
C
C*****************************************************************

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

      INTEGER STAB

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

      DATA AY/ 0.37, 0.30, 0.20, 0.15, 0.10, 0.10, 0.10 /
      DATA AZ/ 0.22, 0.20, 0.15, 0.10, 0.05, 0.05, 0.05 /
      DATA BZ/ 0.0, 0.0, 0.0002, 0.0015, 0.0003, 0.0003, 0.0003 /
      DATA CZ/ 0.0, 0.0, -0.5, -0.5, -1.0, -1.0, -1.0 /

      XVY = (SIGMAY/AY(STAB))**(10.0/9.0)
      IF ( XVY .LT. 10000.0)  THEN

        XEY = XVY + DSMTRI
        IF ( XEY .LT. 10000.0)  THEN
          SIGMAY = AY(STAB) * XEY ** 0.9
        ELSE
          SIGMAY = AY(STAB) *0.398 * XEY
        END IF

      ELSE

        XVY = SIGMAY / ( 0.398 * AY(STAB) )
        XEY = XVY + DSMTRI
        SIGMAY = AY(STAB) * 0.398 * XEY

      ENDIF

C     CHECK SIGMA Z INITIAL SIZE AGAINST MAXIMUM SIZE

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

C     COMPUTE THE VIRTUAL DISTANCE FOR SIGMA Z

      IF ( CZ(STAB) .EQ. 0.0 ) THEN
        XVZ   = SIGMAZ / AZ(STAB)

      ELSE IF ( CZ(STAB) .EQ. -0.5 ) THEN
```

Program Listing 31 Subroutine ASIG

```
        R = SIGMAZ/AZ(STAB)
        R2 = R * R
        XVZ = 0.5 * ( R2 * BZ(STAB) + R * SQRT(R2 * BZ(STAB)**2 + 4.0) )

        ELSE IF ( CZ(STAB) .EQ. -1.0 ) THEN
          IF ( SIGMAZ .GE. AZ(STAB)/BZ(STAB) ) RETURN
          XVZ = SIGMAZ / (AZ(STAB)-BZ(STAB)*SIGMAZ)
      ENDIF

C     COMPUTE SIGMAZ

      XEZ = XVZ + DSMTRI
      SIGMAZ = AZ(STAB) * XEZ * ( 1.0 + BZ(STAB) * XEZ ) ** CZ(STAB)
      IF (SIGMAZ .GT. SZLIM)  SIGMAZ = SZLIM

      RETURN
      END
```

Subroutine ASIG (cont'd)

```
C*****************************************************************
C
C     BSIG                                         MESOI  VERSION 2.0
C     Brigg's Open Country Diffusion Curves
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Computes new diffusion coefficients given the
C       last values, atmospheric stability, mixing layer thickness
C       and distance moved.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     DIFDEP
C
C*****************************************************************

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

      INTEGER STAB

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

      DATA AY/ 0.22, 0.16, 0.11, 0.08, 0.06, 0.04, 0.027 /
      DATA AZ/ 0.20, 0.12, 0.08, 0.06, 0.03, 0.016, 0.011 /
      DATA BZ/ 0.0, 0.0, 0.0002, 0.0015, 0.0003, 0.0003, 0.0003 /
      DATA CZ/ 0.0, 0.0, -0.5, -0.5, -1.0, -1.0, -1.0 /

C  **  COMPUTE THE CHANGE IN SIGMA Y

      BY  = 1.0 E-4
      CY  = -0.5
      R   = SIGMAY/AY(STAB)
      R2  = R * R
      XVY = 0.5 * ( R2 * BY + R * SQRT(R2 * BY**2 + 4.0) )
      XEY = XVY + DSMTRI
      SIGMAY = AY(STAB) * XEY * ( 1.0 + BY * XEY )**CY

C  **  SIGMA Z

C  **  CHECK SIGMA Z INITIAL SIZE AGAINST MAXIMUM SIZE

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

C  ** COMPUTE THE VIRTUAL DISTANCE FOR SIGMA Z

      IF ( CZ(STAB) .EQ. 0.0 ) THEN
        XVZ   = SIGMAZ / AZ(STAB)

      ELSE IF ( CZ(STAB) .EQ. -0.5 ) THEN
        R = SIGMAZ/AZ(STAB)
        R2 = R * R
        XVZ = 0.5 * ( R2 * BZ(STAB) + R *
     +        SQRT(R2 * BZ(STAB)**2 + 4.0) )

      ELSE IF ( CZ(STAB) .EQ. -1.0 ) THEN
```

Program Listing 32 Subroutine BSIG

```
  IF ( SIGMAZ .GE. AZ(STAB)/BZ(STAB) ) RETURN
    XVZ = SIGMAZ / (AZ(STAB)-BZ(STAB)*SIGMAZ)
ENDIF

XEZ = XVZ + DSMTRI
SIGMAZ = AZ(STAB) * XEZ * ( 1.0 + BZ(STAB) * XEZ ) ** CZ(STAB)
IF (SIGMAZ .GT. SZLIM)  SIGMAZ = SZLIM

RETURN
EMD
```

Subroutine BSIG (cont'd)

```
C********************************************************************
C
C     DSIG                                          MESOI  VERSION 2.0
C     Desert Diffusion Curves
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Computes new diffusion coefficients given the
C       last values, atmospheric stability, mixing layer thickness
C       and distance moved.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     DIFDEP
C
C********************************************************************

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

      REAL AY(7), BY(7), AZ(7), BZ(7)

      INTEGER STAB

      DATA AY/  0.718, 0.425, 0.349, 0.267, 0.299, 0.401, 0.401 /
      DATA AZ/  0.100, 0.105, 0.128, 0.146, 0.331, 0.812, 0.812 /
      DATA BY/ 23.0,  13.6,  11.2,   8.55,  9.57, 12.8,  12.8   /
      DATA BZ/  1.033, 0.975, 0.891, 0.824, 0.567, 0.307, 0.307 /

C  **  SIGMA Y COMPUTATION

      XVY = ( SIGMAY / AY(STAB) )**(1.0/0.85)
      IF ( XVY .LT. 20000 ) THEN
        XEY = XVY + DSMTRI
        IF ( XEY .LT. 20000 ) THEN
          SIGMAY = AY(STAB) * XEY**0.85
        ELSE
          SIGMAY = BY(STAB) * SQRT(XEY)
        ENDIF
      ELSE
        XVY = ( SIGMAY / BY(STAB) )**2
        XEY = XVY + DSMTRI
        SIGMAY = BY(STAB) * SQRT(XEY)
      ENDIF

C  **  SIGMA Z COMPUTATION

C  **  CHECK SIGMA Z INITIAL SIZE AGAINST MAXIMUM SIZE

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

      XZCHNG = ( 0.465 * LDEPTH / AZ(STAB) )**(1.0/BZ(STAB))
      IF ( SIGMAZ/LDEPTH .LT. 0.465 ) THEN

        XVZ = ( SIGMAZ / AZ(STAB) )**( 1.0/BZ(STAB) )
        XEZ = XVZ + DSMTRI
```

Program Listing 33 Subroutine DSIG

```
   IF ( XEZ .LE. XZCHNG ) THEN
     SIGMAZ = AZ(STAB) * XEZ**BZ(STAB)
   ELSE
     SIGMAZ = ( 0.465 + 0.335 * ( XEZ - XZCHNG ) / XZCHNG )
+                 * LDEPTH
   END IF

 ELSE

   XVZ = ( ( SIGMAZ/LDEPTH - 0.465 ) / 0.335 + 1.0 )
+             * XZCHNG
   XEZ = XVZ + DSMTRI
   SIGMAZ = ( 0.465 + 0.335 * ( XEZ - XZCHNG ) / XZCHNG )
+                * LDEPTH

 ENDIF
 IF (SIGMAZ .GT. SZLIM) SIGMAZ = SZLIM

 RETURN
 END
```

Subroutine DSIG (cont'd)

```
C*****************************************************************
C
C     NRCSIG                                       MESOI  VERSION 2.0
C     Diffusion Curves As Used in XOQDOQ and PAVAN
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Computes new diffusion coefficients given the
C       last values, atmospheric stability, mixing layer thickness
C       and distance moved.
C
C     Relationsnip to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     DIFDEP
C
C*****************************************************************

      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,
     +         0.00066,0.0382,0.113, 0.222, 0.211, 0.086, 0.052,
     +         0.00024,0.055, 0.113, 1.26,  6.73, 18.05, 10.53 /
      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) )**1.107
      XEY = XVY + DSMTRI
      SIGMAY = AY(STAB) * XEY**0.9031

C  **  SIGMA Z COMPUTATIONS

C  **  CHECK INITIAL SIGMA Z SIZE AGAINST MAXIMUM

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

C  ** COMPUTE VIRTUAL DISTANCE

       XVZ = ( SIGMAZ/AZ(STAB,1) )**(1.0 / BZ(STAB,1))

       IF ( XVZ .GE. 100.0 ) THEN
         XVZ= ( ( SIGMAZ - CZ(STAB,2) ) / AZ(STAB,2) )**
     +          ( 1.0/ BZ(STAB,2) )
         IF ( XVZ .GE. 1000.0 ) THEN
           XVZ= ( ( SIGMAZ - CZ(STAB,3) ) / AZ(STAB,3) )**
     +          ( 1.0  / BZ(STAB,3) )
         ENDIF
       ENDIF
```

Program Listing 34 Subroutine NRCSIG

```
 XEZ = XVZ + DSMTRI

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

 ENDIF
 IF (SIGMAZ .GT. SZLIM)  SIGMAZ = SZLIM

RETURN
END
```

Subroutine NRCSIG (cont'd)

GENERAL UTILITY SUBROUTINES

There are eight subroutines used in Version 2.0 of MESOI that are most properly described as general utility subroutines. Table 12 lists these subroutines along with their memory requirements and a brief description of their use. The subroutines are listed in the table and discussed in this section in alphabetical order.

CLEAN

Subroutine CLEAN is used to reduce the number of puffs that are being tracked by MESOI. It is called from MESOI whenever the number of puffs at the end of an hour exceeds twice the number of puffs released during the hour.

Puffs are deleted for one of two reasons. They are deleted if the active puff flag has been turned off because the puff has left the grid or the maximum concentration in the puff has decreased below the minimum concentration of interest. They may also be deleted if puffs from the same source are sufficiently close together that they may be combined without causing a significant gap between puffs (see Figure 8.).

When two puffs are to be combined, the active puff flag for the puff being deleted is turned off and the other puff is assigned source terms that are the sums of the source terms for the two puffs. The sigma y value and radius of the puff that will remain are increased to the square root of the sum of the squares of the individual puff's sigma y values and radii, and the puff that is to remain is assigned the larger of the individual puff sigma z's. The other puff characteristics assigned to the combined puff are averages of the characteristics of the puffs being combined.

After all puff combinations and deletions have been completed, CLEAN recomputes the number of puffs remaining in the active puff list.

The code for subroutine CLEAN is contained in Program Listing 35.

DIRSPO

DIRSPD is a utility module which takes the U and V components of the wind and computes an equivalent speed and direction. The subroutine operates by first determining which quadrant contains the wind vector. Direction in degrees from through north, is computed based on the ratio of U and V. Speed is computed using the basic Pythagorean relationship.

DIRSPD is used only by the TERRA module in determining relative angles of the wind to terrain features. There are more elegant ways to perform the necessary calculations within a program. However, the method used is easy to follow and understand.

TABLE 12. General Utility Subroutines

NAME	CODE	LOCAL DATA	COMMON DATA	USE
CLEAN	1000	88	28068	Deletes unneeded puffs, consolidates overlapping puffs.
DIRSPD	218			Converts wind U and V components to direction and speed.
DTCHAR	334	191		Converts integer dates and times to character stings for titles, etc.
FILOPN	228	357	64	Opens disk files required for MESOI execution.
JULIAN	58	52		Converts day, month and year to Julian date.
LOCATE	2174	961	64	Finds correct meteorological data records in data files.
MODEND	860	679	64	Permits model restart, closes disk files, handles file disposition.
SHIFT	139	8	25020	Moves meteorological data from current hour to previous hour variables.

Listing 36 contains the subroutine DIRSPD code.

DTCHAR

DTCHAR is a utility module to convert the date and time integers obtained from the data records into character strings. These strings are more easily handled for display by the plotting modules. In its present form, DTCHAR creates a 9 character date string (e.g. 22-APR-83) and a 7 character time string (e.g. 1230 LT). Other similar tasks could be added as needed.

The basic process of the module is to convert each digit of the various integers into its equivalent ASCII character or a substitute character string. The CHAR function of FORTRAN is used to make the direct integer to character conversion. For example, the ASCII representation of the digits 0 - 9 are obtained by CHAR (digit + 48). The characters and strings are then assembled into the final string using the concatenation operation (//).

Program listing 37 contains the DTCHAR code.

FILOPN

FILOPN is a utility module to open all the files required by the model with the exception of the plot output files. Use of the graphic files will probably be system dependent. There is no user interaction with this module.

FILOPN consists of a single loop which is executed once for each file to be opened. Currently, twelve files are opened for model use. This is controlled by the value of NFILES. The loop range is from 3 to NFILES+2 since the first two channels assigned are for terminal I/O. The unit numbers, names and status of the files are contained in arrays. FILOPN obtains the necessary information for opening files from the specified elements of the arrays. After each pass through the loop a message is sent to the terminal indicating the result of the OPEN operation.

Inability to open a file does not stop program execution. FILOPN displays an error message and continues. Errors are likely to occur later as other modules attempt to perform I/O operations to the unopened file.

Program listing 38 contains the FILOPN code.

JULIAN

This subroutine accepts integer values for the day, month and year and computes the Julian day. MESOI data files contain the date as a Julian value. However, when a user is specifying a date for simulation or release, it is easier to input a month, day and year. JULIAN is used to convert the user input date into a form which can be compared with the data date.

JULIAN uses an array DYBFOR to define the total number of days which have preceeded any given month. This value for a given month plus the number of the days within the month plus any leap year correction sums to the JULIAN day.

Program listing 39 contains the JULIAN code.

LOCATE

The LOCATE subroutine searches the meteorological data files for the correct records to start simulation. The model is designed to require a set of observed data for its initialization. After that initial input, forecast data can be used.

The calling program passes LOCATE the year, day, hour and minute of the data record to be located and a unit index to specify which file is to be searched. Initially, LOCATE is called by INIT to position the observed data file at the first record needed for simulation. The records are located by reading thru the file in a forward direction to match dates and time. When the correct record is located a message is printed giving the current position of the observed data file and the starting record of the forecast file.

If a matching record is not located, a message is printed indicating the lack of matching data and control is returned to the calling program. Checks are made during the read-through to determine when no more data is available (Julian date = 888) and to assure correct file structure and record order. These checks are minimal since it is assumed that the data file will be created and error checked by data maintenance programs.

If during the simulation, DATRD detects that the Julian date on the next record is 888, then LOCATE is called by MESOI to position the forecast file at the next record in the time sequence. This branch of LOCATE is accessed when the unit index passed to the subroutine is 4.

Program listing 40 contains the LOCATE code.

MODEND

MODEND is the model termination module. It is designed to handle the rewinding or closing of disk files after simulation has ended. First, the module asks the user about restarting the simulation. If restart is requested, the two meteorological data files are rewound and the subroutine returns to MESOI. The output disk files remain open at their current locations.

If execution is to be stopped, MODEND closes the meteorological data files and then queries the user about the disposition of the generated output and exposure files (MESOUT, EXPCUM, AXPADV, and BXPADV). The four options are displayed as follows:

```
SPECIFY DISPOSITIDN FOR OUTPUT AND EXPOSURE FILES:

     1 = KEEP
     2 = DELETE
     3 = PRINT
     4 = PRINT/DELETE
```

One of the four options must be selected; other responses are ignored. The files are then closed accordingly.

If the deposition flag (DPFLG) has been set, the disposition query is repeated regarding the depletion and deposition files (AXPADL, BXPADL, ADEPOS, BDEPOS, ADPADV, and BDPADV). These files are closed as specified by the user and the program stops. This is the normal termination point for the model.

Program listing 41 contains the MODEND code.

SHIFT

SHIFT is a simple utility module that is accessed at the end of the advection loop in MESOI before any further meteorological data is read. It saves the wind grid components, air temperature and data date and hour for later use in the temporal interpolations. The naming convention used is for double letter variables to represent the current values and single letter variables of the previous interval. For example, TT might be the air temperature at 9:15 and T the temperature at 9:00.

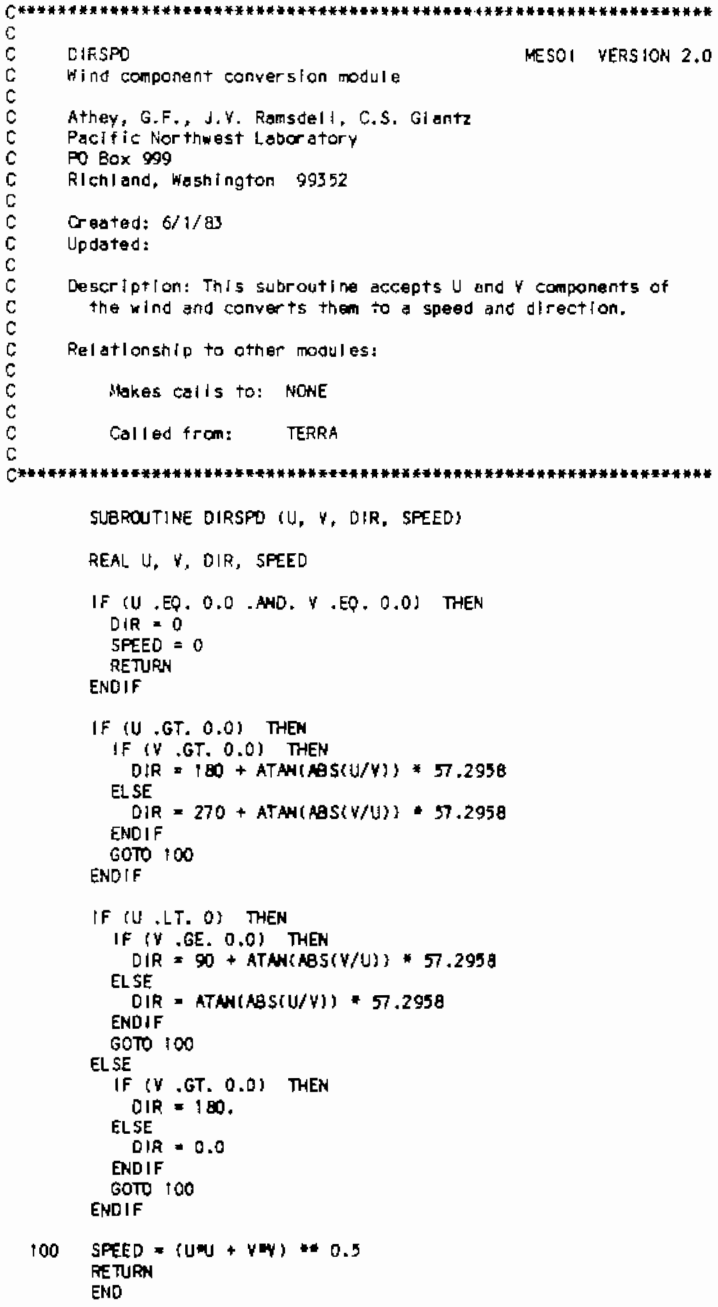
Program listing 42 contains the SHIFT code.

```
C*****************************************************************
C
C     CLEAN                                        MESOI  VERSION 2.0
C     Puff elimination and merge module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Called only after more than 2 * NPH puffs have been
C       released. All those puffs which have the flag MF set to zero
C       are dropped. Merges puffs of the same source whose separation
C       is less than the sum of their sigma y's.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C*****************************************************************

      SUBROUTINE CLEAN

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

C   **  MERGE LOOP; CHECK PUFFS STARTING WITH YOUNGEST

      I = TPUFFS
      J = TPUFFS - 1

C   **  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

C   **  CALCULATE SEPARATION; COMBINE PUFFS IF SEPARATION IS LESS THAN
C   **    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
          QP(I) = QP(I) + QP(J)
          ADQP(I) = ADQP(I) + ADQP(J)
          BDQP(I) = BDQP(I) + BDQP(J)

          XP(I)  = (XP(I)  + XP(J))  / 2.0
          YP(I)  = (YP(I)  + YP(J))  / 2.0
          ZP(I)  = (ZP(I)  + ZP(J))  / 2.0
          AGE(I) = (AGE(I) + AGE(J)) / 2.0
          DXS(I) = (DXS(I) + DXS(J)) / 2.0
          DYS(I) = (DYS(I) + DYS(J)) / 2.0

          SIGMAY(I) = SQRT (SIGMAY(I) ** 2 + SIGMAY(J) ** 2)
          SIGMAZ(I) = AMAX1(SIGMAZ(I), SIGMAZ(J))

          RADP(I) = SQRT (RADP(I) ** 2 + RADP(J) ** 2)
```

Program Listing 35 Subroutine CLEAN

```
        ENDIF
      ENDIF

  120 I = I - 1
      J = I - 1
      IF (I .GT. 1)  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)
      DXS(J) = DXS(I)
      DYS(J) = DYS(I)
      ADQP(J) = ADQP(I)
      BDQP(J) = BDQP(I)
      RADP(J) = RADP(I)
      AGE(J) = AGE(I)
      J = J + 1

  220 I = I + 1
      IF (I .LE. TPUFFS)  GOTO 200

      WRITE (LUN(2),240) TPUFFS-(J-1),J-1
  240 FORMAT (5X, 'CLEAN CALLED --> ', I4, ' PUFFS DROPPED',
     +  4X, I4, ' PUFFS REMAIN ON THE GRID',/)
      TPUFFS = J - 1

      RETURN
      END
```

Subroutine CLEAN (cont'd)



```
C*****************************************************************
C
C     DIRSPD                                       MESO1  VERSION 2.0
C     Wind component conversion module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: This subroutine accepts U and V components of
C       the wind and converts them to a speed and direction.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     TERRA
C
C*****************************************************************

      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
```

Program Listing 36 Subroutine DIRSPD

```
C*******************************************************************
C
C     DTCHAR                                        MESOI  VERSION 2.0
C     Date and time character string generator
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Sox 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to convert groups of integers
C       (e.g. year, month etc.) into more easily displayable
C       character strings.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     PLOTZ, WNDFLD, WNDPLT
C
C*******************************************************************

        SUBROUTINE DTCHAR (DYR, DDAY, DHR, DMIN, DDATE, DTIME)

        REAL LYFLG

        INTEGER DYR, DDAY, DHR, DMIN, FIRSTD(12), LASTD(12)

        CHARACTER*9 DDATE, DTIME
        CHARACTER*5 IMO(12)
        CHARACTER*1 I1, I2, I3, I4

        DATA FIRSTD /  1, 32,60, 91,121,152,182,213,244,274,305,335 /
        DATA LASTD  / 31, 59,90,120,151,181,212,243,273,304,334,365 /

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

C   **  CREATE DATE STRING

        LYFLG = FLOAT(DYR)/4 - DYR/4
        IF (LYFLG .EQ. 0)  THEN
          DO 100 I = 3, 12
            FIRSTD(I) = FIRSTD(I) + 1
  100     CONTINUE

          DO 110 I = 2, 12
            LASTD(I) = LASTD(I) + 1
  110     CONTINUE
        ENDIF

        DO 200 I = 1, 12
          IF (DDAY .GE. FIRSTD(I) .AND. DDAY .LE. LASTD(I))  GOTD 210
  200   CONTINUE

  210   IDAY = DDAY - (FIRSTD(I) - 1)
        I1 = CHAR(IDAY/10 + 48)
        I2 = CHAR(JMDD(IDAY,10) + 48)
        I3 = CHAR(DYR/10 + 48)
        I4 = CHAR(JMOD(DYR,10) + 48)
```

Program Listing 37 Subroutine DTCHAR

```
      DDATE = I1//I2//IMO(I)//I3//I4

C  ** CREATE TIME STRING

      I1 = CHAR(DHR/10 + 48)
      I2 = CHAR(JMOD(DHR,10) + 48)
      I3 = CHAR(DMIN/10 + 48)
      I4 = CHAR(JMOD(DMIN,10) + 48)

      DTIME = I1//I2//I3//I4//' LT'

      RETURN
      END
```

Subroutine DTCHAR (cont'd)

```
C************************************************************
C
C     FILOPN                                   MESOI  VERSION 2.0
C     File opening module
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Module to open all the disk files required for
C          model operation.
C
C     Relationship to other modules:
C
C        Makes calls to:  NONE
C
C        Called from:     MESOI
C
C************************************************************

       SUBROUTINE FILOPN

       INCLUDE 'UNITS.INC'

       CHARACTER*10 FILES(12)
       CHARACTER*3  FTYPE(12)

       DATA FILES / 'OBSDAT.DAT', 'FORDAT.DAT', 'MESOUT.DAT',
     +              'EXPCUM.DAT', 'AXPDPL.DAT', 'BXPDPL.DAT',
     +              'AXPADV.DAT', 'BXPADV.DAT', 'ADEPOS.DAT',
     +              'BDEPOS.DAT', 'ADPADV.DAT', 'BDPADV.DAT'/

       DATA FTYPE / 'OLD', 'OLD', 'NEW', 'NEW', 'NEW', 'NEW',
     +              'NEW', 'NEW', 'NEW', 'NEW', 'NEW', 'NEW'/

       NFILES = 12

       DO 200 N = 3, NFILES+2
         OPEN (UNIT=LUN(N), FORM='FORMATTED', STATUS=FTYPE(N-2),
     +        FILE=FILES(N-2), IOSTAT=IER)
         IF (IER .NE. 0)  THEN
           WRITE (LUN(2),110) IER, FILES(N-2), LUN(N)
  110      FORMAT (/5X, 'ERROR NUMBER ', I4, ' OPENING FILE ', A10,
     +             ' ON UNIT NUMBER ', I2)
         ELSE
           WRITE (LUN(2),120) FILES(N-2), LUN(N)
  120      FORMAT (/5X, 'FILE ', A10, ' OPENED ON UNIT NUMBER ', I2)
         ENDIF
  200  CONTINUE
       PAUSE 'TYPE C TO CONTINUE'
       RETURN
       END
```

Program Listing 38 Subroutine FILOPN

```
C******************************************************************
C
C     JULIAN                                       MESOI  VERSION 2.0
C     Date converter
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Accepts year, month and day; returns Julian date
C
C     Relationship to other modules:
C
C        Makes calls to:  NONE
C
C        Called from:      INIT, RELEAS
C
C******************************************************************

        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
```

Program Listing 39 Subroutine JULIAN

```
C*****************************************************************
C
C     LOCATE                                       MESOI  VERSION 2.0
C     Data record locator
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Program to search the data files for the required
C       starting records. The observation records are searched
C       first for day, then hour and record number. The starting
C       date and time of the forecast file is then reported.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI, INIT
C
C*****************************************************************

      SUBROUTINE LOCATE (SYR, SDAY, SHR, SMIN, LERROR, LUINDX, INTFLG)

      INCLUDE 'UNITS.INC'

      INTEGER SYR, SDAY, SHR, SMIN, IYR, DAY, HOUR, MIN, DREC,
     +        IDUM, LERROR

      IF (LUINDX .EQ. 4)  GOTO 200
      WRITE (LUN(2),100)
      WRITE (LUN(5),100)
  100 FORMAT (/5X, 'METEOROLOGICAL DATA FILE SEARCH -- ',/)

  110 READ (LUN(3),120,END=600,ERR=800) IYR, DAY, HOUR, MIN, DREC
  120 FORMAT (1X, I2, I3, I2, I2, 1X, I1)
      IF (DAY .EQ. 888)  GOTO 600
      IF (DREC .NE .1)  GOTO 820
C   **  SIMULATION IS REQUIRED TO START ON THE HOUR
      IF (MIN .NE. 0)  GOTO 660
      IF (SYR .LT. IYR .OR. SYR .GT. (IYR+1))  GOTO 620
      IF (DAY .GT. SDAY .AND. SDAY .GT. 2) GOTO 620
      IF (DAY .EQ. SDAY) GOTO 150
      READ (LUN(3),130) IDUM
      READ (LUN(3),130) IDUM
  130 FORMAT (A1)
      IF (INTFLG .EQ. 0)  GOTO 110

      DO 140 I = 1, 9
        READ (LUN(3),130) IDUM
  140 CONTINUE
      GOTO 110

C   **  LOCATE PROPER HOUR

  150 IF (HOUR .EQ. SHR)  GOTO 170
      READ (LUN(3),130) IDUM
      READ (LUN(3),130) IDUM
      IF (INTFLG .EQ. 0)  GOTO 110

      DO 160 I = 1, 9
```

Program Listing 40 Subroutine LOCATE

```
      READ (LUN(3),130) IDUM
160 CONTINUE
    GOTO 110

170 WRITE (LUN(2),180) IYR, DAY, HOUR, MIN, DREC
    WRITE (LUN(5),180) IYR, DAY, HOUR, MIN, DREC
180 FORMAT (7X, 'OBSV FILE POSITIONED AT:   JULIAN DAY = ',I2,1X,I3,
   +  3X,' TIME = ', I2, ':', I2, 3X,' RECORD ',I1/)
    BACKSPACE LUN(3)

    READ (LUN(4),120,END=640,ERR=840) IYR, DAY, HOUR, MIN, DREC
    BACKSPACE LUN(4)

    WRITE (LUN(2),190) IYR, DAY, HOUR, MIN, DREC
    WRITE (LUN(5),190) IYR, DAY, HOUR, MIN, DREC
190 FORMAT (7X, 'FORECAST FILE STARTS AT:   JULIAN DAY = ', I2,
   +  1X, I3, 3X, ' TIME = ', I2, ':', I2, 3X, ' RECORD ', I1/)
    LERROR = 0
    RETURN

200 WRITE (LUN(2),210)
    WRITE (LUN(5),210)
210 FORMAT (/5X, 'FORECAST DATA FILE SEARCH -- ',/)
220 READ (LUN(4),120,END=640,ERR=840)  IYR, DAY, HOUR, MIN, DREC
    IF (DREC .NE. 1)  GOTO 820
    IF (MIN .NE. 0)  GOTO 860
    IF (DAY .EQ. SDAY)  GOTO 240

    READ (LUN(4),130) IDUM
    READ (LUN(4),130) IDUM
    IF (INTFLG .EQ. 0)  GOTO 220
    DO 230 I = 1, 9
      READ (LUN(4),130) IDUM
230 CONTINUE
    GOTO  220

240 IF (HOUR .EQ. SHR)  GOTO 260
    READ (LUN(4),130) IDUM
    READ (LUN(4),130) IDUM
    IF (INTFLG .EQ. 0)  GOTO 220

    DO 250 I = 1, 9
      READ (LUN(4),130) IDUM
250 CONTINUE
    GOTO 220

260 IF (MIN .EQ. SMIN .OR. INTFLG .EQ. 0)  GOTO 270
    READ (LUN(4),130) IDUM
    READ (LUN(4),130) IDUM
    READ (LUN(4),120,END=640,ERR=840) IYR, DAY, HOUR, MIN, DRRC
    GOTO 260

270 DO 280 I = 1, 2
      READ (LUN(4),130) IDUM
280 CONTINUE
    READ (LUN(4),120,END=640,ERR=840) IYR, DAY, HOUR, MIN, DREC

    WRITE (LUN(2),290) IYR, DAY, HOUR, MIN, DREC
    WRITE (LUN(5),290) IYR, DAY, HOUR, MIN, DREC
290 FORMAT (3X, 'FORECAST DATA FILE POSITIONED AT:  JDAY = ',I2,
   +  1X, I3, 3X, ' TIME = ' ,I2, ':', I2, 3X, ' RECORD ', I1/)
    BACKSPACE (LUN(4))
    LERROR = 0
    RETURN

600 WRITE (LUN(2),610) SYR, SDAY, SHR
```

Subroutine LOCATE (cont'd)

```
    WRITE (LUN(5),610) SYR, SDAY, SHR
610 FORMAT (/5X, 'LOCATE >> EOF ENCOUNTERED IN OBSERVATION DATA FILE',
   +  ' DURING SEARCH', /7X, 'FOR DAY ', I2, 1X, I3, ' HOUR ', I2,
   +  /7X, 'SIMULATION WITH SPECIFIED DATE AND TIME IS IMPOSSIBLE',//)
    LERROR = 1
    REWIND (UNIT=LUN(3))
    REWIND (UNIT=LUN(4))
    RETURN

620 WRITE (LUN(2),630) IYR, DAY, SYR, SDAY
    WRITE (LUN(5),630) IYR, DAY, SYR, SDAY
630 FORMAT (/5X, 'DATA FILE STARTS ON DAY ', I2, 1X, I3, 5X,
   +  'CANNOT SIMULATE DAY ', I2, 1X, I3//)
    LERROR = 1
    REWIND (UNIT=LUN(3))
    REWIND (UNIT=LUN(4))
    RETURN

640 STOP ' ERROR 1 (IN LOCATE)  -- EOF IN FORECAST FILE'

660 STOP ' ERROR 2 (IN LOCATE)  -- UNRECOGNIZED DATA SEQUENCE (OBSV)'

800 STOP ' ERROR 3 (IN LOCATE)  -- READ ERROR IN OBS FILE'

820 STOP ' ERROR 4 (IN LOCATE)  -- INCOMPLETE DATA RECORD'

840 STOP ' ERROR 5 (IN LOCATE)  -- READ ERROR IN FCST FILE'

860 STOP ' ERROR 6 (IN LOCATE)  -- UNRECOGNIZED DATA SEQUENCE (FCST)'

    END
```

Subroutine LOCATE (cont'd)

```
C***********************************************************************
C
C     MODEND                                          MESOI  VERSION 2.0
C     Simulation termination handler
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Subroutine to handle the restart of the model and
C       the disposition of the disk files.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI
C
C***********************************************************************

      SUBROUTINE MODEND (*, *, DPFLG)

      INCLUDE 'UNITS.INC'

      INTEGER IDISP, DPFLG

      CHARACTER*12 DISPO(4)
      CHARACTER*1  SELECT

      DATA DISPO /'KEEP        ',
     +            'DELETE      ',
     +            'PRINT       ',
     +            'PRINT/DELETE'/

  100 WRITE (LUN(2),110)
  110 FORMAT (/5X, 'DO YOU WISH TO START ANOTHER SIMULATION? Y OR N')
      READ (LUN(1),120,END=100,ERR=100) SELECT
  120 FORMAT (A1)

      IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  THEN
        REWIND LUN(3)
        REWIND LUN(4)
        WRITE (LUN(5),130)
  130   FORMAT (/5X, '** RESTART REQUESTED **'/)

  140   WRITE (LUN(2),150)
  150   FORMAT (/5X, 'DO YOU WISH TO REINITIALIZE THE GRID? Y OR N')
        READ (LUN(1),120,END=140,ERR=140)  SELECT
        IF (SELECT .EQ. 'Y' .OR. SELECT .EQ. 'y')  RETURN 1
        WRITE (LUN(5),160)
  160   FORMAT (/5X, '** NO GRID REINITIALIZATION **'/)
        RETURN 2
      ENDIF

      CLOSE (LUN(3))
      CLOSE (LUN(4))
      PRINT *, ' METEOROLOGICAL DATA FILES CLOSED'

  200 WRITE (LUN(2),210)
  210 FORMAT (/5X, 'SPECIFY DISPOSITION FOR OUTPUT AND EXPOSURE ',
     +      ' FILES:', /7X, '1 = KEEP',  /7X, '2 = DELETE',
```

Program Listing 41 Subroutine MODEND

```
   +                 /7X, '3 = PRINT', /7X, '4 = PRINT/DELETE'/)
      READ (LUN(1),*) IDISP
      IF (IDISP .LT. 1 .OR. IDISP .GT. 4)  GOTO 200

      CLOSE (LUN(5),  STATUS = DISPO(IDISP))
      CLOSE (LUN(6),  STATUS = DISPO(IDISP))
      CLOSE (LUN(9),  STATUS = DISPO(IDISP))
      CLOSE (LUN(10), STATUS = DISPO(IDISP))
      PRINT *, 'OUTPUT AND EXPOSURE FILES CLOSED'

      IF (DPFLG .EQ. 0)  THEN
        IDISP = 2
      ELSE
220     WRITE (LUN(2),230)
230     FORMAT (/5X, 'SPECIFY DISPOSITION FOR DEPLETED AND DEPOSITION',
   +       ' FILES:', /7X, '1 = KEEP',  /7X, '2 = DELETE',
   +                 /7X, '3 = PRINT', /7X, '4 = PRINT/DELETE'/)
        READ (LUN(1),*) IDISP
        IF (IDISP .LT. 1 .OR. IDISP .GT. 4)  GOTO 220
      ENDIF

      CLOSE (LUN(7),  STATUS = DISPO(IDISP))
      CLOSE (LUN(8),  STATUS = DISPO(IDISP))
      CLOSE (LUN(11), STATUS = DISPO(IDISP))
      CLOSE (LUN(12), STATUS = DISPO(IDISP))
      CLOSE (LUN(13), STATUS = DISPO(IDISP))
      CLOSE (LUN(14), STATUS = DISPO(IDISP))

      PRINT *, 'DEPLETION AND DEPOSITION FILES CLOSED'
      STOP 'END MESOI EXECUTION'
      END
```

Subroutine MODEND (cont'd)

```
C*****************************************************************
C
C     SHIFT                                          MESOI  VERSION 2.0
C     Save wind field, times and temperature
C
C     Athey, G.F., J.V. Ramsdell, C.S. Glantz
C     Pacific Northwest Laboratory
C     PO Box 999
C     Richland, Washington  99352
C
C     Created: 6/1/83
C     Updated:
C
C     Description: Module to save parameters from the previous
C           hour. Called at the end of each simulation hour.
C
C     Relationship to other modules:
C
C         Makes calls to:  NONE
C
C         Called from:     MESOI, INIT
C
C*****************************************************************

      SUBROUTINE SHIFT

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

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

      PREVDY = DDAY
      PREVHR = DHR
          UG = UUG
          VG = VVG
           T = TT

      RETURN
      END
```

Program Listing 42 Subroutine SHIFT

COMMON BLOCKS

MESOI Version 2.0 makes extensive use of labelled COMMON blocks for passing data between program elements. There are 12 of these blocks. They are listed in Table 13 along with their memory requirements and a brief description of their contents. More than half of the total MESOI memory requirement is taken for storage of the data in the MATRIX, PUFFS, and WINDS blocks.

Each COMMON block is defined in an INCLUDE file that is referenced in the program elements where the block is needed. In addition to the COMMON statements, the INCLUDE files contain INTEGER, REAL and DATA statements as appropriate.

Program Listing 43 contains the INCLUDE files. If MESOI is to be run on a computer system with a FORTRAN 77 implementation that does not support the use of INCLUDE files, the COMMON, INTEGER, REAL and DATA statements in the INCLUDE files must be inserted in each program element where the INCLUDE file is referenced.

TABLE 13. Common Blocks

COMMON BLOCK	SIZE (BYTES)	CONTENTS
ARRIV	1560	Information related to the status of time-integrated concentrations at selected check points.
CHAR	187	Character variables including the simulation title, wind station names, and the date and time that the simulation was run.
CONST	324	Constants used in computations including the deposition velocity and washout coefficients.
DATIM	188	Information related to the simulated time and current meteorological conditions other than wind.
DECAY	760	Variables related to decay and ingrowth.
MATRIX	34596	Nine arrays for accumulation of time-integrated concentrations and surface contamination at grid nodes.
PUFFS	28004	Information related to individual puffs including original source indentification, position, source strength, diffusion coefficient values, radius, status and age.
REL	224	Information related to releases, source position, strength, effulent flow rate and temperature, time of release, duration of release (in number of puffs).
STATN	484	Information on position and status of wind observation stations.
TOPOGR	384B	Topography for use in diffusion computations.
UNITS	64	Logical unit assignments for model input and output.
WINDS	24832	Wind data, distances from grid nodes to closest wind observation points.

```
C*******************************************************************
C
C     ARRIV.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Checkpoint array, threshold values and flag
C
C     Included in modules: MESOI, ARRIVN, ARRTIM, EXPSUM
C
C*******************************************************************

        REAL ARRIVL(11,35), THRSH1, THRSH2

        INTEGER ARRFL, EXPFLG, NCP

        COMMON /ARRIV/ ARRIVL, THRSH1, THRSH2, ARRFL, EXPFLG, NCP
```

```
C*******************************************************************
C
C     CHAR.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Character variable include file
C
C     Included in modules: MESOI, DIFDEP, GRIDIN, INIT,   PLMRIZ,
C                          PLOTZ, PRINTE, REARNG, SCREEN, STRAY,
C                          TERRA, WIND,   WNDFLD, WNDFLT
C
C*******************************************************************

        CHARACTER*1 TITLE(50)
        CHARACTER*4 NAMST(30)
        CHARACTER*8 RTIME
        CHARACTER*9 RDATE

        COMMON /CHAR/ TITLE, NAMST, RDATE, RTIME
```

```
C*******************************************************************
C
C     CONST.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Constant include file
C
C     Included in modules: MESOI, ARRTIM, DIFDEP, INIT, PUFFM,
C                          PUFFR, RELEAS, WINSPD
C
C*******************************************************************

        REAL CHIMIN, ACNST, PAC, ADT, DELXY, DV, TWOPI, WC(6),
     +      XS(31), YS(31), RCH, ALPHAU, ALPHAV

        INTEGER NPH, ALPHFL, INTFLG

        COMMON /CONST/ CHIMIN, ACNST, PAC, ADT, DELXY, DV, TWOPI, WC,
     +                 XS, YS, RCH, ALPHAU, ALPHAV, NPH,
     +                 ALPHFL, INTFLG
```

Program Listing 43 Include Files

```
C***********************************************************************
C
C     DATIM.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Data, date and time include file
C
C     Included in modules: MESOI, ARRTIM, DATRD,  DATWR, DIFDEP,
C                          INIT,  PLMRIZ, PLOTZ,  PUFFM, SCREEN,
C                          SHIFT, WIND,   WNDFLD, WNDPLT
C
C***********************************************************************

        REAL CONV

        INTEGER DYR, DDAY, DHR, DMIN, DREC, STAB, LDEPTH, PRECIP,
     +     UPWIND, DATA(30), TEMPA, TEMPB, T, TT, PREVDY, PREVHR,
     +     NXTDAY

        COMMON /DATIM/ CONV, DYR, DDAY, DHR, DMIN, DREC, STAB,
     +                 LDEPTH, PRECIP, UPWIND, DATA, TEMPA, TEMPB,
     +                 T, TT, PREVDY, PREVHR, NXTDAY

        DATA CONV / 1.0 /



C***********************************************************************
C
C     DECAY.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Radioactive decay variable include file
C
C     Included in modules: MESOI, DIFDEP, RDK, RDKIN, RELEAS
C
C***********************************************************************

        REAL ADK(60), BDK(60), BDIN(60), ADECA1, ADECA2, BDECA1,
     +       BDECA2, BINAS, BINSI, ALMBDA, BLMBDA, DKCOEF

        INTEGER RDKFL

        COMMON /DECAY/ ADK, BDK, BDIN, ADECA1, ADECA2, BDECA1, BDECA2,
     +                 BINAS, BINSI, ALMBDA, BLMBDA, DKCOEF, RDKFL


C***********************************************************************
C
C     MATRIX.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Grid matrix array include file
C
C     Included in modules: MESOI,  ARRTIM, DIFDEP, EXPSUM, INIT,
C                          PRINTE, SCREEN
C
C***********************************************************************

        REAL EXPCUM(31,31), AXPDPL(31,31), BXPDPL(31,31),
     +       ADEPOS(31,31), BDEPOS(31,31), AXPADV(31,31),
     +       BXPADV(31,31), ADPADV(31,31), BDPADV(31,31)

        COMMON / MATRIX / EXPCUM, AXPDPL, BXPDPL, AXPADV, BXPADV,
     +                    ADEPOS, BDEPOS, ADPADV, BDPADV
```

Include Files (cont'd)

```
C******************************************************************
C
C     PUFFS.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Puff parameter include file
C
C     Included in modules: MESOI, CLEAN, DIFDEP, INIT, PLOTZ,
C                          PUFFM, PUFFR, TESTM
C
C******************************************************************

       REAL XP(500), YP(500), ZP(500), QP(500), DXS(500), DYS(500),
     +    ADQP(500), BDQP(500), SIGMAY(500), SIGMAZ(500), RADP(500)

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

       COMMON / PUFFS / XP, YP, ZP, QP, DXS, DYS, ADQP, BDQP, SIGMAY,
     +                  SIGMAZ, RADP, TPUFFS, MF, AGE, SP


C******************************************************************
C
C     REL.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Release parameter include file
C
C     Included in modules: MESOI,  INIT,   PLMRIZ, PLOTZ, PUFFR,
C                          RELEAS, SCREEN, WINSPD
C
C******************************************************************

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

       INTEGER NSOURC, NPUFFS(4), RELDAY(4), RELHR(4), PARTHR(4),
     +         MAXPUF(4), PLMRFL(4), DPFLG, PUFLG(4)

       COMMON /REL/ XSOURC, YSOURC, ZSOURC, QSOURC, QH, STEMPC, EFFLUX
     +              NSOURC, NPUFFS, RELDAY, RELHR,  PARTHR,  MAXPUF,
     +              PLMRFL, DPFLG,  PUFLG


C******************************************************************
C
C     STATN.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Station parameter include file
C
C     Included in modules: GRIDIN, REARNG, STRAY, WIND, WNDFLD,
C                          WNDPLT
C
C******************************************************************

       REAL XSTA(30), YSTA(30)

       INTEGER ACTSTA, STATUS(30), ELEV(30)

       COMMON /STATN/ XSTA, YSTA, ACTSTA, STATUS, ELEV
```

Include Files (cont'd)

```
C*********************************************************************
C
C     TOPOGR.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Topography data include file
C
C     Included in modules: MESOI, DIFDEP, DTOPO, TOPFIL
C
C*********************************************************************

        INTEGER TOPO(31,31), TOPFLG

        COMMON /TOPOGR/ TOPO, TOPFLG
```

```
C*********************************************************************
C
C     UNITS.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Logical unit assignment include file
C
C     Included in modules: MESOI,  ARRIVN, ARRTIM, CLEAN,  DATRD,
C                          DATWR,  EXPSUM, FILOPN, GRIDIN, INIT,
C                          LOCATE, MODEND, PLMRIZ, PLOTZ,  PRINTE,
C                          RDKIN,  RELEAS, SCREEN, TERRA,  TESTM,
C                          TOPFIL, WIND,   WNDFLD, WNDPLT
C
C*********************************************************************

        INTEGER LUN(16)

        COMMON / UNITS / LUN

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

```
C*********************************************************************
C
C     WINDS.INC
C
C     Created: 6/1/83
C     Updated:
C
C     Description:  Wind grid include file
C
C     Included in modules: MESOI, INIT, PUFFM,  SHIFT,  STRAY,
C                          TERRA, WIND, WINSPD, WNDFLD, WNDPLT
C
C*********************************************************************

        REAL STDIST(10,16,16), U(16,16), V(16,16), UG, VG, UU(16,16),
     +       VV(16,16), UUG, VVG

        INTEGER STNUM(10,16,16), SPD(30), DIR(30), TERFLG

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

Include Files (cont'd)

MESOI CUSTOMIZATION

MESOI has been developed to meet the needs of the U.S. Department of Energy related to its operations at Hanford, Washington and to meet the needs of the U.S. Nuclear Regulatory Commission Intermediate Dose Assessment System. Both of these organizations have requirements for model features and flexibility that may be beyond the requirements of other users, and they have access to relatively large minicomputer systems. As a result, it is likely that other potential users of MESOI may want to customize it to their specific needs and systems. This section outlines areas of the program that should be considered in model customization. These areas include 1) computer system related features, 2) data files, 3) model control parameters, 4) model computational features, 5) model output, and 6) run time initialization.

SYSTEM RELATED FEATURES

In preparing the MESOI Version 2.0 computer code, care has been taken to use ANSI Standard FORTRAN 77 (American National Standards Institute 1978) and limit the number of extensions to the standard that are used. As a result, the code should be highly transportable between computers. Most of the code should also be compatible with extended versions of FORTRAN 66 (FORTRAN IV). Three areas where program modifications may be necessary are: 1) the use of INCLUDE files to supply specification statements to the various program elements where they are needed, 2) the use of a COMMON block to make logical unit assignments to input and output devices, and 3) the use of the block IF-THEN-ELSE structure. Of these areas only the use of INCLUDE files is an extension to the 1977 standard, and the modifications needed to eliminate any problems should be straightforward.

Three other areas where the user's system might require modifications to MESOI are in the terminal control sequence, the plot commands used in the output routines, and the sequences and commands used to open and close files.

DATA FILES

MESOI Version 2.0 makes extensive use of data files. Model execution requires that many of these files exist and be immediately available to the model. The files that must be available include meteorological data files (both observed and forecast) and a file containing the locations of the sources of wind data to be used. The meterological data file formats must conform to the formats used to read in data in subroutines LOCATE and DATRD. The proper factor to convert the wind speeds from the units in which they are reported to meters/second must be included in the DATIM common block. The format for the wind observation station file must match the format in the input statement in subroutine GRIDIN.

If the model is to be fully customized, two topographic files and a check point file should be prepared and available at model execution time. The model can be run without these files if they are not available, but it will not make use all of its features.

One of the topographic files contains the elevations at each of the nodes of the grid used to accumulate time-integrated concentrations, and the other contains the information needed to adjust the winds at locations where the topography is expected to affect the wind. Preparation of the first of the files is straightforward, preparation of other requires a good deal of meteorological insight.

The checkpoint file specifies locations of receptors of specific interest. These receptors might include institutions where evacuation could be a problem such as hospitals and jails; they might include population centers, key transportation centers, and significant agricultural or water resource areas. The format for the checkpoint file is specified in subroutine ARRIVN.

MODEL CONTROL PARAMETERS

A number of model control parameters for MESOI are set in the code without user interaction. These parameters include the default grid spacing, the number of puffs to be released per hour, the maximum number of sources, the maximum duration of the period that can be simulated in a single model run, the maximum number of puffs that can be followed, and the minimum concentration of interest. The user may wish to modify any or all of these parameters permanently, or to alter the code so they may be changed interactively on a run-by-run basis.

The default grid spacing is set in subroutine GRIDIN as 5000 meters. This spacing applies specifically to the grid used for the surface wind field. The spacing of the grid used for accumulation of the time-integrated concentrations and surface concentrations is one half of the wind grid spacing, i.e. 2500 meters. An opportunity to change the wind grid spacing is presented during model initialization. If the spacing is changed, separate topographic files should be provided for the new spacing.

The number of puffs that are released each hour is set at the beginning of subroutine INIT. In the code as published, four puffs will be released each hour, unless the release duration is less than one hour. The acceptable release rates are the integer factors of 60. If the puff release rate is changed, the DATA statements that define the acceptable sampling intervals and number of sampling intervals per advection period must also be changed. The acceptable values for these variables are the integer cofactors of the advection period duration when expressed in minutes.

MESOI Version 2.0 will accommodate releases from four release points without modification. There is no fundamental reason that the limit could not be increased. If a higher limit is desired it can be set by changing the upper limit in a logical IF statement that occurs at the beginning of subroutine RELEAS. The text string in FORMAT statement 140 should also be changed to reflect the new range for number of release points. Similarly, the dimensions for variables containing release information in subroutine RELEAS and the REL.INC file should be increased to the new maximum.

The default limit on the duration of any simulation is set by the upper limit of the loop that starts with DO 400 in MESOI. This limit may be increased or decreased without affecting any of the model's computations.

Puff release rate, the number of sources and the duration of the simulation determine the maximum number of puffs that can be tracked at anytime by MESOI is determined by the dimensions for the variables that store the puff attributes. These variables are found in the PUFFS.INC file. The current dimension is 500, which is not sufficient to accommodate all of the puffs that could be released by MESOI in a 48 hour simulation. As a practical matter, however, the dimension is more than adequate because puffs are combined as they overlap and are deleted as their concentration decreases or they move off the grid. If the puff release rate, maximum number of sources or maximum simulation duration is changed, consideration should be given to revising the dimensions in the PUFFS.INC file.

The minimum time-integrated concentration of interest, in mass-hours per cubic meter, is set in the program line just after the puff release rate is set. The current value is 10^{-15}. The division by 60 is necessary because the sampling interval is expressed in minutes and is not a constant.

MODEL COMPUTATIONAL PROCEDURES

Unless a major modification of MESOI Version 2.0 is being undertaken in addition to customization, major changes should not be required in the MESOI computational procedures. A change in the computational procedures for determining the diffusion coefficients can be made by substituting a different SIGMA subroutine. The SIGMA subroutine used in the test case examples in Appendix B is the one identified as NRCSIG. A different subroutine may also be substituted for ALPHA if linear vertical interpolation of the horizontal winds is not acceptable.

The user may wish to change the initial values assigned to the diffusion coefficients. At present, sigma y and sigma z are assigned initial values of 1 m and 0.1 m, respectively. This assignment is made in PUFFR. Along the same line, the user may wish to change the deposition velocity or the washout coefficients. These parameters are set in subroutine INIT. The deposition velocity must be in meters per hour, and the washout coefficients must be in hours^{-1}. Finally, unless otherwise instructed, MESOI assumes that 1 mass unit of effluent is released from each source each hour. This default release rate is set in subroutine RELEAS.

MESOI INITIALIZATION

The interactive initialization of MESOI is a compromise between the conflicting goals of minimum model execution time and the flexibility required to handle likely release scenarios. When the model is customized for other users or specific applications, the interactive initialization portion of MESOI should be re-evaluated. If variables, such as the potential release locations, can be defined prior to model execution, those portions of the initialization that are used to define the variables can be eliminated. By eliminating unnecessary interactive initialization, the time required to obtain model output and the chances for incorrect initialization will be decreased. It may also be appropriate to add to the interactive initialization if there are conditions in specific scenarios that are subject to change that are assumed to constant in the current version of MESOI.

MESOI OUTPUT

Modification of MESOI output is one of the stages of model customization. As a minimum, this modification should include changes in the text associated with the inquiries made during the interactive model initialization and with the output displays, graphics product and printed files. Specific locations where the changes are needed are found through-out the code.

REFERENCES

American Meteorological Society. 1978. "Accuracy of Dispersion Models - A Position Paper of the AMS 1977 Committee on Atmospheric Turbulence and Diffusion." Bull. Am. Meteorol. Soc. 59:1025-1026.

American National Standards Institute. 1978. American National Standard FORTRAN, X3.9-1978 (FORTRAN 77). New York, NY.

Athey, G.F. and J.V. Ramsdell. 1982. User's Guide to the MESOI Diffusion Model -- Version 1.1 (For Data General Eclipse S/230 with AFOS). PNL-4369, Pacific Northwest Laboratory, Richland, Washington.

Bander, T.J. 1982. PAVAN: An Atmospheric Dispersion Program for Evaluating Design Basis Accidental Releases of Radioactive Materials from Nuclear Power Stations. NUREG/CR-2858 (PNL-4413), U.S. Nuclear Regulatory Commission, Washington, D.C.

Briggs, G.A. 1969. Plume Rise, USAEC Critical Review Series, TID-25075.

Briggs, G.A. 1973. Diffusion Estimation for Small Emissions. ATDL Contribution File No. 79. Atmospheric Turbulence and Diffusion Laboratory, NOAA, P.O. Box E, Oak Ridge, TN.

Csanady, G.T. 1973. Turbulent Diffusion in the Environment. D. Reidel, Boston, 295 p.

Eimutis, E.C. and M.G. Konicek. 1972. "Derivations of Continuous Functions for the Lateral and Vertical Atmospheric Dispersion Coefficients." Atmos. Environ., 6:859-863.

Engelmann, R.J. 1968. "The Calculation of Precipitation Scavenging, in Meteorology and Atmoic Energy - 1968. TID-24190, U.S. Atomic Energy Commission. pp.202-221.

Gifford, F.A. 1961. "Use of Routine Meterological Observations for Estimating Atmospheric Dispersion." Nucl. Safety, 2:47-51.

Gifford, F.A. 1968. "An Outline of Theories of Diffusion in the Lower Layers of the Atmosphere." in Meteorology and Atomic Energy - 1968. TID-24190, U.S. Atomic Energy Commission. pp. 65-116.

Gifford, F.A. 1975. "Atmospheric Dispersion Models for Environmental Pollution Applications." in Lectures on Air Pollution and Environmental Impact Analyses. American Meteorological Society, Boston, MA.

Gifford, F.A. 1976. "Turbulent Diffusion Typing Schemes - a Review." Nucl. Safety, 17:68-86.

Goodin, W.R., G.J. McRae and J.H. Seinfeld. 1979. "A Comparison of Interpolation Methods for Sparse Data: Application to Wind and Concentration Fields." J. Appl. Meteor., 18:761-771.

Hanna, S.R., G.A. Briggs, J. Deardorff, G.A. Egan, F.A. Gifford and F. Pasquill. 1977. "AMS Workshop on Stability Classification Schemes and Sigma Curves -- Summary of Classification Recommendations." Bull. Am. Meteorol. Soc., 58:1305-1309.

Hanna, S.R., G.A. Briggs and R.P. Hosker, Jr. 1982. Handbook on Atmospheric DIffusion. DOE/TIC-11223, U.S. Department of Energy, Washington, D.C.

Horst, T.W. 1977. "A Surface Depletion Model for Deposition from a Gaussian Plume." Atmos. Environ., 14:41-46.

Irwin, J.A. 1983. "Estimating Plume Dispersion -- A Comparison of Several Sigma Schemes." J. Climate and Appl. Meteor., 22:92-114.

Lague, J.S., E.M. Irvine and T.F. Lavery. 1980. "Validation of Alternate Turbulence Typing Models for Estimating Air Quality Effects of Coastal Sources Near Rough Terrain." presented at Second Joint Conference on Applications of Air Pollution Meteorology, and Second Conference on Industrial Meteorology, New Orleans, LA, American Meterological Society, Boston, MA.

Lalas, D.P., V. Catsoulis and M. Petrakis. 1979. "On the Consistency of Stability Classification Schemes When Applied to Non-homogeneous Terrain." Atmos. Environ., 13:687-691.

Luna, R.E. and H.W. Church. 1972. "A Comparison of Turbulence Intensity and Stability Ratio Measurements to Pasquill Stability Classes." J. Appl. Meterol., 11:663-339.

Martin, D.O. and J.A. Tikvart. 1968. "A General Atmospheric Diffusion Model for Estimating the Effects on Air Quality of One or More Source." presented at 61st Annual Meeting of the Air Pollution Control Association for NAPCA, St.Paul, MN, 18p.

Mitchell, A.E., Jr. 1982. "A Comparison of Short-Term Dispersion Estimates Resulting from Various Atmospheric Stability Classification Methods." Atmos. Environ., 16:765-773.

Pasquill, F. 1961. "The Estimation of the Dispersion of Windborne Material" Meteorol. Mag., 90:33-49.

Pasquill, F. 1974. Atmospheric Diffusion (2nd Ed.). Halstead Press, New York, 429 p.

Pasquill, F. 1979. "Atmospheric Dispersion Modeling." J. Air Poll. Control Assoc., 29:117-119.

Powell, D.C., H.L. Wegley and T.D. Fox. 1979. MESODIF-II: A Variable Trajectory Plume Segment Model to Assess Ground-Level Air Concentrations and Deposition of Effluent Releases from Nuclear Power Facilities. NUREG/CR-0523 (PNL-2419), U.S. Nuclear Regulatory Commission, Washington, D.C.

Ramsdell, J.V. and G.F. Athey. 1981. MESOI: An Interactive Lagrangian Trajectory Puff Diffusion Model. PNL-3998, Pacific Northwest Laboratory, Richland, Washington. 86p.

Ramsdell, J.V., S.R. Hanna and H.E. Cramer. 1982. Turbulent Diffusion Coefficients Within and Above the Urban Domain. final report to the U.S. Army Atmospheric Sciences Laboratory under Battelle, Pacific Northwest Laboratory contract 23111 05480, 95p.

Sagendorf, J.F., J.T.. Goll and W.F. Sandusky. 1982. XOQDOQ: Computer Program for the Meteorological Evaluation of Routine Effluent Releases at Nuclear Power Stations. NUREG/CR-2919 (PNL-4380), U.S. Nuclear Regulatory Commission, Washington, D.C.

Sedefian, L., and E. Bennett. 1980. "A Comparison of Turbulence Classification Schemes." Atmos. Environ., 14:741-750.

Sehmel, G.A. 1980. "Particle and Gas Dry Deposition: A Review." Atmos. Environ., 14:983-1011.

Skaggs, D.L. and E. Robinson. 1976. "A Comparison of Methods for Estimating Atmospheric Stability and Diffusion Coefficients." J. Air Poll. Control Assoc., 26:888-891.

Slinn, W.G.N. 1978. "Parameterizations for Resuspension and for Wet and Dry Deposition of Particles and Gases for Use in Radiation Dose Calculations." Nucl. Safety, 19:205-219.

Start, G.E. and L.L. Wendell. 1974. Regional Effluent Dispersion Calculations Considering Spatial and Temporal Meterological Variations. NOAA Tech. Memo. ERL ARL-44, 63 p.

Tadmor, J. and Y. Gur. 1969. "Analytical Expressions for Vertical and Lateral Dispersion Coefficients in Atmospheric Diffusion." Atmos. Environ., 3:688-689.

Turner, D.B. 1964. "A Diffusion Model for an Urban Area." J. Appl. Meteor., 3:83-91.

U.S. Nuclear Regulatory Commission. 1974. Onsite Meteorological Programs. Regulatory Guide 1.23 (formerly Safety Guide 23). Washington, D.C.

U.S. Nuclear Regulatory Commission. 1982. *Atmospheric Dispersion Models for Potential Accident Consequence Assessments at Nuclear Power Plants*. Regulatory Guide 1.145, Revision 1. Washington, D.C.

Van der Hoven, I. 1968. "Deposition of Particles and Gases." in *Meteorology and Atomic Energy - 1968*. TID-25190, U.S. Atomic Energy Commission pp. 202-208.

Weil, J.C. 1979. "Applicability of Stability Classification Schemes and Associated Parameters to Dispersion of Tall Stack Plumes in Maryland." *Atmos. Environ.*, 13:819-831.

Wendell, L.L. 1972. "Mesoscale Wind Fields and Transport Estimated Determined from a Network of Wind Towers," *Mon. Wea. Rev.*, 100:565-578.

Yanskey, G.R., E.H. Markee and A.P. Richter. 1966. *Climatography of the National Reactor Testing Station*. IDO-12048.

APPENDIX A

APPENDIX A

MESOI VERSION 2.0 GLOBAL VARIABLES

Name	Type	Where Defined	Where Used	How Passed	Contents
ACNST	Real	INIT	PUFFR	CONST	Constants in the Gaussian puff model
ACTSTA	Int	GRIDIN	WIND, WNDFLD, WNDPLT	STATN, Call	Number of active wind stations
ADECA1	Real	RDKIN, RELEAS	MESOI	DECAY	Fraction of A remaining after decaying for one advection period
ADECA2	Real	RDK	DIFDEP	DECAY	Fraction of A remaining at the end of a sampling interva
ADEDOS	Real	MESOI, INIT	PRINTE, SCREEN	MATRIX	Surface concentration of A (mass/m^2) accumulated prior to current advection period-corrected for decay
ADK	Real	RKD	DIFDEP	DECAY	Fraction of A remaining at the end of an advection period gives the current sampling interval
ADPADV	Real	MESOI, DIFDEP, INIT	PRINTE, SCREEN	MATRIX	Amount of A deposited in the current advection period (mass/m^2)
ADQP	Real	CLEAN, DIFDEP, PUFFR	TESTM	PUFFS	Source term for A corrected for depletion
ADT	Real	INIT	MESOI, PUFFM, RDKIN, RELEAS	CONST, Call	Duration of the advection period in hours
AGE	Int	CLEAN, DIFDEP PUFFR	TESTM	PUFFS	Time since puff release in minutes
AIRT	Real	MESOI	PLMRIZ	Call	Air temperature (°C) at time puff release
ALMBDA	Real	RDKIN, RELEAS	RDK	DECAY	Decay coefficient for A (primary effluent) (min^{-1})
ALPHAU	Real	ALPHA, INIT	PUFFM	CONST, Call	Exponent for power-law interpolation of E-W wind component
ALPHAV	Real	ALPHA, INIT	PUFFM	CONST, Call	Exponent for power-law interpolation of N-S wind component
ALPHFL	Int	INIT	MESOI	CONST	Flag to select power law vertical interpolation of wind components
ANGØ	Real	DIRSPD, TERRA		Call	Wind direction in DIRSPD converted to transport direction in TERRA
ANG1	Real	DIRSPD	TERRA	Call	Transport angle relative to terrain orientation prior to adjustment
ANG2	Real	DIRSPD	TERRA	Call	Transport angle of adjusted wind relative to the terrain orientation
ARRFL	Int	ARRIVN	MESOI	ARRIV	Flag used to indicate existence of check-point file.
ARRIVL	Real	ARRIVN, ARRTIM	EXPSUM	ARRIV	Array with check point location and status
AXPADV	Real	MESOI, DIFDEP, INIT	PRINTE, SCREEN	MATRIX	Time-integrated ground-level air concentration of A durin the current advection period (mass -hr/m^3)
AXPDPL	Real	MESOI, INIT	PRINTE, SCREEN	MATIRX	Time-integrated ground-level air concentration of A durin simulation to current advection period (mass-hr/m^3)
BDECA1	Real	RDKIN, RELEAS	MESOI	DECAY	Fraction of B remaining after decaving for one advection period
BDECA2	Real	RDK	DIFDEP	DECAY	Fraction of B remaining at the end of a sampling interval
BDEPOS	Real	MESOI, INIT	PRINTE, SCREEN	MATRIX	Surface concentration of B (mass/m^2) accumulated prior to current advection period corrected for decay
BDIN	Real	RDK	DIFDEP	DECAY	Fraction of A decaying to B during remainder of advection period given the current sampling interval
BDK	Real	RDK	DIFDEP	DECAY	Fraction of B remaining at the end of an advection period given the current sampling interval
BDPADV	Real	MESOI, DIFDEP, INIT	PRINTE, SCREEN	MATRIX	Amount of B deposited in the current advection period (mass/m^2)
BDQP	Real	CLEAN, DIFDEP, PUFFR	TESTM	PUFFS	Source term for B corrected for depletion

Name	Type	Where Defined	Where Used	How Passed	Contents
BINAS	Real	RDKIN	MESOI	DECAY	Fraction of A decaying to B during an advection period corrected for B decay
BINSI	Real	RDK	DIFDEP	DECAY	Fraction of A decaying to B during a sampling interval corrected for B decay.
BLMBDA	Real	RDKIN, RELEAS	RDK	DECAY	Decay coefficient for B (daughter) (min^{-1})
BXPADV	Real	MESOI, DIFDEP INIT	PRINTE, SCREEN	MATRIX	Time-integrated ground-level air concentration of B during current advection period (mass-hr/m^3)
BXPDPL	Real	MESOI, INIT	PRINTE, SCREEN	MATRIX	Time-integrated ground-level air concentration of B for simulation to the current advection period
CHIMIN	Real	INIT	DIFDEP	CONST	Minimum puff concentration of interest (mass/m^3)
CONV	Real	DATIM.INC	WIND	DATIM	Conversion factor for wind speed to give transport in m/s
DATA	Int	DATRD	DATWR, WIND	DATIM	Input surface wind data array
DDATE	Char	DTCHAR	PLOTZ, WNDFLD, WNDPLT	Call	Date character string - model time
DDAY	Int	DATRD	MESOI, DATWR, DTCHAR, PLOTZ, PRINTE, RELEAS SCREEN, SHIFT, WIND, WNDFLD, WNDPLT	DATIM, Call	Day of month of input meteorlogical data
DELXY	Real	GRIDIN	MESOI, ARRIVN, DIFDEP, INIT, RELEAS, TESTM, TOPFIL	CONST, Call	Grid node spacing (m) (wind grid)
DHR	Int	DATRD	MESOI, DATWR, DTCHAR, PLOTZ, PRINTE, SCREEN, SHIFT, WIND, WNDFLD, WNDPLT	DATIM, Call	Hour of the day for input meterological data
DIR	Int	WIND, WNDPLT	WNDFLD	WINDS	Wind direction (direction from which wind comes)
DKCOEF	Real	RDKIN	RDK, RELEAS	DECAY	ALMBDA/(BLMBDA-ALMBDA)
DMIN	Int	DATRD	MESOI, DATWR, DTCHAR, PLOTZ, PRINTE, WIND, WNDFLD, WNDPLT	DATIM, Call	Minutes of the hour associated with the input meteorological data
DPFLG	Int	RELEAS	DIFDEP, MODEND PRINTE, PUFFR	REL, Call	Flag used to indicate depositing effluent
DREC	Int	DATRD		DATIM	Sequential record for meteorological data - not used
DSMTR	Real	MESOI	DIFDEP	Call	Distance moved by a puff in an advection period (m)
DSMTRI	Real	DIFDEP	SIGMA	Call	Distance moved by a puff in a sampling interval (m)
DT	Int	DIFDEP	RDK	Call	Sampling interval in minutes
DTIME	Char	DTCHAR	PLOTZ, WNDFLD WNDPLT	Call	Time character string (model time)
DV	Real	INIT	DIFDEP	CONST	Deposition velocity (m/hr)
DXS	Real	CLEAN, PUFFM	MESOI, DIFDEP DTOPO, TESTM	PUFFS, Call	East-west distance moved by puff in an advection period (wind grid units)
DYR	Int	DATRD	DATWR, DTCHAR, PLOTZ, SCREEN, WNDFLD, WNDPLT	DATIM, Call	Year of the input meteorological data
DYS	Real	CLEAN, PUFFM	MESOI, DIFDEP, DTOPO, TESTM	PUFFS, Call	North-south distance moved by a puff in an advection period (wind grid units)
EFFLUX	Real	RELEAS		REL	Stack flow rate (m^3/s)
ELEV	Int	GRIDIN, REARING	WNDFLD, WNDPLT	STATN	Elevation of wind stations in meters above sea-level
EXPCUM	Real	DIFDEP, INIT	ARRTIM, EXPSUM PRINTE, SCREEN	MATRIX	Time-integrated ground-level air concentration assuming no decay or deposition
EXPFLG	Int	ARRIVN, ARRTIM	EXPSUM	ARRIV	Flag to indicate time-integrated, ground-level air concentration threshold has been exceeded at a checkpoint

Name	Type	Where Defined	Where Used	How Passed	Contents
FAC	Real	INIT, RELEAS	PUFFM	CONST	Conversion factor to convert m/s to grid units/advection period
FILE1	Char	ARRIVN, GRIDIN TOPFIL		Call	Name of file with meteorological observation (wind) station information
IADV	Int	MESOI	DATRD, PRINTE, TESTM	Call	Number of the advection period within the hour
IDAY	Int	INIT	JULIAN	Call	Input day of the month for conversion to Julian date
IDY	Int	RELEAS	JULIAN	Call	Input day of the month for conversion to Julian date
IMO	Int	INIT, RELEAS	JULIAN	Call	Input month of year for conversion to Julian date
IMTFLG	Int	INIT	MESOI, LOCATE, PUFFM, WINSPD	CONST, Call	Flag to indicate time period associated with meteorological data (0 = 1 hr, 1 = 15 min)
IPSUM	Int	MESOI	PRINTE, SCREEN	Call	Number of active puffs
IYR	Int	INIT, RELEAS	JULIAN, LOCATE	Call	Input year for use in conversion to Julian date
LDEPTH	Int	MESOI, DATRD INIT	MESOI, SIGMA, DATWR, DIFDEP, PLMRIZ, PUFFM, PUFFR, WNDFLD, WNDPLT	DATIM, Call	Mixing layer thickness (m)
LERROR	Int	LOCATE	INIT	Call	Flag associated with the results of searches of the meteorological data files
LUINDX	Int	MESOI, INIT	DATRD, LOCATE	Call	Logical unit associated with the meteorological data files (3 = observed data, 4 = forecast data)
LUN	Int	UNITS.INC	See Table B	UNITS, Call	Logical unit assignments
M	Int	MESOI	DIFDEP, PUFFM	Call	Puff number
MAXPUF	Int	RELEAS	MESOI	REL	Number of puffs to be released from a source
MF	Int	CLEAN, DIFDEP, INIT, PUFFR	MESOI, PLOTZ	PUFFS	Flag to indicate the status of a puff, 1 = active
NAMST	Char	GRIDIN, REARNG	WNDFLD, WNDPLT	CHAR, Call	Name of the wind observation stations
NCP	Int	ARRIVN		ARRIV	Number of active checkpoints
NPH	Int	INIT, RELEAS	MESOI	CONST	Number of puffs released per hour
NPUFFS	Int	INIT, PUFFR	MESOI	REL	Number of puffs released from a given source
NS	Int	MESOI	PUFFR, WNDSPD	Call	Number associated with specific source points
NSB	Int	ASCND, STRAY		Call	Identifying number for wind station in meteorological data file
NS1	Int	DIFDEP	RDK	Call	Number of sampling intervals in an advection period
NSOURC	Int	RELEAS	MESOI, PLOTZ, SCREEN	REL	Number of sources (4 maximum)
NUMSTA	Int	GRIDIN	ASCND, STRAY WIND	Call	Number of active meteorological (wind) stations
NXTDAY	Int	MESOI, DATRD		DATIM	Day associated with meteorological data for the observations that follow the current advection period
PARTHR	Int	RELEAS	MESOI	REL	Fraction of hour passing before initial puff release
PLMRFL	Int	INIT, RELEAS	MESOI	REL	Flag that enables plume rise computation
PRECIP	Int	DATRD	DATWR, DIFDEP	DATIM	Indicator of precipitation type and rate
PREVDY	Int	SHIFT	MESOI, ARRTIM	DATIM	Day associated with meteorological observations that precede the current puff advection period

Name	Type	Where Defined	Where Used	How Passed	Contents
PREVHR	Int	SHIFT	MESOI, ARRTIM	DATIM	Hour associated with meteorological observations that precede current puff advection period
PRISE	Real	MESOI, PLMRIZ	PUFFR	Call	Estimated plume rise above release height (m)
PUFLG	Int	MESOI, INIT		REL	Flag that enables puff release
QH	Real	RELEAS	PLMRI7	REL	Stack flow rate (m^3/s)
QP	Real	CLEAN, PUFFR	DIFDEP, TESTM	PUFFS	Undepleted source term multiplied by ACNST
QSOURC	Real	RELEAS	PUFFR	REL	Undepleted mass in each puff
RADP	Real	CLEAN, DIFDEP	PLOTZ	PUFFS	Radius of puff used for puff plotting
RCH	Real		WIND	CONST	Maximum distance (grid units) between grid point and location of wind observation to be used to interpolate wind at grid point
RDATE	Char	MESOI	INIT, PRINTE SCREEN	CHAR, Call	Real date that a MESOI simulation is made
RDKFL	Int	DIFDEP, RELEAS	PUFFR	DECAY	Flag to indicate that the primary effluent decays
RELDAY	Int	JULIAN	MESOI, RELEAS	REL, Call	Day that a release starts
RELHR	Int	RELEAS	MESOI	REL	Hour that a release starts
RT	Real	ASCND, STRAY		Call	Radial distance (wind grid units) between a grid point and a wird observation station
RTIME	Char	MESOI	INIT, PRINTE, SCREEN	CHAR, Call	Real-time-of-day that a MESOI simulation is made
S	Real	DIRSPD	TERRA	Call	Wind speed (m/s)
SHR	Int	MESOI	DATRD, PRINTE SCREEN, TESTM	Call	Simulation hour (hour within the period of the simulatio
SIGMAY	Real	SIGMA, CLEAN PUFFR	DIFDEP, TESTM	PUFFS, Call	Horizontal diffusion coefficient (m)
SIGMAZ	Real	SIGMA, CLEAN PUFFR	DIFDEP, TESTM	PUFFS, Call	Vertical diffusion coefficient (m)
SP	Int	CLEAN, PUFFR	TESTM	PUFFS, Call	Source associated with each puff
SPD	Int	PLMRIZ, WIND	WNDFLD, WNDPLT	WINDS, Call	Surface layer wind speed
SPEED	Real	DIRSPD, WINSPD	TERRA, WNDFLD	Call	Wind speed (m/s)
STAB	Int	DATRD	SIGMA, DATWR PLMRIZ, WNDFLD WNDPLT	DATIM, Call	Stability class (1 through 7)
STATUS	Int	GRIDIN	REARNG, WND	STATN	Status of a wind observation station (0 = active)
STDAY	Int	JULIAN	INIT, RELEAS LOCATE	Call	Julian date of year
STDIST	Real	STRAY	WIND	WINDS	Radial distance (wind gride units) between grid point an wind observation stations
STEMPC	Real	RELEAS	MESOI, PLMRIZ	REL	Temperature of stack effluent in °C
STHR	Int	INIT	LOCATE	Call	Hour of meteorological data associated with the starting time of MESOI simulation
STNUM	Int	STRAY	WIND	WINDS	Identifying number for wind observation stations
T	Int	SHIFT	MESOI	DATIM	Surface temperature for meteorological observation period preceding current advection period (°C)
TEMPA	Int	DATRD	DATWR	DATIM	Surface temperature (°C)
TEMPB	Int	DATRD	MESOI, DATWR, INIT	DATIM	Air temperature (°C)
THRSH1	Real	ARRIVN	ARRTIM, EXPSUM	ARRIV	Lower time-integrated ground-level air concentration threshold for determining status at checkpoint
THRSH2	Real	ARRIVN	ARRTIM, EXPSUM	ARRIV	Higher time-integrated ground-level air concentration threshold for determining status at checkpoint

Name	Type	Where Defined	Where Used	How Passed	Contents
TINC	Real	MESOI	PUFFM, WINSPD	Call	Time past the hour associated with current advection period (fraction of an hour)
TITLE	Char	INIT	MESOI, EXPSUM, PLMRIZ, SCREEN, TERRA, WIND	CHAR, Call	Descriptive title for MESOI simulation
TOPFLG	Int	TOPFIL	MESOI	TOPOGR	Flag to indicate availability of topographic data for diffusion computations
TOPO	Int	TOPFIL	DIFDEP, DTOPO	TOPOGR	Grid point locations in meters above sea level
TPMF	Real	DTOPO	MESOI, DIFDEP	Call	Elevation of terrain beneath the center of a puff at the end of an advection period
TPMI	Real	DROPO	DIFDEP	Call	Elevation of terrain beneath the center of a puff at the beginning of an advection period
TPUFF	Int	CLEAN, INIT, PUFFR	MESOI, PLOTZ, TESTM	PUFFS	Number of puffs currently being tracked
TT	Int	MESOI, INIT	SHIFT	DATIM	Surface air temperature for meteorological data period following current advection period
TWOPI	Real	INIT	DIFDEP	CONST	2 * π = 6.283185
U	Real	SHIFT	PUFFM, WINSPD	WINDS	East-west component of surface transport vecotr for meteorological data preceding current advection period (m/s)
UG	Real	SHIFT	PUFFM	WINDS	East-west component of upper level transport vector for meteorological data preceding current advection period (m/s)
UPWIND	Int	DATRD	DATWR, WIND	DATIM	Upper level wind
UU	Real	TERRA, WIND	PUFFM, SHIFT, WINSPD, WNDFLD	WINDS, Call	East-west component of surface transport vector for meteorological data following the current advection period (m/s)
UUG	Real	WIND	PUFFM, SHIFT	WINDS	East-west component of upper level transport vector for meteorological data following the current advection period (m/s)
U1	Real	TERRA	DIRSPD	Call	Component of transport vector parallel to terrain feature prior to adjustment (m/s)
U2	Real	TERRA	DIRSPD	Call	Component of transport vector parallel to terrain feature after adjustment (m/s)
V	Real	SHIFT	PUFFM, WINSPD	WINDS	North-south component of surface transport vector for meteorological data preceding the current advection period (m/s)
VG	Real	SHIFT	PUFFM	WINDS	North-south component of upper level transport vector for meteorological data preceding the current advection period (m/s)
VV	Real	TERRA, WIND	PUFFM, SHIFT WINSPD, WNDFLD	WINDS, Call	North-south component of surface transport vector for meteorological data following the current advection period (m/s)
VVG	Real	WIND	PUFFM, SHIFT	WINDS	North-south component of upper level transport vector for meteorological data following the current advection period (m/s)
V1	Real	TERRA	DIRSPD	Call	Component of transport vector perpendicular to terrain feature prior to adjustment
V2	Real	TERRA	DIRSPD	Call	Component of transport vector perpendicular to terrain feature following adjustment
WC	Real	INIT	DIFDEP	CONST	Washout coefficients (hr^{-1})
XP	Real	CLEAN, DIFDEP PUFFR	MESOI, DTOPO, PLOTZ, PUFFM, TESTM	PUFFS, Call	East-west position of center of puff (grid)
XSOURC	Real	RELEAS	PLOTZ, PUFFR, SCREEN, WINSPD	WEAL	East-west grid coordinates of sources (wind grid)
XSTA	Real	GRIDIN, REARNG	STRAY, WNDPLT	STATN	East-west grid coordinate of wind stations (wind grid)

Name	Type	Where Defined	Where Used	How Passed	Contents
YP	Real	CLEAN, DIFDEP PUFFR	MESOI, DTOPO, PLOTZ, PUFFM, TESTM	PUFFS, Call	North-south position of center of a puff (wind grid)
YSOURC	Real	RELEAS	PLOTZ, PUFFR, SCREEN, WINSPD	REAL	North-south grid coordinates of sources (wind grid)
YSTA	Real	GRIDIN, REARNG	STRAY, WNDPLT	STATN	North-south grid coordinates of wind stations (wind grid)
ZP	Real	CLEAN, PUFFR	DIFDEP, PUFFM TESTM	PUFFS	Effective release height of puff (m agl)
ZSOURC	Real	RELEAS	PLMRIZ, PUFFR	REAL	Actual release height of puff (m agl)

APPENDIX B

MODEL COMPARISON DATA

The tables in this appendix contain data abstracted from ten MESOI Version 2.0 test cases. The first four cases were used to demonstrate that the puff movement is correct, the next five cases were used to evaluate the diffusion and depletion portion of the model, and the last case provides model output for real meteorological data input. In all cases, MESOI was run using the NRCSIG subroutine to obtain diffusion coefficient estimates. The abstracted data provide a basis for evaluating model performance following installation on other computer systems. To obtain data for comparison for the first nine cases, it will be necessary to run MESOI in the test mode. This is done by using an asterisk (*) as the first character of the run title.

The comparison data for the transport test cases are contained in Table B-1. For the first three cases, a single puff is released at ground-level at wind grid position 3,8 (fine grid position 7,17). Flat terrain is assumed. The meteorological conditions are B stability and a 1000 m mixing layer thickness. The lower and upper level winds are identical and are varied, in time, as indicated in the table. In these cases, the puffs move across the grid, out and back and in a square pattern around the grid, respectively. In Case 4, puffs are released simultaneously at three levels over wind grid location 3,12 (fine grid position 7,25). The release levels are 10, 110 and 210 m. The mixing layer thickness is 210 m. Therefore, the release elevations correspond to the top of the surface layer, the middle of the mixing layer, and the top of the mixing layer respectively. The surface layer wind is from the north at 4 m/s, and the top layer wind is from the west at 4 m/s. Thus, the bottom and top puffs move with the input winds, and the middle puff moves along a diagonal.

Table B-2 and B-3 contain the abstracted comparison data for the diffusion and depletion test cases. In each case, the wind field is spatially uniform with a wind from the west at 3 m/s and D atmospheric stability is assumed.

The variables ADQP and BDQP in Table B-2 are directly related to the mass of the parent and daughter species remaining in a puff. The value of the corresponding variable for a non-depositing, non-decaying puff is a constant 5.29-4 (5.29×10^{-4}). Cases 5 through 7 are surface releases of four puffs each at wind grid position 3,8. The mixing layer thickness is assumed to be 1000 m. For Case 5, the released material is subjected to dry deposition. Case 6 adds precipitation (Class 2 -- moderate rain), and Case 7 adds radioactive decay. In Case 7 the half-life of the parent is 3600 s, and that of the daughter is 10,800 s. Cases 8 and 9 are elevated releases at 100 m with a 300 m mixing layer thickness. Case 8 involves only dry deposition, while Case 9 involves both dry deposition and decay. The half-lives for Case 9 are the same as for Case 5.

Table B-3 contains portions of the time integrated, ground-level air concentration and surface concentration matrices for Case 9. The grid positions indicated are for the fine grid, i.e. 2.5 km spacing. The left hand column in the table is at the end of the first hour, and the right is at the end of the third hour.

Case 10 provides comparison data for MESOI run the real meteorological data. The case simluates an hour release (4 puffs) from an elevation of 50 m at wind grid location 7.5, 11.5 (fine grid position 16,24). The released material is assumed to deposit, but not decay. Table B-4 gives the meteorological conditions for the period of release, and Table B-5 gives the MESOI time-integrated output for the end of the sixth hour. Note that the wind speeds are given in miles per hour.

Case 10 was run with modification of the wind field to account for the effects of terrain at Hanford. Considering the general wind direction, the effects of the wind field adjustment should only affect the output minimally.

Table B-1. Transport Comparison Data

	CASE 1		CASE 2		CASE 3		
Time (min)	Wind	X-point (grid)	Wind	X-point (grid)	Wind	x-point (grid)	y-point (grid)
0	270/2	3.00	270/2	3.00	270/2	3.00	8.00
15		3.36		3.36		3.36	8.00
30		3.72		3.72		3.72	8.00
60		4.44		4.44		4.44	8.00
120		5.88		5.88	270/2	5.88	8.00
135		6.24		6.24	360/2	6.06	7.82
150		6.60		6.60		6.06	7.46
180		7.32	270/2	7.32		6.06	6.74
195		7.86	090/2	7.32		6.06	6.38
240		8.76		6.24	360/2	6.06	5.30
255		9.12		5.88	090/2	5.88	5.12
270		9.48		5.52		5.52	5.12
300		10.20		4.80		4.80	5.12
360	270/2	11.64	090/2	3.36	090/2	3.36	5.12

CASE 4

Time (min)	z = 10 m		z = 110 m		z = 210 m	
	x-posit	y-posit	x-posit	y-posit	x-posit	y-posit
0	3.00	12.00	3.00	12.00	3.00	12.00
15	3.00	11.28	3.36	11.64	3.72	12.00
30	3.00	10.56	3.72	11.28	4.44	12.00
60	3.00	9.12	4.44	10.56	5.88	12.00
120	3.00	6.24	5.88	9.12	8.76	12.00
180	3.00	3.36	7.32	7.68	11.64	12.00
240	3.00	0.48	8.76	6.24	14.52	12.00
300	3.00	-2.40*	10.20	4.80	17.40*	12.00
360	3.00	-5.28*	11.64	3.36	20.28*	12.00

*off the grid

TABLE B-2. Diffusion and Depletion Comparison Data

		CASE 5			CASE 6	CASE 7		CASE 8		CASE 9	
Time (min)	Dist. (m)	Sigma y (m)	Sigma z (m)	ADQP	ADQP	ADQP	BDQP	Sigma Z	ADQP	ADQP	BDQP
15	2700	182.9	61.3	4.08-4	2.33-4	1.96-4	3.59-5	61.3	5.23-4	4.40-4	8.08-5
30	5400	336.3	93.3	3.72-4	1.21-4	8.57-5	3.34-5	93.3	5.03-4	3.55-4	1.39-4
45	8100	485.3	118.0	3.48-4	6.33-5	3.76-5	2.34-5	118.0	4.81-4	2.86-4	1.78-4
60	10800	630.0	138.9	3.29-4	3.26-5	1.63-5	1.44-5	138.9	4.62-4	2.31-4	2.03-4
90	16200	906.2	174.3	3.00-4	8.84-6	3.13-6	4.69-6	174.3	4.28-4	1.51-4	2.27-4
120	21600	1169.0	204.3	2.79-4	2.43-6	6.08-7	1.39-6	204.3	4.00-4	1.00-4	2.28-4
180	32400	1679.5	254.8	2.46-4	1.29-7	1.61-8	7.26-8	240.0	3.53-4	4.41-5	1.99-4
240	43200	2179.5	297.7	2.22-4	4.77-9	2.98-10	2.39-9	240.0	3.12-4	1.95-5	1.57-4
300	54000	2661.3	335.6	2.03-4	1.79-10	5.59-12	7.61-11	240.0	2.77-4	8.64-6	1.18-4
360	64800	3128.1	370.0	1.87-4	6.76-12	1.06-13	2.38-12	240.0	2.45-4	3.83-6	8.61-5

TABLE B-3. Time-Integrated Comparison Data for Case 9

	Elapsed Time = 1 hr					Elapsed Time = 3 hr					
	8	9	10	11	12	10	11	12	13	14	15
a. Undepleted Air (EXPCUM)											
19											4.92.15
18				1.08-14	3.70-16		4.96-14	6.06-13	2.42-12	5.55-12	9.41-12
17	6.71-10	3.94-10	1.89-10	6.97-11	9.75-13	3.80-10	2.84-10	2.19-10	1.77-10	1.47-10	1.26-10
16				1.08-14	3.70-16		4.96-14	6.06-13	2.42-12	5.55-12	9.41-12
15											4.92-15
b. Parent Air (AXPDPL)											
19											1.04-15
18				5.23-15	1.73-16		2.37-14	2.40-13	7.92-13	1.50-12	2.11-12
17	5.73-10	2.78-10	1.12-10	3.48-11	4.57-13	2.25-10	1.42-10	8.98-11	5.95-11	4.09-11	2.90.11
16				5.23-15	1.73-16		2.37.14	2.40.13	7.92-13	1.50-12	2.11-12
15											1.04-15
c. Daughter Air (BXPDPL)											
19											2.13-15
18				3.93-15	1.37-16		1.82-14	2.42-13	1.01-12	2.39-12	4.08-12
17	9.04-11	9.48-11	5.81-11	2.48-11	3.61-13	1.17-10	1.01-10	8.65-11	7.93-11	6.43-11	5.48-11
16				3.93-15	1.37-16		1.82-14	2.42-13	1.01-12	2.39-12	4.08-12
											2.13-15
d. Parent Surface (ADEPOS)											
19											2.25-14
18				1.73-13	5.89-15		2.62-13	3.09-12	1.19-11	2.65-11	4.35-11
17	1.58-8	8.16-9	3.45-9	1.12-12	1.55-11	2.09-9	1.51-9	1.12-9	8.75-10	7.06-10	5.86-10
16				1.73-13	5.89-15		2.62-13	3.09-12	1.19-11	2.65-11	4.35-11
15											2.25-14
e. Daughter Surface (BEDPOS)											
19											7.80-14
18				1.53-13	5.19-15		9.06-13	1.07-11	4.13-11	9.16-11	1.51-10
17	7.49-9	4.93-9	2.55-9	9.98-10	1.37-11	7.25-9	5.24-9	3.89-9	3.03-9	2.44-9	2.03-9
16				1.53-13	5.19-15		9.06-13	1.07-11	4.13-11	9.16-11	1.51-10
15											7.80-14

TABLE B-4. Meteorological Data for Model Comparison Case 10

		HOUR						
		1	2	3	4	5	6	7
Stability Class		5	6	6	7	7	6	4
Mixing Layer (m)		1500	120	120	120	120	300	460
Upper Wind*		350/15	350/15	350/15	350/15	350/15	350/15	350/15
		Surface Winds*						
x(grid)	y(grid)							
8.79	5.05	340/14	350/16	180/3	240/3	010/2	010/2	340/7
6.86	5.13	330/16	310/12	310/5	300/7	320/11	030/12	330/14
6.75	7.27	300/10	290/7	300/7	280/4	300/6	280/5	300/8
4.44	7.73	270/5	240/9	240/9	240/8	240/6	230/4	310/4
9.14	9.37	320/11	020/11	260/2	330/2	330/6	330/6	320/5
7.22	8.75	320/12	290/10	280/9	290/7	290/9	300/11	320/7
5.53	8.49	310/8	290/6	290/5	290/3	250/2	280/3	320/8
2.52	12.88	290/8	260/6	280/6	290/3	290/6	310/4	250/3
9.70	6.04	320/16	340/7	050/4	310/5	310/3	350/3	330/5
4.10	9.34	290/5	280/8	270/7	290/6	310/11	310/13	320/13
10.70	4.37	220/5	170/3	240/5	240/6	240/6	200/3	160/3
9.14	7.05	300/16	300/8	240/3	280/9	290/9	290/5	300/9
6.66	11.39	280/8	290/6	190/4	230/6	260/3	260/2	220/3
9.89	6.75	310/10	340/4	050/3	320/3	280/1	360/1	320/3
11.47	5.53	330/10	010/5	340/8	360/5	030/6	360/7	340/5
8.10	9.71	320/16	340/15	010/11	060/11	010/11	360/13	020/10
11.56	8.36	310/12	010/6	020/3	040/3	050/5	060/5	130/1
10.43	2.98	320/9	130/4	240/4	280/4	290/3	290/4	090/2
10.68	10.86	300/6	350/6	050/6	030/3	030/4	010/4	330/5
5.98	5.19	360/18	360/13	010/13	010/17	030/15	030/18	020/20
6.03	8.95	320/10	270/8	290/7	320/7	320/9	330/11	330/12
13.24	2.05					330/4	310/6	310/7

*speed in mph

TABLE B-5. Time-Integrated Comparison Data for Case 10

a. Undepleted Air (EXPCUM)

	17	18	19	20	21	22	23	24	25	26	27
23	1.72-10										
22	2.92-10	3.32-11									
21	2.25-10	7.01-12	1.43-16								
20	3.35-11	9.91-11	3.75-11								
19	4.67-17	2.48-10	6.74-11	5.71-15							
18		9.40-10	4.65-10	2.59-11							
17		5.88-16	3.16-10	2.86-10	5.98-11	8.10-16					
16			5.01-13	5.65-10	3.30-10	7.92-12					
15				4.38-11	3.87-10	2.73-10	8.73-14				
14				6.65-13	1.82-10	2.46-10	1.14-11				
13					6.35-13	1.41-10	7.59-11	2.86-15			
12					9.60-16	2.49-11	1.83-10	1.75-13			
11						2.46-12	2.59-10	4.39-12			
10						1.32-13	1.58-10	7.61-11	1.22-13		
9						2.10-15	2.23-11	3.03-10	3.48-11	1.01-13	
8							8.56-13	1.01-10	2.38-10	1.95-11	9.41-14
7							5.73-15	4.18-12	1.15-10	1.55-10	1.41-11
6								1.34-14	6.24-12	1.04-10	1.01-10
5									6.97-14	1.18-11	9.36-11
4									3.31-16	5.00-13	2.29-11
3										7.38-15	1.89-12
2											6.25-14
1											7.22-16

Undepleted Time-Integrated Concentration

b. Depleted Air (AXPDPL)

	17	18	19	20	21	22	23	24	25	26	27
23	1.69-10										
22	2.79-10	3.12-11									
21	2.06-10	3.63-12	1.27-16								
20	2.95-11	8.58-11	3.22-11								
19	3.94-17	2.04-10	5.56-11	4.64-15							
18		7.35-12	3.23-10	1.63-11							
17		4.25-16	2.18-10	1.86-10	2.88-11	3.96-16					
16			3.30-13	3.42-10	1.64-10	3.44-12					
15				2.38-11	1.99-10	1.12-10	3.45-14				
14				3.24-13	7.79-11	9.39-11	4.28-12				
13					2.37-13	5.02-11	2.68-11	9.83-16			
12					3.26-16	8.39-12	6.15-11	5.73-14			
11						8.01-13	8.42-11	1.38-12			
10						4.08-14	4.90-11	2.32-11	3.56-14		
9						6.10-16	6.52-12	8.74-11	9.62-12	2.73-14	
8							2.34-13	2.74-11	6.25-11	4.94-12	2.23-14
7							1.52-15	1.07-12	2.87-11	3.76-11	3.33-12
6								3.28-15	1.48-12	2.43-11	2.32-11
5									1.59-14	2.65-12	2.07-11
4									7.62-17	1.80-13	4.88-12
3										1.54-15	3.90-13
2											1.25-14
1											1.39-16

Time-Integrated Concentration Depleted Puffs

c. Surface (ADEPOS)

	17	18	19	20	21	22	23	24	25	26	27
23	6.09-9										
22	1.00-8	1.12-9									
21	7.42-9	2.29-10	4.57-15								
20	1.06-9	3.09-0	1.16-9								
19	1.42-15	7.34-9	2.00-9	1.67-13							
18		2.64-10	1.16-8	5.88-10							
17		1.53-14	7.85-9	6.68-9	1.04-9	1.33-14					
16			1.19-11	1.23-8	5.89-9	1.24-10					
15				8.57-10	7.16-9	4.02-9	1.24-12				
14				1 17-11	2.80-9	3.38-9	1.54-10				
13					8.54-12	1.81-9	9.65-10	3.54-14			
12					1.17-14	3.02-10	2.21-9	2.06-12			
11						2.88-11	3.03-9	4.98-11			
10						1.47-12	1.77-9	8.33-10	1.28-12		
9						2.20-14	2.35-10	3.15-9	3.46-10	9.83-13	
8							8.44-12	9.85-10	2.25-9	1.78-10	8.21-13
7							5.48-14	3.87-11	1.03-9	1.35-9	1.20-10
6								1.18-13	5.35-11	8.76-10	8.37-10
5									5.72-13	9.54-11	7.45-10
4									6.62-15	3.88-12	1.76-10
3										5.56-14	1.40-11
2											4.50-13
1											5.00-15

Deposition

DISTRIBUTION

No. of
Copies

OFFSITE

U.S. Nuclear Regulatory Commission
Division of Technical Information
and Document Control

Mr. B. Zalcman
Division of Emergency Preparedness and Engineering Response
Office of Inspection and Enforcement
U.S. Nuclear Regulatory Commission
Washington, DC 20555

ONSITE

DOE Richland Operations Office

D.R. Elle

49 Pacific Northwest Laboratory

KJ Allwine
GL Andrews
GF Athey (5)
DW Dragnich
JG Droppo
CS Glantz (5)
RK Hadlock
DJ Hoitink (8)
CG Lindsey
JV Ramsdell (15)
WF Sandusky
CD Whiteman
RK Woodruff
Publishing Coordination (2)
Technical Information (5)

NRC FORM 335
(11-81)

U.S. NUCLEAR REGULATORY COMMISSION

BIBLIOGRAPHIC DATA SHEET

1. REPORT NUMBER *(Assigned by DDC)*
NUREG/CR-3344
PNL-4753

4. TITLE AND SUBTITLE *(Add Volume No., if appropriate)*
MESOI Version 2.0: An Interactive Mesoscale Lagrangian Puff Dispersion Model with Deposition and Decay

2. *(Leave blank)*

3. RECIPIENT'S ACCESSION NO.

7. AUTHOR(S)
J.V. Ramsdell, G.F. Athey, and C.S. Glantz

5. DATE REPORT COMPLETED
MONTH September | YEAR 1983

9. PERFORMING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*
Pacific Northwest Laboratory
PO Box 999
Richland, WA 99352

DATE REPORT ISSUED
MONTH November | YEAR 1983

6. *(Leave blank)*

8. *(Leave blank)*

12. SPONSORING ORGANIZATION NAME AND MAILING ADDRESS *(Include Zip Code)*
Division of Emergency Preparedness and Engineering Response
Office of Inspection and Enforcement
U.S. Nuclear Regulatory Commission
Washington, DC 20555

10. PROJECT/TASK/WORK UNIT NO.

11. FIN NO.
B1282

13. TYPE OF REPORT
Technical

PERIOD COVERED *(Inclusive dates)*

15. SUPPLEMENTARY NOTES

14. *(Leave blank)*

16. ABSTRACT *(200 words or less)*

MESOI Version 2.0 is an interactive Lagrangian puff model for estimating the transport, diffusion, deposition and decay of effluents released to the atmosphere. The model is capable of treating simultaneous releases from as many as four release points, which may be elevated or at ground-level. The puffs are advected by a horizontal wind field that is defined in three dimensions. The wind field may be adjusted for expected topographic effects. The concentration distribution within the puffs is initially assumed to be Gaussian in the horizontal and vertical. However, the vertical concentration distribution is modified by assuming reflection at the ground and the top of the atmospheric mixing layer. Material is deposited on the surface using a source depletion, dry deposition model and a washout coefficient model. The model also treats the decay of a primary effluent species and the ingrowth and decay of a single daughter species using a first order decay process.

The report is divided in two parts. The first part discusses the theoretical and mathematical bases upon which MESOI Version 2.0 is based. The second part contains the MESOI computer code.

17. KEY WORDS AND DOCUMENT ANALYSIS

17a. DESCRIPTORS

17b. IDENTIFIERS/OPEN-ENDED TERMS

18. AVAILABILITY STATEMENT
unlimited

19. SECURITY CLASS *(This report)*
unclassified

20. SECURITY CLASS *(This page)*
unclassified

21. NO. OF PAGES

22. PRICE
$