PNL-4369
UC-11

User's Guide to the MESOI Diffusion Model—Version 1.1 (For Data General Eclipse S/230 with AFOS)

G. F. Athey
J. V. Ramsdell

September 1982

Prepared for the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Operated for the U.S. Department of Energy
by Battelle Memorial Institute

Battelle

PNL-4369

DISCLAIMER

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC06-76RLO 1830

Printed in the United States of America
Available from
National Technical Information Service
United States Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151

NTIS Price Codes
Microfiche A01

Printed Copy

Pages	Price Codes
001-025	A02
026-050	A03
051-075	A04
076-100	A05
101-125	A06
126-150	A07
151-175	A08
176-200	A09
201-225	A010
226-250	A011
251-275	A012
276-300	A013

PNL-4369
UC-11

USER'S GUIDE TO THE MESOI DIFFUSION MODEL -- VERSION 1.1 (FOR DATA GENERAL ECLIPSE S/230 WITH AFOS)

G. F. Athey
J. V. Ramsdell

September 1982

Prepared for
the U.S. Department of Energy
under Contract DE-AC06-76RLO 1830

Pacific Northwest Laboratory
Richland, Washington 99352

ABSTRACT

MESOI is an interactive, Lagrangian puff trajectory model. The model theory is documented separately (Ramsdell and Athey, 1981). Version 1.1 is a modified form of the original 1.0. It is designed to run on a Data General Eclipse computer. The model has improved support features which make it useful as an emergency response tool. This report is intended to provide the user with the information necessary to successfully conduct model simulations using MESOI version 1.1 and to use the support programs STAPREP and EXPLT.

The user is also provided information on the use of the data file maintenance and review program UPDATE. Examples are given for the operation of the program. Test data sets are described which allow the user to practice with the programs and to confirm proper implementation and execution.

CONTENTS

FIGURES

TABLES

USER'S GUIDE TO MESOI 1.1 AND THE SUPPORT PROGRAMS STAPREP, EXPLT AND UPDATE

INTRODUCTION

MESOI is an interactive, Lagrangian puff trajectory diffusion model. It is an interactive version of the MESOI model, and both models are direct derivatives of the Lagrangian puff trajectory model developed by Start and Wendell (1974) for the National Reactor Testing Station in Idaho. The MESOI model is documented separately (Ramsdell and Athey, 1981). Version 1.0 of the model was designed to operate on a UNIVAC 1104 computer. A separate user's guide is available for that version (Athey, Ramsdell and Allwine, 1981). Version 1.1 was created to run on a Data General Eclipse S/230 computer with AFOS graphics. This report is intended to provide MESOI users with the information necessary to conduct model simulations using version 1.1. It also is intended to familiarize the user with the MESOI support programs: STAPREP, EXPLT and UPDATE.

The user's guide is divided into two parts; the first deals with MESOI and its supporting programs and the second deals with the data file management program UPDATE. Each part includes a section describing the differences between version 1.1 and the previous version. Differences are separated into those which are improvements or changes to the model operational design and those which were necessary for implementation on the DG Eclipse using FORTRAN IV. This discussion is followed by instructions in the details of program execution. This includes data file preparation and the system commands to properly execute the code. Finally, for each program, a complete example is presented for practice and to be used in testing. Appendices contain complete source listings of the various codes as well as a checkout case and a brief description of AFOS graphics.

MESOI -- VERSION 1.1

MESOI Version 1.1 is designed to operate on a Data General Eclipse computer running the RDOS 6.1 operating system. The CLI (Command Line Interpreter) is the user interface to RDOS. The programs are written in Data General Fortran IV which is an implementation of ANSI FORTRAN Standard X3.9-1966 plus extensions. Certain output options of this version are specifically designed to use the National Weather Service AFOS (Automation of Field Operations and Services) graphics software and displays. However, the model should be easily adaptable to any mini-computer which has the memory required to load and execute the programs and which supports FORTRAN IV with extensions comparable to the Data General implementation. Modifications of MESOI Version 1.1 to run with fewer or different FORTRAN IV extensions should not be a difficult task, once the unsupported functions are identified.

SUMMARY OF DIFFERENCES FROM VERSION 1.0

Differences between Versions 1.0 and 1.1 can be categorized into two classes: those which were required to adapt the programs to the different machine (system dependent) and those which were improvements or variations but which can be considered 'system independent'. Table 1.0 summarizes the differences.

MESOI EXECUTION

The execution of MESOI, discussed in the following section, requires four data files. They are the meteorological station location and status file STATIONS, the arrival checkpoint location file CHKPNT, the observed meteorological data file OBSDAT and the forecast data file FORDAT. The program STAPREP may be used to modify the STATIONS file. CHKPNT is changed using the editor. The program UPDATE is used to maintain the files OBSDAT and FORDAT. Structure and formats of the individual data files are described in later sections.

TABLE 1. Summary of MESOI Differences Between Versions 1.0 and 1.1

System Feature	Version 1.0	Version 1.1
DEPENDENT:		
computer	UNIVAC 1100/44	Data General Eclipse S/230
source language	Sperry UNIVAC 1100 Series FORTRAN (ASCII) ANSI X3.9 - 1978	Data General FORTRAN IV plus extensions ANSI X3.9 - 1966
operating system	UNIVAC 1100 Series Executive System Level 36R2	Data General RDOS (Real Time Disk Operating System) Level 6.1
word length	36 bits	16 bits
memory	65K decimal words	32K bytes
file handling	Files assigned using control statements outside the actual programs	Files opened and closed from within the programs
logical units	Defaults differ between the system; see Appendix A for a complete comparison of the channel numbers used.	
graphics software	Line printer graphics; no special software used	AFOS graphic; uses library routines designed for displaying products on AFOS terminals
number of puffs allowed	500	72
INDEPENDENT:		
output option selection	Number keyed selection; only one option selectable each hour	Letter keyed selection; multiple or repetitive selection each hour
exposure checkpoints	Not available	Sets up grid points to be monitored; reports times when predetermined concentrations are exceeded
random receptors	Grid locations specified to measure cumulative exposure over a present interval	Not available
alternate execution modes	Not available	User selectable options to aid in program testing
data format	Year not included -- 5 digit winds - DDDFF	Year included -- 4 digit winds - DDFF
stations selection and modification	Performed in MESOI	Done prior to MESOI execution using the STAPREP program
CHI matrix files	Used by the separate program NEWCONTUR to plot exposure contours over the grid	Used by the separate program EXPLT to plot exposure vs. time for selected grid locations
output subroutines	Single modules -- PRINTE	Multiple modules -- PRINTE, SCREEN, PUFPLT and EXPSUM

The Data General Eclipse computer is dedicated to support of the Hanford AFOS system. MESOI Version 1.1 is designed primarily to operate on this system. The mapped RDOS can support two different programs at a time in a foreground-background operation. Most applications programs run in background while the foreground accomodates the AFOS data handling and display tasks. As a result of the shared resources, execution speeds of MESOI vary with the work load and user or AFOS activity in foreground.

There are no login or logout procedures to be followed with this system. The user need only have access to the terminal (Dasher or ADM3A) which is acting in background. AFOS messages may appear on the background terminal. They always begin with '!!' characters but have no effect on MESOI execution.

To execute MESOI, the current directory must contain an executable version of the model (MESOI.SV). This save file is created by linking the program elements together with the required system and library routines. The load line used with the Eclipse RDOS is:

```
RLDR/R/P CNTRL JULIAN [GRIDIN, ARRIVN, INIT, LOCA1, WIND, RELEAS,
         PUFFR PUFFM LOCA2 PRINTE, SCREEN, EXPSUM, PUFPLT]
         WRG2.LB NSOLIB.LB FORT.LB MESOI/S
```

The square brackets indicate overlay structure used for optimization of storage space. The libraries WRG2 and NSOLIB are part of the AFOS graphics software.

The MESOI model can be executed by entering the command 'MESOI<cr>' at the background terminal. The system should respond with the MESOI header message. The user is then in an interactive mode with the model.

Model execution stops normally under the following conditions:

1) 48 hours of simulation have been completed
2) the user has selected termination (output option = T)
3) the program detects file, data or input errors

In these cases, system or program messages make it clear that processing has ceased. The system is no longer executing MESOI when the prompt 'R' appears. If at any time it is necessary to stop execution at a non-standard point, the system can be interrupted by entering control A.

USER INTERACTION WITH MESOI

The MESOI header information is followed by a pause. When the user hits the carriage return at this point, the program begins prompting for the required initialization information.

Data entry is relatively simple. If illegal characters are entered, the program prompts for input again. If more than one value is requested, the values may be entered with commas between numbers or individually with a <CR> after each. Where possible (e.g., dates or times), checks are made to assure that realistic numbers have been entered. User review of input usually occurs after extensive entry of information. With single parameter entries, only one opportunity for input is provided. If a bad value is entered and accepted by the program, the user may have no choice but to terminate execution and restart.

MESOI is set up with the following conditions as default:

wind grid size of 16 x 16
grid spacing of 5000 meters
22 active wind stations (as preset in the STATIONS file)
source term unknown -- unit release

These parameters remain fixed unless changed by the user. Dates, times and titles must be entered for each simulation.

OUTPUT OPTIONS

Version 1.1 of MESOI incorporates a new output selection scheme. The old version was based upon a numeric menu. The current version uses letters to represent the options available. These are easier to remember and have been implemented in a more flexible manner. The first time the user is prompted to select output options (end of first simulation hour), the following menu is displayed:

S = SCREEN L = LISTING P = PLOT T = TERMINATE
BLANK = NO OUTPUT

The user may select up to four options at any time. They are specified by a string of up to four characters. The string is acted upon from left to right. Examples of output option usage:

a) SLPT would display output on the screen, then write the CHI matrix to the listing file (OUTLIS.), generate a plotfile to show the puffs and then terminate the current simulation.

b) SSSSPT would repeat the screen output four times and then resume simulation; the P and T options would not be acted upon.

c) TSP would terminate before either screen or plot options were acted upon.

If no output is desired, a blank entry will satisfy the prompt for input and the model simulation will resume.

After the P (plot) option has been executed, it displays the name of the plotfile created for that simulation hour. At that time the graphic product is available for display on the graphic display monitor (GDM). To display the file, the following command string is entered at the AFOS console keyboard (foreground terminal):

DSP:DP1:MESOPTAA09

where MESOPT is the predefined prefix for the RDOA filename to be used. The AA is the two character plot ID specified by the user and the 09 is the data hour at which the plot was generated. Use of the ID and hour allows each file to be uniquely identified and makes it easier to recall filenames without reexamining the output listing.

The graphic products generated are designed to be displayed on an AFOS Graphic Display Module (GDM) using a defined portion of the display space. A brief description of the display space utilization is contained in Appendix F.

RESTART AND TERMINATION

After the T output option has been executed, the model displays an exposure summary. This consists of a list of each of the puff arrival checkpoints at which the final exposure exceeded the specified minimum (CHIMIN; usually equal to 1.0E-15). The table gives the times to the nearest 10 minutes when the thresholds were exceeded and gives the final exposure at the point. The summary is displayed on both the interactive device and written to the listing file.

The model then displays the question:

DO YOU WISH TO RESTART? Y OR N

Any response other than 'Y' causes the listing file to be closed and the program stops. If a 'Y' is given to the restart query, the system responds with:

DO YOU WISH TO REINITIALIZE THE GRID? Y OR N

A response of 'Y' returns the program to the beginning of the grid initialization. Any other response returns the program to the primary model initialization, skipping the grid initialization. Execution will continue using the grid specifications and arrival checkpoints of the previous run.

ALTERNATE EXECUTION PATHWAYS

MESOI 1.1 has two alternative pathways built in which change the standard input and output. The selection of these alternatives is controlled by the first character of the user specified run title.

If the title begins with a '*' character, a testing mode is used. In this mode, the model treats incoming wind speeds as having units of meters per second instead of miles per hour. Additional output statements are enabled to provide the following information about each puff:

current grid coordinates (in grid units)

x and y distances (in grid units) the puff has moved in the last advection step

total distance moved from source (meters)
sigma y and sigma z (in meters)

This mode of operation is especially useful when the model is to be compared with hand calculations, e.g., to verify the numerical technique.

If the title begins with the character '#', the error message "FEWER THAN 3 STATIONS' is suppressed in the WIND subroutine. This option is useful when single station tests are run.

EXAMPLE SESSION WITH MESOI

The following example illustrates the prompts and responses during execution of MESOI. The grid size and spacing will remain the same. Simulation starts at 8 AM on 22 April 1982. A unit release of four hours duration beginning at 8:30 AM on April 22nd is simulated. The character '>' is used in the text to show user entries. On the Eclipse, there is no prompt for input.

The data set used in this example is the test case described in Appendix E. It is provided to allow checkout of the model after implementation on a system. A star '*' may be used as the first character in the run title to select the testing mode of operation.

```
R -- the RDOS prompt
MESOI<cr> -- requests execution of the file MESOI.SV
     MESOI -- THE INTERACTIVE EDITION OF MESO
              VERSION 1.1     SEPTEMBER 1982
              TIME = 19:04:35
              DATE = 09/27/82  (current date and time as stored in computer)
PAUSE 'HIT RETURN TO CONTINUE'
```

```
MESOI --->  GRID INITIALIZATION

THE CURRENT WIND GRID IS:
        16 ROWS  16 COLUMNS

MESOI --->  STATION INITIALIZATION
        DELXY = 5000 METERS

THERE ARE CURRENTLY 30 STATIONS WITH 8 DISABLED
```

STA.	NAME	GRIDX	GRIDY	STATUS
1	PROS	10.26	3.60	0
2	EOC	8.33	3.68	0
3	ARMY	8.22	5.82	0
4	RSPG	5.91	6.28	0
5	EDNA	10.61	7.92	0
6	200E	8.69	7.30	0
7	200W	7.00	7.04	0
8	WAHL	3.99	11.43	0
9	FFTF	11.17	4.59	0
10	YAKB	5.57	7.89	0
11	300A	12.17	2.92	0
12	WYEB	10.61	5.60	0
13	100N	8.13	9.94	0
14	WPPS	11.36	5.30	0
15	FRNK	12.94	4.08	0
16	GABL	9.57	8.26	0
17	RING	13.03	6.91	0
18	RICH	11.90	1.53	0
19	SAGE	12.15	9.41	0
20	RMTN	7.47	3.74	0
21	HMS	7.50	7.50	0
22	PASC	14.71	0.60	0
23	XXXX	7.50	7.50	1
24	XXXX	7.50	7.50	1
25	XXXX	7.50	7.50	1
26	XXXX	7.50	7.50	1
27	XXXX	7.50	7.50	1
28	XXXX	7.50	7.50	1
29	XXXX	7.50	7.50	1
30	XXXX	7.50	7.50	1

```
  **END GRID INITIALIZATION**

PAUSE  'HIT RETURN TO CONTINUE'
```

```
        MESOI --->  SET UP ARRIVAL CHECKPOINTS FROM FILE ARRCP.
                    THERE ARE 30 CHECKPOINTS ACTIVE ON THE CURRENT GRID
        MESOI --->  PRIMARY INITIALIZATION
        ENTER RUN IDENTIFICATION TITLE OF UP TO 50 CHARACTERS
> TEST 1A
        ENTER A 2 CHARACTER PLOT ID
>AA
        ENTER DATE FOR START OF SIMULATION ---   MMDDYY
>042282
        JULIAN DATE = 112  1982
        ENTER HOUR FOR START OF SIMULATION
>8
        METEOROLOGICAL DATA FILE SEARCH --
                    OBSV FILE POSITIONED AT:  DAY 112  HOUR 8  RECORD 1
                    FORECAST FIEL STARTS AT:  DAY 112  HOUR 1  RECORD 1
        PAUSE  'HIT RETURN TO CONTINUE'
        MESOI --->  RELEASE INITIALIZATION
        SPECIFY COORDINATES (X,Y) OF SOURCE IN KILOMETERS FROM HMS
>-23.20711,23.20711
        ENTER DATE OF RELEASE --  MMDDYY
>042282
        TIME OF RELEASE?  HOURS,MINUTES
>8,00
        DURATION OF RELEASE?  HOURS,MINUTES
          IF CONTINUOUS,  ENTER 99,99
>2,0
```

```
        12 PUFFS WILL BE RELEASED

        IS THE SOURCE TERM KNOWN?  Y OR N

>N

        SOURCE IS LOCATED AT WIND GRID     5.1    9.9

        RELEASE WILL OCCUR AT  8:00 ON DAY 4/22
                12 PUFFS AT 0.167 CURIES PER PUFF

        PUFF RELEASE FLAT SET -->  AFTER HOUR     8   ADV STEP  4

        END OF SIMULATION HOUR    1      DATA = DAY 112    HOUR  9
                6 PUFFS ACTIVE

        SELECT OUTPUT OPTIONS

        S = SCREEN  L = LISTING  P = PLOT  T = TERMINATE  BLANK = NO OUTPUT

>SLP

        * * * * * * * * * * * * * * * * * *
      *                                     *
      *                                     *
      *                                     *      DAY 112     HOUR 9
      *          X 7                        *       6  PUFFS ACTIVE
      *              7 3                    *
      *              3 6 4                  *
      *                4 6 4                *
      *                  4M5 3              *
      *                    3 2              *
      *                                     *
      *                                     *
      *                                     *
      *                                     *
      *                                     *
      *                                     *
      *                                     *
        * * * * * * * * * * * * * * * * * *

        PAUSE 'HIT RETURN TO CONTINUE'

        OUTPUT SENT TO PRINT FILE

        6 PUFFS PLOTTED

        PLOTFILE 'MESOPTAA09' CREATED
                3434 WORDS USED
```

```
END OF SIMULATION HOUR    2    DATA + DAY 112   HOUR 10
      12 PUFFS ACTIVE

SELECT OUTPUT OPTIONS

(in the sample case, model simulation was run for two more hours
before termination)

>T

END OF SIMULATION

EXPOSURE SUMMARY FOR MESOI --    TEST
```

CHECKPOINT NAME	CHI X,Y COORDINATES		EXCEEDED 3.333E-17 DAY	TIME	3.333E-12 DAY	TIME	FINAL CONCENTRATION
200 EAST	18.	16.	112	850	112	9 0	3.268E-8
200 WEST	15.	15.	112	840	112	9 0	7.315E-8
NSTF	18.	18.	112	850	112	950	8.890E-8
GABLE MT	20.	16.	112	9 0	0	0 0	1.536E-9
LANDFILL	20.	13.	112	910	0	0 0	9.702E-9
ARMY LP	17.	13.	112	9 0	0	0 0	4.711E-11
VERNITA	12.	20.	112	8 0	112	810	6.442E-7
400 AREA	23.	10.	112	950	0	0 0	4.448E-12
WNP-2	24.	12.	112	950	0	0 0	4.457E-11

```
CHI MATRIX FILE CLOSED AND AVAILABLE FOR EXPLT

DO YOU WISH TO RESTART?  Y OR N

>N

R       (RDOS prompt - signals program execution has ceased)
```

MESOI COMPUTER PROGRAM DESCRIPTION

MESOI is a modified version of the regional transport and diffusion model developed by Start and Wendell (1974). The major modifications made to the original program were: to reconstitute the program as an interactive code, to increase the modularization of the code, to revise the numerical algorithms used in the computation of the diffusion parameters, to modify the code to do puff arrival-at-grid-point checking and to do plotting on the AFOS system. Again, no fundamental changes were made to the basic approach taken by Start and Wendell. A full discussion of the theory, numerical approach and sensitivity of the model is contained in Ramsdell and Athey, 1981. A listing of MESOI is contained in Appendix B.

PROGRAM ORGANIZATION

The computer code structure is based upon a set of five nested loops. Three of the loops are transparent to the program user. The apparent loops are the outer two; controlling number of simulations and number of hours in each simulation, respectively. Figure 1 shows a flowchart of the program as it appears to the user. Figure 2 shows a more complete view of activity within MESOI. Detailed discussion of the model loops is contained in the documentation for Version 1.0 (Ramsdell and Athey, 1981).

The major program elements of MESOI are listed in Table 2. A list of the system routines called from the programs is given in Table 3. Common block names and contents are presented in Table 4.

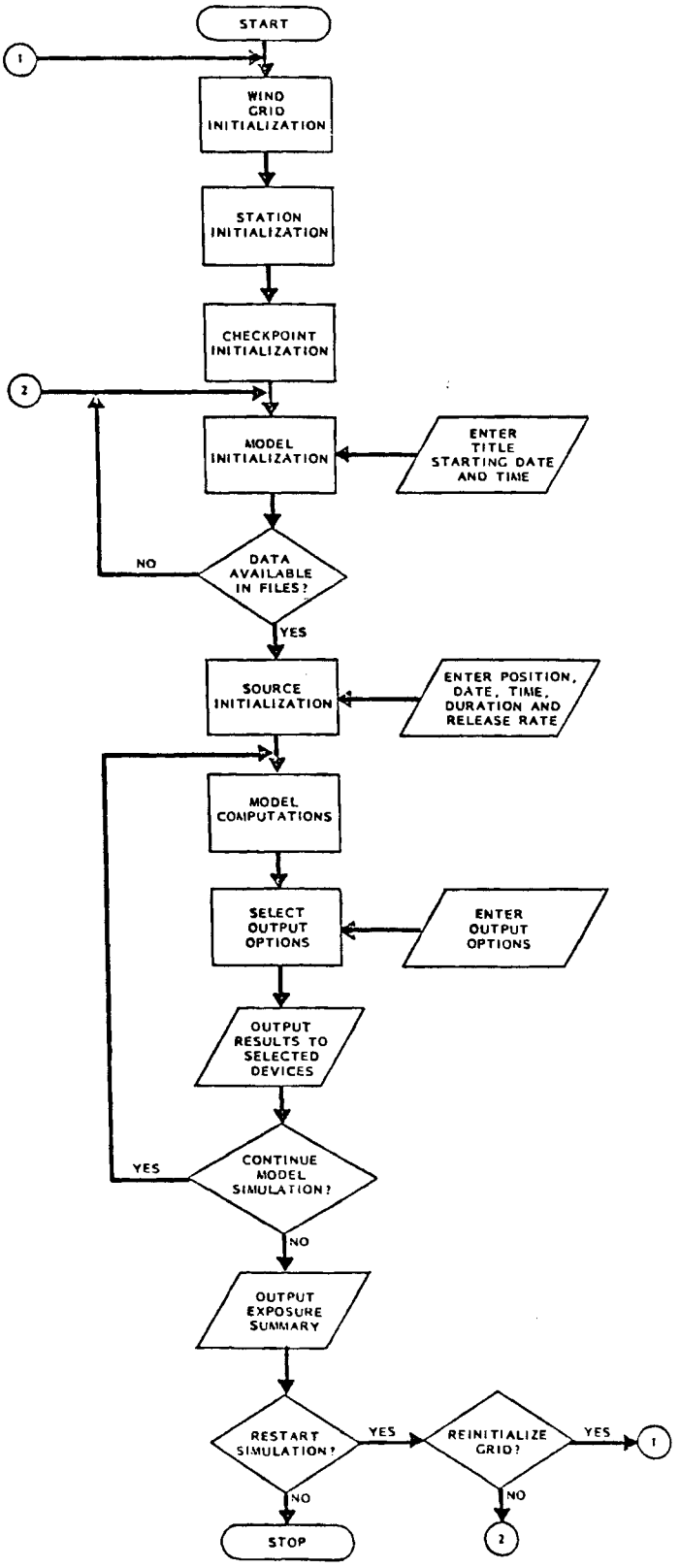

FIGURE 1. General Flowchart for MESOI 1.1.

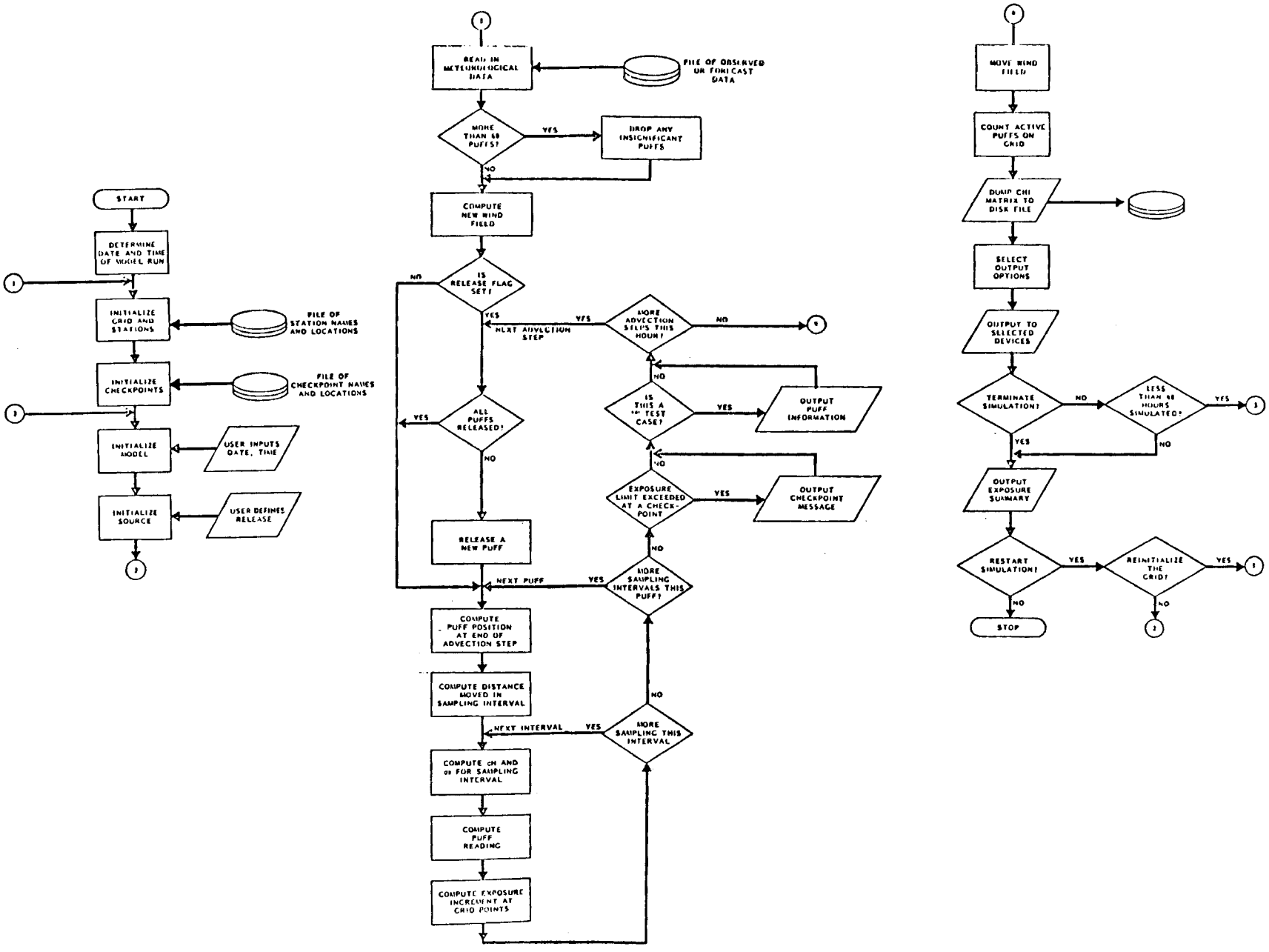

FIGURE 2. MESOI 1.1 Activity Flowchart

TABLE 2. MESOI Program Elements

Module Name	Purpose
CNTRL	Program control; Major computational loops
GRIDIN	Grid and wind station initialization
ARRIVN	Define the exposure checkpoints
INIT	Primary model initializations
JULIAN	Convert month, day and year to Julian date
LOCATE (LOCA1, LOCA2)	Position observed data file (MDATA) for the start of simulation (renamed to allow use as separate overlays)
WIND	Decode wind data and compute the U and V components for the grid points
RELEAS	Define the source location and parameters
PUFFR	Release puffs
PUFFM	Compute puff movement
SIGMA	Compute horizontal and vertical diffusion parameter values
ARRTIM	Keep track of concentrations at checkpoints
CLEAN	Remove puffs which are no longer being considered
SCREEN	Display a simple graphic output on the interactive device
PRINTE	Output the simulation results to the printer file (OUTLIS.)
PUFPLT	Generate an AFOS plotfile
EXPSUM	Summarize the exposure history at the checkpoints
AFOS plot routines:	
COMMH	Write header to plotfile for use by AFOS
GPD1	
MOVEVEC	Draw lines between sets of x,y pairs
TEXT	Plot text strings

TABLE 3. System Routines Used

OPEN	CALL OPEN (channel no., filename, mode, error) a sequentially organized file is opened or created
CLOSE	CALL CLOSE (channel no., error) Close the specified channel
OVOPN	CALL OVOPN (channel no., filename, error) open the overlay file associated with the program; must be called before any overlays are loaded
OVLOD	CALL OVLOD (channel no., overlay name, flag, error) channel number must contain the overlay file; flag indicates conditional(0) or unconditional(1) load of the overlay segment
CHSAV	CALL CHSAV (channel no., start word) used to save the channel status in a three word block such that the file can be reread later
CHRST	CALL CHRST (channel no., start word) restores the channel status from the block and allows the file to be read again from the same location
TIME	CALL TIME (array, error) puts time of day into a 3-element integer array
DATE	CALL DATE (array, error) puts date into a 3-element integer array

TABLE 4. MESOI Common Blocks

Name	Contents
MASTER	Model run identification, control parameters
REL	Source term specifications
PUFFS	Puff characteristics and positions
WINDS	Identification and distances to closest wind stations for each grid point in the wind grid; interpolated wind field
STATN	Wind station positions, names and status
ARR	Arrival checkpoint locations and times
PLT	Plot ID characters and puff plot radii
GRID	CHI matrix values (exposure accumulation array)
VECTOR	U and V component winds for current hour and previous hour
PXTRA	Arrays used locally in PUFPLT
WXTRA	Arrays used locally in WIND
CON	Constants used locally in SIGMA
PRI	Constants used locally in SCREEN

UPDATE VERSION 1.1

UPDATE is the utility program used to maintain the observed and forecast meteorological data files for use by the MESOI diffusion model. The program is interactive and is designed to minimize the work of keeping current data available for use. The simple question and answer format simplifies the background and training necessary to become proficient in the use of the programs.

SUMMARY OF DIFFERENCES FROM VERSION 1.0

Only minor changes were made to the data management programs which are assembled under the name UPDATE. The largest modification has been the elimination of the log files. No record keeping utility is now used to document the changes made in the data files. Wind data is now entered as DDFF instead of DDDFF. This reduction in wind direction resolution was prompted by the limitations of single precision integer size in 16-bit computers. Rather than rework the entire model, the last place of the wind direction was dropped. Other minor changes were made as a result of the shift to DG Fortran IV.

EXECUTION OF UPDATE

The execution of UPDATE requires the availability of the two MESOI data files, MDATA and FDATA. As with MESOI, the current directory must contain an executable version of the program or be linked to such a version. The data management program is started by typing 'UPDATE<cr>' on the background terminal. The program should respond with the UPDATE header messages.

EXAMPLE SESSION WITH UPDATE

The following example illustrates the prompts and responses during an update session. The character '>' is used in the test to show user entries. On the Eclipse, there is no prompt for input. In response, to yes-or-no questions, a single letter response of 'Y' or 'N' will work. Generally, any response other than 'Y' will be interpreted as a NO.

```
        UPDATE          VERSION 1.1          SEPTEMBER 1982

        METEOROLOGICAL DATA ENTRY PROGRAM FOR MESOI

        DO YOU WISH TO ENTER OR REVISE OBSERVED DATA?

>Y

        ENTER YEAR OF OBSERVATION TO BE ENTERED:  YY

>82

        THE MONTH, DAY AND HOUR OF THE LAST OBSERVATION ENTERED ARE:  4 22 24

        THE NEW ENTRIES MUST FOLLOW THIS TIME.

        ENTER MONTH AND DAY OF OBSERVATION TO BE ENTERED:  MM,DD

>4,23

        ENTER HOUR OF THE OBSERVATION: HH

>1

        THE DATE AND HOUR FOR OBSERVATION TO BE ENTERED ARE:  42382 1

        ARE THESE CORRECT?  ENTER Y OR N

>Y

        THE JULIAN DATE IS 112

        ENTER STABILITY CLASS:   1 THROUGH 7

>4

        ENTER MIXING DEPTH IN TENS OF METERS:

>100

        ENTER WIND DATA AS DDFF, WITH 0000=CALM AND 9999=MISSING.
                 DO NOT USE LT/VAR OR CALM
                 DO NOT ESTIMATE THE AVERAGE SPEED AND DIRECTION

        ENTER WIND FOR PROS BAR

>3107
```

```
        ENTER WIND FOR ARMY LPR

>3107

        The sequence of prompts is repeated for the remaining stations.

        DO YOU WISH TO ENTER OBSERVATIONS FOR ANOTHER TIME?

>N

        DO YOU WISH TO REVIEW THE OBSERVED DATA FILE?

>Y

        DO YOU WISH TO REVIEW THE ENTIRE OBSERVED DATA FILE?

>N

        DO YOU WISH TO REVIEW DATA FOR SPECIFIC HOURS?

>Y

        ENTER DATE OF DATA TO BE REVIEWED:  MM,DD,YY

>4,22,82

        ENTER HOUR OF THE DATA TO BE REVIEWED:  HH

>8

        DATA ARE DISPLAYED IN THE FOLLOWING ORDER:

        JULIAN DATA, HOUR, STABILITY, MIXING DEPTH, WINDS.

        THE ORDER OF THE WIND STATIONS IS:

             PROS   EOC   ARMY   RSPG   EDNA   200E   200W   WAHL
             FFTF   YAKB  300A   WYEB   100N   WPPS   FRNK   GABL
             RING   RICH  SAGE   RMTN   HMS    PASC

82112 8 4 120 3107   3107  3107   3107   3107   3107   3107   3107
              3107   3107  3107   3107   3107   3107   3107   3107
              3107   3107  3107   3107   3107   3107

        TYPE N<CR> FOR NEXT RECORD OR JUST <CR> TO EXIT

>N
```

```
82112 9 4  120  3107  3107  3107  3107  3107  3107  3107  3107
                3107  3107  3107  3107  3107  3107  3107  3107
                3107  3107  3107  3107  3107  3107

          TYPE N<CR> FOR NEXT RECORD OR JUST <CR> TO EXIT

>

          DO YOU WISH TO REVIEW ADDITIONAL HOURS?

>N

          DO YOU WISH TO REVISE ANY DATA ENTRIES?

>Y

          ENTER DATE OF OBSERVATION TO BE REVISED:  MM,DD,YY

>4,22,82

          THE JULIAN DATE FOR THE REVISION IS 112

          ENTER HOUR OF THE OBSERVATION TO BE REVISED:  HH
                A 99 CAN BE USED TO ESCAPE FROM THE DATA REVISION SUBROUTINE

>12

82112 12 4 100  3107  3107  3107  3107  3107  3107  3107  3107
                3107  3107  3107  3107  3107  3107  3107  3107
                3107  3107  3107  3107  3107  3107

          IS THIS THE RECORD TO BE REVISED?  Y OR N

>Y

          REVISIONS TO WIND DATA ONLY?  Y OR N

>N

          DO YOU WISH TO REVISE THE STABILITY?
                  ENTER Y OR N

>N

          DO YOU WISH TO REVISE THE MIXING DEPTH?
                  ENTER Y OR N

>Y
```

```
          ENTER REVISED MIXING DEPTH IN TENS OF METERS:
                  1 THRU 300

>150

          DO YOU WISH TO REVISE THE WIND DATA?
                  ENTER Y OR N

>Y

          ENTER WIND DATA AS DDFF, WITH 0000=CALM AND 9999=MISSING
                  DO NOT USE LT/VAR OR CALM
                  DO NOT GUESS AT THE AVERAGE SPEED AND DIRECTION

          ENTER STATION NUMBER TO BE REVISED, 1 - 22,  0 = EXIT

>16

          ENTER REVISED WIND DATA FOR GABL

>3210

          ENTER STATION NUMBER TO BE REVISED, 1 - 22,  0 = EXIT

>0

          DO YOU WISH TO REVISE ANOTHER RECORD?

>N

          DO YOU WISH TO REVIEW THE OBSERVED DATA FILE?

>N
```

------At this point, control is in the main program. The update of the observation file is complete and the file is rewritten. The program then queries about any updates to the forecast file.

```
          DO YOU WISH TO ENTER OR REVISE FORECAST DATA?
```

------If a yes answer is given, the FDATA file is read in and updating proceeds in a similar fashion to that described for the observed data file.

```
>N

          THIS UPDATE SESSION IS OVER
```

As can be seen from the example, UPDATE is designed to continue the input, review and revision cycle until the user is satisfied with the content of the data file.

When entering a new forecast the complete 48 hour period is filled with the data entered. Each follow-on entry will fill the file from the hour of the forecast to the end of the 48 hour period. Any revision to the forecast will cause all the data between the revision start hour and the start hour plus the hours of persistence to be modified.

STRUCTURE AND FILES

The following section describes the organization of the UPDATE program. Source listings of the modules are contained in Appendix G. UPDATE consists of a main program (MASTER) and ten subroutines.

MASTER calls the following primary subroutines directly:

NUDAT RVUDAT REVDAT NUFCST RVUFST REVFST

These six subroutines in turn call one or more of four secondary subroutines as shown in Table 5.

TABLE 5. Subroutines Called by the Primary Subroutines of UPDATE

Calling Subroutine	Secondary Subroutine JULIAN	CAL	INDEX	ARCHIV
NUDAT	X	X		X
RVUDAT	X	X		
REVDAT	X	X		
NUFCST	X	X	X	
RVUFST	X	X	X	
REVFST	X	X	X	

Following are brief descriptions of each subroutine used by the main program:

NUDAT -- Enables the addition of new observations to the MDATA file. User enters wind data station by station. Any number of hourly observations can be entered.

RVUDAT -- Reviews the observation file (MDATA). Single records or 6 hour blocks are displayed.

REVDAT -- Uses a question and answer format to allow the user to make changes to the MDATA file contents.

NUFCST -- Similar to NUDAT in operation with minor variations, except it acts on the forecast data file (FDATA).

RVUFST -- Same as RVUDAT except it displays the file contents if FDATA.

REVFST -- Same as REVDAT except it allows modifications to the forecast data file (FDATA).

JULIAN -- Converts month, day and year into a Julian date.

CAL -- Converts Julian date and year into a month and day.

INDEX -- Given the Julian date and an hour, determines the array subscript for the location into which forecast values are to be stored.

ARCHIV -- Write the previous 24 hours of meteorological observation data to a disk file name ARCHIV. The subroutine is called when the first hourly observation of a new day is entered.

Three disk files are used by UPDATE: MDATA, FDATA and ARCHIV. MDATA is the meteorological observation data file. It consists of 48 hours of observations with 3 records per hour. FDATA is the forecast data file. It has the same structure as MDATA. Both files are input via logical unit 1. ARCHIV is the long term data storage file which is used to automatically save the data. It is not required for the operation

of MESOI, but is useful in restoring data for simulations of releases more than 48 hours in the past.

The file record structure is the same for each data set. Each hour of data consists of three records. Table 6 presents a sample three record set and describes each element of the records.

TABLE 6. Data Record Structure

```
12345678901234567890123456789012345678901234567890123456789012345678901234567890
0        1         2         3         4         5         6         7         8
--------------------------------------------------------------------------------

 82112 11 4 120       3107  3107  3107  3107  3107  3107  3107  3107  3107  3107
 82112 12 4 120       3107  3107  3107  3107  3107  3107  3107  3107  3107  3107
```

APPENDIX A
Summary of Specific FORTRAN Differences

APPENDIX A -- Summary of specific changes made to MESOI and UPDATE to implement on DG Eclipse with FORTRAN IV.

1) replace block IF statements

2) replace BACKSPACE with calls to CHSAV and CHRST
   (channel status saves and restores)

3) change all A formats to S

4) alter intinsic function calls as needed

5) eliminate CHARACTER as a variable type; replace with INTEGER or REAL

6) use calls to OPEN and CLOSE for file management within the programs; eliminates need for UNIVAC ASG's and USE's

7) eliminate * from the list directed READs

8) additional checks on user input; the ERR branch on the READ statement does not catch bad character entries. In DG Fortran, the ERR transfer occurs only after an I/O error at the driver level is detected.

9) modify variable and array names to avoid reserved words (e.g., DATA)

10) put all variables to be initialized by DATA statements into labeled COMMON

11) put large arrays into labeled COMMON to keep run-time stack small†

12) make subscripts of array elements in I/O statements integer type variables; cannot be expression

13) use overlays to keep memory utilization down

14) Redefine the logical units (channel numbers)

† This modification eliminated a significant number of problems encountered in early attempts to load and run MESOI and UPDATE on the DG Eclipse computer.

Device or File	Channel Number Ver 1.0	Ver 1.1	
keyboard input	5	11	
screen or printer output	6	10	
MDATA. file	10	1	
FDATA. file	11	2	
output listing	12	3	(in MESOI)
overlay file	none	9	
CHI matrix file	none	4	
ARRCP and STATIONS files	none	8	
MDATA. file	10	1	
MDL. file	11	none	(in UPDATE)
FDATA. file	12	1	
FDL. file	13	none	

APPENDIX B

MESOI Version 1.1 Program Listing and Notes

The following section provides a complete listing of the source code for each module of the MESOI model. A description of each section was presented in the Version 1.0 documentation (Ramsdell and Athey, 1981). Discussion is presented here only for those modules which are new or significantly changed in operation. All but the simplest sections have been modified in some way to use the DG Fortran.

CNTRL

CNTRL is the main program of the MESOI model. It provides the basic model structure for simulation. Overlays are used to keep the memory requirements low.

```
C    ** MESOI.CNTRL
C
C    ** MAIN CONTROLLING ROUTINE FOR THE MESOI MODEL
C
C ***************************************************************
C
C    ** MESOI  ---  INTERACTIVE VERSION OF MESO
C    **             VERSION 1.1   SEPTEMBER 1982
C
C    **             FOR DG ECLIPSE S/230 WITH FORTRAN IV AND RDOS 6.0
C
C    **  LOAD LINE:
C    **    RLDR/R/P CNTRL JULIAN [GRIDIN,ARRIVN,INIT LOCA1,WIND,RELEAS,
C    **              PUFFR PUFFM LOCA2 CLEAN SIGMA ARRTIM, PRINTE, SCREEN,
C    **             EXPSUM, PUFPLT ] ARG2.LB NSOLIB.LB FORT.LB MESOI/S
C
C ***************************************************************
C
      REAL CH1(32,32),  STOTAL(72), QP(72)
      REAL ARRIVL(11,35), RADP(72)
      REAL XP(72),YP(72),DXS(72),DYS(72),SIGMAY(72),SIGMAZ(72)
      REAL XSOURC,YSOURC,Q,XS(31),YS(31)
      REAL XSTA(30),YSTA(30),NAMST(30),USTA(30),VSTA(30),
     +       XDIST(30),YDIST(30)
      REAL STDIST(10,16,16)
      REAL U(16,16),V(16,16),UU(16,16),VV(16,16)
C
      INTEGER RDATE(3), RTIME(3), IBLK(3), SELECT
      INTEGER DT,PFLAG, DFLAG, PLTID,IOPT(4),SHR,SDAY
      INTEGER RELDAY,RELTIM,RELHR,PARTHR,MAXPUF
      INTEGER ACTSTA, STATUS(30)
      INTEGER NF(72), TPUFFS, NPUFFS, METDAT(30)
      INTEGER DYR,DDAY, DHR, DREC, STAB, PREVHR, PREVDY
      INTEGER TITLE(50), STNUM(10,16,16)
C
```

```
      COMMON/MASTER/RDATE,RTIME,TITLE,NPH,DT,CHIMIN,DDAY,DHR,NSI,
     1 ACMST,SDAY,SHR
      COMMON/REL/XSOURC,YSOURC,RELDAY,RELHR,PARTHR,
     1    MAXPUF,Q
      COMMON/STATN/XSTA,YSTA,NAMST,STATUS,RCH,ACTSTA,USTA,VSTA,XDIST,YDIST
      COMMON/PUFFS/MF,SIGMAY,SIGMAZ,DP,STOTAL,XP,YP,TPUFFS,
     1    NPUFFS,DXS,DYS,FAC,ADT
      COMMON/WINDS/SIDIST,STNUM
      COMMON/ARR/ARRIVL
      COMMON/VECTOR/U,V,UU,VV
      COMMON/GRID/CHI
      COMMON/PLT/PLTID,RADP
C
C     **  OVERLAY ASSIGNMENTS
C
C     **  SB0 - ARRIVN
C     **  SB1 - GRIDIN
C     **  SB2 - INIT (+ LOCATE)
C     **  SB3 - RELEAS
C     **  SB4 - PUFFR (+PUFFM, SIGMA, ARRTIM,LOCATE,CLEAN)
C     **  SB5 - SCREEN
C     **  SB6 - PRINTE
C     **  SB7 - PUFPLT
C     **  SB8 - EXPSUM
C     **  SB9 - WIND
C
      EXTERNAL SB0,SB1,SB2,SB3,SB4,SB5,SB6,SB7,SB8,SB9
C
C     ** DEFINE AND INITIALIZE GRID, MODEL, RECEPTORS AND RELEASE
C
      CALL TIME(RTIME,IER)
      CALL DATE(RDATE,IER)
      CALL OVOPN(9,'MESOI.OL',IER)
```

```
      IF(IER.NE.1)  WRITE(10,1) IER
    1 FORMAT(/5X,'ERROR OPENING FILE MESOI.OL - IER = ',I3)
C
      CALL DFILE('OUTLIS.',IER)
      CALL OPEN(3,'OUTLIS.',2,IER)
      IF(IER.NE.1)  WRITE(10,2) IER
    2 FORMAT(/5X,'ERROR OPENING FILE OUTLIS. - CODE = ',I3)
C
      WRITE(10,4) RTIME, RDATE
      WRITE(3,4) RTIME, RDATE
    4 FORMAT(//////,15X,'MESOI -- THE INTERACTIVE EDITION OF MESO',
     +  /,26X,'VERSION 1.1     SEPTEMBER, 1982',
     +  //,26X,'TIME = ',I2,':',I2,':',I2,//,26X,'DATE = ',
     +  I2,'/',I2,'/',I2,////)
      PAUSE 'HIT RETURN TO CONTINUE'
C
   10 CALL OVLOD(9,SB1,1,IER)
      IF(IER.NE.1)  TYPE 'ERROR LOADING OVERLAY SB1 - IER = ',IER
      CALL GRIDIN(DELXY,NUMSTA)
C
      CALL OVLOD(9,SB0,1,IER)
      IF(IER.NE.1)  TYPE 'ERROR LOADING OVERLAY SB0 - IER = ',IER
      CALL ARRIVN(DELXY)
C
      NXTDAY = 0
   20 CALL OVLOD(9,SB2,1,IER)
      IF(IER.NE.1) TYPE 'ERROR LOADING OVERLAY SB2 - IER = ',IER
      CALL INIT(FAC,ADT,DELXY,XS,YS,NUMSTA,NXTDAY,METDAT)
C
      CALL OVLOD(9,SB9,1,IER)
      IF(IER.NE.1) TYPE 'ERROR LOADING OVERLAY SB9 - IER = ',IER
      CALL WIND(STDAY,STHR,METDAT,U,V,NUMSTA,TITLE(1))
C
      CALL OVLOD(9,SB3,1,IER)
      IF(IER.NE.1) TYPE 'ERROR LOADING OVERLAY SB3 - IER = ',IER
      CALL RELEAS(NPH,DELXY,SHR,SDAY)
```

```
C
      ICOL = 16
      IROW = 16
C
      DO 30 I = 1, 72
        MF(I) = 0
   30 CONTINUE
      IPUFFS = 0
      NPUFFS = 0
      PFLAG = 0
      DFLAG = 0
      PREVDY = DDAY
      PREVHR = DHR
C
C    ** HOURLY LOOP FOR SIMULATION
      DO 100 SHR = 1,48
        CALL OVLOD(9,SB4,1,IER)
        IF(IER.NE.1) TYPE 'ERROR LOADING OVERLAY SB4 - IER = ',IER
        IF(DFLAG.EQ.1)  GOTO 123
C    **  CHECK TO SEE IF OBSERVED METDAT (LU=16) IS EXHAUSTED
C    **  IF SO, SWITCH TO FORECAST METDAT (LU=17) AND SET DFLAG
        IF(NXTDAY.NE.888)   GOTO 105
          DFLAG = 1
          CALL LOCA2(DYR,DDAY,DHR,LERROR,DFLAG)
          GOTO 123
  105   READ(1,110,END=500,ERR=600) DYR, DDAY,DHR,DREC,STAB,LDEPTH,
     1    (METDAT(I),I=1,10)
  110   FORMAT(I2,I3,I2,I1,1X,I1,1X,I3,5X,10(1X,I5))
C
        READ(1,120,END=500,ERR=600) (METDAT(I),I=11,20)
        READ(1,120,END=500,ERR=600) (METDAT(I),I=21,30)
  120   FORMAT(19X,10(1X,I5))
      CALL CHSAV(1,IRLK(1))
        READ(1,122,END=500,ERR=600) NXTDAY
  122   FORMAT(2X,I3)
```

```
      CALL CHRST(1,IBLK(1))
        GOTO 124
C
  123   READ(2,110,END=500,ERR=600) DYR,DDAY,DHR,DREC,STAB,LDEPTH,
     1     (METDAT(I),I=1,10)
        READ(2,120,END=500,ERR=600)  (METDAT(I),I=11,20)
        READ(2,120,END=500,ERR=600)  (METDAT(I),I=21,30)
C
C
C   **  LIST THE DATA IN THE OUTPUT FILE
C
  124 LDEPTH = LDEPTH * 10
      WRITE(3,125) DYR, DDAY, DHR
  125 FORMAT(/5X,'DATA FOR DAY: ',I2,1X,I3,'  HOUR ',I2,' <PST>'/)
      WRITE(3,126) STAB, LDEPTH
  126 FORMAT(8X,'STABILITY = ',I1,3X,'MIXING DEPTH = ',I4,
     +   ' METERS',/)
      WRITE(3,127) (METDAT(I),I=1, 10)
  127 FORMAT(8X, 'WINDS',8X,10(1X,I4))
      WRITE(3,128) (METDAT(I),I=11,20)
      WRITE(3,128) (METDAT(I),I=21,30)
  128 FORMAT(21X,10(1X,I4))
C
        IF(TPUFFS.GT.(10*NPH))  CALL CLEAN
C
        CALL OVLOD(9,SB9,1,IER)
        CALL WIND(DDAY,DHR,METDAT,UU,VV,NUMSTA,TITLE(1))
        CALL OVLOD(9,SB4,1,IER)
C
        DO 340 IADV = 1, NPH
          TINC = (IADV - 1) * ADT
          IF(PFLAG.EQ.1)  GOTO 130
          IF(RELDAY.NE.PREVDY) GOTO 140
          IF(RELHR.LT.PREVHR)  GOTO 140
          IF(RELHR.NE.PREVHR.OR.IADV.NE.PARTHR)  GOTO 140
            PFLAG = 1
            WRITE(10,135) PREVHR, IADV
```

```
            WRITE(3,135) PREVHR, IADV
  135       FORMAT(/5X,'PUFF RELEASE FLAG SET ---> AFTER HOUR ',I2,
     1         '  ADV STEP ',I2,//)
C
  130     IF(NPUFFS.LT.MAXPUF)  CALL PUFFR(ACNST)
  140     DO 330 M = 1, TPUFFS
            IF(MF(M).EQ.0)  GOTO 330
C
            IF(XP(M).LE.0.0.OR.XP(M).GT.15.0)  GOTO 150
            IF(YP(M).LE.0.0.OR.YP(M).GT.15.0)  GOTO 150
            MOFG = 0
C
            CALL PUFFM(M,DXM,DYM,TIMC,MOFG)
C
            IF(MOFG.NE.0)  GOTO 150
            DXS(M) = DXM
            DYS(M) = DYM
            GOTO 160
  150       DXM = DXS(M)
            DYM = DYS(M)
  160       DS = SQRT(DXM**2 + DYM**2)
            DXM = DXM / NSI
            DYM = DYM / NSI
C
            DSMTR = DS / NSI * DELXY
            DO 320 JN = 1, NSI
              CALL SIGMA(DSMTR,M,TPUFFS,STAB,LDEPTH,JN,
     +           SIGMAZ(M),SIGMAY(M),STOTAL(M))
C
              SIGYSQ = SIGMAY(M) * SIGMAY(M)
              HSGSQ = (-0.5 * (DELXY / 2.0) ** 2) / SIGYSQ
              PUFCHI = QP(M) / (SIGYSQ * SIGMAZ(M))
C     **  SET PUFF PLOT RADIUS TO 1 PIXEL; PUFF CENTER ONLY
              RADP(M) = 0.020
            IF(PUFCHI.GT.1.0E-14)
```

```
     +          RADP(M) = (SIGMAY(M) * SQRT(-2.*ALOG(1.0E-14/
     +            PUFCHI))/DELXY)
                IF(PUFCHI.LE.CHIMIN)  GOTO 200
C
                RP = (SIGMAY(M) * SQRT(-2.*ALOG(CHIMIN/PUFCHI))/DELXY)
                XP(M) = XP(M) + DXM
                YP(M) = YP(M) + DYM
C
                IF(XP(M)+RP.LT.0.0)  GOTO 200
                IF(XP(M)-RP.GT.15.0) GOTO 200
                IF(YP(M)+RP.LT.0.0)  GOTO 200
                IF(YP(M)-RP.LE.15.0) GOTO 210
  200           MF(M) = 0
                RADP(M) = 0.020
                GOTO 330
C
  210           RP = RP * 2.0
                ISTRT = (XP(M) * 2.0)  - RP + 2
                ISTP  = (XP(M) * 2.0)  + RP + 1
                JSTRT = (YP(M) * 2.0)  - RP + 2
                JSTP  = (YP(M) * 2.0)  + RP + 1
C
                IF(ISTRT.LT.1) ISTRT = 1
                IF(ISTP.GT.31) ISTP  = 31
                IF(JSTRT.LT.1) JSTRT = 1
                IF(JSTP.GT.31) JSTP  = 31
C
C    ** CALCULATE NEW CONCENTRATIONS FOR EACH CHI GRID POINT
C    **   INFLUENCED BY THE CURRENT PUFF
C    **  PCHI HAS UNITS OF CURIE * HOUR / M**3
C
                DO 310 I = ISTRT, ISTP
                  DO 300 J = JSTRT, JSTP
                    RSQ = (((XP(M)*2.0) - XS(I))**2) +
     1                    (((YP(M)*2.0) - YS(J))**2)
                    IF(RSQ.GT.(RP*RP))  GOTO 300
```

```
                  PCHI = PUFCHI * EXP(HSGSQ * RSQ)
                  CHI(I,J) = CHI(I,J) + PCHI
  300           CONTINUE
  310         CONTINUE
  320       CONTINUE
  330     CONTINUE
          CALL ARRTIM(CHIMIN,TINC,PREVDY,PREVHR)
C    **  TESING MODE OUTPUT SECTION
          IF(TITLE(1).NE.'*')  GOTO 340
          IF(IADV.NE.1)  GOTO 334
           MHR = PREVHR + RELHR + 1
           WRITE(3,332) MHR
  332      FORMAT(1H1,///'RELEASE HOUR',I4,///T14,'ADV STP',T27,'PUFF',
     +     T35,'GRID X',T45,'GRID Y',T55,'MOVE X',T65,'MOVE Y',
     +     T76,'TOTAL DIST',T94,'SIGMA Y',T104,'SIGMA Z'///)
C
  334     DO 338 MM = 1, IPUFFS
           WRITE(3,336) IADV,MM,XP(MM),YP(MM),DXS(MM),DYS(MM),STOTAL(MM),
     +     SIGMAY(MM),SIGMAZ(MM)
  336      FORMAT(1H ,T17,I4,T27,I4,T35,F6.2,T45,F6.2,T54,F7.4,T64,F7.4,
     +      T78,F8.1,T94,F7.1,T104,F7.1)
  338    CONTINUE
         WRITE(3,339)
  339    FORMAT(//)
C
  340   CONTINUE
C
        DO 650 I = 1, 16
          DO 640 J = 1, 16
            U(I,J) = UU(I,J)
            V(I,J) = VV(I,J)
  640     CONTINUE
  650   CONTINUE
        PREVDY = DDAY
        PREVHR = DHR
```

```
C
C    COUNT THE ACTIVE PUFFS AND BLANK THE OPTION SELECT ARRAY
          IPSUM = 0
          DO 670 M = 1, IPUFFS
            IPSUM = IPSUM + MF(M)
  670     CONTINUE
          DO 680 K=1,4
            IOPT(K) = ' '
  680     CONTINUE
C
          WRITE(10,685) SHR,DDAY,DHR,IPSUM
  685     FORMAT(/5X,'END OF SIMULATION HOUR ',I3,5X,
     +      'DATA = DAY ',I3,' HOUR ',I2,' <PST>',/7X,I3,' PUFFS ACTIVE',/)
C
C    **   DUMP CHI MATRIX TO FILE FOR LATER USE IN THE
C    **     POINT EXPOSURE PLOTS
C
        WRITE BINARY(4) DHR, CHI
C
          WRITE(10,690)
  690     FORMAT(/5X,'SELECT OUTPUT OPTIONS')
          IF(SHR.EQ.1)  WRITE(10,695)
  695     FORMAT(/5X,'S = SCREEN      L = LISTING      P = PLOT',
     +                '      T = TERMINATE      BLANK = NO OUTPUT'/)
          READ(11,700) (IOPT(K),K=1,4)
  700     FORMAT(4S1)
          DO 800 K = 1, 4
            IF(IOPT(K).EQ.' ')  GOTO 800
            IF(IOPT(K).NE.'S')  GOTO 740
              CALL OVLOD(9,SBS,1,IER)
              IF(IER.NE.1) TYPE'SBS LOAD ERROR - IER = ',IER
              CALL SCREEN(IPSUM,DELXY)
              PAUSE 'HIT RETURN TO CONTINUE'
              GOTO 800
```

```
C
  740     IF(IOPT(K).NE.'L')  GOTO 760
             CALL OVLOD(9,SR6,1,IER)
             IF(IER.NE.1) TYPE 'SR6 LOAD ERROR - IER = ',IER
             CALL PRINTE(IPSUM,DELXY)
              WRITE(10,750)
  750         FORMAT(/5X,'OUTPUT SENT TO PRINT FILE')
             GOTO 800
C
  760      IF(IOPT(K).NE.'P')  GOTO 780
             CALL OVLOD(9,SR7,1,IER)
             IF(IER.NE.1) TYPE 'SR7 LOAD ERROR - IER = ',IER
             CALL PUFPLT(IPSUM,DHR,TITLE,XP,YP,DELXY)
C
  780      IF(IOPT(K).EQ.'T')  GOTO 390
  800    CONTINUE
C
  100 CONTINUE
C
  390 WRITE(10,400)
      WRITE(3,400)
  400 FORMAT(/5X,'END OF SIMULATION',///)
      CALL OVLOD(9,SRR,1,IER)
      IF(IER.NE.1) TYPE 'ERROR LOADING OVERLAY SRR - IER = ',IER
      CALL EXPSUM(TITLE,CHIMIN)
      ENDFILE 4
      CALL CLOSE(4,IER)
      CALL CLOSE(1,IER)
      CALL CLOSE(2,IER)
      WRITE(10,410)
  410 FORMAT(/5X,'CHI MATRIX FILE CLOSED AND AVAILABLE FOR EXPLT')
      GOTO 515
C
  500 WRITE(10,510) SHR
  510 FORMAT(5X,'EOF ENCOUNTERED IN DATA DURING HOUR ',I2,
     1    ' OF THE SIMULATION')
```

```
  515 WRITE(10,520)
  520 FORMAT(5X,'DO YOU WISH TO RESTART? Y OR N')
      READ(11,530,END=515,ERR=515) SELECT
  530 FORMAT(S1)
      IF(SELECT.NE.'Y')  GOTO 999
      WRITE(3,538)
  538 FORMAT(/5X,'***  RESTART REQUESTED  ***',/)
C
  535 WRITE(10,540)
  540 FORMAT(5X,'DO YOU WISH TO REINITIALIZE THE GRID? Y OR N')
      READ(11,530,END=535,ERR=535) SELECT
      IF(SELECT.EQ.'Y')  GOTO 10
      WRITE(3,550)
  550 FORMAT(/5X,'***  NO GRID REINITIALIZATION  ***',/)
      GOTO 20
C
  600 WRITE(10,610) SHR
  610 FORMAT(5X,'READ ERROR IN DATA DURING HOUR ',I2,
     1   ' OF THE SIMULATION --> STOP')
  999 CALL CLOSE(3,IER)
      STOP
      END
```

GRIDIN

The GRIDIN subroutine initializes the wind grid and reads in the station data. MESOI is currently configured to accept wind data from thirty locations. The grid spacing and station names, locations, status and elevation are input from the file STATIONS (see Appendix C). GRIDIN also reads in the arrays STNUM and STDIST. These arrays contain the information to be used in interpolating winds to each grid point.

Subroutine GRIDIN

```
      OVERLAY S61
C    ** GRIDIN  **  MESOI VERSION 1.1
C
C    ** INITIALIZATION OF THE GRID AND SET UP OF STATIONS
C
      SUBROUTINE GRIDIN(DELXY,NUMSTA)
C
      REAL NAMST(30)
      REAL XSTA(30),YSTA(30)
      REAL XDIST(30), YDIST(30)
      REAL STDIST(10,16,16), USTA(30), VSTA(30)
C
      INTEGER STATUS(30), ACTSTA,STANUM(30), SEL, STNUM(10,16,16)
C
      COMMON/STATN/XSTA,YSTA,NAMST,STATUS,RCH,ACTSTA,USTA,VSTA,XDIST,YDIST
      COMMON/WINDS/STDIST,STNUM
C
      DATA RCH/5.0/
      NCOLS = 16
      NROWS = 16
C
      DELXY = 5000.0
      WRITE(10,100)
      WRITE(3,100)
  100 FORMAT(/5X,'MESOI ---> GRID INITIALIZATION',/)
      WRITE(10,110) NCOLS, NROWS
  110 FORMAT(5X,'THE CURRENT WIND GRID IS:',/,8X,I3,' ROWS',5X,
     1  I3,' COLUMNS',//)
C
      WRITE(10,310)
  310 FORMAT(/5X,'MESOI ---> STATION INITIALIZATION',/)
      NUMSTA = 0
      NSTAT  = 0
      CALL OPEN(8,'STATIONS.',1,IER)
```

```
      IF(IER.EQ.1)  GOTO 330
        WRITE(10,320) IER
320     FORMAT(/5X,'ERROR OPENING STATIONS FILE= IER= ',I3)
        STOP
C
330   READ(8,340) DELXY
340   FORMAT(1X,F8.2)
      WRITE(10,350) DELXY
      WRITE(3,350) DELXY
350   FORMAT(8X,'DELXY = ',F8.1,' METERS'/)
      DO 400 I = 1, 30
        READ(8,370) STNAM,XD,YD,ISTAT
370     FORMAT(1X,S4,2(4X,F6.2),4X,I1)
        IF(STNAM.EQ.'XXXX')  GOTO 400
          XSTA(I) = (XD * 1000. / DELXY) + 7.5
          YSTA(I) = (YD * 1000. / DELXY) + 7.5
          STATUS(I) = ISTAT
          NAMST(I) = STNAM
          NUMSTA = NUMSTA + 1
          IF(ISTAT.EQ.1)  NSTAT = NSTAT + 1
400     CONTINUE
C
      READ BINARY(8) STDIST
      READ BINARY(8) STNUM
      CALL CLOSE(8,IER)
C
C
      WRITE(10,440) NUMSTA, NSTAT
440   FORMAT(5X,'THERE ARE CURRENTLY ',I2,' STATIONS WITH ',
     +    I2,' DISABLED'//)
      WRITE(10,450)
450   FORMAT(6X,'STA NAME',4X,'GRIDX',3X,'GRIDY',4X,'STATUS')
C
```

```
      ACTSTA = NUMSTA - NSTAT
      DO 470 I = 1, NUMSTA
        WRITE(10,460) I,NAMST(I),XSTA(I),YSTA(I),STATUS(I)
  460   FORMAT(6X,I2,2X,S4,3X,F6.2,2X,F6.2,6X,I1)
  470 CONTINUE
C
      DO 610 I = 1, 30
        USTA(I) = 0.0
        VSTA(I) = 0.0
  610 CONTINUE
C
      WRITE(10,700)
  700 FORMAT(/5X,'**  END GRID INITIALIZATION **'/)
      PAUSE 'HIT RETURN TO CONTINUE'
C
C   WRITE COMPLETED GRID INFORMATION TO OUTPUT FILE
      WRITE(3,110) NCOLS,NROWS
      WRITE(3,440) NUMSTA, NSTAT
      WRITE(3,450)
      DO 750 I = 1, NUMSTA
        WRITE(3,460) I,NAMST(I),XSTA(I),YSTA(I),STATUS(I)
  750 CONTINUE
      WRITE(3,700)
      RETURN
      END
```

ARRIVN

Subroutine ARRIVN sets up the exposure checkpoints and initializes the array ARRIVL(11,35). Eleven pieces of information are stored for each of up to 35 points:

1) X coordinate (km from grid center)
2) Y coordinate (km from grid center)
3) status flag; set if thresholds have been exceeded
4) first 4 characters of name
5) second 4 characters of name
6) day
7) hour — when first threshold exceeded
8) minute
9) day
10) hour — when second threshold exceeded
11) minute

Up to 35 checkpoints may be predefined in the file CHKPNT. The file consists of single line records each containing three elements:

1) X distance in kilometers from grid center
2) Y distance in kilometers from grid center
3) name of checkpoint -- up to 8 characters

The record format for reading is: (1X,F6.1,1X,F6.1,2S4).

The subroutine reads each record in turn and checks to see if the defined point is within the current grid. If it is, then the first three elements of the ARRIVL array are set. Input stops when an EOF is encountered in CHKPNT or when all 35 available checkpoints have been defined.

Subroutine ARRIVN

```
      OVERLAY SR0
C     **  ARRIVN  **  MESOI VERSION 1.1
C
C     **  MODULE WHICH SETS UP THE ARRIVAL CHECKPOINTS
C     **    THE ARRAY 'ARRIVL' IS INITIALIZED WITH THE FOLLOWING
C     **     INFORMATION:
C
C     **     1) X DISTANCE IN KILOMETERS FROM GRID CENTER
C     **     2) Y DISTANCE IN KILOMETERS FROM GRID CENTER
C     **     3) A STATUS FLAG; 0=INACTIVE,1=ACTIVE,2=>CHIMIN,3=>CHIMINES
C     **     4) FIRST 4 CHARACTERS OF THE CHECKPOINT NAME
C     **     5) LAST 4 CHARACTERS OF NAME
C     **     6) DAY, HOUR AND MINUTE WHEN CHIMIN WAS EXCEEDED
C     **     7) DAY, HOUR AND MINUTE WHEN CHIMIN*ES WAS EXCEEDED
C
      SUBROUTINE ARRIVN(DELXY)
C
      REAL ARRIVL(11,35), CPNAME(2)
C
      COMMON/ARR/ARRIVL
      COMMON/AXTRA/CPNAME
C
      WRITE(10,50)
      WRITE(3,50)
   50 FORMAT(/5X,'MESOI --> SET UP ARRIVAL CHECKPOINTS FROM ',
     +      'CHKPNT FILE')
C
      DO 120 J=1,35
        ARRIVL(3,J) = 0.0
        DO 110 I=6,11
          ARRIVL(I,J) = 0.0
  110   CONTINUE
  120 CONTINUE
C
```

```
      CALL OPEN(8,'CHKPNT.',1,IER)
      IF(IER.EQ.1)  GOTO 140
        WRITE(10,130) IER
  130   FORMAT(/5X,'ERROR OPENING CHKPNT FILE - CODE = ',I3)
        RETURN
C
  140 DO 200 J = 1, 35
  160   READ(8,180,END=300) X, Y, CPNAME
  180   FORMAT(1X,F6.1,1X,F6.1,1X,2S4)
        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 160
        IF(Y.LT.1.OR.Y.GT.31)  GOTO 160
        ARRIVL(1,J) = X
        ARRIVL(2,J) = Y
        ARRIVL(3,J) = 1.0
        ARRIVL(4,J) = CPNAME(1)
        ARRIVL(5,J) = CPNAME(2)
  200 CONTINUE
C
  300 CALL CLOSE(8,IER)
      NCP = 0
      DO 600 J=1,35
        NCP = NCP + ARRIVL(3,J)
  600 CONTINUE
      WRITE(10,610) NCP
      WRITE(3,610) NCP
  610 FORMAT(/5X,I2,' CHECKPOINTS ACTIVE ON THE CURRENT GRID'/)
      RETURN
      END
```

Subroutine INIT

```
      OVERLAY SR2
C    **  INIT  ** MESOI VERSION 1.1
C
C    **  PRIMARY INITIALIZATIONS OF THE MODEL, INCLUDING:
C    **    IDENTIFICATION TITLE, CURRENT DATE AND TIME,
C    **    AND BASIC SIMULATION PARAMETERS
C
      SUBROUTINE INIT(FAC,ADT,DELXY,XS,YS,NUMSTA,NXTDAY,METDAT)
C
      REAL CHI(32,32), XS(31),YS(31)
C
      INTEGER IBLK(3),RDATE(3), RTIME(3)
      INTEGER PLTID
      INTEGER  SDAY,SHR,NPH,DT,NSI,DDAY,DHR,STHR,STDAY,DFLAG,STAR
      INTEGER TITLE(50), METDAT(30)
C
C
      COMMON/MASTER/RDATE,RTIME,TITLE,NPH,DT,CHIMIN,DDAY,DHR,NSI,
     1   ACNST,STDAY,STHR
      COMMON/PLT/PLTID,RP
      COMMON/GRID/CHI
C
      WRITE(10,90)
   90 FORMAT(/5X,'MESOI ---> PRIMARY INITIALIZATION',/)
C
  100 WRITE(10,110)
  110 FORMAT(/5X,'ENTER RUN IDENTIFICATION TITLE OF UP TO',
     1   ' 50 CHARACTERS')
      READ(11,120,END=100,ERR=100) TITLE
  120 FORMAT(50S1)
C
  130 WRITE(10,140)
  140 FORMAT(/5X,'ENTER A 2 CHARACTER PLOT ID')
      READ(11,150,ERR=130,END=130) PLTID
  150 FORMAT(S2)
```

INIT continued

```
C
C    **  SET MODEL PARAMETERS:
C    **    NPH ----- NUMBER OF PUFFS PER HOUR
C    **    DT ------ NUMBER OF MINUTES BETWEEN PUFF SAMPLINGS
C    **    CHIMIN -- INITIAL MINIMUM SIGNIFICANT EXPOSURE (CURIE HR/M**3)
C    **    NSI ----- NUMBER OF TIMES PUFF IS SAMPLED BETWEEN RELEASES
C
      NPH = 6
       DT = 2
      CHIMIN = 1.0E-15
C    **  USER SPECIFIES CHIMIN IF IN TESTING MODE
C
      IF(TITLE(1).NE.'*')  GOTO 190
        WRITE(10,160)
  160   FORMAT(/5X,'SPECIFY BASIC MAGNITUDE FOR CHIMIN ',
     +      ' -- USE E FORMAT')
        READ(11) CHIMIN
        WRITE(10,170) CHIMIN
        WRITE(3,170) CHIMIN
  170   FORMAT(/5X,'TESTING - BASIC CHIMIN SET AT ',E10.3)
  190 CHIMIN = CHIMIN * (DT / 60.)
      NSI = 60 / (DT * NPH)
C
      IF((MOD(60,NPH)).NE.0)  GOTO 800
      IF(DT.GT.(60/NPH))  GOTO 800
      IF(MOD((60/NPH),DT).NE.0)  GOTO 800
C
      DO 200 I = 1, 31

      DO 200 J = 1, 31
      XS(I) = I - 1
        CHI(I,J) = 0.0
        YS(J) = J - 1
200 CONTINUE
```

```
C
      ADT = 1.0 / NPH
      FAC = ADT * 3600 / DELXY
      ACNST = 2.0 * (DT / 60.0) / (6.283185 ** 1.5)
      CALL OPEN(1,'OBSDAT.',1,IER)
      IF(IER.EQ.1)  GOTO 258
        WRITE(10,256) IER
  256   FORMAT(/5X,'ERROR OPENING OBSV DATA FILE - CODE = ',I3)
        STOP
C
  258 CALL OPEN(2,'FORDAT.',1,IER)
      IF(IER.EQ.1)  GOTO 205
        WRITE(10,259) IER
  259   FORMAT(/5X,'ERROR OPENING FORECAST DATA FILE - CODE = ',I3)
        STOP
C
  205 WRITE(10,210)
  210 FORMAT(/5X,'ENTER DATE FOR START OF SIMULATION --> MMDDYY')
      READ(11,215,END=205,ERR=205) IMO, IDY, IYR
  215 FORMAT(10I2)
      CALL JULIAN(IYR,IMO,IDY,STDAY)
C
      IF(IMO.LT.1.OR.IMO.GT.12.OR.IDY.LT.1.OR.IDY.GT.31)  GOTO 219
        WRITE(10,225) STDAY, IYR
  225   FORMAT(/5X,'JULIAN DATE = ',I3,4X,'19',I2,/)
        GOTO 230
  219   WRITE(10,220) IMO, IDY
  220   FORMAT(5X,'A DATE OF ',2I2,' IS IMPOSSIBLE ',
     1     '--> TRY AGAIN',//)
        GOTO 205
C
  230 WRITE(10,240)
  240 FORMAT(/5X,'ENTER HOUR <PST> FOR START OF SIMULATION')
      READ(11,END=230,ERR=230) STHR
      IF(STHR.GE.1.AND.STHR.LE.24)  GOTO 255
        WRITE(10,250) STHR
```

```
  250   FORMAT(5X,'A STARTING HOUR OF ',I3,' IS IMPOSSIBLE ',
     1     '--> TRY AGAIN',//)
        GOTO 230
C
  255 DFLAG = 0
      CALL LOCA1(IYR,STDAY,STHR,LERROR,DFLAG)
      IF(LERROR.EQ.1)  GOTO 205
C
      READ(1,260,END=900,ERR=900) IYR,STDAY,STHR,STAB,LDEPTH,
     +  (METDAT(I),I=1,10)
  260 FORMAT(I2,I3,I2,2X,I1,1X,I3,5X,10(1X,I5))
      LDEPTH = LDEPTH * 10
      READ(1,270,END=900,ERR=900) (METDAT(I),I=11,20)
      READ(1,270,END=900,ERR=900) (METDAT(I),I=21,30)
  270 FORMAT(19X,10(1X,I5))
      CALL CHSAV(1,IBLK(1))
      READ(1,272,END=900,ERR=900) NXTDAY
  272 FORMAT(2X,I3)
      CALL CHRST(1,IBLK(1))
C   **  LIST FIRST DATA RECORDS ON OUTPUT FILE
      WRITE(3,280) IYR, STDAY, STHR
  280 FORMAT(5X,'DATA FOR DAY: ',I2,1X,I3,'  HOUR ',I2,' <PST>'/)
      WRITE(3,285) STAB, LDEPTH
  285 FORMAT(8X,'STABILITY = ',I1,3X,'MIXING DEPTH = ',I4,' METERS',/)
      WRITE(3,290) (METDAT(I),I=1,10)
  290 FORMAT(8X,'WINDS',8X,10(1X,I4))
      WRITE(3,295) (METDAT(I),I=11,20)
      WRITE(3,295) (METDAT(I),I=21,30)
  295 FORMAT(21X,10(1X,I4))
      DDAY = STDAY
      DHR  = STHR
      CALL DFILW('CHIDMP',IER)
      CALL OPEN(4,'CHIDMP.',2,IER)
      IF(IER.NE.1) TYPE 'ERROR OPENING CHIDMP FILE - IER = ',IER
```

```
C
      PAUSE 'HIT RETURN TO CONTINUE'
      WRITE(3,300) TITLE,RDATE,RTIME
  300 FORMAT(//5X,50A1,5X,3I2,2X,3I2)
      WRITE(3,310) STHR, IMO, IDY
  310 FORMAT(/5X,'STMULATION STARTS AT HOUR ',I2,' <PST> ON DAY ',
     +  I2,':',I2,//)
      RETURN
C
  800 WRITE(10,810)
  810 FORMAT(5X,'INCORRECT RELATIONSHIP OR BAD VALUES FOR ',
     1  'VARIABLE NPH AND/OR DT',/,7X,'CHECK SPECIFICATIONS')
      STOP
C
  900 WRITE(10,910)
  910 FORMAT(/5X,'EOF OR ERROR DURING FIRST DATA READ')
      STOP
      END
```

Subroutine LOCATE

```
C    ** LOCATE  **  MESOI VERSION 1.1
C
C    ** 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.
C
      SUBROUTINE LOCA1(SYR,SDAY, SHR, LERROR,DFLAG)
C
      INTEGER IBLK(3), JBLK(3)
      INTEGER SYR,SDAY, SHR, DAY, HOUR, RECNO, DFLAG, YR
C
      IF(DFLAG.EQ.1)  GOTO 300
      WRITE(10,100)
      WRITE(5,100)
  100 FORMAT(/5X,'METEOROLOGICAL DATA FILE SEARCH -- ',/)
C
  105 CALL CHSAV(1,IBLK(1))
      READ(1,110,END=600,ERR=800) YR,DAY, HOUR, RECNO
  110 FORMAT(I2,I3,I2,I1)
      IF(YR.NE.SYR)  GOTO 620
      IF(DAY.EQ.888)  GOTO 600
      IF(RECNO.NE.1)  GOTO 820
      IF(DAY.GT.SDAY.AND.SDAY.GT.2) GOTO 620
      IF(DAY.EQ.SDAY) GOTO 120
      READ(1,1) IDUM
      READ(1,1) IDUM
    1 FORMAT(A1)
      GOTO 105
C
C    **  LOCATE PROPER HOUR
C
  120 IF(HOUR.EQ.SHR)  GOTO 150
      READ(1,1) IDUM
      READ(1,1) IDUM
      GOTO 105
```

```
C
  150 WRITE(10,200) DAY, HOUR, RECNO
      WRITE(3,200) DAY, HOUR, RECNO
  200 FORMAT(7X,'OBSV FILE POSITIONED AT:   DAY ',I3,3X,' HOUR ',I2,
     1   3X,' RECORD ',I1/)
      CALL CHRST(1,IBLK(1))
C
      CALL CHSAV(2,JBLK(1))
      READ(2,110,END=640,ERR=840) YR,DAY,HOUR,RECNO
      CALL CHRST(2,JBLK(1))
      WRITE(10,210) DAY, HOUR, RECNO
      WRITE(3,210) DAY, HOUR, RECNO
  210 FORMAT(7X,'FORECAST FILE STARTS AT:   DAY ',I3,3X,
     +   ' HOUR ',I2,3X,' RECORD ',I1,/)
      DFLAG = 1
      LERROR = 0
      RETURN
C
  300 WRITE(10,310)
      WRITE(3,310)
  310 FORMAT(/5X,'FORECAST DATA FILE SEARCH -- ',/)
  315 READ(2,110,END=640,ERR=840)  YR, DAY, HOUR, RECNO
      IF(RECNO.NE.1)  GOTO 820
      IF(DAY.EQ.SDAY)  GOTO 320
      READ(2,1) IDUM
      READ(2,1) IDUM
      GOTO  315
C
  320 READ(2,1) IDUM
      READ(2,1) IDUM
      IF(HOUR.NE.SHR)  GOTO 315
      CALL CHSAV(2,JBLK(1))
      READ(2,110,END=640,ERR=840) YR,DAY, HOUR, RECNO
      CALL CHRST(2,JBLK(1))
```

```
C
      WRITE(10,340) DAY, HOUR, RECNO
      WRITE(3,340) DAY, HOUR, RECNO
  340 FORMAT(7X,'FORECAST DATA FILE POSITIONED AT:   DAY ',I3,
     +  3X,' HOUR ',I2,3X,' RECORD ',I1,/)
      LERROR = 0
      RETURN
C
C
  600 WRITE(10,610) SDAY, SHR
      WRITE(3,610) SDAY, SHR
  610 FORMAT(/5X,'EOF ENCOUNTERED IN OBSERVATION DATA FILE DURING  ',
     1'SEARCH',/,7X,'FOR DAY ',I3,' HOUR ',I2,/,7X,
     2'SIMULATION WITH SPECIFIED DATE AND TIME IS IMPOSSIBLE',//)
      LERROR = 1
      REWIND 1
      REWIND 2
      RETURN
C
  620 WRITE(10,630) YR, DAY, SYR, SDAY
      WRITE(3,630) YR, DAY, SYR,SDAY
  630 FORMAT(/5X,'DATA FILE STARTS ON DAY ',I2,I3,5X,
     1'CANNOT SIMULATE DAY ',I2,I3,//)
      LERROR = 1
      REWIND 1
      REWIND 2
      RETURN
C
  640 WRITE(10,650)
      WRITE(3,650)
  650 FORMAT(/5X,'EOF ENCOUNTERED IN FORECAST DATA FILE DURING SKIP',
     1' TO STARTING POSITION',/,7X,'FILE DOES NOT MATCH OBS FILE',
     2  ' --> STOP',//)
      STOP
```

```
C
  800 WRITE(10,810)
      WRITE(3,810)
  810 FORMAT(/5X,'READ ERROR IN OBSERVATION DATA FILE --> STOP',//)
      STOP
  820 WRITE(10,830) DAY, HOUR, RECNO
      WRITE(3,830) DAY, HOUR, RECNO
  830 FORMAT(/5X,'INCOMPLETE DATA RECORD --> STOP ',/,
     17X,'DAY ',I3,'  HOUR ',I2,'  RECORD ',I1)
      STOP
  840 WRITE(10,850)
      WRITE(3,850)
  850 FORMAT(/5X,'READ ERROR IN FORECAST DATA FILE -->',
     +    ' STOP',//)
      STOP
      END
```

```
      OVERLAY SR3
C    ** RELEAS
C
C    ** DEFINITION OF THE RELEASE PARAMETERS AND SOURCE LOCATION
C
      SUBROUTINE RELEAS(NPH,DELXY,STRTH,STRTD)
C
      REAL XSOURC, YSOURC,Q
C
      INTEGER RELDAY, STRTD, RELHR, RELMIN, PARTHR,SELECT,
     1  DURHR, DURMIN, MAXPUF, STRTH
C
      COMMON/REL/XSOURC,YSOURC,RELDAY,RELHR,
     1  PARTHR,MAXPUF,Q
C
C
      WRITE(10,110)
  110 FORMAT(/5X,'MESO ---> RELEASE INITIALIZATION'/)
  115 WRITE(10,120)
  120 FORMAT(/5X,'SPECIFY COORDINATES (X,Y) OF SOURCE',
     +  ' IN KILOMETERS FROM HMS',/)
C
      READ(11,END=115,ERR=115) XDIST, YDIST
      XSOURC = XDIST * 1000. / DELXY + 7.5
      YSOURC = YDIST * 1000. / DELXY + 7.5
C
      IF(XSOURC.LT.0.0.OR.XSOURC.GT.15.0)  GOTO 800
      IF(YSOURC.LT.0.0.OR.YSOURC.GT.15.0)  GOTO 800
C
  140 WRITE(10,150)
  150 FORMAT(/5X,'ENTER DATE OF RELEASE ---> MMDDYY')
      READ(11,155,END=140,ERR=140) IMO, IDY, IYR
  155 FORMAT(1012)
```

```
C
      IF(IMO.LT.1.OR.IMO.GT.12)  GOTO 820
      IF(IDY.LT.1.OR.IDY.GT.31)  GOTO 820
      CALL JULTAN(IYR,IMO,IDY,RELDAY)
C
      IF(RELDAY.GT.(STRTD+2))  GOTO 840
      IF(RELDAY.LT.STRTD.AND.RELDAY.GT.2)  GOTO 840
C
  175 WRITE(10,180)
  180 FORMAT(/5X,'TIME OF RELEASE?  HOURS,MINUTES <PST> ')
      READ(11,END=175,ERR=175)  RELHR, RELMIN
      IF(RELHR.LT.1.OR.RELHR.GT.24)  GOTO 860
      IF(RELHR.LT.STRTH.AND.RELDAY.EQ.STRTD)  GOTO 840
      IF(RELMIN.LT.0.OR.RELMIN.GT.59)  GOTO 880
      PARTHR = RELMIN / (60/NPH) + 1
C
  195 WRITE(10,200)
  200 FORMAT(/5X,'DURATION OF RELEASE?  HOURS,MINUTES',/,
     1    7X,'IF CONTINUOUS,  ENTER 99,99 ')
      READ(11,END=230,ERR=230) DURHR, DURMIN
      IF(DURHR.EQ.99.AND.DURMIN.EQ.99)  GOTO 230
C
      IF(DURHR.LT.0.OR.DURMIN.LT.0)  GOTO 900
      MAXPUF = (DURHR * NPH) + (DURMIN / (60/NPH))
      WRITE(10,220) MAXPUF
  220 FORMAT(5X,I4,' PUFFS WILL BE RELEASED')
      GOTO 300
  230 DURHR = 0
      DURMIN = 0
      MAXPUF = 1000
      WRITE(10,240) NPH
  240 FORMAT(5X,'RELEASE WILL BE CONTINUOUS AT ',I3,
     1    ' PUFFS PER HOUR')
```

```
C
C
C   **  DEFAULT SOURCE TERM = 1.0 CURIE PER HOUR
C
  300 Q = 1.0 / NPH
C
      WRITE(3,400) XSOURC, YSOURC
      WRITE(10,400) XSOURC, YSOURC
  400 FORMAT(/5X,'SOURCE IS LOCATED AT WIND GRID ',F4.1,1X,F4.1)
      WRITE(3,410) RELHR,RELMIN, IMO, IDY, MAXPUF, Q
      WRITE(10,410) RELHR,RELMIN,IMO,IDY,MAXPUF,Q
  410 FORMAT(/5X,'RELEASE WILL OCCUR AT ',I2,':',I2,' <PST> ON ',
     +  I2,'/',I2,/,7X,I4,' PUFFS AT ',F10.3,' CURIES PER PUFF',//)
      RETURN
C
  800 WRITE(10,810) XSOURC, YSOURC
  810 FORMAT(/5X,'SOURCE IS OFF GRID --  X = ',F10.2,5X,'Y = ',
     1  F10.2,//)
      GOTO 115
  820 WRITE(10,830) IMO, IDY
  830 FORMAT(/5X,'A DATE OF ',2I2,' IS NOT VALID')
      GOTO 140
  840 WRITE(10,850) STRTD,STRTH, RELDAY,RELHR
  850 FORMAT(/5X,'DATA STARTS ON JDAY ',I3,' AT HOUR ',I2,/,5X,
     1  'A RELEASE ON JDAY ',I3,' AT HOUR ',I2,' CANNOT BE SIMULATED')
      GOTO 140
  860 WRITE(10,870) RELHR
  870 FORMAT(/5X,'SPECIFIED HOUR OF RELEASE IS NOT POSSIBLE --> ',
     1    I5)
      GOTO 175
  880 WRITE(10,890) RELMIN
  890 FORMAT(/5X,'SPECIFIED MINUTES OF RELEASE NOT POSSIBLE --> ',
     1    I5)
      GOTO 175
  900 WRITE(10,910) DURHR, DURMIN
  910 FORMAT(/5X,'NEGATIVE VALUES OF DURATION NOT ALLOWED --> ',2I5,/)
      GOTO 195
      END
```

```
      OVERLAY S09
C     ** WIND  **  MESOI VERSION 1.1
C
C     ** PROGRAM TO DECODE WIND DATA ELEMENTS INTO DIRECTION
C     **   AND SPEEDS THEN INTERPOLATE TO EACH GRID LOCATION
C
      SUBROUTINE WIND(JDAY,HOUR,IDATA,UG,VG,NUMSTA,ITITL)
C
      REAL NAMST(30)
      REAL DIR(30), SPD(30), USTA(30), VSTA(30), UG(16,16),
     1   VG(16,16),XSTA(30),YSTA(30)
      REAL XDIST(30), YDIST(30),STDIST(10,16,16)
C
      INTEGER IDATA(30), STATUS(30), LFLAG(30)
      INTEGER STNUM(10,16,16),JDAY,HOUR,ACTSTA
C
      COMMON/WXTRA/LFLAG,DIR,SPD
      COMMON/STATN/XSTA,YSTA,NAMST,STATUS,RCH,ACTSTA,USTA,VSTA,XDIST,YDIST
      COMMON/WINDS/STDIST,STNUM
C
C     ** DECODE WIND DIRECTION AND SPEED FOR EACH ACTIVE STATION
C     **   CHECK FOR MISSING DATA (999'S)
C
      CONV = 0.44704
      M = 0
      DO 100 J = 1, NUMSTA
        IF(STATUS(J).EQ.1)  GOTO 100
        M = M + 1
        DIR(M) = IDATA(J) / 100
        SPD(M) = IDATA(J) - DIR(M) * 100
        DIR(M) = DIR(M) * 10
C
        IF(DIR(M).EQ.990.AND.SPD(M).EQ.99)  GOTO 90
        IF(DIR(M).EQ.0.AND.SPD(M).EQ.0)     GOTO 90
C
```

```
          LFLAG(M) = 0
C
          IF(DIR(M).LT.0.0.OR.DIR(M).GT.360.0)
     1        WRITE(10,500) J, DIR(M), JDAY, HOUR
          IF(SPD(M).LT.0.0.OR.SPD(M).GT.45.0)
     1        WRITE(10,510) J, SPD(M), JDAY, HOUR
C
C    **  CALCULATE U AND V COMPONENTS AND CONVERT TO METERS/SEC
C
          ANG = (270. - DIR(M)) * (3.1415927 / 180.)
          IF(ITITL.EQ.'*')  CONV = 1.0
          USTA(M) = (SPD(M) * COS(ANG)) * CONV
          VSTA(M) = (SPD(M) * SIN(ANG)) * CONV
          GOTO 100
C
   90     LFLAG(M) = 1
          IF(M.GE.ACTSTA)  GOTO 200
  100  CONTINUE
C
  500 FORMAT(5X,'WARNING -- STATION NUMBER ',I2,/,7X,
     1    'DIRECTION OUT OF NORMAL RANGE --> ',F6.1,/,
     2    7X,'DAY ',I3,2X,'HOUR ',I2,//)
  510 FORMAT(5X,'WARNING -- STATION NUMBER ',I2,/,7X,
     1    'SPEED OUT OF NORMAL RANGE --> ',F6.1,/,7X,
     2    'DAY ',I3,2X,'HOUR ',I2,//)
C
C    ** WIND INTERPOLATION SECTION
C
  200 DO 280 I = 1, 16
          DO 270 J = 1, 16
            SNU = 0.0
            SNV = 0.0
            SND = 0.0
             NS = 0
            LEND = MIN0(ACTSTA,10)
C
```

```
            DO 220 L = 1, LEND
              LS = STNUM(L,I,J)
              IF(LS.LT.1.OR.LS.GT.30)
     +          TYPE 'STNUM OUT OF RANGE ',LS
              IF(LFLAG(LS).EQ.1)  GOTO 220
              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
C
  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.LT.3.AND.ITITL.NE.'#')  WRITE(10,240) NS, I, J, JDAY, HOUR
  240         FORMAT(5X,'WARNING -- ONLY ',I2,' STATIONS AT ',
     1          'WIND GRID POINT ',I2,',',I2,/,7X,'DAY ',I3,
     2          5X,'HOUR ',I2,//)
            GOTO 260
C
  250       UG(I,J) = USTA(LS)
            VG(I,J) = VSTA(LS)
            GOTO 270
  260       UG(I,J) = SNU / SND
            VG(I,J) = SNV / SND
  270     CONTINUE
  280   CONTINUE
C
        RETURN
        END
```

Subroutine PUFFR

```
      OVERLAY SR4
C    ** PUFFR  ** MESOI VERSION 1.1
C
C    ** PUFF RELEASE SUBROUTINE  SPECIFIES INITIAL ATTRIBUTES
C
      SUBROUTINE PUFFR(ACNST)
C
      REAL XP(72), YP(72), STOTAL(72), QP(72), SIGMAZ(72),
     1   SIGMAY(72), DXS(72), DYS(72)
      REAL XSOURC,YSOURC,QSOURC
C
      INTEGER MF(72),TPUFFS,NPUFFS
      INTEGER RELDAY,RELTIM,RELHR,PARTHR,MAXPUF
C
      COMMON/PUFFS/MF,SIGMAY,SIGMAZ,QP,STOTAL,XP,YP,TPUFFS,
     1  NPUFFS,DXS,DYS,FAC,ADT
      COMMON/REL/XSOURC,YSOURC,RELDAY,RELHR,PARTHR,
     1   MAXPUF,QSOURC
C
      NPUFFS = NPUFFS + 1
      TPUFFS = TPUFFS + 1
      MF(TPUFFS) = 1
      XP(TPUFFS) = XSOURC
      YP(TPUFFS) = YSOURC
      QP(TPUFFS) = QSOURC * ACNST
      STOTAL(TPUFFS) = 1.0
      SIGMAZ(TPUFFS) = 0.1
      SIGMAY(TPUFFS) = 1.0
C
      RETURN
      END
```

```
C     ** PUFFM  **  MESOI VERSION 1.1
C
C     ** CALCULATES MOVEMENT OF THE PUFF
C
      SUBROUTINE PUFFM(M,DXM,DYM,TINC,MOFG)
C
      REAL XP(72),YP(72),STOTAL(72),SIGMAZ(72),
     1   SIGMAY(72),GP(72),DXS(72),DYS(72)
      REAL U(16,16),V(16,16),UU(16,16),
     1  VV(16,16), XG(2), YG(2), DX(2), DY(2)
C
      INTEGER MF(72),IPUFFS,NPUFFS
C
      COMMON/VECTOR/U,V,UU,VV
      COMMON/PUFFS/MF,SIGMAY,SIGMAZ,GP,STOTAL,XP,YP,IPUFFS,
     1   NPUFFS,DXS,DYS,FAC,ADT
C
      XG(1) = XP(M)
      YG(1) = YP(M)
C
      DO 100 K = 1, 2
        C2 = TINC + (FLOAT(K) - 1.0) * ADT
        C1 = 1.0 - C2
        I = XG(K) + 1.0
        IF(I.LT.1.OR.I.GE.16)  GOTO 110
        J = YG(K) + 1.0
        IF(J.LT.1.OR.J.GE.16)  GOTO 110
        RX = XG(K) - (I-1)
        RY = YG(K) - (J-1)
        RX1 = 1.0 - RX
        RY1 = 1.0 - RY
C
          CST = (RX1*RY1) * (C1*U(I,J) + C2*UU(I,J))
          CST = CST + (RY*RX1) * (C1*U(I,J+1) + C2*UU(I,J+1))
          CST = CST + (RX*RY) * (C1*U(I+1,J+1) + C2*UU(I+1,J+1))
          CST = CST + (RX*RY1) * (C1*U(I+1,J) + C2*UU(I+1,J))
        DX(K) = CST * FAC
```

```
C
          CST = (RX1*RY1) * (C1*V(I,J) + C2*VV(I,J))
          CST = CST + (RY*RX1) * (C1 * V(I,J+1) + C2*VV(I,J+1))
          CST = CS1 + (RX*RY) * (C1*V(I+1,J+1) + C2*VV(I+1,J+1))
          CST = CST + (RX*RY1) * (C1*V(I+1,J) + C2*VV(I+1,J))
        DY(K) = CST * FAC
C
        IF(K.EQ.2)  GOTO 100
        XG(K+1) = XG(K) + DX(K)
        YG(K+1) = YG(K) + DY(K)
  100 CONTINUE
C
      DXM = 0.5 * (DX(1) + DX(2))
      DYM = 0.5 * (DY(1) + DY(2))
      RETURN
C
  110 MOFG = 1
      RETURN
      END
```

Subroutine SIGMA

```
C    ** SIGMA  **  MESOI VERSION 1.1
C
C    ** CALCULATES VALUES FOR SIGMA Z AND SIGMA Y
C
      SUBROUTINE SIGMA(DSMTR,M,TPUFFS,STAB,LDEPTH,
     1  JN,SIGMAZ,SIGMAY,STOTLM)
C
      REAL CONST(6,4)
C
      INTEGER STAB, TPUFFS
C
      COMMON/CON/CONST,SYX
C
      DATA SYX/20000.0/
      DATA CONST/0.718,0.425,0.349,0.267,0.299,0.401,
     1           0.100,0.105,0.128,0.146,0.331,0.812,
     2           1.033,0.975,0.891,0.824,0.567,0.307,
     3           0.968,1.026,1.122,1.214,1.764,3.257/
C
      IF(JN.NE.1) GOTO 100
        A = CONST(STAB,1)
        A1 = A * SYX ** 0.35
        B = CONST(STAB,2)
        QG = CONST(STAB,3)
        QGI = CONST(STAB,4)
C
        CCHGPT = 0.335 * LDEPTH
        SZSMAX = 0.465 * LDEPTH
        SMAX = (SZSMAX/B) ** QGI
        SZMAX = 0.8 * LDEPTH
        IFLAG = 1
        IF(STOTLM.LT.SYX) IFLAG = 0
C
  100 STOTLM = STOTLM + DSMTR
```

SIGMA continued

```
C
C    ** CALCULATE SIGMA Z (IF NECESSARY) USING LINEAR OR POWER FORMULA
C
       IF(SIGMAZ.GE.SZMAX) GOTO 300
         IF(SIGMAZ.LT.SZSMAX) GOTO 230
           DELTA = (CCHGPT / SMAX) * DSMTR
           DELTA = AMIN1(DELTA,SZMAX-SIGMAZ)
         GOTO 280
  230     EFDIST = (SIGMAZ/B) ** QGI
           ENDIST = EFDIST + DSMTR
           IF(ENDIST.GT.SMAX)  GOTO 250
             HDIST = EFDIST + DSMTR / 2
             DELTA = B * QG * HDIST ** (QG-1) * DSMTR
           GOTO 280
  250        F1 = (SMAX - EFDIST) / DSMTR
             HDIST = (EFDIST + SMAX) / 2.0
             DELTA = (F1 * B * QG * HDIST ** (QG-1) +
     +                (1.0-F1) * (CCHGPT/SMAX)) * DSMTR
  280    SIGMAZ = SIGMAZ + DELTA
         IF(SIGMAZ.GT.SZMAX) SIGMAZ = SZMAX
C
C    ** CALCULATE SIGMA Y
C
  300  IF(SIGMAY.GE.1.0)  GOTO 320
         EFDIST = 0.0
         GOTO 340
  320    EFDIST = (SIGMAY / A) ** (1.0 /0.85)
  340    IF(EFDIST.GT.SYX)  GOTO 380
           IF(EFDIST+DSMTR.GT.SYX)  GOTO 360
         HDIST = EFDIST + (DSMTR / 2.0)
         SIGMAY = SIGMAY + (0.85 * A) * DSMTR / HDIST ** 0.15
         RETURN
```

```
C
  360         DS1 = (SYX - EFDIST) / DSMTR
              DS2 = 1.0 - DS1
              HDIST1 = EFDIST + DS1 * DSMTR
              HDIST2 = SYX + DS2 * DSMTR
              SIGMAY = SIGMAY + 0.85 * DS1 * DSMTR * A / HDIST1 ** 0.15
     +          + 0.5 * A1 * DS2 * DSMTR / HDIST2 ** 0.5
              RETURN
C
  380       EFDIST = (SIGMAY / A1) ** 2.0
            HDIST = EFDIST + DSMTR / 2
            SIGMAY = SIGMAY + 0.5 * A1 * DSMTR / HDIST ** 0.5
        RETURN
        END
```

ARRTIM

The ARRTIM subroutine is called from CNTRL each advection step to monitor the defined checkpoints. If the first threshold is exceeded, the Julian date, hour and minute are entered into elements 6, 7, and 8 of the ARRIVL array. Elements 9, 10, 11 are similarly filled when the second threshold is exceeded. A message is displayed on the interactive device and written into the output file noting that the threshold was exceeded.

```
C     **   ARRTIM  **  MESOI - VERSION 1.1
C
C     **   SUBROUTINE TO REPORT THE ARRIVAL OF TRANSPORTED
C     **     MATERIAL AT SPECIFIED CHECKPOINTS ON THE CHI GRID.
C     **     THE POINTS ARE CHECKED AT THE END OF EACH ADVECTION
C     **     STEP. A REPORT IS MADE WHEN CHIMIN AND 1E5*CHIMIN
C     **     ARE EXCEEDED.
C
        SUBROUTINE ARRTIM(CHIMIN,TINC,PREVDY,PREVHR)
C
        REAL ARRIVL(11,35), CHI(32,32)
        INTEGER PREVDY, PREVHR
C
        COMMON/GRID/CHI
        COMMON/ARR/ARRIVL
C
        MIN = TINC * 60
        DO 500 J = 1, 35
          IF(ARRIVL(3,J).EQ.0)  GOTO 500
          IF(ARRIVL(3,J).GE.3)  GOTO 500
          IF(ARRIVL(3,J).EQ.2)  GOTO 250
C
C     **   ROUND TO NEAREST INTEGER; THUS GRID POINT WILL BE AT MOST
C     **   0.25 X DELXY AWAY FROM CHECKPOINT.
C
          L = ARRIVL(1,J) + 0.5
          M = ARRIVL(2,J) + 0.5
          IF(CHI(L,M).LT.CHIMIN)  GOTO 500
C
```

```
            ARRIVL(3,J) = 2
            WRITE(10,100) PREVDY,PREVHR,MIN,ARRIVL(4,J),
     +                    ARRIVL(5,J), CHIMIN
  100       FORMAT(/1X,'DAY ',I3,1X,'AT ',I2,':',I2,' <PST>  EXPOSURE  AT ',
     +      2S4,' EXCEEDS ',1PE10.3,' CURIE HOUR/M**3',/)
            WRITE(3,100) PREVDY,PREVHR,MIN,ARRIVL(4,J),
     +                    ARRIVL(5,J), CHIMIN
            ARRIVL(6,J) = PREVDY
            ARRIVL(7,J) = PREVHR
            ARRIVL(8,J)  = MIN
            GOTO 500
C
  250       L = ARRIVL(1,J) + 0.5
            M = ARRIVL(2,J) + 0.5
            CHIMES = CHIMIN * 1E5
            IF(CHI(L,M).LT.CHIMES)  GOTO 500
C
            ARRIVL(3,J) = 3
            WRITE(10,100) PREVDY,PREVHR,MIN,ARRIVL(4,J),
     +                    ARRIVL(5,J), CHIMES
            WRITE(3,100) PREVDY, PREVHR,MIN,ARRIVL(4,J),
     +                    ARRIVL(5,J), CHIMES
            ARRIVL(9,J) = PREVDY
            ARRIVL(10,J) = PREVHR
            ARRIVL(11,J) = MIN
C
  500 CONTINUE
      RETURN
      END
```

SCREEN

The SCREEN subroutine provides a simple display of the CHI matrix on the interactive device being used. Previously (i.e., Version 1.0), these functions were performed in the PRINTE module. Output from the SCREEN subroutine includes:

date time title simulation hour number of puffs active

plus a crude graphic plot of exposures on the grid

This output option is accessed by selecting the 'S' as one of the desired options.

Subroutine SCREEN

```
      OVERLAY SB5
C    **  SCREEN  ** MESOI VERSION 1.1
C
C    **  OUTPUT ROUTINE FOR DISPLAY TO THE INTERACTIVE
C    **    DEVICE BEING USED
C
      SUBROUTINE SCREEN(IPSUM,DELXY)
      REAL CHI(32,32), VALUE(11), BORDER(16)
C
      INTEGER SHR,SDAY,DT,DDAY,DHR, PLTID
      INTEGER  DISPL(50), RTIME(3), RDATE(3)
      INTEGER TITLE(50), SYMBOL(10)
C
      COMMON/MASTER/RDATE,RTIME,TITLE,NPH,DT,CHIMIN,
     1   DDAY,DHR,NSI,ACNST,SDAY,SHR
      COMMON/GRID/CHI
      COMMON/REL/XSOURC,YSOURC
      COMMON/PRI/SYMBOL,VALUE,BORDER
C
      DATA BORDER/3H * ,3H * ,3H * ,3H * ,3H * ,3H * ,
     +              3H * ,3H * ,3H * ,3H * ,3H * ,3H * ,
     +              3H * ,3H * ,3H * ,3H * /
      DATA SYMBOL/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/
      DATA VALUE/1.0E-17,1.0E-16,1.0E-15,1.0E-14,1.0E-13,1.0E-12,
     1          1.0E-11,1.0E-10,1.0E-09,1.0E-08,1.0E-07/
C
C    **  SCREEN DISPLAY OF HOURLY OUTPUT
      WRITE(10,200) TITLE,RDATE,RTIME
  200 FORMAT(/5X,50S1,/,60X,3I2,2X,3I2,/)
      WRITE(10,210) SHR,DDAY,DHR,IPSUM
  210 FORMAT(5X,'SIMULATION HOUR ',I2,15X,'DAY ',I3,3X,
     1  'HOUR ',I2,/,7X,I4,' PUFFS ACTIVE',/)
C
```

SCREEN continued

```
      WRITE(10,500) BORDER
  500 FORMAT(2X,165$)
      DISPL(1) = 1H*
      DISPL(50)= 1H*
C
      DO 600 N = 1, 16
        DO 510 I = 2, 49
          DISPL(I) = 1H
  510   CONTINUE
C
        DO 580 I = 1, 16
          IF(CHI(I*2-1,33-(N*2)).LT.VALUE(1))  GOTO 580
          DO 570 K = 2, 11
            IF(CHI(I*2-1,33-(N*2)).GE.VALUE(K))  GOTO 570
              DISPL(I*3) = SYMBOL(K-1)
              GOTO 580
  570     CONTINUE
C
          DISPL(I*3) = 1H+
  580   CONTINUE
C
C        PUT CHARACTER 'X' AT RELEASE POINT
C        PUT CHARACTER 'M' AT GRID CENTER [MET TOWER]
C
        IF(INT(YSOURC).EQ.16-N)  DISPL(INT((XSOURC+1.0)*3+0.5)) = 1HX
        IF(N.EQ.9)  DISPL(26) = 1HM
        WRITE(10,590) DISPL
  590   FORMAT(1X,50A1)
C
  600 CONTINUE
      WRITE(10,610) BORDER
  610 FORMAT(2X,165$)
      RETURN
      END
```

PRINTE

This subroutine formerly (Version 1.0) handled most of the output of MESOI. In VERSION 1.1, its task is limited to writing out the CHI grid to the OUTLIS file.

```
      OVERLAY SB6
C    ** PRINTE  **  MESOI VERSION 1.1
C
C    **  HANDLES LISTING OF THE CHI MATRIX TO THE PRINT FILE
C
      SUBROUTINE PRINTE(IPSUM,DELXY)
C
      REAL CHI(32,32)
C
      INTEGER SHR,SDAY,DT,DDAY,DHR
      INTEGER  RTIME(3), RDATE(3)
      INTEGER TITLE(50)
C
      COMMON/MASTER/RDATE,RTIME,TITLE,NPH,DT,CHIMIN,
     1   DDAY,DHR,NSI,ACNST,SDAY,SHR
      COMMON/GRID/CHI
C
C    **  CONVERT CHI UNITS TO CURIES * SEC / M**3
C
      DO 110 I = 1, 32
        DO 100 J = 1, 32
          CHI(I,J) = CHI(I,J) * 3600.
  100   CONTINUE
  110 CONTINUE
      IPLUS = 0
      WRITE(3,200) TITLE, RDATE, RTIME
  200 FORMAT(5X,'RUN TITLE = ',50S1/,5X,'MODEL WAS RUN ON ',
     +  3I2,' AT ',3I2,' <GMT>',/)
      WRITE(3,205)
  205 FORMAT(5X,'EXPOSURE MATRIX -- CURIES SECONDS / M**3',
     +  5X,'[Q = 1.0 CURIES PER HOUR]',/)
      WRITE(3,210) SHR, DDAY, DHR, IPSUM
  210 FORMAT(5X,'SIMULATION HOUR ',I2,15X,'DAY ',I3,3X,
     +   'HOUR ',I2,/,7X,I4,' PUFFS ACTIVE',/)
      DO 350 L = 1, 4
```

PRINTE continued

```
      I = IPLUS + 1
      II = IPLUS + 8
      WRITE(3,310) (N,N=I,II)
310   FORMAT(/1X,'Y GRID',25X,'X GRID COORDINATE',
     +          /1X,'COORD',6X,I2,7(7X,I2)/)
C
      DO 340 J = 1, 31
        J32 = 32 - J
        IP1 = 1 + IPLUS
        IP11 = 8 + IPLUS
        WRITE(3,330) J32,(CHI(I,J32),I=IP1,IP11)
330     FORMAT(I3,5X,8(1X,1PE8.2))
340   CONTINUE
      IPLUS = IPLUS + 8
350 CONTINUE
    RETURN
    END
```

PUFPLT

This is the AFOS graphics routine. Individual puffs are drawn in the display space using circles centered on the puff center of mass. The program uses the low-level graphics software developed for AFOS.

Subroutine PUFPLT

```
      OVERLAY SB7
C     **  PUFPLT  **  MESOI VERSION 1.1
C
C     **  SUBROUTINE DESIGNED TO PLOT THE LOCATIONS OF EACH ACTIVE PUFF.
C     **     USES THE PUFF CENTER X,Y COORDINATES AND THE PUFF RADIUS.
C     **     GRAPHICS ARE CREATED BY EACH CALL.
C
      SUBROUTINE PUFPLT(IPSUM, DHR, TITLE,XP,YP,DELXY)
C
      REAL XP(72), YP(72), RP(72)
C
      INTEGER BORDER(12), PTITL(6), FTITL(6), PLTID, DIGIT(60),
     +        GRAFIX(4096),SCRIPT(30), CIRCLE(76),
     +        TITLE(50), DHR, HEAD1(30),HEAD2(30),HEAD3(30),HEAD4(30)
C
      INTEGER RELDAY,RELHR,PARTHR
C
      COMMON/PXTRA/BORDER,HEAD1,HEAD2,HEAD3,HEAD4,DIGIT,
     +           GRAFIX,CIRCLE,SCRIPT
      COMMON/REL/XSOURC,YSOURC,RELDAT,RELHR,PARTHR
      COMMON/PLT/PLTID,RP
C
      DATA BORDER/50,50,3050,50,3050,3050,50,3050,50,50,-1,-1/
C
      DATA HEAD1/'MESOI PUFF PLOT FILENAME IS:  '/
C
      DATA DIGIT/'01','02','03','04','05','06','07','08','09','10',
     +           '11','12','13','14','15','16','17','18','19','20',
     +           '21','22','23','24','25','26','27','28','29','30',
     +           '31','32','33','34','35','36','37','38','39','40',
     +           '41','42','43','44','45','46','47','48','49','50',
     +           '51','52','53','54','55','56','57','58','59','60'/
      NS = 4096
```

```
C
C    **  DEFINE TWO PLOT LABELS:
C    **     1) PTITL = THE AFOS PRODUCT TITLE
C    **     2) FTITL =  THE RDOS DISK FILENAME
C
       PTITL(1) = 'HN'
       PTITL(2) = 'FM'
       PTITL(3) = 'ES'
       PTITL(4) = PLTID
       PTITL(5) = DIGIT(DHR)
       PTITL(6) = 0
C
       FTITL(1) = 'ME'
       FTITL(2) = 'SO'
       FTITL(3) = 'PT'
       FTITL(4) = PLTID
       FTITL(5) = DIGIT(DHR)
       FTITL(6) = 0
C
C    **  DEFINE THE LOWER LEFT CORNER AND THE UPPER RIGHT CORNER
C    **      AS X AND Y PIXELS-- 200 PIXELS PER GRID UNIT
C
       IXORG = 50
       IYORG = 50
       IXMAX = 3050
       IYMAX = 3050
C
       DTR = 0.0174532925
C
C    **  INITIALIZE PLOTTING ROUTINES AND SET UP FILES
```

```
C
      CALL COMMH(GRAFIX,PTITL,NWDS,NS,IER)
      IF(IER.EQ.1)  GOTO 200
        WRITE(10,120)
  120   FORMAT(/5X,'ERROR IN PUFPLT -- CALL TO COMMH')
        RETURN
C
  200 CALL GPD1(GRAFIX,NWDS,NS,0)
C
C   **  DRAW IN THE LOCATION OF THE SOURCE
C   **  THE PASSED VALUES OF XSOURC AND YSOURC HAVE
C   **    A RANGE OF 0 - 15 INCLUSIVE
C
      IXSP = IXORG + (XSOURC * 200.0)
      IYSP = IYORG + (YSOURC * 200.0)
C
      SCRIPT(1) = 22K
      SCRIPT(2) = 7
      SCRIPT(3) = 21K
      SCRIPT(4) = 0
C
      ISIZ = 2
       IZT = 1
C
      CALL TEXT(ISIZ,GRAFIX,IXSP,IYSP,SCRIPT,NWDS,NS,IZT,IER)
      IF(IER.EQ.1)  GOTO 300
        WRITE(10,280)
  280   FORMAT(/5X,'ERROR IN PUFPLT -PLOTTING SOURCE LOCATIONS')
        RETURN
C   **  PLOT A CIRCLE FOR EACH ACTIVE PUFF
C
  300 DO 500 I = 1, IPSUM
```

```
C
      PX = IXORG + (XP(I) * 200.0)
      PY = IYORG + (YP(I) * 200.0)
      PLEN = RP(I) * 200.0
C
C    **  RADIUS IS PASSED IN GRID UNITS THEN CONVERTED TO PIXELS
C
C    **  DRAW A CIRCLE USING 36 LINE SEGMENTS OF RADIUS PLEN
C    **      WITH CENTER AT PX,PY
C
      K = 1
      DO 400 J = 10,370,10
C
        RADJ = (J-10) * DTR
        CIRCLE(K) = PX + (SIN(RADJ) * PLEN)
        IF(CIRCLE(K).LT.IXORG)  CIRCLE(K) = IXORG
        IF(CIRCLE(K).GT.IXMAX)  CIRCLE(K) = IXMAX
C
        CIRCLE(K+1) = PY + (COS(RADJ) * PLEN)
        IF(CIRCLE(K+1).LT.IYORG)  CIRCLE(K+1) = IYORG
        IF(CIRCLE(K+1).GT.IYMAX)  CIRCLE(K+1) = IYMAX
C
        K = K + 2
  400 CONTINUE
C
      CIRCLE(75) = -1
      CIRCLE(76) = -1
      CALL MOVFVEC(GRAFIX,CIRCLE,NWDS,NS,1,IER)
      IF(IER.EQ.1)  GOTO 500
        WRITE(10,480) I,PX,PY,PLEN
  480   FORMAT(/5X,'ERROR IN PUFPLT -- PLOTTING PUFF',/,
```

```
     +            'NUMBER ',I3,/,
     +          /7X,'PIXEL COORDINATES ARE ',2(I5,3X),'RADIUS = ',I5,' PIXELS')
  500 CONTINUE
      WRITE(10,510)
  510 FORMAT(/5X,'ALL PUFFS PLOTTED')
C
C    **  PLOT HEADER INFORMATION TO THE RIGHT OF THE PUFF
C    **    PLOT AREA; USE PIXEL AREA: X 3060 - 4080
C    **                               Y 50 - 3050
C
      IP = 3080
      JP = 3000
      IZT = 1
      ISIZ = 0
C
      CALL TEXT(ISIZ,GRAFIX,IP,JP,TITLE,NWDS,NS,IZT,IER)
C
      IP = 3080
      JP = 2550
C
      CALL TEXT(ISIZ,GRAFIX,IP,JP,HEAD1,NWDS,NS,IZT,IER)
C
      IP = 3080
      JP = 2500
C
      CALL TEXT(ISIZ,GRAFIX,IP,JP,FTITL,NWDS,NS,IZT,IER)
      IXSOR = INT(XSOURC+0.5)
      IYSOR = INT(YSOURC+0.5)
C
C    **  DRAW THE GRAPHIC BORDER
```

```
C
      CALL MOVEVEC(GRAFIX,BORDER,NWDS,NS,1,IER)
C
      CALL UTFDS(FTITL,GRAFIX,NWDS,NS)
      WRITE(3,610) FTITL, NWDS
      WRITE(10,610) FTITL,NWDS
  610 FORMAT(/5X,'PLOT FILE GENERATION COMPLETE -- ',6S2/,
     +   I5,' WORDS USED'/)
      RETURN
      END
```

EXPSUM

The subroutine EXPSUM writes out an exposure summary for the defined checkpoints. The array ARRIVL with appropriate headings is sent to both the interactive device and the output file. The subroutine also outputs the final exposure at each checkpoint.

Subroutine EXPSUM

```
      OVERLAY SBB
C     **  EXPSUM  **  MESOI VERSION 1.1
C
C     **  AN EXPOSURE SUMMARY IS DISPLAYED AND PRINTED
C     **    GIVING FOR EACH CHECKPOINT:
C
C     **      1) NAME OF LOCATION
C     **      2) CHI GRID X AND Y COORDINATES
C     **      3) DATE AND TIME MINIMUM CONCENTRATION WAS EXCEEDED
C     **      4) DATE AND TIME MINIMUM * E5 WAS EXCEEDED
C     **      5) FINAL CONCENTRATION
C
      SUBROUTINE EXPSUM(TITLE,CHIMIN)
C
      REAL CHI(32,32), ARRIVL(11,35)
C
      INTEGER TITLE(50)
C
      COMMON/ARR/ARRIVL
      COMMON/GRID/CHI
C
      CHIME5 = CHIMIN * 1.0E5
C
      WRITE(10,420) TITLE
      WRITE(3,420) TITLE
  420 FORMAT(//5X,'EXPOSURE SUMMARY FOR MESOI ==> ',50S1//)
      WRITE(10,430)
      WRITE(3,430)
  430 FORMAT(/5X,'CHECKPOINT',5X,'CHI X,Y',7X,'EXPOSURE EXCEEDED',
     +      11X,'FINAL')
C
      WRITE(10,440) CHIMIN, CHIME5
      WRITE(3,440) CHIMIN, CHIME5
```

```
  440 FORMAT(8X,'NAME',6X,'COORDINATES',3X,1PE10.3,3X,1PE10.3,7X,
     +      'EXPOSURE [CI SEC/M**3]')
      WRITE(10,445)
      WRITE(3,445)
  445 FORMAT(33X,'CI HR/M**3',3X,'CI HR/M**3')
      WRITE(10,450)
      WRITE(3,450)
  450 FORMAT(33X,'DAY TIME',5X,'DAY TIME'/)
C
      DO 480 J=1, 35
        IF(ARRIVL(3,J).EQ.0.)  GOTO 480
        L = ARRIVL(1,J)
        IF(L.LT.1) L=1
        IF(L.GT.31) L=31
        M = ARRIVL(2,J)
        IF(M.LT.1) M=1
        IF(M.GT.31) M =31
        IF(CHI(L,M).EQ.0.0)  GOTO 480
        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(10,460) (ARRIVL(I,J),I=4,5),(ARRIVL(I,J),I=1,2),
     +    IDAY1,IHR1,IMIN1,IDAY2,IHR2,IMIN2,CHI(L,M)
        WRITE(3,460) (ARRIVL(I,J),I=4,5),(ARRIVL(I,J),I=1,2),
     +      IDAY1,IHR1,IMIN1,IDAY2,IHR2,IMIN2,CHI(L,M)
  460   FORMAT(/6X,2S4,4X,2(1X,F4.0),2(5X,I3,1X,2I2),6X,1PE10.3)
  480 CONTINUE
      RETURN
      END
```

Subroutine CLEAN

```
C    **  CLEAN  **  MESOI VERSION 1.1
C
C    **  ROUTINE TO ELIMINATE PUFFS WHICH HAVE LEFT THE GRID
C
      SUBROUTINE CLEAN
C
      REAL SIGMAY(72), SIGMAZ(72), Q(72), STOTAL(72), XP(72),
     1   YP(72), DXS(72), DYS(72)
C
      INTEGER MF(72), TPUFFS
C
      COMMON/PUFFS/MF,SIGMAY,SIGMAZ,Q,STOTAL,XP,YP,TPUFFS,NPUFFS,
     1   DXS,DYS,FAC,ADT
C
      I = 1
      J = 1
  100 IF(MF(I).EQ.0)  GOTO 120
      MF(J) = MF(I)
      SIGMAY(J) = SIGMAY(I)
      Q(J) = Q(I)
      STOTAL(J) = STOTAL(I)
      SIGMAZ(J) = SIGMAZ(I)
      XP(J) = XP(I)
      YP(J) = YP(I)
      DXS(J) = DXS(I)
      DYS(J) = DYS(I)
      J = J + 1
C
  120 I = I + 1
      IF(I.LE.TPUFFS)  GOTO 100
```

```
C
      JM1 = J - 1
      ITJM1 = TPUFFS - JM1
      WRITE(10,140) ITJM1, JM1
  140 FORMAT(5X,'CLEAN CALLED --> ',I4,' PUFFS DROPPED',
     1  4X,I4,' PUFFS REMAIN ON THE GRID',/)
      TPUFFS = J - 1
      RETURN
      END
```

Subroutine JULIAN

```
      SUBROUTINE JULIAN(IYR,MO,IDAY,JDATE)
C
C     JULIAN CONVERTS YEAR, MONTH AND DAY TO JULIAN DATE
C
C     CHECK FOR MONTH INDEX IN RANGE
C
      IF(MO.LE.0.OR.MO.GE.13) GO  TO 30
C
C     CONVERT TO JULIAN DATE
C
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12),MO
    1 JDATE=IDAY
      GO TO 40
    2 JDATE=IDAY+31
      GO TO 20
    3 JDATE=IDAY+59
      GO TO 20
    4 JDATE=IDAY+90
      GO TO 20
    5 JDATE=IDAY+120
      GO TO 20
    6 JDATE=IDAY+151
      GO TO 20
    7 JDATE=IDAY+181
      GO TO 20
    8 JDATE=IDAY+212
      GO TO 20
    9 JDATE=IDAY+243
      GO TO 20
   10 JDATE=IDAY+273
      GO TO 20
   11 JDATE=IDAY+304
      GO TO 20
   12 JDATE=IDAY+334
   20 CONTINUE
```

```
C
C       ADJUST FOR LEAP YEAR
C
        A=FLOAT(IYR)/4-IYR/4
        IF(A.EQ.0.AND.MO.GE.3) JDATE=JDATE+1
        GO TO 40
   30   CONTINUE
        JDATE=0
   40   CONTINUE
        RETURN
        END
```

APPENDIX C

STAPREP Program Description and Listing

The STAPREP program is essentially the interactive portion of the GRIDIN subroutine plus the subroutines STRAY, ASCND and REARNG in MESOI Version 1.0. They were extracted and implemented as a separate program to allow the user the flexibility of constructing a variety of STATION files which cover the situations likely to be modeled. As such, the decision making and interaction required to set up the station arrays can be done before MESOI execution. In addition, computations done in computing the station distances to grid points are very repetitive and time consuming (esp. on a computer competing for resources with other programs). If a 'standard' set of stations is used, there is no reason to repeat the calculations for each model execution.

The file STATIONS consists of two major parts:

1) the formatted section; defining grid spacing, station names, distances, status and elevations.

2) the binary section; containing the arrays STDIST and STNUM (as described in the MESOI variables discussion).

The formatted section always consists of 31 records; the grid spacing in the first record followed by 30 station records. Non-existent stations should be designated with name = XXXX and with status set to 1.

STAPREP requires that the formatted section of the STATIONS file exist before execution. Initially, it can be created with an editor or cards to create a data file. The format and a sample of contents is shown following:

Column	Format	Contents
1	1X	blank
2-5	S4	4 character name of station
10-15	F6.2	X distance (km) from grid center
20-25	F6.2	Y distance (km) from grid center
30	I1	station status (0=active, 1-disabled)
35-38	I4	station elevation (meters MSL)

Record No.	Column Numbers 12345678901234567890123456789012345678901234567890 1 2 3 4 5				
1	5000.00				
2	PROS	10.26	3.60	0	146
3	EOC	8.33	3.68	0	378
4	ARMY	8.22	5.82	0	172
•	•	•	•	•	•
•	•	•	•	•	•
31	XXXX	0.0	0.0	1	0

The binary section of the file is transparent to the user. It is produced by STAPREP after the completion of the review and edit of the formatted information.

Execution of STAPREP on the DG Eclipse is started by typing STAPREP<cr>. The program responds with a title and asks for the name of the input file. The file must exist and have the required structure, even if the information is dummy. Once a valid file is opened, the formatted portion is read and displayed for user review. Changes to any element of any record can be accomplished via the interactive dialog. An unlimited number of reviews and changes are permitted.

When satisfied with the station descriptions, the user specifies a filename for output. It may be a new name or the same as the input file. The formatted information is written to the file. STAPREP then uses the subroutines STRAY, ASCND and REARNG to calculate the arrays STDIST and STNUM. They are then written in binary to the output file, the file is closed and program execution is terminated. The following pages contain a listing of STAPREP source code and its support subroutines.

Program STRAPREP

```
C    **  STAPREP  **  SUPPORTING MESOI VERSION 1.1
C
C    **  STATION FILE PREPARATION PROGRAM
C    **    SETS UP THE STATION NAMES AND COORDINATES
C    **    COMPUTES THE STATION ARRAY
C
C    **  LOAD LINE:
C    **    RLDR/P STAPREP STRAY ASCND REARNG FORT.LB
C
      REAL XDIST(30), YDIST(30), NAMST(30), XSTA(30), YSTA(30),
     +   STDIST(10,16,16)
C
      INTEGER FILNAM(7),ELEV(30)
      INTEGER STATUS(30), STNUM(10,16,16), ACTSTA, SEL, STANUM(30)
C
C
      WRITE(10,50)
   50 FORMAT(//5X,'STATION FILE PREPARATION PROGRAM'//)
   55 WRITE(10,60)
   60 FORMAT(/5X,'ENTER NAME OF THE STATIONS FILE TO BE REVIEWED',
     +       ' OR MODIFIED  >',Z)
      READ(11,70) FILNAM(1)
   70 FORMAT(S13)
C
      CALL OPEN(3,FILNAM,2,IER)
      IF(IER.EQ.1)  GOTO 100
        WRITE(10,90) IER
   90   FORMAT(/5X,'ERROR OPENING SPECIFIED FILE - IER = ',I3)
        GOTO 55
```

```
C
  100 READ(3,110) SETXY
  110 FORMAT(1X,F8.2)
      DO 150 J=1, 30
        READ(3,120) NAMST(J),XDIST(J),YDIST(J),STATUS(J),ELEV(J)
  120   FORMAT(1X,S4,2(4X,F6.2),4X,I1,4X,I4)
  150 CONTINUE
C
  160 WRITE(10,170)
  170 FORMAT(/6X,'STA NAME',4X,'XDIST',3X,'YDIST',4X,'STATUS',
     +        4X,'ELEV')
      DO 200 J=1, 30
        WRITE(10,190) J,NAMST(J),XDIST(J),YDIST(J),STATUS(J),ELEV(J)
  190   FORMAT(6X,I2,2X,S4,5X,F6.2,2X,F6.2,6X,I1,7X,I4)
  200 CONTINUE
C
  210 WRITE(10,220)
  220 FORMAT(/5X,'ANY CHANGES TO STATION RECORDS? Y OR N')
      READ(11,230,END=210,ERR=210) SEL
  230 FORMAT(S1)
      IF(SEL.NE.'Y') GOTO 510
C
  240 WRITE(10,250)
  250 FORMAT(/5X,'HOW MANY STATIONS TO BE CHANGED?')
      READ(11,END=240,ERR=240) ICHG
      IF(ICHG.EQ.0)  GOTO 510
      IF(ICHG.LT.0.OR.ICHG.GT.30)  GOTO 240
  260 WRITE(10,270)
  270 FORMAT(/5X,'ENTER STATION NUMBERS TO BE CHANGED, N,N,N ..')
      READ(11,END=260,ERR=260) (STANUM(I),I=1,ICHG)
C
      DO 500 I=1, ICHG
        IF(STANUM(I).LT.1.OR.STANUM(I).GT.30)  GOTO 500
        J = STANUM(I)
```

```
      WRITE(10,300) J
  300 FORMAT(/5X,'CHANGE NAME FOR STATION ',I3,' Y OR N')
      READ(11,230) SEL
      IF(SEL.NE.'Y')  GOTO 330
      WRITE(10,310)
  310 FORMAT(/5X,'ENTER 4 CHARACTER NAME')
      READ(11,320) XNAM
  320 FORMAT(S4)
      NAMST(J) = XNAM
C
  330 WRITE(10,340) J
  340 FORMAT(/5X,'CHANGE X AND Y DISTANCES FOR STATION ',I3,
     +       ' Y OR N')
      READ(11,230) SEL
      IF(SEL.NE.'Y')  GOTO 360
      WRITE(10,350)
  350 FORMAT(/5X,'ENTER X AND Y DISTANCES FROM HMS IN KM')
      READ(11) XDIST(J), YDIST(J)
C
  360 WRITE(10,370) J
  370 FORMAT(/5X,'CHANGE STATUS ON STATION ',I3,' Y OR N')
      READ(11,230) SEL
      IF(SEL.NE.'Y')  GOTO 400
C
  380 WRITE(10,390)
  390 FORMAT(/5X,'ENTER STATUS -- 0 OR 1')
      READ(11) STATUS(J)
      IF(STATUS(J).LT.0.OR.STATUS(J).GT.1)  GOTO 380
C
  400 WRITE(10,410) J
  410 FORMAT(/5X,'CHANGE ELEVATION ON STATION ',I3,'Y OR N >',Z)
      READ(11,230) SEL
      IF(SEL.NE.'Y')  GOTO 500
```

```
  420 WRITE(10,430)
  430 FORMAT(/5X,'ENTER ELEVATION IN METERS(MSL) >',Z)
      READ(11) ELEV(J)
      IF(ELEV(J).LT.0.OR.ELEV(J).GT.6000)  GOTO 420
C
  500 CONTINUE
      GOTO 160
C
  510 WRITE(10,520)
  520 FORMAT(/5X,'SPECIFY THE DELTA XY TO BE USED WITH THIS DATA')
      READ(11) SETXY
C
      CALL CLOSE(3,IER)
  550 WRITE(10,560)
  560 FORMAT(/5X,'ENTER FILENAME FOR OUTPUT  >',Z)
      READ(11,570) FILNAM(1)
  570 FORMAT(S13)
      CALL OPEN(3,FILNAM,0,IER)
      IF(IER.EQ.1)  GOTO 610
        WRITE(10,580) IER
  580   FORMAT(/5X,'ERROR OPENING SPECIFIED FILE - IER = ',I3)
        GOTO 550
C
  610 WRITE(3,620) SETXY
  620 FORMAT(2X,F8.2)
      DO 650 I=1,30
        WRITE(3,630) NAMST(I),XDIST(I),YDIST(I),STATUS(I),ELEV(I)
  630   FORMAT(2X,S4,2(4X,F6.2),4X,I1,4X,I4)
  650 CONTINUE
C
```

```
      NUMSTA = 0
       NSTAT = 0
      DO 600 I=1, 30
        IF(NAMST(I).EQ.'XXXX')  GOTO 600
        XSTA(I) = (XDIST(I) * 1000. / SETXY) + 7.5
        YSTA(I) = (YDIST(I) * 1000. / SETXY) + 7.5
        NUMSTA = NUMSTA + 1
        IF(STATUS(I).EQ.1)  NSTAT = NSTAT + 1
  600 CONTINUE
C
      ACTSTA = NUMSTA - NSTAT
      IF(NSTAT.NE.0)  CALL REARNG(NUMSTA,XSTA,YSTA,NAMST,STATUS)
      CALL STRAY(ACTSTA,XSTA,YSTA,STDIST,STNUM)
C
      WRITE BINARY(3) STDIST
      WRITE BINARY(3) STNUM
      CALL CLOSE(3,IER)
      WRITE(10,900)
  900 FORMAT(/5X,'END PROGRAM STAPREP')
      STOP
      END
```

```
C    ** REARNG  **  STAPREP
C
C    ** REMOVES STATIONS WITH STATUS FLAG SET FROM THE
C    **   SET OF ACTIVE STATIONS TO BE USED IN WIND INTERPOLATION
C
      SUBROUTINE REARNG(NUMSTA,XSTA,YSTA,NAMST,STATUS)
C
      REAL NAMST(30)
      REAL   XSTA(30), YSTA(30)
C
      INTEGER STATUS(30), ACTSTA, STANUM(30)
C
      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)
  100 CONTINUE
C
C
      WRITE(10,200)
  200 FORMAT(/5X,'REARNG:  DISABLED STATIONS REMOVED FROM LIST',/)
      RETURN
      END
```

Subroutine STRAY

```
C    ** STRAY  **  STAPREP
C
C    ** SETS UP THE STATION ARRAY FOR EACH WIND GRID POINT
C
      SUBROUTINE STRAY(NUMSTA,XSTA,YSTA,STDIST,STNUM)
C
      REAL XSTA(30),YSTA(30)
      REAL STDIST(10,16,16)
      REAL RT(30)
C
      INTEGER  STNUM(10,16,16), NSB(30)
C
      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
C
          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
C
      WRITE(10,400) NUMSTA
  400 FORMAT(5X,'STRAY:  STATION ARRAY SET UP FOR EACH GRID ',
     1   'POINT',/,15X,I3,' ACTIVE STATIONS',//)
      RETURN
      END
```

Subroutine ASCND

```
C    ** ASCND  **  STAPREP
C
C    ** FOR A GIVEN GRID POINT, STATIONS ARE ARRANGED
C    **   IN ASCENDING ORDER BY DISTANCE FROM THE POINT
C
C    **    X -- DISTANCE (X**2 + Y**2)
C    **    N -- STATION NUMBER
C    **    NP - TOTAL NUMBER OF STATIONS
C
      SUBROUTINE ASCND(X,N,NP)
C
      REAL X(30)
      INTEGER N(30)
C
      NPM1 = NP - 1
      DO 110 I = 1,NPM1
        J = 0
        DO 100 K = 1, NPM1
          IF(X(K+1).GT.X(K))  GOTO 100
          J = K + 1
          TMP = X(K)
          NTMP = N(K)
          X(K) = X(J)
          N(K) = N(J)
          X(J) = TMP
          N(J) = NTMP
  100   CONTINUE
        IF(J.EQ.0) GOTO 120
  110 CONTINUE
C
  120 RETURN
      END
```

APPENDIX D
EXPLT Program Description and Listing

The EXPLT program is an AFOS graphics generation code designed to improve MESOI's utility as an emergency response tool. MESOI writes the values of the CHI matrix to an output file following each hour's computation. In Version 1.0, this file is used by the NEWCONTUR program to generate a line-printer plot of accumulated exposure on the grid. Version 1.1 uses AFOS graphics (PUFPLT subroutine) software to generate a more detailed view of material transport.

EXPLT allows the user to plot exposure time histories for any selected point on the CHI grid (31 x 31). It requires the following:

1) that at least 2 hours of MESOI simulation has been completed

2) that final exposures at the selected points exceed 1.0E-14 sec/m**3

3) the CHI matrix file CHIDMP is available for use

Execution on the DG Eclipse is started by typing EXPLT<cr> at the background terminal. The program responds with a title and then requests a two character plot ID. This identification is used to uniquely name the plotfiles created. Then, the user is asked for the X and Y coordinates of the CHI grid point for which exposure is to be plotted. The data file is read until all hourly records have been input. If the final exposure is too low, no plot is generated and the program loops back to the query for X and Y.

If a plot is generated, the message:

EXPOSURE PLOT EXPLIDXXYY COMPLETE

is printed, where ID is the 2 specified characters and XX and YY are the CHI coordinates used. The plotfile is then available for display on the AFOS GDM while the program requests new coordinates for plotting. The program is terminated by entering an XX value of 99.

Figure D-1 shows a sample plot generated from the CHIDMP file. Following is a listing of the source code.

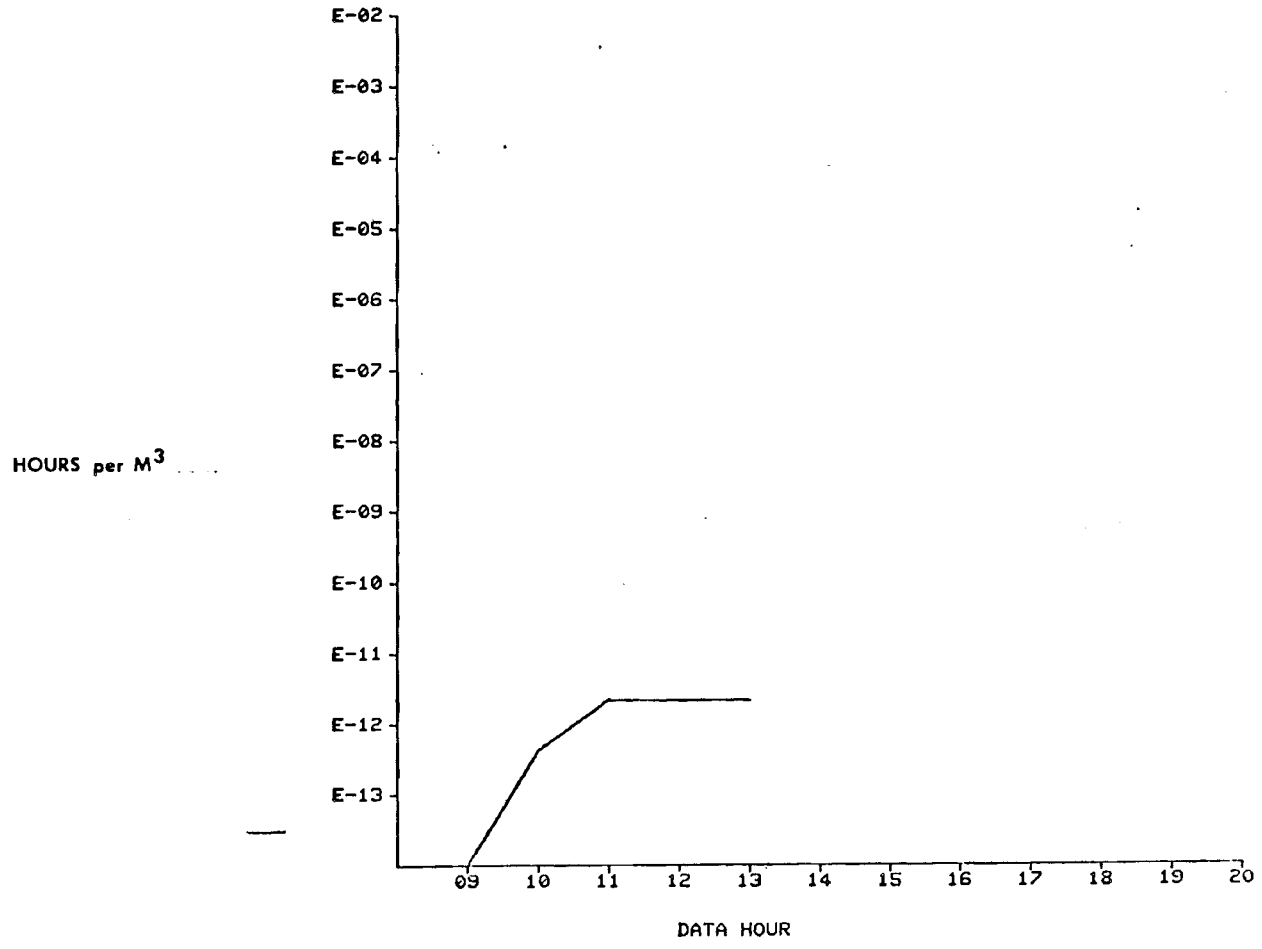

FIGURE D-1. Sample of Exposure Plot Generated by EXPLT.

Program EXPLT

```
C    **  EXPLT  **  SUPPORTING MESOI VERSION 1.1
C
C    **  EXPOSURE PLOTTER
C
C    **  GENERATES AFOS PLOTFILES FOR USER SPECIFIED LOCATIONS
C    **    ON THE CHI GRID(31,31). THE PROGRAM USES A BINARY FILE
C    **    WRITTEN BY MESOI AND AVAILABLE AT THE END OF A SIMULATION RUN
C
      REAL CHIMAT(32,32), CHI(48)
C
      INTEGER PTITL(6), FTITL(6), GRAFIX(2048),BORDER(12), FNAME(20),
     +  DIGIT(50), ILINE(100), HOUR(48),HEAD1(50),HEAD2(50),SCRIPT(5)
C
      COMMON/EXPL/DIGIT,HEAD1,HEAD2,GRAFIX,CHIMAT,CHI,ILINE,
     +              BORDER, SCRIPT,FTITL,PTITL,FNAME
C
      DATA HEAD1/'DATA HOUR'/
      DATA HEAD2/'HOURS PER CU M'/
      DATA BORDER/50,50,4000,50,4000,3050,50,3050,50,50,-1,-1/
C
      DATA DIGIT/'01','02','03','04','05','06','07','08','09','10',
     +           '11','12','13','14','15','16','17','18','19','20',
     +           '21','22','23','24','25','26','27','28','29','30',
     +           '31','32','33','34','35','36','37','38','39','40',
     +           '41','42','43','44','45','46','47','48','49','50'/
C
      WRITE(10,100)
  100 FORMAT(/5X,'EXPLT -- EXPOSURE PLOTTING PROGRAM')
C
      NS = 2048
C
C    **  READ IN THE CHI MATRICES AND TIMES
C
```

```
      CALL OPEN(1,'CHIDMP.',1,IER)
      IF(IER.NE.1) TYPE 'ERROR OPENING CHIDMP FILE - IER = ',IER
      WRITE(10,180)
  180 FORMAT(/5X,'ENTER 2 CHARACTER RUN ID     >',Z)
      READ(11,190) ID
  190 FORMAT(S2)
C
  200 WRITE(10,210)
  210 FORMAT(/5X,'ENTER X AND Y CHI COORDINATES FOR THE POINT'/,
     +     10X,'USE X=99 TO EXIT     >',Z)
      READ(11) ICHIX, ICHIY
      IF(ICHIX.EQ.99)  GOTO 800
      IF(ICHIX.LT.1.OR.ICHIX.GT.31)  GOTO 200
      IF(ICHIY.LT.1.OR.ICHIY.GT.31)  GOTO 200
      NREC = 0
  212 READ BINARY(1,END=215) IHR, CHIMAT
      NREC = NREC + 1
      HOUR(NREC) = IHR
      CHI(NREC) = CHIMAT(ICHIX,ICHIY)
      GOTO 212
C
C
  215 IF(CHI(NREC).GT.1.0E-14)  GOTO 250
        WRITE(10,220) NREC, CHI(NREC)
  220   FORMAT(/5X,'CHI / Q AT THAT POINT NEVER EXCEEDED 1.0E-14',/
     +       10X,'NREC = ',I3,5X,'CHI/Q = ',E10.3,5X,'  NO PLOT')
        REWIND 1
        GOTO 200
C
  250 PTITL(1) = 'EX'
```

```
      PTITL(2) = 'PL'
      PTITL(3) = ID
      PTITL(4) = DIGIT(ICHIX)
      PTITL(5) = DIGIT(ICHIY)
      PTITL(6) = 0
C
      FTITL(1) = 'EX'
      FTITL(2) = 'PL'
      FTITL(3) = ID
      FTITL(4) = DIGIT(ICHIX)
      FTITL(5) = DIGIT(ICHIY)
      FTITL(6) = 0
C
      DO 255 I = 1, 20
        FNAME(I) = 0
  255 CONTINUE
      DO 260 I = 1, 6
        FNAME(I) = FTITL(I)
  260 CONTINUE
C
      CALL COMMH(GRAFIX,PTITL,NWDS,NS,IER)
      CALL GP01(GRAFIX,NWDS,NS,0)
      WRITE(10,270) ID, DIGIT(ICHIX), DIGIT(ICHIY)
  270 FORMAT(/5X,'CREATING EXPL PLOTFILE FOR ',3S2)
C
C   **  DRAW AXES AND TICK MARKS
C
      IOX = 1200
      IOY = 600
C
      IMX = 3600
      IMY = 3000
C
```

```
      ISIZ = 1
      ITLEN = -20
      IZT = 1
      LOGMIN = -14
      ISPACE = 200
      IF(NREC.GT.12)  ISPACE = 100
      IF(NREC.GT.24)  ISPACE = 50
      ISHIFT = 200 / ISPACE
C
      CALL VEC(GRAFIX,IOX,IOY,IMX,IOY,NWDS,NS,IZT,IER)
      CALL VEC(GRAFIX,IOX,IOY,IOX,IMY,NWDS,NS,IZT,IER)
C
      DO 300 I=1,12
        IXTICK = IOX + (200 * I)
        IYTICK = IOY + ITLEN
        CALL VEC(GRAFIX,IXTICK,IOY,IXTICK,IYTICK,NWDS,NS,IZT,IER)
        NLABL = (ISHIFT * (I-1)) + HOUR(ISHIFT)
        IF(NLABL.GT.24)  NLABL=NLABL -24
        IXLAB = IXTICK - 30
        IYLAB = IOY - 60
        SCRIPT(1) = DIGIT(NLABL)
        SCRIPT(2) = 0
        CALL TEXT(ISIZ,GRAFIX,IXLAB,IYLAB,SCRIPT,NWDS,NS,IZT,IER)
  300 CONTINUE
C
      DO 320 I = 1, 12
        IYTICK = IOY + (200 * I)
        IXTICK = IOX + ITLEN
        CALL VEC(GRAFIX,IOX,IYTICK,IXTICK,IYTICK,NWDS,NS,IZT,IER)
        NLABL = (LOGMIN + I) * (-1)
```

```
        IXLAB = IOX - 180
        IYLAB = IYTICK - 10
        SCRIPT(1) = 105K * 400K + 55K
        SCRIPT(2) = DIGIT(NLABL)
        SCRIPT(3) = 0
        CALL TEXT(ISIZ,GRAFIX,IXLAB,IYLAB,SCRIPT,NWDS,NS,IZT,IER)
C
  320 CONTINUE
C
C   **  GENERATE X,Y PAIRS FOR THE POINTS TO BE PLOTTED
C
      J = 1
      JP1 = J+1
C
      DO 410 N=1, NREC
        ILINE(J) = IOX + (N * ISPACE)
        YLOG = FLOAT(LOGMIN)
        IF(CHI(N).GT.0.0)  YLOG = ALOG10(CHI(N))
        ILINE(JP1) = IOY + IFIX((YLOG - FLOAT(LOGMIN)) * 200.0)
C
        J = J + 2
        JP1 = JP1 + 2
  410 CONTINUE
C
      ILINE(J) = -1
      ILINE(JP1) = -1
      CALL MOVEVEC(GRAFIX,ILINE,NWDS,NS,IZT,IER)
C   **  PLOT THE AXES LABELS
C
      IXLAB = 1700
      IYLAB = 200
      ISIZ = 3
      CALL TEXT(ISIZ,GRAFIX,IXLAB,IYLAB,FNAME,NWDS,NS,IZT,IER)
```

```
C
      ISIZ = 1
      IXLAB = 2000
      IYLAB = 400
      CALL TEXT(ISIZ,GRAFIX,IXLAB,IYLAB,HEAD1,NWDS,NS,IZT,IER)
C
      IXLAB = 400
      IYLAB = 1800
      CALL TEXT(ISIZ,GRAFIX,IXLAB,IYLAB,HEAD2,NWDS,NS,IZT,IER)
C
      CALL MOVEVEC(GRAFIX,BORDER,NWDS,NS,1,IER)
C
      CALL UTFDS(FTITL,GRAFIX,NWDS,NS)
      WRITE(10,420) FTITL, NWDS
  420 FORMAT(/5X,'EXPOSURE PLOT ',6S2,' COMPLETE'/,
     +     10X,I5,' WORDS USED')
      REWIND 1
      GOTO 200
C
  800 CALL CLOSE(1,IER)
      STOP
      END
```

APPENDIX E

Test Data Set and Sample Output

A very simple data set is used to test the operation of MESOI. The test case requires a data set with the following conditions in each hourly set of three records:

STAB = 2 (stability class)

LDEPTH = 100 (mixing depth of 1000 meters)

winds of 310 degrees from N at 10 mph

The following conditions are specified during initialization:

source at -23.20711, 23.20711

release for 2 hours (12 puffs)

simulation runs for 2 hours

A listing was requested at hours 1 and 2. Puffs were plotted at hour 2. The following pages show the output listing of the checkout run. Figure E-1 shows a sample puff plot. It is overlayed on a Hanford area background.

Output from Sample/Test Case

```
MESOI -- THE INTERACTIVE EDITION OF MESO
          VERSION 1.1     SEPTEMBER, 1982

          TIME = 19: 4:35

          DATE =  9/29/82


MESOI ---> GRID INITIALIZATION

   DELXY =   5000.0 METERS

THE CURRENT WIND GRID IS:
    16 ROWS       16 COLUMNS

THERE ARE CURRENTLY 30 STATIONS WITH  8 DISABLED

 STA  NAME     GRIDX    GRIDY     STATUS
  1   PROS    10.26     3.60        0
  2   EOC      8.33     3.68        0
  3   ARMY     8.22     5.82        0
  4   RSPG     5.91     6.28        0
  5   EDNA    10.61     7.92        0
  6   200E     8.69     7.30        0
  7   200W     7.00     7.04        0
  8   WAHL     3.99    11.43        0
  9   FFTF    11.17     4.59        0
 10   YAKB     5.57     7.89        0
```

11	300A	12.17	2.92	0
12	WYEB	10.61	5.60	0
13	100N	8.13	9.94	0
14	WPPS	11.36	5.30	0
15	FRNK	12.94	4.08	0
16	GARL	9.57	8.26	0
17	RING	13.03	6.91	0
18	RICH	11.90	1.53	0
19	SAGE	12.15	9.41	0
20	RMTN	7.47	3.74	0
21	HYS	7.50	7.50	0
22	PASC	14.71	0.60	0
23	XXXX	7.50	7.50	1
24	XXXX	7.50	7.50	1
25	XXXX	7.50	7.50	1
26	XXXX	7.50	7.50	1
27	XXXX	7.50	7.50	1
28	XXXX	7.50	7.50	1
29	XXXX	7.50	7.50	1
30	XXXX	7.50	7.50	1

** END GRID INITIALIZATION **

```
MESOI --> SET UP ARRIVAL CHECKPOINTS FROM CHKPNT FILE

30 CHECKPOINTS ACTIVE ON THE CURRENT GRID


METEOROLOGICAL DATA FILE SEARCH --

  OBSV FILE POSITIONED AT:   DAY 112    HOUR  8    RECORD 1

  FORECAST FILE STARTS AT:   DAY 112    HOUR  1    RECORD 1

DATA FOR DAY: 82 112  HOUR  8 <PST>

   STABILITY = 2   MIXING DEPTH = 1000 METERS

   WINDS          3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                  3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                  3110 3110 3110 3110 3110 3110 3110 3110 3110 3110


TEST 1A      92982  19 435

SIMULATION STARTS AT HOUR  8 <PST> ON DAY  4:22


SOURCE IS LOCATED AT WIND GRID  5.1  9.9

RELEASE WILL OCCUR AT  8: 0 <PST> ON  4/22
    12 PUFFS AT      0.167 CURIES PER PUFF
```

```
     DATA FOR DAY: 82 112  HOUR  9 <PST>

        STABILITY = 2   MIXING DEPTH = 1000 METERS

        WINDS          3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                       3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                       3110 3110 3110 3110 3110 3110 3110 3110 3110 3110

     PUFF RELEASE FLAG SET ---> AFTER HOUR  8  ADV STEP  1


DAY 112 AT  8: 0 <PST>  EXPOSURE  AT VERNITA  EXCEEDS  3.333E-17 CURIE HOUR/M***3

DAY 112 AT  8:10 <PST>  EXPOSURE  AT VERNITA  EXCEEDS  3.333E-12 CURIE HOUR/M***3

DAY 112 AT  8:40 <PST>  EXPOSURE  AT 200 WEST EXCEEDS  3.333E-17 CURIE HOUR/M***3

DAY 112 AT  8:50 <PST>  EXPOSURE  AT 200 EAST EXCEEDS  3.333E-17 CURIE HOUR/M***3

DAY 112 AT  8:50 <PST>  EXPOSURE  AT  MSTF    EXCEEDS  3.333E-17 CURIE HOUR/M***3

     RUN TITLE = TEST 1A
     MODEL WAS RUN ON  92982 AT 19 435 <GMT>

     EXPOSURE MATRIX -- CURIES SECONDS / M***3      (0 = 1.0 CURIES PER HOUR)
```

SIMULATION HOUR 1 DAY 112 HOUR 9
6 PUFFS ACTIVE

Y GRID COORD	X GRID COORDINATE 1	2	3	4	5	6	7	8
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE 9	10	11	12	13	14	15	16
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	6.06E -7	1.46E -8	9.77E-11	3.67E-13	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	1.79E-10	9.71E -8	3.65E -8	2.01E -9	8.55E-11
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	1.63E -9	3.18E -8	2.19E -8	3.45E -9
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	1.09E-11	1.90E -9	1.16E -8	8.02E -9
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	4.34E-11	9.88E-10	2.10E -9
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	2.90E-13	2.01E-11	7.11E-11
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	2.06E-13
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE 17	18	19	20	21	22	23	24
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	2.80E-12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	2.11E-10	3.25E-12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	1.10E -9	2.30E-11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
16	5.13E-10	1.33E-11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
15	2.29E-11	5.84E-13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE 25	26	27	28	29	30	31	32
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

```
     DATA FOR DAY: 82 112  HOUR 10 <PST>

        STABILITY = 2   MIXING DEPTH = 1000 METERS

        WINDS          3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                       3110 3110 3110 3110 3110 3110 3110 3110 3110 3110
                       3110 3110 3110 3110 3110 3110 3110 3110 3110 3110

DAY 112 AT  9: 0 <PST>  EXPOSURE  AT 200 EAST EXCEEDS  3.333E-12 CURIE HOUR/M**3

DAY 112 AT  9: 0 <PST>  EXPOSURE  AT 200 WEST EXCEEDS  3.333E-12 CURIE HOUR/M**3

DAY 112 AT  9: 0 <PST>  EXPOSURE  AT GABLE MT EXCEEDS  3.333E-17 CURIE HOUR/M**3

DAY 112 AT  9: 0 <PST>  EXPOSURE  AT ARMY LP  EXCEEDS  3.333E-17 CURIE HOUR/M**3

DAY 112 AT  9:10 <PST>  EXPOSURE  AT LANDFILL EXCEEDS  3.333E-17 CURIE HOUR/M**3

DAY 112 AT  9:50 <PST>  EXPOSURE  AT  NSTF    EXCEEDS  3.333E-12 CURIE HOUR/M**3

DAY 112 AT  9:50 <PST>  EXPOSURE  AT 400 AREA EXCEEDS  3.333E-17 CURIE HOUR/M**3

DAY 112 AT  9:50 <PST>  EXPOSURE  AT  WNP-2   EXCEEDS  3.333E-17 CURIE HOUR/M**3

     RUN TITLE = TEST 1A
     MODEL WAS RUN ON  82982 AT 19 435 <GMT>

     EXPOSURE MATRIX -- CURIES SECONDS / M**3      [Q = 1.0 CURIES PER HOUR]

     SIMULATION HOUR  2                DAY 112   HOUR 10
         12 PUFFS ACTIVE
```

Y GRID COORD	X GRID COORDINATE 1	2	3	4	5	6	7	8
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE 9	10	11	12	13	14	15	16
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	2.18E -3	5.24E -5	3.52E -7	1.32E -9	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	6.44E -7	3.50E -4	1.32E -4	7.24E -6	3.08E -7
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	5.89E -6	1.15E -4	7.90E -5	1.25E -5
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	3.93E -8	6.85E -6	4.17E -5	2.89E -5
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	1.56E -7	3.56E -6	7.60E -6
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	1.05E -9	7.32E -8	2.63E -7
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	3.16E-11	1.56E -9
13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	4.96E-11
12	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE							
	17	18	19	20	21	22	23	24
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	1.01E -8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	7.62E -7	1.20E -8	3.08E-11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	3.99E -6	8.89E -8	9.68E-10	1.11E-10	7.90E-12	0.00E 0	0.00E 0	0.00E 0
16	1.89E -6	7.21E -8	7.65E -9	1.54E -9	2.09E-10	1.91E-11	5.16E-13	0.00E 0
15	1.07E -7	3.27E -8	1.95E -8	7.14E -9	1.61E -9	2.26E-10	1.80E-11	5.24E-13
14	5.73E -9	1.54E -8	1.93E -8	1.27E -8	4.72E -9	9.69E-10	1.02E-10	4.78E-12
13	6.94E-10	3.64E -9	8.50E -9	9.70E -9	5.62E -9	1.60E -9	2.10E-10	1.13E-11
12	4.71E-11	4.54E-10	1.83E -9	3.35E -9	2.81E -9	1.04E -9	1.60E-10	9.41E-12
11	0.00E 0	3.13E-11	2.00E-10	5.30E-10	5.84E-10	2.60E-10	4.46E-11	2.72E-12
10	0.00E 0	5.34E-13	1.05E-11	3.67E-11	4.88E-11	2.43E-11	4.45E-12	2.03E-13
9	0.00E 0	0.00E 0	0.00E 0	8.69E-13	1.41E-12	7.15E-13	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

Y GRID COORD	X GRID COORDINATE 25	26	27	28	29	30	31	32
31	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
30	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
29	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
28	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
27	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
26	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
25	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
24	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
23	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
22	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
21	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
20	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
19	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
18	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
17	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
16	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
15	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
14	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
13	1.38E-13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
12	1.35E-13	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
11	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
10	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
9	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
8	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
7	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
6	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
5	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
4	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
3	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
2	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0
1	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0	0.00E 0

END OF SIMULATION

EXPOSURE SUMMARY FOR MESOI ==> TEST 1A

CHECKPOINT NAME	CHI X,Y COORDINATES	EXPOSURE EXCEEDED 3.333E-17 CI HR/M***3 DAY TIME	EXPOSURE EXCEEDED 3.333E-12 CI HR/M***3 DAY TIME	FINAL EXPOSURE (CI SEC/M***3)
200 EAST	18. 16.	112 850	112 9 0	3.268E -8
200 WEST	15. 15.	112 840	112 9 0	7.315E -8
NSTF	18. 18.	112 850	112 950	8.890E -8
GABLE MT	20. 16.	112 9 0	0 0 0	1.536E -9
LANDFILL	20. 13.	112 910	0 0 0	9.702E -9
ARMY LP	17. 13.	112 9 0	0 0 0	4.711E-11
VERNITA	12. 20.	112 8 0	112 810	6.442E -7
400 AREA	23. 10.	112 950	0 0 0	4.448E-12
WNP-2	24. 12.	112 950	0 0 0	4.457E-11

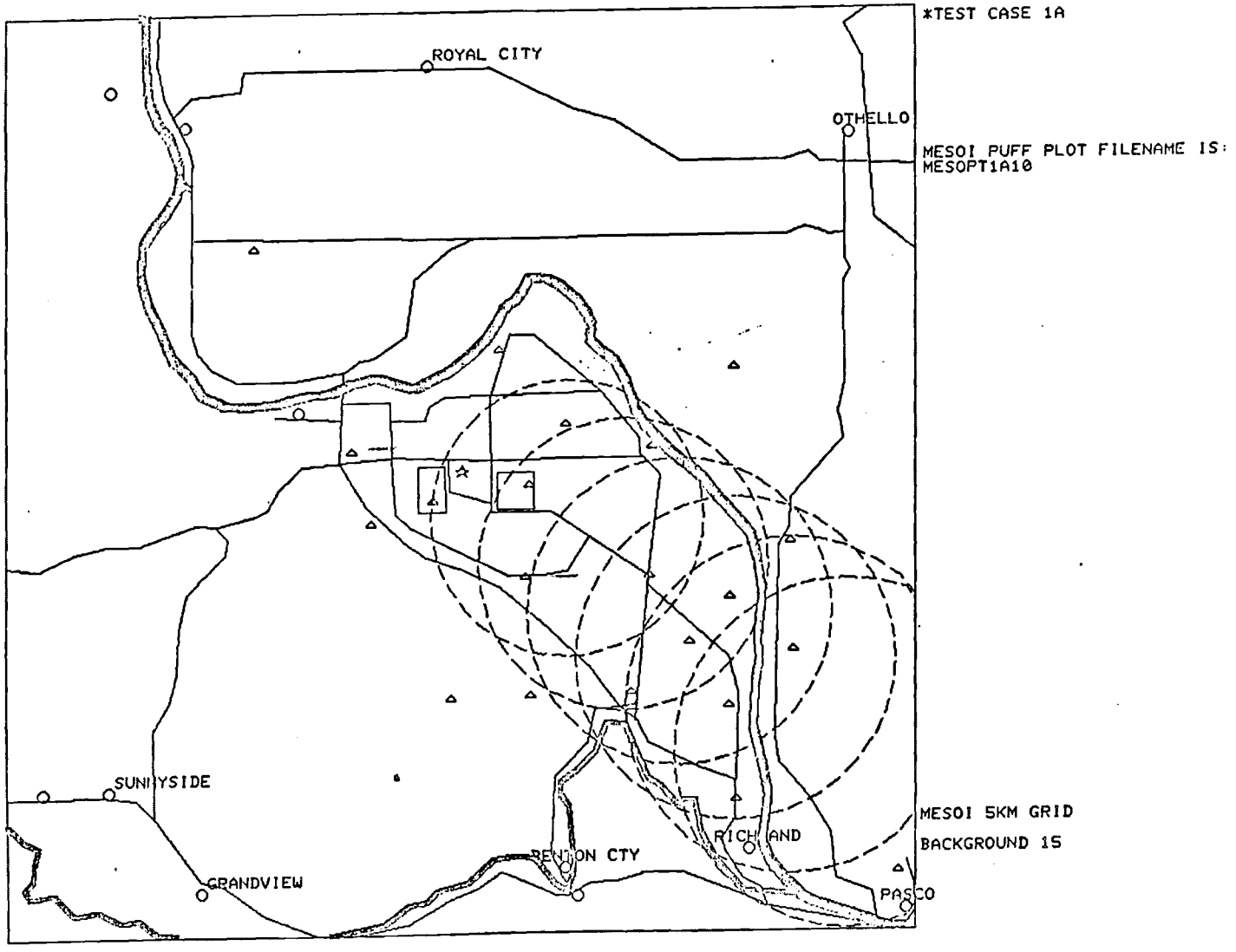

FIGURE E-1. PUFPLT Output for a Sample Run.

APPENDIX F
Description of AFOS Graphics and GDM Display Space

All the graphics routines used in MESOI Version 1.1 make use of low level plotting software in which each element of a picture must be individually created. Plots are constructed by repeated calls to the subroutines MOVEVEC (for drawing lines) and TEXT (for drawing alphanumeric characters). The graphics screen is logically addressed using a Cartesian grid with an abscissa(X) of 0 to 4095 and an ordinate(Y) of 0 to 3071. The actual display CRT has a resolution of 1024 x 768 pixels.

Documentation of the plotting software used in the Version 1.1 programs is limited. A similar software set is described by MacDonald (1981) and could be used to produce similar graphic products. Future software development for the AFOS system graphics will use Tektronics PLOT-10 now available to convert the resultant graphic product into a form displayable on the AFOS GDM.

In developing graphics for use with MESOI, the available display space was partitioned in defined areas. All plotting is done within a square defined by the points:

0,0 0,3050 3050,3050 3050,0

The remaining space on the right of the screen is used for labels, keys and other descriptive information.

On a three GDM setup such as in the Hanford Forecast Center, one of the screens has limited color capability. The unit can display products in red, green or white, but the user cannot select the color using plot software or have multiple colors within a single displayable plot file. To make use of the color features, components of the final picture must be produced as separate files and then overlayed at the GDM with different colors. For example, with MESOI the background may be displayed in white and the puffs in red and green.

There is no 'real time' graphics capability with the AFOS. That is, the user cannot watch a program draw a product and see components added as the simulation progresses. Each complete product file must be created by the generation program, then called up for display from the foreground terminal.

APPENDIX G

UPDATE Version 1.1 Program Listing and Notes

The data management programs have changed very little in their modification to make Version 1.1. Thus, no separate discussion is presented for each section of the code. Only the actual source code listings are given.

```
C     **  MASTER  **  UPDATE VERSION 1.1
C
C     **  THIS IS THE MAIN PROGRAM OF THE DATA FILE MANAGEMENT
C     **    PROGRAM ASSOCIATED WITH THE MESOI MODEL
C     **    VERSION FOR DG ECLIPSE WITH FORTRAN IV
C
      INTEGER DARRAY(35,48), SELECT, YR(48)
C
      REAL STNAME(30)
C
      COMMON/DAT1/DARRAY,YR
C
      COMMON/NAMES/STNAME,NSTA
      EXTERNAL SB1,SB2,SB3,SB4,SB5,SB6,SB7
C     **  OVERLAY ASSIGNMENTS:
C
C     **     SB1 = NUDAT
C     **     SB2 = ARCHIV
C     **     SB3 = RVUDAT
C     **     SB4 = REVDAT
C     **     SB5 = NUFCST
C     **     SB6 = RVUFST
C     **     SB7 = REVFST
C
      DATA STNAME/'PROS',' EOC','ARMY','RSPG',
     +            'EDNA','200E','200W','WAHL',
     +             'FFTF','YAKB','300A','WYEB',
     +             '100N','WPPS','FRNK','GABL',
     +             'RING','RICH','SAGE','RMTN','HMST','PASC'/
      NSTA=22
      CALL OVOPN(9,'UPDATE.OL',IER)
      WRITE(10,9000)
```

```
 9000 FORMAT(1H1)
      WRITE(10,9011)
 9011 FORMAT(///5X,'UPDATE    VERSION 1.1     MARCH 1982',//)
   10 WRITE(10,9002)
 9002 FORMAT(/10X,'DO YOU WISH TO ENTER OR REVISE OBSERVED DATA?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=10,ERR=10) SELECT
 8000 FORMAT(S1)
      IF(SELECT.NE.'Y') GO TO 200
C
C     READ IN OLD OBSERVED  DATA FILE AND PLACE IN DATA ARRAYS
C
      CALL OPEN(1,'MDATA',2,IER)
      IF(IER.NE.1)  WRITE(10,400) IER
  400   FORMAT(/5X,'ERROR OPENING MDATA FILE - CODE = ',I3)
      DO 20 I=1,48
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,15)
 8001   FORMAT(I2,I3,I2,I1,1X,I1,1X,I3,5X,10(2X,I4))
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=16,25)
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=26,35)
   20 CONTINUE
      REWIND 1
   35 WRITE(10,9003)
 9003 FORMAT(/10X,'DO YOU WISH TO ENTER A NEW SET OF OBSERVATIONS?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=35,ERR=35) SELECT
      IF(SELECT.NE.'Y') GO TO 45
C
C     CALL SUBROUTINE TO ENTER NEW OBSERVATIONS
C
      CALL OVLOD(9,SB1,1,IER)
```

```
      CALL NUDAT
C
   45 WRITE(10,9004)
 9004 FORMAT(/10X,'DO YOU WISH TO REVIEW THE OBSERVED DATA FILE?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=45,ERR=45) SELECT
      IF(SELECT.NE.'Y') GO TO 55
C
C     CALL SUBROUTINE TO REVIEW THE OBSERVED DATA FILE
C
   50 CALL OVLOD(9,SB3,1,IER)
      CALL RVUDAT

C
   55 WRITE(10,9005)
 9005 FORMAT(/10X,'DO YOU WISH TO REVISE ANY DATA ENTRIES?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=55,ERR=55) SELECT
      IF(SELECT.NE.'Y') GO TO 60
C
C     CALL SUBROUTINE TO REVISE OBSERVED DATA FILE
C
      CALL OVLOD(9,SB4,1,IER)
      CALL REVDAT

C
   57 WRITE(10,9004)
      READ(11,8000,END=57,ERR=57) SELECT
      IF(SELECT.EQ.'Y') GO TO 50
   60 CONTINUE
C
C     WRITE OUT NEW OBSERVED DATA FILE
C
```

```
      DO 70 I=1,48
        DARRAY(3,I)=1
 8801   FORMAT(1X,I2,I3,I2,I1,1X,I1,1X,I3,5X,10(2X,I4))
        WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,15)
        DARRAY(3,I)=2
        WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=16,25)
        DARRAY(3,I)=3
        WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=26,35)
   70 CONTINUE
      CALL CLOSE(1,IER)
      IF(IER.NE.1)  WRITE(10,420) IER
  420   FORMAT(/5X,'ERROR CLOSING MDATA FILE - CODE = ',I3)
  200 CONTINUE
C
C     SECTION TO UPDATE FORECAST DATA
C
  205 WRITE(10,9006)
 9006 FORMAT(/10X,'DO YOU WISH TO ENTER OR REVISE FORECAST DATA?'/
     +      10X,'ENTER Y RO N')
      READ(11,8000,END=205,ERR=205) SELECT
      IF(SELECT.NE.'Y')  GOTO 300
C
C     READ IN OLD FORECAST DATA FILE AND PLACE IN ARRAYS
C
      CALL OPEN(1,'FDATA',2,IER)
      IF(IER.NE.1)  WRITE(10,440) IER
  440   FORMAT(/5X,'ERROR OPENING FDATA FILE - CODE = ',I3)
      DO 220 I=1,48
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,15)
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=16,25)
```

```
        READ(1,8001) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=26,35)
  220 CONTINUE
      REWIND 1
  235 WRITE(10,9007)
 9007 FORMAT(/10X,'DO YOU WISH TO ENTER A NEW FORECAST?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=235,ERR=235) SELECT
      IF(SELECT.NE.'Y') GO TO 240
C
C     CALL SUBROUTINE TO ENTER A NEW FORECAST
C
      CALL OVLOD(9,SR5,1,IER)
      CALL NUFCST

C
  240 CONTINUE
  245 WRITE(10,9008)
 9008 FORMAT(/10X,'DO YOU WISH TO REVIEW THE CURRENT FORECAST?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=245,ERR=245) SELECT
      IF(SELECT.NE.'Y') GO TO 255
C
C     CALL SUBROUTINE TO REVIEW FORECASTS
C
  250 CALL OVLOD(9,SR6,1,IER)
      CALL RVUFST
  255 WRITE(10,9009)
 9009 FORMAT(/10X,'DO YOU WISH TO REVISE THE FORECAST?'/
     +       10X,'ENTER Y OR N.')
      READ(11,8000,END=255,ERR=255) SELECT
      IF(SELECT.NE.'Y') GO TO 260
C
```

```
C      CALL SUBROUTINE TO REVISE THE FORECAST FILE
C
       CALL OVLOD(9,SB7,1,IER)
       CALL REVFST
C
  257  WRITE(10,9008)
       READ(11,8000,END=257,ERR=257) SELECT
       IF(SELECT.EQ.'Y') GO TO 250
  260  CONTINUE
C
C      WRITE OUT THE FORECAST DATA FILE
C
       DO 270 I=1,48
         DARRAY(3,I)=1
         WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,15)
         DARRAY(3,I)=2
         WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=16,25)
         DARRAY(3,I)=3
         WRITE(1,8801) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=26,35)
  270  CONTINUE
C
       CALL CLOSE(1,IER)
       IF(IER.NE.1)  WRITE(10,460) IER
  460    FORMAT(/5X,'ERROR CLOSING FDATA FILE - CODE = ',I3)
  300  CONTINUE
       WRITE(10,9020)
 9020  FORMAT(//10X,'THIS UPDATE SESSION IS OVER.')
       CALL CLOSE(9,IER)
       END
```

Subroutine CAL

```
      SUBROUTINE CAL(IYR,JDATE,MO,IDAY)
C
C     CAL CONVERTS JULIAN DATE AND YEAR TO CALENDER MONTH AND DAY.
C
      COMMON/DUMMY/ISTART,ISTOP,LSTART,LSTOP
      DIMENSION ISTART(12),ISTOP(12),LSTART(12),LSTOP(12)
      DATA ISTART/1,32,60,91,121,152,182,213,244,274,305,335/
      DATA ISTOP/31,59,90,120,151,181,212,243,273,304,334,365/
      DATA LSTART/1,32,61,92,122,153,183,214,245,275,306,336/
      DATA LSTOP/31,60,91,121,152,182,213,244,274,305,335,366/
C
C     CHECK FOR JULIAN DATE OUT OF RANGE.
C
      IF(JDATE.LT.1.OR.JDATE.GT.366) GO TO 90
C
C     DETERMINE IF LEAP YEAR.
C
      A=FLOAT(IYR)/4 - IYR/4
      IF(A.EQ.0.) GO TO 25
C
C     DETERMINE MONTH AND DAY.
C
      DO 10 J=1,12
      IF(JDATE.GE.ISTART(J).AND.JDATE.LE.ISTOP(J)) GO TO 20
   10 CONTINUE
   20 MO=J
      IDAY=JDATE-(ISTART(J)-1)
      GO TO 100
C
   25 DO 30 J=1,12
      IF(JDATE.GE.LSTART(J).AND.JDATE.LE.LSTOP(J)) GO TO 40
   30 CONTINUE
   40 MO=J
      IDAY=JDATE-(LSTART(J)-1)
      GO TO 100
   90 CONTINUE
      MO=0
      IDAY=0
  100 CONTINUE
      RETURN
      END
```

Subroutine INDEX

```
      SUBROUTINE INDEX(MDD,MH,INDX)
      INTEGER DARRAY(35,48)
      COMMON /DAT1/DARRAY
      K=0
      DO 10 I=1,48
      IF(MDD.EQ.DARRAY(1,I)) GO TO 20
      K=K+1
   10 CONTINUE
      IF(K.EQ.48) GO TO 50
   20 CONTINUE
      INDX=K
      IB=K+1
      IE=IB+(24-DARRAY(2,IB)+1)
      IF(IE.GT.48) IE=48
      K=1
      DO 30 I=IB,IE
      IF(MH.EQ.DARRAY(2,I)) GO TO 40
      K=K+1
   30 CONTINUE
      IF(K.GT.IE) GO TO 50
   40 CONTINUE
      INDX=INDX+K
      GO TO 60
   50 CONTINUE
      INDX=0
   60 CONTINUE
      RETURN
      END
```

```
      OVERLAY SB1
C    **  NUDAT  * UPDATE VERSION 1.1
C
      SUBROUTINE NUDAT
C
      INTEGER DARRAY(35,48), SELECT,YR(48)
C
      REAL STNAME(50)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
C
    1 WRITE(10,1025)
 1025 FORMAT(/10X,'ENTER YEAR OF OBSERVATION TO BE ENTERED: YY'/
     +       10X,'(F.G. - 81).')
      READ(11,ERR=1,END=1)MY
      IYRP1 = YR(1)+1
      IF(MY.EQ.YR(1).OR.MY.EQ.IYRP1)  GOTO 270
        WRITE(10,260)
  260   FORMAT(/5X,'YEAR SPECIFIED DOES NOT MATCH YEAR OF DATA',/)
        GOTO 1
C
C    DETERMINE MONTH, DAY AND HOUR OF LAST ENTRY.
C
  270 IF(DARRAY(1,25).EQ.0) GO TO 200
      DO 280 J=25,48
        IF(DARRAY(2,J).GT.24) GO TO 290
  280 CONTINUE
C
  290 CONTINUE
      J=J-1
      MDD=DARRAY(1,J)
      CALL CAL(MY,MDD,MM,MD)
      WRITE(10,1026) MM,MD,DARRAY(2,J)
```

```
 1026 FORMAT(/10X,'THE MONTH, DAY AND HOUR OF THE LAST OBSERVATION'/
     +       10X,' ENTERED ARE: ',3I3,'  .'//
     +       10X,'THE NEW ENTRIES MUST FOLLOW THIS TIME.')
      GO TO 10
  200 WRITE(10,1027)
 1027 FORMAT(/10X,'NO OLD OBSERVATIONS ARE TABULATED.  THE NEW ENTRIES'/
     +       10X,'MUST START WITH HOUR 01.')
C
   10 WRITE(10,1000)
 1000 FORMAT(/10X,'ENTER MONTH AND DAY OF OBSERVATIONS TO BE ENTERED:'/
     +       10X,'MM,DD  (E.G. - 7,29).')
      READ(11,ERR=10,END=10)MM,MD
      IF(MM.LT.1.OR.MM.GT.12) GO TO 10
      IF(MD.LT.1.OR.MD.GT.31) GO TO 10
   20 WRITE(10,1002)
 1002 FORMAT(/10X,'ENTER HOUR OF THE OBSERVATION: HH .   99 MAY BE'/
     +       10X,'USED TO ESCAPE FROM THE DATA ENTRY ROUTINE.')
      READ(11,ERR=20,END=20)MH
      IF(MH.EQ.99) GO TO 140
      IF(MH.LT.1.OR.MH.GT.24) GO TO 20
   25 WRITE(10,1022) MM,MD,MY,MH
 1022 FORMAT(/10X,'THE DATE AND HOUR FOR OBSERVATION TO BE'/
     +       10X,' ENTERED ARE:  ',3I2,3X,I2//
     +       10X,'ARE THESE CORRECT?   ENTER Y OR N.')
      READ(11,1011,END=25,ERR=25) SELECT
      IF(SELECT.NE.'Y') GO TO 10
      CALL JULIAN(MY,MM,MD,JDATE)
C
      IF((JDATE.LT.MDD.AND.JDATE.NE.1).OR.JDATE.GT.(MDD+1))  GOTO 3000
         GOTO 3010
```

```
 3000 WRITE(10,1001)
 1001   FORMAT(/10X,'DATE MUST BE SAME OR +1 DAY TO THE ',
     +  'LAST OBSERVATION DAY',/)
        GOTO 10
 3010 MDD = JDATE
      WRITE(10,1012) MY,MDD
 1012 FORMAT(/10X,'THE JULIAN DATE OF THE DATA IS',I2,2X,I3)
C
C     SHIFT DATA IN ARRAY IF OBSERVATION TO BE ENTERED IS FIRST OF
C     THE DAY.    OTHERWISE SKIP THE SHIFT.
C
C     IF DATA ARE SHIFTED FILL THE REMAINDER OF THE ARRAY WITH 8888.
C
      IF(NH.GT.1) GO TO 50
C
      CALL ARCHIV
C
      DO 40 I=1,24
        II=I+24
        DO 30 J=1,35
          YR(I) = YR(II)
          DARRAY(J,I)=DARRAY(J,II)
   30   CONTINUE
        DO 35 J=6,35
          DARRAY(J,II)=8888
   35   CONTINUE
        YR(II) = 88
        DARRAY(1,II)=888
        DARRAY(2,II)=88
        DARRAY(4,II)=8
        DARRAY(5,II)=888
   40 CONTINUE
   50 CONTINUE
```

```
C
C       ENTER DATE, TIME, STABILITY AND MIXING DEPTH IN DARRAY
C
        IH=MH+24
        YR(IH) = MY
        DARRAY(1,IH)=MDD
        DARRAY(2,IH)=MH
   60   WRITE(10,1003)
 1003   FORMAT(/10X,'ENTER STABILITY CLASS: 1 THROUGH 7 .'2X)
        READ(11,ERR=60,END=60)ISTB
        IF(ISTB.LT.1.OR.ISTB.GT.7) GO TO 60
        DARRAY(4,IH)=ISTB
C
   70   WRITE(10,1004)
 1004   FORMAT(/10X,'ENTER MIXING DEPTH IN TENS OF METERS:'/
     +         10X,' 1 THRU 300 . ',2X)
        READ(11,ERR=70,END=70)LDEPTH
        IF(LDEPTH.LT.1.OR.LDEPTH.GT.300) GO TO 70
        DARRAY(5,IH)=LDEPTH
C
C       ENTER WIND OBSERVATION BY STATIONS
C
        WRITE(10,1006)
 1006   FORMAT(/10X,'ENTER WIND DATA AS DDFF, WITH 0000=CALM AND'
     +          ,/10X,'9999=MISSING.  DO NOT USE LT/VAR, USE CALM',/
     +       10X,'OR ESTIMATE THE AVERAGE SPEED AND DIRECTION.')
        NSTAP5 = NSTA + 5
        DO 120    J=6,NSTAP5
        J1=J-5
        IX=0
```

```
   80 WRITE(10,1007) SINAME(J1)
 1007 FORMAT(/10X,'ENTER WIND FOR',2X,S4,2X)
      READ(11,ERR=80,END=80)IDDFF
      IF(IDDFF.EQ.0.OR.IDDFF.EQ.9999) GO TO 110
      IDD=IDDFF/100
      IFF=IDDFF-IDD*100
      IDD = IDD * 10
      IF(IDD.LT.0.OR.IDD.GT.360) GO TO 90
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) GO TO 100
      GO TO 110
   90 WRITE(10,1008)
 1008 FORMAT(/10X,'WIND DIRECTION OUT OF RANGE')
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) WRITE(10,1009)
 1009 FORMAT(/10X,'WIND SPEED ABNORMALLY HIGH')
      IX=1
      GO TO 80
C
C
  100 WRITE(10,1009)
      IX=1
      GO TO 80
  110 CONTINUE
      DARRAY(J,IH)=IDDFF
  120 CONTINUE
  130 WRITE(10,1010)
 1010 FORMAT(/10X,'DO YOU WISH TO ENTER OBSERVATIONS FOR ANOTHER TIME?'/
     +        10X,'ENTER Y OR N.')
      READ(11,1011,END=130,ERR=130) SELECT
 1011 FORMAT(S1)
      IF(SELECT.EQ.'Y') GO TO 10
  140 CONTINUE
      RETURN
      END
```

Subroutine ARCHIV

```
      OVERLAY SB2
C    **  ARCHIV  **  UPDATE VERSION 1.1
C
C    **  THIS PROGRAM WRITE 24 HOURS OF METEOROLOGICAL DATA
C    **    TO AN ARCHIVE FILE BEFORE IT IS DELETED FROM THE
C    **    ACTIVE DATA SET AVAILABLE TO MESOI
C
      SUBROUTINE ARCHIV
C
      INTEGER DARRAY(35,48), SELECT,YR(48)
C
      COMMON/DAT1/DARRAY,YR
C
 1000 WRITE(10,1010)
 1010 FORMAT(/5X,'DO YOU WISH TO ARCHIVE THE OBSERVATION DATA? ',
     + /,' Y OR N      IF PRACTICE SESSION ANSWER ''N''  > ')
      READ(11,1020,ERR=1000,END=1000) SELECT
 1020 FORMAT(S1)
      IF(SELECT.NE.'Y'.AND.SELECT.NE.'N')  GOTO 1000
      IF(SELECT.EQ.'N')  RETURN
C
      CALL APPEND(8,'ARCHIV',2,IER)
      IF(IER.EQ.1)  GOTO 1090
         WRITE(10,1080) IER
 1080    FORMAT(/5X,'ERROR CODE = ',I3,' ARCHIV FILE NOT OPENED')
        RETURN
C
 1090 WRITE(10,1100)
 1100 FORMAT(/5X,'DATA ARCHIVE FILE OPENED',//5X,
     +  'COPYING LAST 24 HOURS OF OBSERVATIONS TO FILE ARCHIV.')
C
      DO 1170 I = 1, 24
```

```
        DARRAY(3,I) = 1
        WRITE(8,1120) YR(I),(DARRAY(J,I),J=1,15)
 1120   FORMAT(1X,I2,I3,I2,I1,1X,I1,1X,I3,5X,10(1X,I5))
        DARRAY(3,I) = 2
        WRITE(8,1120) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=16,25)
        DARRAY(3,I) = 3
        WRITE(8,1120) YR(I),(DARRAY(J,I),J=1,5),(DARRAY(J,I),J=26,35)
 1170 CONTINUE
      ENDFILE 8
C
      CALL CLOSE(8)
      WRITE(10,1180)
 1180 FORMAT(/5X,'24 HOURS OF DATA ADDED TO ARCHIVES ',//)
      RETURN
      END
```

```
      OVERLAY SB3
C    **  RVUDAT  **  UPDATE VERSION 1.1
C
      SUBROUTINE RVUDAT
C
      INTEGER DARRAY(35,48),SELECT,YR(48)
C
      REAL STNAME(30)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
      NSTAP5 = NSTA + 5
      Y='Y'
   10 WRITE(10,1000)
 1000 FORMAT(/10X,'DO YOU WISH TO REVIEW THE ENTIRE OBSERVED DATA FILE?',
     +       /10X,'ENTER Y OR N. ')
      READ(11,1001,END=10,ERR=10) SELECT
 1001 FORMAT(S1)
      IF(SELECT.NE.'Y') GO TO 50
C
C     THIS SECTION PERMITS THE USER TO REVIEW THE ENTIRE OBSERVED DATA
C     FILE IN 6 HOUR BLOCKS.
C
      WRITE(10,1002) (STNAME(I),I=1,NSTA)
 1002 FORMAT(/10X,'THE OBSERVED FILE WILL BE DISPLAYED IN 6',
     + ' HOUR BLOCKS.',/10X,'   HITTING N<CR> WILL DISPLAY THE ',
     + 'NEXT BLOCK.',/10X,'  THE FORECAST ORDER IS: JULIAN DATE,'/
     + 10X,'HOUR, STABILITY, MIXING DEPTH, WINDS.  THE WIND'/
     + 10X,'STATION ORDER IS:',/ 4(10X,8(S4,2X)/)/)
      DO 50 I=1,8
        II=(I-1)*6+1
        IIP5 = II + 5
        DO 20 J=II,IIP5
```

```
          WRITE(10,1003)YR(I),(DARRAY(K,J),K=1,2),(DARRAY(K,J),K=4,NSTAP5)
 1003     FORMAT(3X,I2,I3,2X,I2,2X,I1,2X,I3,5X,8(I5,1X)/3(25X,8(I5,1X)/))
   20    CONTINUE
   22    WRITE(10,2000)
 2000    FORMAT(/5X,'TYPE N <CR> FOR NEXT BLOCK OR JUST <CR> TO EXIT')
         READ(11,1001,END=22,ERR=22) SELECT
         IF(SELECT.NE.'N')  GOTO 50
   30 CONTINUE
C
C      THIS SECTION REVIEWS THE DATA ONE HOUR AT A TIME
C
   50 WRITE(10,1004)
 1004 FORMAT(/10X,'DO YOU WISH TO REVIEW DATA FOR SPECIFIC HOURS?'/
     +        10X,'ENTER Y OR N. ')
      READ(11,1001,END=50,ERR=50) SELECT
      IF(SELECT.NE.'Y') GO TO 100
   60 WRITE(10,1005)
 1005 FORMAT(/10X,'ENTER DATE OF DATA TO BE REVIEWED: MM,DD,YY'/
     +        10X,'(E.G. - 7,29,81).')
      READ(11,ERR=60,END=60)MM,MD,MY
      IF(MM.LT.1.OR.MM.GT.12) GO TO 60
      IF(MD.LT.1.OR.MD.GT.31) GO TO 60
      CALL JULIAN(MY,MM,MD,MDD)
      IF(DARRAY(1,1).EQ.0.AND.DARRAY(1,25).EQ.0)  GOTO 95
      IF(DARRAY(1,1).EQ.MDD.AND.YR(1).EQ.MY)  GOTO 70
      IF(DARRAY(1,25).EQ.MDD.AND.YR(25).EQ.MY)  GOTO 70
      CALL CAL(MY,DARRAY(1,1),MO1,IDAY1)
      CALL CAL(MY,DARRAY(1,25),MO2,IDAY2)
      WRITE(6,1020) YR(1),MO1,IDAY1,YR(25),MO2,IDAY2
 1020 FORMAT(/10X,'THIS DATE IS NOT ON RECORD.'/
     +10X,'THE RECORDED MONTHS-DAYS (YYMMDD) ARE: ',3I2,' AND ',3I2)
```

```
      GOTO 60
C
   70 WRITE(10,1007)
 1007 FORMAT(/10X,'ENTER HOUR OF THE DATA TO BE REVIEWED: HH .')
      READ(11,ERR=70,END=70)MH
      IF(MH.LT.1.OR.MH.GT.24)  GOTO 70
      INDX = MH
      IF(MDD.EQ.DARRAY(1,25)) INDX = MH + 24
      WRITE(10,1008)(STNAME(I),I=1,NSTA)
 1008 FORMAT(/10X,'DATA ARE DISPLAYED IN THE FOLLOWING ORDER:'/
     +       10X,'JULIAN DATE, HOUR, STABILITY, MIXING DEPTH, WINDS,'/
     +       10X,'THE ORDER OF THE WIND FCST IS: '/4(10X,8(S4,2X)/))
 2005 WRITE(10,1003)YR(INDX),(DARRAY(J,INDX),J=1,2),(DARRAY(J,INDX),J=4,NSTAP5)
   72 WRITE(10,2010)
 2010 FORMAT(/5X,'TYPE N <CR> FOR NEXT RECORD OR JUST <CR> TO EXIT')
      READ(11,1001,END=72,ERR=72) SELECT
      IF(SELECT.NE.'N') GOTO 80
        INDX = INDX + 1
        IF(INDX.LE.48)  GOTO 2005
          WRITE(10,2020)
 2020     FORMAT(/5X,'END OF DATA')
C
   80 WRITE(10,1009)
 1009 FORMAT(/10X,'DO YOU WISH TO REVIEW ADDITIONAL HOURS?'/
     +       10X,'ENTER Y OR N. ')
      READ(11,1001,END=80,ERR=80) SELECT
      IF(SELECT.NE.'Y') GO TO 100
C
   90 WRITE(10,1010)
 1010 FORMAT(/10X,'THE SAME DAY? ENTER Y OR N. ')
      READ(11,1001,END=90,ERR=90) SELECT
      IF(SELECT.EQ.'Y') GO TO 70
      GO TO 60
```

```
C
   95 WRITE(10,1025)
 1025 FORMAT(/10X,'THIS OBSERVATION FILE IS EMPTY.')
  100 CONTINUE
      RETURN
      END
```

Subroutine REVDAT

```
      OVERLAY SB4
C    **  REVDAT  ** UPDATE 1.1
C
      SUBROUTINE REVDAT
C
      INTEGER DARRAY(35,48), SELECT,YR(48)
C
      REAL STNAME(30)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
C
      Y='Y'
   10 WRITE(10,1000)
 1000 FORMAT(/10X,'ENTER DATE OF OBSERVATION TO BE REVISED: MM,DD,YY'/
     +       10X,'(E.G. - 7,29,81).')
      READ(11,ERR=10,END=10)MM,MD,MY
      IF(MY.LT.60.OR.MY.GT.85)  GOTO 10
      IF(MM.LT.1.OR.MM.GT.12) GO TO 10
      IF(MD.LT.1.OR.MD.GT.31) GO TO 10
      CALL JULIAN(MY,MM,MD,MDD)
      IF(DARRAY(1,1).EQ.0.AND.DARRAY(1,25).EQ.0) GO TO 175
      IF(DARRAY(1,1).EQ.MDD.AND.MY.EQ.YR(1))  GOTO 15
      IF(DARRAY(1,25).EQ.MDD.AND.MY.EQ.YR(25))  GOTO 15
C
      CALL CAL(MY,DARRAY(1,1),MO1,IDAY1)
      CALL CAL(MY,DARRAY(1,25),MO2,IDAY2)
      WRITE(10,1022) YR(1),MO1,IDAY1,YR(25),MO2,IDAY2
 1022 FORMAT(/10X,'THIS DATE IS NOT ON RECORD.'/
     + 10X,'THE RECORDED MONTHS-DAYS (MMDD) ARE: ',3I2,' AND ',3I2)
      GO TO 10
C
```

```
   15 WRITE(10,1002) MDD
 1002 FORMAT(/10X,' THE JULIAN DATE FOR THE REVISION IS ',2X,I3)
   20 WRITE(10,1003)
 1003 FORMAT(/10X,'ENTER HOUR OF THE OBSERVATION TO BE REVISED: HH .'/
     +        10X,'A 99 CAN BE USED TO ESCAPE FROM THE DATA REVISION'/
     +        10X,'SUBROUTINE.',//)
      READ(11,ERR=20,END=20)MH
      IF(MH.EQ.99) GO TO 180
      IF(MH.LT.1.OR.MH.GT.24) GO TO 20
      INDX=MH
      IF(MDD.EQ.DARRAY(1,25)) INDX=MH+24
C
      NSTAP5 = NSTA + 5
      WRITE(10,2000) YR(INDX),(DARRAY(K,INDX),K=1,2),
     +               (DARRAY(K,INDX),K=4,NSTAP5)
 2000 FORMAT(3X,I2,I3,2X,I2,2X,I1,2X,I3,5X,8(I4,2X)/3(25X,8(I4,2X)/))
 2005 WRITE(10,2010)
 2010 FORMAT(/5X,'IS THIS THE RECORD TO BE REVISED? Y OR N > ')
      READ(11,1005,END=2005,ERR=2005) SELECT
      IF(SELECT.NE.'Y')  GOTO 20
C
C     ENTER REVISIONS
C
   25 WRITE(10,2100)
 2100 FORMAT(/5X,'REVISIONS TO WIND DATA ONLY? Y OR N > ')
      READ(11,1005,END=25,ERR=25) SELECT
      IF(SELECT.EQ.'Y')  GOTO 100
C
   30 WRITE(10,1004)
 1004 FORMAT(/10X,'DO YOU WISH TO REVISE THE STABILITY?'/
     +        10X,'ENTER Y OR N.')
      READ(11,1005,END=30,ERR=30) SELECT
```

```
 1005 FORMAT(S1)
      IF(SELECT.NE.'Y') GO TO 60
   40 WRITE(10,1006)
 1006 FORMAT(/10X,'ENTER REVISED STABILITY: 1 THROUGH 7 .')
      READ(11,ERR=40,END=40)ISTB
      IF(ISTB.LT.1.OR.ISTB.GT.7) GO TO 40
      DARRAY(4,INDX)=ISTB
C
   60 WRITE(10,1007)
 1007 FORMAT(/10X,'DO YOU WISH TO REVISE THE MIXING DEPTH?'/
     +        10X,'ENTER Y OR N.')
      READ(11,1005,END=60,ERR=60) SELECT
      IF(SELECT.NE.'Y') GO TO 90
   70 WRITE(10,1008)
 1008 FORMAT(/10X,'ENTER REVISED MIXING DEPTH IN TENS OF METERS:'/
     +        10X,' 1 THRU 300 . ',2X)
      READ(11,ERR=70,END=70)LDEPTH
      IF(LDEPTH.LT.1.OR.LDEPTH.GT.300) GO TO 70
      DARRAY(5,INDX)=LDEPTH
C
   90 WRITE(10,1010)
 1010 FORMAT(/10X,'DO YOU WISH TO REVISE WIND DATA?'/
     +        10X,'ENTER Y OR N.')
      READ(11,1005,END=90,ERR=90) SELECT
      IF(SELECT.NE.'Y') GO TO 160
  100 WRITE(10,1011)
 1011 FORMAT(//10X,'ENTER WIND DATA AS DDDFF, WITH 0000=CALM AND'/
     +        10X,'9999=MISSING.  DO NOT USE LT/VAR, USE CALM OR'/
     +          10X,'A GUESS AT THE AVERAGE SPEED AND DIRECTION.'//)
C
 2040 WRITE(10,2050) NSTA
 2050 FORMAT(/5X,'ENTER STATION NUMBER TO BE REVISED ',
     +  '1 - ',I2,'  0 = EXIT > ')
      READ(11,END=2040,ERR=2040) J1
      IF(J1.GT.NSTA)  GOTO 2040
```

```
      IF(J1.LT.1)  GOTO 160
      IX=0
  110 WRITE(10,1013) STNAME(J1)
 1013 FORMAT(/10X,'ENTER REVISED WIND DATA FOR',2X,S4)
      READ(11,ERR=110,END=110)IDDFF
      IF(IDDFF.EQ.0.OR.IDDFF.EQ.9999) GO TO 140
      IDD=IDDFF/100
      IFF=IDDFF-IDD*100
      IDD = IDD * 10
      IF(IDD.LT.0.OR.IDD.GT.360) GO TO 120
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) GO TO 130
      GO TO 140
C
  120 WRITE(10,1014)
 1014 FORMAT(/10X,'WIND DIRECTION OUT OF RANGE')
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) WRITE(10,1015)
 1015 FORMAT(/10X,'WIND SPEED ABNORMALLY HIGH')
      IX=1
      GO TO 110
C
  130 WRITE(10,1015)
      IX=1
      GO TO 110
C
  140 CONTINUE
      DARRAY(J1+5,INDX)=IDDFF
      GOTO 2040
C
```

```
  160 WRITE(10,1016)
 1016 FORMAT(/10X,'DO YOU WISH TO REVISE ANOTHER RECORD?'/
     +       10X,'ENTER Y OR N.')
      READ(11,1005,END=160,ERR=160) SELECT
      IF(SELECT.NE.'Y') GO TO 180
C
  170 WRITE(10,1017)
 1017 FORMAT(/10X,'SAME DAY?  ENTER Y OR N.')
      READ(11,1005,END=170,ERR=170) SELECT
      IF(SELECT.EQ.'Y') GO TO 20
      GO TO 10
C
  175 WRITE(10,1025)
 1025 FORMAT(/10X,'THIS OBSERVATION FILE IS EMPTY.')
  180 CONTINUE
      RETURN
      END
```

```
      OVERLAY SB5
C    **  NUFCST  **  UPDATE 1.1
C
      SUBROUTINE NUFCST
C
      INTEGER DARRAY(35,48), SELECT, YR(48)
C
      REAL STNAME(30)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
C
      NEW=1
C
      Y='Y'
   10 WRITE(10,950)
  950 FORMAT(/10X,'ENTER START DATE OF NEW FORECAST: MM,DD,YY'/
     +       10X,'(E.G. = 7,29,81).')
 1000 FORMAT(/10X,'ENTER DATE OF FORECASTS TO BE ENTERED: MM,DD,YY'/
     +        10X,'(E.G. = 7,29,81).')
      READ(11,ERR=10,END=10)MM,MD,MY
      IF(MY.LT.60.OR.MY.GT.85)  GOTO 10
      IF(MM.LT.1.OR.MM.GT.12) GO TO 10
      IF(MD.LT.1.OR.MD.GT.31) GO TO 10
C
   20 WRITE(10,951)
  951 FORMAT(/10X,'ENTER START HOUR OF NEW FORECAST: HH .')
 1002 FORMAT(/10X,'ENTER HOUR OF THE FORECAST: HH .   99 MAY BE'/
     +        10X,'USED TO ESCAPE FROM THE DATA ENTRY ROUTINE.')
      READ(11,ERR=20,END=20)MH
      IF(MH.EQ.99) RETURN
      IF(MH.LT.1.OR.MH.GT.24) GO TO 20
      CALL JULTAN(MY,MM,MD,MDD)
      WRITE(10,1012) MDD
 1012 FORMAT(/10X,'THE JULTAN DATE OF THE FCST IS',2X,I3,'.')
```

NUFCST continued

```
C
C     ENTER DATES AND TIMES INTO DARRAY
C
      JDD=MDD
      KH=MH+1
      DO 40 I=1,48
        KH=KH+1
        IF(  KH.LE.24) GO TO 30
      KH=1
        JDD=JDD+1
        IF(JDD.LE.365) GO TO 30
        IF(  JDD.EQ.366.AND.MOD(MY,4).EQ.0.AND.MY.NE.0) GO TO 30
      JDD=1
   30   CONTINUE
        YR(I) = MY
        DARRAY(1,I)=JDD
        DARRAY(2,I)=KH
   40 CONTINUE
      GO TO 50
   42 CONTINUE
   44 WRITE(10,1000)
      READ(11,ERR=44,END=44)MM,MD,MY
      IF(MM.LT.1.OR.MM.GT.12) GO TO 44
      IF(MD.LT.1.OR.MD.GT.31) GO TO 44
   46 WRITE(10,1002)
      READ(11,ERR=46,END=46)MH
      IF(MH.EQ.99) RETURN
      IF(MH.LT.1.OR.MH.GT.24) GO TO 46
      CALL JULIAN(MY,MM,MD,MDD)
```

```
   50 CONTINUE
      CALL INDEX(MDD,MH,INDX)
      IF(INDX.NE.0) GO TO 54
      CALL CAL(MY,DARRAY(1,1),MO,IDAY)
      WRITE(10,1030) YR(1),MO,IDAY,DARRAY(2,1)
 1030 FORMAT(/10X,'THE DATE AND TIME IS NOT WITHIN THE 48HR'/
     +        10X,'FORECAST PERIOD.  THE YEAR,MONTH, DAY AND HOUR OF'/
     +        10X,'FORECAST START ARE: ',4I3)
      GO TO 42
   54 CONTINUE
C
C     ENTER STABILITY AND MIXING DEPTH IN DARRAY
C
      IF(NEW.EQ.1) GO TO 60
   55 WRITE(10,1019) DARRAY(4,INDX)
 1019 FORMAT(/10X,'THE CURRENT STABILITY FCST IS',2X,I1,2X, 'DO YOU'/
     +  10X,'WISH TO REVISE THE FCST?  ENTER Y OR N.')
      READ(11,1011,END=55,ERR=55) SELECT
      IF(SELECT.NE.'Y') GO TO 66
   60 WRITE(10,1003)
 1003 FORMAT(/10X,'ENTER STABILITY CLASS: 1 THROUGH 7 .'2X)
      READ(11,ERR=60,END=60)ISTB
      IF(ISTB.LT.1.OR.ISTB.GT.7) GO TO 60
      DO 63 I=INDX,48
        DARRAY(4,I)=ISTB
   63 CONTINUE
C
   66 CONTINUE
      IF(NEW.EQ.1) GO TO 70
   68 WRITE(10,1013) DARRAY(5,INDX)
 1013 FORMAT(/10X,'THE CURRENT MIXING DEPTH FCST IS',2X,I4,'.  DO YOU'/
     +  10X,'WISH TO REVISE THE FCST?  ENTER Y OR N.')
      READ(11,1011,END=68,ERR=68) SELECT
      IF(SELECT.NE.'Y') GO TO 77
```

```
   70 WRITE(10,1004)
 1004 FORMAT(/10X,'ENTER MIXING DEPTH IN TENS OF METERS:'/
     +       10X,' 1 THRU 300 . ',2X)
      READ(11,ERR=70,END=70)LDEPTH
      IF(LDEPTH.LT.1.OR.LDEPTH.GT.300) GO TO 70
      DO 73 I=INDX,48
        DARRAY(5,I)=LDEPTH
   73 CONTINUE
C
C     ENTER WIND OBSERVATION BY STATIONS
C
   77 WRITE(10,1006)
 1006 FORMAT(/10X,'ENTER WIND FCST AS DDFF, WITH 0000=CALM AND'
     +         ,/10X,'9999=MISSING.  DO NOT USE LT/VAR, USE CALM',/
     +      10X,'OR ESTIMATE THE AVERAGE SPEED AND DIRECTION.')
      NSTAP5 = NSTA + 5
      DO 120 J=6,NSTAP5
        J1=J-5
        IX=0
        IF(NE  W.EQ.1) GO TO 80
   78 WRITE(10,1014) STNAME(J1),DARRAY(J,INDX)
 1014   FORMAT(/  10X,'THE CURRENT WIND FCST FOR ',S4,' IS ',I5,'.'/
     + 10X,'DO YOU WISH TO CHANGE IT?  ENTER Y OR N.')
        READ(11,1011,END=78,ERR=78)SELECT
        IF(SELECT.NE.'Y') GO TO 120
   80   WRITE(10,1007) STNAME(J1)
 1007   FORMAT(/10X,'ENTER WIND FOR',2X,S4,2X)
        READ(11,ERR=80,END=80)IDDFF
        IF(IDDFF.EQ.0.OR.IDDFF.EQ.9999) GO TO 110
        IDD=IDDFF/100
        IFF=IDDFF-IDD*100
        IDD = IDD * 10
        IF(IDD.LT.0.OR.IDD.GT.360) GO TO 90
        IF(IX.EQ.0.AND.IABS(IFF).  GT.50) GO TO 100
        GO TO 110
```

```
C
   90    WRITE(10,1008)
 1008    FORMAT(/10X,'WIND DIRECTION OUT OF RANGE')
         IF(IX.EQ.0.AND.IABS(IFF).GT.50) WRITE(10,1009)
 1009    FORMAT(/10X,'WIND SPEED ABNORMALLY HIGH')
         IX=1
         GO TO 80
C
  100    WRITE(10,1009)
         IX=1
         GO TO 80
C
  110    DO 115 I=INDX,48
           DARRAY(J,I)=IDDFF
  115    CONTINUE
  120 CONTINUE
C
  130 WRITE(10,1010)
 1010 FORMAT(/10X,'DO YOU WISH TO ENTER FORECASTS FOR ANOTHER TIME?'/
     +        10X,'ENTER Y OR N.')
      READ(11,1011,END=130,ERR=130) SELECT
 1011 FORMAT(S1)
      IF(SELECT.NE.'Y')  RETURN
        NEW = 1
  150   WRITE(10,160)
  160   FORMAT(/5X,'DO YOU NEED TO REVIEW CURRENT VALUES?',
     +                ' Y OR N',/)
      READ(11,1011,ERR=150,END=150)  SELECT
      IF(SELECT.EQ.'Y')  NEW = 0
      GOTO 42
      END
```

```
      OVERLAY SB6
C    **  RVUFST  **  UPDATE 1.1
C
      SUBROUTINE RVUFST
C
      INTEGER DARRAY(35,48), SELECT,YR(48)
C
      REAL STNAME(30)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
C
      NSTAP5 = NSTA + 5
C
      Y='Y'
   10 WRITE(10,1000)
 1000 FORMAT(/10X,'DO YOU WISH TO REVIEW THE ENTIRE FORECAST DATA FILE?',
     +       /10X,'ENTER Y OR N. ')
      READ(11,1001,END=10,ERR=10) SELECT
 1001 FORMAT(S1)
      IF(SELECT.NE.'Y') GO TO 50
C
C     THIS SECTION PERMITS THE USER TO REVIEW THE ENTIRE FORECAST DATA
C     FILE IN 6 HOUR BLOCKS.
C
      WRITE(10,1002) (STNAME(I),I=1,NSTA)
 1002 FORMAT(/10X,'THE FORECAST FILE WILL BE DISPLAYED IN 6',
     + ' HOUR BLOCKS.',/10X,'   HITTING N<CR> WILL DISPLAY THE',
     + ' NEXT BLOCK.',/10X,'  THE FORECAST ORDER IS: JULIAN DATE,'/
     + 10X,'HOUR, STABILITY, MIXING DEPTH, WINDS.  THE WIND'/
     + 10X,'STATION ORDER IS:',/ 4(10X,8(S4,2X)/)/)
      DO 30 I=1,8
```

```
        II=(I-1)*6+1
        IIP5 = II + 5
        DO 20 J=II,IIP5
          WRITE(10,1003)YR(J),(DARRAY(K,J),K=1,2),(DARRAY(K,J),K=4,NSTAP5)
 1003     FORMAT(3X,I2,I3,2X,I2,2X,I1,2X,I3,5X,8(I5,1X)/3(25X,8(I5,1X)/))
   20   CONTINUE
   22   WRITE(10,2000)
 2000   FORMAT(/5X,'TYPE N <CR> FOR NEXT BLOCK OR JUST <CR> TO EXIT')
        READ(11,1001,END=22,ERR=22) SELECT
        IF(SELECT.NE.'N')  GOTO 50
   30 CONTINUE
C
C      THIS SECTION REVIEWS THE DATA ONE HOUR AT A TIME
C
   50 WRITE(10,1004)
 1004 FORMAT(/10X,'DO YOU WISH TO REVIEW FORECASTS FOR SPECIFIC HOURS?'/
     +        10X,'ENTER Y OR N. ')
      READ(11,1001,END=50,ERR=50) SELECT
      IF(SELECT.NE.'Y') GO TO 100
      IF(DARRAY(1,1).EQ.0) GO TO 95
C
   60 WRITE(10,1005)
 1005 FORMAT(/10X,'ENTER DATE OF FORECAST TO BE REVIEWED: MM,DD,YY'/
     +        10X,'(E.G. - 7,29,81).')
      READ(11,ERR=60,END=60)MM,MD,MY
      IF(MM.LT.1.OR.MM.GT.12) GO TO 60
      IF(MD.LT.1.OR.MD.GT.31) GO TO 60
C
      CALL JULIAN(MY,MM,MD,MDD)
   70 WRITE(10,1007)
 1007 FORMAT(/10X,'ENTER HOUR OF THE FORECAST TO BE REVIEWED: HH .')
      READ(11,ERR=70,END=70)MH
```

```
      IF(MH.LT.1.OR.MH.GT.24) GO TO 70
      CALL INDEX(MOD,MH,INDX)
      IF(INDX.NE.0) GO TO 74
      CALL CAL(MY,DARRAY(1,1),MO,IDAY)
      WRITE(10,1030) YR(1),MO,IDAY,DARRAY(2,1)
 1030 FORMAT(/10X,'THE DATE AND TIME IS NOT WITHIN THE 48HR FORECAST'/
     + 10X,'PERIOD.  THE YEAR,MONTH, DAY AND HOUR OF FORECAST START ARE:'/
     + 10X,4I3)
      GO TO 60
C
   74 WRITE(10,1008)(STNAME(I),I=1,NSTA)
 1008 FORMAT(/10X,'THE FCST IS DISPLAYED IN THE FOLLOWING ORDER:'/
     +       10X,'JULIAN DATE, HOUR, STABILITY, MIXING DEPTH, WINDS.'/
     +       10X,'THE ORDER OF THE WIND FCST IS: '/4(10X,8(S4,2X)/))
 2005 WRITE(10,1003)YR(INDX),(DARRAY(J,INDX),J=1,2),(DARRAY(J,INDX)
     +     ,J=4,NSTAPS)
   72 WRITE(10,2010)
 2010 FORMAT(/5X,'TYPE N <CR> FOR NEXT RECORD OR JUST <CR> TO EXIT')
      READ(11,1001,END=72,ERR=72) SELECT
      IF(SELECT.NE.'N') GOTO 80
        INDX = INDX + 1
        IF(INDX.LE.48)  GOTO 2005
          WRITE(10,2020)
 2020     FORMAT(/5X,'END OF DATA')
C
   80 WRITE(10,1009)
 1009 FORMAT(/10X,'DO YOU WISH TO REVIEW ADDITIONAL HOURS?'/
     +       10X,'ENTER Y OR N. ')
      READ(11,1001,END=80,ERR=80) SELECT
      IF(SELECT.NE.'Y') GO TO 100
   90 WRITE(10,1010)
 1010 FORMAT(/10X,'THE SAME DAY? ENTER Y OR N. ')
      READ(11,1001,END=90,ERR=90) SELECT
      IF(SELECT.EQ.'Y') GO TO 70
      GO TO 60
```

```
C
   95 WRITE(10,1025)
 1025 FORMAT(/10X,'THIS FORECAST FILE IS EMPTY.')
  100 CONTINUE
      RETURN
      END
```

Subroutine REVFST

```
      OVERLAY SB7
C    **  REVFST  **  UPDATE 1.1
C
      SUBROUTINE REVFST
C
      INTEGER DARRAY(35,48), SELECT,YR(48)
C
      REAL STNAME(30)
C
      COMMON /DAT1/DARRAY,YR/NAMES/STNAME,NSTA
C
      Y='Y'
      IF(DARRAY(1,1).EQ.0) GO TO 175
   10 WRITE(10,1000)
 1000 FORMAT(/10X,'ENTER DATE OF FORECAST TO BE REVISED: MM,DD,YY'/
     +       10X,'(E.G. = 7,29,81).')
      READ(11,ERR=10,END=10)MM,MD,MY
      IF(MM.LT.1.OR.MM.GT.12) GO TO 10
      IF(MD.LT.1.OR.MD.GT.31) GO TO 10
      IF(MY.LT.60.OR.MY.GT.85)  GOTO 10
      CALL JULIAN(MY,MM,MD,MDD)
C
      WRITE(10,1002) MDD
 1002 FORMAT(/10X,' THE JULIAN DATE FOR THE REVISION IS ',2X,I3)
   20 WRITE(10,1003)
 1003 FORMAT(/10X,'ENTER HOUR OF THE FORECAST TO BE REVISED: HH .'/
     +       10X,'A 99 CAN BE USED TO ESCAPE FROM THE FCST REVISION'/
     +       10X,'SUBROUTINE.')
      READ(11,ERR=20,END=20)MH
      IF(MH.EQ.99) GO TO 180
      IF(MH.LT.1.OR.MH.GT.24) GO TO 20
 1036 WRITE(10,1018)
```

```
 1018 FORMAT(/10X,'ENTER FORECAST PERSISTENCE: HH . (1 TO 48)')
      READ(11,ERR=1036,END=1036)IP
      IF(IP.LT.1.OR.IP.GT.48)  GOTO 1036
      CALL INDEX(MDD,MH,INDX)
      IF(INDX.NE.0) GO TO 24
      CALL CAL(MY,DARRAY(1,1),MO,IDAY)
      WRITE(10,1030) YR(1),MO,IDAY,DARRAY(2,1)
 1030 FORMAT(/10X,'THE DATE AND TIME IS NOT WITHIN THE 48HR FORECAST'/
     + 10X,'PERIOD.  YEAR MONTH, DAY AND HOUR OF FORECAST START ARE:'/
     + 10X,3I3)
      GO TO 10
C
   24 CONTINUE
      IND=INDX+IP-1
      IF(IND.GT.48) IND=48
C
      NSTAP5 = NSTA + 5
      WRITE(10,2000) YR(INDX),(DARRAY(K,INDX),K=1,2),
     +               (DARRAY(K,INDX),K=4,NSTAP5)
 2000 FORMAT(3X,I2,I3,2X,I2,2X,I1,2X,I3,5X,8(I4,2X)/3(25X,8(I4,2X)/))
 2005 WRITE(10,2010)
 2010 FORMAT(/5X,'IS THIS THE RECORD TO BE REVISED? Y OR N > ')
      READ(11,1005,END=2005,ERR=2005) SELECT
      IF(SELECT.NE.'Y')  GOTO 20
C
C     ENTER REVISIONS
C
   25 WRITE(10,2100)
 2100 FORMAT(/5X,'REVISIONS TO WIND DATA ONLY? Y OR N > ')
      READ(11,1005,END=25,ERR=25) SELECT
      IF(SELECT.EQ.'Y')  GOTO 100
```

```
C
   30 WRITE(10,1004)
 1004 FORMAT(/10X,'DO YOU WISH TO REVISE THE STABILITY?'/
     +       10X,'ENTER Y OR N.')
      READ(11,1005,END=30,ERR=30) SELECT
 1005 FORMAT(S1)
      IF(SELECT.NE.'Y') GO TO 60
C
   40 WRITE(10,1006)
 1006 FORMAT(/10X,'ENTER REVISED STABILITY: 1 THROUGH 7 .')
      READ(11,ERR=40,END=40)ISTB
      IF(ISTB.LT.1.OR.ISTB.GT.7) GO TO 40
      DO 45 I=INDX,IND
        DARRAY(4,I)=ISTB
   45 CONTINUE
C
   60 WRITE(10,1007)
 1007 FORMAT(/10X,'DO YOU WISH TO REVISE THE MIXING DEPTH?'/
     +       10X,'ENTER Y OR N.')
      READ(11,1005,END=60,ERR=60) SELECT
      IF(SELECT.NE.'Y') GO TO 90
   70 WRITE(10,1008)
 1008 FORMAT(/10X,'ENTER REVISED MIXING DEPTH IN TENS OF METERS:'/
     +       10X,' 1 THRU 300 . ',2X)
      READ(11,ERR=70,END=70)LDEPTH
      IF(LDEPTH.LT.1.OR.LDEPTH.GT.300) GO TO 70
      DO 75 I=INDX,IND
        DARRAY(5,I)=LDEPTH
   75 CONTINUE
C
```

```
   90 WRITE(10,1010)
 1010 FORMAT(/10X,'DO YOU WISH TO REVISE WIND FCSTS?'/
     +       10X,'ENTER Y OR N.')
      READ(11,1005,END=90,ERR=90) SELECT
      IF(SELECT.NE.'Y') GO TO 160
  100 WRITE(10,1011)
 1011 FORMAT(//10X,'ENTER WIND FCST AS DDFF, WITH 0000=CALM AND'/
     +       10X,'9999=MISSING.  DO NOT USE LT/VAR, USE CALM OR'/
     +         10X,'A GUESS AT THE AVERAGE SPEED AND DIRECTION.'//)
C
 2040 WRITE(10,2050) NSTA
 2050 FORMAT(/5X,'ENTER STATION NUMBER TO BE REVISED ',
     +  ' 1 - ',I2,' 0 = EXIT  > ')
      READ(11,END=2040,ERR=2040) J1
      IF(J1.GT.NSTA)  GOTO 2040
      IF(J1.LT.1)  GOTO 160
C
      IX=0
  110 WRITE(10,1013) SINAME(J1)
 1013 FORMAT(/10X,'ENTER REVISED WIND FCST FOR',2X,S4)
      READ(11,ERR=110,END=110)IDDFF
      IF(IDDFF.EQ.0.OR.IDDFF.EQ.9999) GO TO 140
      IDD=IDDFF/100
      IFF=IDDFF-IDD*100
      IDD = IDD * 10
      IF(IDD.LT.0.OR.IDD.GT.360) GO TO 120
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) GO TO 130
      GO TO 140
C
  120 WRITE(10,1014)
 1014 FORMAT(/10X,'WIND DIRECTION OUT OF RANGE')
      IF(IX.EQ.0.AND.IABS(IFF).GT.50) WRITE(10,1015)
 1015 FORMAT(/10X,'WIND SPEED ABNORMALLY HIGH')
```

```
      IX=1
      GO TO 110
C
  130 WRITE(10,1015)
      IX=1
      GO TO 110
C
  140 DO 145 I=INDX,IND
        DARRAY(J1+5,I)=IDDFF
  145 CONTINUE
      GOTO 2040
C
  160 WRITE(10,1016)
 1016 FORMAT(/10X,'DO YOU WISH TO ENTER MORE REVISIONS?'/
     +       10X,'ENTER Y OR N.')
      READ(11,1005,END=160,ERR=160) SELECT
      IF(SELECT.NE.'Y') GO TO 180
  170 WRITE(10,1017)
 1017 FORMAT(/10X,'SAME DAY?  ENTER Y OR N.')
      READ(11,1005,END=170,ERR=170) SELECT
      IF(SELECT.EQ.'Y') GO TO 20
      GO TO 10
C
  175 WRITE(10,1029)
 1029 FORMAT(/10X,'THIS FORECAST FILE IS EMPTY.')
  180 CONTINUE
      RETURN
      END
```

PNL-4369
UC-11

DISTRIBUTION

No. of Copies

OFFSITE

27 DOE Technical Information Center

ONSITE

2 DOE Richland Operations Office

H. E. Ramson
D. R. Elle

63 Pacific Northwest Laboratory

K. J. Allwine
G. F. Athey (15)
T. J. Bander
J. G. Droppo
O. P. Gifford
R. K. Hadlock
D. J. Hoitink (15)
C. G. Lindsey
B. J. McMurray
P. M. Potter
J. V. Ramsdell (15)
W. F. Sandusky
M. J. Walters
R. K. Woodruff
Publishing Coordination (2)
Technical Information (5)