

NOAA Technical Memorandum ERL ARL-112

USER'S GUIDE FOR A LONG-RANGE MULTI-LAYER
ATMOSPHERIC TRANSPORT AND DISPERSION MODEL

Roland R. Draxler

Air Resources Laboratories
Silver Spring, Maryland
May 1982



UNITED STATES
DEPARTMENT OF COMMERCE

Malcolm Baldrige,
Secretary

NATIONAL OCEANIC AND
ATMOSPHERIC ADMINISTRATION

John V. Byrne,
Administrator

Environmental Research
Laboratories

George H. Ludwig
Director

TABLE OF CONTENTS

ABSTRACT iv

1. TRANSPORT AND DISPERSION MODEL 1

 1.1 Model Structure 1

 1.2 Advection Calculation Method 2

2. MODEL PARAMETERS 3

 2.1 Input Control Cards 3

 2.2 Program Code Changes 4

3. EXAMPLE OUTPUT 4

4. AVAILABILITY 5

5. ACKNOWLEDGEMENTS 5

6. REFERENCES 6

APPENDIX 11

ABSTRACT

A long-range transport and dispersion model which is responsive to the effects of wind shear is documented in this report. The model is especially suitable for calculating concentrations for puff travel times of greater than one day. The effect of wind shear on dispersion is obtained by dividing the pollutant puff within the mixed layer into 300m sublayers at the beginning of each night. Each layer is tracked by means of a separate trajectory. Vertical mixing is resumed during the next day. Further puff divisions occur each night. Example calculations using the model are presented. Descriptions of necessary input parameters are related to the example calculations. The computer code is listed in the Appendix.

USER'S GUIDE FOR A LONG-RANGE MULTI-LAYER ATMOSPHERIC TRANSPORT AND DISPERSION MODEL

Roland R. Draxler

National Oceanic and Atmospheric Administration
Air Resources Laboratories
6010 Executive Blvd.
Rockville, Maryland 20852

May 1982

1. TRANSPORT AND DISPERSION MODEL

A brief discussion of the model is presented in this report. The long-range transport and dispersion model was described in detail by Draxler (1982). In that paper, model calculations were compared with measured concentrations from the 1974 Midwest Experiment first reported by Ferber et al. (1977). From January to May twice-daily measurements of Kr-85 air concentrations were made at 11 sites along a north-south line about 1500km from the source. Model calculations at the more southern sampling sites compared very favorably with the measured data. Incorporating wind-shear effects in long-range transport calculations was shown to reduce concentration overcalculations from models (as tested by Ferber et al., 1977) that do not consider wind-shear.

The wind-shear effect on a pollutant puff results from the absence of vertical mixing at night. During this time each vertical layer decouples and continues as a separate trajectory. The greater the wind-shear the greater the displacement of these layers from each other. During the daytime vertical mixing resumes and these elevated layers mix to the surface to affect air concentrations. Draxler and Taylor (1982) have shown that this approach to wind-shear simulation using actual meteorological data gives average puff growths that are similar to the theoretical growth of a puff in an Ekman boundary layer.

1.1 Model Structure

The model is essentially a simple Lagrangian puff trajectory model. Twice-daily rawinsonde observations are used to calculate trajectories from a single source at 3-h intervals. However, to efficiently compute the trajectories of many more puffs the position of each element (endpoints, receptors, rawinsondes, etc.) of the calculations is indexed to a three-dimensional grid with a one-degree horizontal resolution and 10 vertical coordinates spaced at 300m increments. In this way, advection and air concentration calculations can be performed quickly without having to search an entire data array for the nearby receptors and rawinsondes.

Trajectory calculations, although time consuming due to the daily exponential increase in the number of trajectories, are not unduly

expensive. Several other features of the computer code, in addition to the indexing by grid position, were designed to increase computational efficiency. For example,

- 1 - Sines and cosines are pre-computed and stored by whole degrees (the resolution of the wind directions);
- 2 - no spatial or temporal interpolation of meteorological data is performed;
- 3 - the masses of two or more puffs at the same grid coordinates are combined; and
- 4 - any individual puffs whose mass has become too small to resolve in the air concentrations are eliminated.

1.2 Advection Calculation Method

Wind shear effects are incorporated by applying the diurnal difference in vertical mixing to the winds in the layer used to calculate advection. If the release is at night, the puff is constrained to the lowest transport layer (0 to 300m). During the day, all puffs are assumed to mix instantaneously to the top of the mixed layer. During the next night, when no vertical mixing is assumed, the mass of a puff is divided into 300m layers within the previous day's mixed layer. The interval of 300m was chosen for the maximum vertical resolution because the wind data are reported at approximately that interval.

After a puff is divided, the layers are followed as separate trajectories for all subsequent calculations. During the next day, the layers become fully mixed to the surface. If the mixed layer during subsequent day-time periods is lower than the previous day some puffs may remain above the new mixed layer. These puffs will be advected with the appropriate wind in that upper layer and will not affect surface air concentrations until the mixed layer reaches that height. Mixed layer heights are determined within the model from the observed temperature soundings.

The separation of these vertical layered puffs simulates the wind shear effect. However, turbulent diffusion still plays a role in spreading the pollutant puff during the first 24 hours before wind shear begins to dominate horizontal dispersion. Therefore, the turbulent dispersion is modeled by assuming that each puff expands with time as defined by Heffter (1965). Concentrations are assumed to be uniform across this disk. The concentration in the disk is then just the mass (variable - QTERM) of the pollutant divided by the cross-sectional area of the disk times its depth. Concentration calculations are performed at the advection time step (3h) at all grid locations (every one-degree latitude and longitude intersection).

2. MODEL PARAMETERS

Most model calculations can be controlled through the input parameters rather than modifying the code. About 150k bytes of computer core are required if the maximum number of puffs is 5000. Computer time requirements also vary according to the number of puffs. Slower wind speeds during the summer months mean that the number of puffs that pass off the computational grid are less than those in winter. Puffs are retained in the calculations until they pass off the grid. On an IBM 360/195, cpu time varies from 1.5 to 3.0 min per month of simulation time.

Unit numbers 5 through 10 are necessary for input and output. Unit 5 is reserved for the input cards to control the simulation. Units 6 through 9 are reserved for printing model output. Input meteorological data are read from unit 10. Meteorological data are available from the National Climatic Center, Digital Products Section, Asheville, North Carolina 28801. Each NCC tape contains 6 months of rawinsonde data and is identified by the time period. The tapes are called NAMED-WINDTEMP and are archived under the identifying number TD-9743.

2.1 Input Control Cards

Only three input cards are necessary to control most model simulations. Additional cards are needed to specify each fixed point receptor. Under most circumstances few other changes are necessary once the code is set to run for a particular region. The three necessary cards contain the parameters:

Card #1 - (variable; columns; format; description)

IBDA	06-09	xxxx	- Beginning day of computations (01-31)
IBMO	11-14	xxxx	- Beginning month of computations (01-12)
NDTR	16-19	xxxx	- Number of days of computations
NAVG	21-24	xxxx	- Number of 3hr periods for concentrations
MAPS	26-29	xxxx	- Flag for short (1) or long (0) map output (80 or 132 columns)
MAPC	31-34	xxxx	- Flag for concentration maps to be printed at NAVG interval (yes -1; no - 0)
MAPT	36-39	xxxx	- Number of 3hr periods between puff maps

Card #2 - (variable; columns; format; description)

OLAT	06-09	xx.x	- Origin latitude (degrees and tenths)
OLON	11-15	xxx.x	- Origin longitude (degrees and tenths)
QTERM	17-21	xxxx.	- Emissions per 3hr puff (arbitrary units)
NQTRM	23-27	xxxxx	- Number of 3hr puffs emitted starting at IBDA
ALATT	29-33	xxx.x	- Top latitude of calculational grid
ALATB	35-39	xxx.x	- Bottom latitude of computational grid
ALONL	41-45	xxx.x	- Left longitude of grid
ALONR	47-51	xxx.x	- Right longitude of grid

Card #3 - (variable; columns; format; description)
NREC 06-09 xxxx - Number of fixed point receptor positions

Card #4 to 'NREC+3' - (variable; columns; format; description)
RLAT 06-09 xx.x - Receptor latitude
RLON 11-15 xxx.x - Receptor longitude

The internal computational grid has been defined to be 25 degrees latitude by 60 degrees longitude. Therefore, the top internal boundary is always 25 degrees greater than the bottom. The left longitude just controls the alignment of all the output maps which are scaled by the top and bottom latitudes. All four boundaries are used to determine when puffs have passed off the computational grid. The intersection of the bottom and right boundaries is the origin (1,1) position of the internal computational grid.

2.2 Program Code Changes

There are some program constants within the code that would usually be changed only when the program is set up for a particular pollutant the first time. These involve the concentration factors needed to derive the correct units. The numeric value of 'CNFACT' determines the units of the input source term. All concentrations are multiplied by 'CNFACT' before printing. This value is determined by the user so that the concentration output will be in whole integers from 1 to 999.

The variable 'PMIN' is the minimum puff mass in units determined by the user (in conjunction with 'CNFACT'). Puffs with less than this mass are eliminated from the calculations after each day.

3. EXAMPLE OUTPUT

The output on unit 6 from a 2 day simulation with concentration and puff maps at 3-hour intervals is illustrated in Fig. 1. The data from the input cards are printed and some of the meteorological data, read every 12-hours, are summarized. These values are read from the input data tape to confirm that the correct meteorological data are being used in the simulation. Printed twice a day are the number of stations reporting, the minimum, mean, and maximum mixing depths (meters) for all stations, and the number of puffs on the computational grid at that time.

A sequence of puff position maps for the example calculation of Fig. 1 is shown in Fig. 2. For the purposes of this illustration the maps have been combined on a single page in which time goes from left to right and down at 3 hour intervals starting at the end of day 1. The time of the map is in the upper left corner of each box. Maps can be printed at the advection interval (3hr) or at any 3hr interval chosen by the user. In this series, each map shown is approximately 500km wide.

The printed digit identifies the layer of the puff. During the day puffs within the mixed layer, ones that intercept the surface to give ground-level air concentrations, are identified by the digit '1'; puffs with a number other than '1' are above the mixed layer and do not contribute to surface air concentrations. Each increment above '1' represents a height increment of 300m. The top layer (2700-3000m) is identified by the zero digit. Note that at the end of the first nocturnal advection step (3Z - 2nd box) the previous day's fully-mixed puffs ('1') are now separating at different levels.

The same sequence of maps for puff concentrations is shown in Fig. 3. Again the time of the map is given in the upper left. The averaging interval is the interval between maps. Concentrations are given by as many as three digits. The units have been adjusted by the variable 'CNFACT'. Concentrations greater than 999 are represented by '###'. The concentrations are printed at all whole degree latitude-longitude grid intersections. This is the current minimum resolution of the model concentration calculations. Fixed receptors are internally defined by the one-degree intersection point. The concentrations shown in Fig. 3 only correspond to puffs in Fig. 2 at the surface identified by '1'. As in the previous example the additional puffs (index .1) at night (3Z) do not contribute to concentration at the surface until the next day (15Z map).

The table of receptor concentrations is shown in Fig. 4. The receptors numbered one to 'NREC' are those given in the input. The concentrations at these points are printed at the interval specified, in this case every three hours. These numbers also correspond with the concentrations shown in Fig. 3 at the appropriate grid intersection. The concentrations at these fixed receptors is taken internally from the computational grid.

4. AVAILABILITY

A copy of the computer code (FORTRAN) is listed in the Appendix. Magnetic tape copies may be obtained from the author.

5. ACKNOWLEDGEMENTS

This work was supported by the U.S. Department of Energy, Office of Health and Environmental Research.

6. REFERENCES

- Draxler R.R. (1982) Measuring and modeling the transport and dispersion of Krypton-85 1500km from a point source, to be published in Atm. Environm.
- Draxler R.R. and Taylor A.D. (1982) Horizontal dispersion parameters for long-range transport modeling, J. Appl. Meteorol., March.
- Ferber G.J. Telegadas K. Heffter J.L. and Smith M.E. (1977) Air concentrations of Krypton-85 in the midwestern United States during January-May 1974, Atm. Environm., 11:379-385.
- Heffter J.L. (1965) The variation of horizontal diffusion parameters with time for travel periods of one hour or longer, J. Appl. Meteorol., 4:153-156.

INPUT PARAMETERS

IBDA IBMO NDTR NAVG MAPS MAPC MAPT

1 1 2 1 0 1 1

OLAT OLON QTERM NQTRM ALATT ALATB ALONL ALONR

40.0 100.0 3000. 16 50.0 25.0 110.0 65.0

NUMBER OF FIXED POINT RECEPTORS

NREC

1

RLAT RLON

43.0 95.0

FINISHED READING METEOROLOGICAL DATA MIXING DEPTH DATA (meters)

MONTH	YEAR	DAY-HR	REPORTS	MIN	MEAN	MAX	PUFFS
JAN	1975	1 12	55	530	1620.	4456	0
JAN	1975	2 0	62	491	1736.	4943	4
JAN	1975	2 12	72	417	1426.	4799	34
JAN	1975	3 0	64	580	1586.	5147	38

Figure 1. Illustration of output from unit 6 for sample calculation. Data from input cards are printed and certain meteorological data each time period are summarized as read from the input tape.

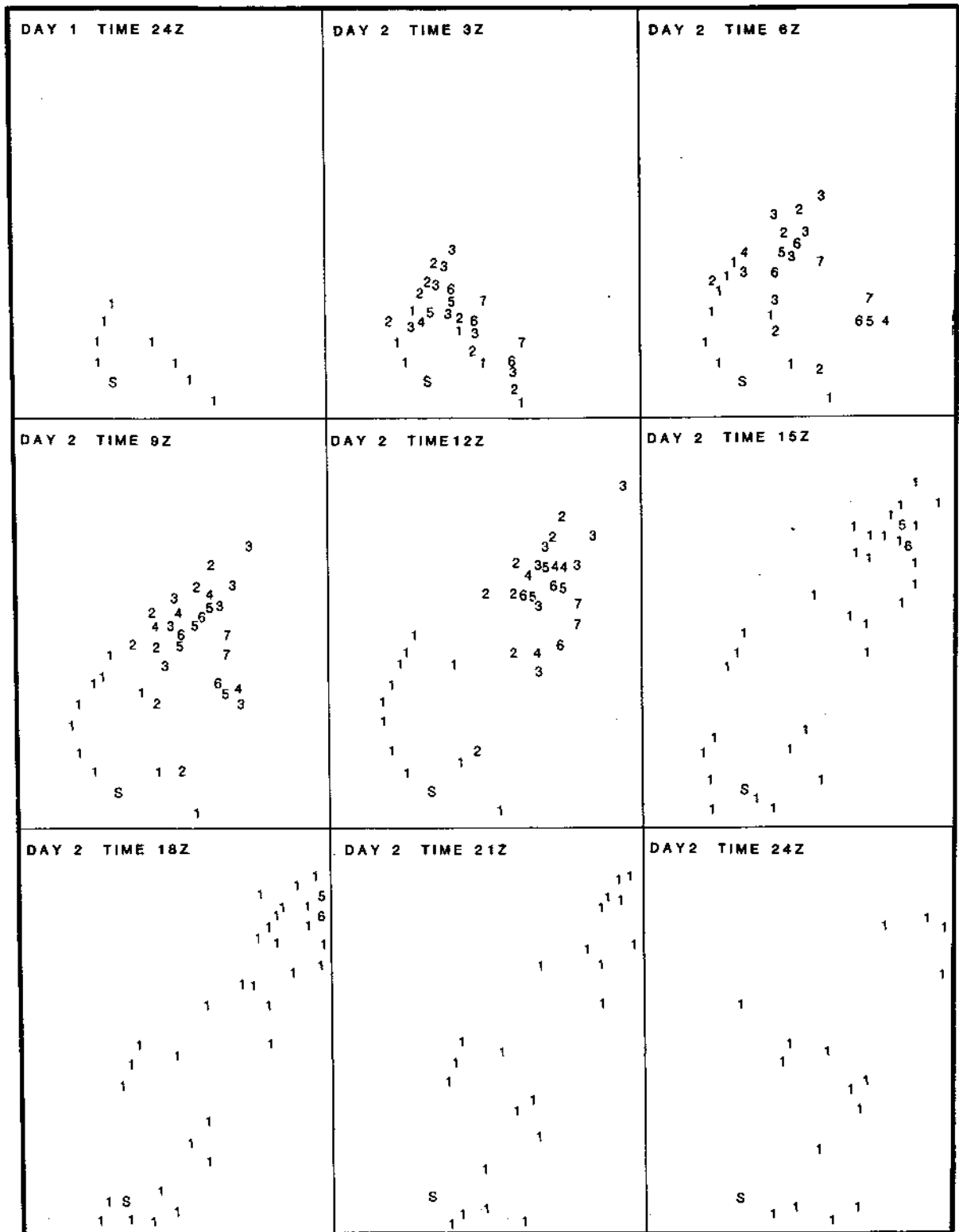


Figure 2. Illustration of output from unit 8 for sample calculation. Shown are the positions of the puffs at three hour intervals. Puffs in the surface layer are identified by the digit '1'. Elevated puffs have digits from 2 to 9. The height is the value of the digit times 300m. The position of the source is indicated by 'S'.

<p>DAY 1 TIME 24Z</p> <p>661</p> <p>***** 212</p> <p>*** 240</p> <p>S 71</p>	<p>DAY 2 TIME 3Z</p> <p>423</p> <p>***</p> <p>*** 147</p> <p>*** 185</p> <p>S 56</p>	<p>DAY 2 TIME 6Z</p> <p>423 294</p> <p>*** 108</p> <p>***</p> <p>*** 142</p> <p>S 45</p>
<p>DAY 2 TIME 9Z</p> <p>216</p> <p>717 83</p> <p>***</p> <p>***</p> <p>*** 112</p> <p>S 37</p>	<p>DAY 2 TIME 12Z</p> <p>381</p> <p>254 294 65</p> <p>***</p> <p>***</p> <p>*** 91 31</p> <p>S 31</p>	<p>DAY 2 TIME 15Z</p> <p>33</p> <p>22 41 52 54</p> <p>28 44 52</p> <p>20 15 13</p> <p>7 12 5</p> <p>33 9 12 3</p> <p>24 4</p> <p>4</p> <p>127 18 12 13</p> <p>198 13</p> <p>353 S *** 9</p>
<p>DAY 2 TIME 18Z</p> <p>57 31 76</p> <p>27 60 43</p> <p>6 20 47 2</p> <p>14 6 5 16 2</p> <p>6 8 2</p> <p>15 12 5 4</p> <p>18 4 4</p> <p>10 12</p> <p>10 12</p> <p>588 10 8</p> <p>S</p> <p>101 159 *****</p>	<p>DAY 2 TIME 21Z</p> <p>9 54</p> <p>27 54</p> <p>10 19 13</p> <p>19</p> <p>4 2</p> <p>21 4 3</p> <p>15 4 9 3</p> <p>12 7 3</p> <p>12 3 3</p> <p>331 3 3 3</p> <p>S</p> <p>*****</p>	<p>DAY 2 TIME 24Z</p> <p>16</p> <p>20 16</p> <p>16</p> <p>8 3</p> <p>36 6 11 3</p> <p>8 12 17 9</p> <p>9 17 9</p> <p>212 6 6 6</p> <p>S ***** 661</p>

Figure 3. Illustration of the output from unit 9 for the sample calculation. Shown are the 3-hour average concentrations. Only puffs in the surface layer (digit - 1 in Fig. 2) contribute to air concentrations.

END SAMPLING		RECEPTORS
DAY	HR	1
1	3	0.0
1	6	0.0
1	9	0.0
1	12	0.0
1	15	0.0
1	18	0.0
1	21	0.0
1	24	0.0
2	3	0.0
2	6	0.0
2	9	0.0
2	12	0.0
2	15	0.0
2	18	3.9
2	21	8.9
2	24	12.3

Figure 4. Illustration of output from unit 7 for the sample calculation. Shown are the 3-hour concentrations (arbitrary units) at a fixed receptor identified in the input cards shown in Fig. 1.

APPENDIX

Listing of Code

```

00000100C-----
00000200C*****MAIN PROGRAM FOR LONG-RANGE WIND SHEAR TRAJECTORIES
00000300      INTEGER*2 IDSTN(25,60)
00000400      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00000500      COMMON/MSTNS/IDSTN,DUMMY(1800)
00000600      COMMON/CON/TS,CNFACT
00000700      COMMON/RCPEX/RLAT(10),RLON(10),NREC
00000800      DATA IMAPS/0/
00000900C-----CONCENTRATION FACTOR TO GIVE INTEGER UNITS ON OUTPUT
00001000      CNFACT=1.0E+12
00001100C-----MINIMUM PUFF MASS (ARBITRARY UNITS)
00001200      PMIN=1.0
00001300C-----PROGRAM CONTROL INPUT PARAMETERS
00001400C      IBDA - BEGINING DAY OF COMPUTATIONS
00001500C      IBMO - BEGINING MONTH OF COMPUTATIONS
00001600C      NDTR - NUMBER OF DAYS OF COMPUTATIONS
00001700C      NAVG - NUMBER OF 3HR CONCENTRATION AVERAGES
00001800C      MAPS - FLAG FOR SHORT OR LONG MAP OUTPUT (80 OR 132)
00001900C           0 - 132 COLUMN PRINTER; 1 - 80 COLUMN PRINTER
00002000C      MAPC - CONCENTRATION MAP FLAG (0/1) AT NAVG INTERVAL
00002100C      MAPT - NUMBER OF 3HR PERIODS BETWEEN PUFF MAPS
00002200C      OLAT - ORIGIN LATITUDE (DEG-TENTHS)
00002300C      OLON - ORIGIN LONGITUDE (DEG-TENTHS)
00002400C      QTERM- EMISSIONS PER 3 HOUR PUFF
00002500C      NQTRM- NUMBER OF 3 HOUR PUFFS EMITTED
00002600C      ALATT- TOP LATITUDE OF GRID (DEG-TENTHS)
00002700C      ALATB- BOTTOM LATITUDE ...
00002800C      ALONL- LEFT LONGITUDE ...
00002900C      ALONR- RIGHT LONGITUDE ...
00003000C-----INPUT/OUTPUT FILE REQUIREMENTS
00003100C      FT05F001 - PROGRAM CONTROL INPUT PARAMETERS
00003200C      FT06F001 - PRINTER OUTPUT FOR RUN STATISTICS
00003300C      FT07F001 - PRINTER OUTPUT FOR FIXED SAMPLING SITES
00003400C      FT08F001 - PRINTER OUTPUT FOR PUFF POSITION MAPS
00003500C      FT09F001 - PRINTER OUTPUT FOR CONCENTRATION MAPS
00003600C      FT10F001 - INPUT METEOROLOGICAL DATA
00003700C           NAMER-WINDTEMP TAPES (TD-9743)
00003800C           N.C.C., DIGITAL PRODUCTS
00003900C           ASHVILLE, NC 28801
00004000      WRITE(6,20)
00004100      20 FORMAT(1X,'INPUT PARAMETERS')
00004200      WRITE(6,25)
00004300      25 FORMAT(6X,'IBDA IBMO NDTR NAVG MAPS MAPC MAPT')
00004400      READ(5,50)IBDA,IBMO,NDTR,NAVG,MAPS,MAPC,MAPT
00004500      50 FORMAT(5X,7(I4,1X))
00004600      WRITE(6,60)IBDA,IBMO,NDTR,NAVG,MAPS,MAPC,MAPT
00004700      60 FORMAT(6X,7(I4,1X))
00004800      WRITE(6,75)
00004900      75 FORMAT(6X,'OLAT OLON QTERM NQTRM ALATT ALATB ALONL',
00004910      * 'ALONR')
00005000      READ(5,100)OLAT,OLON,QTERM,NQTRM,ALATT,ALATB,ALONL,ALONR
00005100      100 FORMAT(5X,F4.1,1X,F5.1,1X,F5.0,1X,I5,4F6.1)
00005200      WRITE(6,120)OLAT,OLON,QTERM,NQTRM,ALATT,ALATB,ALONL,
00005210      * ALONR
00005300      120 FORMAT(6X,F4.1,1X,F5.1,1X,F5.0,1X,I5,4F6.1)
00005400      WRITE(6,110)
00005500      110 FORMAT(' NUMBER OF FIXED POINT RECEPTORS',/,6X,'NREC')
00005600      READ(5,125)NREC
00005700      125 FORMAT(5X,I4)
00005800      WRITE(6,126)NREC
00005900      126 FORMAT(6X,I4)
00006000      IF(NREC.EQ.0)GO TO 149
00006100      WRITE(6,128)
00006200      128 FORMAT('          RLAT RLON')

```

```

00006300      DO 140 I=1,NREC
00006400      READ(5,130)RLAT(I),RLON(I)
00006500 130      FORMAT(5X,F4.1,1X,F5.1)
00006600      WRITE(6,135)RLAT(I),RLON(I)
00006700 135      FORMAT(6X,F4.1,1X,F5.1)
00006800 140      CONTINUE
00006900      WRITE(7,144)(I,I=1,NREC)
00007000 144      FORMAT(' END SAMPLING RECEPTORS',/, ' DAY HR',5X,10I5)
00007100      WRITE(7,146)
00007200 146      FORMAT(' ')
00007300 149      CALL CNZERO
00007400      CALL MLOT(MAPS)
00007500      CALL INITLZ(NPUFF)
00007600      CALL PSTNTP(IBMO,IBDA)
00007700      CALL SKIP(2)
00007800      WRITE(6,150)
00007900 150      FORMAT(1H0,' FINISHED READING METEOROLOGICAL DATA',14X,
00008000 *      ' MIXING DEPTH DATA (M)',/,6X,' MONTH',2X,' YEAR',1X,
00008100 *      ' DAY-HR',18X,' REPORTS',4X,
00008200 *      ' MIN',1X,' MEAN',3X,' MAX',5X,' PUFFS')
00008300      IF(NAVG.EQ.1)CALL CNZERO
00008400          DO 1000 I=1,NDTR
00008500              DO 900 ITIME=3,24,3
00008600                  IF(IMAPS.GE.MAPT)IMAPS=0
00008700                  IMAPS=IMAPS+1
00008800                  IF(IFIX(TS).GE.NAVG)CALL CNZERO
00008900                  TS=TS+1.0
00009000                  IGMT=ITIME-3
00009100                  CALL RDMTDT(NPUFF,IGMT,&800)
00009200                  CALL ASSIGM(IDSTN)
00009300 800          CALL ADVCTN(NPUFF,IGMT,OLAT,OLON,OTERM,NOTRM)
00009400                  IF(IMAPS.GE.MAPT)CALL PPLOT(I,ITIME,NPUFF,
00009410 *                  OLAT,OLON)
00009500 *                  IF(MAPC.EQ.1.AND.IFIX(TS).GE.NAVG)CALL CNPLOT
00009600 *                  (I,ITIME,OLAT,OLON)
00009700                  IF(IFIX(TS).GE.NAVG)CALL CNPRT(I,ITIME)
00009800 900          CONTINUE
00009900                  CALL PCNVRG(NPUFF)
00010000                  CALL PSORT(NPUFF,PMIN)
00010100                  CALL PSPLIT(NPUFF)
00010200 1000         CONTINUE
00010300      STOP
00010400      END
00010500C-----
00010600C*****SUB TO READ NAMER WIND-TEMP TAPE
00010700C      SET TO READ ONLY 0 OR 12 GMT DATA (IGMT)
00010800      SUBROUTINE RDMTDT(NPUFF,IGMT,*)
00010900      INTEGER*2 IZHT(20),IDIR(20),ISPD(20),ITEMP(20),IPPP(20),
00011000 A      DUM1(20),DUM2(20)
00011100      COMMON/PRFL/IZHT,IDIR,ISPD,ITEMP,IPPP,DUM1,DUM2
00011200      IF(IGMT.EQ.0.OR.IGMT.EQ.12)GO TO 10
00011300      IF(IGMT.EQ.6.OR.IGMT.EQ.18)CALL SKIP(1)
00011400      RETURN 1
00011500 10      NSTA=0
00011600          NTEMP=0
00011700          AVGMX=0.0
00011800          MINMX=3000
00011900          MAXMX=0
00012000          CALL INITLZ
00012100              DO 300 I=1,2
00012200                  READ(10,30,END=400,ERR=412)MON,IYR,IDY,IHR,NREPT
00012300 30          FORMAT(A3,I4,I2,I2,I4)
00012400                  IF(NREPT.EQ.0)GO TO 300
00012500                  DO 200 J=1,NREPT
00012600                      READ(10,40,END=400,ERR=412)LATSTA,LONSTA,
00012610 *                      IHTSTA,NLVLS
00012700 40          FORMAT(5X,I5,I7,I5,5X,I2)
00012800                      IF(NLVLS.EQ.0)GO TO 200
00012900                      DO 100 K=1,NLVLS
00013000                          GO TO(50,75),I
00013100 50          READ(10,45,END=400,ERR=400)IHT,IWDR,IWSP
00013200 45          FORMAT(I5,I3,I4)
00013300                      IF(K.GT.20)GO TO 100
00013400                      IZHT(K)=MAX0(IHT-IHTSTA,0)

```

```

00013500      ISPD(K)=IWSP/10.0+0.5
00013600      IDIR(K)=IWDR
00013700      IF(IDIR(K).EQ.0)IDIR(K)=360
00013800      GO TO 100
00013900      75 READ(10,47,END=400,ERR=400)IHT,IPRES,IDEG
00014000      47 FORMAT(I4,I5,I4)
00014100      IF(K.GT.20)GO TO 100
00014200      IZHT(K)=MAX0(IHT-IHTSTA,0)
00014300      IPPP(K)=IPRES/10.0+0.5
00014400      ITEMP(K)=IDEG
00014500      100 CONTINUE
00014600      IF(NLVLS.GT.20)NLVLS=20
00014700      IF(NLVLS.LE.1)GO TO 200
00014800      STALAT=LATSTA/100.0
00014900      STALON=LONSTA/100.0
00015000      CALL INDEX1(STALAT,STALON,ILAT,JLON,IREM,&200)
00015100      GO TO(110,120),I
00015200      110 CALL WBANW(NSTA,ILAT,JLON)
00015300      CALL WNDTMP(NLVLS,NSTA)
00015400      GO TO 200
00015500      120 CALL WBANT(NSTA,ILAT,JLON,&200)
00015600      CALL TMPRFL(NLVLS,NSTA,MIX)
00015700      NTEMP=NTEMP+1
00015800      AVGMX=AVGMX+MIX
00015900      MINMX=MIN0(MINMX,MIX)
00016000      MAXMX=MAX0(MAXMX,MIX)
00016100      200 CONTINUE
00016200      300 CONTINUE
00016300      CALL WBANF
00016400      AVGMX=AVGMX/NTEMP
00016500      WRITE(6,550)MON,IYR,IDY,IHR,NTEMP,MINMX,AVGMX,MAXMX,
00016510      * NPUFF
00016600      550 FORMAT(6X,A3,3X,I4,2(2X,I2),18X,I5,5X,I4,1X,F5.0,1X,I4,
00016610      * 5X,I5)
00016700      GO TO 420
00016800      400 WRITE(6,410)
00016900      410 FORMAT(1H0,'EOF ON UNIT 10 FROM RDMTDT')
00017000      STOP
00017100      412 WRITE(6,414)
00017200      414 FORMAT(1H0,'I/O ERROR ON UNIT 10 FROM RDMTDT')
00017300      STOP
00017400      420 RETURN
00017500      END
00017600C-----
00017700C*****SUB TO POSITION TAPE TO FIRST INPUT DAY
00017800C      TAPES HAVE TWO FILES PER MONTH
00017900C      DAYS 1 - 15 AND DAYS 16 TO END OF MONTH
00018000      SUBROUTINE PSTNTP(IBMO,IBDA)
00018100      IF(IBDA.EQ.1.OR.IBDA.EQ.16)RETURN
00018200      ISKIP=(IBDA-1)*4
00018300      IF(IBDA.GE.16)ISKIP=(IBDA-16)*4
00018400      ENTRY SKIP(ISKIP)
00018500      DO 100 I=1,ISKIP
00018600      DO 100 J=1,2
00018700      READ(10,20,END=110,ERR=130)NREC
00018800      20  FORMAT(15X,I5)
00018900      IF(NREC.EQ.0)GO TO 100
00019000      DO 90 K=1,NREC
00019100      READ(10,30,END=110,ERR=130)
00019200      30  FORMAT(1X,I5)
00019300      90  CONTINUE
00019400      100 CONTINUE
00019500      RETURN
00019600      110 WRITE(6,120)
00019700      120 FORMAT(1H0,'EOF UNIT 10 FROM PSTNTP')
00019800      STOP
00019900      130 WRITE(6,140)
00020000      140 FORMAT(1H0,'I/O ERROR ON UNIT 10 FROM PSTNTP')
00020100      STOP
00020200      END
00020300C-----
00020400C*****SUB TO ADD SEQUENCE NUMBER OF METEOROLOGICAL STATION
00020500C*****TO DATA GRID (INDEXED BY POSITION IN WHOLE DEGREES)
00020600C      POSITION SEQUENCE NUMBER IDENTIFIES ARRAY LOCATION

```



```

00020700C      EACH VARIABLE IN MSTNS COMMON BLOCK
00020800      SUBROUTINE WBANK
00020900      INTEGER*2 IDSTN(25,60)
00021000      COMMON/MSTNS/IDSTN,DUMMY(1800)
00021100      ENTRY WBANW(NSTA,ILAT,JLON)
00021200      NSTA=NSTA+1
00021300      IDSTN(ILAT,JLON)=NSTA
00021400      RETURN
00021500      ENTRY WBANT(NSTA,ILAT,JLON,*)
00021600      IF(IDSTN(ILAT,JLON).EQ.0)RETURN 1
00021700      NSTA=IDSTN(ILAT,JLON)
00021800      IDSTN(ILAT,JLON)=-IDSTN(ILAT,JLON)
00021900      RETURN
00022000      ENTRY WBANF
00022100          DO 100 I=1,25
00022200          DO 100 J=1,60
00022300          IF(IDSTN(I,J).LE.0)GO TO 100
00022400          IDSTN(I,J)=0
00022500      100 CONTINUE
00022600      RETURN
00022700      END
00022800C-----
00022900C*****SUB TO PUT WIND SOUNDING IN MET DATA ARRAY
00023000C      IN COMMON BLOCK MSTNS
00023100      SUBROUTINE WNDTMP(NLVLS,NSTA)
00023200      INTEGER*2 IZHT(20),IDIR(20),ISPD(20),DUM1(20),ITEMP(20)
00023300      INTEGER*2 DUM2(20),IDSTN(25,60),IPPP(20)
00023400      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100)
00023500      COMMON/PRFL/IZHT,DIR,ISPD,ITEMP,IPPP,DUM1,DUM2
00023600      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00023700      IDF=0
00023800          DO 100 I=1,NLVLS
00023900          IDI=IZHT(I)/300+1
00024000          IF(IDI.GT.10)GO TO 100
00024100          IF(IDI.EQ.IDF)GO TO 100
00024200          IDF=IDI
00024300          WDIR(NSTA,IDF)=IDIR(I)
00024400          WSPD(NSTA,IDF)=ISPD(I)
00024500      100 CONTINUE
00024600      RETURN
00024700      END
00024800C-----
00024900C*****SUB TO DETERMINE INVERSION HEIGHT AT EACH RAWINSONDE
00025000C      AND PUT MIXH IN COMMON BLOCK MSTNS
00025100      SUBROUTINE TMPRFL(NLVLS,NSTA,MIX)
00025200      INTEGER*2 IZHT(20),ITEMP(20),DUM2(20),IDIR(20),ISPD(20),
00025300      * DUM1(20),WDIR(100,10),WSPD(100,10),MIXH(100),
00025310      * IPPP(20),IDSTN(25,60)
00025400      INTEGER*4 LB(2),LE(2)
00025500      COMMON/PRFL/IZHT,DIR,ISPD,ITEMP,IPPP,DUM1,DUM2
00025600      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00025700      DZL=5.0
00025800      10 IFLG=0
00025900          T1=ITEMP(1)/10.0*(1000.0/IPPP(1))*0.286
00026000          DO 100 I=2,NLVLS
00026100          T2=ITEMP(I)/10.0*(1000.0/IPPP(I))*0.286
00026200          IF(IZHT(I-1).LT.300)GO TO 90
00026300          IF(IZHT(I)-IZHT(I-1).LE.0)GO TO 100
00026400          DTDZ=(T2-T1)/(IZHT(I)-IZHT(I-1))*1000.0
00026500          IF(DTDZ.LT.DZL)GO TO 90
00026600          IF(IFLG.EQ.0)DT=0.0
00026700          IFLG=1
00026800          DT=DT+(T2-T1)
00026900          IF(DT.GT.2.0)GO TO 200
00027000          T1=T2
00027100          GO TO 100
00027200      90 T1=T2
00027300          IFLG=0
00027400      100 CONTINUE
00027500          DZL=DZL-1.0
00027600          IF(DZL.GE.1.0)GO TO 10
00027700          MIX=2999
00027800          GO TO 210
00027900      200 MIX=IZHT(I)

```

```

00028000 210 MIXH(NSTA)=MINO(MIX,2999)
00028100 RETURN
00028200 END
00028300C-----
00028400C*****SUB TO COMPUTE HORIZONTAL AVERAGED WIND FOR ADVECTION
00028500C NOTE THAT AFTER STMT 60 THE SECTION IS SET TO SCAN
00028600C SOUNDING IF THERE ARE MISSING DATA AT A LEVEL
00028700C OTHERWISE NO ADVECTION IS CALCULATED ON THE 3RD PASS
00028800 SUBROUTINE WINDWT(NWIND,LBT,LTT,XTS,YTS,*)
00028900 INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
00028910 * IDSTN(25,60)
00029000 REAL*4 SIN360(360),COS360(360),COSLAT(25)
00029100 COMMON/TRANS/COS360,SIN360,COSLAT
00029200 COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00029300 MIX=MIXH(NWIND)/300+1
00029400 MIX=MINO(MIX,10)
00029500 IC=0
00029600 LB=LBT
00029700 LT=LTT
00029800 10 XAW=0.0
00029900 YAW=0.0
00030000 INW=0
00030100 DO 50 K=LB,LT
00030200 ISPD=WSPD(NWIND,K)
00030300 IDIR=WDIR(NWIND,K)
00030400 IF(IDIR.LT.0.OR.ISPD.LT.0)GO TO 50
00030500 XAW=XAW-ISPD*SIN360(IDIR)
00030600 YAW=YAW-ISPD*COS360(IDIR)
00030700 INW=INW+1
00030800 50 CONTINUE
00030900 IF(INW.NE.0)GO TO 100
00031000 IC=IC+1
00031100 GO TO(60,70,200),IC
00031200 60 IF(LB.GT.MIX)GO TO 65
00031300 LB=1
00031400 GO TO 10
00031500 65 LB=MINO(MIX+1,10)
00031600 GO TO 10
00031700 70 IF(LB.GT.MIX)GO TO 75
00031800 LT=MIX
00031900 GO TO 10
00032000 75 LT=10
00032100 GO TO 10
00032200 100 XAW=XAW/INW
00032300 YAW=YAW/INW
00032400 XTS=1.94*XAW*3.0
00032500 YTS=1.94*YAW*3.0
00032600 RETURN
00032700 200 RETURN 1
00032800 END
00032900C-----
00033000C*****SUBROUTINE TO CALCULATE TRAJECTORY ADVECTION
00033100C AND START NEW PUFFS WHICH ARE PUT IN COMMON BLOCK
00033200C CONTAINING MASS, POSITION, INDEX, AND DIAMETER
00033300C SUBROUTINE ADVCTN(NPUFF,IGMT,OLAT,OLON,QTERM,NQTRM)
00033400 INTEGER*2 POSTN(05000),ARRAY(05000),IDSTN(25,60)
00033500 INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
00033510 * SIGMA(05000)
00033600 REAL*4 MASS(05000),COS360(360),SIN360(360),COSLAT(25)
00033700 COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
00033800 COMMON/TRANS/COS360,SIN360,COSLAT
00033900 COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00034000 DATA IQCNT/0/
00034100 CALL INDEX1(OLAT,OLON,ILAT,JLON,JREM,&700)
00034200 IQCNT=IQCNT+1
00034300 IF(IQCNT.GT.NQTRM)GO TO 200
00034400 NPUFF=NPUFF+1
00034500 MASS(NPUFF)=QTERM
00034600 POSTN(NPUFF)=JREM
00034700 ARRAY(NPUFF)=ILAT*1000+JLON*10+1
00034800 SIGMA(NPUFF)=0
00034900 200 IF(NPUFF.EQ.0)GO TO 600
00035000 DO 500 L=1,NPUFF
00035100 IF(MASS(L).EQ.0.0)GO TO 500

```

```

00035200      IT1=ARRAY(L)/1000
00035300      JT1=ARRAY(L)/10-IT1*100
00035400      KT1=ARRAY(L)-ARRAY(L)/10*10
00035500      NW1=IDSTN(IT1,JT1)
00035600      NW1=IABS(NW1)
00035700      IF(NW1.EQ.0)GO TO 375
00035800      SIGMA(L)=SIGMA(L)+5
00035900      CALL LEVEL(NW1,KT1,IGMT,LB,LT)
00036000  300    CALL WINDWT(NW1,LB,LT,XTS,YTS,&325)
00036100      IREM=POSTN(L)
00036200      CALL INDEX2(IREM,IT1,JT1,TLAT1,TLON1)
00036300      TLAT2=TLAT1+YTS/60.0
00036400      TLON2=TLON1-(XTS/(60.0*COSLAT(IT1)))
00036500      CALL INDEX1(TLAT2,TLON2,IT2,JT2,IREM,&375)
00036600      POSTN(L)=IREM
00036700      ARRAY(L)=IT2*1000+JT2*10+KT1
00036800      IT1=IT2
00036900      JT1=JT2
00037000  325    TSIGMA=SIGMA(L)*1000.0
00037100      IF(LB.EQ.1)CALL CNSUM(IGMT,LT,MASS(L),IT1,JT1,TSIGMA)
00037200      GO TO 500
00037300  375    MASS(L)=0.0
00037400  500    CONTINUE
00037500  600    RETURN
00037600  700    WRITE(6,800)
00037700  800    FORMAT(1H0,'ERROR ORIGIN POSITION ')
00037800      STOP
00037900      END
00038000C-----
00038100C*****SUB TO SELECT BOTTOM AND TOP OF ADVECTION LAYER
00038200C      INDIVIDUAL LEVELS AT NIGHT AND ABOVE MIXED LAYER
00038300C      AND THE AVERAGE WITHIN THE MIXED LAYER DURING THE DAY
00038400      SUBROUTINE LEVEL(NW1,KT1,IGMT,LB,LT)
00038500      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
00038510      * IDSTN(25,60)
00038600      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00038700      LB=KT1
00038800      IF(LB.EQ.0)LB=10
00038900      LT=KT1
00039000      IF(LT.EQ.0)LT=10
00039100      MIX=MIXH(NW1)/300+1
00039200      MIX=MIN0(MIX,10)
00039300      IF(LB.GT.MIX)GO TO 100
00039400      IF(IGMT.LT.12)GO TO 100
00039500      LB=1
00039600      LT=MIX
00039700  100    KT1=MOD(LB,10)
00039800      RETURN
00039900      END
00040000C-----
00040100C*****SUB TO ARRANGE FOR PLOTTING DAILY PUFF POSITIONS
00040200C      PUFFS THAT CONTRIBUTE TO SURFACE CONCENTRATION
00040300C      ARE INDICATED BY INDEX 1, OTHERS HAVE LEVEL NUMBERS
00040400C      THAT ARE MULTIPLES OF 300 METERS
00040500      SUBROUTINE PPLOT(IDAY,ITIME,NPUFF,OLAT,OLON)
00040600      INTEGER*2 POSTN(05000),ARRAY(05000),SIGMA(05000)
00040700      LOGICAL*1 G,IJOFF,NUM(10)
00040800      REAL*4 MASS(05000)
00040900      COMMON/MAPVAL/XGPD,YGPD,TALATT
00041000      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00041100      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
00041200      COMMON/MAPGRD/G(130,80)
00041300      DATA NUM/'1','2','3','4','5','6','7','8','9','0'/
00041400      T(ALAT)=ALOG(TALATT/TAN(0.785+ALAT*0.00873))
00041500      IJOFF(I,J)=I.LT.1.OR.I.GT.130.OR.J.LT.1.OR.J.GT.80
00041600      CALL PLTZRO
00041700      IF(NPUFF.EQ.0)GO TO 600
00041800      DO 500 L=1,NPUFF
00041900      IF(MASS(L).EQ.0.0)GO TO 500
00042000      IREM=POSTN(L)
00042100      ITRAJ=ARRAY(L)/1000
00042200      JTRAJ=ARRAY(L)/10-ITRAJ*100
00042300      KTRAJ=ARRAY(L)-ARRAY(L)/10*10
00042400      IF(KTRAJ.EQ.0)KTRAJ=10

```

```

00042500      CALL INDEX2(IREM,ITRAJ,JTRAJ,TLAT,TLON)
00042600      IX=(ALONL-TLON)*XGPD+0.5
00042700      JX=T(TLAT)*YGPD+0.5
00042800      IF(IJOFF(IX,JX))GO TO 500
00042900      G(IX,JX)=NUM(KTRAJ)
00043000      CONTINUE
00043100      500
00043200      600 CALL PLTSRC(OLAT,OLON)
00043300      CALL DO PLOT(8,IDAY,ITIME)
00043400      RETURN
00043500      END
00043600C-----
00043700C*****SUB TO ARRANGE FOR PLOTTING DAILY CONCENTRATION MAPS
00043800C      VARIABLE TS IS THE SAMPLING COUNTER, INCREMENTED EACH
00043900C      ADVECTION STEP. THE CONCENTRATION FACTOR OF E+12 IS
00044000C      APPLIED BEFORE PLOTTING TO GET WHOLE NUMBERS
00044100      SUBROUTINE CNCON
00044200      REAL*4 CONC(25,60),COSLAT(25),SIN360(360),COS360(360)
00044300      DIMENSION FXCON(10)
00044400      LOGICAL*1 G,IJOFF,NUM(10),AST,BLANK,OVRPRT
00044500      COMMON/MAPVAL/XGPD,YGPD,TALATT
00044600      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00044700      COMMON/TRANS/COS360,SIN360,COSLAT
00044800      COMMON/MAPGRD/G(130,80)
00044900      COMMON/CON/TS,CNFACT
00045000      COMMON/RCPFX/RLAT(10),RLON(10),NREC
00045100      DATA YD/111180.0/
00045200      DATA NUM/'1','2','3','4','5','6','7','8','9','0'/
00045300      DATA AST/'*','/','BLANK/'/'/'/
00045400      T(ALAT)=ALOG(TALATT/TAN(0.785+ALAT*0.00873))
00045500      *OVRPRT(I,J)=G(I-1,J).NE.BLANK.OR.
00045600      G(I,J).NE.BLANK.OR.G(I+1,J).NE.BLANK
00045700      IJOFF(I,J)=I.LT.1.OR.I.GT.130.OR.J.LT.1.OR.J.GT.80
00045800      ENTRY CNZERO
00045900      DO 100 I=1,25
00046000      DO 100 J=1,60
00046100      CONC(I,J)=0.0
00046200      100 CONTINUE
00046300      TS=0.0
00046400      RETURN
00046500      ENTRY CNSUM(IGMT,LT,TMASS,ITRAJ,JTRAJ,TSIGMA)
00046600      ZD=300.0
00046700      IF(IGMT.GE.12)ZD=300*LT
00046800      ISCAN=(2.0*TSIGMA)/YD
00046900      JSCAN=(2.0*TSIGMA)/(YD/COSLAT(ITRAJ))
00047000      IMIN=MAX0(ITRAJ-ISCAN,1)
00047100      IMAX=MIN0(ITRAJ+ISCAN,25)
00047200      JMIN=MAX0(JTRAJ-JSCAN,1)
00047300      JMAX=MIN0(JTRAJ+JSCAN,60)
00047400      DO 150 I=IMIN,IMAX
00047500      DO 150 J=JMIN,JMAX
00047600      CONC(I,J)=CONC(I,J)+TMASS/(12.6*ZD*TSIGMA*TSIGMA)
00047700      150 CONTINUE
00047800      RETURN
00047900      ENTRY CNPLOT(IDAY,ITIME,OLAT,OLON)
00048000      CALL PLTZRO
00048100      IF(TS.EQ.0.0)GO TO 500
00048200      DO 400 IL=1,25
00048300      DO 400 JL=1,60
00048400      SLAT=IL+(ALATB-1.0)+0.5
00048500      SLON=JL+(ALONR-1.0)+0.5
00048600      I=(ALONL-SLON)*XGPD+0.5
00048700      J=T(SLAT)*YGPD+0.5
00048800      IF(IJOFF(I-1,J))GO TO 400
00048900      IF(IJOFF(I+1,J))GO TO 400
00049000      IF(OVRPRT(I,J))GO TO 400
00049100      ICONC=CONC(IL,JL)/TS*CNFACT+0.5
00049200      IF(ICONC.LE.0)GO TO 400
00049300      IF(ICONC.LT.1000)GO TO 325
00049400      G(I-1,J)=AST
00049500      G(I,J)=AST
00049600      G(I+1,J)=AST
00049700      GO TO 400
00049800      325 DO 350 IP=1,3
          IV=ICONC-ICONC/10*10

```

```

00049900          IF(IV.EQ.0)IV=10
00050000          ICONC=ICONC/10
00050100          G(I+2-IP,J)=NUM(IV)
00050200          IF(ICONC.EQ.0)GO TO 400
00050300 350      CONTINUE
00050400 400      CONTINUE
00050500 500      CALL PLTSRC(OLAT,OLON)
00050600          CALL DOPLLOT(9,IDAY,ITIME)
00050700          RETURN
00050800          ENTRY CNPRNT(IDAY,ITIME)
00050900          IF(NREC.EQ.0)RETURN
00051000          DO 600 I=1,NREC
00051100          FXCON(I)=-1.0
00051200          CALL INDEX1(RLAT(I),RLON(I),ILAT,JLON,IR,&600)
00051300          FXCON(I)=CONC(ILAT,JLON)/TS*CNFACT
00051400 600      CONTINUE
00051500          WRITE(7,700)IDAY,ITIME,(FXCON(I),I=1,NREC)
00051600 700      FORMAT(1X,I3,1X,I2,5X,10F5.1)
00051700          RETURN
00051800          END
00051900C-----
00052000C*****SUB TO PLOT DATA ON STANDARD PLANE PROJECTION
00052100          SUBROUTINE MPLOT(MAPS)
00052200          REAL*4 XLON(120),YLAT(76)
00052300          INTEGER*2 NI(2),NJ(2),LPI(2)
00052400          LOGICAL*1 G,BLANK,PLUS,SOURCE
00052500          COMMON/MAPVAL/XGPD,YGPD,TALATT
00052600          COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00052700          COMMON/MAPGRD/G(130,80)
00052800          DATA BLANK/' ','/','PLUS/'+' ','SOURCE/'S'/
00052900          DATA NI/120,70/,NJ/76,40/,LPI/8,6/
00053000          T(ALAT)=ALOG(TALATT/TAN(0.7853982+ALAT*0.008726646))
00053100          TALATT=TAN(0.7853982+ALATT*0.008726646)
00053200          TALATB=TAN(0.7853982+ALATB*0.008726646)
00053300          NIX=NI(MAPS+1)
00053400          NJY=NJ(MAPS+1)
00053500          XDPI=(-57.29578*LPI(MAPS+1)/NJY)*ALOG(TALATB/TALATT)
00053600          XGPD=10.0/XDPI
00053700          YGPD=LPI(MAPS+1)/(XDPI*0.01745329)
00053800          XLON(1)=ALONL
00053900          DO 100 LX=2,NIX
00054000          XLON(LX)=XLON(LX-1)-XDPI/10.0
00054100 100      CONTINUE
00054200          DO 200 J=1,NJY
00054300          YLAT(J)=114.5916*(ATAN(EXP(-J/YGPD)*TALATT))-90.0
00054400 200      CONTINUE
00054500          RETURN
00054600          ENTRY PLTZRO
00054700          DO 300 I=1,130
00054800          DO 300 J=1,80
00054900          G(I,J)=BLANK
00055000 300      CONTINUE
00055100          RETURN
00055200          ENTRY PLTSRC(OLAT,OLON)
00055300          I=(ALONL-OLON)*XGPD+1.5
00055400          J=T(OLAT)*YGPD+0.5
00055500          G(I,J)=SOURCE
00055600          RETURN
00055700          ENTRY DOPLLOT(KUNIT,IDAY,ITIME)
00055800          WRITE(KUNIT,400)IDAY,ITIME
00055900 400      FORMAT(1H1,'DAY ',I2,5X,'TIME ',I2,'Z')
00056000          WRITE(KUNIT,610)(XLON(LX),LX=1,NIX,10)
00056100 610      FORMAT(1H0,1X,13(F7.2,3X))
00056200          WRITE(KUNIT,620)ALATT,(PLUS,LX=1,NIX,10)
00056300 620      FORMAT(1H ,F5.2,13(A1,9X))
00056400          KS=1
00056500          KE=3
00056600          DO 900 J=4,NJY,4
00056700          DO 820 K=KS,KE
00056800          WRITE(KUNIT,810)(G(I,K),I=1,NIX)
00056900 810      FORMAT(1H ,5X,120A1)
00057000 820      CONTINUE
00057100          KS=J+1
00057200          KE=J+3

```

```

00057300      G(1,J)=PLUS
00057400      WRITE(KUNIT,830)YLAT(J),(G(I,J),I=1,NIX)
00057500      830      FORMAT(1H ,F5.2,120A1)
00057600      900      CONTINUE
00057700      RETURN
00057800      END
00057900C-----
00058000C*****SUB TO CONVERT GRID I/J TO LAT LON
00058100C      DEFAULT SET TO 25 BY 60 GRID
00058200C      REMAINDER IN TENTHS AND HUNDREDS OF A DEGREE IN IREM
00058300      SUBROUTINE INDEX
00058400      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00058500      ENTRY INDEX1(TLAT,TLON,ILAT,JLON,IREM,*)
00058600      MLAT=TLAT*100.0
00058700      MLON=TLON*100.0
00058800      ILAT=TLAT-(ALATB-1.0)
00058900      JLON=TLON-(ALONR-1.0)
00059000      IREM=MOD(MLAT,100)*100+MOD(MLON,100)
00059100      IF(ILAT.GT.25.OR.ILAT.LT.1)RETURN 1
00059200      IF(JLON.GT.60.OR.JLON.LT.1)RETURN 1
00059300      RETURN
00059400      ENTRY INDEX2(IREM,ILAT,JLON,TLAT,TLON)
00059500      TLAT=ILAT+(ALATB-1.0)+(IREM/100)/100.0
00059600      TLON=JLON+(ALONR-1.0)+(IREM-IREM/100*100)/100.0
00059700      RETURN
00059800      END
00059900C-----
00060000C*****SUB TO INITILIZE CERTAIN ARRAYS
00060100C      FUNCTION VALUES ARE PUT IN TABLES (SINE AND COSINE)
00060200C      MISSING DATA INDEX (-1) IN COMMON BLOCK BEFORE READ
00060300      SUBROUTINE INITLZ(NPUFF)
00060400      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
00060410      * IDSTN(25,60)
00060500      REAL*4 COS360(360),SIN360(360),COSLAT(25)
00060600      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
00060700      COMMON/TRANS/COS360,SIN360,COSLAT
00060800      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00060900      NPUFF=0
00061000      DO 50 I=1,360
00061100      ANG=I*0.0174533
00061200      SIN360(I)=SIN(ANG)
00061300      COS360(I)=COS(ANG)
00061400      50 CONTINUE
00061500      DO 75 I=1,25
00061600      ANG=(I+ALATB-0.5)*0.0174533
00061700      COSLAT(I)=COS(ANG)
00061800      75 CONTINUE
00061900      RETURN
00062000      ENTRY INITL2
00062100      DO 100 I=1,25
00062200      DO 100 J=1,60
00062300      IDSTN(I,J)=0
00062400      100 CONTINUE
00062500      DO 200 I=1,100
00062600      MIXH(I)=-1
00062700      DO 200 J=1,10
00062800      WDIR(I,J)=-1
00062900      WSPD(I,J)=-1
00063000      200 CONTINUE
00063100      RETURN
00063200      END
00063300C-----
00063400C*****DIVIDE DAYTIME MIXED LAYER INTO NOCTURNAL SUBLAYERS
00063500C      BELOW PREVIOUS DAYTIME MIXED DEPTH, PUFF MASS IS
00063600C      DIVIDED ACCORDINGLY
00063700      SUBROUTINE PSPLIT(NPUFF)
00063800      REAL*4 MASS(05000)
00063900      INTEGER*2 POSTN(05000),ARRAY(05000),IDSTN(25,60)
00064000      INTEGER*2 WSPD(100,10),WDIR(100,10),MIXH(100),
00064010      * SIGMA(05000)
00064100      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
00064200      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
00064300      IF(NPUFF.LT.1)RETURN
00064400      L2=NPUFF+1

```

```

00064500      L1=1
00064600  100  IT1=ARRAY(L1)/1000
00064700      JT1=ARRAY(L1)/10-IT1*100
00064800      KT1=ARRAY(L1)-ARRAY(L1)/10*10
00064900      IF(KT1.NE.1)GO TO 300
00065000      KMET=IDSTN(IT1,JT1)
00065100      KMET=IABS(KMET)
00065200      KMIX=MIXH(KMET)/300+1
00065300      KMIX=MIN0(KMIX,10)
00065400      MASS(L1)=MASS(L1)/KMIX
00065500          DO 200 K=2,KMIX
00065600          KM=K
00065700          MASS(L2)=MASS(L1)
00065800          POSTN(L2)=POSTN(L1)
00065900          KM=MOD(KM,10)
00066000          ARRAY(L2)=IT1*1000+JT1*10+KM
00066100          SIGMA(L2)=SIGMA(L1)
00066200          L2=L2+1
00066300  200  CONTINUE
00066400  300  L1=L1+1
00066500          IF(L1.LE.NPUFF)GO TO 100
00066600          NPUFF=L2-1
00066700          IF(NPUFF.GT.05000)GO TO 400
00066800          RETURN
00066900  400  WRITE(6,500)NPUFF
00067000  500  FORMAT(IH0,'EXCEEDS TRAJECTORY LIMIT IN PSPLIT BY ',I5)
00067100      STOP
00067200      END
00067300C-----
00067400C*****COMBINE COINCIDENT TRAJECTORY SUBLAYERS
00067500C      THE MASS OF PUFFS AT THE SAME POSITION IS COMBINED
00067600C      PUFFS WITH THE GREATEST MASS RETAIN THEIR POSITION
00067700C      AND DIAMETER
00067800      SUBROUTINE PCNVRG(NPUFF)
00067900      INTEGER*2 IDSTN(25,60),POSTN(05000),ARRAY(05000)
00068000      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100)
00068100      INTEGER*2 KDSTN(25,60),SIGMA(05000)
00068200      REAL*4 MASS(05000)
00068300      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
00068400      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,KDSTN
00068500      IF(NPUFF.LT.2)RETURN
00068600          DO 900 LOOP=1,2
00068700              DO 50 I=1,25
00068800                  DO 50 J=1,60
00068900                      KDSTN(I,J)=0
00069000  50  CONTINUE
00069100              DO 500 L=1,NPUFF
00069200                  IF(MASS(L).EQ.0.0)GO TO 500
00069300                  IN=ARRAY(L)/1000
00069400                  JN=ARRAY(L)/10-IN*100
00069500                  KN=ARRAY(L)-ARRAY(L)/10*10
00069600                  IF(KN.EQ.0)KN=10
00069700                  IF(KDSTN(IN,JN))100,100,200
00069800  100  KDSTN(IN,JN)=L
00069900                  GO TO 500
00070000  200  LO=KDSTN(IN,JN)
00070100                  KO=ARRAY(LO)-ARRAY(LO)/10*10
00070200                  IF(KO.EQ.0)KO=10
00070300                  GO TO(210,220),LOOP
00070400  210  IF(KO.EQ.KN)GO TO 250
00070500                  GO TO 500
00070600  220  KMET=IDSTN(IN,JN)
00070700                  KMET=IABS(KMET)
00070800                  KMIX=MIXH(KMET)/300+1
00070900                  KMIX=MIN0(KMIX,10)
00071000                  IF(KO.GT.KMIX.OR.KN.GT.KMIX)GO TO 500
00071100                  ARRAY(LO)=IN*1000+JN*10+1
00071200  250  IF(MASS(LO).GT.MASS(L))GO TO 300
00071300                  POSTN(LO)=POSTN(L)
00071400                  SIGMA(LO)=SIGMA(L)
00071500  300  MASS(LO)=MASS(LO)+MASS(L)
00071600                  MASS(L)=0.0
00071700  500  CONTINUE
00071800  900  CONTINUE

```

```

00071900      RETURN
00072000      END
00072100C-----
00072200C*****ZERO MASS OR LOW MASS PUFFS ARE SORTED OUT OF ARRAY
00072300C      LOW MASS VALUE CURRENTLY SET TO 1.0 UNITS
00072400      SUBROUTINE PSORT(NPUFF,PMIN)
00072500      INTEGER*2 POSTN(05000),ARRAY(05000),SIGMA(05000)
00072600      REAL*4 MASS(05000)
00072700      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
00072800      IF(NPUFF.LT.2)RETURN
00072900      LB=1
00073000      LE=1
00073100      10 IF(MASS(LE).GT.PMIN)GO TO 50
00073200      LE=LE+1
00073300      IF(LE.GT.NPUFF)GO TO 60
00073400      GO TO 10
00073500      50 MASS(LB)=MASS(LE)
00073600      POSTN(LB)=POSTN(LE)
00073700      ARRAY(LB)=ARRAY(LE)
00073800      SIGMA(LB)=SIGMA(LE)
00073900      LB=LB+1
00074000      LE=LE+1
00074100      IF(LE.LE.NPUFF)GO TO 10
00074200      60 NPUFF=LB-1
00074300      RETURN
00074400      END
00074500C-----
00074600C*****ASSIGN MET DATA TO ALL ARRAY POSITIONS
00074700C      ALL I,J OF KDSTN ASSIGNED SEQUENCE NUMBER OF NEAREST
00074800C      METEOROLOGICAL STATION, EXACT I,J OF A STATION
00074900C      HAS A NEGATIVE SEQUENCE NUMBER
00075000      SUBROUTINE ASSIGM(KDSTN)
00075100      REAL*4 COS360(360),SIN360(360),COSLAT(25)
00075200      INTEGER*2 KDSTN(25,60)
00075300      COMMON/TRANS/COS360,SIN360,COSLAT
00075400      KB=0
00075500      DO 100 I=1,25
00075600      DO 100 J=1,60
00075700      IF(KDSTN(I,J).NE.0)GO TO 100
00075800      DO 85 KGRID=1,20
00075900      KGRDI=KGRID
00076000      KGRDJ=KGRID*COSLAT(I)+0.5
00076100      IB=MAX0(I-KGRDI,1)
00076200      IE=MIN0(I+KGRDI,25)
00076300      JB=MAX0(J-KGRDJ,1)
00076400      JE=MIN0(J+KGRDJ,60)
00076500      JBI=MIN0(JB+1,60)
00076600      JEI=MAX0(JE-1,1)
00076700      DO 70 II=IB,IE
00076800      KSTA=KDSTN(II,JB)
00076900      IF(KSTA.LT.0)GO TO 90
00077000      KSTA=KDSTN(II,JE)
00077100      IF(KSTA.LT.0)GO TO 90
00077200      70 CONTINUE
00077300      IF(JBI.GE.J.OR.JEI.LE.J)GO TO 85
00077400      DO 75 JJ=JBI,JEI
00077500      KSTA=KDSTN(IB,JJ)
00077600      IF(KSTA.LT.0)GO TO 90
00077700      KSTA=KDSTN(IE,JJ)
00077800      IF(KSTA.LT.0)GO TO 90
00077900      75 CONTINUE
00078000      85 CONTINUE
00078100      KB=KB+1
00078200      GO TO 100
00078300      90 KDSTN(I,J)=-KSTA
00078400      100 CONTINUE
00078500      IF(KB.EQ.0)GO TO 200
00078600      WRITE(6,150)KB
00078700      150 FORMAT(27X,14,' UNASSIGNED POSITIONS ON GRID')
00078800      200 RETURN
00078900      END

```