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

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 begining 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 NAMER-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 - Begining day of computations (01-31)
    IBMO 11-14  xxxx - Begining 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 calcualtional 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 longtitude 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 incrememt 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 minumum 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 sumarized as read from the input tape.

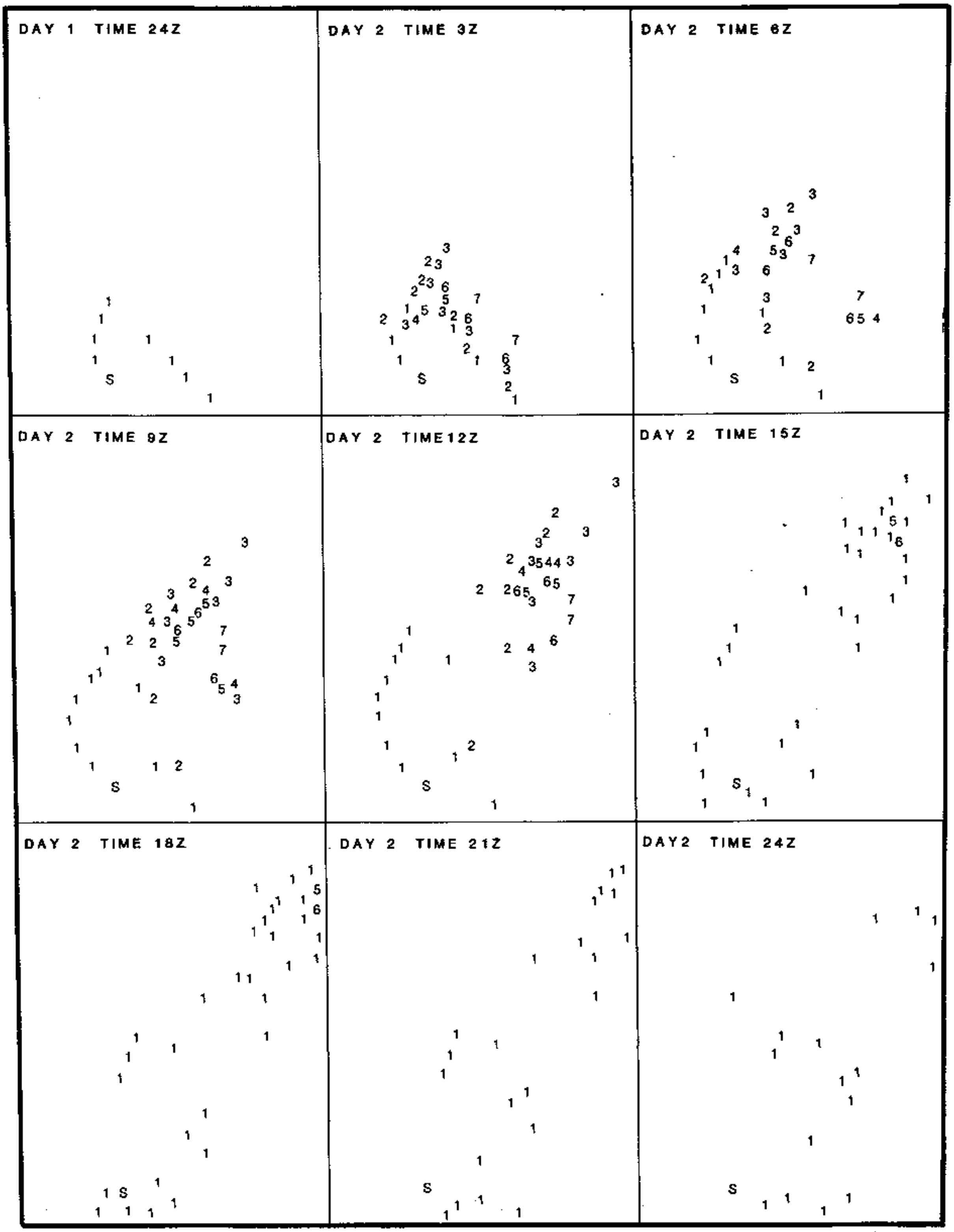

Figure 2. Illustration of output from unit 8 for sample calculation. Shown are the positions of the puffs at three hour intervals. Puffs in the surface layer are identified by the digit '1'. Elevated puffs have digits from 2 to 0. The height is the value of the digit times 300m. The position of the source is indicated by 'S'.

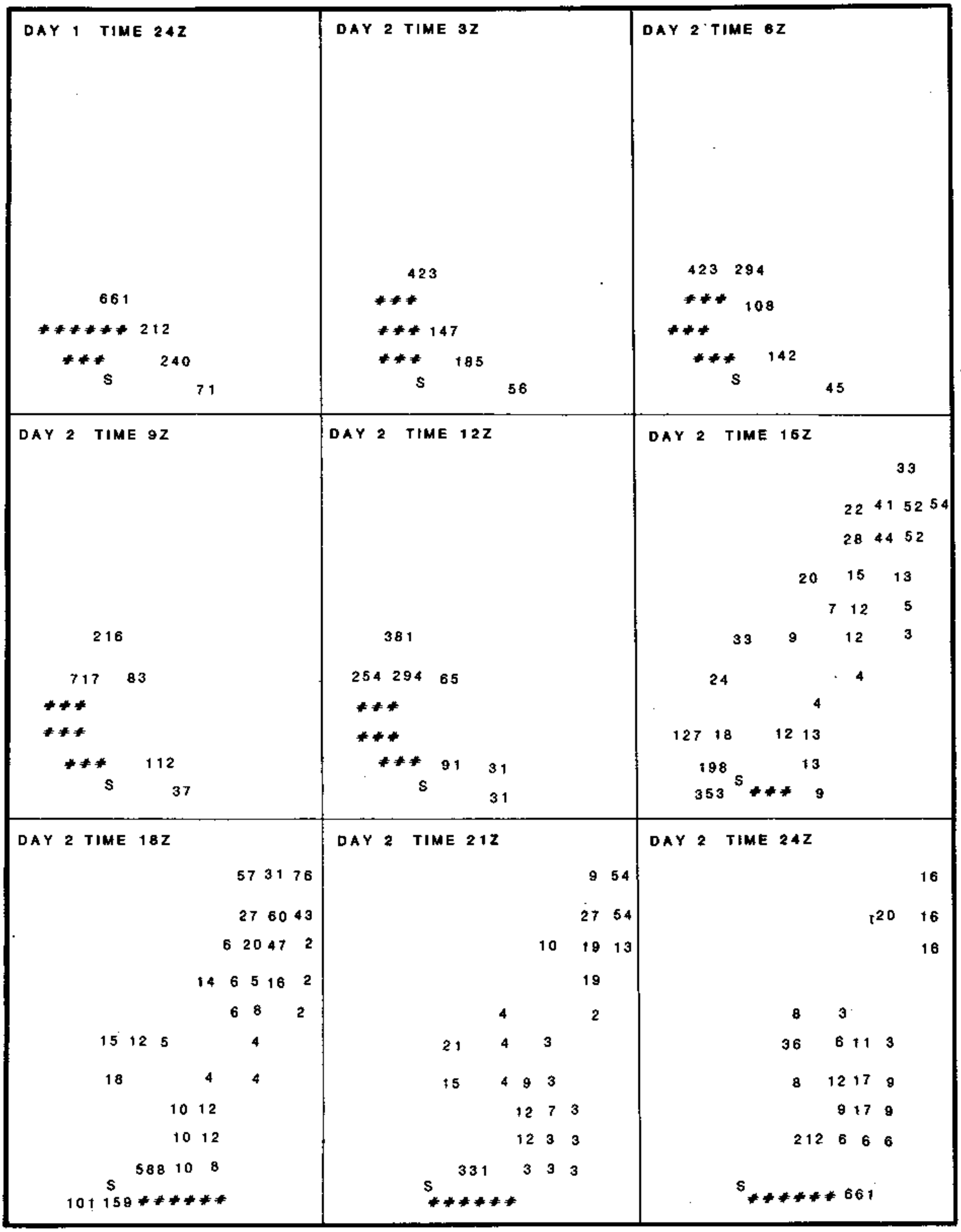

Figure 3. Illustration of the output from unit 9 for the sample calculation. Shown are the 3-hour average concentrations. Only puffs in the surface layer (digit - 1 in Fig. 2) contribute to air concentrations.

END SAMPLING DAY	HR	RECEPTORS 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

```
C-----------------------------------------------------------------------
C********MAIN PROGRAM FOR LONG-RANGE WIND SHEAR TRAJECTORIES
      INTEGER*2 IDSTN(25,60)
      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
      COMMON/MSTNS/IDSTN,DUMMY(1800)
      COMMON/CON/TS,CNFACT
      COMMON/RCPFX/RLAT(10),RLON(10),NREC
      DATA IMAPS/0/
C---------CONCENTRATION FACTOR TO GIVE INTEGER UNITS ON OUTPUT
      CNFACT=1.0E+12
C---------MINIMUM PUFF MASS (ARBITRARY UNITS)
      PMIN=1.0
C---------PROGRAM CONTROL INPUT PARAMETERS
C         IBDA - BEGINING DAY OF COMPUTATIONS
C         IBMO - BEGINING MONTH OF COMPUTATIONS
C         NDTR - NUMBER OF DAYS OF COMPUTATIONS
C         NAVG - NUMBER OF 3HR CONCENTRATION AVERAGES
C         MAPS - FLAG FOR SHORT OR LONG MAP OUTPUT (80 OR 132)
C                0 - 132 COLUMN PRINTER; 1 - 80 COLUMN PRINTER
C         MAPC - CONCENTRATION MAP FLAG (0/1) AT NAVG INTERVAL
C         MAPT - NUMBER OF 3HR PERIODS BETWEEN PUFF MAPS
C         OLAT - ORIGIN LATITUDE (DEG-TENTHS)
C         OLON - ORIGIN LONGITUDE (DEG-TENTHS)
C         QTERM- EMISSIONS PER 3 HOUR PUFF
C         NQTRM- NUMBER OF 3 HOUR PUFFS EMITTED
C         ALATT- TOP LATITUDE OF GRID (DEG-TENTHS)
C         ALATB- BOTTOM LATITUDE ...
C         ALONL- LEFT LONGITUDE ...
C         ALONR- RIGHT LONGITUDE ...
C---------INPUT/OUTPUT FILE REQUIREMENTS
C         FT05F001 - PROGRAM CONTROL INPUT PARAMETERS
C         FT06F001 - PRINTER OUTPUT FOR RUN STATISTICS
C         FT07F001 - PRINTER OUTPUT FOR FIXED SAMPLING SITES
C         FT08F001 - PRINTER OUTPUT FOR PUFF POSITION MAPS
C         FT09F001 - PRINTER OUTPUT FOR CONCENTRATION MAPS
C         FT10F001 - INPUT METEOROLOGICAL DATA
C                    NAMER-WINDTEMP TAPES (TD-9743)
C                    N.C.C., DIGITAL PRODUCTS
C                    ASHVILLE, NC   28801
      WRITE(6,20)
   20 FORMAT(1X,'INPUT PARAMETERS')
      WRITE(6,25)
   25 FORMAT(6X,'IBDA IBMO NDTR NAVG MAPS MAPC MAPT')
      READ(5,50)IBDA,IBMO,NDTR,NAVG,MAPS,MAPC,MAPT
   50 FORMAT(5X,7(I4,1X))
      WRITE(6,60)IBDA,IBMO,NDTR,NAVG,MAPS,MAPC,MAPT
   60 FORMAT(6X,7(I4,1X))
      WRITE(6,75)
   75 FORMAT(6X,'OLAT  OLON QTERM NQTRM ALATT ALATB ALONL',
     * ' ALONR')
      READ(5,100)OLAT,OLON,QTERM,NQTRM,ALATT,ALATB,ALONL,ALONR
  100 FORMAT(5X,F4.1,1X,F5.1,1X,F5.0,1X,I5,4F6.1)
      WRITE(6,120)OLAT,OLON,QTERM,NQTRM,ALATT,ALATB,ALONL,
     * ALONR
  120 FORMAT(6X,F4.1,1X,F5.1,1X,F5.0,1X,I5,4F6.1)
      WRITE(6,110)
  110 FORMAT(' NUMBER OF FIXED POINT RECEPTORS',/,6X,'NREC')
      READ(5,125)NREC
  125 FORMAT(5X,I4)
      WRITE(6,126)NREC
  126 FORMAT(6X,I4)
      IF(NREC.EQ.0)GO TO 149
      WRITE(6,128)
  128 FORMAT('      RLAT  RLON')
```

```
      DO 140 I=1,NREC
      READ(5,130)RLAT(I),RLON(I)
  130 FORMAT(5X,F4.1,1X,F5.1)
      WRITE(6,135)RLAT(I),RLON(I)
  135 FORMAT(6X,F4.1,1X,F5.1)
  140 CONTINUE
      WRITE(7,144) (I,I=1,NREC)
  144 FORMAT('  END SAMPLING  RECEPTORS',/,' DAY HR',5X,10I5)
      WRITE(7,146)
  146 FORMAT(' ')
  149 CALL CNZERO
      CALL MPLOT(MAPS)
      CALL INITLZ(NPUFF)
      CALL PSTNTP(IBMO,IBDA)
      CALL SKIP(2)
      WRITE(6,150)
  150 FORMAT(1H0,'FINISHED READING METEOROLOGOCAL DATA',14X,
     *   'MIXING DEPTH DATA (M)',/,6X,'MONTH',2X,'YEAR',1X,
     *    'DAY-HR',18X,'REPORTS',4X,
     *    'MIN',1X,'MEAN',3X,'MAX',5X,'PUFFS')
      IF(NAVG.EQ.1)CALL CNZERO
             DO 1000 I=1,NDTR
                    DO 900 ITIME=3,24,3
                    IF(IMAPS.GE.MAPT)IMAPS=0
                    IMAPS=IMAPS+1
                    IF(IFIX(TS).GE.NAVG)CALL CNZERO
                    TS=TS+1.0
                    IGMT=ITIME-3
                    CALL RDMTDT(NPUFF,IGMT,&800)
                    CALL ASSIGM(IDSTN)
  800               CALL ADVCTN(NPUFF,IGMT,OLAT,OLON,QTERM,NQTRM)
                    IF(IMAPS.GE.MAPT)CALL PFPLOT(I,ITIME,NPUFF,
     *              OLAT,OLON)
                    IF(MAPC.EQ.1.AND.IFIX(TS).GE.NAVG)CALL CNPLOT
     *              (I,ITIME,OLAT,OLON)
                    IF(IFIX(TS).GE.NAVG)CALL CNPRNT(I,ITIME)
  900               CONTINUE
             CALL PCNVRG(NPUFF)
             CALL PSORT(NPUFF,PMIN)
             CALL PSPLIT(NPUFF)
 1000        CONTINUE
      STOP
      END
C-----------------------------------------------------------------------
C*******SUB TO READ NAMER WIND-TEMP TAPE
C            SET TO READ ONLY 0 OR 12 GMT DATA (IGMT)
      SUBROUTINE RDMTDT(NPUFF,IGMT,*)
      INTEGER*2 IZHT(20),IDIR(20),ISPD(20),ITEMP(20),IPPP(20),
     A    DUM1(20),DUM2(20)
      COMMON/PRFL/IZHT,IDIR,ISPD,ITEMP,IPPP,DUM1,DUM2
      IF(IGMT.EQ.0.OR.IGMT.EQ.12)GO TO 10
      IF(IGMT.EQ.6.OR.IGMT.EQ.18)CALL SKIP(1)
      RETURN 1
   10 NSTA=0
      NTEMP=0
      AVGMX=0.0
      MINMX=3000
      MAXMX=0
      CALL INITL2
             DO 300 I=1,2
             READ(10,30,END=400,ERR=412)MON,IYR,IDY,IHR,NREPT
   30        FORMAT(A3,I4,I2,I2,I4)
             IF(NREPT.EQ.0)GO TO 300
                    DO 200 J=1,NREPT
                    READ(10,40,END=400,ERR=412)LATSTA,LONSTA,
     *              IHTSTA,NLVLS
   40               FORMAT(5X,I5,I7,I5,5X,I2)
                    IF(NLVLS.EQ.0)GO TO 200
                           DO 100 K=1,NLVLS
                           GO TO(50,75),I
   50                      READ(10,45,END=400,ERR=400)IHT,IWDR,IWSP
   45                      FORMAT(I5,I3,I4)
                           IF(K.GT.20)GO TO 100
                           IZHT(K)=MAX0(IHT-IHTSTA,0)
```

```
                  ISPD(K)=IWSP/10.0+0.5
                  IDIR(K)=IWDR
                  IF(IDIR(K).EQ.0)IDIR(K)=360
                  GO TO 100
 75               READ(10,47,END=400,ERR=400)IHT,IPRES,IDEG
 47               FORMAT(I4,I5,I4)
                  IF(K.GT.20)GO TO 100
                  IZHT(K)=MAX0(IHT-IHTSTA,0)
                  IPPP(K)=IPRES/10.0+0.5
                  ITEMP(K)=IDEG
100               CONTINUE
              IF(NLVLS.GT.20)NLVLS=20
              IF(NLVLS.LE.1)GO TO 200
              STALAT=LATSTA/100.0
              STALON=LONSTA/100.0
              CALL INDEX1(STALAT,STALON,ILAT,JLON,IREM,&200)
              GO TO(110,120),I
110           CALL WBANW(NSTA,ILAT,JLON)
              CALL WNDTMP(NLVLS,NSTA)
              GO TO 200
120           CALL WBANT(NSTA,ILAT,JLON,&200)
              CALL TMPRFL(NLVLS,NSTA,MIX)
              NTEMP=NTEMP+1
              AVGMX=AVGMX+MIX
              MINMX=MIN0(MINMX,MIX)
              MAXMX=MAX0(MAXMX,MIX)
200           CONTINUE
300       CONTINUE
      CALL WBANF
      AVGMX=AVGMX/NTEMP
      WRITE(6,550)MON,IYR,IDY,IHR,NTEMP,MINMX,AVGMX,MAXMX,
     * NPUFF
550   FORMAT(6X,A3,3X,I4,2(2X,I2),18X,I5,5X,I4,1X,F5.0,1X,I4,
     * 5X,I5)
      GO TO 420
400   WRITE(6,410)
410   FORMAT(1H0,'EOF ON UNIT 10 FROM RDMTDT')
      STOP
412   WRITE(6,414)
414   FORMAT(1H0,'I/O ERROR ON UNIT 10 FROM RDMTDT')
      STOP
420   RETURN
      END
C-----------------------------------------------------------------------
C*******SUB TO POSITION TAPE TO FIRST INPUT DAY
C           TAPES HAVE TWO FILES PER MONTH
C           DAYS 1 - 15 AND DAYS 16 TO END OF MONTH
      SUBROUTINE PSTNTP(IBMO,IBDA)
      IF(IBDA.EQ.1.OR.IBDA.EQ.16)RETURN
      ISKIP=(IBDA-1)*4
      IF(IBDA.GE.16)ISKIP=(IBDA-16)*4
      ENTRY SKIP(ISKIP)
          DO 100 I=1,ISKIP
          DO 100 J=1,2
          READ(10,20,END=110,ERR=130)NREC
 20       FORMAT(15X,I5)
          IF(NREC.EQ.0)GO TO 100
          DO 90 K=1,NREC
          READ(10,30,END=110,ERR=130)
 30       FORMAT(' ')
 90       CONTINUE
100       CONTINUE
      RETURN
110   WRITE(6,120)
120   FORMAT(1H0,'EOF UNIT 10 FROM PSTNTP')
      STOP
130   WRITE(6,140)
140   FORMAT(1H0,'I/O ERROR ON UNIT 10 FROM PSTNTP')
      STOP
      END
C-----------------------------------------------------------------------
C*******SUB TO ADD SEQUENCE NUMBER OF METEOROLOGICAL STATION
C*******TO DATA GRID (INDEXED BY POSITION IN WHOLE DEGREES)
C           POSITION SEQUENCE NUMBER IDENTIFIES ARRAY LOCATION
```

```
C          EACH VARIABLE IN MSTNS COMMON BLOCK
      SUBROUTINE WBANX
      INTEGER*2 IDSTN(25,60)
      COMMON/MSTNS/IDSTN,DUMMY(1800)
      ENTRY WBANW(NSTA,ILAT,JLON)
      NSTA=NSTA+1
      IDSTN(ILAT,JLON)=NSTA
      RETURN
      ENTRY WBANT(NSTA,ILAT,JLON,*)
      IF(IDSTN(ILAT,JLON).EQ.0)RETURN 1
      NSTA=IDSTN(ILAT,JLON)
      IDSTN(ILAT,JLON)=-IDSTN(ILAT,JLON)
      RETURN
      ENTRY WBANF
          DO 100 I=1,25
          DO 100 J=1,60
          IF(IDSTN(I,J).LE.0)GO TO 100
          IDSTN(I,J)=0
  100     CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
C*******SUB TO PUT WIND SOUNDING IN MET DATA ARRAY
C          IN COMMON BLOCK MSTNS
      SUBROUTINE WNDTMP(NLVLS,NSTA)
      INTEGER*2 IZHT(20),IDIR(20),ISPD(20),DUM1(20),ITEMP(20)
      INTEGER*2 DUM2(20),IDSTN(25,60),IPPP(20)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100)
      COMMON/PRFL/IZHT,IDIR,ISPD,ITEMP,IPPP,DUM1,DUM2
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      IDF=0
          DO 100 I=1,NLVLS
          IDI=IZHT(I)/300+1
          IF(IDI.GT.10)GO TO 100
          IF(IDI.EQ.IDF)GO TO 100
          IDF=IDI
          WDIR(NSTA,IDF)=IDIR(I)
          WSPD(NSTA,IDF)=ISPD(I)
  100     CONTINUE
      RETURN
      END
C-----------------------------------------------------------------
C*******SUB TO DETERMINE INVERSION HEIGHT AT EACH RAWINSONDE
C          AND PUT MIXH IN COMMON BLOCK MSTNS
      SUBROUTINE TMPRFL(NLVLS,NSTA,MIX)
      INTEGER*2 IZHT(20),ITEMP(20),DUM2(20),IDIR(20),ISPD(20),
     *    DUM1(20),WDIR(100,10),WSPD(100,10),MIXH(100),
     *    IPPP(20),IDSTN(25,60)
      INTEGER*4 LB(2),LE(2)
      COMMON/PRFL/IZHT,IDIR,ISPD,ITEMP,IPPP,DUM1,DUM2
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      DZL=5.0
   10 IFLG=0
      T1=ITEMP(1)/10.0*(1000.0/IPPP(1))**0.286
          DO 100 I=2,NLVLS
          T2=ITEMP(I)/10.0*(1000.0/IPPP(I))**0.286
          IF(IZHT(I-1).LT.300)GO TO 90
          IF(IZHT(I)-IZHT(I-1).LE.0)GO TO 100
          DTDZ=(T2-T1)/(IZHT(I)-IZHT(I-1))*1000.0
          IF(DTDZ.LT.DZL)GO TO 90
          IF(IFLG.EQ.0)DT=0.0
          IFLG=1
          DT=DT+(T2-T1)
          IF(DT.GT.2.0)GO TO 200
          T1=T2
          GO TO 100
   90     T1=T2
          IFLG=0
  100     CONTINUE
      DZL=DZL-1.0
      IF(DZL.GE.1.0)GO TO 10
      MIX=2999
      GO TO 210
  200 MIX=IZHT(I)
```

```
  210 MIXH(NSTA)=MIN0(MIX,2999)
      RETURN
      END
C-----------------------------------------------------------------------
C*******SUB TO COMPUTE HORIZONTAL AVERAGED WIND FOR ADVECTION
C           NOTE THAT AFTER STMT 60 THE SECTION IS SET TO SCAN
C           SOUNDING IF THERE ARE MISSING DATA AT A LEVEL
C           OTHERWISE NO ADVECTION IS CALCULATED ON THE 3RD PASS
      SUBROUTINE WINDWT(NWIND,LBT,LTT,XTS,YTS,*)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
     * IDSTN(25,60)
      REAL*4 SIN360(360),COS360(360),COSLAT(25)
      COMMON/TRANS/COS360,SIN360,COSLAT
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      MIX=MIXH(NWIND)/300+1
      MIX=MIN0(MIX,10)
      IC=0
      LB=LBT
      LT=LTT
   10 XAW=0.0
      YAW=0.0
      INW=0
           DO 50 K=LB,LT
           ISPD=WSPD(NWIND,K)
           IDIR=WDIR(NWIND,K)
           IF(IDIR.LT.0.OR.ISPD.LT.0)GO TO 50
           XAW=XAW-ISPD*SIN360(IDIR)
           YAW=YAW-ISPD*COS360(IDIR)
           INW=INW+1
   50      CONTINUE
      IF(INW.NE.0)GO TO 100
      IC=IC+1
      GO TO(60,70,200),IC
   60 IF(LB.GT.MIX)GO TO 65
      LB=1
      GO TO 10
   65 LB=MIN0(MIX+1,10)
      GO TO 10
   70 IF(LB.GT.MIX)GO TO 75
      LT=MIX
      GO TO 10
   75 LT=10
      GO TO 10
  100 XAW=XAW/INW
      YAW=YAW/INW
      XTS=1.94*XAW*3.0
      YTS=1.94*YAW*3.0
      RETURN
  200 RETURN 1
      END
C-----------------------------------------------------------------------
C*******SUBROUTINE TO CALCULATE TRAJECTORY ADVECTION
C           AND START NEW PUFFS WHICH ARE PUT IN COMMON BLOCK
C           CONTAINING MASS, POSITION, INDEX, AND DIAMETER
      SUBROUTINE ADVCTN(NPUFF,IGMT,OLAT,OLON,QTERM,NQTRM)
      INTEGER*2 POSTN(05000),ARRAY(05000),IDSTN(25,60)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
     * SIGMA(05000)
      REAL*4 MASS(05000),COS360(360),SIN360(360),COSLAT(25)
      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
      COMMON/TRANS/COS360,SIN360,COSLAT
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      DATA IQCNT/0/
      CALL INDEX1(OLAT,OLON,ILAT,JLON,JREM,&700)
      IQCNT=IQCNT+1
      IF(IQCNT.GT.NQTRM)GO TO 200
      NPUFF=NPUFF+1
      MASS(NPUFF)=QTERM
      POSTN(NPUFF)=JREM
      ARRAY(NPUFF)=ILAT*1000+JLON*10+1
      SIGMA(NPUFF)=0
  200 IF(NPUFF.EQ.0)GO TO 600
           DO 500 L=1,NPUFF
           IF(MASS(L).EQ.0.0)GO TO 500
```

```
                IT1=ARRAY(L)/1000
                JT1=ARRAY(L)/10-IT1*100
                KT1=ARRAY(L)-ARRAY(L)/10*10
                NW1=IDSTN(IT1,JT1)
                NW1=IABS(NW1)
                IF(NW1.EQ.0)GO TO 375
                SIGMA(L)=SIGMA(L)+5
                CALL LEVEL(NW1,KT1,IGMT,LB,LT)
 300            CALL WINDWT(NW1,LB,LT,XTS,YTS,&325)
                IREM=POSTN(L)
                CALL INDEX2(IREM,IT1,JT1,TLAT1,TLON1)
                TLAT2=TLAT1+YTS/60.0
                TLON2=TLON1-(XTS/(60.0*COSLAT(IT1)))
                CALL INDEX1(TLAT2,TLON2,IT2,JT2,IREM,&375)
                POSTN(L)=IREM
                ARRAY(L)=IT2*1000+JT2*10+KT1
                IT1=IT2
                JT1=JT2
 325            TSIGMA=SIGMA(L)*1000.0
                IF(LB.EQ.1)CALL CNSUM(IGMT,LT,MASS(L),IT1,JT1,TSIGMA)
                GO TO 500
 375            MASS(L)=0.0
 500            CONTINUE
 600  RETURN
 700  WRITE(6,800)
 800  FORMAT(1H0,'ERROR ORIGIN POSITION ')
      STOP
      END
C------------------------------------------------------------------------
C*******SUB TO SELECT BOTTOM AND TOP OF ADVECTION LAYER
C         INDIVIDUAL LEVELS AT NIGHT AND ABOVE MIXED LAYER
C         AND THE AVERAGE WITHIN THE MIXED LAYER DURING THE DAY
      SUBROUTINE LEVEL(NW1,KT1,IGMT,LB,LT)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
     * IDSTN(25,60)
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      LB=KT1
      IF(LB.EQ.0)LB=10
      LT=KT1
      IF(LT.EQ.0)LT=10
      MIX=MIXH(NW1)/300+1
      MIX=MIN0(MIX,10)
      IF(LB.GT.MIX)GO TO 100
      IF(IGMT.LT.12)GO TO 100
      LB=1
      LT=MIX
 100  KT1=MOD(LB,10)
      RETURN
      END
C------------------------------------------------------------------------
C*******SUB TO ARRANGE FOR PLOTTING DAILY PUFF POSITIONS
C         PUFFS THAT CONTRIBUTE TO SURFACE CONCENTRATION
C         ARE INDICATED BY INDEX 1, OTHERS HAVE LEVEL NUMBERS
C         THAT ARE MULTIPLES OF 300 METERS
      SUBROUTINE PFPLOT(IDAY,ITIME,NPUFF,OLAT,OLON)
      INTEGER*2 POSTN(05000),ARRAY(05000),SIGMA(05000)
      LOGICAL*1 G,IJOFF,NUM(10)
      REAL*4 MASS(05000)
      COMMON/MAPVAL/XGPD,YGPD,TALATT
      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
      COMMON/MAPGRD/G(130,80)
      DATA NUM/'1','2','3','4','5','6','7','8','9','0'/
      T(ALAT)=ALOG(TALATT/TAN(0.785+ALAT*0.00873))
      IJOFF(I,J)=I.LT.1.OR.I.GT.130.OR.J.LT.1.OR.J.GT.80
      CALL PLTZRO
      IF(NPUFF.EQ.0)GO TO 600
           DO 500 L=1,NPUFF
           IF(MASS(L).EQ.0.0)GO TO 500
           IREM=POSTN(L)
           ITRAJ=ARRAY(L)/1000
           JTRAJ=ARRAY(L)/10-ITRAJ*100
           KTRAJ=ARRAY(L)-ARRAY(L)/10*10
           IF(KTRAJ.EQ.0)KTRAJ=10
```

```
                 CALL INDEX2(IREM,ITRAJ,JTRAJ,TLAT,TLON)
                 IX=(ALONL-TLON)*XGPD+0.5
                 JX=T(TLAT)*YGPD+0.5
                 IF(IJOFF(IX,JX))GO TO 500
                 G(IX,JX)=NUM(KTRAJ)
  500            CONTINUE
  600 CALL PLTSRC(OLAT,OLON)
      CALL DOPLOT(8,IDAY,ITIME)
      RETURN
      END
C------------------------------------------------------------------------
C********SUB TO ARRANGE FOR PLOTTING DAILY CONCENTRATION MAPS
C           VARIABLE TS IS THE SAMPLING COUNTER, INCREMENTED EACH
C           ADVECTION STEP. THE CONCENTRATION FACTOR OF E+12 IS
C           APPLIED BEFORE PLOTTING TO GET WHOLE NUMBERS
      SUBROUTINE CNCON
      REAL*4 CONC(25,60),COSLAT(25),SIN360(360),COS360(360)
      DIMENSION FXCON(10)
      LOGICAL*1 G,IJOFF,NUM(10),AST,BLANK,OVRPRT
      COMMON/MAPVAL/XGPD,YGPD,TALATT
      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
      COMMON/TRANS/COS360,SIN360,COSLAT
      COMMON/MAPGRD/G(130,80)
      COMMON/CON/TS,CNFACT
      COMMON/RCPFX/RLAT(10),RLON(10),NREC
      DATA YD/111180.0/
      DATA NUM/'1','2','3','4','5','6','7','8','9','0'/
      DATA AST/'*'/,BLANK/' '/
      T(ALAT)=ALOG(TALATT/TAN(0.785+ALAT*0.00873))
      OVRPRT(I,J)=G(I-1,J).NE.BLANK.OR.
     *            G(I,J).NE.BLANK.OR.G(I+1,J).NE.BLANK
      IJOFF(I,J)=I.LT.1.OR.I.GT.130.OR.J.LT.1.OR.J.GT.80
      ENTRY CNZERO
           DO 100 I=1,25
           DO 100 J=1,60
           CONC(I,J)=0.0
  100      CONTINUE
      TS=0.0
      RETURN
      ENTRY CNSUM(IGMT,LT,TMASS,ITRAJ,JTRAJ,TSIGMA)
      ZD=300.0
      IF(IGMT.GE.12)ZD=300*LT
      ISCAN=(2.0*TSIGMA)/YD
      JSCAN=(2.0*TSIGMA)/(YD/COSLAT(ITRAJ))
      IMIN=MAX0(ITRAJ-ISCAN,1)
      IMAX=MIN0(ITRAJ+ISCAN,25)
      JMIN=MAX0(JTRAJ-JSCAN,1)
      JMAX=MIN0(JTRAJ+JSCAN,60)
           DO 150 I=IMIN,IMAX
           DO 150 J=JMIN,JMAX
           CONC(I,J)=CONC(I,J)+TMASS/(12.6*ZD*TSIGMA*TSIGMA)
  150      CONTINUE
      RETURN
      ENTRY CNPLOT(IDAY,ITIME,OLAT,OLON)
      CALL PLTZRO
      IF(TS.EQ.0.0)GO TO 500
           DO 400 IL=1,25
           DO 400 JL=1,60
           SLAT=IL+(ALATB-1.0)+0.5
           SLON=JL+(ALONR-1.0)+0.5
           I=(ALONL-SLON)*XGPD+0.5
           J=T(SLAT)*YGPD+0.5
           IF(IJOFF(I-1,J))GO TO 400
           IF(IJOFF(I+1,J))GO TO 400
           IF(OVRPRT(I,J))GO TO 400
           ICONC=CONC(IL,JL)/TS*CNFACT+0.5
           IF(ICONC.LE.0)GO TO 400
           IF(ICONC.LT.1000)GO TO 325
           G(I-1,J)=AST
           G(I,J)=AST
           G(I+1,J)=AST
           GO TO 400
  325           DO 350 IP=1,3
                IV=ICONC-ICONC/10*10
```

```
                      IF(IV.EQ.0)IV=10
                      ICONC=ICONC/10
                      G(I+2-IP,J)=NUM(IV)
                      IF(ICONC.EQ.0)GO TO 400
  350                 CONTINUE
  400           CONTINUE
  500     CALL PLTSRC(OLAT,OLON)
          CALL DOPLOT(9,IDAY,ITIME)
          RETURN
          ENTRY CNPRNT(IDAY,ITIME)
          IF(NREC.EQ.0)RETURN
          DO 600 I=1,NREC
          FXCON(I)=-1.0
          CALL INDEX1(RLAT(I),RLON(I),ILAT,JLON,IR,&600)
          FXCON(I)=CONC(ILAT,JLON)/TS*CNFACT
  600     CONTINUE
          WRITE(7,700)IDAY,ITIME,(FXCON(I),I=1,NREC)
  700     FORMAT(1X,I3,1X,I2,5X,10F5.1)
          RETURN
          END
C-------------------------------------------------------------------
C*******SUB TO PLOT DATA ON STANDARD PLANE PROJECTION
          SUBROUTINE MPLOT(MAPS)
          REAL*4 XLON(120),YLAT(76)
          INTEGER*2 NI(2),NJ(2),LPI(2)
          LOGICAL*1 G,BLANK,PLUS,SOURCE
          COMMON/MAPVAL/XGPD,YGPD,TALATT
          COMMON/GRID/ALONL,ALONR,ALATT,ALATB
          COMMON/MAPGRD/G(130,80)
          DATA BLANK/' '/,PLUS/'+'/,SOURCE/'S'/
          DATA NI/120,70/,NJ/76,40/,LPI/8,6/
          T(ALAT)=ALOG(TALATT/TAN(0.7853982+ALAT*0.008726646))
          TALATT=TAN(0.7853982+ALATT*0.008726646)
          TALATB=TAN(0.7853982+ALATB*0.008726646)
          NIX=NI(MAPS+1)
          NJY=NJ(MAPS+1)
          XDPI=(-57.29578*LPI(MAPS+1)/NJY)*ALOG(TALATB/TALATT)
          XGPD=10.0/XDPI
          YGPD=LPI(MAPS+1)/(XDPI*0.01745329)
                XLON(1)=ALONL
                DO 100 LX=2,NIX
                XLON(LX)=XLON(LX-1)-XDPI/10.0
  100           CONTINUE
                DO 200 J=1,NJY
                YLAT(J)=114.5916*(ATAN(EXP(-J/YGPD)*TALATT))-90.0
  200           CONTINUE
          RETURN
          ENTRY PLTZRO
                DO 300 I=1,130
                DO 300 J=1,80
                G(I,J)=BLANK
  300           CONTINUE
          RETURN
          ENTRY PLTSRC(OLAT,OLON)
          I=(ALONL-OLON)*XGPD+1.5
          J=T(OLAT)*YGPD+0.5
          G(I,J)=SOURCE
          RETURN
          ENTRY DOPLOT(KUNIT,IDAY,ITIME)
          WRITE(KUNIT,400)IDAY,ITIME
  400     FORMAT(1H1,'DAY ',I2,5X,'TIME ',I2,'Z')
          WRITE(KUNIT,610)(XLON(LX),LX=1,NIX,10)
  610     FORMAT(1H0,1X,13(F7.2,3X))
          WRITE(KUNIT,620)ALATT,(PLUS,LX=1,NIX,10)
  620     FORMAT(1H ,F5.2,13(A1,9X))
          KS=1
          KE=3
                DO 900 J=4,NJY,4
                      DO 820 K=KS,KE
                      WRITE(KUNIT,810)(G(I,K),I=1,NIX)
  810                 FORMAT(1H ,5X,120A1)
  820                 CONTINUE
                KS=J+1
                KE=J+3
```

```
            G(1,J)=PLUS
            WRITE(KUNIT,830)YLAT(J),(G(I,J),I=1,NIX)
  830       FORMAT(1H ,F5.2,120A1)
  900       CONTINUE
      RETURN
      END
C------------------------------------------------------------------
C********SUB TO CONVERT GRID I/J TO LAT LON
C           DEFAULT SET TO 25 BY 60 GRID
C           REMAINDER IN TENTHS AND HUNDREDS OF A DEGREE IN IREM
      SUBROUTINE INDEX
      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
      ENTRY INDEX1(TLAT,TLON,ILAT,JLON,IREM,*)
      MLAT=TLAT*100.0
      MLON=TLON*100.0
      ILAT=TLAT-(ALATB-1.0)
      JLON=TLON-(ALONR-1.0)
      IREM=MOD(MLAT,100)*100+MOD(MLON,100)
      IF(ILAT.GT.25.OR.ILAT.LT.1)RETURN 1
      IF(JLON.GT.60.OR.JLON.LT.1)RETURN 1
      RETURN
      ENTRY INDEX2(IREM,ILAT,JLON,TLAT,TLON)
      TLAT=ILAT+(ALATB-1.0)+(IREM/100)/100.0
      TLON=JLON+(ALONR-1.0)+(IREM-IREM/100*100)/100.0
      RETURN
      END
C------------------------------------------------------------------
C********SUB TO INITILIZE CERTAIN ARRAYS
C           FUNCTION VALUES ARE PUT IN TABLES (SINE AND COSINE)
C           MISSING DATA INDEX (-1) IN COMMON BLOCK BEFORE READ
      SUBROUTINE INITLZ(NPUFF)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100),
     * IDSTN(25,60)
      REAL*4 COS360(360),SIN360(360),COSLAT(25)
      COMMON/GRID/ALONL,ALONR,ALATT,ALATB
      COMMON/TRANS/COS360,SIN360,COSLAT
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      NPUFF=0
      DO 50 I=1,360
      ANG=I*0.0174533
      SIN360(I)=SIN(ANG)
      COS360(I)=COS(ANG)
   50 CONTINUE
      DO 75 I=1,25
      ANG=(I+ALATB-0.5)*0.0174533
      COSLAT(I)=COS(ANG)
   75 CONTINUE
      RETURN
      ENTRY INITL2
      DO 100 I=1,25
      DO 100 J=1,60
      IDSTN(I,J)=0
  100 CONTINUE
      DO 200 I=1,100
      MIXH(I)=-1
      DO 200 J=1,10
      WDIR(I,J)=-1
      WSPD(I,J)=-1
  200 CONTINUE
      RETURN
      END
C------------------------------------------------------------------
C********DIVIDE DAYTIME MIXED LAYER INTO NOCTURNAL SUBLAYERS
C           BELOW PREVIOUS DAYTIME MIXED DEPTH, PUFF MASS IS
C           DIVIDED ACCORDINGLY
      SUBROUTINE PSPLIT(NPUFF)
      REAL*4 MASS(05000)
      INTEGER*2 POSTN(05000),ARRAY(05000),IDSTN(25,60)
      INTEGER*2 WSPD(100,10),WDIR(100,10),MIXH(100),
     * SIGMA(05000)
      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,DUMMY(750)
      IF(NPUFF.LT.1)RETURN
      L2=NPUFF+1
```

```
      L1=1
  100 IT1=ARRAY(L1)/1000
      JT1=ARRAY(L1)/10-IT1*100
      KT1=ARRAY(L1)-ARRAY(L1)/10*10
      IF(KT1.NE.1)GO TO 300
      KMET=IDSTN(IT1,JT1)
      KMET=IABS(KMET)
      KMIX=MIXH(KMET)/300+1
      KMIX=MIN0(KMIX,10)
      MASS(L1)=MASS(L1)/KMIX
          DO 200 K=2,KMIX
          KM=K
          MASS(L2)=MASS(L1)
          POSTN(L2)=POSTN(L1)
          KM=MOD(KM,10)
          ARRAY(L2)=IT1*1000+JT1*10+KM
          SIGMA(L2)=SIGMA(L1)
          L2=L2+1
  200     CONTINUE
  300 L1=L1+1
      IF(L1.LE.NPUFF)GO TO 100
      NPUFF=L2-1
      IF(NPUFF.GT.05000)GO TO 400
      RETURN
  400 WRITE(6,500)NPUFF
  500 FORMAT(1H0,'EXCEEDS TRAJECTORY LIMIT IN PSPLIT BY ',I5)
      STOP
      END
C-----------------------------------------------------------------
C*********COMBINE COINCIDENT TRAJECTORY SUBLAYERS
C         THE MASS OF PUFFS AT THE SAME POSITION IS COMBINED
C         PUFFS WITH THE GREATEST MASS RETAIN THEIR POSITION
C         AND DIAMETER
      SUBROUTINE PCNVRG(NPUFF)
      INTEGER*2 IDSTN(25,60),POSTN(05000),ARRAY(05000)
      INTEGER*2 WDIR(100,10),WSPD(100,10),MIXH(100)
      INTEGER*2 KDSTN(25,60),SIGMA(05000)
      REAL*4 MASS(05000)
      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
      COMMON/MSTNS/IDSTN,WDIR,WSPD,MIXH,KDSTN
      IF(NPUFF.LT.2)RETURN
          DO 900 LOOP=1,2
              DO 50 I=1,25
              DO 50 J=1,60
              KDSTN(I,J)=0
   50         CONTINUE
              DO 500 L=1,NPUFF
              IF(MASS(L).EQ.0.0)GO TO 500
              IN=ARRAY(L)/1000
              JN=ARRAY(L)/10-IN*100
              KN=ARRAY(L)-ARRAY(L)/10*10
              IF(KN.EQ.0)KN=10
              IF(KDSTN(IN,JN))100,100,200
  100         KDSTN(IN,JN)=L
              GO TO 500
  200         LO=KDSTN(IN,JN)
              KO=ARRAY(LO)-ARRAY(LO)/10*10
              IF(KO.EQ.0)KO=10
              GO TO(210,220),LOOP
  210         IF(KO.EQ.KN)GO TO 250
              GO TO 500
  220         KMET=IDSTN(IN,JN)
              KMET=IABS(KMET)
              KMIX=MIXH(KMET)/300+1
              KMIX=MIN0(KMIX,10)
              IF(KO.GT.KMIX.OR.KN.GT.KMIX)GO TO 500
              ARRAY(LO)=IN*1000+JN*10+1
  250         IF(MASS(LO).GT.MASS(L))GO TO 300
              POSTN(LO)=POSTN(L)
              SIGMA(LO)=SIGMA(L)
  300         MASS(LO)=MASS(LO)+MASS(L)
              MASS(L)=0.0
  500         CONTINUE
  900     CONTINUE
```

```
      RETURN
      END
C-----------------------------------------------------------------------
C********ZERO MASS OR LOW MASS PUFFS ARE SORTED OUT OF ARRAY
C           LOW MASS VALUE CURRENTLY SET TO 1.0 UNITS
      SUBROUTINE PSORT(NPUFF,PMIN)
      INTEGER*2 POSTN(05000),ARRAY(05000),SIGMA(05000)
      REAL*4 MASS(05000)
      COMMON/PUFFS/MASS,POSTN,ARRAY,SIGMA
      IF(NPUFF.LT.2)RETURN
      LB=1
      LE=1
   10 IF(MASS(LE).GT.PMIN)GO TO 50
      LE=LE+1
      IF(LE.GT.NPUFF)GO TO 60
      GO TO 10
   50 MASS(LB)=MASS(LE)
      POSTN(LB)=POSTN(LE)
      ARRAY(LB)=ARRAY(LE)
      SIGMA(LB)=SIGMA(LE)
      LB=LB+1
      LE=LE+1
      IF(LE.LE.NPUFF)GO TO 10
   60 NPUFF=LB-1
      RETURN
      END
C-----------------------------------------------------------------------
C********ASSIGN MET DATA TO ALL ARRAY POSITIONS
C           ALL I,J OF KDSTN ASSIGNED SEQUENCE NUMBER OF NEAREST
C           METEOROLOGICAL STATION, EXACT I,J OF A STATION
C           HAS A NEGATIVE SEQUENCE NUMBER
      SUBROUTINE ASSIGM(KDSTN)
      REAL*4 COS360(360),SIN360(360),COSLAT(25)
      INTEGER*2 KDSTN(25,60)
      COMMON/TRANS/COS360,SIN360,COSLAT
      KB=0
           DO 100 I=1,25
           DO 100 J=1,60
           IF(KDSTN(I,J).NE.0)GO TO 100
                DO 85 KGRID=1,20
                KGRDI=KGRID
                KGRDJ=KGRID*COSLAT(I)+0.5
                IB=MAX0(I-KGRDI,1)
                IE=MIN0(I+KGRDI,25)
                JB=MAX0(J-KGRDJ,1)
                JE=MIN0(J+KGRDJ,60)
                JB1=MIN0(JB+1,60)
                JE1=MAX0(JE-1,1)
                     DO 70 II=IB,IE
                     KSTA=KDSTN(II,JB)
                     IF(KSTA.LT.0)GO TO 90
                     KSTA=KDSTN(II,JE)
                     IF(KSTA.LT.0)GO TO 90
   70                CONTINUE
                IF(JB1.GE.J.OR.JE1.LE.J)GO TO 85
                     DO 75 JJ=JB1,JE1
                     KSTA=KDSTN(IB,JJ)
                     IF(KSTA.LT.0)GO TO 90
                     KSTA=KDSTN(IE,JJ)
                     IF(KSTA.LT.0)GO TO 90
   75                CONTINUE
   85           CONTINUE
           KB=KB+1
           GO TO 100
   90      KDSTN(I,J)=-KSTA
  100      CONTINUE
      IF(KB.EQ.0)GO TO 200
      WRITE(6,150)KB
  150 FORMAT(27X,I4,' UNASSIGNED POSITIONS ON GRID')
  200 RETURN
      END
```