LA-UR-09-1601

Approved for public release;
distribution is unlimited.

Title:	POSTMAX V2.0 User's Guide
Author(s):	Raymond F. Sartor
Intended for:	

Table of Contents

List of Appendices

List of Figures

List of Tables

Acknowledgements

The computer code presented in this report, POSTMAX V2.0, is a revision of the POSTMAX code developed by Bruce C. Letellier and Scott G. Ashbaugh. Their work on POSTMAX and LA-UR-01-1461 is greatly appreciated.

The Coupled LPF-Dispersion Analysis option in POSTMAX V2.0 was originally developed by Hans Jordan as a modified version of the original POSTMAX code. His work produced a rigorous and consistent method for dose consequences when the atmospheric dispersion (χ/Q) and the leak path factor (LPF) are both dependent on wind direction and speed.

1. Purpose

POSTMAX can be part of a two-step process to determine the 95th percentile MEOI χ/Q for a facility and accident. In the first step, MACCS2 is used to determine the dispersion values (χ/Q) according to the facility parameters and the hourly meteorological data. The models and instructions for MACCS2 are provided in NUREG/CR-4691 and NUREG/CR-6613. MACCS2 generates a table of χ/Q according to distance for each hour evaluated. POSTMAX reads the MACCS2 output, meteorological data, and site boundary distances to calculate the Maximally-Exposed Offsite Individual (MEOI) χ/Q for each hour evaluated. The MEOI χ/Q values are sorted to generate the cumulative distribution function (CDF) and the 95th percentile value is identified.

2. Program Version

This manual documents Version 2.0 of POSTMAX. The POSTMAX V2.0 executable file (for the Windows XP operating system) is dated **Tue 2/03/2009, 17:52:21 GMT**.

> For the purposes of this manual, "POSTMAX V2.0" and "POSTMAX" are equivalent, and both names are used in this manual.

3. Installation

3.1 Program

To install POSTMAX V2.0, copy the POSTMAX2.exe file from the CD to a directory on your Windows XP workstation. POSTMAX will be most useful when installed in the directory where the MACCS2 executable (MACCS2.exe) is located.

3.2 Test Cases

Copy the directory "Test" and file "Test.bat" to the directory where POSTMAX V2.0 has been placed. Execute "Test.bat" from a Windows XP command prompt window or Windows Explorer window.

To return the "Test" directory to original condition, copy and execute the file Clean.bat.

3.3 Verify Test Results

Copy the directory "Benchmark" and files "Compare.exe" and "filelist.txt" to the directory where POSTMAX V2.0 has been placed. Execute Compare from a Windows XP command prompt window or Windows Explorer window. Using the filenames in filelist.txt, the Compare program does a line by line comparison of the files in directories Benchmark and Test. Compare allows the file time-stamps in the two files to be different; if any other difference is found, Compare aborts. If Compare runs to "Normal completion" then the two sets are identical (except for some file time-stamps). Compare writes its' output to the screen and file "Compare.txt".

4. Program Execution

POSTMAX V2.0 is designed to execute within a Windows XP command prompt window (often called a DOS window) or batch file. Normally, POSTMAX V2.0 does not require any user input beyond the command line. The only time additional user input is required is when the output cdo file exists when POSTMAX is executed; in this case, POSTMAX asks whether the file should be overwritten.

The POSTMAX V2.0 command line must be in the following forms:

```
POSTMAX2  1  infile bndfile <cdofile>
```

```
POSTMAX2  2  lpffile infile bndfile <cdofile>
```

```
POSTMAX2  3  outfile cdofile1 cdofile2 <cdofile3> <cdofile4> <cdofile5> <cdofile6>
```

where

option	=	1 for analysis of MACCS2 results,
	=	2 for analysis of MACCS2 and LPF results,
	=	3 for merging/sorting previous POSTMAX2 results,
infile.out	=	MACCS2 output file used for POSTMAX2 input,
bndfile.txt	=	site boundary input file,
cdofile.cdo	=	optional POSTMAX2 output file; input/output for the multi-year statistical analysis,
lpffile.txt	=	LPF table input file (required only when option = 2),
outfile.cdo	=	POSTMAX2 output file; output for the multi-year statistical analysis,
cdofile#.cdo	=	POSTMAX2 single-year output file; input for the multi-year statistical analysis, and
<filename>	=	optional filename.

Chapter 7 discussed two types of analyses, the constant source term analysis (to determine the 95th percentile atmospheric dispersion values) and the coupled LPF-dispersion analysis (to determine the 95th percentile of LPF times the atmospheric dispersion values). The following sections will explain how to perform these analyses and the third option for POSTMAX V2.0.

4.1 Constant Source Term Analysis

The constant source term analysis is performed to determine the 95th percentile values for ground-level χ/Q without deposition, ground-level time-integrated air concentration, and ground-level puff release χ/Q without deposition. POSTMAX V2.0 performs this type of analysis when the command line option (first argument) is equal to one (1).

4.1.1 Single-Year Analysis

The 95th percentile atmospheric dispersion values for a single year are determined when the MACCS output and site boundary distance files are specified in the command line.

4.1.2 Multi-Year Analysis

When an output file from a previous POSTMAX V2.0 run is specified (in addition to the MACCS output and site boundary distance files), POSTMAX

- determines the 95th percentile values for the single year contained in the MACCS output file, and
- determines the 95th percentile values for the years contained in the MACCS output file and POSTMAX V2.0 output file.

4.2 Coupled LPF-Dispersion Analysis

The coupled LPF-dispersion analysis is performed to determine the 95th percentile values of LPF times ground-level χ/Q without deposition, ground-level time-integrated air concentration, and ground-level puff release χ/Q without deposition. POSTMAX V2.0 performs this type of analysis when the command line option (first argument) is equal to two (2).

4.2.1 Single-Year Analysis

The 95th percentile of LPF times the atmospheric dispersion values for a single year is determined when the LPF table file, MACCS output, and site boundary distance files are specified in the command line.

4.2.2 Multi-Year Analysis

When an output file from a previous POSTMAX V2.0 run is specified (in addition to the LPF table file, MACCS output, and site boundary distance files), POSTMAX

- determines the 95th percentile of LPF times the atmospheric dispersion values for the single year contained in the MACCS output file, and
- determines the 95th percentile of LPF times the atmospheric dispersion values for the years contained in the MACCS output file and POSTMAX output file.

4.3 Multi-Year Analysis from Single-Year Output Files

When two to six single-year POSTMAX V2.0 output files are available, POSTMAX can read these single-year cdo files, sort the data, and determine the multi-year 95th percentile values. These type of analysis is performed with option three (3) in the command line. The analysis can be performed with POSTMAX files from either constant source term or coupled LPF-dispersion analyses. POSTMAX does not perform any consistency checks, therefore, it is incumbent on the user to ensure that the POSTMAX output specified in the command line are all from one type of analysis.

4.4 Trace File Option

When the POSTMAX V2.0 option is negative, POSTMAX will perform as described above for the positive option value. In addition, POSTMAX will write a trace file. The trace file reports the intermediate actions and results performed by POSTMAX. More information on the trace file is provided in Section 6.4.

5. Input Files

This chapter explains the input provided to POSTMAX V2.0. The POSTMAX inputs (and outputs) for the constant source term and coupled LPF-dispersion analyses (discussed in Chapter 7) are shown in Figure 1. POSTMAX can also evaluate multiple years using single-year distributions previously calculated by POSTMAX. The POSTMAX inputs (and outputs) for this third type of analysis are shown in Figure 2.

5.1 MACCS Output File

MACCS2 calculates straight-line Gaussian dispersion results using site-specific meteorological data and generates an output file. POSTMAX V2.0 uses the MACCS2 output file to calculate the 95th percentile MEOI values. MACCS2 output files have the filename extension "out". When running POSTMAX, include the filename without the extension in the POSTMAX command line.

The file may be in a different directory from POSTMAX. If the file is in a different directory from POSTMAX, include the path with the filename in the POSTMAX command line. All POSTMAX output files will be written to the directory where the MACCS2 output file is located.

POSTMAX has been validated with MACCS2 version 1.13.1. To use POSTMAX with newer versions of MACCS2, the validation cases will have to be re-performed.

5.2 Site Boundary File

The site boundary file is a user-generated text file. The filename extension must be "txt". When running POSTMAX, include the filename without the extension in the POSTMAX command line. The file may be in a different directory from POSTMAX. If the file is in a different directory from POSTMAX, include the path with the filename in the POSTMAX command line.

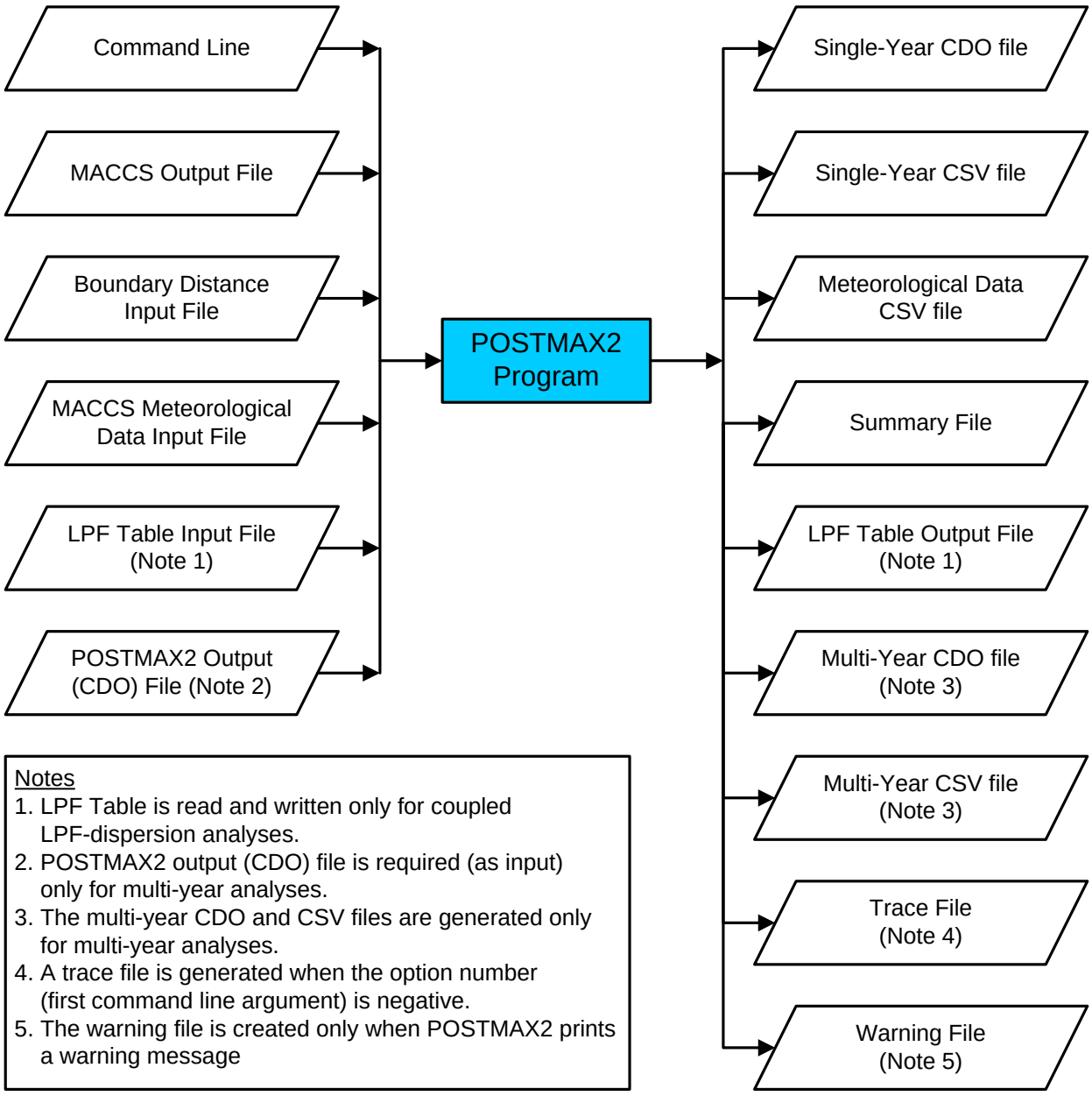

Figure 1
Input/Output for Constant Source Term and Coupled LPF-Dispersion Analyses

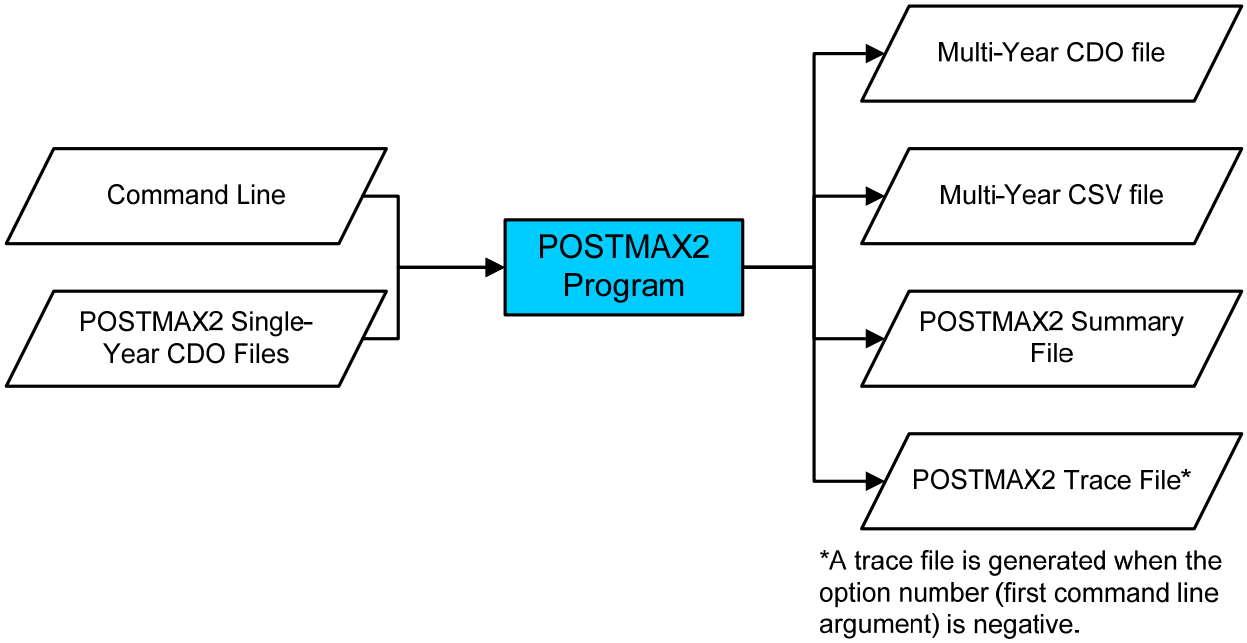

Figure 2
Input/Output for Multi-Year Analysis from Single-Year Output Files

The first two lines of the site boundary file is the file header. The file header can be any text and has no effect on the POSTMAX results. Lines 3 through 18 are the site boundary distances for the compass directions, starting with North going clockwise. See Section 8.7 for additional discussion on the site boundary distances.

5.3 MACCS2 Meteorological Data Input File

A meteorological data file was created for input to MACCS2. POSTMAX uses the same meteorological data input file. However, the user does not directly provide the MACCS2 meteorological data input filename. The MACCS2 output file identifies the meteorological data file used by MACCS2. POSTMAX retrieves the filename (including the path) from the MACCS2 output file.

5.4 LPF Table File

The LPF table file is a user-generated text file. The filename extension must be "txt". For coupled LPF-dispersion analyses, include the filename without the extension in the POSTMAX V2.0 command line. The file may be in a different directory from POSTMAX. If the file is in a different directory from POSTMAX, include the path with the filename in the POSTMAX command line.

The first two lines of the LPF table file is the file header. The file header can be any text and has no effect on the POSTMAX results. The LPF table begins on line 3. The LPF table must have at least one row. A table row may be placed on one or more lines in the

test file. The maximum number of rows is limited only by the array dimension in POSTMAX; the current limit is 25 rows. Each row of the LPF table is a wind speed value (in meters per second) followed with 16 LPF values. The LPF values are for the 16 compass directions, starting with North going clockwise. The first wind speed must be greater than zero and less than or equal to 0.5. The wind speed must be in increasing order. Identical wind speeds are not allowed in the LPF table.

5.5 POSTMAX Output File

The primary output of POSTMAX are CDO files containing a cumulative distribution frequency (CDF) table. In order to determine the 95th percentile MEOI values for multiple years, POSTMAX reads the CDF table for previous years from the CDO files. With options 1 and 2, the POSTMAX output file in the command line may be a single-year CDO or multi-year CDO. Also, with options 1 and 2, POSTMAX replaces the POSTMAX output file with the new multi-year CDO. Option 3 uses only single-year POSTMAX CDO files for input and generates a new file for the multi-year CDO.

The filename extension must be "cdo". For multi-year analyses, include the filename without the extension in the POSTMAX command line. The file may be in a different directory from POSTMAX. If the file is in a different directory from POSTMAX, include the path with the filename in the POSTMAX command line.

6. Output Files

This chapter explains the output generated by POSTMAX V2.0. The POSTMAX outputs (and inputs) for the constant source term and coupled LPF-dispersion analyses (discussed in Section 1) are shown in Figure 1. POSTMAX can also evaluate multiple years using single-year distributions previously calculated by POSTMAX. The POSTMAX outputs (and inputs) for this third type of analysis are shown in Figure 2.

6.1 CDO Files

The primary result of POSTMAX is the CDO file. Single-year CDO files are named after the MACCS output file with the extension changed to "cdo". For constant source term and coupled LPF-dispersion analyses (POSTMAX V2.0 option 1 or 2), the CDO filename specified in the command line for input is also used for output, i.e., the input CDO file is overwritten with the new multi-year CDF. For single-year distributions previously calculated by POSTMAX (option 3), the multi-year CDO filename is specified in the command line. POSTMAX places the CDO output file in the same directory as the MACCS2 output file. Despite the "cdo" filename extension, the CDO file is an ASCII text file.

The CDO file reports the cumulative distribution frequency (CDF) calculated by POSTMAX V2.0 in plain text. The CDF may be for a single year or for multiple years. In either case, the CDO file has three sections.

6.1.1 Execution Information

The top of the file reports information on the execution of POSTMAX V2.0. For single-year CDO files, the execution information begins with an echo of the command line. Then each input file is identified with the file date and time; time is given in Greenwich Mean Time (GMT). For multi-year CDO files, the execution information for each POSTMAX V2.0 execution that formed the multi-year CDF is listed. For constant source term and coupled LPF-dispersion analyses (POSTMAX option 1 or 2), the execution information is in chronological order. For single-year distributions previously calculated by POSTMAX (option 3), the execution information follows the file order in the command line.

The MACCS2 meteorological data input files are included in the list of input files. Multi-year CDO files will have several meteorological data input files listed. Each meteorological data input file is assigned a number (1 through 6). These numbers are used in the cumulative distribution frequency table to identify the meteorological data for each hourly result.

6.1.2 Cumulative Distribution Frequency

The cumulative distribution frequency (CDF) is a table of the following hourly results.

Table 1
POSTMAX CDF Table

Column	Value
1	Normalized Probability
2	Ground-Level χ/Q Without Deposition*
3	corresponding meteorological data file number
4	corresponding meteorological file day
5	corresponding meteorological file hour
6	Ground-Level Puff Release χ/Q Without Deposition*
7	corresponding meteorological data file number
8	corresponding meteorological file day
9	corresponding meteorological file hour
10	Ground-Level Time-Integrated Air Concentration*
11	corresponding meteorological data file number
12	corresponding meteorological file day
13	corresponding meteorological file hour
14	Normalized Probability (same as column 1)

*For coupled LPF-dispersion analyses, these columns will be the value times LPF for the meteorological conditions of the hour.

6.1.3 95th Percentile Summary

For constant source term analyses, the 95th percentile results for ground-level χ/Q without deposition, ground-level time-integrated air concentration, and ground-level puff release χ/Q without deposition are listed.

In the case of coupled LPF-dispersion analyses, the 95th percentile results for LPF times ground-level χ/Q without deposition, ground-level time-integrated air concentration, and ground-level puff release χ/Q without deposition are listed.

6.2 CSV (Comma Separated Value) Files

The CSV output file is identical to the CDO file, except the CDF table is written with commas between the column headers and values. POSTMAX V2.0 places the CSV file in the same directory as the MACCS2 output file. The comma separated value (CSV) format is more convenient than the plain text in the CDO file when importing the CDF table into Excel for plotting.

6.3 Meteorological Data CSV File

The meteorological data CSV file is an echo of the meteorological data used by POSTMAX V2.0. The meteorological data CSV file is ASCII text with the values separated by commas. The meteorological data CSV file is named after the MACCS2 meteorological data input file with the filename extension changed to "csv". POSTMAX places the meteorological data CSV file in the same directory as the MACCS2 output file. The comma separated value (CSV) format is more convenient than plain text when importing the meteorological data into Excel for plotting.

6.4 Summary File

POSTMAX V2.0 generates a summary file named "POSTMAX_Summary.txt" in each output directory. The summary file lists the 95th percentile results of each (uniquely named) CDO file written to the directory. Table 2 lists the information written to the summary file. If POSTMAX_Summary.txt does not exist when POSTMAX writes the CDO file to the directory, POSTMAX will create POSTMAX_Summary.txt, write the headers, and record the results of this CDO file. When POSTMAX writes a new CDO file to the directory, the 95th percentile information is appended to POSTMAX_Summary.txt. If POSTMAX replaces an existing CDO file, then the old information for this CDO file is deleted from POSTMAX_Summary.txt and the new information is appended.

Table 2
POSTMAX Summary File

Column	Value
1	95th percentile Ground-Level χ/Q Without Deposition*
2	95th percentile Ground-Level Puff Release χ/Q Without Deposition*
3	95th percentile Ground-Level Time-Integrated Air Concentration*
4	Date for the CDO file (see column 6)
5	Time (in GMT) for the CDO file (see column 6)
6	Name of the CDO file with the CDF for these results

*For coupled LPF-dispersion analyses, these columns will be the 95th percentile value of LPF times this parameter.

After the POSTMAX_Summary.txt is created, POSTMAX assumes that only one type of analysis, constant source term or coupled LPF-dispersion, is performed in the directory. If the two types of analyses are mixed up in the directory, the column headers will not be correct for some CDO files.

6.5 LPF Table Output File

If a coupled LPF-dispersion analysis is performed, POSTMAX V2.0 echoes the LPF table to an output file. The LPF table output file is named after the MACCS output file with the filename extension changed to "lpf". POSTMAX places the LPF table output file in the same directory as the MACCS2 output file. Despite the "lpf" filename extension, the LPF table is an ASCII text file.

6.6 Trace File

The trace file option is provided as a tool for identifying and solving problems with POSTMAX V2.0. The trace file is generated when the POSTMAX option (first command line argument) is negative. The trace file is named after the MACCS output file with the extension changed to "trc". POSTMAX places the trace file in the same directory as the MACCS2 output file. Despite the "trc" filename extension, the trace file is an ASCII text file.

The trace file provides the following information:

- The execution information (as explained for the CDO file),
- An echo of the text in the site boundary distance file,
- The values read from the site boundary distance file,
- An echo of the text in the MACCS2 meteorological data input file,

- The day, hour, wind direction, wind speed, and stability class values read from the MACCS2 meteorological data input file,
- Parameter checking of the MACCS output file, i.e., the lines containing variables OCIDEBUG001, M1METCOD001, GENUMRAD001, and M4NSMPLS001; and the values read for these variables,
- For each MACCS trial the following are listed:
    - the MACCS trial results,
    - the wind direction and boundary distance,
    - the results by lookup or interpolation for σ_x, ground-level χ/Q without deposition, and ground-level time-integrated air concentration,
    - the calculated ground-level puff release χ/Q without deposition, and
    - for a coupled LPF-dispersion analysis, the LPF value and product of LPF and ground-level χ/Q without deposition, ground-level time-integrated air concentration, and ground-level puff release χ/Q without deposition.
- The unsorted hourly values are listed,
- The sort orders are listed, and
- The sorted hourly values are listed (similar to the CDF table written to the CDO and CSV files).

6.7 Warning File

At times, POSTMAX V2.0 may encounter anomalous conditions that may, or may not, affect the 95th percentile results. The anomalous conditions POSTMAX V2.0 checks for are:

- For a particular MACCS2 trial, the χ/Q without deposition values consistently increase from the site boundary to the last (farthest) location analyzed. POSTMAX is unable to determine the χ/Q without deposition value for the MEOI for this trial (hour).
- For a particular MACCS2 trial, the ground-level air concentrations consistently increase from the site boundary to the last (farthest) location analyzed. POSTMAX is unable to determine the ground-level air concentration for the MEOI for this trial (hour).
- For a particular MACCS2 trial, the quotient of the χ/Q without deposition values and crosswind standard deviation (σ_y) consistently increase from the site boundary to the last (farthest) location analyzed. POSTMAX is unable to

determine the puff release χ/Q (without deposition) value for the MEOI for this trial (hour).

These warning messages may indicate that the spatial grid in the MACCS2 input file needs to be revised. Or it may be that the values for this particular trial (hour) are already insignificant compared to the 95th percentile results and a more precise value for this trial is not beneficial. The warning file is created to record these conditions and enable the analyst to determine if MACCS2 input revisions are required.

7. Error Reporting

If a problem or error is encountered with POSTMAX V2.0, please contact one of the following:

Contact	Phone	E-mail
Analytical Support Team (Ray Sartor)	505-665-3166	sartor@lanl.gov
Technical Services Group (Ron Selvage)	505-665-0810	rselvage@lanl.gov
Safety Basis Division	505-665-1951	

8. Theory

This chapter explains how POSTMAX V2.0 supports a dose consequence analyses for a Documented Safety Analysis.

8.1 DSA Requirements

The contractor responsible for a hazard category 1, 2, or 3 DOE nuclear facility must obtain approval from DOE for the methodology used to prepare the documented safety analysis for the facility unless the contractor uses a methodology set forth in Table 2 of Appendix A to 10 CFR 830 [10 CFR 830.204(a)]. In Table 2 of Appendix A to 10 CFR 830, DOE Standard 3009 [STD-3009] is the specified methodology for a DOE nonreactor nuclear facility. DOE Standard 3009 is also listed for other types of DOE facilities.

DOE Standard 3009 states that the 95th percentile of the distribution of doses to the maximally-exposed offsite individual (MEOI) is the comparison point for assessment against the Evaluation Guideline (EG). The dose distribution must account for variations in distance to the site boundary as a function of direction. The method used should be consistent with the statistical treatment of calculated χ/Q values described in regulatory position 3 of NRC Regulatory Guide 1.145 for the evaluation of consequences along the exclusion area boundary. [STD-3009, Sec. A.3.3]

8.2 Dose Consequences

The equation for calculating the dose consequence from a release is:

$$D = ST \times \frac{\chi}{Q} \times BR \times DCF \quad (1)$$

Where

ST = Source Term (the amount of material released from the facility),

χ/Q = atmospheric dispersion parameter,

BR = Breathing Rate, and

DCF = Dose Conversion Factor.

The equation for calculating the source term is:

$$ST = MAR \times DR \times RF \times ARF \times LPF \quad (2)$$

Where

MAR = Material At Risk,

DR = Damage Ratio,

RF = Release Fraction,

ARF = Airborne Release Fraction, and

LPF = Leak Path Factor.

8.3 MACCS2 Capabilities and Limitations

DOE-STD-3009 states "Accident phenomenology may be modeled assuming straight-line Gaussian dispersion characteristics, applying meteorological data representing a 1-hour average for the duration of the accident." MACCS2 is one way of performing straight-line Gaussian dispersion using site-specific meteorological data, but MACCS2 does not provide the 95th percentile χ/Q.

The user defines radial regions and MACCS calculates the dispersion parameter (χ/Q) for the center of each region. χ/Q values depend on the wind speed and stability class; MACCS2 can obtain the wind speed and stability class by sampling a meteorological input data file. For a DSA analysis, each hour of the year is analyzed, i.e., one MACCS run produces 8760 results tables.

The first limitation of MACCS2 is that the MACCS2 χ/Q values correspond to locations on the concentric circles. MACCS2 χ/Q values do not correspond to the site boundary. Also, each hourly χ/Q value must correspond to the site boundary distance for the direction the wind is blowing that hour.

The second limitation is MACCS2 does not calculate the cumulative probability distribution and determine the 95th percentile results. The cumulative distribution for the year is developed by sorting the χ/Q values in ascending order and selecting the 95th percentile value.

However, MACCS2 has several benefits, including a dry deposition model. If dry deposition was neglected from the dispersion analysis, the resulting χ/Q would be unrealistically over-estimated. However, the user must properly use the MACCS2 output file to obtain a χ/Q with deposition. The MACCS output file is explained later.

8.4 Statistical Dose Consequence Analysis (Constant Source Term)

In most cases, χ/Q is the only parameter considered to be dependent on the meteorological conditions. In this case, the form of eq. 1 for the statistical analysis is:

$$D_{95\%} = ST \times \left(\frac{\chi}{Q}\right)_{95\%} \times BR \times DCF \tag{3}$$

That is, by determining and using the 95th percentile of χ/Q, the 95th percentile dose consequence is calculated. Later on, we will examine the special case where the source term is dependent on the meteorological conditions.

8.5 Atmospheric Dispersion

The atmospheric dispersion parameter (χ/Q) is typically calculated using the Gaussian dispersion model. The following version of the Gaussian dispersion equation is used in MACCS [NUREG/CR-4691; Sec. 2.9.1]:

$$\begin{aligned}\chi(x, y=0, z) = \frac{Q}{2\pi\sigma_y\sigma_z\bar{u}}\Bigg(&\exp\left[-\frac{1}{2}\left(\frac{z-H}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H}{\sigma_z}\right)^2\right] \\ &+ \sum_{n=1}^{5}\Bigg\{\exp\left[-\frac{1}{2}\left(\frac{z-H-2nL}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H-2nL}{\sigma_z}\right)^2\right] \\ &+ \exp\left[-\frac{1}{2}\left(\frac{z-H+2nL}{\sigma_z}\right)^2\right] + \exp\left[-\frac{1}{2}\left(\frac{z+H+2nL}{\sigma_z}\right)^2\right]\Bigg\}\Bigg)\end{aligned} \tag{4}$$

where

$\chi(x, y=0, z)$ = the time-integrated air concentration (Bq-s/m^3) at the downwind location (x, y=0, z), along the plume centerline,

Q = the source strength (Bq),

$\bar{u}$ = the mean wind speed (m/s),

σ_y and σ_z = the standard deviations (m) of the normal crosswind and vertical concentration distributions of plume materials,

H = $h + \Delta h$ = the height of the plume centerline (m),

h = the initial release height (m) of the plume before plume rise,

Δh = the amount of plume rise (m), and

L = the height (m) of the capping inversion layer (mixing height).

For the ground-level concentrations (z=0) reported by MACCS and stack releases, the dispersion equation simplifies to:

$$\chi(x, y=0, z=0) = \frac{Q}{\pi \sigma_y \sigma_z \bar{u}} \left(\exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z} \right)^2 \right] + \sum_{n=1}^{5} \left\{ \exp\left[-\frac{1}{2}\left(\frac{H - 2nL}{\sigma_z} \right)^2 \right] + \exp\left[-\frac{1}{2}\left(\frac{H + 2nL}{\sigma_z} \right)^2 \right] \right\} \right) \quad (5)$$

It is obvious from the equations above that the atmospheric dispersion parameter (χ/Q) depends on the wind speed when the release occurs.

8.6 Standard Deviations of the Crosswind And Vertical Concentration Distributions

The standard deviations of the crosswind and vertical concentration distributions depend on the meteorological conditions, specifically, the stability class. Stability class is a measure of the turbulence in the wind; turbulence increases dispersion and decreases χ/Q. There are several correlations to calculate the standard deviations. The Tadmor and Gur correlation is explained below to illustrate the dependency on stability class.

The Tadmor-Gur standard deviations of the plume spread over flat terrain covered by grass (surface roughness length (z_0) of 3 cm) are calculated with the equations [NUREG/CR-4691; Sec. 2.9.2]:

$$\sigma_y(x) = a_i \, x^{b_i} \quad (6)$$

$$\sigma_z(x, 3\,\text{cm}) = c_i \, x^{d_i} \quad (7)$$

where

x = distance from the release point in the downwind direction,

$\sigma_y(x)$ = standard deviation of the horizontal crosswind plume spread (m) at point x, and

$\sigma_z(x, z_0)$ = standard deviation of the vertical plume spread (m) at point x with a surface roughness of z_0.

where the constants are chosen according to Stability Class in Table 3.

Table 3
Tadmor-Gur Correlation Coefficients

Stability Class	a_i	b_i	c_i	d_i
A	0.3658	0.9031	0.00025	2.125
B	0.2751	0.9031	0.0019	1.6021
C	0.2089	0.9031	0.2	0.8543
D	0.1474	0.9031	0.3	0.6532
E	0.1046	0.9031	0.4	0.6021
F	0.0722	0.9031	0.2	0.602

The Tadmor-Gur standard deviations are illustrated in Figure 3 and Figure 4.

8.7 Distance to the Site Boundary

The MEOI is a member of the public located on (or beyond) the site boundary. The MEOI is initially assumed to be located at the site boundary. If a higher dose occurs past the site boundary, then the maximum dose is used.

Regulatory Guide 1.145, Regulatory Position 1.2 requires the site boundary to be divided into sixteen 45º sectors. (The sector overlap for conservatism.) Within each sector, the point on the site boundary closest to the facility is the site boundary distance. Therefore, the site boundary distance depends on the wind direction. Since site boundary distance must be used to calculate the χ/Q value (Sec. 8.3 and 8.6), the χ/Q value depends on the wind direction.

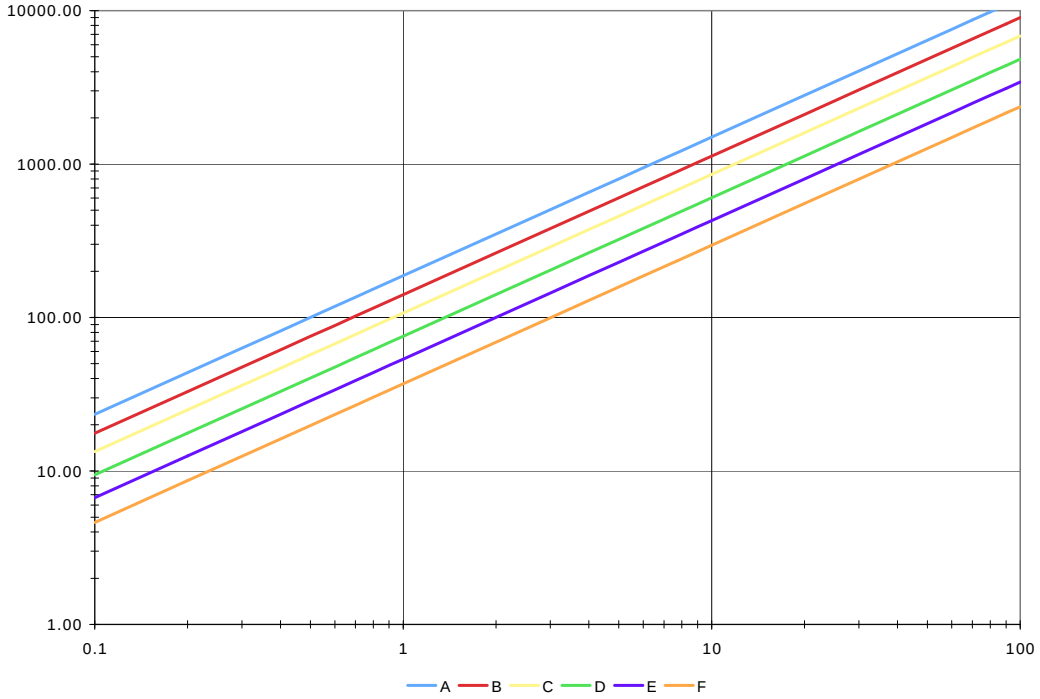

Figure 3
Tadmor-Gur Crosswind Standard Deviation (σ_y)

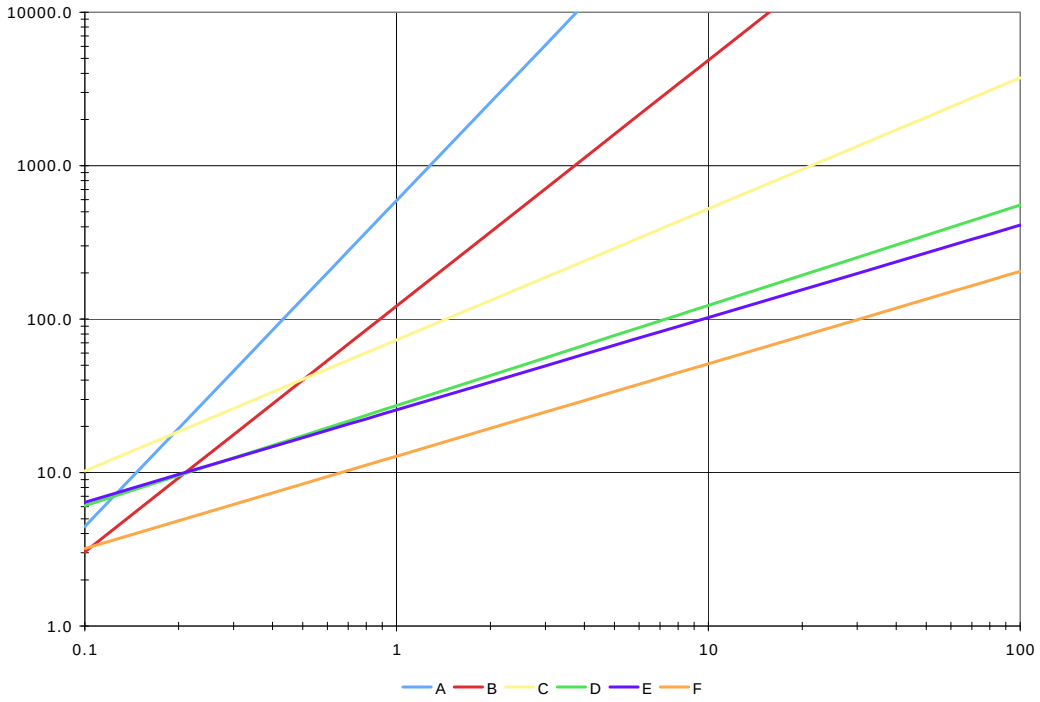

Figure 4
Tadmor-Gur Vertical Standard Deviation (σ_z)

8.8 MACCS2 Results Without Deposition

For DSA calculations, MACCS2 is used to analyze the hourly meteorological conditions for a full year. For each meteorological condition analyzed, MACCS2 generates a table of results in the output file. The first column in the results table is the distance; the results in each row correspond to the distance in column 1. The results table has one column labeled "GL X/Q"; this column is the ground-level χ/Q without deposition.

8.9 Hourly χ/Q (Without Deposition)

8.9.1 χ/Q at the Site Boundary

For some releases, the airborne concentration and χ/Q value decrease with distance. For example, this occurs with a ground-level release and a ground-level receptor. In this case, the maximum dose occurs at the minimum distance from the facility or release to an offsite individual. Therefore, the minimum distance to the site boundary in the wind direction, as explained in Section 8.7, determines the MEOI dose. Unless the site boundary aligns with the midpoint of the MACCS2 radial grid, interpolation of the MACCS2 results will be necessary.

The atmospheric dispersion equation (eq. 5) does not have an explicit dependence on the downwind distance (x). The x-dependency is in the standard deviations of the crosswind and vertical concentration distributions. The Pasquill-Gifford σ_y and σ_z dispersion values are linear on a log-log scale (Section 8.6). Therefore, χ/Q will be approximately linear on a log-log scale.

To determine the χ/Q value at the site boundary, interpolate the MACCS2 results table to get the ground-level concentration value at the site boundary distance. Since the χ/Q is approximately linear on a log-log scale, the interpolation equation is:

$$\frac{\chi}{Q}(x) = \exp\left[\ln\left(\frac{\chi}{Q}(x_2)\right) + \frac{x - x_2}{x_2 - x_1}\left(\ln\left(\frac{\chi}{Q}(x_2)\right) - \ln\left(\frac{\chi}{Q}(x_1)\right)\right)\right] \quad (8)$$

where

x = site boundary distance,

x_1 = MACCS spatial grid location closest to x while $x > x_1$,

x_2 = MACCS spatial grid location closest to x while $x < x_2$,

$\chi/Q(x_1)$ = ground-level concentration at distance x_1,

$\chi/Q(x_2)$ = ground-level concentration at distance x_2, and

$\chi/Q(x)$ = ground-level concentration at distance x.

8.9.2 χ/Q Beyond the Site Boundary

For some releases, the maximum χ/Q value occurs beyond the site boundary. This may happen if the release is elevated above the ground or if the plume is heated, making the plume buoyant. In these cases, the plume may effectively be high in the atmosphere at the site boundary, but touchdown on the ground past the site boundary.

8.9.3 MEOI χ/Q

The hourly χ/Q value for the MEOI is the maximum value of either the interpolated site boundary value or a MACCS output value for a distance greater than the site boundary.

8.10 95th Percentile χ/Q (Without Deposition)

Once the hourly χ/Q values have been calculated, the 95th percentile value of the year (or several years) can be determined. The hourly values are sorted in ascending order and the 95th percentile value is identified. That is, the value that is greater than or equal to 95% of the hourly values is the 95th percentile result.

8.11 MACCS2 Results With Deposition

In addition to the ground-level χ/Q results without deposition (GL X/Q), MACCS2 also reports the ground-level air concentration (labeled "GL AIRCON"). The ground-level air concentration excludes any material which has already been deposited on the ground. The ground-level χ/Q *with* deposition can be calculated from the MACCS2 ground-level air concentration by:

$$\chi/Q_{w/dep} = \frac{GLCON}{(CORINV)(CORSCA)} \tag{9}$$

where

GLCON = the (plume-centerline) ground-level time integrated concentration from POSTMAX (Bq-s/m^3),

CORINV = the core inventory, or source term, (Becquerels) specified in the MACCS2 input file, and

CORSCA = the scaling factor on the core inventory specified in the MACCS2 input file.

Since CORINV and CORSCA are constant for a MACCS2 run, the 95th percentile ground-level air concentration can be used to calculate the 95th percentile χ/Q value with deposition, i.e.,

$$(\chi/Q_{w/dep})_{95th\%} = \frac{(GLCON)_{95th\%}}{(CORINV)(CORSCA)} \tag{10}$$

8.12 95th Percentile Ground-Level Air Concentration

The methods and equations presented in Sections 8.9 and 8.10 for the 95th percentile χ/Q value are applicable to calculating the 95th percentile ground-level air concentration from the hourly ground-level air concentrations.

Note: If multiple MACCS2 runs are used to determine the 95th percentile χ/Q value with deposition, then it is essential that the CORINV and CORSCA values are not changed between MACCS2 input files.

8.13 Atmospheric Dispersion for a Puff Release

The Gaussian dispersion equation for a puff release is [Dobbins; eq. 10.29] [Turner; eq. 4.29]:

$$\chi_{puff}(x,y,0;H) = \frac{2\,Q_T}{(2\pi)^{3/2}\,\sigma_x(t)\,\sigma_y(t)\,\sigma_z(t)} \times \exp\left[-\frac{1}{2}\left(\frac{x-ut}{\sigma_x(t)}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{y}{\sigma_y(t)}\right)^2\right] \exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z(t)}\right)^2\right] \quad (11)$$

Note the standard deviations (σ_x, σ_y, σ_z) for a puff release are expressed as functions of time after release (t), not the receptor location (x). Since the velocity of the puff (u) is assumed to be constant, the standard deviations could be expressed as functions of the puff location.

The maximum ground-level concentration occurs when the puff is directly over the receptor, i.e., x = ut. In this case, the ground-level concentration is [Turner; eq. 4.30]:

$$\chi_{puff}^{max}(x,0,0;H) = \frac{2\,Q_T}{(2\pi)^{3/2}\,\sigma_x(x)\,\sigma_y(x)\,\sigma_z(x)} \exp\left[-\frac{1}{2}\left(\frac{H}{\sigma_z(x)}\right)^2\right] \quad (12)$$

Since this equation is for, and only for, a puff directly over the receptor, the standard deviations (σ_x, σ_y, σ_z) can revert to functions of the receptor distance.

Dobbins states that the values of σ_y and σ_z for puff releases are derived from the steady flow experiments and refers the Pasquill-Gifford figures and Tadmor-Gur correlations. Turner (Section 4.19) and Wark, et. al.(Section 4-8-D) do not agree that the continuous release σ_y and σ_z are appropriate for puff releases. Turner states the σ_y and σ_z for a puff release are less than those for a continuous release and lists σ_y and σ_z values provided by D.H. Slade.

If one accepts that the σ_y and σ_z values for continuous releases are appropriate for puff releases (and neglects the summation term in eq. 5 as a minor contributor), then the correlation between a puff release and a continuous release is:

$$\frac{\chi_{puff}^{max}}{\chi_{cont}} = \frac{1}{\sqrt{2\pi}} \frac{\bar{u}}{\sigma_x} \frac{Q_T}{Q} \tag{13}$$

Although correlations for σ_x are unknown, it is expected that σ_x is approximately the same as σ_y. With this approximation, the puff release concentration can be calculated by

$$\left(\frac{\chi}{Q_T}\right)_{puff}^{max} = \frac{1}{\sqrt{2\pi}} \frac{\bar{u}}{\sigma_y} \left(\frac{\chi}{Q}\right)_{cont} \tag{14}$$

Where the previous assumption that the σ_y and σ_z values for continuous releases are appropriate for puff releases.

8.14 95th Percentile χ/Q for a Puff Release (Without Deposition)

The methods and equations presented in Sections 8.9 and 8.10 for the 95th percentile χ/Q value are applicable to calculating the 95th percentile χ/Q value for a puff release. Since the puff release is a quotient of the continuous release χ/Q and σ_y, both functions of x, the interpolation may be performed in one of two ways:

- Calculate the $(\chi/Q_T)_{puff}^{max}$ values at x_1 and x_2, then interpolate between them, or
- Interpolate the χ/Q and σ_y values, then calculate the $(\chi/Q_T)_{puff}^{max}$ value.

Either method is valid since χ/Q and σ_y are both linear on a log-log scale.

8.15 Leak Path Factor

LPF is the fraction of the airborne material inside the facility that is transported to the exterior environment (outside the facility). Typically, LPF is a constant value for a particular scenario. However, there are also analyses that model LPF as a value that varies according to the wind direction and speed. In these cases, LPF increases with wind speed while χ/Q decreases with wind speed. If the analysis independently selected 95th percentile values for LPF and χ/Q, the analysis would be based on simultaneous high and low wind speeds, a physical impossibility. To properly account for wind-dependent LPFs, the couple LPF-dispersion model is required.

8.16 Statistical Dose Consequence for Coupled LPF-Dispersion Analysis

As mentioned in Section 8.13, there are analyses that model LPF as a value that varies according to the wind direction and speed. In this case, the 95th percentile dose equation, from equations 1 and 2, is:

$$D_{95\%} = MAR \times DR \times RF \times ARF \times \left(LPF \times \frac{\chi}{Q}\right)_{95\%} \times BR \times DCF \tag{15}$$

That is, by determining and using the 95th percentile of the product of LPF and χ/Q, the 95th percentile dose consequence is calculated. When calculating the product value, the same meteorological conditions must be used for determining LPF and χ/Q.

8.17 LPF Table

Since this analysis method considers LPF to vary according to the wind direction and speed, a two-dimensional table of LPF values is required. The table format used here is: the wind speed is in column 1 followed by 16 LPF values for the wind direction. Table 4 is an example of an LPF table.

Table 4
LPF Table Example

	Wind Direction					
Wind Speed (m/s)	1	2	3	…	15	16
0.5	0.70	0.67	0.64	…	0.64	0.67
1.0	0.75	0.72	0.69	…	0.69	0.72
…	…	…	…	…	…	…

The MACCS2 meteorological data input file provides the wind direction and wind speed. The LPF table has a column for each wind direction, therefore, an exact match for wind direction can be found. However, the LPF table may not, probably will not, have an exact match for wind speed. For a hourly wind speed bracketed by two wind speeds in the LPF table, the two LPF values for the wind direction are examined and the maximum value is selected. If the hourly wind speed exceeds the last table row, the LPF value is selected from the last row.

8.18 95th Percentile Results for Coupled LPF-Dispersion Analysis

The coupled LPF-dispersion analysis is a minor variation of the constant source term analysis. The hourly values of ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release ground-level χ/Q without deposition are calculated as discussed above. The LPF for the same hour is determined as discussed in Section 8.17. The ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release ground-level χ/Q without deposition are multiplied with the LPF. The products (hourly values) are sorted in ascending order and the 95th percentile value is identified.

9. Functional Requirements

This chapter lists the functional requirements established for the POSTMAX V2.0 program.

1. POSTMAX correctly reads the MACCS output file.
    - When the MACCS ground-level χ/Q without deposition values decrease with distance and the site boundary distance matches a MACCS radial grid location, POSTMAX reports the MACCS values for ground-level χ/Q without deposition.
    - When the MACCS ground-level time-integrated air concentration values decrease with distance and the site boundary distance matches a MACCS radial grid location, POSTMAX reports the MACCS values for ground-level time-integrated air concentration.
    - When the quotient of the MACCS ground-level χ/Q without deposition and crosswind standard deviation (σ_y) values decreases with distance and the site boundary distance matches a MACCS radial grid location, POSTMAX reports the puff release ground-level χ/Q without deposition calculated from the MACCS ground-level χ/Q without deposition and crosswind standard deviation (σ_y) values.
2. POSTMAX correctly interpolates between the MACCS results.
    - When the MACCS ground-level χ/Q without deposition values decrease with distance and the site boundary distance is between two MACCS radial grid locations, POSTMAX interpolates between the MACCS values for ground-level χ/Q without deposition.
    - When the MACCS ground-level time-integrated air concentration values decrease with distance and the site boundary distance is between two MACCS radial grid locations, POSTMAX interpolates between the MACCS values for ground-level time-integrated air concentration.
    - When the quotient of the MACCS ground-level χ/Q without deposition and crosswind standard deviation (σ_y) values decreases with distance and the site boundary distance is between two MACCS radial grid locations, POSTMAX interpolates the MACCS ground-level χ/Q without deposition and crosswind standard deviation (σ_y) values and reports the resultant puff release ground-level χ/Q without deposition value.
3. POSTMAX correctly calculates the MEOI values.
    - If the site boundary ground-level χ/Q without deposition is not greater than the ground-level χ/Q without deposition at all MACCS distances greater than the site boundary, POSTMAX reports the maximum ground-level χ/Q without deposition past the site boundary.
    - If the site boundary ground-level time-integrated air concentration is not greater than the ground-level time-integrated air concentration at all

MACCS distances greater than the site boundary, POSTMAX reports the maximum ground-level time-integrated air concentration past the site boundary.

- If the site boundary puff release ground-level χ/Q without deposition is not greater than the puff release ground-level χ/Q without deposition at all MACCS distances greater than the site boundary, POSTMAX reports the maximum puff release ground-level χ/Q without deposition past the site boundary.

4. POSTMAX reports when the MEOI value (ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release χ/Q without deposition) is from the last MACCS radial grid. For this case, POSTMAX generates a warning file with a message for each iteration and consequence where the MEOI value is in the last MACCS radial grid.
5. POSTMAX correctly generates the single-year CDFs from a MACCS2 output file.
    - POSTMAX sorts the hourly ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the hourly ground-level time-integrated air concentration values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the puff release ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
6. POSTMAX correctly generates the multi-year CDFs from a MACCS2 output file and a single-year POSTMAX CDF file.
    - POSTMAX sorts the hourly ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the hourly ground-level time-integrated air concentration values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the puff release ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
7. POSTMAX correctly generates the multi-year CDFs from a MACCS2 output file and a multi-year POSTMAX CDF file.
    - POSTMAX sorts the hourly ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.

- POSTMAX sorts the hourly ground-level time-integrated air concentration values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
- POSTMAX sorts the puff release ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.

8. POSTMAX correctly generates the multi-year CDFs from two or more single-year POSTMAX cdo files.
    - POSTMAX sorts the hourly ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the hourly ground-level time-integrated air concentration values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
    - POSTMAX sorts the puff release ground-level χ/Q without deposition values in increasing order; the meteorological file number, day, hour, and MEOI distance associated with each value must be sorted also.
9. For each directory used by POSTMAX for output, POSTMAX will create and update a summary file listing the 95^{th} percentile results for each cdo file written (by POSTMAX) to the directory.
10. When POSTMAX regenerates a cdo file, i.e., overwrites a previously generated cdo file, then the previous 95^{th} percentile results must be erased from the summary file.
11. POSTMAX correctly reads the LPF input file (when specified).
    - When the wind speed for the trial has a matching wind speed in the LPF table, the LPF for this trial will be the LPF table entry matching the wind speed and direction of this trial.
    - The ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release ground-level χ/Q without deposition results for this trial are multiplied with this LPF.
12. POSTMAX correctly selects the LPF value from the input file (when specified).
    - When the wind speed for the trial does not have a matching wind speed in the LPF table, the LPF for this trial will be the maximum of the 2 LPFs for the trial wind direction and the wind speeds below and above the trial wind speed.
    - The ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release ground-level χ/Q without deposition results for this trial are multiplied with this LPF.
13. POSTMAX identifies the input files used. POSTMAX documents within the cdo and csv output files:

- the POSTMAX command(s) that generated the file,
- the time stamp on the POSTMAX executable file, and
- the time stamp on each input file.

14. POSTMAX write the meteorological data to a file.
    - The meteorological data is written as comma separated values (csv file type) to assist data imports by Excel.
    - The file is created in the POSTMAX output directory.
    - If the meteorological data output file already exists, POSTMAX will overwrite the file (without a message or query to the user).
15. POSTMAX generates a file of the LPF table in the POSTMAX output directory.
    - The file is created in the POSTMAX output directory.
    - If the LPF output file already exists, POSTMAX will overwrite the file (without a message or query to the user).
16. If POSTMAX is executed with option 1 or 2, and the POSTMAX single-year cdo output file already exists, then POSTMAX should query whether the user wishes to continue (and overwrite the POSTMAX output files).
17. If POSTMAX is executed with option 3, and the POSTMAX multi-year cdo output file already exists, then POSTMAX should query whether the user wishes to continue (and overwrite the POSTMAX output files).

10. Interface Requirements

This chapter lists the interface requirements established for the POSTMAX V2.0 program.

1. Command Line Errors
    a. If POSTMAX is executed with option 1 and the number of files specified is less than 2, then POSTMAX should terminate with an error message to the terminal.
    b. If POSTMAX is executed with option 1 and the number of files specified is greater than 3, then POSTMAX should terminate with an error message to the terminal.
    c. If POSTMAX is executed with option 2 and the number of files specified is less than 3, then POSTMAX should terminate with an error message to the terminal.
    d. If POSTMAX is executed with option 2 and the number of files specified is greater than 4, then POSTMAX should terminate with an error message to the terminal.

e. If POSTMAX is executed with option 3 and the number of files specified is less than 3, then POSTMAX should terminate with an error message to the terminal.

f. If POSTMAX is executed with option 3 and the number of files specified is greater than 7, then POSTMAX should terminate with an error message to the terminal.

g. If the first POSTMAX command line argument (POSTMAX option) is nonnumeric, then POSTMAX should terminate with an error message to the terminal.

h. If the first POSTMAX command line argument (POSTMAX option) is numeric but not 1, 2, or 3; then POSTMAX should terminate with an error message to the terminal.

i. If POSTMAX is executed with option 1, and the filenames (POSTMAX command line arguments) for the single-year CDO output file and the multi-year CDO output file are identical, then POSTMAX should terminate with an error message to the terminal.

j. If POSTMAX is executed with option 2, and the filenames (POSTMAX command line arguments) for the single-year CDO output file and the multi-year CDO output file are identical, then POSTMAX should terminate with an error message to the terminal.

2. Missing Input File Errors

a. If POSTMAX is executed with option 1 and the POSTMAX input (MACCS output) file does not exist, then POSTMAX should terminate with an error message to the terminal.

b. If POSTMAX is executed with option 2 and the POSTMAX input (MACCS output) file does not exist, then POSTMAX should terminate with an error message to the terminal.

c. If POSTMAX is executed with option 1 and the boundary distance file does not exist, then POSTMAX should terminate with an error message to the terminal.

d. If POSTMAX is executed with option 2 and the boundary distance file does not exist, then POSTMAX should terminate with an error message to the terminal.

e. If POSTMAX is executed with option 1 and the MACCS meteorological data file (named in the POSTMAX input/MACCS output file) does not exist, then POSTMAX should terminate with an error message to the terminal.

f. If POSTMAX is executed with option 2 and the MACCS meteorological data file (named in the POSTMAX input/MACCS output file) does not exist, then POSTMAX should terminate with an error message to the terminal.

g. If POSTMAX is executed with option 1 and the POSTMAX aggregate multi-year CDO file given in the command line does not exist, then POSTMAX should terminate with an error message to the terminal.

h. If POSTMAX is executed with option 2 and the POSTMAX aggregate multi-year CDO file given in the command line does not exist, then POSTMAX should terminate with an error message to the terminal.

i. If POSTMAX is executed with option 2 and the LPF table file does not exist, then POSTMAX should terminate with an error message to the terminal.

j. If POSTMAX is executed with option 3 and a POSTMAX CDO files for input does not exist, then POSTMAX should terminate with an error message to the terminal.

3. MACCS Output File Errors

a. If POSTMAX is executed with option 1, and MACCS output file does not contain the keyword OCIDEBUG001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

b. If POSTMAX is executed with option 2, and MACCS output file does not contain the keyword OCIDEBUG001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

c. If POSTMAX is executed with option 1, and MACCS output file was not generated with OCIDEBUG001 = 2, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

d. If POSTMAX is executed with option 2, and MACCS output file was not generated with OCIDEBUG001 = 2, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

e. If POSTMAX is executed with option 1, and MACCS output file does not contain the keyword M1METCOD001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

f. If POSTMAX is executed with option 2, and MACCS output file does not contain the keyword M1METCOD001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

g. If POSTMAX is executed with option 1, and MACCS output file was not generated with M1METCOD001 = 5, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

h. If POSTMAX is executed with option 2, and MACCS output file was not generated with M1METCOD001 = 5, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

i. If POSTMAX is executed with option 1, and MACCS output file does not contain the keyword GENUMRAD001, then POSTMAX should terminate

with an error message to the terminal and POSTMAX output (cdo and csv) files.

j. If POSTMAX is executed with option 2, and MACCS output file does not contain the keyword GENUMRAD001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

k. If POSTMAX is executed with option 1, and MACCS output file does not contain the keyword M4NSMPLS001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

l. If POSTMAX is executed with option 2, and MACCS output file does not contain the keyword M4NSMPLS001, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

m. If POSTMAX is executed with option 1, and the minimum distance in the boundary distance file is less than the first spatial grid in the POSTMAX input/MACCS output file, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

n. If POSTMAX is executed with option 2, and the minimum distance in the boundary distance file is less than the first spatial grid in the POSTMAX input/MACCS output file, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

o. If POSTMAX is executed with option 1, and the maximum distance in the boundary distance file is greater than the last spatial grid in the POSTMAX input/MACCS output file, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

p. If POSTMAX is executed with option 2, and the maximum distance in the boundary distance file is greater than the last spatial grid in the POSTMAX input/MACCS output file, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

4. Boundary Distance File Errors

a. If POSTMAX is executed with option 1 and the boundary distance file is less than 2 lines (for the title), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

b. If POSTMAX is executed with option 2 and the boundary distance file is less than 2 lines (for the title), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

c. If POSTMAX is executed with option 1 and the boundary distance file is less than 18 lines (2 for the title and 16 for the distances in each compass direction), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

d. If POSTMAX is executed with option 2 and the boundary distance file is less than 18 lines (2 for the title and 16 for the distances in each compass

direction), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

e. If POSTMAX is executed with option 1 and the boundary distance file contains a zero distance, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

f. If POSTMAX is executed with option 2 and the boundary distance file contains a zero distance, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

g. If POSTMAX is executed with option 1 and the boundary distance file contains a negative distance, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

h. If POSTMAX is executed with option 2 and the boundary distance file contains a negative distance, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

5. LPF Table File Errors

a. If POSTMAX is executed with option 2 and the LPF table file is less than 2 lines (for the title), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

b. If POSTMAX is executed with option 2 and the LPF table file is less than 3 lines (2 for the title and 1 for the LPF table), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

c. If POSTMAX is executed with option 2 and the LPF table file begins with a wind speed greater than 0.5 m/s, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

d. If POSTMAX is executed with option 2 and the LPF table file begins with a zero wind speed, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

e. If POSTMAX is executed with option 2 and the LPF table file begins with a negative wind speed, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

f. If POSTMAX is executed with option 2 and the LPF table file contains a wind speed less than the previous wind speed, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

g. If POSTMAX is executed with option 2 and the LPF table file contains a zero LPF, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

h. If POSTMAX is executed with option 2 and the LPF table file contains a negative LPF, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo, csv, and lpf) files.

6. CDO Input Files
   a. If POSTMAX is executed with option 3 and the POSTMAX single-year CDO files were not generated with the same POSTMAX option (i.e., all with 1 or all with 2), then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.
   b. If POSTMAX is executed with option 3 and a POSTMAX multi-year CDO file is specified for input, then POSTMAX should terminate with an error message to the terminal and POSTMAX output (cdo and csv) files.

11. Test Cases

Test cases were developed to verify each functional requirement in Chapter 9 and each interface requirement in Chapter 10. The test cases for the functional requirements (Chapter 9) are described in Appendix A. For the interface requirements, one test case was created for each requirement; Chapter 10 is considered sufficient explanation.

Directory "Test" contains the input files necessary for the test cases. Directory "Test" does not contain the output files for the test case to execute the test cases after installation. Directory "Benchmark" contains the test case output files, in addition to the input files.

Appendix B is the batch file (Test.bat) for executing all test cases. Test.bat includes test cases to demonstrate that the user is prompted in the event that the cdo output file already exists. To complete execution of the test cases, the user must respond to these questions.

POSTMAX V2.0 has options -1, -2, and -3 in addition to options 1 ,2 and 3. The only difference between the positive and negative option values is the trace file generated with the negative values. Since the trace file is intended to be a debugging tool, only the positive option values are used in the test cases.

After the test cases are run, the output files can be verified with program Compare. Compare uses filelist.txt to identify the files in directory Test to compare against files in directory Benchmark. If Compare runs to "Normal completion" then the two sets are identical (except for some file time-stamps). Compare writes its' output to the screen and file "Compare.txt".

12. Program Structure

The main steps performed in POSTMAX V2.0 are shown in Figure 5.

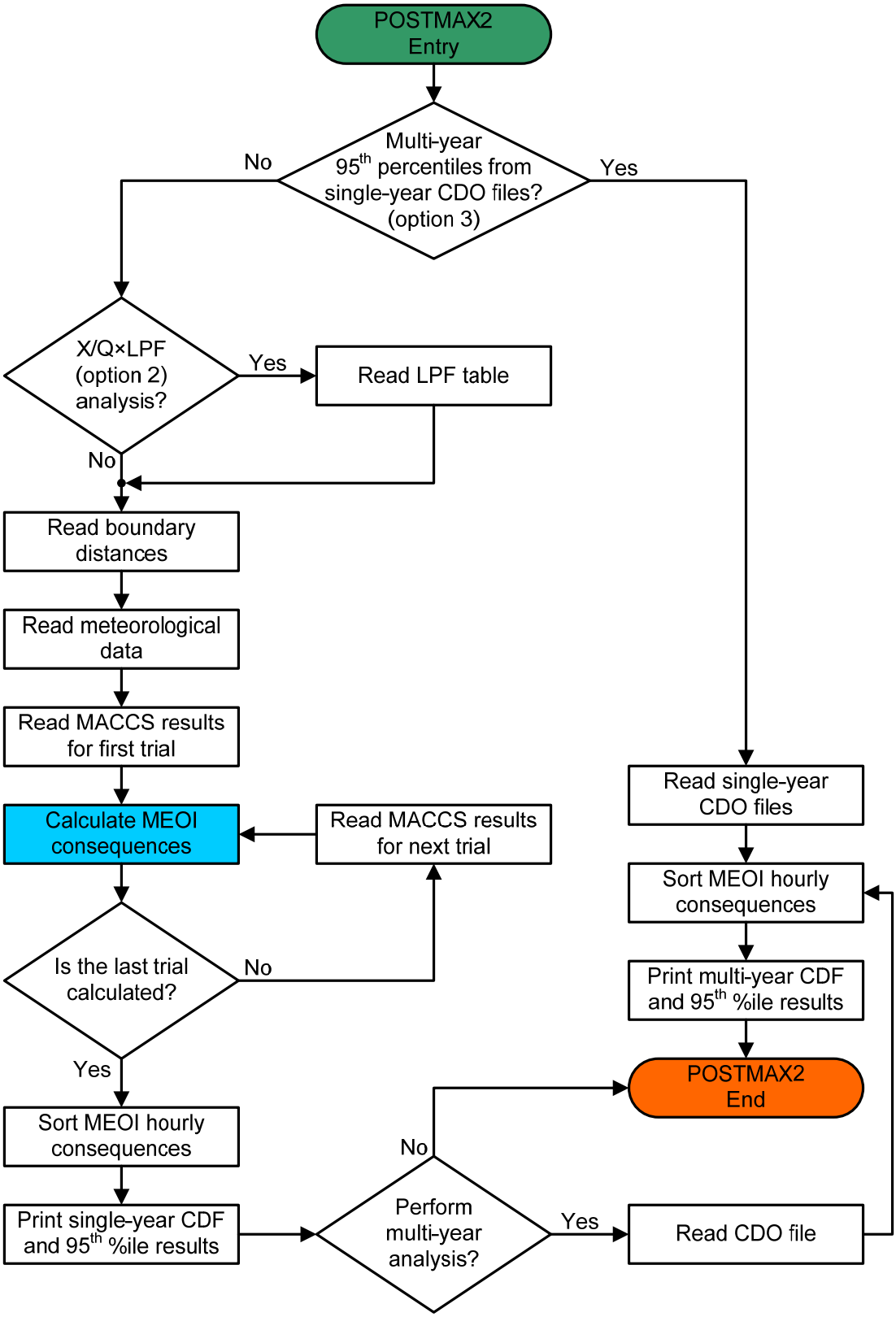

Figure 5
POSTMAX V2.0 Flow Diagram

12.1 MEOI Consequences

The block labeled "Calculate MEOI Consequences" in Figure 5 represents most of the computations performed by POSTMAX. Figure 6 and Figure 7 illustrate these computation steps. The reader is directed to Chapter 8, Theory, for further explanations of these computations.

12.2 Sort Routine

The heapsort algorithm is used to sort the hourly results for ground-level χ/Q without deposition, ground-level time-integrated air concentration, and puff release ground-level χ/Q without deposition in ascending order. The sorted values is the cumulative distribution frequency (CDF) for each result.

In POSTMAX V2.0, the ground-level χ/Q without deposition values are not actually sorted (moved within their array); instead a pointer array is sorted. The sorted pointer array sorts the ground-level χ/Q without deposition values, but also points at the corresponding meteorological file number, day, hour, and MEOI distance values. If the pointer array was not used, these 4 values would have to be moved within their respective arrays. Same for the ground-level time-integrated air concentration and puff release ground-level χ/Q without deposition arrays.

The heapsort subroutine within POSTMAX V2.0 is based on Numerical Recipes (2nd Edition, pp. 329) and work by Bruce C. Letellier (5/15/97). As mentioned above, the subroutine was modified to sort a pointer array. In addition, comments were inserted and variable names changed to clarify the operation of the heapsort subroutine. The subroutine comments are based on http://www.cis.upenn.edu/~matuszek/cit594-2006/Lectures/25-heapsort.ppt.

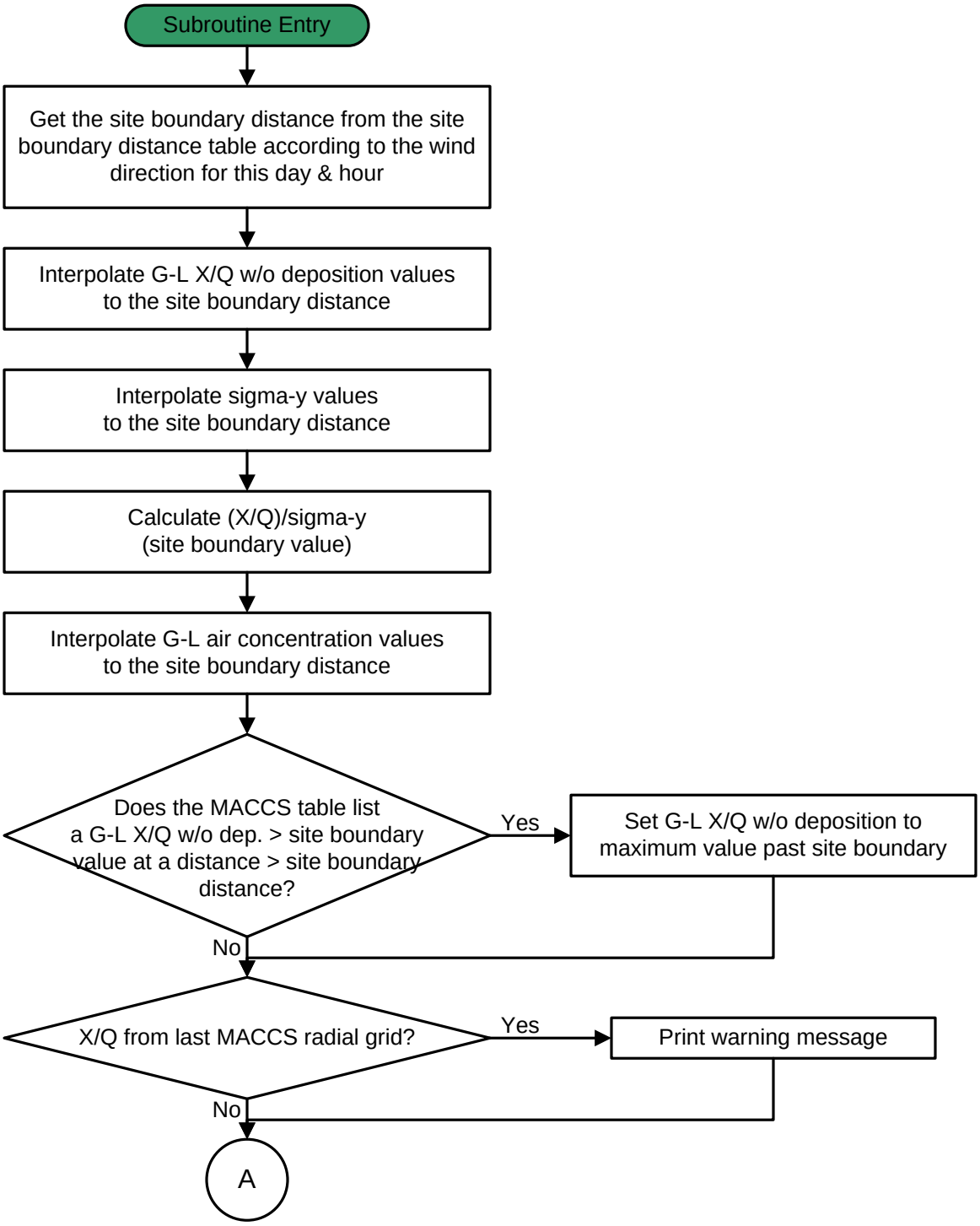

Figure 6
Calculate MEOI Consequences Flow Diagram (Part 1 of 2)

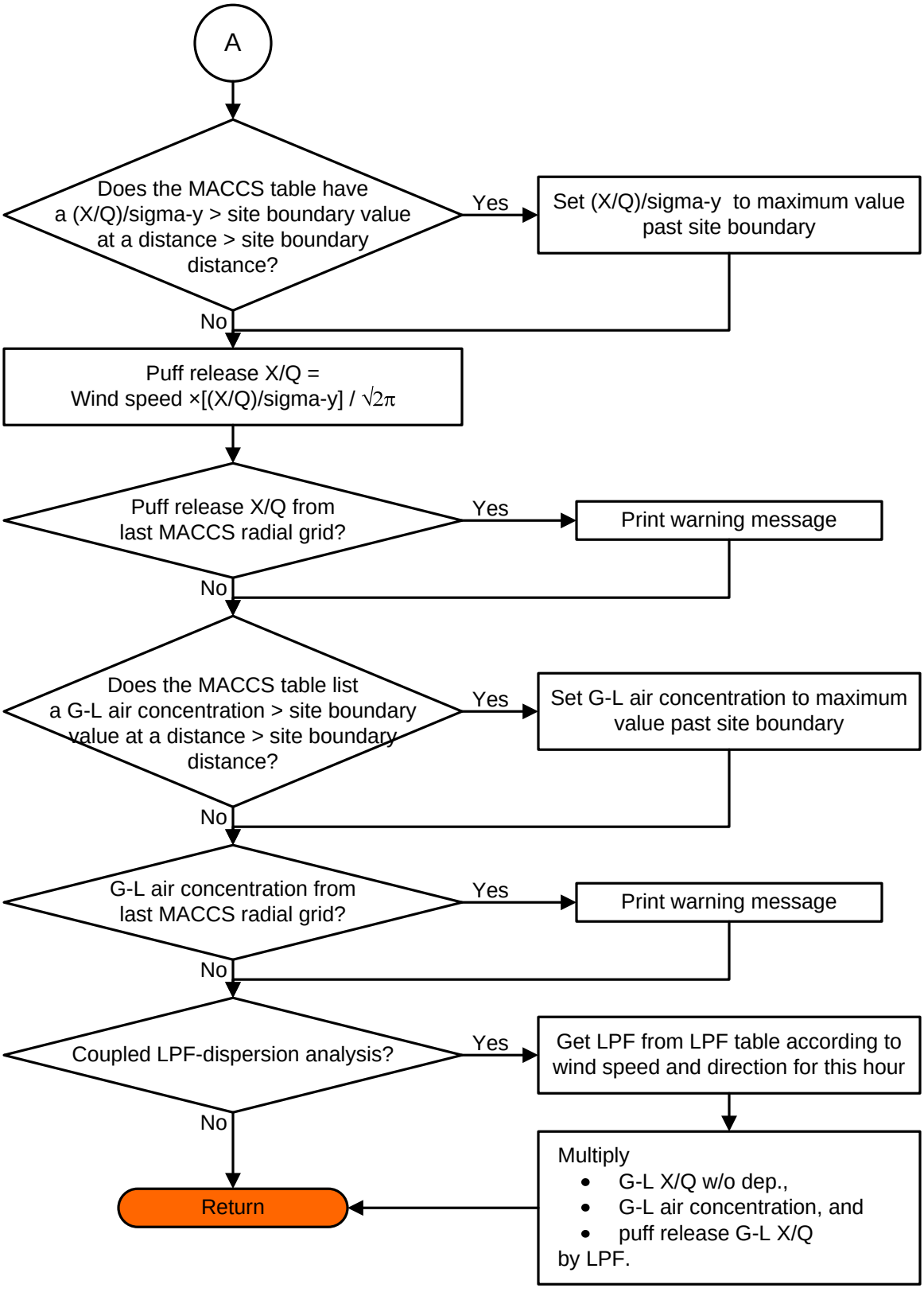

Figure 7
Calculate MEOI Consequences Flow Diagram (Part 2 of 2)

13. Fortran Compiler

POSTMAX V2.0 was compiled with Absoft Pro Fortran for Windows V10.0.

14. Program Analyzers

14.1 Forcheck

The POSTMAX V2.0 Fortran source code was evaluated with the Forcheck program. The Forcheck review did not generate any error (E), overflow (O), or warning (W) messages. There are a few informative (I) messages, but the corresponding code is acceptable, or even necessary.

14.2 plusFORT

The POSTMAX V2.0 Fortran source code was evaluated with the plusFORT program. The messages generated by plusFORT were determined to be acceptable.

15. References

10 CFR 830 — Code of Federal Regulations, Title 10, **Energy**, Part 830, *Nuclear Safety Management*

Dobbins — Richard A. Dobbins, *Atmospheric Motion and Air Pollution*, John Wiley & Sons, Inc., 1979

LA-UR-01-1461 — Software Quality Assurance Verification Report for the POSTMAX Dispersion Analysis Postprocessor, Bruce C. Letellier and Scott G. Ashbaugh, Los Alamos National Laboratory, March 2001

NUREG/CR-4691 — *MELCOR Accident Consequence Code System (MACCS), Model Description*, Sandia National Laboratory, SAND86-1562, February 1990

NUREG/CR-6613 — *Code Manual for MACCS2: Volume 1, User's Guide*, Sandia National Laboratory, SAND97-0594, May 1998

STD-3009 — DOE-STD-3009-94, Change Notice No. 3, *Preparation Guide For U.S Department Of Energy Nonreactor Nuclear Facility Documented Safety Analyses*, U.S. Department of Energy, March 2006

Turner — D. Bruce Turner, *Workbook of Atmospheric Dispersion Estimates*, Second Ed., Trinity Consultants, Inc., 1994

Wark, et. al. — Kenneth Wark, Cecil F. Warker, and Wayne T. Davis, *Air Pollution*, Third Ed., Addison Wesley Longman, Inc., 1998

Functional Requirements
Test Cases

1. POSTMAX correctly reads the MACCS2 ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and crosswind standard deviation (σ_y) at the site boundary distance; and calculates the puff release χ/Q without deposition. (Functional Requirement 1)

   To demonstrate this software Functional Requirement without unnecessary complications, the site boundary distance will be equal to one of the MACCS2 radial grid center distances. In these cases, interpolation of the ground-level χ/Q (without deposition), ground-level time-integrated air concentration,and crosswind standard deviation (σ_y) is not required.

1.1. Input Data

1.1.a. POSTMAX option = 1

1.1.b. MACCS output file – A file containing the required POSTMAX input values was manually created to test this requirement. The MACCS output file contains the following information for each trial.

Trial Line	DISTANCE	GL AIRCON	GL X/Q	PLSIGY
1	1.00E+02	1.60E+01	1.60E+00	1.01E+00
2	2.00E+02	1.50E+01	1.50E+00	1.02E+00
3	3.00E+02	1.40E+01	1.40E+00	1.03E+00
4	4.00E+02	1.30E+01	1.30E+00	1.04E+00
5	5.00E+02	1.20E+01	1.20E+00	1.05E+00
6	6.00E+02	1.10E+01	1.10E+00	1.06E+00
7	7.00E+02	1.00E+01	1.00E+00	1.07E+00
8	8.00E+02	9.00E+00	9.00E-01	1.08E+00
9	9.00E+02	8.00E+00	8.00E-01	1.09E+00
10	1.00E+03	7.00E+00	7.00E-01	1.10E+00
11	1.10E+03	6.00E+00	6.00E-01	1.11E+00
12	1.20E+03	5.00E+00	5.00E-01	1.12E+00
13	1.30E+03	4.00E+00	4.00E-01	1.13E+00
14	1.40E+03	3.00E+00	3.00E-01	1.14E+00
15	1.50E+03	2.00E+00	2.00E-01	1.15E+00
16	1.60E+03	1.00E+00	1.00E-01	1.16E+00

   The GL AIRCON, GL X/Q, and product of GL X/Q and PLSIGY values decrease with distance to demonstrate that POSTMAX correctly identifies the site boundary distance and retrieves the corresponding MACCS2 values.

1.1.c. MACCS meteorological data file – Files containing the meteorological data was manually created to test this requirement. Sixteen files were created to test each wind direction and trial distance. Each file has a different wind speed to demonstrate the puff release χ/Q uses the meteorological data file wind speed.

   In each file, one wind direction and wind speed is entered for all 8760 hours, as follows.

Subcase	Wind Direction	Wind Speed
1	1	1.0
2	2	1.1
3	3	1.2
4	4	1.3
5	5	1.4
6	6	1.5
7	7	1.6
8	8	1.7
9	9	1.8
10	10	1.9
11	11	2.0
12	12	2.1
13	13	2.2
14	14	2.3
15	15	2.4
16	16	2.5

1.1.d. Boundary file – A file containing the site boundary distances was manually created with the following values.

Wind Dir.	Site Boundary Distance
1	100.0
2	200.0
3	300.0
4	400.0
5	500.0
6	600.0
7	700.0
8	800.0
9	900.0
10	1000.0
11	1100.0
12	1200.0
13	1300.0
14	1400.0
15	1500.0
16	1600.0

1.2. Benchmark Values

From the input values, the results should be:

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	1.6000E+00	1.6000E+01	6.3199E-01
2	1.5000E+00	1.5000E+01	6.4535E-01
3	1.4000E+00	1.4000E+01	6.5070E-01
4	1.3000E+00	1.3000E+01	6.4828E-01
5	1.2000E+00	1.2000E+01	6.3831E-01
6	1.1000E+00	1.1000E+01	6.2100E-01
7	1.0000E+00	1.0000E+01	5.9655E-01
8	9.0000E-01	9.0000E+00	5.6517E-01
9	8.0000E-01	8.0000E+00	5.2704E-01
10	7.0000E-01	7.0000E+00	4.8236E-01
11	6.0000E-01	6.0000E+00	4.3129E-01
12	5.0000E-01	5.0000E+00	3.7401E-01
13	4.0000E-01	4.0000E+00	3.1068E-01
14	3.0000E-01	3.0000E+00	2.4147E-01
15	2.0000E-01	2.0000E+00	1.6652E-01
16	1.0000E-01	1.0000E+00	8.5979E-02

1.3. Output Files and Results

The results of this test case are in directory Test-01. The input and output files for each subcase are in a subdirectory. The files for subcase 1 are are subdirectory Dir-01, etc.

1.3.a. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

1.3.b. MACCS.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

1.3.c. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

1.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

2. POSTMAX correctly interpolates the MACCS2 ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and crosswind standard deviation (σ_y) at the site boundary distance; and calculates puff release χ/Q without deposition. (Functional Requirement 2)

   To demonstrate this software Functional Requirement, the site boundary distance will not be equal to any of the MACCS2 radial grid center distances.

2.1. Input Data

2.1.a. POSTMAX option = 1

2.1.b. MACCS output file – A file containing the required POSTMAX input values was manually created to test this requirement. The MACCS output file contains the following information for each trial.

Trial Line	DISTANCE	GL AIRCON	GL X/Q	PLSIGY
1	1.00E+02	1.70E+01	1.70E+00	1.01E+00
2	2.00E+02	1.60E+01	1.60E+00	1.02E+00
3	3.00E+02	1.50E+01	1.50E+00	1.03E+00
4	4.00E+02	1.40E+01	1.40E+00	1.04E+00
5	5.00E+02	1.30E+01	1.30E+00	1.05E+00
6	6.00E+02	1.20E+01	1.20E+00	1.06E+00
7	7.00E+02	1.10E+01	1.10E+00	1.07E+00
8	8.00E+02	1.00E+01	1.00E+00	1.08E+00
9	9.00E+02	9.00E+00	9.00E-01	1.09E+00
10	1.00E+03	8.00E+00	8.00E-01	1.10E+00
11	1.10E+03	7.00E+00	7.00E-01	1.11E+00
12	1.20E+03	6.00E+00	6.00E-01	1.12E+00
13	1.30E+03	5.00E+00	5.00E-01	1.13E+00
14	1.40E+03	4.00E+00	4.00E-01	1.14E+00
15	1.50E+03	3.00E+00	3.00E-01	1.15E+00
16	1.60E+03	2.00E+00	2.00E-01	1.16E+00
17	1.70E+03	1.00E+00	1.00E-01	1.17E+00

The GL AIRCON, GL X/Q, and product of GL X/Q and PLSIGY values decrease with distance to demonstrate that POSTMAX correctly identifies the site boundary distance and retrieves the corresponding MACCS2 values.

2.1.c. MACCS meteorological data file – Files containing the meteorological data was manually created to test this requirement. Sixteen files were created to test each wind direction and trial distance.

In each file, one wind direction and wind speed is entered for all 8760 hours, as follows.

Subcase	Wind Direction	Wind Speed
1	1	1.0
2	2	1.1
3	3	1.2

Subcase	Wind Direction	Wind Speed
4	4	1.3
5	5	1.4
6	6	1.5
7	7	1.6
8	8	1.7
9	9	1.8
10	10	1.9
11	11	2.0
12	12	2.1
13	13	2.2
14	14	2.3
15	15	2.4
16	16	2.5

2.1.d. Boundary file – A file containing the site boundary distances was manually created with the following values.

Wind Dir.	Site Boundary Distance
1	150.0
2	250.0
3	350.0
4	450.0
5	550.0
6	650.0
7	750.0
8	850.0
9	950.0
10	1050.0
11	1150.0
12	1250.0
13	1350.0
14	1450.0
15	1550.0
16	1650.0

2.2. Benchmark Values

From the input values, the results should be:

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	1.6492E+00	1.6492E+01	6.4824E-01
2	1.5492E+00	1.5492E+01	6.6327E-01
3	1.4491E+00	1.4491E+01	6.7029E-01
4	1.3491E+00	1.3491E+01	6.6954E-01

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
5	1.2490E+00	1.2490E+01	6.6123E-01
6	1.1489E+00	1.1489E+01	6.4557E-01
7	1.0488E+00	1.0488E+01	6.2276E-01
8	9.4868E-01	9.4868E+00	5.9300E-01
9	8.4853E-01	8.4853E+00	5.5647E-01
10	7.4833E-01	7.4833E+00	5.1333E-01
11	6.4807E-01	6.4807E+00	4.6376E-01
12	5.4772E-01	5.4772E+00	4.0789E-01
13	4.4721E-01	4.4721E+00	3.4582E-01
14	3.4641E-01	3.4641E+00	2.7760E-01
15	2.4495E-01	2.4495E+00	2.0306E-01
16	1.4142E-01	1.4142E+00	1.2107E-01

2.3. Output Files and Results

The results of this test case are in directory Test-02. The input and output files for each subcase are in a subdirectory. The files for subcase 1 are are subdirectory Dir-01, etc.

2.3.a. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

2.3.b. MACCS.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

2.3.c. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

2.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

3. POSTMAX correctly finds or calculates the peak values for ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition past the site boundary distance. (Functional Requirement 3)

3.1. Input Data

3.1.a. POSTMAX option = 1

3.1.b. MACCS output file – A file containing the required POSTMAX input values was manually created to test this requirement. The MACCS output file contains the following information for each trial.

Trial Line	DISTANCE	GL AIRCON	GL X/Q	PLSIGY
1	1.00E+02	1.00E+00	1.00E-01	1.01E+00
2	2.00E+02	2.00E+00	2.00E-01	1.02E+00
3	3.00E+02	3.00E+00	3.00E-01	1.03E+00
4	4.00E+02	4.00E+00	4.00E-01	1.04E+00
5	5.00E+02	5.00E+00	5.00E-01	1.05E+00
6	6.00E+02	6.00E+00	6.00E-01	1.06E+00
7	7.00E+02	7.00E+00	7.00E-01	1.07E+00
8	8.00E+02	8.00E+00	8.00E-01	1.08E+00
9	9.00E+02	9.00E+00	9.00E-01	1.09E+00
10	1.00E+03	8.00E+00	8.00E-01	1.10E+00
11	1.10E+03	7.00E+00	7.00E-01	1.11E+00
12	1.20E+03	6.00E+00	6.00E-01	1.12E+00
13	1.30E+03	5.00E+00	5.00E-01	1.13E+00
14	1.40E+03	4.00E+00	4.00E-01	1.14E+00
15	1.50E+03	3.00E+00	3.00E-01	1.15E+00
16	1.60E+03	2.00E+00	2.00E-01	1.16E+00
17	1.70E+03	1.00E+00	1.00E-01	1.17E+00

The GL AIRCON, GL X/Q, and product of GL X/Q and PLSIGY values increase with distance to 900 meters, then decrease with distance.

3.1.c. MACCS meteorological data file – Files containing the meteorological data was manually created to test this requirement. Sixteen files were created to test each wind direction and trial distance.

In each file, one wind direction and wind speed is entered for all 8760 hours, as follows.

Subcase	Wind Direction	Wind Speed
1	1	1.0
2	2	1.1
3	3	1.2
4	4	1.3
5	5	1.4
6	6	1.5
7	7	1.6

Subcase	Wind Direction	Wind Speed
8	8	1.7
9	9	1.8
10	10	1.9
11	11	2.0
12	12	2.1
13	13	2.2
14	14	2.3
15	15	2.4
16	16	2.5

3.1.d. Boundary file – A file containing the site boundary distances was manually created with the following values.

Wind Dir.	Site Boundary Distance
1	150.0
2	250.0
3	350.0
4	450.0
5	550.0
6	650.0
7	750.0
8	850.0
9	950.0
10	1050.0
11	1150.0
12	1250.0
13	1350.0
14	1450.0
15	1550.0
16	1650.0

3.2. Benchmark Values

For cases 1 through 8, with distances less than 900 meters, POSTMAX should report the maximum results at 900 meters. For cases 9 through 16, the results should be the same as Test 3.

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	9.0000E-01	9.0000E+00	3.2940E-01
2	9.0000E-01	9.0000E+00	3.6234E-01
3	9.0000E-01	9.0000E+00	3.9528E-01
4	9.0000E-01	9.0000E+00	4.2822E-01
5	9.0000E-01	9.0000E+00	4.6116E-01
6	9.0000E-01	9.0000E+00	4.9410E-01

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
7	9.0000E-01	9.0000E+00	5.2704E-01
8	9.0000E-01	9.0000E+00	5.5998E-01
9	8.4853E-01	8.4853E+00	5.5647E-01
10	7.4833E-01	7.4833E+00	5.1333E-01
11	6.4807E-01	6.4807E+00	4.6376E-01
12	5.4772E-01	5.4772E+00	4.0789E-01
13	4.4721E-01	4.4721E+00	3.4582E-01
14	3.4641E-01	3.4641E+00	2.7760E-01
15	2.4495E-01	2.4495E+00	2.0306E-01
16	1.4142E-01	1.4142E+00	1.2107E-01

3.3. Output Files and Results

The results of this test case are in directory Test-03. The input and output files for each subcase are in a subdirectory. The files for subcase 1 are are subdirectory Dir-01, etc.

3.3.a. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

3.3.b. MACCS.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

3.3.c. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

3.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

4. POSTMAX correctly finds or calculates the peak values for ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition located in the last MACCS2 radial grid. For this case, POSTMAX generates a warning file with a message for each iteration and consequence where the peak value is in the last MACCS radial grid. (Functional Requirement 4)

4.1. Input Data

4.1.a. POSTMAX option = 1

4.1.b. MACCS output file – A file containing the required POSTMAX input values was manually created to test this requirement. The MACCS output file contains the following information for each trial.

Trial Line	DISTANCE	GL AIRCON	GL X/Q	PLSIGY
1	1.00E+02	1.00E+00	1.00E-01	1.01E+00
2	2.00E+02	2.00E+00	2.00E-01	1.02E+00
3	3.00E+02	3.00E+00	3.00E-01	1.03E+00
4	4.00E+02	4.00E+00	4.00E-01	1.04E+00
5	5.00E+02	5.00E+00	5.00E-01	1.05E+00
6	6.00E+02	6.00E+00	6.00E-01	1.06E+00
7	7.00E+02	7.00E+00	7.00E-01	1.07E+00
8	8.00E+02	8.00E+00	8.00E-01	1.08E+00
9	9.00E+02	9.00E+00	9.00E-01	1.09E+00
10	1.00E+03	1.00E+01	1.00E+00	1.10E+00
11	1.10E+03	1.10E+01	1.10E+00	1.11E+00
12	1.20E+03	1.20E+01	1.20E+00	1.12E+00
13	1.30E+03	1.30E+01	1.30E+00	1.13E+00
14	1.40E+03	1.40E+01	1.40E+00	1.14E+00
15	1.50E+03	1.50E+01	1.50E+00	1.15E+00
16	1.60E+03	1.60E+01	1.60E+00	1.16E+00

The GL AIRCON, GL X/Q, and product of GL X/Q and PLSIGY values increase with distance to demonstrate that POSTMAX correctly identifies the last values as the maximum MACCS2 values.

4.1.c. MACCS meteorological data file – Files containing the meteorological data was manually created to test this requirement. Sixteen files were created to test each wind direction and trial distance. Each file has a different wind speed to demonstrate the puff release χ/Q uses the meteorological data file wind speed.

In each file, one wind direction and wind speed is entered for all 8760 hours, as follows.

Subcase	Wind Direction	Wind Speed
1	1	1.0
2	2	1.1
3	3	1.2
4	4	1.3

Subcase	Wind Direction	Wind Speed
5	5	1.4
6	6	1.5
7	7	1.6
8	8	1.7
9	9	1.8
10	10	1.9
11	11	2.0
12	12	2.1
13	13	2.2
14	14	2.3
15	15	2.4
16	16	2.5

4.1.d. Boundary file – A file containing the site boundary distances was manually created with the following values.

Wind Dir.	Site Boundary Distance
1	100.0
2	200.0
3	300.0
4	400.0
5	500.0
6	600.0
7	700.0
8	800.0
9	900.0
10	1000.0
11	1100.0
12	1200.0
13	1300.0
14	1400.0
15	1500.0
16	1600.0

4.2. Benchmark Values

From the input values, the results should be:

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	1.6000E+01	1.6000E+00	5.5027E-01
2	1.6000E+01	1.6000E+00	6.0529E-01
3	1.6000E+01	1.6000E+00	6.6032E-01
4	1.6000E+01	1.6000E+00	7.1534E-01
5	1.6000E+01	1.6000E+00	7.7037E-01

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
6	1.6000E+01	1.6000E+00	8.2540E-01
7	1.6000E+01	1.6000E+00	8.8042E-01
8	1.6000E+01	1.6000E+00	9.3545E-01
9	1.6000E+01	1.6000E+00	9.9048E-01
10	1.6000E+01	1.6000E+00	1.0455E+00
11	1.6000E+01	1.6000E+00	1.1005E+00
12	1.6000E+01	1.6000E+00	1.1556E+00
13	1.6000E+01	1.6000E+00	1.2106E+00
14	1.6000E+01	1.6000E+00	1.2656E+00
15	1.6000E+01	1.6000E+00	1.3206E+00
16	1.6000E+01	1.6000E+00	1.3757E+00

4.3. Output Files and Results

The results of this test case are in directory Test-03. The input and output files for each subcase are in a subdirectory. The files for subcase 1 are are subdirectory Dir-01, etc.

4.3.a. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

4.3.b. MACCS.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

Each value in the CDF and the reported 95th percentile values match the corresponding benchmark value.

4.3.c. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

4.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

4.3.e. MACCS.wrn

This output file contains a warning message for each iteration and consequence where the maximum result is located in the last MACCS2 radial grid. For this test case, a warning message is printed for each iteration and consequence. The file ends with the 95th percentile values.

5. Verify POSTMAX correctly sorts ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results. (Functional Requirement 5)

5.1. Input Data

5.1.a. POSTMAX option = 1

5.1.b. MACCS output file – A file containing the required POSTMAX input values was manually created to test this requirement. The MACCS output file contains the following information for the first trial.

Trial Line	DISTANCE	GL AIRCON	GL X/Q	PLSIGY
1	1.00E+02	3.80E+02	3.80E+01	1.01E+00
2	2.00E+02	3.79E+02	3.79E+01	1.02E+00
3	3.00E+02	3.78E+02	3.78E+01	1.03E+00
4	4.00E+02	3.77E+02	3.77E+01	1.04E+00
5	5.00E+02	3.76E+02	3.76E+01	1.05E+00
6	6.00E+02	3.75E+02	3.75E+01	1.06E+00
7	7.00E+02	3.74E+02	3.74E+01	1.07E+00
8	8.00E+02	3.73E+02	3.73E+01	1.08E+00
9	9.00E+02	3.72E+02	3.72E+01	1.09E+00
10	1.00E+03	3.71E+02	3.71E+01	1.10E+00
11	1.10E+03	3.70E+02	3.70E+01	1.11E+00
12	1.20E+03	3.69E+02	3.69E+01	1.12E+00
13	1.30E+03	3.68E+02	3.68E+01	1.13E+00
14	1.40E+03	3.67E+02	3.67E+01	1.14E+00
15	1.50E+03	3.66E+02	3.66E+01	1.15E+00
16	1.60E+03	3.65E+02	3.65E+01	1.16E+00

In each subsequent trial, the GL AIRCON value (on each line) decreases by 1, and the GL X/Q value decreases by 0.1. For a given distance, the trial results are in decreasing order. The last trial (number 365) is identical to the data in Test 1.

5.1.c. MACCS meteorological data file – Files containing the meteorological data was manually created to test this requirement. Sixteen files were created to test each wind direction and trial distance. Each file has a different wind speed to demonstrate the puff release χ/Q uses the meteorological data file wind speed.

In each file, one wind direction and wind speed is entered for all 8760 hours, as follows.

Subcase	Wind Direction	Wind Speed
1	1	1.0
2	2	1.1
3	3	1.2
4	4	1.3
5	5	1.4
6	6	1.5

Subcase	Wind Direction	Wind Speed
7	7	1.6
8	8	1.7
9	9	1.8
10	10	1.9
11	11	2.0
12	12	2.1
13	13	2.2
14	14	2.3
15	15	2.4
16	16	2.5

5.1.d. Boundary file – A file containing the site boundary distances was manually created with the following values.

Wind Dir.	Site Boundary Distance
1	100.0
2	200.0
3	300.0
4	400.0
5	500.0
6	600.0
7	700.0
8	800.0
9	900.0
10	1000.0
11	1100.0
12	1200.0
13	1300.0
14	1400.0
15	1500.0
16	1600.0

5.2. Benchmark Values

5.2.a. Sort Order

The POSTMAX results should be sorted from day 365 to day 1.

5.2.b. CDF First Entry

Since the CDF has 365 entries, the normalized probability for the first entry is 1/365 ≈ 0.00274. The results for the first entry should be identical to the Test 1 benchmark values.

5.2.c. CDF Last Entry

The results for the last (normalized probability = 1.00000) entry should be:

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	3.8000E+01	3.8000E+02	1.5010E+01
2	3.7900E+01	3.7900E+02	1.6306E+01
3	3.7800E+01	3.7800E+02	1.7569E+01
4	3.7700E+01	3.7700E+02	1.8800E+01
5	3.7600E+01	3.7600E+02	2.0000E+01
6	3.7500E+01	3.7500E+02	2.1170E+01
7	3.7400E+01	3.7400E+02	2.2311E+01
8	3.7300E+01	3.7300E+02	2.3423E+01
9	3.7200E+01	3.7200E+02	2.4508E+01
10	3.7100E+01	3.7100E+02	2.5565E+01
11	3.7000E+01	3.7000E+02	2.6596E+01
12	3.6900E+01	3.6900E+02	2.7602E+01
13	3.6800E+01	3.6800E+02	2.8583E+01
14	3.6700E+01	3.6700E+02	2.9539E+01
15	3.6600E+01	3.6600E+02	3.0472E+01
16	3.6500E+01	3.6500E+02	3.1382E+01

5.2.d. 95^{th} Percentile Results

The CDF will not have an exact entry for the 95^{th} percentile (that is 0.95×365 is a non-integer value, 346.75). The CDF results with a normalized probability of 0.95068 (from 347/365) should be reported as the 95^{th} percentile values. Since the trials in MACCS.OUT are in increasing order, the 95^{th} percentile results correspond to trial 19. The 95^{th} percentile results are:

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
1	3.6200E+01	3.6200E+02	1.4299E+01
2	3.6100E+01	3.6100E+02	1.5531E+01
3	3.6000E+01	3.6000E+02	1.6732E+01
4	3.5900E+01	3.5900E+02	1.7903E+01
5	3.5800E+01	3.5800E+02	1.9043E+01
6	3.5700E+01	3.5700E+02	2.0154E+01
7	3.5600E+01	3.5600E+02	2.1237E+01
8	3.5500E+01	3.5500E+02	2.2293E+01
9	3.5400E+01	3.5400E+02	2.3322E+01
10	3.5300E+01	3.5300E+02	2.4325E+01
11	3.5200E+01	3.5200E+02	2.5302E+01
12	3.5100E+01	3.5100E+02	2.6255E+01
13	3.5000E+01	3.5000E+02	2.7185E+01
14	3.4900E+01	3.4900E+02	2.8090E+01
15	3.4800E+01	3.4800E+02	2.8974E+01

Subcase	Ground-Level χ/Q Without Deposition	Ground-Level Time-Integrated Air Concentration	Puff Release Ground-Level χ/Q Without Deposition
16	3.4700E+01	3.4700E+02	2.9835E+01

5.3. Output Files and Results

The results of this test case are in directory Test-05. The input and output files for each subcase are in a subdirectory. The files for subcase 1 are are subdirectory Dir-01, etc.

5.3.a. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The CDF values at probability 0.00274 are equal to the Test 1 benchmark values. The CDF values at probability 1.0 are equal to the benchmark values listed above. The CDF values at probability 0.95068 are equal to the benchmark values listed above.

The 95th percentile summary matches the CDF and benchmark values.

5.3.b. MACCS.csv

The contents of this file are identical to the cdo file.

5.3.c. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

5.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

6. Verify the POSTMAX CDF generated from real world data. (Functional Requirements 1-5)

   This was Test 3.1 in the original PostMax verification report (LA-UR-01-1461).

6.1. Input Data

6.1.a. POSTMAX option = 1

6.1.b. MACCS output file – The MACCS output file (ATMOS1.OUT) was retrieved from TEST3.1 directory on the CD-ROM attached to LA-UR-01-1461. The meteorological data filename was modified to include the directory name (Test-06).

6.1.c. MACCS meteorological data file – The MACCS meteorological data file (TA6_91.INP) was retrieved from TEST3.1 directory on the CD-ROM attached to LA-UR-01-1461.

6.1.d. Boundary file – The boundary file (CNTLBND.BND) was retrieved from TEST3.1 directory on the CD-ROM attached to LA-UR-01-1461.

6.2. Benchmark Values

The input files were imported into Excel file "Benchmark Calc.xls"; see sheets ATMOS1.OUT, TA6_91.INP, and CNTLBND.BND. Excel was used to calculate the ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results. The primary calculations were performed in sheets Calc_Chi-over-Q, Calc_GL_AIRCON, and Calc_Puff_Rel. Sheet Unsorted_List summarizes the calculations (in trial order). The *values* in Unsorted_List were copied to sheet Sorted_Values, and the ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were individually sorted.

6.3. Output Files and Results

The results of this test case are in directory Test-06.

6.3.a. ATMOS1.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

The 95^{th} percentile summary matches the corresponding (0.95068 probability) CDF values.

6.3.b. ATMOS1.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The csv file and the CDF from "Benchmark Calc.xls" were loaded into an Excel file (V&V.xls). The ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were compared in sheets

Chi-over-Q, AIRCON, and Puff_Rel. Each CDF was identical with the benchamrk values. The meteorological data day and hour, and MEOI distance were also compared to verify these POSTMAX results.

The 95th percentile summary matches the corresponding (0.95068 probability) CDF values.

6.3.c. TA6_91.csv

This output file echoes the meteorological data which was input to POSTMAX.

6.3.d. POSTMAX_Summary.txt

The 95th percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

7. Additional verification of a POSTMAX CDF generated from real world data. (Software Functional Requirements 1-5)

   This was Test 3.2 in the original PostMax verification report (LA-UR-01-1461).

7.1. Input Data

7.1.a. POSTMAX option = 1

7.1.b. MACCS output file – The MACCS output file (ATMOS1.OUT) was retrieved from TEST3.2 directory on the CD-ROM attached to LA-UR-01-1461. The meteorological data filename was modified to include the directory name (Test-07).

7.1.c. MACCS meteorological data file – The MACCS meteorological data file (TA6_94.INP) was retrieved from TEST3.2 directory on the CD-ROM attached to LA-UR-01-1461.

7.1.d. Boundary file – The boundary file (CNTLBND.BND) was retrieved from TEST3.2 directory on the CD-ROM attached to LA-UR-01-1461.

7.2. Benchmark Values

The input files were imported into Excel file "Benchmark Calc.xls"; see sheets ATMOS1.OUT, TA6_94.INP, and CNTLBND.BND. Excel was used to calculate the ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results. The primary calculations were performed in sheets Calc_Chi-over-Q, Calc_GL_AIRCON, and Calc_Puff_Rel. Sheet Unsorted_List summarizes the calculations (in trial order). The *values* in Unsorted_List were copied to sheet Sorted_Values, and the ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were individually sorted.

7.3. Output Files and Results

The results of this test case are in directory Test-07.

7.3.a. ATMOS1.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

The 95^{th} percentile summary matches the corresponding (0.95 probability) CDF values.

7.3.b. ATMOS1.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The csv file and the CDF from "Benchmark Calc.xls" were loaded into an Excel file (V&V.xls). The ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were compared in sheets Chi-over-Q, AIRCON, and Puff_Rel. Each CDF was identical with the benchmark

values. The meteorological data day and hour, and MEOI distance were also compared to verify these POSTMAX results.

The 95th percentile summary matches the corresponding (0.95 probability) CDF values.

7.3.c. ATMOS1.wrn

The MACCS2 output file contains two trials (505 and 953) in which the maximum ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition past the site boundary are in the last MACCS2 radial grid. These conditions are reported in ATMOS1.wrn.

7.3.d. TA6_91.csv

This output file echoes the meteorological data which was input to POSTMAX.

7.3.e. POSTMAX_Summary.txt

The 95th percentile values for ATMOS1.cdo and Cumulatv.cdo listed in this file match the corresponding benchmark values. (Functional Requirement 9)

8. Verify POSTMAX correctly generates aggregate multi-year ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from a MACCS2 output file and a single-year POSTMAX CDF file. (Functional Requirement 6)

   Verify POSTMAX correctly generates aggregate multi-year ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from a MACCS2 output file and a multi-year POSTMAX CDF file. (Functional Requirement 7)

   This was Test 3.3 in the original PostMax verification report (LA-UR-01-1461).

8.1. Input Data

8.1.a. POSTMAX option = 1

8.1.b. MACCS output files – The MACCS output files (ATMOS1.OUT, ATMOS2.OUT, ATMOS3.OUT, ATMOS4.OUT, ATMOS5.OUT, and ATMOS6.OUT) were retrieved from TEST3.3 directory on the CD-ROM attached to LA-UR-01-1461. The meteorological data filename in each file was modified to include the directory name (Test-08).

8.1.c. MACCS meteorological data file – The MACCS meteorological data files (TA6_91.INP, TA6_92.INP, TA6_93.INP, TA6_94.INP, TA6_95.INP, and TA6_96.INP) was retrieved from TEST3.3 directory on the CD-ROM attached to LA-UR-01-1461.

8.1.d. Boundary file – The boundary file (CNTLBND.BND) was retrieved from TEST3.3 directory on the CD-ROM attached to LA-UR-01-1461.

8.2. Benchmark Values

Since POSTMAX has been verified for generating single-year CDFs, the benchmark CDFs will be assembled from the single-year cdo files.

8.3. Output Files and Results

The results of this test case are in directory Test-08.

8.3.a. Single-year CDF files ATMOS1.cdo, etc.

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

8.3.b. Single-year CDF files ATMOS1.csv, etc.

The single-year csv files were loaded into sheets of an Excel file (V&V.xls). The meteorological data file number were revised to match the MACCS2 output filenames. The six single year CDFs were combined into a single list (sheet Unsorted_List). The *values* in Unsorted_List were copied to sheet Sorted_Values, and the ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were individually sorted.

8.3.c. Cumulatv.cdo

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

8.3.d. Cumulatv.csv

This CDF files was loaded in Excel (V&V.xls; sheet Cumulatv). Excel sheet Check verifies the CDF generated from the six single-year files (sheet Sorted_Values) and by POSTMAX are identical. The meteorological data file no., day, and hour; and MEOI distance were also compared to verify these POSTMAX results.

8.3.e. ATMOS3.wrn, ATMOS4.wrn, ATMOS5.wrn, and ATMOS6.wrn

Several of the MACCS2 output file contain trials in which the maximum ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition past the site boundary are in the last MACCS2 radial grid. These conditions are reported in the warning files.

8.3.f. TA6_91.csv, etc.

These output files echo the meteorological data which was input to POSTMAX.

8.3.g. POSTMAX_Summary.txt

The 95^{th} percentile values for ATMOS1.cdo, etc., and Cumulatv.cdo listed in this file match the corresponding output files. (Functional Requirement 9)

9. Verify POSTMAX correctly generates aggregate multi-year ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from two or more single-year POSTMAX CDF files. (Functional Requirement 8)

9.1. Input Data

9.1.a. POSTMAX option = 3

9.1.b. ATMOS1.cdo, etc., from Test 8

9.2. Benchmark Values

The benchmark CDFs assembled from the single-year cdo files for Test 8 are also used for this test.

9.3. Output Files and Results

The results of this test case are in directory Test-09.

9.3.a. Cumulatv.cdo

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

9.3.b. Cumulatv.csv

This CDF files was loaded in Excel (V&V.xls; sheet Cumulatv). Excel sheet Check verifies the CDF generated from the six single-year files (sheet Sorted_Values) and by POSTMAX are identical. The meteorological data file no., day, and hour; and MEOI distance were also compared to verify these POSTMAX results.

9.3.c. POSTMAX_Summary.txt

The 95^{th} percentile values in this file match the values in Cumulatv.cdo. (Functional Requirement 9)

10. When POSTMAX regenerates a cdo file, i.e., overwrites a previously generated cdo file, then the old CDF results must be expunged from the summary file. (Functional Requirement 10)

    This Functional Requirement is verified by executing POSTMAX with a MACCS output file and a boundary file. Then POSTMAX is executed with the same MACCS output file but a different boundary file. Since the same MACCS output filename is used for both runs, the POSTMAX output files are overwritten and the original results written to POSTMAX_Summary are superseded. Between POSTMAX executions, file POSTMAX_Summary.txt is copied to POSTMAX_Summary_Old.txt to record the file contents before being updated. The batch file also renames the output file MACCS.cdo to MACCS_Old.cdo between POSTMAX executions to run the batch file to completion without any user prompts. (When the POSTMAX output file exists, POSTMAX asks the user whether to replace the file.)

10.1. Input Data

10.1.a. POSTMAX option = 1

10.1.b. MACCS output file – The MACCS output file (MACCS.out) created for Test 2, direction 1 is used for this test. The meteorological data filename was modified to include the directory name (Test-10).

10.1.c. MACCS meteorological data file – The MACCS meteorological data file (Met_Data.inp) created for Test 2, direction 1 is used for this test.

10.1.d. Boundary file #1 – The boundary file created for Test 1, direction 1, renamed Distance1.bnd, is used for the first execution of POSTMAX in this test.

10.1.e. Boundary file #2 – The boundary file created for Test 2, direction 1, renamed Distance2.bnd, is used for the second execution of POSTMAX in this test.

10.2. Output Files and Results

The results of this test case are in directory Test-10.

10.2.a. POSTMAX_Summary.txt

The 95^{th} percentile values listed in this file match the results in MACCS.cdo. The superseded results are not listed.

10.2.b. MACCS.cdo

The results of the second execution of POSTMAX.

10.2.c. POSTMAX_Summary_Old.txt

This is a copy of the POSTMAX_Summary.txt generated by the first execution of POSTMAX. This copy was made only to demonstrate that POSTMAX_Summary.txt had results after the first POSTMAX execution that were removed by the second execution. The 95^{th} percentile values listed in this file match the results in MACCS_Old.cdo.

10.2.d. MACCS_Old.cdo

Generated by the first execution of POSTMAX.

10.2.e. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

11. When POSTMAX uses the LPF table and the wind speed for the trial has a match in the LPF table, POSTMAX uses the LPF multiplier (according to wind speed and direction) on MACCS2 ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition. (Functional Requirement 11)

    When POSTMAX uses the LPF table and the wind speed for the trial does not have an exact match in the LPF table, POSTMAX uses the maximum LPF multiplier (of the 2 LPFs for the wind direction and the wind speeds below and above the trial wind speed) on MACCS2 ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition. (Functional Requirement 12)

11.1. Input Data

11.1.a. POSTMAX option = 2

11.1.b. MACCS output file – The MACCS output file (MACCS.out) created for Test 1, direction 1 is used for this test. The meteorological data filename was modified to include the directory name (Test-11).

11.1.c. MACCS meteorological data file – A file containing the meteorological data was manually created to test this requirement. The file starts with wind direction 1 and a wind speed of 1.0 m/s. On each consecutive day, the wind direction is incremented. Every 16 days, the wind speed is incremented by 0.1 m/s. The meteorological data for each hour within a day are identical.

11.1.d. Boundary file – The boundary file created for Test 1, direction 1 is used for this test.

11.1.e. LPF Table – A file containing the following LPF values was created. (The following table does not follow the file format.) The LPF file must begin with a wind speed greater than zero, but less than or equal to 0.5 m/s. Since the meteorological data provides wind speeds at or above 1 m/s, this line should have no effect on the calculation. The second wind speed in the table is 1.0 m/s, and subsequent lines increment the wind speed by 0.2 m/s. With these wind speeds, the wind speeds in the meteorological data file will be a combination of matching and not matching the LPF table. The LPF value at 1 m/s, North direction is 0.010. The LPF value is incremented by 0.005 for each increment in direction or wind speed to make each LPF unique.

WindSpeed	North	NNE	NE	ENE	East	ESE	SE	SSE
0.5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0	0.010	0.015	0.020	0.025	0.030	0.035	0.040	0.045
1.2	0.090	0.095	0.100	0.105	0.110	0.115	0.120	0.125
1.4	0.170	0.175	0.180	0.185	0.190	0.195	0.200	0.205
1.6	0.250	0.255	0.260	0.265	0.270	0.275	0.280	0.285
1.8	0.330	0.335	0.340	0.345	0.350	0.355	0.360	0.365
2.0	0.410	0.415	0.420	0.425	0.430	0.435	0.440	0.445
2.2	0.490	0.495	0.500	0.505	0.510	0.515	0.520	0.525
2.4	0.570	0.575	0.580	0.585	0.590	0.595	0.600	0.605
2.6	0.650	0.655	0.660	0.665	0.670	0.675	0.680	0.685
2.8	0.730	0.735	0.740	0.745	0.750	0.755	0.760	0.765
3.0	0.810	0.815	0.820	0.825	0.830	0.835	0.840	0.845

WindSpeed	North	NNE	NE	ENE	East	ESE	SE	SSE
3.2	0.890	0.895	0.900	0.905	0.910	0.915	0.920	0.925

WindSpeed	South	SSE	SW	WSW	West	WNE	NW	NNW
0.5	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.0	0.050	0.055	0.060	0.065	0.070	0.075	0.080	0.085
1.2	0.130	0.135	0.140	0.145	0.150	0.155	0.160	0.165
1.4	0.210	0.215	0.220	0.225	0.230	0.235	0.240	0.245
1.6	0.290	0.295	0.300	0.305	0.310	0.315	0.320	0.325
1.8	0.370	0.375	0.380	0.385	0.390	0.395	0.400	0.405
2.0	0.450	0.455	0.460	0.465	0.470	0.475	0.480	0.485
2.2	0.530	0.535	0.540	0.545	0.550	0.555	0.560	0.565
2.4	0.610	0.615	0.620	0.625	0.630	0.635	0.640	0.645
2.6	0.690	0.695	0.700	0.705	0.710	0.715	0.720	0.725
2.8	0.770	0.775	0.780	0.785	0.790	0.795	0.800	0.805
3.0	0.850	0.855	0.860	0.865	0.870	0.875	0.880	0.885
3.2	0.930	0.935	0.940	0.945	0.950	0.955	0.960	0.965

11.2. Benchmark Values

File Test-11.xls includes 3 sheets (Chi-over-Q, AIRCON, and Puff_Rel) that calculate the trial results for LPF times ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition from the input values. These *values* were copied to other sheets (Chi-over-Q_Sorted, AIRCON_Sorted, and Puff_Rel_Sorted) and sorted to calculate the CDFs.

11.3. Output Files and Results

The results of this test case are in directory Test-11.

11.3.a. MACCS.lpf

This output file echoes the LPF table which was input to POSTMAX.

11.3.b. MACCS.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

The 95th percentile summary matches the corresponding (0.95068 probability) CDF values.

11.3.c. MACCS.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The csv file and the CDFs from "Test-11.xls" were loaded into an Excel file (V&V.xls). The ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were compared in sheets Chi-over-Q_Sorted, AIRCON_Sorted, and Puff_Rel_Sorted. Each CDF was identical with the benchmark values. The meteorological data day and hour, and MEOI distance were also compared to verify these POSTMAX results.

The 95^{th} percentile summary matches the corresponding (0.95068 probability) CDF values.

11.3.d. Met_Data.csv

This output file echoes the meteorological data which was input to POSTMAX.

11.3.e. POSTMAX_Summary.txt

The 95^{th} percentile values listed in this file match the corresponding benchmark values. (Functional Requirement 9)

12. Verify POSTMAX correctly generates aggregate multi-year ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from a MACCS2 output file and a single-year POSTMAX CDF file. (Functional Requirement 6)

    Verify POSTMAX correctly generates aggregate multi-year ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from a MACCS2 output file and a multi-year POSTMAX CDF file. (Functional Requirement 7)

12.1. Input Data

12.1.a. POSTMAX option = 2

12.1.b. MACCS output files – The MACCS output files (MACCS_1.OUT, MACCS_2.OUT, MACCS_3.OUT, MACCS_4.OUT, MACCS_5.OUT, and MACCS_6.OUT) are the same as Test 11 except the meteorological data filename is the corresponding test file (Test-11\Met_Data_1.inp).

12.1.c. MACCS meteorological data file – The MACCS meteorological data files (Met_Data_1.INP, Met_Data_2.INP, Met_Data_3.INP, Met_Data_4.INP, Met_Data_5.INP, and Met_Data_6.INP) were created in Excel file Test-11.xls using the random function for wind direction and speed. (Sheet Random_Met_Data.inp shows the random functions used; sheets Met_Data_1.inp, etc., list the randomly generated values.)

12.1.d. Boundary file – The boundary file (Distances.bnd) was copied form Test 11.

12.1.e. LPF Table – The LPF file was copied from Test 11.

12.2. Benchmark Values

Since POSTMAX has been verified for generating single-year CDFs, the benchmark CDFs will be assembled from the single-year cdo files.

12.3. Output Files and Results

The results of this test case are in directory Test-12.

12.3.a. Single-year CDF files MACCS_1.cdo, etc.

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

12.3.b. Single-year CDF files MACCS_1.csv, etc.

The single-year csv files were loaded into sheets of an Excel file (V&V.xls). The six single year CDFs were combined into a single list (sheet Unsorted_List) with the meteorological data file number revised to match the MACCS2 output filenames. The *values* in Unsorted_List were copied to sheet Sorted_Values, and the LPF times ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition results were individually sorted.

12.3.c. Cumulatv.cdo

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

The 95th percentile summary matches the corresponding (0.95023 probability) CDF values.

12.3.d. Cumulatv.csv

The file header documents (1) the POSTMAX command that generated the file, (2) the time stamp on the POSTMAX executable file, and (3) the time stamp on each input file (Functional Requirement 13).

This CDF files was loaded in Excel (V&V.xls; sheet Cumulatv). Excel sheet Check verifies the CDF generated from the six single-year files (sheet Sorted_Values) and by POSTMAX are identical. The meteorological data file no., day, and hour; and MEOI distance were also compared to verify these POSTMAX results.

The 95th percentile summary matches the corresponding (0.95023 probability) CDF values.

12.3.e. Met_Data_1.csv, etc.

These output files echo the meteorological data which was input to POSTMAX.

12.3.f. POSTMAX_Summary.txt

The 95th percentile values for MACCS_1.cdo, etc., and Cumulatv.cdo listed in this file match the corresponding output files. (Functional Requirement 9)

13. Verify POSTMAX correctly generates aggregate multi-year LPF times ground-level χ/Q (without deposition), ground-level time-integrated air concentration, and puff release χ/Q without deposition distributions from two or more single-year POSTMAX CDF files. (Functional Requirement 8)

13.1. Input Data

13.1.a. POSTMAX option = 3

13.1.b. MACCS_1.cdo, etc., from Test 12

13.2. Benchmark Values

The benchmark CDFs assembled from the single-year cdo files for Test 12 are also used for this test.

13.3. Output Files and Results

The results of this test case are in directory Test-13.

13.3.a. Cumulatv.cdo

The CDF in this file and the csv file are identical. The csv file is used to verify the CDF; see below.

13.3.b. Cumulatv.csv

This CDF files was loaded in Excel (V&V.xls; sheet Cumulatv). Excel sheet Check verifies the CDF generated from the six single-year files (sheet Sorted_Values) and by POSTMAX are identical. The meteorological data file no., day, and hour; and MEOI distance were also compared to verify these POSTMAX results.

13.3.c. POSTMAX_Summary.txt

The 95^{th} percentile values in this file match the values in Cumulatv.cdo. (Functional Requirement 9)

Test Batch File

Note: The line numbers on the following pages are not part of the file.

```
:: Run POSTMAX2 Test Cases
::
:: Part 1 - Performance Tests
::

@echo off

::
:: Test 1
::

set dir=Test\01
goto Loop
:Test\01

::
:: Test 2
::

set dir=Test\02
goto Loop
:Test\02

::
:: Test 3
::

set dir=Test\03
goto Loop
:Test\03

::
:: Test 4
::

set dir=Test\04
goto Loop
:Test\04

::
:: Test 5
::

set dir=Test\05
goto Loop
:Test\05

::
:: Test 6
::

set dir=Test\06
goto Clean
:Test\06

echo on
POSTMAX2 1 %dir%\ATMOS1 %dir%\CNTLBND
```

```
@echo off

::
:: Test 7
::

set dir=Test\07
goto Clean
:Test\07

echo on
POSTMAX2 1 %dir%\ATMOS1 %dir%\CNTLBND
@echo off

::
:: Test 8
::

set dir=Test\08
goto Clean
:Test\08

echo on
POSTMAX2 1 %dir%\ATMOS1 %dir%\CNTLBND

copy       %dir%\ATMOS1.cdo %dir%\Cumulatv.cdo

POSTMAX2 1 %dir%\ATMOS2 %dir%\CNTLBND %dir%\Cumulatv
POSTMAX2 1 %dir%\ATMOS3 %dir%\CNTLBND %dir%\Cumulatv
POSTMAX2 1 %dir%\ATMOS4 %dir%\CNTLBND %dir%\Cumulatv
POSTMAX2 1 %dir%\ATMOS5 %dir%\CNTLBND %dir%\Cumulatv
POSTMAX2 1 %dir%\ATMOS6 %dir%\CNTLBND %dir%\Cumulatv
@echo off

::
:: Test 9
::

set dir=Test\09
goto Clean
:Test\09

set s=Test\08

echo on
POSTMAX2 3 %dir%\Cumulatv %s%\ATMOS1 %s%\ATMOS2 %s%\ATMOS3 %s%\ATMOS4 %s%\ATMOS5 %s%\ATMOS6
@echo off

::
:: Test 10
::

set dir=Test\10
goto Clean
:Test\10

echo on
```

```
115   POSTMAX2 1 %dir%\MACCS %dir%\Distance1
116
117   copy %dir%\PostMax_Summary.txt %dir%\PostMax_Summary_Old.txt
118   ren  %dir%\MACCS.cdo            MACCS_Old.cdo
119
120   POSTMAX2 1 %dir%\MACCS %dir%\Distance2
121   @echo off
122
123   ::
124   :: Test 11
125   ::
126
127   set dir=Test\11
128   goto Clean
129   :Test\11
130
131   echo on
132   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS %dir%\Distances
133   @echo off
134
135   ::
136   :: Test 12
137   ::
138
139   set dir=Test\12
140   goto Clean
141   :Test\12
142
143   echo on
144   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_1 %dir%\Distances
145
146   copy       %dir%\MACCS_1.cdo %dir%\Cumulatv.cdo
147
148   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_2 %dir%\Distances %dir%\Cumulatv
149   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_3 %dir%\Distances %dir%\Cumulatv
150   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_4 %dir%\Distances %dir%\Cumulatv
151   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_5 %dir%\Distances %dir%\Cumulatv
152   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS_6 %dir%\Distances %dir%\Cumulatv
153   @echo off
154
155   ::
156   :: Test 13
157   ::
158
159   set dir=Test\13
160   goto Clean
161   :Test\13
162
163   set s=Test\12
164
165   echo on
166   POSTMAX2 3 %dir%\Cumulatv %s%\MACCS_1 %s%\MACCS_2 %s%\MACCS_3 %s%\MACCS_4 %s%\MACCS_5 %s%\MACCS_6
167   @echo off
168
169   ::
170   :: Test 14
171   ::
```

```
172
173   echo.
174   echo.
175   echo ****************************************************************
176   echo ***   The following command line is a test of the warning   ****
177   echo ***   handling requirement (pre-existing output file)       ****
178   echo ****************************************************************
179
180   set dir=Test\06
181
182   echo on
183   POSTMAX2 1 %dir%\ATMOS1 %dir%\CNTLBND
184   @echo off
185
186   ::
187   :: Test 15
188   ::
189
190   echo.
191   echo.
192   echo ****************************************************************
193   echo ***   The following command line is a test of the warning   ****
194   echo ***   handling requirement (pre-existing output file)       ****
195   echo ****************************************************************
196
197   set dir=Test\11
198
199   echo on
200   POSTMAX2 2 %dir%\LPF_Tbl %dir%\MACCS %dir%\Distances
201   @echo off
202
203   ::
204   :: Test 16
205   ::
206
207   echo.
208   echo.
209   echo ****************************************************************
210   echo ***   The following command line is a test of the warning   ****
211   echo ***   handling requirement (pre-existing output file)       ****
212   echo ****************************************************************
213
214   set dir=Test\09
215   set s=Test\08
216
217   echo on
218   POSTMAX2 3 %dir%\Cumulatv %s%\ATMOS1 %s%\ATMOS2 %s%\ATMOS3 %s%\ATMOS4 %s%\ATMOS5 %s%\ATMOS6
219   @echo off
220
221   ::
222   :: Part 2 - Error Handling Tests
223   ::
224
225   ::
226   :: Error Type 1
227   ::
228
```

```
229   set dir=Test\E1
230   goto Clean
231   :Test\E1
232
233   echo on
234   POSTMAX2 1 dum > Test\E1\Case_1a.txt
235   @echo off
236   type Test\E1\Case_1a.txt
237
238   echo on
239   POSTMAX2 1 dum dum dum dum > Test\E1\Case_1b.txt
240   @echo off
241   type Test\E1\Case_1b.txt
242
243   echo on
244   POSTMAX2 2 dum dum > Test\E1\Case_1c.txt
245   @echo off
246   type Test\E1\Case_1c.txt
247
248   echo on
249   POSTMAX2 2 dum dum dum dum dum > Test\E1\Case_1d.txt
250   @echo off
251   type Test\E1\Case_1d.txt
252
253   echo on
254   POSTMAX2 3 dum dum > Test\E1\Case_1e.txt
255   @echo off
256   type Test\E1\Case_1e.txt
257
258   echo on
259   POSTMAX2 3 dum dum dum dum dum dum dum dum > Test\E1\Case_1f.txt
260   @echo off
261   type Test\E1\Case_1f.txt
262
263   echo on
264   POSTMAX2 A > Test\E1\Case_1g.txt
265   @echo off
266   type Test\E1\Case_1g.txt
267
268   echo on
269   POSTMAX2 0 > Test\E1\Case_1h.txt
270   @echo off
271   type Test\E1\Case_1h.txt
272
273   echo on
274   POSTMAX2 1 Test\E1\inp Test\E1\bnd Test\E1\inp > Test\E1\Case_1i.txt
275   @echo off
276   type Test\E1\Case_1i.txt
277
278   echo on
279   POSTMAX2 2 Test\E1\lpf Test\E1\inp Test\E1\bnd Test\E1\inp > Test\E1\Case_1j.txt
280   @echo off
281   type Test\E1\Case_1j.txt
282
283   ::
284   :: Error Type 2
285   ::
```

```
286
287   set dir=Test\E2
288   goto Clean
289   :Test\E2
290
291   echo on
292   POSTMAX2 1 Test\E2\dum Test\E2\bnd > Test\E2\Case_2a.txt
293   @echo off
294   type Test\E2\Case_2a.txt
295
296   echo on
297   POSTMAX2 2 Test\E2\lpf Test\E2\dum Test\E2\bnd > Test\E2\Case_2b.txt
298   @echo off
299   type Test\E2\Case_2b.txt
300
301   echo on
302   POSTMAX2 1 Test\E2\inp Test\E2\dum > Test\E2\Case_2c.txt
303   @echo off
304   type Test\E2\Case_2a.txt
305
306   echo on
307   POSTMAX2 2 Test\E2\lpf Test\E2\inp Test\E2\dum > Test\E2\Case_2d.txt
308   @echo off
309   type Test\E2\Case_2b.txt
310
311   echo on
312   POSTMAX2 1 Test\E2\inp Test\E2\bnd > Test\E2\Case_2e.txt
313   @echo off
314   type Test\E2\Case_2e.txt
315
316   echo on
317   POSTMAX2 2 Test\E2\lpf Test\E2\inp Test\E2\bnd > Test\E2\Case_2f.txt
318   @echo off
319   type Test\E2\Case_2f.txt
320
321   echo on
322   POSTMAX2 1 Test\E2\inp Test\E2\bnd Test\E2\dum > Test\E2\Case_2g.txt
323   @echo off
324   type Test\E2\Case_2g.txt
325
326   echo on
327   POSTMAX2 2 Test\E2\lpf Test\E2\inp Test\E2\bnd Test\E2\dum > Test\E2\Case_2h.txt
328   @echo off
329   type Test\E2\Case_2h.txt
330
331   echo on
332   POSTMAX2 2 Test\E2\dum Test\E2\inp Test\E2\bnd > Test\E2\Case_2i.txt
333   @echo off
334   type Test\E2\Case_2i.txt
335
336   echo on
337   POSTMAX2 3 Test\E2\out Test\E2\dum Test\E2\dum > Test\E2\Case_2j.txt
338   @echo off
339   type Test\E2\Case_2j.txt
340
341   ::
342   :: Error Type 3
```

```
::

set dir=Test\E3
goto Clean
:Test\E3

set file=Case_3a
echo on
POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3b
echo on
POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3c
echo on
POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3d
echo on
POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3e
echo on
POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3f
echo on
POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3g
echo on
POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3h
echo on
POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
@echo off
type Test\E3\%file%.txt

set file=Case_3i
echo on
POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
```

```
400   @echo off
401   type Test\E3\%file%.txt
402
403   set file=Case_3j
404   echo on
405   POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
406   @echo off
407   type Test\E3\%file%.txt
408
409   set file=Case_3k
410   echo on
411   POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
412   @echo off
413   type Test\E3\%file%.txt
414
415   set file=Case_3l
416   echo on
417   POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
418   @echo off
419   type Test\E3\%file%.txt
420
421   set file=Case_3m
422   echo on
423   POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
424   @echo off
425   type Test\E3\%file%.txt
426
427   set file=Case_3n
428   echo on
429   POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
430   @echo off
431   type Test\E3\%file%.txt
432
433   set file=Case_3o
434   echo on
435   POSTMAX2 1 Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
436   @echo off
437   type Test\E3\%file%.txt
438
439   set file=Case_3p
440   echo on
441   POSTMAX2 2 Test\E3\lpf Test\E3\%file% Test\E3\bnd > Test\E3\%file%.txt
442   @echo off
443   type Test\E3\%file%.txt
444
445   ::
446   :: Error Type 4
447   ::
448
449   set dir=Test\E4
450   goto Clean
451   :Test\E4
452
453   set file=Case_4a
454   echo on
455   POSTMAX2 1 Test\E4\%file% Test\E4\one_line > Test\E4\%file%.txt
456   @echo off
```

```
type Test\E4\%file%.txt

set file=Case_4b
echo on
POSTMAX2 2 Test\E4\lpf Test\E4\%file% Test\E4\one_line > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4c
echo on
POSTMAX2 1 Test\E4\%file% Test\E4\ten_lines > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4d
echo on
POSTMAX2 2 Test\E4\lpf Test\E4\%file% Test\E4\ten_lines > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4e
echo on
POSTMAX2 1 Test\E4\%file% Test\E4\zero_val > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4f
echo on
POSTMAX2 2 Test\E4\lpf Test\E4\%file% Test\E4\zero_val > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4g
echo on
POSTMAX2 1 Test\E4\%file% Test\E4\neg_val > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

set file=Case_4h
echo on
POSTMAX2 2 Test\E4\lpf Test\E4\%file% Test\E4\neg_val > Test\E4\%file%.txt
@echo off
type Test\E4\%file%.txt

::
:: Error Type 5
::

set dir=Test\E5
goto Clean
:Test\E5

set file=Case_5a
echo on
POSTMAX2 2 Test\E5\one_line Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
@echo off
type Test\E5\%file%.txt
```

```
514
515   set file=Case_5b
516   echo on
517   POSTMAX2 2 Test\E5\two_lines Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
518   @echo off
519   type Test\E5\%file%.txt
520
521   set file=Case_5c
522   echo on
523   POSTMAX2 2 Test\E5\high_spd Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
524   @echo off
525   type Test\E5\%file%.txt
526
527   set file=Case_5d
528   echo on
529   POSTMAX2 2 Test\E5\zero_spd Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
530   @echo off
531   type Test\E5\%file%.txt
532
533   set file=Case_5e
534   echo on
535   POSTMAX2 2 Test\E5\neg_spd Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
536   @echo off
537   type Test\E5\%file%.txt
538
539   set file=Case_5f
540   echo on
541   POSTMAX2 2 Test\E5\decr_spd Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
542   @echo off
543   type Test\E5\%file%.txt
544
545   set file=Case_5g
546   echo on
547   POSTMAX2 2 Test\E5\zero_lpf Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
548   @echo off
549   type Test\E5\%file%.txt
550
551   set file=Case_5h
552   echo on
553   POSTMAX2 2 Test\E5\neg_lpf Test\E5\%file% Test\E5\bnd > Test\E5\%file%.txt
554   @echo off
555   type Test\E5\%file%.txt
556
557   goto end
558
559   ::-------------------------------------------------------------------------------
560   :Loop
561
562   set case=Dir-01
563   goto Subr
564   :Dir-01
565
566   set case=Dir-02
567   goto Subr
568   :Dir-02
569
570   set case=Dir-03
```

```
571   goto Subr
572   :Dir-03
573
574   set case=Dir-04
575   goto Subr
576   :Dir-04
577
578   set case=Dir-05
579   goto Subr
580   :Dir-05
581
582   set case=Dir-06
583   goto Subr
584   :Dir-06
585
586   set case=Dir-07
587   goto Subr
588   :Dir-07
589
590   set case=Dir-08
591   goto Subr
592   :Dir-08
593
594   set case=Dir-09
595   goto Subr
596   :Dir-09
597
598   set case=Dir-10
599   goto Subr
600   :Dir-10
601
602   set case=Dir-11
603   goto Subr
604   :Dir-11
605
606   set case=Dir-12
607   goto Subr
608   :Dir-12
609
610   set case=Dir-13
611   goto Subr
612   :Dir-13
613
614   set case=Dir-14
615   goto Subr
616   :Dir-14
617
618   set case=Dir-15
619   goto Subr
620   :Dir-15
621
622   set case=Dir-16
623   goto Subr
624   :Dir-16
625
626   goto %dir%
627
```

```
::-------------------------------------------------------------------------------
:Subr

if exist %dir%\%case%\*.cdo  del %dir%\%case%\*.cdo
if exist %dir%\%case%\*.csv  del %dir%\%case%\*.csv
if exist %dir%\%case%\*.tmp  del %dir%\%case%\*.tmp
if exist %dir%\%case%\*.lpf  del %dir%\%case%\*.lpf
if exist %dir%\%case%\*.wrn  del %dir%\%case%\*.wrn

if exist %dir%\%case%\PostMax_Summary.txt  del %dir%\%case%\PostMax_Summary.txt

echo on
POSTMAX2 1 %dir%\%case%\MACCS %dir%\%case%\Distances
@echo off

goto %case%

::-------------------------------------------------------------------------------
:Clean

if exist %dir%\*.cdo  del %dir%\*.cdo
if exist %dir%\*.csv  del %dir%\*.csv
if exist %dir%\*.tmp  del %dir%\*.tmp
if exist %dir%\*.lpf  del %dir%\*.lpf
if exist %dir%\*.wrn  del %dir%\*.wrn

if exist %dir%\PostMax_Summary*.txt  del %dir%\PostMax_Summary*.txt

goto %dir%

::-------------------------------------------------------------------------------
:end
echo End of batch file.
```

POSTMAX V2.0
Source Code

Note: The line numbers on the following pages are not part of the file.

```
!     ------------------------------------------------------------------
!
!     POSTMAX2.for
!
!     ------------------------------------------------------------------
      program POSTMAX2
      include 'POSTMAX2.fh'
      integer n

      call get_cmd_args
      if (calc_type.lt.0) open(trc,file=trcfile)
*-----------------------------------------------------------------------
      if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
        call get_metfile_name
        call open_single_year_output
        if (abs(calc_type).eq.2) call get_lpf_table
        call get_boundary_distances
        call get_met_data
        call get_maccs_parms
        call queue_trial_results
        do iter=1,nmet
          call read_trial_results
          call calc_consequences
        enddo
        close(in)
        call sort_consequences
        call print_consequences(1)
        call open_summary_file
        call print_summary(outfile)
      endif
*-----------------------------------------------------------------------
      if (multiyear) then
        do n=1,ncdofiles
          call get_previous_cdo(n)
        enddo
        call sort_consequences
        call open_multi_year_output
        call print_consequences(2)
        if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
          call print_summary(cdofile(1))
        else
          call open_summary_file
          call print_summary(outfile)
        endif
      endif
*-----------------------------------------------------------------------
      close(summ)
      if (calc_type.lt.0) close(trc)
      write(crt,98000)
      stop
98000 format(/'Normal completion')
      end
*=======================================================================
      block data
      include 'POSTMAX2.fh'
      data key, crt, bnd, cdo, csv, eco,  in, lpf, met, out, summ, sum2,
     1    temp, trc, wrn
```

```
     1   /   5,   6, 101, 102, 103, 104, 105, 106, 107, 108,  109,  110,
     1     111, 112, 113 /
      data pi / 3.14159265358979d+00 /
      data n_oldhdr / 0 /
      data user_abort / '?' /
      end
*=====================================================================
      subroutine get_cmd_args
*----
*---- 1) Verifies the command line has a valid number of arguements
*---- 2) Get the calculation type and filenames from the cmd line
*---- 3) Verifies input number for calculation type is valid
*---- 4) Appends the required extensions to the filenames
*---- 5) Verifies the input files exist
*----
      include 'POSTMAX2.fh'
*---- local variables
      character cline*100
      integer i, iarg, imax, length
      logical ex
*---- external functions
      integer iargc

*---- get calculation type
      iarg = 1
      call getarg(iarg, cline)
      read (cline,*,err=70000) calc_type

*---- check number of command line arguements
      if (abs(calc_type).eq.1) then
        if (iargc().lt.3.or.iargc().gt.4) goto 70000
      elseif (abs(calc_type).eq.2) then
        if (iargc().lt.4.or.iargc().gt.5) goto 70000
      elseif (abs(calc_type).eq.3) then
        if (iargc().lt.4.or.iargc().gt.8) goto 70000
      else
        goto 70000
      endif

*---- get filename for LPF table
      if (abs(calc_type).eq.2) then
        iarg = iarg + 1
        call getarg(iarg, cline)
        length = len_trim(cline)
        lpffile = cline(1:length)//'.txt'
*----   check if LPF table file exists
        inquire (file=lpffile, exist=ex)
        if (.not.ex) then
          write(crt,90040) trim(lpffile)
          stop
        endif
      else
        lpffile = ""
      endif

*---- get first filename and folder name
      iarg = iarg + 1
```

```
115         call getarg(iarg, cline)
116         length = len_trim(cline)
117   *---- if input file is located in a subdirectory, save the directory name
118         if (index(cline,'\').eq.0) then
119           folder = ' '
120         else
121           do i=1,length
122             if (cline(i:i).eq.'\') imax=i
123           enddo
124           folder = cline(1:imax)
125         endif
126         length = len_trim(cline)
127
128         if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
129
130   *-----  append the filename extension to the input file
131           infile  = cline(1:length)//'.out'
132
133   *-----  create the single-year cdo (text), csv (comma separated values),
134   *-----  trace, and LPF echo output filenames
135           outfile = infile(1:index(infile,'.'))//'cdo'
136           csvfile = infile(1:index(infile,'.'))//'csv'
137           trcfile = infile(1:index(infile,'.'))//'trc'
138           lpfecho = infile(1:index(infile,'.'))//'lpf'
139
140   *-----  check if input file exists
141           inquire (file=infile, exist=ex)
142           if (.not.ex) then
143             write(crt,90040) trim(infile)
144             stop
145           endif
146
147   *-----  get filename for site boundary file
148           iarg = iarg + 1
149           call getarg(iarg, cline)
150           length = len_trim(cline)
151           bndfile = cline(1:length)//'.bnd'
152
153   *-----  check if site boundary file exists
154           inquire (file=bndfile, exist=ex)
155           if (.not.ex) then
156             write(crt,90040) trim(bndfile)
157             stop
158           endif
159
160   *-----  get filename for cdo file (if included in command line)
161           iarg = iarg + 1
162           if (iargc().eq.iarg) then
163   *------   a multi-year analysis is requested
164             multiyear=.true.
165             ncdofiles=1
166
167             call getarg(iarg, cline)
168             length = len_trim(cline)
169             cdofile(1)=cline(1:length)//'.cdo'
170
171   *------   verify input multi-year cdo filename is not identical to
```

```
172   *------   output single-year cdo filename
173             if (outfile.eq.cdofile(1)) then
174               write(crt,90050) outfile
175               stop
176             endif
177
178   *------   verify user-specified CDO file exists
179             inquire (file=cdofile(1), exist=ex)
180             if (.not. ex) then
181               write(crt,90040) trim(cdofile(1))
182               stop
183             endif
184
185           else
186             multiyear=.false.
187           endif
188         else
189   *-----  calc_type = 3
190           multiyear=.true.
191
192   *-----  append the filename extension to the output file
193           outfile = cline(1:length)//'.cdo'
194
195   *-----  create the trace filename
196           trcfile = cline(1:length)//'.trc'
197
198           ncdofiles=iargc()-2
199
200           do iarg=3,iargc()
201             call getarg(iarg, cline)
202             length = len_trim(cline)
203             cdofile(iarg-2) = cline(1:length)//'.cdo'
204
205   *-------  check if CDO file exists
206             inquire (file=cdofile(iarg-2), exist=ex)
207             if (.not.ex) then
208               write(crt,90040) trim(cdofile(iarg-2))
209               stop
210             endif
211           enddo
212         endif
213
214         return
215
216   70000 write(crt,90010)
217         stop
218   90010 format(/'Command line must be in the form of:',
219        1  //'POSTMAX2 1 infile bndfile <cdofile>',
220        1   /'POSTMAX2 2 lpffile infile bndfile <cdofile>',
221        1   /'POSTMAX2 3 outfile cdofile1 cdofile2 <cdofile3> <cdofile4>',
222        1    ' <cdofile5> <cdofile6>',
223        2  //'option = 1 for analysis of MACCS results',
224        3   /'       = 2 for analysis of MACCS and LPF results',
225        3   /'       = 3 for merging/sorting previous POSTMAX2 results',
226        6   /'infile.out  = MACCS output file used for POSTMAX2 input',
227        7   /'bndfile.txt = site boundary input file',
228        8   /'cdofile.cdo = optional POSTMAX2 output file; input/output',
```

```
229        9   /'              for the multi-year statistical analysis',
230        4   /'lpffile.txt = LPF table input file',
231        5   /'              (required only when option-num = 2)',
232        6   /'outfile.cdo = POSTMAX2 output file; output',
233        9   /'              for the multi-year statistical analysis',
234        8   /'cdofile#.cdo= POSTMAX2 single-year output file; input',
235        9   /'              for the multi-year statistical analysis')
236     90040 format(
237          1  /'Error: Input file ',a,
238          2  /'       does not exist.',
239          3 //'Execution terminated.')
240     90050 format(
241          1  /'Error: The POSTMAX input/MACCS2 output file and the',
242          2  /'       the CDO input file (from an earlier POSTMAX run)',
243          3  /'       have the same root filename: ',a,'.',
244          4  /'       The root filenames must be different for POSTMAX',
245          5  /'       to write the single-year output of this run and',
246          6  /'       read the output of the previous run.'
247          7 //'Execution terminated.')
248           end
249     *=====================================================================
250           subroutine get_metfile_name
251           include 'POSTMAX2.fh'
252     *---- local variables
253           logical ex
254
255           open(in,file=infile,status='old')
256
257     10000 read(in,'(a)') metfile
258           if (index(metfile,'METEOROLOGY').eq.0) goto 10000
259
260           close(in)
261
262           metfile = metfile(index(metfile,'=')+2:len_trim(metfile))
263
264     *---- check if meteorological file exists
265           inquire (file=metfile, exist=ex)
266           if (.not.ex) then
267             write(crt,90040) trim(metfile)
268             stop
269           endif
270
271           return
272     90040 format(
273          1  /'Error: Meteorology data file ',a,
274          2  /'       does not exist.',
275          3 //'Execution terminated.')
276           end
277     *=====================================================================
278           subroutine open_single_year_output
279           include 'POSTMAX2.fh'
280     *---- local variables
281           logical ex
282           character answer*3, cline*100, Weekday_Label(7)*3
283           integer i, File_Status(13)
284           integer GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
285          &        GMT_Minute, GMT_Second, GMT_Weekday
```

```
286   *---- external functions
287         integer iargc
288
289   *---- text for weekday values from 1 to 7
290         data Weekday_Label / 'Sun','Mon','Tue','Wed','Thu','Fri','Sat'/
291
292   *---- if output file exists, ask user whether it should be overwritten
293         inquire (file=outfile, exist=ex)
294         if (ex) then
295            write(crt,90020) trim(outfile)
296            read(key,*) answer
297            write(crt,*)
298            if (answer.ne.'y'.and.answer.ne.'yes') then
299               write(crt,90030)
300               stop
301            endif
302         endif
303
304         open(out,file=outfile)
305         open(csv,file=csvfile)
306
307         n_newhdr=1
308         newhdr(n_newhdr)='Command Line:'
309         do i=0, iargc()
310           call getarg(i, cline)
311           newhdr(n_newhdr)=trim(newhdr(n_newhdr))//' '//trim(cline)
312         enddo
313         write(crt,*)
314         write(crt,'(a)') trim(newhdr(n_newhdr))
315         write(out,'(a)') trim(newhdr(n_newhdr))
316         write(csv,'(a)') trim(newhdr(n_newhdr))
317
318         n_newhdr=n_newhdr+1
319         call Get_File_Status('POSTMAX2.exe', crt, File_Status,
320        1     GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
321        2     GMT_Minute, GMT_Second, GMT_Weekday)
322          write(newhdr(n_newhdr),92010) 'POSTMAX2.exe',
323        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
324        2     GMT_Hour, GMT_Minute, GMT_Second
325          write(crt,'(a)') trim(newhdr(n_newhdr))
326          write(out,'(a)') trim(newhdr(n_newhdr))
327          write(csv,'(a)') trim(newhdr(n_newhdr))
328
329          if (abs(calc_type).eq.2) then
330            n_newhdr=n_newhdr+1
331            call Get_File_Status(lpffile, crt, File_Status,
332        1       GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
333        2       GMT_Minute, GMT_Second, GMT_Weekday)
334            write(newhdr(n_newhdr),92010) trim(lpffile),
335        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
336        2     GMT_Hour, GMT_Minute, GMT_Second
337            write(crt,'(a)') trim(newhdr(n_newhdr))
338            write(out,'(a)') trim(newhdr(n_newhdr))
339            write(csv,'(a)') trim(newhdr(n_newhdr))
340          endif
341
342          n_newhdr=n_newhdr+1
```

```
343         call Get_File_Status(infile, crt, File_Status,
344        1     GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
345        2     GMT_Minute, GMT_Second, GMT_Weekday)
346         write(newhdr(n_newhdr),92010) trim(infile),
347        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
348        2     GMT_Hour, GMT_Minute, GMT_Second
349         write(crt,'(a)') trim(newhdr(n_newhdr))
350         write(out,'(a)') trim(newhdr(n_newhdr))
351         write(csv,'(a)') trim(newhdr(n_newhdr))
352
353         n_newhdr=n_newhdr+1
354         call Get_File_Status(bndfile, crt, File_Status,
355        1     GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
356        2     GMT_Minute, GMT_Second, GMT_Weekday)
357         write(newhdr(n_newhdr),92010) trim(bndfile),
358        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
359        2     GMT_Hour, GMT_Minute, GMT_Second
360         write(crt,'(a)') trim(newhdr(n_newhdr))
361         write(out,'(a)') trim(newhdr(n_newhdr))
362         write(csv,'(a)') trim(newhdr(n_newhdr))
363
364         if (multiyear) then
365           n_newhdr=n_newhdr+1
366           call Get_File_Status(cdofile(1), crt, File_Status,
367        1       GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
368        2       GMT_Minute, GMT_Second, GMT_Weekday)
369           write(newhdr(n_newhdr),92010) trim(cdofile(1)),
370        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
371        2     GMT_Hour, GMT_Minute, GMT_Second
372           write(crt,'(a)') trim(newhdr(n_newhdr))
373           write(out,'(a)') trim(newhdr(n_newhdr))
374           write(csv,'(a)') trim(newhdr(n_newhdr))
375         endif
376
377         n_newhdr=n_newhdr+1
378         call Get_File_Status(metfile, crt, File_Status,
379        1     GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
380        2     GMT_Minute, GMT_Second, GMT_Weekday)
381         write(newhdr(n_newhdr),92020) 1, trim(metfile),
382        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
383        2     GMT_Hour, GMT_Minute, GMT_Second
384         write(crt,'(a)') trim(newhdr(n_newhdr))
385         write(out,'(a)') trim(newhdr(n_newhdr))
386         write(csv,'(a)') trim(newhdr(n_newhdr))
387
388         write(crt,91020)
389         write(out,91020)
390         write(csv,91020)
391
392         call print_column_headers(out,calc_type,1,1,1,0)
393         call print_column_headers(csv,calc_type,1,1,1,1)
394
395         if (calc_type.lt.0) then
396           do i=1,n_newhdr
397             write(trc,'(a)') trim(newhdr(i))
398           enddo
399           write(trc,91020)
```

```
      endif

      return
90020 format(/'Warning: File ',a,' already exists - overwrite? (y/n) ',
     1  $)
90030 format(/'Execution terminated.')
91020 format(118('-'))
92010 format('   File Info: ',a,' - ',a3,1x,i2,'/',i2.2,'/',i4,
     1       1x,i2,2(':',i2.2),' GMT')
92020 format('   Met. File',i2,': ',a,' - ',a3,1x,i2,'/',i2.2,'/',i4,
     1       1x,i2,2(':',i2.2),' GMT')
      end
*=======================================================================
      subroutine print_column_headers(file,ctype,nyr,col1,verbose,
     1           comma)
      implicit none
*---- input values
      integer file
*----          is the fileio number to write the headers to
      integer ctype
*----            =1 or -1 calculation is for X/Q (and corr. results)
*----            =2 or -2 calculation is for LPF*X/Q (and corr. results)
      integer nyr
*----            =1 results for a single year
*----            >1 results for multiple years
      integer col1
*----            =1 results listed by probability in column 1
*----            =2 results listed by sequence number in column 1
      integer verbose
*----            =1 headers for single-year and multi-year output files
*----            =2 headers for summary file
      integer comma
*----            =0 no commas
*----            =1 comma separated columns
*----
*---- local variables
      integer i
      character label1(9,2)*7, cnsq1(9)*10, cnsq2(9)*10, cnsq3(9)*10,
     1          disthdr(9)*8,methdr(9)*12, methdr2(9)*14, namehdr(9)*48
      data label1 /
     1  '       ',
     2  '       ',
     3  '       ',
     4  'Normal-',
     5  'ized   ',
     6  'Prob-  ',
     7  'ability',
     8  '(-)    ',
     9  '-------',
     1  '       ',
     2  '       ',
     3  '       ',
     4  '       ',
     5  '       ',
     6  'Trial  ',
     7  'Number ',
     8  '(-)    ',
```

```
     9  '-------'/
      data cnsq1 /
     1  '          ',
     2  '          ',
     3  'Ground-   ',
     4  'level    ',
     5  'Chi/Q    ',
     6  'without  ',
     7  'deposition',
     8  '(s/m**3)  ',
     9  '----------'/
      data cnsq2 /
     1  'Puff      ',
     2  'Release   ',
     3  'Ground-   ',
     4  'level    ',
     5  'Chi/Q    ',
     6  'without  ',
     7  'deposition',
     8  '(1/m**3)  ',
     9  '----------'/
      data cnsq3 /
     1  'Ground-   ',
     2  'level    ',
     3  'time inte-',
     4  'grated air',
     5  'concentra-',
     6  'tion      ',
     7  '(source*  ',
     8  ' s/m**3)  ',
     9  '----------'/
      data methdr /
     1  '           ',
     2  '           ',
     3  '           ',
     4  '           ',
     5  '           ',
     6  '  Met. File',
     7  '  #  Day  Hr',
     8  '           ',
     9  '  -  ---  --'/
      data methdr2 /
     1  '   ,     ,    ',
     2  '   ,     ,    ',
     3  '   ,     ,    ',
     4  '   ,     ,    ',
     5  '   ,     ,    ',
     6  'Met, File,    ',
     7  '  #,  Day,  Hr',
     8  '   ,     ,    ',
     9  '  -,  ---,  --'/
      data disthdr /
     1  '        ',
     2  '        ',
     3  '        ',
     4  '        ',
     5  '        ',
```

```
      6  'MEOI    ',
      7  'Distance',
      8  '(meters)',
      9  '--------'/
       data namehdr /
      1  ' ',
      2  ' ',
      3  ' ',
      4  ' ',
      5  ' ',
      6  ' ',
      7  ' ',
      8  'File Date      File Time     CDO Output Filename',
      9  '-------------- ------------  -------------------'/

       if (verbose.eq.1) then
         if (nyr.eq.1) then
           write(file,91010)
         else
           write(file,91020)
         endif

         if (abs(ctype).eq.1) then
           write(file,91110)
         else
           write(file,91120)
         endif
       endif

       do i=1,9
         if (verbose.eq.1) then
           if(comma.ne.1) then
             write(file,'(a)') label1(i,col1)//'  '//
     1         cnsq1(i)//methdr(i)//'  '//disthdr(i)//'  '//
     2         cnsq2(i)//methdr(i)//'  '//disthdr(i)//'  '//
     3         cnsq3(i)//methdr(i)//'  '//disthdr(i)//'  '//
     4         label1(i,col1)
           else
             write(file,'(a)') label1(i,col1)//',  '//
     1         cnsq1(i)//','//methdr2(i)//',  '//disthdr(i)//',  '//
     2         cnsq2(i)//','//methdr2(i)//',  '//disthdr(i)//',  '//
     3         cnsq3(i)//','//methdr2(i)//',  '//disthdr(i)//',  '//
     4         label1(i,col1)
           endif
         else
           write(file,'(a)') cnsq1(i)//'  '//
     2                       cnsq2(i)//'  '//
     3                       cnsq3(i)//'  '//
     4                       trim(namehdr(i))
         endif
       enddo

       return
91010 format(46('-'),' Current Single-Year CDF ',47('-'))
91020 format(46('-'),' Aggregate Multi-Year CDF ',46('-'))
91110 format(27('-'),' of Chi/Q; Puff Release Chi/Q; ',
     &  'and Integrated Air Concentration ',27('-')/)
```

```
571   91120 format(22('-'),' of LPF times Chi/Q; Puff Release Chi/Q; ',
572        &  'and Integrated Air Concentration ',22('-')/)
573         end
574   *=====================================================================
575         subroutine get_lpf_table
576         include 'POSTMAX2.fh'
577   *---- local variables
578         character*80 header
579         integer i,j
580
581         if (calc_type.lt.0) call copy_file(lpffile,trc)
582
583         open(lpf,file=lpffile,status='old')
584         open(eco,file=lpfecho)
585
586         write(eco,91000) lpffile
587         do i=1,2
588           read(lpf,'(a)',end=70000,err=70000) header
589           write(eco,91010) i, trim(header)
590         enddo
591
592         n_lpf=0
593
594   10000 n_lpf=n_lpf+1
595         if (n_lpf.gt.25) then
596           write(crt,90060) trim(lpffile), 25
597           write(out,90060) trim(lpffile), 25
598           write(csv,90060) trim(lpffile), 25
599           write(eco,90060) trim(lpffile), 25
600           stop
601         endif
602         read(lpf,*,end=20000,err=70000) (lpftbl(n_lpf,i),i=1,17)
603         goto 10000
604
605   20000 n_lpf=n_lpf-1
606         close(lpf)
607
608         write(eco,91020)
609         call print_lpf_table(eco)
610
611         if (calc_type.lt.0) then
612           write(trc,91025)
613           call print_lpf_table(trc)
614         endif
615
616         if (n_lpf.lt.1) then
617           write(crt,90050) trim(lpffile)
618           write(out,90050) trim(lpffile)
619           write(csv,90050) trim(lpffile)
620           write(eco,90050) trim(lpffile)
621           stop
622         endif
623         if (lpftbl(1,1).gt.0.5d+00) then
624           write(crt,90010) trim(lpffile)
625           write(out,90010) trim(lpffile)
626           write(csv,90010) trim(lpffile)
627           write(eco,90010) trim(lpffile)
```

```
628           stop
629         endif
630         if (lpftbl(1,1).le.0.0) then
631           write(crt,90020) trim(lpffile)
632           write(out,90020) trim(lpffile)
633           write(csv,90020) trim(lpffile)
634           write(eco,90020) trim(lpffile)
635           stop
636         endif
637         do i=2,n_lpf
638           if (lpftbl(i,1).le.lpftbl(i-1,1)) then
639             write(crt,90030) trim(lpffile)
640             write(out,90030) trim(lpffile)
641             write(csv,90030) trim(lpffile)
642             write(eco,90030) trim(lpffile)
643             stop
644           endif
645         enddo
646         do i=1,n_lpf
647           do j=2,17
648             if (lpftbl(i,j).le.0.0) then
649               write(crt,90040) trim(lpffile)
650               write(out,90040) trim(lpffile)
651               write(csv,90040) trim(lpffile)
652               write(eco,90040) trim(lpffile)
653               stop
654             endif
655           enddo
656         enddo
657
658         close(eco)
659         return
660
661     70000 write(crt,90050) trim(lpffile)
662         write(out,90050) trim(lpffile)
663         write(csv,90050) trim(lpffile)
664         write(eco,90050) trim(lpffile)
665         stop
666
667     90010 format(
668          1  /'Error: In file ',a,'.',
669          2  /'       LPF table is undefined for low wind speeds.'
670          3  /'       First wind speed must be 0.5 m/s or less.',
671          4 //'Execution terminated.')
672     90020 format(
673          1  /'Error: In file ',a,'.',
674          2  /'       LPF table contains a non-positive wind speed.',
675          3 //'Execution terminated.')
676     90030 format(
677          1  /'Error: In file ',a,'.',
678          2  /'       LPF table contains a wind speed that is less than',
679          3  /'       or equal to the previous wind speed.',
680          4 //'Execution terminated.')
681     90040 format(
682          1  /'Error: In file ',a,'.',
683          2  /'       LPF table contains a non-positive LPF value.',
684          3 //'Execution terminated.')
```

```
685   90050 format(
686        1  /'Error: Reading file ',a,'.',
687        2  /'       LPF files must begin with two title lines.',
688        3  /'       The LPF table has the wind speed in column 1',
689        4  /'       then 16 columns of Leak Path Factors for',
690        5  /'       the 16 wind directions.',
691        6 //'Execution terminated.')
692   90060 format(
693        1  /'Error: In file ',a,'.',
694        2  /'       LPF table appears to exceed program dimension (',
695        2  i2,').')
696   91000 format('LPF File: ',a,/)
697   91010 format('Header',i2,': ',a)
698   91020 format(/'LPF Table'/,9('-'))
699   91025 format('Trace Section - LPF Table')
700         end
701   *=====================================================================
702         subroutine print_lpf_table(dev)
703         include 'POSTMAX2.fh'
704   *---- input values
705         integer dev
706   *----         is the fileio number to write to
707   *---- local variables
708         integer i, j
709
710         write(dev,91031)
711         do i=1,n_lpf
712           write(dev,91041) lpftbl(i,1),(lpftbl(i,j),j=2,5)
713         enddo
714         write(dev,91032)
715         do i=1,n_lpf
716           write(dev,91041) lpftbl(i,1),(lpftbl(i,j),j=6,9)
717         enddo
718         write(dev,91033)
719         do i=1,n_lpf
720           write(dev,91041) lpftbl(i,1),(lpftbl(i,j),j=10,13)
721         enddo
722         write(dev,91034)
723         do i=1,n_lpf
724           write(dev,91041) lpftbl(i,1),(lpftbl(i,j),j=14,17)
725         enddo
726         write(dev,91050)
727         return
728   91031 format(/'  WindSpeed',2x,'North',6x,'NNE',7x,'Northeast',3x,'ENE')
729   91032 format(/'  WindSpeed',2x,'East ',6x,'ESE',7x,'Southeast',3x,'SSE')
730   91033 format(/'  WindSpeed',2x,'South',6x,'SSE',7x,'Southwest',3x,'WSW')
731   91034 format(/'  WindSpeed',2x,'West ',6x,'WNE',7x,'Northwest',3x,'NNW')
732   91041 format(1p,5e11.3)
733   91050 format(50('-'))
734         end
735   *=====================================================================
736         subroutine get_boundary_distances
737         include 'POSTMAX2.fh'
738   *---- local variables
739         character chaff*80
740         integer i
741
```

```
      if (calc_type.lt.0) call copy_file(bndfile,trc)

      open(bnd,file=bndfile,status='old')

      read(bnd,'(a)',end=70000,err=70000) (chaff,i=1,2)

      do i=1,16
        read(bnd,*,end=70000,err=70000) bndry(i)
        if (bndry(i) .le. 0.d0) then
          write(crt,90010) trim(bndfile)
          write(out,90010) trim(bndfile)
          write(csv,90010) trim(bndfile)
          stop
        endif
      enddo
      close(bnd)

      if (calc_type.lt.0) then
        write(trc,91000)
        write(trc,91010) (i,bndry(i),i=1,16)
        write(trc,91020)
      endif

      return

70000 write(crt,90020) trim(bndfile)
      write(out,90020) trim(bndfile)
      write(csv,90020) trim(bndfile)
      stop

90010 format(
     1  /'Error: In file ',a,'.',
     2  /'       Non-positive control boundary dist encountered.',
     3 //'Execution terminated.')
90020 format(
     1  /'Error: Reading file ',a,'.',
     2  /'       Boundary files must begin with two title lines.',
     3  /'       After the title lines, boundary files must have',
     4  /'       16 lines for the distances in the 16 wind directions.',
     5 //'Execution terminated.')
91000 format('Trace Section - Site Boundary Distances')
91010 format('bndry(',i2,') = ',f9.2)
91020 format(80('-'))
      end
*=======================================================================
      subroutine copy_file(infilename,outfilenum)
      include 'POSTMAX2.fh'
*---- input values
      character*(*) infilename
*----               is the name of the source file
*----               which is not open before the subroutine call
*----               and is closed by copy_file
      integer outfilenum
*----               is the fileio number for the destination file
*----               which is open before the subroutine call
*----               and is left open by copy_file
*---- local variables
```

```
      character*256 line

      if (outfilenum.eq.trc) then
*-----  write header information to the trace file
        write(trc,90020) trim(infilename)
        write(trc,90010)
      endif

      open(temp,file=infilename,status='old')

10000 read(temp,'(a)',end=20000) line
      if (outfilenum.ne.summ) then
*-----  every line is copied
        write(outfilenum,'(a)') trim(line)
      else
*-----  do not copy results for the current outfile and cdofile
*-----  to the summary file because new results will be written
        if (index(line,outfile).eq.0.and.index(line,cdofile(1)).eq.0)
     1  then
          write(outfilenum,'(a)') trim(line)
        endif
      endif
      goto 10000

20000 close(temp)

      if (outfilenum.eq.trc) then
*-----  write footer information to the trace file
        write(trc,90010)
        write(trc,90030) trim(infilename)
        write(trc,90010)
      endif

      return
90010 format(80('-'))
90020 format('Trace Section - Echo Input File: ',a)
90030 format('End of File: ',a)
      end
*=======================================================================
      subroutine get_met_data
      include 'POSTMAX2.fh'
*---- local variables
      character*80 header(2)
      integer i, idum, ispd, stablty(365,24)

      if (calc_type.lt.0) call copy_file(metfile,trc)

      open(met,file=metfile,status='old')

      read(met,'(a)') (header(i),i=1,2)

      do day=1,365
        do hour=1,24
          read(met,80010) idum, idum, winddir(day,hour), ispd,
     1                    stablty(day,hour)

          if (ispd.lt.5) ispd=5
```

```
856             windspd(day,hour)=dble(ispd)/10.d0
857           enddo
858         enddo
859         close(met)
860
861         if (calc_type.lt.0) then
862           write(trc,91010)
863           do day=1,365
864             do hour=1,24
865               write(trc,91020) day, hour, winddir(day,hour),
866        1        windspd(day,hour), stablty(day,hour)
867             enddo
868           enddo
869           write(trc,91030)
870         endif
871
872         do i=len_trim(metfile),1,-1
873           if(metfile(i:i).eq.'\') goto 10000
874         enddo
875         i=0
876   10000 continue
877
878         csvfile = trim(folder)
879        1          // metfile(i+1:index(metfile,'.'))
880        2          // 'csv'
881
882         open(met,file=csvfile)
883         write(met,92010) trim(metfile)
884         do i=1,2
885           write(met,92020) i,trim(header(i))
886         enddo
887         write(met,92030)
888         do day=1,365
889           do hour=1,24
890             write(met,92040) day, hour, winddir(day,hour),
891        1      windspd(day,hour), stablty(day,hour)
892           enddo
893         enddo
894         close(met)
895
896         return
897   80010 format(i4,3i3,i1)
898   91010 format('Trace Section - Meteorological Data',
899        1  /'Day Hour Winddir Windspd Stability')
900   91020 format(i3,i5,i8,f8.1,i5)
901   91030 format(80('-'))
902   92010 format('File: ',a,/)
903   92020 format('Header',i2,': ',a)
904   92030 format(/'Day, Hour, Winddir, Windspd, Stability')
905   92040 format(i3,',',i5,',',i8,',',f8.1,',',i2)
906         end
907   *=====================================================================
908         subroutine get_maccs_parms
909         include 'POSTMAX2.fh'
910   *---- local variables
911         integer idebug, metcod, nsmpls
912
```

```
      if (calc_type.lt.0) write(trc,91000)

*---- MACCS must be run with idebug = 2
      call find_int_after_string(infile,'OCIDEBUG001',idebug)
      if (idebug.ne.2) then
        write(crt,90010) trim(infile)
        write(out,90010) trim(infile)
        write(csv,90010) trim(infile)
        stop
      endif

*---- Check the meteorology sampling option
      call find_int_after_string(infile,'M1METCOD001',metcod)
      if (metcod.ne.5) then
        write(crt,90020) trim(infile)
        write(out,90020) trim(infile)
        write(csv,90020) trim(infile)
        stop
      endif

*---- get number of radial distances from MACCS file
      call find_int_after_string(infile,'GENUMRAD001',numrad)

*---- get number of samples per day from MACCS file
      call find_int_after_string(infile,'M4NSMPLS001',nsmpls)

*---- total number of samples in MACCS file
      nmet=365*nsmpls

      return
90010 format(
     1  /'Error: In file ',a,'.',
     2  /'       MACCS must have OCIDEBUG001 = 2 to run POSTMAX.',
     3 //'Execution terminated.')
90020 format(
     1  /'Error: In file ',a,'.',
     2  /'       MACCS must have M1METCOD001 = 5 to run POSTMAX.',
     3 //'Execution terminated.')
91000 format('Trace Section - Checking MACCS Parameters')
      end
*=======================================================================
      subroutine find_int_after_string(filename,string,ival)
      include 'POSTMAX2.fh'
*---- input values
      character*(*) filename
*----               is the file containing the character string
      character*(*) string
*----               is the character string to find in filename
*---- output value
      integer ival
*----         is the integer value found after the string
*---- local variables
      integer file
      character*80 chaff
      data file / 1 /

      open(file,file=filename,status='old')
```

```
970
971   10010 read(file,'(a)',err=50000) chaff
972         if (index(chaff,string).eq.0) goto 10010
973         if (calc_type.lt.0) then
974           write(trc,90020) trim(filename),trim(string)
975           write(trc,'(a)') trim(chaff)
976         endif
977
978         chaff=chaff(index(chaff,string)+11:len(chaff))
979         read(chaff,*) ival
980         if (calc_type.lt.0) then
981           write(trc,90030) ival
982           write(trc,90010)
983         endif
984
985         close(file)
986         return
987
988   50000 write(crt,99000) trim(filename),trim(string)
989         write(out,99000) trim(filename),trim(string)
990         write(csv,99000) trim(filename),trim(string)
991         stop
992
993   90010 format(80('-'))
994   90020 format('Searching file "',a,'" for string "',a,'" found line:')
995   90030 format('and extracted the value:',i4)
996   99000 format(
997        1  /'Error: In file ',a,'.',
998        2  /'       String "',a,'" was not found.',
999        3 //'Execution terminated.')
1000        end
1001  *=====================================================================
1002        subroutine queue_trial_results
1003        include 'POSTMAX2.fh'
1004  *---- local variables
1005        character chaff*80
1006
1007        open(in,file=infile,status='old')
1008  10000 read(in,'(a)') chaff
1009        if (index(chaff,'TRIAL        DAY').eq.0) goto 10000
1010
1011        if(calc_type.lt.0) write(trc,91000)
1012
1013        return
1014  91000 format('Trace Section - MACCS Trial Results')
1015        end
1016  *=====================================================================
1017        subroutine read_trial_results
1018        include 'POSTMAX2.fh'
1019  *---- local variables
1020        character chaff*80, iter_str*4, line*256
1021        integer i, idum
1022        real dum2, dum3, dum4, dum5
1023
1024  *---- convert iteration number to a string
1025        write(iter_str,'(i4)') iter
1026
```

```
*---- read data from each trial
11000 read(in,'(a)') chaff
      if (index(chaff,iter_str).ne.8) goto 11000

      if (calc_type.lt.0) write(trc,91010) trim(chaff)
      backspace(in)
      read(in,*) idum,day,hour
      do i=1,3
        day_time(iter,i,1) = 1
        day_time(iter,i,2) = day
        day_time(iter,i,3) = hour
      enddo
      if (calc_type.lt.0) write(trc,91020) day, hour

12000 read(in,'(a)') chaff
      if (index(chaff,'1').ne.2) goto 12000

      backspace(in)
      do i=1,numrad
        if (calc_type.lt.0) then
          read(in,'(a)') line
          write(trc,*) trim(line)
          backspace(in)
        endif
        read(in,80010)
     1    idum,dist(i),glcon(i),dum2,xovrq(i),dum3,dum4,dum5,sigma(i)
      enddo

      if (iter.gt.1) return

*---- Check minimum maccs2 radial distance is less than the
*---- minimum control boundary distance
      do i=1,16
        if (dist(1).gt.bndry(i)) then
          write(crt,90010) trim(infile)
          write(out,90010) trim(infile)
          write(csv,90010) trim(infile)
          stop
        endif
      enddo

*---- Check to see that radial distances exceed all boundary points
      do i=1,16
        if (dist(numrad).lt.bndry(i)) then
          write(crt,90020) trim(infile)
          write(out,90020) trim(infile)
          write(csv,90020) trim(infile)
          stop
        endif
      enddo

      return
80010 format(i2,4d10.2,2f8.4,2d10.2)
90010 format(
     1  /'Error: In file ',a,'.',
     2  /'       Midpoint of the first MACCS radial grid is',
     2  /'       greater than a control boundary distance.',
```

```
     3  /'       Decrease midpoint of minimum radial grid',
     4  /'       and repeat the MACCS2 calculation.',
     5 //'Execution terminated.')
90020 format(
     1  /'Error: In file ',a,'.',
     2  /'       Midpoint of the last MACCS radial grid is',
     2  /'       less than a control boundary distance.',
     3  /'       Increase or extend the MACCS radial grid'
     4  /'       and repeat the MACCS2 calculation.',
     5 //'Execution terminated.')
91010 format('Trial identifying line:',/a)
91020 format('Trial day =',i4,' hour =',i3)
      end
*=======================================================================
      subroutine calc_consequences
      include 'POSTMAX2.fh'
*---- local variables
      integer imax, imin, ipeak
      real*8 bndry_dist, sigmax

*---- find MACCS distances surrounding the site boundary distance
      bndry_dist=bndry(winddir(day,hour))
      imax=2
      do while (dist(imax).lt.bndry_dist)
        imax=imax+1
      enddo
      imin=imax-1

      if (calc_type.lt.0) then
        write(trc,90010) winddir(day,hour)
        write(trc,90020) bndry_dist
      endif

*---- assume MEIO location is on the site boundary
*---- calculate site boundary results by interpolation
      meoi_dist(iter,1)=bndry_dist
      call loglint(dist(imin),dist(imax),bndry_dist,
     1             xovrq(imin),xovrq(imax),cnsq(iter,1))

      meoi_dist(iter,2)=bndry_dist
      call loglint(dist(imin),dist(imax),bndry_dist,
     1             sigma(imin),sigma(imax),sigmax)
      if (calc_type.lt.0) then
        write(trc,90030)
        call print_trial_conseq(imin,imax,0,sigmax,meoi_dist(iter,2))
      endif
*---- save (Chi/Q)/sigmax for puff instantaneous air concentration
      cnsq(iter,2) = cnsq(iter,1)/sigmax

      meoi_dist(iter,3)=bndry_dist
      call loglint(dist(imin),dist(imax),bndry_dist,
     1             glcon(imin),glcon(imax),cnsq(iter,3))

*---- check if any offsite Chi/Q is greater than site boundary
      call find_peak(xovrq,cnsq(iter,1),meoi_dist(iter,1),imax,ipeak)
      if (calc_type.lt.0) then
        write(trc,90031)
```

```
1141          call print_trial_conseq(imin,imax,ipeak,cnsq(iter,1),
1142         1    meoi_dist(iter,1))
1143          endif
1144          if (ipeak.eq.numrad) then
1145            call print_warning('Chi/Q',cnsq(iter,1))
1146          endif
1147
1148    *---- check if any offsite (Chi/Q)/sigmax is greater than site boundary
1149          call find_peak_fraction(xovrq,sigma,cnsq(iter,2),
1150         1  meoi_dist(iter,2),imax,ipeak)
1151    *---- calc puff instantaneous air concentration
1152          cnsq(iter,2) = cnsq(iter,2)*windspd(day,hour)/dsqrt(2*pi)
1153          if (calc_type.lt.0) then
1154            write(trc,90032)
1155            call print_trial_conseq(imin,imax,ipeak,cnsq(iter,2),
1156         1    meoi_dist(iter,2))
1157          endif
1158          if (ipeak.eq.numrad) then
1159            call print_warning('Puff Chi/Q',cnsq(iter,2))
1160          endif
1161
1162    *---- check if any offsite ground-level air concentration is greater
1163    *---- than site boundary
1164          call find_peak(glcon,cnsq(iter,3),meoi_dist(iter,3),imax,ipeak)
1165          if (calc_type.lt.0) then
1166            write(trc,90033)
1167            call print_trial_conseq(imin,imax,ipeak,cnsq(iter,3),
1168         1    meoi_dist(iter,3))
1169            write(trc,*)
1170          endif
1171          if (ipeak.eq.numrad) then
1172            call print_warning('GLCON',cnsq(iter,3))
1173          endif
1174
1175    *---- if selected, multiply MACCS results by LPF
1176          if (abs(calc_type).eq.2) then
1177            call get_lpf
1178          endif
1179
1180          if (calc_type.lt.0) write(trc,90050)
1181          return
1182    90010 format('Wind direction    =',i6)
1183    90020 format('Boundary distance =',f9.2)
1184    90030 format(' iter type   meoi dist imin imax      sigmax')
1185    90031 format(' iter type   meoi dist imin imax   cnsq(#,1)')
1186    90032 format(' iter type   meoi dist imin imax   cnsq(#,2)')
1187    90033 format(' iter type   meoi dist imin imax   cnsq(#,3)')
1188    90050 format(128('-'))
1189          end
1190    *=====================================================================
1191          subroutine loglint(x1,x2,xnew,y1,y2,ynew)
1192    *----   interpolates values assuming a linear
1193    *----   variation in x and a logarithmic variation in y.  To avoid the
1194    *----   logarithm of very small (and zero) y values, a constant quantity is
1195    *----   assumed that should be set to the smallest representable number.
1196    *----
1197    *----    It is assumed that x1<xnew<x2.  The interpolated value is returned
```

```
*----    in the argument ynew.
*----
*----                                                                bcl 3/8/00
*----
      implicit none
*---- input values
      real*8 x1,x2,xnew,y1,y2
*---- output value
      real*8 ynew
*---- local variables
      real*8 a,b

*---- interpolate and return
      if (y1.le.1.d-99) then
         a=-1.d+99
      else
         a=dlog(y1)
      endif
      if (y2.le.1.d-99) then
         b=-1.d+99
      else
         b=dlog(y2)
      endif
      ynew=dexp((b-a)*(xnew-x2)/(x2-x1)+b)

      return
      end
*=======================================================================
      subroutine find_peak(a,amax,dmax,istart,ipeak)
      include 'POSTMAX2.fh'
*---- subroutine arguements
      real*8 a(*)
*----        input = array in which to find peak value
      real*8 amax
*----        input  = initial value of amax
*----        output = maximum of amax and a(i),i=istart,numrad
      real*8 dmax
*----        output = distance corresponding to new amax
*----                 if amax does not change, dmax does not change
      integer istart
*----         input = begin search for peak value at a(istart)
      integer ipeak
*----         output = pointer to peak a(i) value;
*----                  if amax does not change, ipeak=0
*---- local variables
      integer j

      ipeak = 0
      do j=istart,numrad
        if (a(j).gt.amax) then
          amax=a(j)
          dmax=dist(j)
          ipeak=j
        endif
      enddo
      return
      end
```

```
1255  *=====================================================================
1256        subroutine find_peak_fraction(a,b,abmax,dmax,istart,ipeak)
1257        include 'POSTMAX2.fh'
1258  *---- subroutine arguements
1259        real*8 a(*)
1260  *----         input = numerator array
1261        real*8 b(*)
1262  *----         input = denomator array
1263        real*8 abmax
1264  *----         input  = initial value of a/b
1265  *----         output = maximum of abmax and a(i)/b(i),i=istart,numrad
1266        real*8 dmax
1267  *----         output = distance corresponding to new abmax
1268  *----                  if abmax does not change, dmax does not change
1269        integer istart
1270  *----          input = begin search for peak value at a(istart)/b(istart)
1271        integer ipeak
1272  *----          output = pointer to peak a(i) value;
1273  *----                   if amax does not change, ipeak=0
1274  *---- local variables
1275        integer j
1276        real*8 c
1277
1278        ipeak = 0
1279        do j=istart,numrad
1280          c=a(j)/b(j)
1281          if (c.gt.abmax) then
1282            abmax=c
1283            dmax=dist(j)
1284            ipeak=j
1285          endif
1286        enddo
1287        return
1288        end
1289  *=====================================================================
1290        subroutine print_trial_conseq(imin,imax,ipeak,conseq,dconseq)
1291        include 'POSTMAX2.fh'
1292  *---- input values
1293        integer imin,imax,ipeak
1294        real*8  conseq,dconseq
1295
1296        if (ipeak.eq.0) then
1297          write(trc,90020) iter,'site',dconseq,imin,imax,conseq
1298        else
1299          write(trc,90020) iter,'peak',dconseq,ipeak,ipeak,conseq
1300        endif
1301        return
1302  90020 format(i5,1x,a4,f12.2,2i5,1pe12.4)
1303        end
1304  *=====================================================================
1305        subroutine print_warning(var_name,maxvalu)
1306        include 'POSTMAX2.fh'
1307  *---- local variables
1308        character*(*) var_name
1309        real*8        maxvalu
1310
1311  *---- MACCS results increase out to the last grid point
```

```
*---- the peak Chi/Q has not been found
      write(crt,*)
      write(crt,91010) iter,trim(var_name),maxvalu

      if (user_abort.eq.'?') then
        wrnfile = infile(1:index(infile,'.'))//'wrn'
        open(wrn,file=wrnfile)
        user_abort='n'
        write(wrn,91020)
      endif

      write(wrn,91010) iter,trim(var_name),maxvalu

      return
91010 format(i9,3x,a10,3x,1pe11.4)
91020 format('Warning file',//
     1'For the following iterations and consequences, the maximum ',
     2'MACCS2 consequence',/
     3'is located in the last radial zone.  The peak MEOI consequence ',
     4'may be located',/
     5'beyond the last MACCS2 radial zone.  This may indicate that ',
     6'the MACCS2 radial',/
     7'zones need to be increased.',//
     8'Iteration  Consequence  Maximum Value',/
     9'---------  -----------  -------------')
      end
*=======================================================================
      subroutine get_lpf
      include 'POSTMAX2.fh'
*---- local variables
      integer i, j, k

      j=winddir(day,hour)+1

      do k=1,n_lpf-1
        if (windspd(day,hour).eq.lpftbl(k,1)) then
          i=k
          goto 10000
        endif
        if (windspd(day,hour).gt.lpftbl(k,1).and.
     1      windspd(day,hour).lt.lpftbl(k+1,1)) then
          if (lpftbl(k+1,j).gt.lpftbl(k,j)) then
            i=k+1
          else
            i=k
          endif
          goto 10000
        endif
      enddo
      i=n_lpf

10000 do k=1,3
        cnsq(iter,k)=lpftbl(i,j)*cnsq(iter,k)
      enddo

      if (calc_type.lt.0) then
        write(trc,90010) lpftbl(i,j)
```

```
1369          do k=1,3
1370            write(trc,90020) k
1371            write(trc,90030) iter,meoi_dist(iter,k),cnsq(iter,k)
1372          enddo
1373        endif
1374
1375        return
1376    90010 format('LPF =',f10.6)
1377    90020 format(' iter        meoi dist             cnsq(#,',i1,')')
1378    90030 format(i5,5x,f12.2,10x,1pe12.4)
1379        end
1380    *=====================================================================
1381        subroutine sort_consequences
1382        include 'POSTMAX2.fh'
1383    *---- local variables
1384        integer i, j, k
1385
1386        if (calc_type.lt.0) then
1387          write(trc,91000)
1388          call print_column_headers(trc,calc_type,1,2,1,0)
1389          do i=1,nmet
1390            write(trc,91010)
1391       1      i,(cnsq(i,j),(day_time(i,j,k),k=1,3),int(meoi_dist(i,j)),
1392       2      j=1,3),i
1393          enddo
1394        endif
1395
1396        do j=1,3
1397          call heapsort(nmet,cnsq(1,j),ptr(1,j))
1398        enddo
1399
1400        if (calc_type.lt.0) then
1401          write(trc,91020)
1402          do i=1,nmet
1403            write(trc,91030) i,(ptr(i,j),j=1,3)
1404          enddo
1405          write(trc,91040)
1406          call print_column_headers(trc,calc_type,1,2,1,0)
1407          do i=1,nmet
1408            write(trc,91010) i,
1409       1      (cnsq(ptr(i,j),j),(day_time(ptr(i,j),j,k),k=1,3),
1410       2       int(meoi_dist(ptr(i,j),j)),j=1,3),i
1411          enddo
1412          write(trc,91050)
1413        endif
1414
1415        return
1416    91000 format('Trace Section - Unsorted POSTMAX Results')
1417    91010 format(i7,3(1pe12.4,i3,i5,i4,i10),i9)
1418    91020 format(80('-'),
1419         1  /'Trace Section - POSTMAX Results Sort Order',
1420         2  /80('-'))
1421    91030 format(4i10)
1422    91040 format(80('-'),
1423         1  /'Trace Section - Sorted POSTMAX Results')
1424    91050 format(80('-'))
1425        end
```

```
1426  *=====================================================================
1427        subroutine print_consequences(nyr)
1428        include 'POSTMAX2.fh'
1429  *---- input values
1430        integer nyr
1431  *----            =1 results for a single year
1432  *----            >1 results for multiple years
1433  *---- local variables
1434        integer i, j, k
1435        real*8 above95, below95, prob, r
1436
1437        r=dble(nmet)
1438        below95 = 0.95d0 - 0.5d+0/r
1439        above95 = 0.95d0 + 0.5d+0/r
1440        do i=1,nmet
1441          prob=dble(i)/r
1442          write(out,90010) prob,
1443       1    (cnsq(ptr(i,j),j),(day_time(ptr(i,j),j,k),k=1,3),
1444       2     int(meoi_dist(ptr(i,j),j)),j=1,3), prob
1445          write(csv,90015) prob,
1446       1    (cnsq(ptr(i,j),j),(day_time(ptr(i,j),j,k),k=1,3),
1447       2     int(meoi_dist(ptr(i,j),j)),j=1,3), prob
1448          if (prob.ge.below95.and.prob.le.above95) then
1449            do j=1,3
1450              cnsq95(j)=cnsq(ptr(i,j),j)
1451            enddo
1452          endif
1453        enddo
1454        if (nyr.eq.1) then
1455          write(crt,90020)
1456          write(out,90020)
1457          write(csv,90020)
1458        else
1459          write(crt,90030)
1460          write(out,90030)
1461          write(csv,90030)
1462        endif
1463        write(crt,90040) (cnsq95(i),i=1,3)
1464        write(out,90040) (cnsq95(i),i=1,3)
1465        write(csv,90040) (cnsq95(i),i=1,3)
1466
1467        close(out)
1468        close(csv)
1469
1470        if (nyr.eq.1.and.user_abort.ne.'?') then
1471          write(wrn,90020)
1472          write(wrn,90040) (cnsq95(i),i=1,3)
1473          close(wrn)
1474        endif
1475
1476        return
1477  90010 format(f7.5,3(1pe12.4,i3,i5,i4,i10),0pf9.5)
1478  90015 format(f7.5,3(',',1pe12.4,',',i3,',',i5,',',i4,',',i10),
1479       1  ',',0pf9.5)
1480  90020 format(//,"Current single-year 95th percentile values:")
1481  90030 format(//,"Aggregate multi-year 95th percentile values:")
1482  90040 format(
```

```
1483        1 /,"Ground-level Chi/Q without deposition =",1pe11.4," s/m**3",
1484        2 /,"Puff Release G-L Chi/Q w/o deposition =",1pe11.4," 1/m**3",
1485        3 /,"Ground-level time-integrated air conc =",1pe11.4,
1486        4   " source*s/m**3")
1487         end
1488    *=====================================================================
1489         subroutine open_summary_file
1490         include 'POSTMAX2.fh'
1491    *---- local variables
1492         logical ex
1493
1494         summfile=trim(folder)//'POSTMAX_Summary.txt'
1495         tempfile=trim(folder)//'POSTMAX.tmp'
1496
1497    *---- check if input file exists
1498         inquire (file=summfile, exist=ex)
1499
1500         if (.not.ex) then
1501           open(summ,file=summfile)
1502           call print_column_headers(summ,calc_type,2,1,2,0)
1503         else
1504    *-----  copy summary file to temporary file
1505           open(sum2,file=tempfile)
1506           call copy_file(summfile,sum2)
1507           close(sum2)
1508    *-----  copy temporary file to summary file
1509    *-----  except the lines containing the outfile and cdofile names
1510           open(summ,file=summfile)
1511           call copy_file(tempfile,summ)
1512         endif
1513
1514         return
1515         end
1516    *=====================================================================
1517         subroutine print_summary(filename)
1518         include 'POSTMAX2.fh'
1519    *---- input values
1520         character*(*) filename
1521    *----                 is the file containing the character string
1522    *---- local variables
1523         integer i, File_Status(13)
1524         integer GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
1525        &        GMT_Minute, GMT_Second, GMT_Weekday
1526         character Weekday_Label(7)*3
1527
1528    *---- text for weekday values from 1 to 7
1529         data Weekday_Label / 'Sun','Mon','Tue','Wed','Thu','Fri','Sat'/
1530
1531         call Get_File_Status(filename, crt, File_Status,
1532        1     GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
1533        2     GMT_Minute, GMT_Second, GMT_Weekday)
1534
1535         write(summ,90010) (cnsq95(i),i=1,3),
1536        1     Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
1537        2     GMT_Hour, GMT_Minute, GMT_Second, trim(filename)
1538
1539         return
```

```
90010 format(3(1pe10.4,2x),a3,1x,i2,'/',i2.2,'/',i4,
     1       1x,i2,2(':',i2.2),' GMT  ',a)
      end
*=====================================================================
      subroutine get_previous_cdo(n)
      include 'POSTMAX2.fh'
*---- input values
      integer n
*----           = index to cdofile to get
*---- local variables
      character chaff*80
      integer i,j,int_dist(3)
      real*8 dum, prob

      if (calc_type.lt.0) then
        call copy_file(cdofile(n),trc)
        if (abs(calc_type).eq.3) then
          write(trc,90100)
        endif
      endif

      open(cdo,file=cdofile(n),status='old')

*---- read command line history from the cdo file
*---------------------------------------------------------------------
10000 n_oldhdr=n_oldhdr+1
      read(cdo,'(a256)') oldhdr(n_oldhdr)

      if (abs(calc_type).eq.3 .and.
     1    index(oldhdr(n_oldhdr),'Command Line').gt.0) then
        read(oldhdr(n_oldhdr)(15:256),*) chaff, i
        if (calc_type.lt.0) then
          write(trc,90110) trim(cdofile(n)),trim(oldhdr(n_oldhdr)),i
        endif
        if (n.eq.1) then
          file_calc_type=i
        else
          if(abs(i).ne.abs(file_calc_type)) then
            write(crt,90010) trim(cdofile(n)),i,
     1        trim(cdofile(1)),file_calc_type
            write(out,90010) trim(cdofile(n)),i,
     1        trim(cdofile(1)),file_calc_type
            write(csv,90010) trim(cdofile(n)),i,
     1        trim(cdofile(1)),file_calc_type
            stop
          endif
        endif
      endif

      if (index(oldhdr(n_oldhdr),'Met. File').gt.0) then
        read(oldhdr(n_oldhdr)(13:14),*) nmetfiles
        if (abs(calc_type).eq.3) then
          if (nmetfiles.ne.1) then
            write(crt,90020) trim(cdofile(n))
            write(out,90020) trim(cdofile(n))
            write(csv,90020) trim(cdofile(n))
            stop
```

```
1597            endif
1598            write(oldhdr(n_oldhdr)(13:14),'(i2)') n
1599          endif
1600        endif
1601
1602        if (index(oldhdr(n_oldhdr),'Year CDF').eq.0) goto 10000
1603    *-----------------------------------------------------------------------
1604        n_oldhdr=n_oldhdr-1
1605
1606        if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
1607    *-----  set new file number for most current postmax run results
1608          nmetfiles=nmetfiles+1
1609          do i=1,nmet
1610            do j=1,3
1611              day_time(i,j,1)=nmetfiles
1612            enddo
1613          enddo
1614        endif
1615
1616    *---- read past the column headings in the cdo file
1617        do i=1,11
1618          read(cdo,'(a)') chaff
1619        enddo
1620
1621    *---- read previous cdo results
1622    *-----------------------------------------------------------------------
1623    20000 nmet=nmet+1
1624        read(cdo,*) prob,(cnsq(nmet,i),(day_time(nmet,i,j),j=1,3),
1625       1  int_dist(i),i=1,3),dum
1626        do i=1,3
1627          meoi_dist(nmet,i)=dble(int_dist(i))
1628        enddo
1629        if (abs(calc_type).eq.3) then
1630          do i=1,3
1631            day_time(nmet,i,1)=n
1632          enddo
1633        endif
1634        if (prob.lt.1.0d+00) goto 20000
1635    *-----------------------------------------------------------------------
1636
1637        close(cdo)
1638        return
1639    90010 format(
1640       1  /'Error: Input file ',a,
1641       2  /'       was generated with POSTMAX option',i3,'.',
1642       3  /'       Input file ',a,
1643       4  /'       was generated with POSTMAX option',i3,'.',
1644       5  /'       Can not mix X/Q and LPF*X/Q CDFs.',
1645       6 //'Execution terminated.')
1646    90020 format(
1647       1  /'Error: File ',a,' contains a Multi-Year CDF.',
1648       2  /'       Option 3 is for merging/sorting single-year CDFs',
1649       3  /'       previous generated by POSTMAX.',
1650       4 //'Execution terminated.')
1651    90100 format('Trace Section - Checking POSTMAX Parameters')
1652    90110 format('Searching file "',a,'" for string "Command Line" found ',
1653       1  'line:',/a,/'and extracted the value:',i4,/,80('-'))
```

```
      end
*=====================================================================
      subroutine open_multi_year_output
      include 'POSTMAX2.fh'
*---- local variables
      logical ex
      character answer*3, cline*100, opt3hdr*256, Weekday_Label(7)*3
      integer i, File_Status(13)
      integer GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
     &        GMT_Minute, GMT_Second, GMT_Weekday
*---- external functions
      integer iargc

*---- text for weekday values from 1 to 7
      data Weekday_Label / 'Sun','Mon','Tue','Wed','Thu','Fri','Sat'/

      if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
        open(out,file=cdofile(1))
        csvfile = cdofile(1)(1:index(cdofile(1),'.'))//'csv'
        open(csv,file=csvfile)
      else
*-----  if output file exists, ask user whether it should be overwritten
        inquire (file=outfile, exist=ex)
        if (ex) then
          write(crt,90020) trim(outfile)
          read(key,*) answer
          write(crt,*)
          if (answer.ne.'y'.and.answer.ne.'yes') then
            write(crt,90030)
            stop
          endif
        endif
        open(out,file=outfile)
        csvfile = outfile(1:index(outfile,'.'))//'csv'
        open(csv,file=csvfile)
*-----  echo command line
        opt3hdr='Command Line:'
        do i=0, iargc()
          call getarg(i, cline)
          opt3hdr=trim(opt3hdr)//' '//trim(cline)
        enddo
        write(crt,*)
        write(crt,'(a)') trim(opt3hdr)
        write(out,'(a)') trim(opt3hdr)
        write(csv,'(a)') trim(opt3hdr)
*-----  print postmax2.exe time stamp
        call Get_File_Status('POSTMAX2.exe', crt, File_Status,
     1       GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
     2       GMT_Minute, GMT_Second, GMT_Weekday)
        write(opt3hdr,92010) 'POSTMAX2.exe',
     1       Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
     2       GMT_Hour, GMT_Minute, GMT_Second
        write(crt,'(a)') trim(opt3hdr)
        write(out,'(a)') trim(opt3hdr)
        write(csv,'(a)') trim(opt3hdr)
*-----  print time stamps of input files
        do i=1,ncdofiles
```

```
1711            call Get_File_Status(cdofile(i), crt, File_Status,
1712           1         GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
1713           2         GMT_Minute, GMT_Second, GMT_Weekday)
1714            write(opt3hdr,92010) trim(cdofile(1)),
1715           1       Weekday_Label(GMT_Weekday), GMT_Month, GMT_Day, GMT_Year,
1716           2       GMT_Hour, GMT_Minute, GMT_Second
1717            write(crt,'(a)') trim(opt3hdr)
1718            write(out,'(a)') trim(opt3hdr)
1719            write(csv,'(a)') trim(opt3hdr)
1720          enddo
1721          write(crt,91020)
1722          write(out,91020)
1723          write(csv,91020)
1724         endif
1725
1726         do i=1,n_oldhdr
1727           write(out,'(a)') trim(oldhdr(i))
1728           write(csv,'(a)') trim(oldhdr(i))
1729         enddo
1730         do i=1,n_newhdr
1731           if (index(newhdr(i),'Met. File').gt.0) then
1732             write(newhdr(i)(13:14),'(i2)') nmetfiles
1733           endif
1734           write(out,'(a)') trim(newhdr(i))
1735           write(csv,'(a)') trim(newhdr(i))
1736         enddo
1737         write(out,91020)
1738         write(csv,91020)
1739
1740         if (abs(calc_type).eq.1.or.abs(calc_type).eq.2) then
1741           call print_column_headers(out,calc_type,2,1,1,0)
1742           call print_column_headers(csv,calc_type,2,1,1,1)
1743         else
1744           call print_column_headers(out,file_calc_type,2,1,1,0)
1745           call print_column_headers(csv,file_calc_type,2,1,1,1)
1746         endif
1747
1748         return
1749   90020 format(/'Warning: File ',a,' already exists - overwrite? (y/n) ',
1750        1  $)
1751   90030 format(/'Execution terminated.')
1752   91020 format(118('-'))
1753   92010 format('   File Info: ',a,' - ',a3,1x,i2,'/',i2.2,'/',i4,
1754        1       1x,i2,2(':',i2.2),' GMT')
1755         end
1756   *=====================================================================
1757         include 'Get_File_Status.for'
1758         include 'heapsort.for'
1759   *=====================================================================
1760   *==== End of file POSTMAX2.for
1761   *=====================================================================
1762
```

Common Variable Declarations

File POSTMAX2.FH contains all of the common variable declaration statements used in POSTMAX V2.0. The Fortran code in POSTMAX2.FH is included by the compiler command "include 'POSTMAX2.FH'" in POSTMAX2.FOR.

Note: The line numbers on the following pages are not part of the file.

```
*---- POSTMAX2.fh
      implicit none
      character*80
     1      bndfile, cdofile(6), csvfile, folder, infile, lpfecho,
     2      lpffile, metfile, outfile, summfile, tempfile, trcfile,
     3      wrnfile
      integer
     1      bnd, cdo, crt, csv, eco, in, key, lpf, met, out,
     2      summ, sum2, temp, trc, wrn
      integer
     1      n_newhdr, n_oldhdr, nmetfiles
      character*256
     1      newhdr(10), oldhdr(50)
      integer calc_type, file_calc_type, iter, ncdofiles, nmet,  numrad,
     1  winddir(365,24)
      real*8 bndry(16), pi, windspd(365,24)
      logical multiyear
      integer day, hour
      real*8 dist(35), glcon(35), sigma(35), xovrq(35)
      integer n_lpf
      real*8 lpftbl(25,17)
      integer day_time(52560,3,3), ptr(52560,3)
      real*8  cnsq(52560,3), cnsq95(3), meoi_dist(52560,3)
      character*1 user_abort

      common / filenames /
     1      bndfile, cdofile, csvfile, folder, infile, lpfecho,
     2      lpffile, metfile, outfile, summfile, tempfile, trcfile,
     3      wrnfile
      common / fileio /
     1      bnd, cdo, crt, csv, eco, in, key, lpf, met, out,
     2      summ, sum2, temp, trc, wrn
      common / filehdr1 /
     1      n_newhdr, n_oldhdr, nmetfiles
      common / filehdr2 /
     1      newhdr, oldhdr
      common / idata / calc_type, file_calc_type, iter, ncdofiles,
     1  nmet, numrad, winddir
      common / rdata / bndry, pi, windspd
      common / ldata / multiyear
      common / maccs / day, hour, dist, glcon, sigma, xovrq
      common / lpfac / lpftbl, n_lpf
      common / reslt / day_time, ptr, cnsq, cnsq95, meoi_dist
      common / user  / user_abort
*---- End of POSTMAX2.fh
```

Heapsort Subroutine

File HEAPSORT.FOR contains the heapsort subroutine used in POSTMAX V2.0. The Fortran code in HEAPSORT.FOR is included by the compiler command "include 'HEAPSORT.FOR'" in POSTMAX2.FOR.

Note: The line numbers on the following pages are not part of the file.

```
      subroutine heapsort(nsize,rval,ptr)
*----
*----   determines the ascending order of the input values.
*----   This version does not actually sort the input values
*----   but returns the sort  order in array ptr.
*----   The sort is performed using the heapsort algorithm.
*----
      implicit none
*---- input values
      integer nsize
*----         is the size of arrays rval and ptr
      real*8  rval(*)
*----         are the input values to be sorted.
*---- output values
      integer ptr(*)
*----         is the ascending order of the input values
*----
*---- The heapsort steps are:
*----
*---- 1) The original input values are arranged into an ordered binary
*----    tree (heap).
*----
*----    Binary tree structure of rval(i): In array rval, the first entry
*----    (i=1) is the root of the tree. It may have 2 children, rval(2)
*----    and rval(3). Position two of rval (i=2) may have 2 children,
*----    rval(4) and rval(5). In general, position i may have 2 children,
*----    located at 2i and 2i+1. Since the size of rval is finite,
*----    some positions or nodes will not have children. In an array
*----    of N values, the first trunc(N/2) will have children and
*----    the rest will not.
*----
*----    An ordered tree will have each parent larger (or equal to)
*----    each of its' children, making the root the largest value in
*----    the entire tree.
*----
*----    This step begins at the bottom of the tree and progresses up.
*----
*---- 2) The tree root is moved to the last (unsorted) position in rval
*----    (after saving the last unsorted value) because the root is the
*----    maximum value in the tree.
*----
*---- 3) The size of the unsorted array is decremented (because now the
*----    largest value is in the last position).
*----
*---- 4) Promote either rval(2) or rval(3) to the root position.
*----    Work down the promoted branch of the binary tree until a
*----    position is found for the temporarily saved value.
*----
*---- Code is based on Numerical Recipes (2nd Edition, p. 329) and
*---- work by Bruce C. Letellier (5/15/97).
*----
*---- Comments are based on http://www.cis.upenn.edu/~matuszek/
*---- cit594-2006/Lectures/25-heapsort.ppt
*----
*---- local variables
      integer child, i, n_unsorted, node, tempptr, topnode
      real*8 tempval
```

```
58
59          if (nsize.lt.2) return
60
61      *---- initialize pointer array
62          do i=1,nsize
63            ptr(i) = i
64          enddo
65
66      *---- initial the unsorted size to the entire input array
67          n_unsorted=nsize
68
69      *---- initialize the pointer for the current topnode
70      *---- to one more than the last node with children
71          topnode=nsize/2+1
72
73      *-------------------------------------------------------------------------
74      *---- top of outer loop
75      *-------------------------------------------------------------------------
76      10000 continue
77      *-----
78      *-----  Part 1
79      *-----
80            if (topnode.gt.1) then
81      *-------  outer loop will arrange the original input values
82      *-------  into a ordered binary tree
83
84      *-------  point to next (preceding) node
85               topnode=topnode-1
86      *-------  save the value for this node
87      *-------  if a child is larger than this topnode, the child will be
88      *-------  promoted and this value will be inserted lower in the tree
89               tempptr=ptr(topnode)
90            else
91      *-------  the entire tree is ordered
92      *-------  the largest value (of the unsorted part of the array)
93      *-------  is the tree, ie, in position 1
94
95      *-------  save the last entry in the unsorted array because the
96      *-------  largest (unsorted) value is about to be placed there
97               tempptr=ptr(n_unsorted)
98
99      *-------  move the tree root (which the maximum value in the tree
100     *-------  because the tree is ordered) to the last entry in the
101     *-------  unsorted array
102              ptr(n_unsorted)=ptr(1)
103
104     *-------  largest value has been sorted so decrement unsorted size
105              n_unsorted=n_unsorted-1
106
107              if (n_unsorted.eq.1) then
108     *---------  the sort is finished except for
109                ptr(1)=tempptr
110                return
111              endif
112           endif
113
114           tempval=rval(tempptr)
```

```
115
116   *-----------------------------------------------------------------------
117   *-----  Part 2
118   *-----  order the binary tree under topnode
119
120   *-----  initialize node pointer to the current top node
121           node=topnode
122
123   *-----------------------------------------------------------------------
124   *-----  top of inner loop
125   *-----------------------------------------------------------------------
126   20000   continue
127
128   *-------  set child pointer to the first child under node
129             child=2*node
130
131             if (child.le.n_unsorted) then
132   *---------  this node has at least one child, proceed with tree ordering
133               if (child.lt.n_unsorted) then
134   *-----------  this node has 2 children
135                 if (rval(ptr(child)).lt.rval(ptr(child+1))) then
136   *-------------  set the child pointer to the larger value child
137                   child=child+1
138                 endif
139               endif
140               if (tempval.lt.rval(ptr(child))) then
141   *-----------  tempval is too small to be inserted at this node
142   *-----------  promote the (larger) child to this node
143                 ptr(node)=ptr(child)
144   *-----------  move down tree to find location to insert tempval
145                 node=child
146               else
147   *-----------  tempval should be inserted at this node
148   *-----------  because it is larger than both children
149                 goto 30000
150               endif
151             else
152   *---------  this node does not have any children
153   *---------  this is the last possible node to insert tempval
154               goto 30000
155             endif
156           goto 20000
157   *-----------------------------------------------------------------------
158   *-----  bottom of inner loop
159   *-----------------------------------------------------------------------
160   30000   continue
161
162   *-----  since tempval is greater than the children values
163   *-----  at this node (or there are no children for this node)
164   *-----  insert tempval into the binary tree at this node;
165   *-----  ordering of tree under topnode is completed
166           ptr(node)=tempptr
167
168         goto 10000
169   *-----------------------------------------------------------------------
170   *---- bottom of outer loop
171   *-----------------------------------------------------------------------
```

```
172         end
173     *=====================================================================
174     *==== End of file heapsort.for
175     *=====================================================================
```

Get_File_Status Subroutine

File GET_FILE_STATUS.FOR contains the Get_File_Status subroutine used in POSTMAX V2.0. The Fortran code in GET_FILE_STATUS.FOR is included by the compiler command "include 'GET_FILE_STATUS.FOR'" in POSTMAX2.FOR.

Note: The line numbers on the following pages are not part of the file.

```
      subroutine Get_File_Status( filename, crt, File_Status,
     &           GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
     &           GMT_Minute, GMT_Second, GMT_Weekday)
      implicit none
*----
*---- input values
      character filename*(*)
*----           is the name of the file to status check
      integer   crt
*----           is the logical unit number for error message output
*----
*---- output values
      integer File_Status(13)
*----           1 device on which the file resides
*----           2 the serial number for the file (inode)
*----           3 file mode
*----           4 number of hard links to the file
*----           5 user ID of file owner
*----           6 group ID of file owner
*----           7 device identifier (devices only)
*----           8 size, in bytes, of file
*----           9 last file access time
*----           10 last file modify time
*----           11 last file status change time
*----           12 preferred block size for this file system
*----           13 actual number of blocks allocated
*----           Time values are number of seconds
*----           since 00:00:00 GMT January 1, 1970.
*----           Reference: Absoft Support Libraries, Aids to Porting
*----                      to/from UNIX,VAX/VMS, Absoft Corp., 2002
      integer GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
     &        GMT_Minute, GMT_Second, GMT_Weekday
*----           are the Greenwich Mean Time values for the file
*----
*---- local variables
      integer Return_Code
*----
*---- function declarations
      integer stat
*----           is a Absoft UNIX library function

      Return_Code = stat( filename, File_Status )

      if (Return_Code .ne. 0) then
        write(crt,99999) filename
        stop
      endif

      call Time_Convert( File_Status(10), GMT_Year, GMT_Month, GMT_Day,
     &           GMT_Hour, GMT_Minute, GMT_Second, GMT_Weekday )

      return
99999 format(/"Unexpected error with status check on file: ",a,
     1       /"Program terminated.")
      end
*=====================================================================
      subroutine Time_Convert( Total_Elapsed_Time_Seconds,
```

```
58        &           GMT_Year, GMT_Month, GMT_Day,
59        &           GMT_Hour, GMT_Minute, GMT_Second, GMT_Weekday )
60         implicit none
61    *---- Input Value = elapsed time, in seconds,
62    *----                since Jan. 1, 1970, 00:00:00 GMT
63         integer Total_Elapsed_Time_Seconds
64    *----
65    *---- Output Values = Greenwich Mean Time (GMT) year, month, day,
66    *----                  hour, minute, seconds, and weekday
67         integer GMT_Year, GMT_Month, GMT_Day, GMT_Hour,
68        &        GMT_Minute, GMT_Second, GMT_Weekday
69    *----
70    *---- local variables
71         integer Day_Per_Month(12), i,
72        &        Num_Cycle, Num_Year,
73        &        Sec_Accumulated_Month(12), Sec_Per_Cycle, Sec_Per_Day,
74        &        Sec_Per_Hour, Sec_Per_Minute, Sec_Per_Week, Sec_Per_Year
75
76         data Day_Per_Month / 31, 28, 31, 30, 31, 30,
77        &                     31, 31, 30, 31, 30, 31 /
78
79         Sec_Per_Minute =  60
80         Sec_Per_Hour   =  60 * Sec_Per_Minute
81         Sec_Per_Day    =  24 * Sec_Per_Hour
82         Sec_Per_Week   =   7 * Sec_Per_Day
83         Sec_Per_Year   = 365 * Sec_Per_Day
84         Sec_Per_Cycle  =   4 * Sec_Per_Year + Sec_Per_Day
85
86         Sec_Accumulated_Month(1) = 0
87         do i= 2, 12
88           Sec_Accumulated_Month(i) = Sec_Accumulated_Month(i-1)
89        &                           + Sec_Per_Day * Day_Per_Month(i-1)
90         enddo
91
92    *---- Calculate Weekday (Sun=1, Mon=2, etc.) from Total_Elapsed_Time_Seconds
93    *---- Since Jan. 1, 1970, is a Thursday, add 4 days (in seconds) to account
94    *---- for the prior Sun, Mon, Tue, and Wed.  (With this adjustment, the Sun
95    *---- before Jan. 1, 1970 begins at time 0 seconds.)
96         GMT_Weekday = 1 +
97        & mod( (Total_Elapsed_Time_Seconds + 4*Sec_Per_Day), Sec_Per_Week)
98        & / Sec_Per_Day
99
100   *---- initialize GMT_Second to Total_Elapsed_Time_Seconds
101        GMT_Second = Total_Elapsed_Time_Seconds
102
103   *---- calc number of 4-year cycles completed within the number of seconds
104        Num_Cycle = GMT_Second / Sec_Per_Cycle
105        GMT_Second = GMT_Second - Num_Cycle * Sec_Per_Cycle
106
107   *---- calc number of years completed within the remaining number of seconds
108        Num_Year = GMT_Second / Sec_Per_Year
109        GMT_Second = GMT_Second - Num_Year * Sec_Per_Year
110
111   *---- calc calendar year from baseline, number of 4-year cycles and years
112        GMT_Year = 1970 + 4*Num_Cycle + Num_Year
113
114   *---- calc number of months completed within the remaining number of seconds
```

```
      GMT_Month = 1
      do i = 12, 2, -1
        if (GMT_Second .ge. Sec_Accumulated_Month(i)) then
          GMT_Month = i
          GMT_Second = GMT_Second - Sec_Accumulated_Month(i)
        endif
      enddo

*---- calc number of days completed within the remaining number of seconds
      GMT_Day = GMT_Second / Sec_Per_Day
      GMT_Second = GMT_Second - GMT_Day * Sec_Per_Day
*---- increment number of days to get the date within the month
      GMT_Day = GMT_Day + 1

*---- Adjustment date if February 29th of the current 4-year cycle has passed
      if ((Num_Year .eq. 2 .and. GMT_Month .gt. 2)
     &    .or. (Num_Year .ge. 3)) then
         GMT_Day = GMT_Day - 1
         if (GMT_Day .eq. 0) then
           GMT_Month = GMT_Month - 1
           if (GMT_Month .eq. 0) then
             Num_Year = Num_Year - 1
             GMT_Month = 12
           endif
           if (GMT_Month .eq. 2 .and. Num_Year .eq. 2) then
             GMT_Day = 29
           else
             GMT_Day = Day_Per_Month(GMT_Month)
           endif
         endif
      endif

*---- calc number of hours completed within the remaining number of seconds
      GMT_Hour = GMT_Second / Sec_Per_Hour
      GMT_Second = GMT_Second - GMT_Hour * Sec_Per_Hour

*---- calc number of minutes completed within the remaining number of seconds
      GMT_Minute = GMT_Second / Sec_Per_Minute
      GMT_Second = GMT_Second - GMT_Minute * Sec_Per_Minute

      return
      end
*=====================================================================
      integer function Weekday( Year, Month, Day )
      implicit none
*---- Input Values:
      integer Year, Month, Day
*---- local variables
      integer Day_Accumulated_Month(12), Num_Cycle, Num_Day, Num_Year

      data Day_Accumulated_Month /   0,  31,  59,  90, 120, 151,
     &                             181, 212, 243, 273, 304, 334 /

*---- calc number of 4-year cycles and years since Jan. 1, 1970
      Num_Year  = Year - 1970
      Num_Cycle = Num_Year / 4
      Num_Year  = Num_Year - Num_Cycle * 4
```

```
172
173   *---- calc total number of days from Jan. 1, 1970 to date
174   *---- (Jan. 1, 1970 is Day 0.)
175         Num_Day   = Num_Cycle*(4*365+1) + Num_Year*365
176        &            + Day_Accumulated_Month(Month) + Day - 1
177         if (Num_Year .gt. 1) then
178   *----   the Feb. 29 of the current 4-year cycle has past
179           Num_Day = Num_Day + 1
180         endif
181
182   *---- Jan. 1, 1970, is Thursday
183   *---- add 4 days to account for the prior Sun, Mon, Tue, and Wed
184   *---- (Sunday before Jan. 1, 1970 is Consecutive Day 0.)
185         Num_Day = Num_Day + 4
186   *---- calc weekday using Sun=1
187         Weekday = mod( Num_Day, 7 ) + 1
188
189         return
190         end
191   *======================================================================
192   *---- End of File Get_File_Status.for
193   *======================================================================
```

COMPARE
Source Code

Note: The line numbers on the following pages are not part of the file.

```
      program COMPARE
      implicit none
      integer key, crt, lst, mrk, out, tst
      character*256 filename, mrkfile, tstfile
      character*256 mrkline, tstline
      logical caveat, ex

      data key, crt, lst, mrk, out, tst / 5, 6, 7, 8, 9, 10 /

      open(lst,file="filelist.txt")
      open(out,file="COMPARE.txt")
*-----------------------------------------------------------------------
10000 read(lst,'(a)',end=50000,err=72000) filename

      mrkfile = 'benchmark\'//filename
      inquire (file=mrkfile, exist=ex)
      if (.not.ex) then
        write(crt,'("File does not exist: ",a)') trim(mrkfile)
        write(crt,'("COMPARE aborted.")')
        write(out,'("File does not exist: ",a)') trim(mrkfile)
        write(out,'("COMPARE aborted.")')
        stop
      endif
      open(mrk,file=mrkfile,err=73000)

      tstfile = 'test\'//filename
      inquire (file=tstfile, exist=ex)
      if (.not.ex) then
        write(crt,'("File does not exist: ",a)') trim(tstfile)
        write(crt,'("COMPARE aborted.")')
        write(out,'("File does not exist: ",a)') trim(tstfile)
        write(out,'("COMPARE aborted.")')
        stop
      endif
      open(tst,file=tstfile,err=73000)

      caveat=.false.
*-----------------------------------------------------------------------
20000 read(mrk,'(a)',end=30000) mrkline
      read(tst,'(a)',end=75000) tstline
      if (index(filename,'POSTMAX_Summary').gt.0) then
        caveat=.true.
        if (mrkline(1:36).ne.tstline(1:36)) goto 76000
      else
        if (mrkline.ne.tstline) then
          if (index(mrkline,'File Info:').gt.0 .and.
     1        index(tstline,'File Info:').gt.0 .and.
     2        mrkline(1:index(mrkline,' - ')) .eq.
     3        tstline(1:index(tstline,' - '))
     4       ) then
            caveat=.true.
          else
            goto 76000
          endif
        endif
      endif
      goto 20000
```

```
*-----------------------------------------------------------------------
30000 read(tst,'(a)',end=40000) tstline
      goto 77000

40000 close(mrk)
      close(tst)
      write(crt,'("Compare complete: ",a)') trim(filename)
      write(out,'("Compare complete: ",a)') trim(filename)
      if (caveat) then
        write(crt,'(18x,"Time-stamp differences allowed.")')
        write(out,'(18x,"Time-stamp differences allowed.")')
      endif
      goto 10000
*-----------------------------------------------------------------------

50000 write(crt,'("End of file list.")')
      write(crt,'("Normal completion.")')
      write(out,'("End of file list.")')
      write(out,'("Normal completion.")')
      close(out)
      stop
72000 write(crt,'("Error reading filelist.txt")')
      write(out,'("Error reading filelist.txt")')
      stop
73000 write(crt,'("Error opening file: ",a)') trim(mrkfile)
      write(out,'("Error opening file: ",a)') trim(mrkfile)
      stop
74000 write(crt,'("Error opening file: ",a)') trim(tstfile)
      write(out,'("Error opening file: ",a)') trim(tstfile)
      stop
75000 write(crt,'("Benchmark file has more lines than test file.")')
      write(crt,'("File: ",a)') trim(mrkfile)
      write(crt,'("Line: ",a)') trim(mrkline)
      write(crt,'("COMPARE aborted.")')
      write(out,'("Benchmark file has more lines than test file.")')
      write(out,'("File: ",a)') trim(mrkfile)
      write(out,'("Line: ",a)') trim(mrkline)
      write(out,'("COMPARE aborted.")')
      stop
76000 write(crt,'("Lines do not match.")')
      write(crt,'("File: ",a)') trim(mrkfile)
      write(crt,'("Line: ",a)') trim(mrkline)
      write(crt,'("File: ",a)') trim(tstfile)
      write(crt,'("Line: ",a)') trim(tstline)
      write(crt,'("COMPARE aborted.")')
      write(out,'("Lines do not match.")')
      write(out,'("File: ",a)') trim(mrkfile)
      write(out,'("Line: ",a)') trim(mrkline)
      write(out,'("File: ",a)') trim(tstfile)
      write(out,'("Line: ",a)') trim(tstline)
      write(out,'("COMPARE aborted.")')
      stop
77000 write(crt,'("Test file has more lines than benchmark file.")')
      write(crt,'("File: ",a)') trim(tstfile)
      write(crt,'("Line: ",a)') trim(tstline)
      write(crt,'("COMPARE aborted.")')
      write(out,'("Test file has more lines than benchmark file.")')
```

```
115         write(out,'("File: ",a)') trim(tstfile)
116         write(out,'("Line: ",a)') trim(tstline)
117         write(out,'("COMPARE aborted.")')
118         stop
119         end
```

COMPARE Input File

Note: The line numbers on the following pages are not part of the file.

```
 1  01\Dir-01\Distances.bnd
 2  01\Dir-01\MACCS.cdo
 3  01\Dir-01\MACCS.csv
 4  01\Dir-01\MACCS.out
 5  01\Dir-01\Met_Data.csv
 6  01\Dir-01\Met_Data.inp
 7  01\Dir-01\POSTMAX_Summary.txt
 8  01\Dir-02\Distances.bnd
 9  01\Dir-02\MACCS.cdo
10  01\Dir-02\MACCS.csv
11  01\Dir-02\MACCS.out
12  01\Dir-02\Met_Data.csv
13  01\Dir-02\Met_Data.inp
14  01\Dir-02\POSTMAX_Summary.txt
15  01\Dir-03\Distances.bnd
16  01\Dir-03\MACCS.cdo
17  01\Dir-03\MACCS.csv
18  01\Dir-03\MACCS.out
19  01\Dir-03\Met_Data.csv
20  01\Dir-03\Met_Data.inp
21  01\Dir-03\POSTMAX_Summary.txt
22  01\Dir-04\Distances.bnd
23  01\Dir-04\MACCS.cdo
24  01\Dir-04\MACCS.csv
25  01\Dir-04\MACCS.out
26  01\Dir-04\Met_Data.csv
27  01\Dir-04\Met_Data.inp
28  01\Dir-04\POSTMAX_Summary.txt
29  01\Dir-05\Distances.bnd
30  01\Dir-05\MACCS.cdo
31  01\Dir-05\MACCS.csv
32  01\Dir-05\MACCS.out
33  01\Dir-05\Met_Data.csv
34  01\Dir-05\Met_Data.inp
35  01\Dir-05\POSTMAX_Summary.txt
36  01\Dir-06\Distances.bnd
37  01\Dir-06\MACCS.cdo
38  01\Dir-06\MACCS.csv
39  01\Dir-06\MACCS.out
40  01\Dir-06\Met_Data.csv
41  01\Dir-06\Met_Data.inp
42  01\Dir-06\POSTMAX_Summary.txt
43  01\Dir-07\Distances.bnd
44  01\Dir-07\MACCS.cdo
45  01\Dir-07\MACCS.csv
46  01\Dir-07\MACCS.out
47  01\Dir-07\Met_Data.csv
48  01\Dir-07\Met_Data.inp
49  01\Dir-07\POSTMAX_Summary.txt
50  01\Dir-08\Distances.bnd
51  01\Dir-08\MACCS.cdo
52  01\Dir-08\MACCS.csv
53  01\Dir-08\MACCS.out
54  01\Dir-08\Met_Data.csv
55  01\Dir-08\Met_Data.inp
56  01\Dir-08\POSTMAX_Summary.txt
57  01\Dir-09\Distances.bnd
```

```
01\Dir-09\MACCS.cdo
01\Dir-09\MACCS.csv
01\Dir-09\MACCS.out
01\Dir-09\Met_Data.csv
01\Dir-09\Met_Data.inp
01\Dir-09\POSTMAX_Summary.txt
01\Dir-10\Distances.bnd
01\Dir-10\MACCS.cdo
01\Dir-10\MACCS.csv
01\Dir-10\MACCS.out
01\Dir-10\Met_Data.csv
01\Dir-10\Met_Data.inp
01\Dir-10\POSTMAX_Summary.txt
01\Dir-11\Distances.bnd
01\Dir-11\MACCS.cdo
01\Dir-11\MACCS.csv
01\Dir-11\MACCS.out
01\Dir-11\Met_Data.csv
01\Dir-11\Met_Data.inp
01\Dir-11\POSTMAX_Summary.txt
01\Dir-12\Distances.bnd
01\Dir-12\MACCS.cdo
01\Dir-12\MACCS.csv
01\Dir-12\MACCS.out
01\Dir-12\Met_Data.csv
01\Dir-12\Met_Data.inp
01\Dir-12\POSTMAX_Summary.txt
01\Dir-13\Distances.bnd
01\Dir-13\MACCS.cdo
01\Dir-13\MACCS.csv
01\Dir-13\MACCS.out
01\Dir-13\Met_Data.csv
01\Dir-13\Met_Data.inp
01\Dir-13\POSTMAX_Summary.txt
01\Dir-14\Distances.bnd
01\Dir-14\MACCS.cdo
01\Dir-14\MACCS.csv
01\Dir-14\MACCS.out
01\Dir-14\Met_Data.csv
01\Dir-14\Met_Data.inp
01\Dir-14\POSTMAX_Summary.txt
01\Dir-15\Distances.bnd
01\Dir-15\MACCS.cdo
01\Dir-15\MACCS.csv
01\Dir-15\MACCS.out
01\Dir-15\Met_Data.csv
01\Dir-15\Met_Data.inp
01\Dir-15\POSTMAX_Summary.txt
01\Dir-16\Distances.bnd
01\Dir-16\MACCS.cdo
01\Dir-16\MACCS.csv
01\Dir-16\MACCS.out
01\Dir-16\Met_Data.csv
01\Dir-16\Met_Data.inp
01\Dir-16\POSTMAX_Summary.txt
02\Dir-01\Distances.bnd
02\Dir-01\MACCS.cdo
```

```
115   02\Dir-01\MACCS.csv
116   02\Dir-01\MACCS.out
117   02\Dir-01\Met_Data.csv
118   02\Dir-01\Met_Data.inp
119   02\Dir-01\POSTMAX_Summary.txt
120   02\Dir-02\Distances.bnd
121   02\Dir-02\MACCS.cdo
122   02\Dir-02\MACCS.csv
123   02\Dir-02\MACCS.out
124   02\Dir-02\Met_Data.csv
125   02\Dir-02\Met_Data.inp
126   02\Dir-02\POSTMAX_Summary.txt
127   02\Dir-03\Distances.bnd
128   02\Dir-03\MACCS.cdo
129   02\Dir-03\MACCS.csv
130   02\Dir-03\MACCS.out
131   02\Dir-03\Met_Data.csv
132   02\Dir-03\Met_Data.inp
133   02\Dir-03\POSTMAX_Summary.txt
134   02\Dir-04\Distances.bnd
135   02\Dir-04\MACCS.cdo
136   02\Dir-04\MACCS.csv
137   02\Dir-04\MACCS.out
138   02\Dir-04\Met_Data.csv
139   02\Dir-04\Met_Data.inp
140   02\Dir-04\POSTMAX_Summary.txt
141   02\Dir-05\Distances.bnd
142   02\Dir-05\MACCS.cdo
143   02\Dir-05\MACCS.csv
144   02\Dir-05\MACCS.out
145   02\Dir-05\Met_Data.csv
146   02\Dir-05\Met_Data.inp
147   02\Dir-05\POSTMAX_Summary.txt
148   02\Dir-06\Distances.bnd
149   02\Dir-06\MACCS.cdo
150   02\Dir-06\MACCS.csv
151   02\Dir-06\MACCS.out
152   02\Dir-06\Met_Data.csv
153   02\Dir-06\Met_Data.inp
154   02\Dir-06\POSTMAX_Summary.txt
155   02\Dir-07\Distances.bnd
156   02\Dir-07\MACCS.cdo
157   02\Dir-07\MACCS.csv
158   02\Dir-07\MACCS.out
159   02\Dir-07\Met_Data.csv
160   02\Dir-07\Met_Data.inp
161   02\Dir-07\POSTMAX_Summary.txt
162   02\Dir-08\Distances.bnd
163   02\Dir-08\MACCS.cdo
164   02\Dir-08\MACCS.csv
165   02\Dir-08\MACCS.out
166   02\Dir-08\Met_Data.csv
167   02\Dir-08\Met_Data.inp
168   02\Dir-08\POSTMAX_Summary.txt
169   02\Dir-09\Distances.bnd
170   02\Dir-09\MACCS.cdo
171   02\Dir-09\MACCS.csv
```

```
172   02\Dir-09\MACCS.out
173   02\Dir-09\Met_Data.csv
174   02\Dir-09\Met_Data.inp
175   02\Dir-09\POSTMAX_Summary.txt
176   02\Dir-10\Distances.bnd
177   02\Dir-10\MACCS.cdo
178   02\Dir-10\MACCS.csv
179   02\Dir-10\MACCS.out
180   02\Dir-10\Met_Data.csv
181   02\Dir-10\Met_Data.inp
182   02\Dir-10\POSTMAX_Summary.txt
183   02\Dir-11\Distances.bnd
184   02\Dir-11\MACCS.cdo
185   02\Dir-11\MACCS.csv
186   02\Dir-11\MACCS.out
187   02\Dir-11\Met_Data.csv
188   02\Dir-11\Met_Data.inp
189   02\Dir-11\POSTMAX_Summary.txt
190   02\Dir-12\Distances.bnd
191   02\Dir-12\MACCS.cdo
192   02\Dir-12\MACCS.csv
193   02\Dir-12\MACCS.out
194   02\Dir-12\Met_Data.csv
195   02\Dir-12\Met_Data.inp
196   02\Dir-12\POSTMAX_Summary.txt
197   02\Dir-13\Distances.bnd
198   02\Dir-13\MACCS.cdo
199   02\Dir-13\MACCS.csv
200   02\Dir-13\MACCS.out
201   02\Dir-13\Met_Data.csv
202   02\Dir-13\Met_Data.inp
203   02\Dir-13\POSTMAX_Summary.txt
204   02\Dir-14\Distances.bnd
205   02\Dir-14\MACCS.cdo
206   02\Dir-14\MACCS.csv
207   02\Dir-14\MACCS.out
208   02\Dir-14\Met_Data.csv
209   02\Dir-14\Met_Data.inp
210   02\Dir-14\POSTMAX_Summary.txt
211   02\Dir-15\Distances.bnd
212   02\Dir-15\MACCS.cdo
213   02\Dir-15\MACCS.csv
214   02\Dir-15\MACCS.out
215   02\Dir-15\Met_Data.csv
216   02\Dir-15\Met_Data.inp
217   02\Dir-15\POSTMAX_Summary.txt
218   02\Dir-16\Distances.bnd
219   02\Dir-16\MACCS.cdo
220   02\Dir-16\MACCS.csv
221   02\Dir-16\MACCS.out
222   02\Dir-16\Met_Data.csv
223   02\Dir-16\Met_Data.inp
224   02\Dir-16\POSTMAX_Summary.txt
225   03\Dir-01\Distances.bnd
226   03\Dir-01\MACCS.cdo
227   03\Dir-01\MACCS.csv
228   03\Dir-01\MACCS.out
```

```
229   03\Dir-01\Met_Data.csv
230   03\Dir-01\Met_Data.inp
231   03\Dir-01\POSTMAX_Summary.txt
232   03\Dir-02\Distances.bnd
233   03\Dir-02\MACCS.cdo
234   03\Dir-02\MACCS.csv
235   03\Dir-02\MACCS.out
236   03\Dir-02\Met_Data.csv
237   03\Dir-02\Met_Data.inp
238   03\Dir-02\POSTMAX_Summary.txt
239   03\Dir-03\Distances.bnd
240   03\Dir-03\MACCS.cdo
241   03\Dir-03\MACCS.csv
242   03\Dir-03\MACCS.out
243   03\Dir-03\Met_Data.csv
244   03\Dir-03\Met_Data.inp
245   03\Dir-03\POSTMAX_Summary.txt
246   03\Dir-04\Distances.bnd
247   03\Dir-04\MACCS.cdo
248   03\Dir-04\MACCS.csv
249   03\Dir-04\MACCS.out
250   03\Dir-04\Met_Data.csv
251   03\Dir-04\Met_Data.inp
252   03\Dir-04\POSTMAX_Summary.txt
253   03\Dir-05\Distances.bnd
254   03\Dir-05\MACCS.cdo
255   03\Dir-05\MACCS.csv
256   03\Dir-05\MACCS.out
257   03\Dir-05\Met_Data.csv
258   03\Dir-05\Met_Data.inp
259   03\Dir-05\POSTMAX_Summary.txt
260   03\Dir-06\Distances.bnd
261   03\Dir-06\MACCS.cdo
262   03\Dir-06\MACCS.csv
263   03\Dir-06\MACCS.out
264   03\Dir-06\Met_Data.csv
265   03\Dir-06\Met_Data.inp
266   03\Dir-06\POSTMAX_Summary.txt
267   03\Dir-07\Distances.bnd
268   03\Dir-07\MACCS.cdo
269   03\Dir-07\MACCS.csv
270   03\Dir-07\MACCS.out
271   03\Dir-07\Met_Data.csv
272   03\Dir-07\Met_Data.inp
273   03\Dir-07\POSTMAX_Summary.txt
274   03\Dir-08\Distances.bnd
275   03\Dir-08\MACCS.cdo
276   03\Dir-08\MACCS.csv
277   03\Dir-08\MACCS.out
278   03\Dir-08\Met_Data.csv
279   03\Dir-08\Met_Data.inp
280   03\Dir-08\POSTMAX_Summary.txt
281   03\Dir-09\Distances.bnd
282   03\Dir-09\MACCS.cdo
283   03\Dir-09\MACCS.csv
284   03\Dir-09\MACCS.out
285   03\Dir-09\Met_Data.csv
```

```
286   03\Dir-09\Met_Data.inp
287   03\Dir-09\POSTMAX_Summary.txt
288   03\Dir-10\Distances.bnd
289   03\Dir-10\MACCS.cdo
290   03\Dir-10\MACCS.csv
291   03\Dir-10\MACCS.out
292   03\Dir-10\Met_Data.csv
293   03\Dir-10\Met_Data.inp
294   03\Dir-10\POSTMAX_Summary.txt
295   03\Dir-11\Distances.bnd
296   03\Dir-11\MACCS.cdo
297   03\Dir-11\MACCS.csv
298   03\Dir-11\MACCS.out
299   03\Dir-11\Met_Data.csv
300   03\Dir-11\Met_Data.inp
301   03\Dir-11\POSTMAX_Summary.txt
302   03\Dir-12\Distances.bnd
303   03\Dir-12\MACCS.cdo
304   03\Dir-12\MACCS.csv
305   03\Dir-12\MACCS.out
306   03\Dir-12\Met_Data.csv
307   03\Dir-12\Met_Data.inp
308   03\Dir-12\POSTMAX_Summary.txt
309   03\Dir-13\Distances.bnd
310   03\Dir-13\MACCS.cdo
311   03\Dir-13\MACCS.csv
312   03\Dir-13\MACCS.out
313   03\Dir-13\Met_Data.csv
314   03\Dir-13\Met_Data.inp
315   03\Dir-13\POSTMAX_Summary.txt
316   03\Dir-14\Distances.bnd
317   03\Dir-14\MACCS.cdo
318   03\Dir-14\MACCS.csv
319   03\Dir-14\MACCS.out
320   03\Dir-14\Met_Data.csv
321   03\Dir-14\Met_Data.inp
322   03\Dir-14\POSTMAX_Summary.txt
323   03\Dir-15\Distances.bnd
324   03\Dir-15\MACCS.cdo
325   03\Dir-15\MACCS.csv
326   03\Dir-15\MACCS.out
327   03\Dir-15\Met_Data.csv
328   03\Dir-15\Met_Data.inp
329   03\Dir-15\POSTMAX_Summary.txt
330   03\Dir-16\Distances.bnd
331   03\Dir-16\MACCS.cdo
332   03\Dir-16\MACCS.csv
333   03\Dir-16\MACCS.out
334   03\Dir-16\Met_Data.csv
335   03\Dir-16\Met_Data.inp
336   03\Dir-16\POSTMAX_Summary.txt
337   04\Dir-01\Distances.bnd
338   04\Dir-01\MACCS.cdo
339   04\Dir-01\MACCS.csv
340   04\Dir-01\MACCS.out
341   04\Dir-01\MACCS.wrn
342   04\Dir-01\Met_Data.csv
```

```
343   04\Dir-01\Met_Data.inp
344   04\Dir-01\POSTMAX_Summary.txt
345   04\Dir-02\Distances.bnd
346   04\Dir-02\MACCS.cdo
347   04\Dir-02\MACCS.csv
348   04\Dir-02\MACCS.out
349   04\Dir-02\MACCS.wrn
350   04\Dir-02\Met_Data.csv
351   04\Dir-02\Met_Data.inp
352   04\Dir-02\POSTMAX_Summary.txt
353   04\Dir-03\Distances.bnd
354   04\Dir-03\MACCS.cdo
355   04\Dir-03\MACCS.csv
356   04\Dir-03\MACCS.out
357   04\Dir-03\MACCS.wrn
358   04\Dir-03\Met_Data.csv
359   04\Dir-03\Met_Data.inp
360   04\Dir-03\POSTMAX_Summary.txt
361   04\Dir-04\Distances.bnd
362   04\Dir-04\MACCS.cdo
363   04\Dir-04\MACCS.csv
364   04\Dir-04\MACCS.out
365   04\Dir-04\MACCS.wrn
366   04\Dir-04\Met_Data.csv
367   04\Dir-04\Met_Data.inp
368   04\Dir-04\POSTMAX_Summary.txt
369   04\Dir-05\Distances.bnd
370   04\Dir-05\MACCS.cdo
371   04\Dir-05\MACCS.csv
372   04\Dir-05\MACCS.out
373   04\Dir-05\MACCS.wrn
374   04\Dir-05\Met_Data.csv
375   04\Dir-05\Met_Data.inp
376   04\Dir-05\POSTMAX_Summary.txt
377   04\Dir-06\Distances.bnd
378   04\Dir-06\MACCS.cdo
379   04\Dir-06\MACCS.csv
380   04\Dir-06\MACCS.out
381   04\Dir-06\MACCS.wrn
382   04\Dir-06\Met_Data.csv
383   04\Dir-06\Met_Data.inp
384   04\Dir-06\POSTMAX_Summary.txt
385   04\Dir-07\Distances.bnd
386   04\Dir-07\MACCS.cdo
387   04\Dir-07\MACCS.csv
388   04\Dir-07\MACCS.out
389   04\Dir-07\MACCS.wrn
390   04\Dir-07\Met_Data.csv
391   04\Dir-07\Met_Data.inp
392   04\Dir-07\POSTMAX_Summary.txt
393   04\Dir-08\Distances.bnd
394   04\Dir-08\MACCS.cdo
395   04\Dir-08\MACCS.csv
396   04\Dir-08\MACCS.out
397   04\Dir-08\MACCS.wrn
398   04\Dir-08\Met_Data.csv
399   04\Dir-08\Met_Data.inp
```

```
400   04\Dir-08\POSTMAX_Summary.txt
401   04\Dir-09\Distances.bnd
402   04\Dir-09\MACCS.cdo
403   04\Dir-09\MACCS.csv
404   04\Dir-09\MACCS.out
405   04\Dir-09\MACCS.wrn
406   04\Dir-09\Met_Data.csv
407   04\Dir-09\Met_Data.inp
408   04\Dir-09\POSTMAX_Summary.txt
409   04\Dir-10\Distances.bnd
410   04\Dir-10\MACCS.cdo
411   04\Dir-10\MACCS.csv
412   04\Dir-10\MACCS.out
413   04\Dir-10\MACCS.wrn
414   04\Dir-10\Met_Data.csv
415   04\Dir-10\Met_Data.inp
416   04\Dir-10\POSTMAX_Summary.txt
417   04\Dir-11\Distances.bnd
418   04\Dir-11\MACCS.cdo
419   04\Dir-11\MACCS.csv
420   04\Dir-11\MACCS.out
421   04\Dir-11\MACCS.wrn
422   04\Dir-11\Met_Data.csv
423   04\Dir-11\Met_Data.inp
424   04\Dir-11\POSTMAX_Summary.txt
425   04\Dir-12\Distances.bnd
426   04\Dir-12\MACCS.cdo
427   04\Dir-12\MACCS.csv
428   04\Dir-12\MACCS.out
429   04\Dir-12\MACCS.wrn
430   04\Dir-12\Met_Data.csv
431   04\Dir-12\Met_Data.inp
432   04\Dir-12\POSTMAX_Summary.txt
433   04\Dir-13\Distances.bnd
434   04\Dir-13\MACCS.cdo
435   04\Dir-13\MACCS.csv
436   04\Dir-13\MACCS.out
437   04\Dir-13\MACCS.wrn
438   04\Dir-13\Met_Data.csv
439   04\Dir-13\Met_Data.inp
440   04\Dir-13\POSTMAX_Summary.txt
441   04\Dir-14\Distances.bnd
442   04\Dir-14\MACCS.cdo
443   04\Dir-14\MACCS.csv
444   04\Dir-14\MACCS.out
445   04\Dir-14\MACCS.wrn
446   04\Dir-14\Met_Data.csv
447   04\Dir-14\Met_Data.inp
448   04\Dir-14\POSTMAX_Summary.txt
449   04\Dir-15\Distances.bnd
450   04\Dir-15\MACCS.cdo
451   04\Dir-15\MACCS.csv
452   04\Dir-15\MACCS.out
453   04\Dir-15\MACCS.wrn
454   04\Dir-15\Met_Data.csv
455   04\Dir-15\Met_Data.inp
456   04\Dir-15\POSTMAX_Summary.txt
```

```
457   04\Dir-16\Distances.bnd
458   04\Dir-16\MACCS.cdo
459   04\Dir-16\MACCS.csv
460   04\Dir-16\MACCS.out
461   04\Dir-16\MACCS.wrn
462   04\Dir-16\Met_Data.csv
463   04\Dir-16\Met_Data.inp
464   04\Dir-16\POSTMAX_Summary.txt
465   05\Dir-01\Distances.bnd
466   05\Dir-01\MACCS.cdo
467   05\Dir-01\MACCS.csv
468   05\Dir-01\MACCS.out
469   05\Dir-01\Met_Data.csv
470   05\Dir-01\Met_Data.inp
471   05\Dir-01\POSTMAX_Summary.txt
472   05\Dir-02\Distances.bnd
473   05\Dir-02\MACCS.cdo
474   05\Dir-02\MACCS.csv
475   05\Dir-02\MACCS.out
476   05\Dir-02\Met_Data.csv
477   05\Dir-02\Met_Data.inp
478   05\Dir-02\POSTMAX_Summary.txt
479   05\Dir-03\Distances.bnd
480   05\Dir-03\MACCS.cdo
481   05\Dir-03\MACCS.csv
482   05\Dir-03\MACCS.out
483   05\Dir-03\Met_Data.csv
484   05\Dir-03\Met_Data.inp
485   05\Dir-03\POSTMAX_Summary.txt
486   05\Dir-04\Distances.bnd
487   05\Dir-04\MACCS.cdo
488   05\Dir-04\MACCS.csv
489   05\Dir-04\MACCS.out
490   05\Dir-04\Met_Data.csv
491   05\Dir-04\Met_Data.inp
492   05\Dir-04\POSTMAX_Summary.txt
493   05\Dir-05\Distances.bnd
494   05\Dir-05\MACCS.cdo
495   05\Dir-05\MACCS.csv
496   05\Dir-05\MACCS.out
497   05\Dir-05\Met_Data.csv
498   05\Dir-05\Met_Data.inp
499   05\Dir-05\POSTMAX_Summary.txt
500   05\Dir-06\Distances.bnd
501   05\Dir-06\MACCS.cdo
502   05\Dir-06\MACCS.csv
503   05\Dir-06\MACCS.out
504   05\Dir-06\Met_Data.csv
505   05\Dir-06\Met_Data.inp
506   05\Dir-06\POSTMAX_Summary.txt
507   05\Dir-07\Distances.bnd
508   05\Dir-07\MACCS.cdo
509   05\Dir-07\MACCS.csv
510   05\Dir-07\MACCS.out
511   05\Dir-07\Met_Data.csv
512   05\Dir-07\Met_Data.inp
513   05\Dir-07\POSTMAX_Summary.txt
```

```
514   05\Dir-08\Distances.bnd
515   05\Dir-08\MACCS.cdo
516   05\Dir-08\MACCS.csv
517   05\Dir-08\MACCS.out
518   05\Dir-08\Met_Data.csv
519   05\Dir-08\Met_Data.inp
520   05\Dir-08\POSTMAX_Summary.txt
521   05\Dir-09\Distances.bnd
522   05\Dir-09\MACCS.cdo
523   05\Dir-09\MACCS.csv
524   05\Dir-09\MACCS.out
525   05\Dir-09\Met_Data.csv
526   05\Dir-09\Met_Data.inp
527   05\Dir-09\POSTMAX_Summary.txt
528   05\Dir-10\Distances.bnd
529   05\Dir-10\MACCS.cdo
530   05\Dir-10\MACCS.csv
531   05\Dir-10\MACCS.out
532   05\Dir-10\Met_Data.csv
533   05\Dir-10\Met_Data.inp
534   05\Dir-10\POSTMAX_Summary.txt
535   05\Dir-11\Distances.bnd
536   05\Dir-11\MACCS.cdo
537   05\Dir-11\MACCS.csv
538   05\Dir-11\MACCS.out
539   05\Dir-11\Met_Data.csv
540   05\Dir-11\Met_Data.inp
541   05\Dir-11\POSTMAX_Summary.txt
542   05\Dir-12\Distances.bnd
543   05\Dir-12\MACCS.cdo
544   05\Dir-12\MACCS.csv
545   05\Dir-12\MACCS.out
546   05\Dir-12\Met_Data.csv
547   05\Dir-12\Met_Data.inp
548   05\Dir-12\POSTMAX_Summary.txt
549   05\Dir-13\Distances.bnd
550   05\Dir-13\MACCS.cdo
551   05\Dir-13\MACCS.csv
552   05\Dir-13\MACCS.out
553   05\Dir-13\Met_Data.csv
554   05\Dir-13\Met_Data.inp
555   05\Dir-13\POSTMAX_Summary.txt
556   05\Dir-14\Distances.bnd
557   05\Dir-14\MACCS.cdo
558   05\Dir-14\MACCS.csv
559   05\Dir-14\MACCS.out
560   05\Dir-14\Met_Data.csv
561   05\Dir-14\Met_Data.inp
562   05\Dir-14\POSTMAX_Summary.txt
563   05\Dir-15\Distances.bnd
564   05\Dir-15\MACCS.cdo
565   05\Dir-15\MACCS.csv
566   05\Dir-15\MACCS.out
567   05\Dir-15\Met_Data.csv
568   05\Dir-15\Met_Data.inp
569   05\Dir-15\POSTMAX_Summary.txt
570   05\Dir-16\Distances.bnd
```

```
571   05\Dir-16\MACCS.cdo
572   05\Dir-16\MACCS.csv
573   05\Dir-16\MACCS.out
574   05\Dir-16\MACCS.wrn
575   05\Dir-16\Met_Data.csv
576   05\Dir-16\Met_Data.inp
577   05\Dir-16\POSTMAX_Summary.txt
578   06\ATMOS1.cdo
579   06\ATMOS1.csv
580   06\ATMOS1.OUT
581   06\CNTLBND.BND
582   06\POSTMAX_Summary.txt
583   06\ta6_91.csv
584   06\TA6_91.INP
585   07\ATMOS1.cdo
586   07\ATMOS1.csv
587   07\ATMOS1.OUT
588   07\ATMOS1.wrn
589   07\CNTLBND.BND
590   07\POSTMAX_Summary.txt
591   07\ta6_94.csv
592   07\TA6_94.INP
593   08\ATMOS1.cdo
594   08\ATMOS1.csv
595   08\ATMOS1.OUT
596   08\ATMOS2.cdo
597   08\ATMOS2.csv
598   08\ATMOS2.OUT
599   08\ATMOS3.cdo
600   08\ATMOS3.csv
601   08\ATMOS3.OUT
602   08\ATMOS3.wrn
603   08\ATMOS4.cdo
604   08\ATMOS4.csv
605   08\ATMOS4.OUT
606   08\ATMOS4.wrn
607   08\ATMOS5.cdo
608   08\ATMOS5.csv
609   08\ATMOS5.OUT
610   08\ATMOS5.wrn
611   08\ATMOS6.cdo
612   08\ATMOS6.csv
613   08\ATMOS6.OUT
614   08\ATMOS6.wrn
615   08\CNTLBND.BND
616   08\Cumulatv.cdo
617   08\Cumulatv.csv
618   08\POSTMAX_Summary.txt
619   08\ta6_91.csv
620   08\TA6_91.INP
621   08\ta6_92.csv
622   08\TA6_92.INP
623   08\ta6_93.csv
624   08\TA6_93.INP
625   08\ta6_94.csv
626   08\TA6_94.INP
627   08\ta6_95.csv
```

```
628   08\TA6_95.INP
629   08\ta6_96.csv
630   08\TA6_96.INP
631   09\Cumulatv.cdo
632   09\Cumulatv.csv
633   09\POSTMAX_Summary.txt
634   10\Distance1.bnd
635   10\Distance2.bnd
636   10\MACCS.cdo
637   10\MACCS.csv
638   10\MACCS.out
639   10\MACCS_Old.cdo
640   10\Met_Data.csv
641   10\Met_Data.inp
642   10\POSTMAX_Summary.txt
643   10\POSTMAX_Summary_Old.txt
644   11\Distances.bnd
645   11\LPF_Tbl.txt
646   11\MACCS.cdo
647   11\MACCS.csv
648   11\MACCS.lpf
649   11\MACCS.out
650   11\Met_Data.csv
651   11\Met_Data.inp
652   11\POSTMAX_Summary.txt
653   12\Cumulatv.cdo
654   12\Cumulatv.csv
655   12\Distances.bnd
656   12\LPF_Tbl.txt
657   12\MACCS_1.cdo
658   12\MACCS_1.csv
659   12\MACCS_1.lpf
660   12\MACCS_1.out
661   12\MACCS_2.cdo
662   12\MACCS_2.csv
663   12\MACCS_2.lpf
664   12\MACCS_2.out
665   12\MACCS_3.cdo
666   12\MACCS_3.csv
667   12\MACCS_3.lpf
668   12\MACCS_3.out
669   12\MACCS_4.cdo
670   12\MACCS_4.csv
671   12\MACCS_4.lpf
672   12\MACCS_4.out
673   12\MACCS_5.cdo
674   12\MACCS_5.csv
675   12\MACCS_5.lpf
676   12\MACCS_5.out
677   12\MACCS_6.cdo
678   12\MACCS_6.csv
679   12\MACCS_6.lpf
680   12\MACCS_6.out
681   12\Met_Data_1.csv
682   12\Met_Data_1.inp
683   12\Met_Data_2.csv
684   12\Met_Data_2.inp
```

```
685   12\Met_Data_3.csv
686   12\Met_Data_3.inp
687   12\Met_Data_4.csv
688   12\Met_Data_4.inp
689   12\Met_Data_5.csv
690   12\Met_Data_5.inp
691   12\Met_Data_6.csv
692   12\Met_Data_6.inp
693   12\POSTMAX_Summary.txt
694   13\Cumulatv.cdo
695   13\Cumulatv.csv
696   13\POSTMAX_Summary.txt
697   E1\bnd.bnd
698   E1\Case_1a.txt
699   E1\Case_1b.txt
700   E1\Case_1c.txt
701   E1\Case_1d.txt
702   E1\Case_1e.txt
703   E1\Case_1f.txt
704   E1\Case_1g.txt
705   E1\Case_1h.txt
706   E1\Case_1i.txt
707   E1\Case_1j.txt
708   E1\inp.out
709   E1\lpf.txt
710   E2\bnd.bnd
711   E2\Case_2a.txt
712   E2\Case_2b.txt
713   E2\Case_2c.txt
714   E2\Case_2d.txt
715   E2\Case_2e.txt
716   E2\Case_2f.txt
717   E2\Case_2g.txt
718   E2\Case_2h.txt
719   E2\Case_2i.txt
720   E2\Case_2j.txt
721   E2\inp.out
722   E2\lpf.txt
723   E3\bnd.bnd
724   E3\Case_3a.cdo
725   E3\Case_3a.csv
726   E3\Case_3a.out
727   E3\Case_3a.txt
728   E3\Case_3b.cdo
729   E3\Case_3b.csv
730   E3\Case_3b.lpf
731   E3\Case_3b.out
732   E3\Case_3b.txt
733   E3\Case_3c.cdo
734   E3\Case_3c.csv
735   E3\Case_3c.out
736   E3\Case_3c.txt
737   E3\Case_3d.cdo
738   E3\Case_3d.csv
739   E3\Case_3d.lpf
740   E3\Case_3d.out
741   E3\Case_3d.txt
```

```
742   E3\Case_3e.cdo
743   E3\Case_3e.csv
744   E3\Case_3e.out
745   E3\Case_3e.txt
746   E3\Case_3f.cdo
747   E3\Case_3f.csv
748   E3\Case_3f.lpf
749   E3\Case_3f.out
750   E3\Case_3f.txt
751   E3\Case_3g.cdo
752   E3\Case_3g.csv
753   E3\Case_3g.out
754   E3\Case_3g.txt
755   E3\Case_3h.cdo
756   E3\Case_3h.csv
757   E3\Case_3h.lpf
758   E3\Case_3h.out
759   E3\Case_3h.txt
760   E3\Case_3i.cdo
761   E3\Case_3i.csv
762   E3\Case_3i.out
763   E3\Case_3i.txt
764   E3\Case_3j.cdo
765   E3\Case_3j.csv
766   E3\Case_3j.lpf
767   E3\Case_3j.out
768   E3\Case_3j.txt
769   E3\Case_3k.cdo
770   E3\Case_3k.csv
771   E3\Case_3k.out
772   E3\Case_3k.txt
773   E3\Case_3l.cdo
774   E3\Case_3l.csv
775   E3\Case_3l.lpf
776   E3\Case_3l.out
777   E3\Case_3l.txt
778   E3\Case_3m.cdo
779   E3\Case_3m.csv
780   E3\Case_3m.out
781   E3\Case_3m.txt
782   E3\Case_3n.cdo
783   E3\Case_3n.csv
784   E3\Case_3n.lpf
785   E3\Case_3n.out
786   E3\Case_3n.txt
787   E3\Case_3o.cdo
788   E3\Case_3o.csv
789   E3\Case_3o.out
790   E3\Case_3o.txt
791   E3\Case_3p.cdo
792   E3\Case_3p.csv
793   E3\Case_3p.lpf
794   E3\Case_3p.out
795   E3\Case_3p.txt
796   E3\lpf.txt
797   E3\Met_Data.csv
798   E3\Met_Data.inp
```

```
799   E4\Case_4a.cdo
800   E4\Case_4a.csv
801   E4\Case_4a.out
802   E4\Case_4a.txt
803   E4\Case_4b.cdo
804   E4\Case_4b.csv
805   E4\Case_4b.lpf
806   E4\Case_4b.out
807   E4\Case_4b.txt
808   E4\Case_4c.cdo
809   E4\Case_4c.csv
810   E4\Case_4c.out
811   E4\Case_4c.txt
812   E4\Case_4d.cdo
813   E4\Case_4d.csv
814   E4\Case_4d.lpf
815   E4\Case_4d.out
816   E4\Case_4d.txt
817   E4\Case_4e.out
818   E4\Case_4e.txt
819   E4\Case_4f.cdo
820   E4\Case_4f.csv
821   E4\Case_4f.lpf
822   E4\Case_4f.out
823   E4\Case_4f.txt
824   E4\Case_4g.cdo
825   E4\Case_4g.csv
826   E4\Case_4g.out
827   E4\Case_4g.txt
828   E4\Case_4h.cdo
829   E4\Case_4h.csv
830   E4\Case_4h.lpf
831   E4\Case_4h.out
832   E4\Case_4h.txt
833   E4\lpf.txt
834   E4\Met_Data.inp
835   E4\neg_val.bnd
836   E4\one_line.bnd
837   E4\ten_lines.bnd
838   E4\zero_val.bnd
839   E5\bnd.bnd
840   E5\Case_5a.cdo
841   E5\Case_5a.csv
842   E5\Case_5a.lpf
843   E5\Case_5a.out
844   E5\Case_5a.txt
845   E5\Case_5b.cdo
846   E5\Case_5b.csv
847   E5\Case_5b.lpf
848   E5\Case_5b.out
849   E5\Case_5b.txt
850   E5\Case_5c.cdo
851   E5\Case_5c.csv
852   E5\Case_5c.lpf
853   E5\Case_5c.out
854   E5\Case_5c.txt
855   E5\Case_5d.cdo
```

```
856   E5\Case_5d.csv
857   E5\Case_5d.lpf
858   E5\Case_5d.out
859   E5\Case_5d.txt
860   E5\Case_5e.out
861   E5\Case_5e.txt
862   E5\Case_5f.cdo
863   E5\Case_5f.csv
864   E5\Case_5f.lpf
865   E5\Case_5f.out
866   E5\Case_5f.txt
867   E5\Case_5g.cdo
868   E5\Case_5g.csv
869   E5\Case_5g.lpf
870   E5\Case_5g.out
871   E5\Case_5g.txt
872   E5\Case_5h.cdo
873   E5\Case_5h.csv
874   E5\Case_5h.lpf
875   E5\Case_5h.out
876   E5\Case_5h.txt
877   E5\decr_spd.txt
878   E5\high_spd.txt
879   E5\incomplete.txt
880   E5\lpf.txt
881   E5\neg_lpf.txt
882   E5\neg_spd.txt
883   E5\one_line.txt
884   E5\two_lines.txt
885   E5\zero_lpf.txt
886   E5\zero_spd.txt
887
```