ESL-TR-92-04/02

CALIBRATING DOE-2 TO WEATHER AND NON-WEATHER-DEPENDENT LOADS FOR A COMMERCIAL BUILDING, VOLUME 2: DATA PROCESSING ROUTINES TO CALIBRATE A DOE-2 MODEL

Written by:
John Douglas Bronson

May 1992

Table of Contents

DATA PROCESSING ROUTINES

Extracting Columnar Data from DOE-2 Hourly Output Reports

DOE-2 yields hourly data on specific variables provided the user specifies the HOURLY-REPORT instruction. Analyzing the simulation results with hourly data gives a more detailed picture of how well the model is predicting the monitored energy consumption. The difficulties of using hourly data to calibrate a model are the extraction of data from DOE-2's well documented output reports and processing the data into graphs which are meaningful. This chapter demonstrates the data processing routines that extract hourly end-use energy consumption and weather data from DOE-2's hourly output reports and process the data into three-dimensional plots and temperature-specific humidity carpet plots.

Figure 1 is an outline of the methodology to process the hourly data into single column data files with a date stamp. Often, to expedite efficient execution of a parametric study the PARAMETRIC-INPUT instruction is used, requiring additional data processing of the DOE-2 output report. The output report from parametric runs consists of several simulation outputs appended together into one file. The SEP-PARA.AWK (SEParate PARAmetric runs) routine (Flag A.2 in Figure 1) divides the output report so that the hourly output from each parametric run is in a separate file. The command line for SEP-PARA.AWK routine is:

nawk -f sep-para.awk <source> <parameter> <output> <# of parametric runs>

<source> is the name of the DOE-2 output report containing the output from the parametric runs.

<parameter> is the sub-program name in which the parametric study was performed. If a parametric study was performed on a LOADS parameter, <parameter> is LDL. Likewise, <parameter> is SDL for a SYSTEMS parametric run and PDL for a PLANT parametric run.

Note: The flags ("A1", "A.2", etc.) appended to the blocks representing the routines and data files in the diagrams label the figure numbers containing examples of the data files and hard copies of the routines. They are also used as references in the text. As an example, the block representing the DOE-2 SIMULATION OUTPUT in Figure 1 is flagged with the flag "A.1".

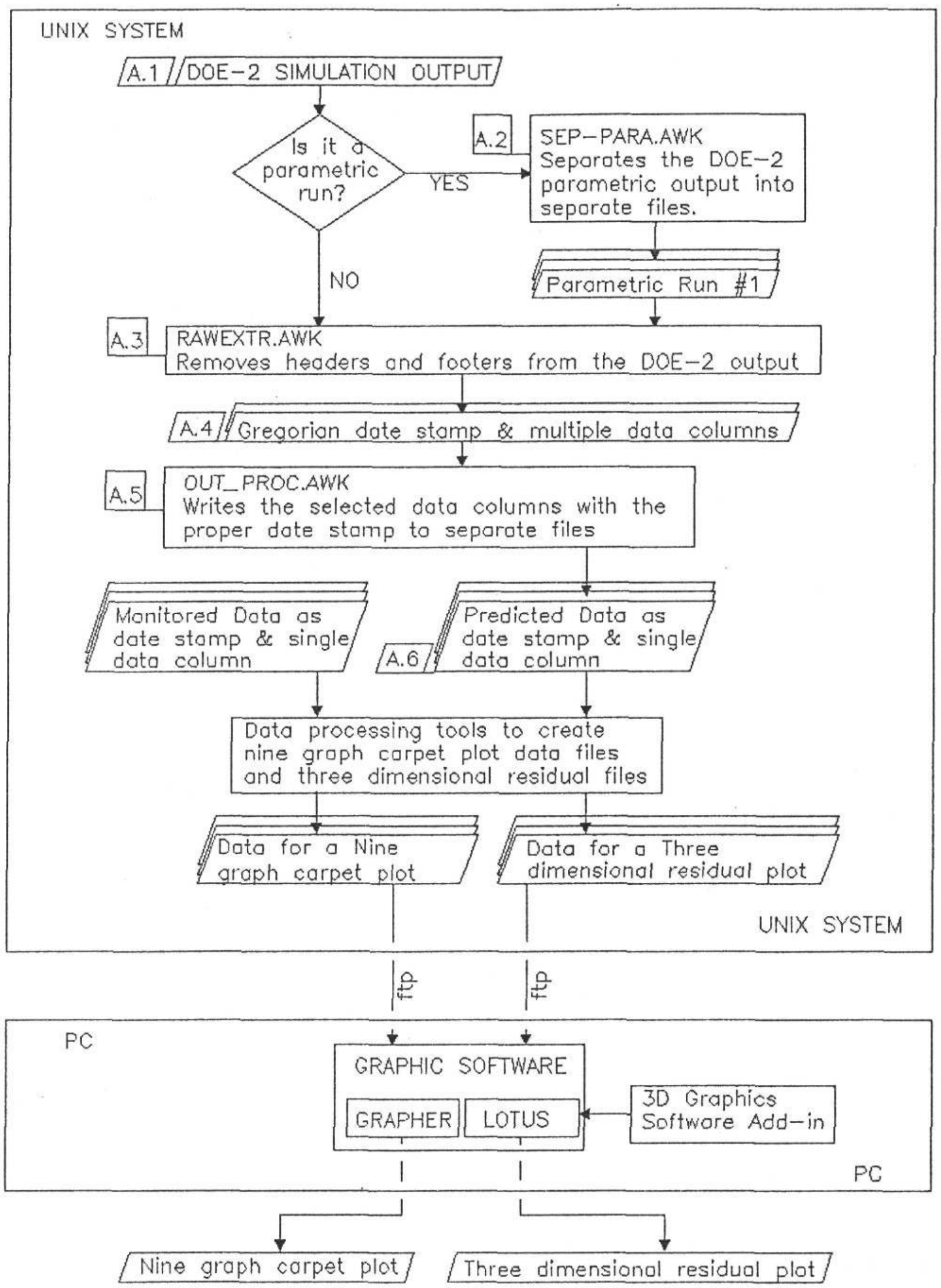

Figure 1 **Processing DOE-2 Output** -- Flow diagram for processing DOE-2 hourly output reports into data files with a date stamp and a single data column. If necessary, a routine separates the output from a parametric study. Other routines create contiguous data files by removing the columnar headers and footers in the DOE-2 output report. These hourly data files are used to create the comparative plots. The flag labels of "A.1", "A.2", etc. in the diagram denotes the figure number containing an example of the data file or routine.

<output> is a unique name given to the files containing the separated parametric run output. Output from the first parameter run will be printed to a file called <output>"-1.out" and the output from the second parameter run will printed to a file called <output>-2.out", etc..

<# of parametric runs> is the number of parametric run performed by the simulation.

DOE-2's hourly output reports contain formatted column headers, footers, and summaries which make them easily to read. The column headers, footers, and summaries need to be removed so that the hourly data is a contiguous columnar data file and ready to be processed into graphical data files. The RAWEXTR.AWK (RAW data EXTRaction) routine (Flag A.3) extracts the hourly data from between the columnar headers and footers, creating a contiguous multi-column hourly data output file. The command line for the RAWEXTR.AWK routine is:

nawk -f rawextr.awk <source> <report1> <report2> <report3> <report4>

<source> is the name of the DOE-2 output file.

<report1>, <report2>, etc. are the U-names of the HOURLY-REPORT in the sub-programs. If hourly data needs to be extracted from less than four HOURLY-REPORTs, unused **<reports?>** need to be filled in with an ambiguous names, i.e., "XXXX".

Example:
nawk -f rawextr.awk Fig-A1.txt HR-3 XXXX XXX XX

Fig-A.1.txt is a two day segment of a DOE-2 hourly output report; shown in Figure A.1. The RAWEXTR.AWK routine generates an output file called "HR-3.out" which is Figure A.4 or Fig-A4.txt. Depending on the number of variables the user requested in the HOURLY-REPORT instruction, the continuous hourly data output file contains an equal number of hourly data columns plus a Gregorian date stamp without a year designator. This output file requires further processing to generate the graphical data files.

The next routine, OUT_PROC.AWK (OUTput PROCessed data) (Flag A.5), calculates a decimal date from the Gregorian date stamp, removes any daylight savings shift in the data, and writes the desired columns of hourly data to separate single data column files. The single data column files can be given either the LoanSTAR date stamp (Figure A.7) or only the decimal date (Flag A.6). Hourly data monitored in the LoanSTAR project is given a seven field date stamp show in Table 1 which is placed at the beginning of each hourly data record. The command line for the OUT_PROC.AWK routine is:

TABLE 1 The LoanSTAR Date Stamp

Field #	Description	Format	Example
1	Site Number	XXX	001
2	Month	MM	09
3	Day-of-the-Month	DD	01
4	Year	YY	89
5	Julian Date	YYDDD	89244
6	Decimal Date	XXXX.XXXX	3531.5833
7	Hour-of-the-Day	HH	14

nawk -f out_proc.awk <source> <stamp> <yr> <begins> <ends> <col> <factor> <output>

<source>	is the contiguous multi-column hourly data file generated by RAWEXTR.AWK.
<stamp>	"0" or "1" to indicate the desired date stamp - "1" for the LoanSTAR date stamp - "2" for only the Decimal date.
<yr>	the year that the data represents, i.e., "89".
<begins>	Julian Date for the first day of daylight savings - First Sunday in April.
<ends>	Julian Date for the last day of daylight savings - Last Sunday in October.
<col>	is the data column's position from the farthest right column.
<factor>	is a conversion factor that the data column is multiplied by.
<output>	is output file name for the single data column.

Example:

```
nawk -f out_proc.awk   HR-3.out   2   89   91   302   2   0.000001   cl.dat
```

The hourly cooling load data is copied to a single column data file with only the decimal date date stamp; "cl.dat" is generated which is Figure A.6 or Fig-A.6.txt. To extract more than one column from the contiguous multi-column data file, repeat **<col> <factor> <output>** columns as many times necessary.

OUT_PROC.AWK routine processes the DOE-2 hourly data into a format identical to the monitored data. Identical format of the predicted and monitored data files facilitates the creation of subsequent data processing tools that handled all the data files without concern for their origin. With the hourly data processed into a contiguous file and with the desired date stamp, other routines are used to create the three-dimensional residual plots and the temperature-specific humidity carpet plots.

Three-Dimensional Residual Plots

The methodology to create the three-dimensional residual plots is outlined in Figure 2. Data for the three-dimensional residual plots are the difference between the monitored (Flag A.8) and the predicted (Flag A.9) whole-building electricity consumption data. The RESIDUAL.C routine (creates RESIDUAL data files, Flag A.10) generates these data files The command line for RESIDUAL.C routine is:

residual <predicted> <monitored> <positive> <negative>

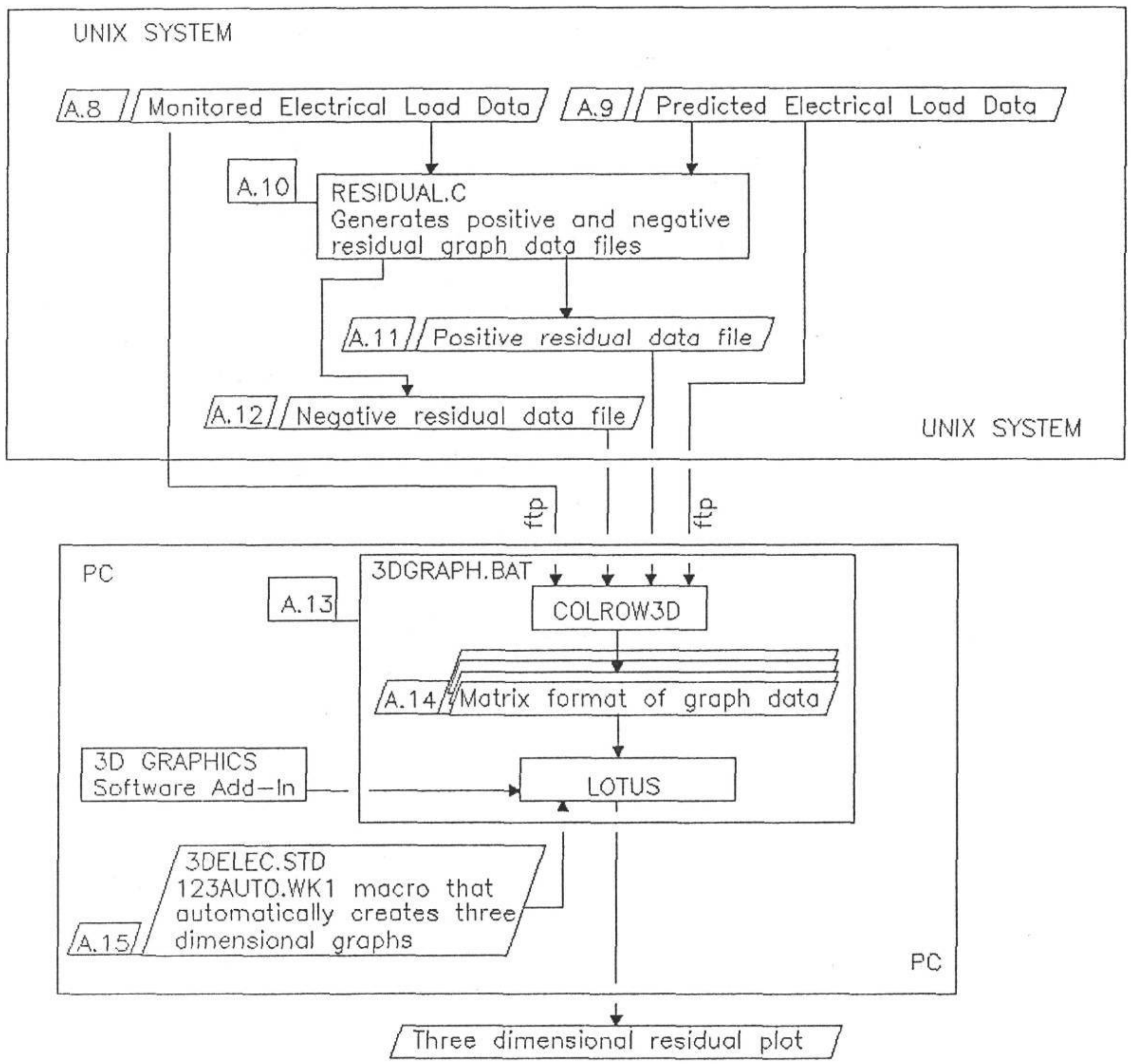

Figure 2 **Creating a Three-Dimensional Residual Plot** -- Flow diagram for producing a three-dimensional residual plot. Residual data files are generated on the UNIX system and then file transferred (ftp) to the PC along with the hourly monitored and predicted electricity consumption data. On the PC the data files are processed into a matrix where they are then imported into LOTUS which, together with the 3DGRAPHIC Add-In package and a LOTUS instruction macro, generates the four *.PIC files that create the plot.

<predicted> is name of the predicted whole-building electricity consumption, (Flag A.8).

<monitored> is name of the monitored whole-building electricity consumption, (Flag A.9).

<positive> is name of the outfile for the positive residual data file.

<negative> is name of the outfile for the negative residual data file.

Example:

residual Fig-A9.txt Fig-A8.txt p-resid.dat n-resid.dat

Two residual data files are generated; "p-resid.dat" and "n-resid.dat" are the positive residual data file (Figure A.11) and a negative residual data file (Figure A.12), respectively. The values in the positive residual graph data file represent the hourly occurrences when the model overestimates whole-building electricity consumption. In a like manner, the values in the negative residual graph data file represent the hourly occurrences when the model underestimates whole-building electricity consumption.

The residual graph data files are created on the UNIX system. The two residual graph data files and the monitored and predicted whole-building electrical load data files are then transferred to the PC. A batch file, 3DGRAPH.BAT (Flag A.13), processes each graph data file into a 3D graph using the COLROW3D program and the LOTUS spreadsheet package with the Intex Solutions 3D-Graphics Add-In software to create three-dimensional graphs. The command line for the 3DGRAPH.BAT routine is:

3dgraph <source> <0 or 1> <macro name>

<source> is the name for the hourly columnar data file without its *.dat extension.

<0 or 1> Choose "0" to process one year's worth of data, filling in missing data if necessary. Choose "1" to generate the smallest n x 24 matrix containing the input data.

<macro name> name of the 123AUTO.WK1 macro without its *.std extension.

Example:

3dgraph Fig-A8 1 3delec

COLROW3D converts columnar data into a matrix format (Flag A.14) which in the format required by the Intex Solutions 3D-Graphics software. The matrix consists of 25 columns. The first column is the day-of-the-year followed by 24 columns for each hour-of-the-day. For a non-leap year, the matrix can have up to 366 rows; one row for each day of the year present in the data file and one row along the top for graphic labels. The data matrix file is imported into LOTUS and three-dimensional graphs are automatically generated using a three-dimensional add-in package, 3D GRAPHICS, and a 123AUTO.WK1 macro (Flag A.15). The 123AUTO.WK1 macro performs the keystrokes necessary to retrieve a matrix file and select, label, and scale the data ranges to generate a three-dimensional graph.

Temperature-Specific Humidity Carpet Plots

The data files used to create the temperature-specific humidity carpet plots originate from five hourly data files extracted from the DOE-2 simulation output (Flag A.9 and Flags A.16 to A.19). Figure 3 outlines the required data processing to create the data files for the temperature-specific humidity carpet plot. First, the hourly outdoor dry bulb temperature and specific humidity data are used by DOE-WEA.C (DOE-2 WEAther, Flag A.20) to create a single weather data file containing the decimal date, outdoor dry bulb temperature, specific humidity, and relative humidity. The command line for DOE-WEA.C routine is:

doe-wea <pressure> <temperature> <humidity> <output>

<pressure> is the building's standard atmospheric pressure (psia).
<temperature> is the hourly dry bulb temperature data file.
<humidity> is the hourly specific humidity data file.
<outfile> is the name of the outfile.

Example:

doe-wea 14.55 Fig-A16.txt Fig-A17.txt weather.dat

The DOE-WEA.C routine generates a data file called "weather.dat" which is Figure A.21. Next, the hourly energy load data files are merged with the weather data file by ADD-LOAD.C (ADD LOADs, Flag A.22) to create a single graph data file containing weather and energy consumption data (Flag A.23). The command line for ADD-LOAD.C routine is:

add-load <weather> <cooling> <heating> <electricity> <output>

<weather> is the hourly weather data file generated by DOE-WEA.C
<cooling> is the hourly chilled water energy consumption data file.
<heating> is the hourly hot water energy consumption data file.
<electricity> is the hourly whole-building electricity energy consumption data file.
<outfile> is the name of the outfile.

Example:

add-load weather.dat Fig-A18.txt Fig-A19.txt Fig-A9.txt wea-lds.dat

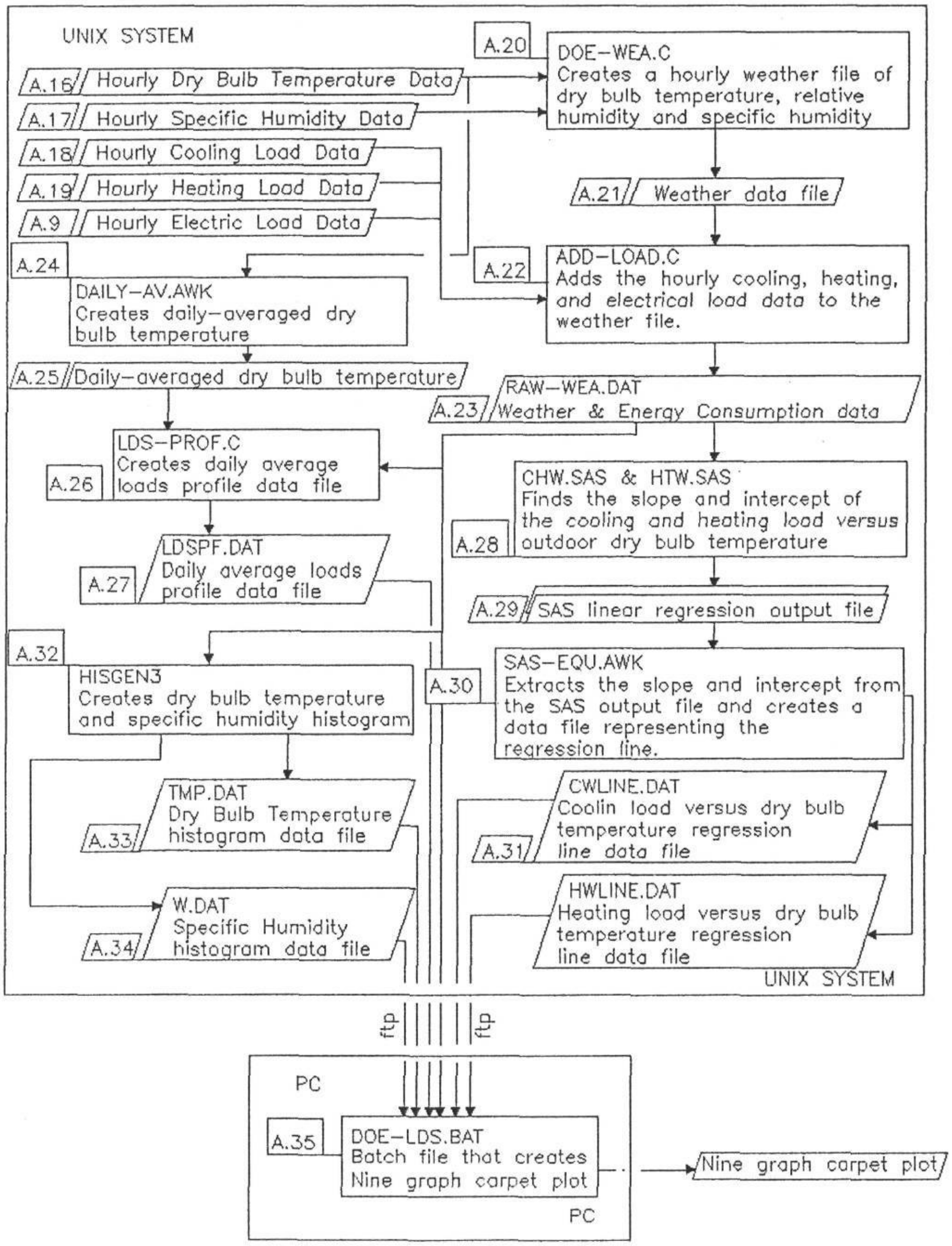

Figure 3 **Creating a Temperature-Specific Humidity Carpet Plot** -- Flow diagram for producing a temperature-specific humidity carpet plot. The graph data files of hourly weather data and energy consumption for the carpet plot are generated on the UNIX system and then transferred (ftp) to the PC. On the PC a batch routine calls GRAPHER to generate the graph files, and then copies, scales, and positions the graphs in the proper order on the page.

The ADD-LOAD.C routine generates a weather and energy consumption data file called "wea-lds.dat" which is Figure A.23. This weather and energy consumption data file is used to create five of the graphs in temperature-specific humidity carpet plot. The file contains the hourly data used to plot ambient conditions on a psychrometric chart template, to create a cooling and heating load versus outdoor dry bulb temperature graph, and cooling and heating load versus outdoor specific humidity graph.

The weather and energy consumption data file is also used to create the data files for the other four graphs in the carpet. The file is used to create the outdoor dry-bulb temperature and specific humidity histogram data files (Flag A.33 and Flag A.34), the daily average load profile data file (Flag A.27) and linear regression models of chilled water and hot water consumption versus outdoor dry-bulb temperature (A.29). The SAS statistical software programs, CHW.SAS and HTW.SAS (Flag A.28), determine the slope and the y-intercept of the predicted chilled water and hot water consumption versus outdoor dry bulb temperature. The SAS programs look for a data file called "input.dat", therefore the weather and energy consumption data file needs to be copied to "input.dat" before executing the SAS programs. The command lines for the CHW.SAS and HTW.SAS routines are:

copy wea-lds.dat input.dat

sas **chw**.sas

sas **htw**.sas

Output from CHW.SAS and HTW.SAS routines are files named "chwsas.out" (Figure A.29) and "htwsas.out" respectively.

The SAS-EQU.AWK routine (SAS EQUations, Flag A.30) extracts the slope and y-intercept values from the SAS output files and uses them to generate two data files representing the linear regression models of cooling and heating load versus dry bulb temperature. The command line for the SAS-EQU.AWK routine is:

nawk -f sas-equ.awk <sas output> <output>

<sas output> is the output file from CHW.SAS and HTW.SAS routines.

<output> is the name of the output data file.

Example:

nawk -f sas-equ.awk Fig-A29.txt cw-line.dat

The SAS-EQU.AWK routine generates a graph data file called "cwline.dat" which is Figure A.31. These lines reflect the central tendency of the predicted chilled and hot water consumption versus dry bulb temperature and are plotted in the cooling and heating load versus dry bulb temperature graph as solid lines labeled CW Model and HW Model.

The daily average energy load profile graphs require daily-averaged dry bulb temperature to sort the hourly energy consumption according to daily-averaged temperature less than or greater than 60 °F (Flag A.25). DAILY-AV.AWK (creating DAILY AVerage data from hourly data, Flag A.24) generates a daily-averaged dry bulb temperature data file from the hourly data. The command line for the DAILY-AV.AWK routine is:

nawk -f daily-av.awk <source> <output>

<source> is the hourly dry bulb temperature data file.

<output> is name of the created daily average data file.

Example:

nawk -f daily-av.awk Fig-A16.txt db.ave

The DAILY-AV.AWK routines generates a daily-averaged data file called "db.ave" which is Figure A.25. LDS-PROF.C (Flag A.26) uses the daily-averaged dry bulb temperature data file and the weather and energy consumption data file (Flag A.21) to generate daily-average load profiles. Daily-average weekday and weekend load profiles are generated from the hourly cooling, heating, and electricity load data for days with a daily-averaged temperatures above and below 60 °F. The command line for the LDS-PROF.C routine is:

lds-prof <daily-average> <source> <outfile>

<daily-average> is the daily average dry bulb temperature file.

<source> is the weather and energy consumption data file.

<outfile> is the name of the outfile data file.

Example:

lds-prof db.ave wea-lds.dat ldspf.dat

The LDS-PROF.C routine generates a graph data file called ldspf.dat which is Figure A.27.

The graph data files for the two histograms are generated using the routine HISGEN3 (Flag A.32). The command line for the HISGEN3 routine is:

nawk -f hisgen3 <source>

<source> is the weather and energy consumption data file.

Example:

nawk -f hisgen3 wea-lds.dat

HISGEN3 generates two output files, "tmp.dat" (Figure A.33) and "w.dat" (Figure A.34), which are the graph data files for the temperature and specific humidity histograms, respectively.

In all, six graph data files are required to create the temperature-specific humidity carpet plot. The carpet plots are created using the GRAPHER graphics software. The six data graphs files are transferred from the UNIX system to the PC. The six graph data files need to be copied to the file names which are expected by the graphing batch routine DOE-LDS.BAT (DOE-2 LoaDS graphing routine Figure A.35) and in the resident directory of the GRAPHER graph data files (*.grf) and scaling and positioning files (*.tem). The weather and energy consumption data file from ADD-LOAD.C (wea-lds.dat) needs to be copied to **raw-wea.dat**. The graph data files representing the cooling and heating load linear regression models generated by SAS-EQU.AWK need to be copy to **cwline.dat** and **hwline.dat**, respectively. The daily average load profiles data file from LDS-PROF.C needs to be copied to **ldspf.dat**. The two histogram data files, **tmp.dat** and **w.dat**, do not need to be renamed. DOE-LDS.BAT generates the GRAPHER files, copies, scales, and positions the graphs in proper order to create a temperature-specific humidity carpet plot. The command line for the DOE-LDS.BAT routine is:

doe-lds <date 1> <date 2>

<date 1> and <date 2> are the Gregorian or similar date stamp that represents the beginning and ending dates of the data. These dates are the X-axis title of all the time series graphs. An AWK routine called CHANGEX (Figure A.36) edits the *.grf time series files to include the correct X-axis title.

Example:

doe-lds Sept-1/1989 Sept-2/1989

DOE-LDS.BAT generates a postscript output file called DOE-LDS.OUT. Figure A.37 shows the proper combination of graph data files and GRAPHER files to create each graph in the plot.

'PACKING' SITE MONITORED WEATHER DATA INTO TRY

A methodology to pack site monitored weather data into an existing TRY (Test Reference Year) weather file has been developed using a FORTRAN program called LS2TRY.FOR (LoanSTAR weather data into a Test Reference Year weather file) and the DOE-2 weather packer developed by Lawrence Berkeley Laboratory. Site monitored weather data in the format shown in Table 1 is processed into a binary file usable by the DOE-2 program. The methodology is outlined in Figure 1. In Figure 1, the blocks representing the data files and programs in the methodology are appended with flags to identify the figure numbers containing examples of the data files and hard copies of the programs.

A command procedure (PACKWEATHER.COM) was written to perform the necessary steps to pack the site monitored weather data. The command procedure does file management, then executes LS2TRY.FOR, and finally calls the DOE-2 weather packer. The DOE-2 weather packer resides on the University VAX, therefore it's used to pack the site monitored weather data. The command line to pack monitored weather data is:

@PACKWEATHER **<Monitored Data> <Base TRY> <LS2TRY Instruction> <Packer Instruction>**

<Monitored Data>	is the site-monitored weather data file (B.3).
<Base TRY>	is the base TRY weather file (B.4). The base TRY weather file is in the ASCII format and contains one-whole year of data.
<LS2TRY Instruction>	is the program instruction file (B.2) for LS2TRY.FOR. The instruction file gives the methodology the flexibility to process weather data from different weather sites.
<Packer Instruction>	is the DOE-2 weather packer instruction file (B.5) .

Example:

@PACKWEATHER Fig-B3.txt Fig-B4.txt Fig-B2.txt Fig-B5.txt

where Fig-B3.txt is the two day example the monitored data file shown in Figure B.3.

To pack site monitored weather data, these four data files and PACKWEATHER.COM need to be in the same directory. To complete a successful run, the Monitored Data file needs to have several attributes. The Monitored Data file, **<Monitored Data>**, needs to be in columnar format with a 1-24 hour time stamp. The monitored data file can be as small as one hourly data, record but no larger than one year (365 days). The data needs to be in sequence from the earliest time stamp to the latest; based on the Month, Day-of-the-Month, Hour-of-the-Day fields. The

TABLE 1 Format of the ZEC Monitored Weather Data File

Field #	Description	Format
1	Station Number	XXXXX
2	Month	MM
3	Day-of-the-Month	MM
4	Year	YY
5	Julian Date	YYDDD
6	Decimal Date	XXXX.XXXX
7	Hour-of-the-Day	HHMM
8	Relative Humidity	[%]
9	Dry Bulb Temperature	[F]
10	Total Horizontal Insolation	[W/m^2]
11	Wind Speed	[mph]

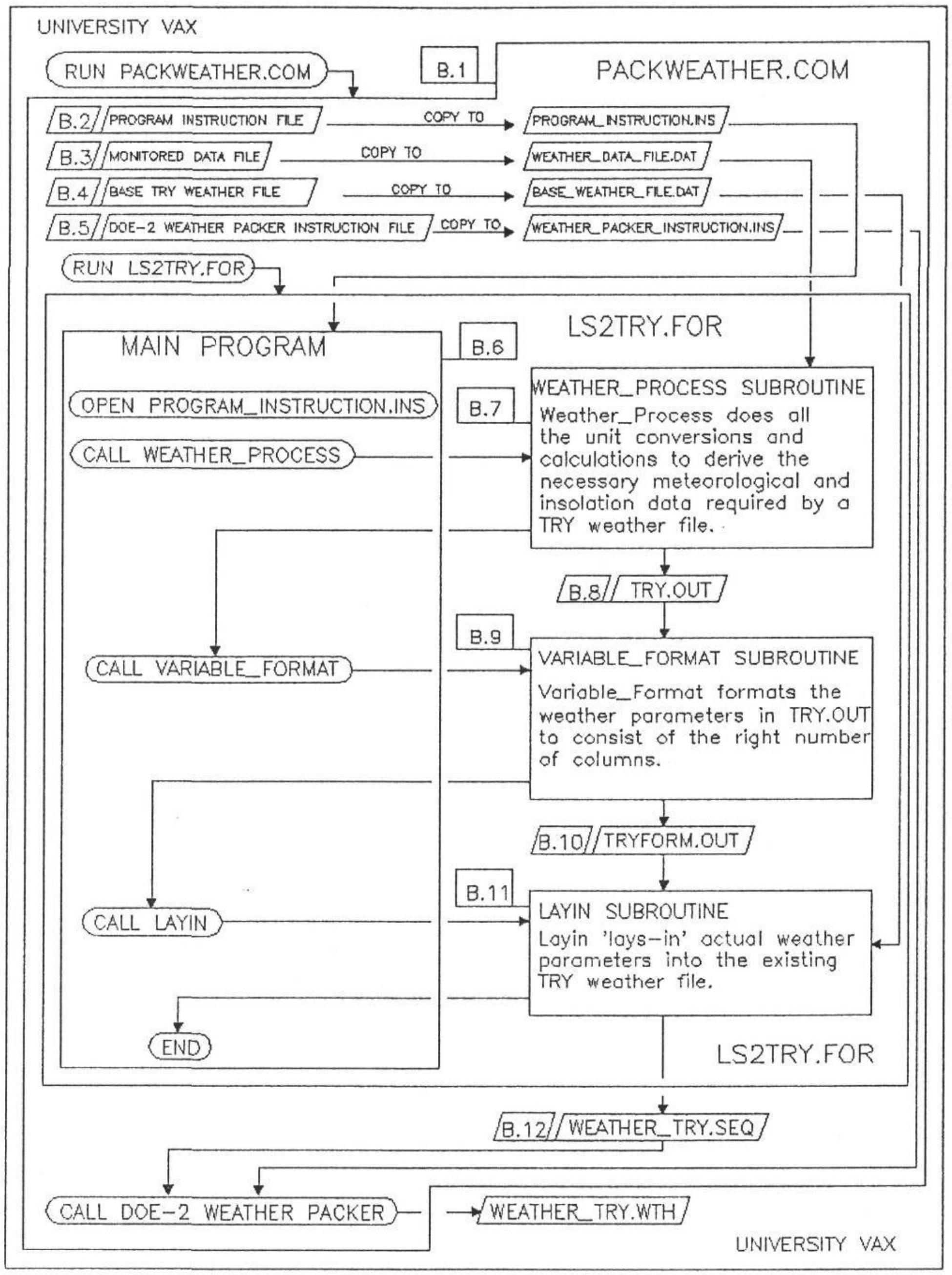

Figure 1 **Methodology to Incorporate Site Monitored Weather Data** -- Flow diagram of the procedure to incorporate site monitored data into a DOE-2 readable binary weather file on the University VAX. One command procedure, PACKWEATHER.COM, performs the data file management, runs the FORTRAN program LS2TRY.FOR, and calls the DOE-2 weather packer to incorporate site monitored data into a binary weather file.

year which the data comes from is not significant to the order which it appears in the data file. The procedure requires the data records to be marked with whole hours, i.e. zero minutes. This is the format of the TRY weather files and the Hour-of-the-Day field needs to be in the format of HHMM. There should be no records with identical time stamps and all non-existing monitored weather data should be marked as '-99.0'. The monitored data file should be robustly processed using data processing routines that check for missing hourly records, duplicate records, and hourly records out of sequence.

The Base TRY weather file, <Base TRY> is any TRY weather file in ASCII format containing 365 days of data. The DOE-2 weather packer requires the weather file to contain one full year of data. Actual weather data is laid-into the base weather file to assure the resulting weather file contains a full year of data.

The program instruction file, <LS2TRY Instruction>, was created to give the methodology the flexibility to process weather data from different weather sites. The instruction file consists of two records. The first record, detailed in Table 2, is data instruction for the WEATHER_PROCESS subroutine and the second record, detailed in Table 3, consists of Y's and N's for yes and no to tell the LAYIN subroutine whether or not to replace a weather parameter on the base weather file with actual data. All numeric data except the STATION NUMBER is input as a real variable, the decimal place specified: the STATION NUMBER is input as an integer.

The instruction file for the DOE-2 weather packer, <Packer Instruction>, contains the necessary data to pack an ASCII weather file. For more information on creating an instruction file, reference the DOE-2 Reference Manual, Part 2, Appendix VIII.C, pp VIII.37 to VIII.38 (LBL 1982).

Table 2 Description of the Program Instruction File's First Record

	Columns	First Record Description
1	1 - 6	The standard meridian for the local time zone (i.e. Central Time Zone is 90.0°)
2	7 - 12	The longitude of the weather station in degrees west
3	13 - 18	The latitude of the weather station in degrees north*
4	19 - 24	The station's standard atmospheric pressure for the station's altitude [psia]
5	25 - 30	First day of Daylight Savings [Julian date], first Sunday in April**
6	31 - 36	Last day of Daylight Savings [Julian date], last Sunday in October**
7	37 - 42	5 digit number specifying the STATION NUMBER***
8	43 - 48	Dominant Wind Direction - '999' if missing wind direction data

* LS2TRY.FOR code is for weather stations in the Northern Hemisphere.

** The value of these two parameters should be set to 0.0 for a monitored weather file that does not have a Daylight Saving time shift.

*** This STATION NUMBER needs to be the same number specified in the data instruction file for the DOE-2 weather packer.

Table 3 Description the Program Instruction File's Second Record

	Columns	Second Record Description
1	1 - 2	Y or N to replace Dry-Bulb temperature
2	3 - 4	Y or N to replace Wet-Bulb temperature
3	5 - 6	Y or N to replace Dew Point temperature
4	7 - 8	Y or N to replace Wind Direction
5	9 - 10	Y or N to replace Wind Speed
6	11 - 12	Y or N to replace Station Pressure
7	13 - 14	Y or N to replace Global Horizontal Radiation
8	15 - 16	Y or N to replace Direct Normal Radiation

APPENDIX A

Data Processing Routines' Example Data Files and Routine Hard-copies


```
1 LoneSTAR Project  ZEC              DOE-2.1D   4/23/1992      15:45:07  PDL RUN 1
Energy Systems Laboratory          Texas A&M University                     JDB
HR-3 = HOURLY-REPORT                                                  PAGE  1-  1
---------------------------------------------------------------------------------
 MMDDHH     PLANT             PLANT             PLANT

            TOTAL             TOTAL             TOTAL
            HEATING           COOLING           ELECTRIC
            BTU/HR            BTU/HR            BTU/HR

            ----( 8)          ----( 9)          ----(10)
  9 1 1           169.          3689051.          2036128.
  9 1 2           170.          3664612.          2020038.
  9 1 3           171.          3772806.          2010383.
  9 1 4           173.          3629084.          2003947.
  9 1 5           174.          3639808.          2008774.
  9 1 6           174.          3555864.          2061874.
  9 1 7           174.          3891487.          2197038.
  9 1 8           172.          3861645.          2438401.
  9 1 9           167.          3885045.          2546210.
  9 110           162.          3857863.          2591265.
  9 111           158.          3529573.          2602529.
  9 112           154.          3367484.          2583220.
  9 113           153.          3383961.          2600919.
  9 114           151.          3442571.          2621838.
  9 115           149.          3381674.          2607356.
  9 116           148.          3349710.          2549429.
  9 117           148.          3363630.          2354729.
  9 118           148.          3370410.          2253356.
  9 119           151.          3402521.          2224392.
  9 120           153.          3459111.          2219565.
  9 121           157.          3628390.          2192210.
  9 122           159.          3639908.          2151983.
  9 123           162.          3661303.          2108538.
  9 124           164.          3705388.          2061874.
0 DAILY SUMMARY (SEP  1)
     MN           148.          3349710.          2003947.
     MX           174.          3891487.          2621838.
     SM          3860.         86132904.         55045988.
     AV           161.          3588871.          2293583.
```

Figure A.1 Two Day Example of DOE-2 HOURLY-REPORT Output


```
1 LoneSTAR Project  ZEC              DOE-2.1D   4/23/1992     15:45:07  PDL RUN 1
Energy Systems Laboratory           Texas A&M University                     JDB
HR-3 = HOURLY-REPORT                                                   PAGE  2-  1
-------------------------------------------------------------------------------
             PLANT              PLANT              PLANT

             TOTAL              TOTAL              TOTAL
             HEATING            COOLING            ELECTRIC
             BTU/HR             BTU/HR             BTU/HR

             ----( 8)           ----( 9)           ----(10)
  9 2 1           165.         3635715.           1991074.
  9 2 2           167.         3657901.           1981419.
  9 2 3           169.         3790060.           1973374.
  9 2 4           170.         3939914.           1968547.
  9 2 5           171.         3883365.           1966938.
  9 2 6           172.         3813105.           1963720.
  9 2 7           173.         3649263.           1947629.
  9 2 8           173.         3946251.           1957283.
  9 2 9           169.         3795406.           1986247.
  9 210           165.         3532216.           2018429.
  9 211           162.         3341847.           2040956.
  9 212           159.         3119392.           2055438.
  9 213           158.         3153871.           2073138.
  9 214           157.         3128918.           2082792.
  9 215           156.         3126860.           2090838.
  9 216           155.         3220465.           2089228.
  9 217           153.         3366613.           2074747.
  9 218           154.         3266043.           2065092.
  9 219           156.         3265660.           2061874.
  9 220           159.         3258212.           2069919.
  9 221           162.         3492583.           2068310.
  9 222           164.         3609627.           2055438.
  9 223           166.         3692564.           2036128.
  9 224           168.         3588905.           2008774.
0 DAILY SUMMARY (SEP  2)
     MN           153.         3119392.           1947629.
     MX           173.         3946251.           2090838.
     SM          3923.        84274760.          48627328.
     AV           163.         3511448.           2026139.
```

Figure A.1 Continued

```
# ! /usr/local/bin/gawk -f
# %W% %G%
# *************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: SEP-PARA.AWK
# Version: %I%
# Last Update: %G%
#
# Description:
# This program divides the DOE-2 output from parametric study into
# seperate files.
#
# Usage:
#  SEP-PARA.AWK <source> <parameter> <output> <# of runs>
#
# <source> is the name of the DOE-2 output file
# <parameter> is the name of the sub-program in DOE-2; LDL for LOADS,
#  SDL for SYSTEMS, and PDL of PLANT which the PARAMETRIC-INPUT
#  instruction was used
# <output> output file name - a numeral is attach to this output file
#  name to distinguished between PARAMETRIC-INPUT instructions
# <# of runs> number of PARAMETRIC-INPUT instructions
#
#
# History:
#       Design: Doug Bronson
#       Code: Doug Bronson
#
# Distribution Rights
#       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#               Texas A & M University., College Station, Texas 77843-3123,
#               (409)- 845-1560
#       SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#       from being claimed by any other party. Texas A & M University intends
#       that the program be placed in the public domain and grants
#       permission for it to be used and redistributed, provided that:
#           1) the source code is distributed
#           2) this notice is retained in all copies of the source code, and
#           3) the program is not sold for profit without written approval
#               from TEES.
#       The program is distributed "as is". TEES provides no warranty or
#       support service unless special arrangements have been made to do so.
#       Certain manufacturers and trade names are mentioned in this code for
#       the purpose of describing their communications protocol. This does
#       not constitute an endorsement or recommendation of such equipment,
#       but is provided for informantional purposes only.
#
#**************************************************************************
# SEP-PARA.AWK seperates DOE-2 parametric runs into seperate files to be
# processed by OUT_PROC.AWK
#
# input form
#   gawk -f sep-para.awk
# 1 source file with extension
# 2 the parameter run made LDL for LOADS, SDL for SYSTEMS and PDL for PLANT
# 3 a 3-4 letter code word for naming the runs
# 4 the number of runs - 1,2,3, etc
#
```

Figure A.2 Hard-copy of the SEP-PARA.AWK Routine


```
BEGIN {
      number_of_runs = ARGV[4];
      code_word      = ARGV[3];
      para_type      = ARGV[2]
      ARGC = 2;
#
      i = 1;
      while (i <= number_of_runs) {
            output[i] = code_word"-"i".out";
            print output[i];
            i++;
      }
}
#
{
      if ((NF > 2) && ($(NF-1) == "RUN") && ($(NF-2) == para_type)) print_file
= $NF;
#
      if ((print_file >= 1) && (print_file <= number_of_runs)) print $0 >
output[print_file];
}
#
END {
      while (i <= number_of_runs) {
            close(output[i]);
            i++;
      }
}
```

Figure A.2 Continued


```
# ! /usr/local/bin/nawk -f
# %W% %G%
# ************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: RAWEXTR.AWK
# Version: %I%
# Last Update: %G%
#
# Description:
# This program removes the column headers and footers from the DOE-2 hourly
# report to create contiguous hourly data file.
#
# Usage:
#  RAWEXTR.AWK <source> <report1> <report2> <report3> <report4>
#
# <source> name of the DOE-2 output file
# RAWEXTR.AWK is setup to extract hourly data from four DOE-2 hourly reports.
# <report1>, <report2>, etc. are the U-names of the HOURLY-REPORT in the
# subprograms. If hourly data needs to be extracted from less than four
# HOURLY-REPORTs, unused <outfile?> need to be filled in with an ambiguous
# names, i.e., "XXXX"
#
# History:
#       Design: Doug Bronson
#       Code: Doug Bronson
#
# Distribution Rights
#       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#               Texas A & M University., College Station, Texas 77843-3123,
#               (409)- 845-1560
#       SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#       from being claimed by any other party. Texas A & M University intends
#       that the program be placed in the public domain and grants
#       permission for it to be used and redistributed, provided that:
#           1) the source code is distributed
#           2) this notice is retained in all copies of the source code, and
#           3) the program is not sold for profit without written approval
#              from TEES.
#       The program is distributed "as is". TEES provides no warranty or
#       support service unless special arrangements have been made to do so.
#       Certain manufacturers and trade names are mentioned in this code for
#       the purpose of describing their communications protocol. This does
#       not constitute an endorsement or recommendation of such equipment,
#       but is provided for informantional purposes only.
#
#*************************************************************************
# RAWEXTR.AWK
# THIS FILE EXTRACT DATA FROM A STANDARD DOE2.1 OUTPUT FILE AND PLACE
# IT IN TO A FILE SOURCE_NAME.AWK
#
# input format
# gawk -f rawextr.awk
#  1 source_file
#  2  report1
#  3  report2
#  4  report3
#  5  report4
# a designator must be at each report location, real or false
```

Figure A.3 Hard-copy of the RAWEXTR.AWK Routine


```
BEGIN {
      errorflag = 0;
      if (ARGC != 6) {
      printf(" \n");
      printf("USAGE :\n");
      printf("nawk -f RAWEXTR.AWK <source> <report1> <report2> <report3>
<report4>\n");
      printf(" \n");
      errorflag = 1;
      exit 4;
      }

      number_of_files = ARGC - 2;
      print " ";
      print "The specified Hourly Reports are in file(s):";
      i = 1;
      while (i <= number_of_files) {
            group_name[i] = ARGV[ARGC - (number_of_files + 1) + i];
            outfile[i] = group_name[i]".out";
            print outfile[i];
            i++;
      }
      ARGC = ARGC - number_of_files;
}
            $1 == group_name[1],$2 == "DAILY" {
                  if (($1~/[1-9]+/ && $1!~/[A-Za-z]/ && $1!~/[---]/) &&
($2~/[1-9.]+/ && $2!~/[A-Za-z]/)){
                  print $0 > outfile[1]
                  }
            }
#
            $1 == group_name[2],$2 == "DAILY" {
                  if (($1~/[1-9]+/ && $1!~/[A-Za-z]/ && $1!~/[---]/) &&
($2~/[1-9.]+/ && $2!~/[A-Za-z]/)){
                  print $0 > outfile[2]
                  }
            }
#
            $1 == group_name[3],$2 == "DAILY" {
                  if (($1~/[1-9]+/ && $1!~/[A-Za-z]/ && $1!~/[---]/) &&
($2~/[1-9.]+/ && $2!~/[A-Za-z]/)){
                  print $0 > outfile[3]
                  }
            }
#
            $1 == group_name[4],$2 == "DAILY" {
                  if (($1~/[1-9]+/ && $1!~/[A-Za-z]/ && $1!~/[---]/) &&
($2~/[1-9.]+/ && $2!~/[A-Za-z]/)){
                  print $0 > outfile[4]
                  }
            }
#
END {
      if (errorflag == 1) exit;
      i = 1;
#
      while (i <= number_of_files) {
            close(outfile[i]);
            i++;
      }
}
# END
```

Figure A.3 Continued

MM DAY HR	Heating Load (Btu/hr)	Cooling Load (Btu/hr)	Electricity Consumption (Btu/hr)
9 1 1	169.	3689051.	2036128.
9 1 2	170.	3664612.	2020038.
9 1 3	171.	3772806.	2010383.
9 1 4	173.	3629084.	2003947.
9 1 5	174.	3639808.	2008774.
9 1 6	174.	3555864.	2061874.
9 1 7	174.	3891487.	2197038.
9 1 8	172.	3861645.	2438401.
9 1 9	167.	3885045.	2546210.
9 110	162.	3857863.	2591265.
9 111	158.	3529573.	2602529.
9 112	154.	3367484.	2583220.
9 113	153.	3383961.	2600919.
9 114	151.	3442571.	2621838.
9 115	149.	3381674.	2607356.
9 116	148.	3349710.	2549429.
9 117	148.	3363630.	2354729.
9 118	148.	3370410.	2253356.
9 119	151.	3402521.	2224392.
9 120	153.	3459111.	2219565.
9 121	157.	3628390.	2192210.
9 122	159.	3639908.	2151983.
9 123	162.	3661303.	2108538.
9 124	164.	3705388.	2061874.
9 2 1	165.	3635715.	1991074.
9 2 2	167.	3657901.	1981419.
9 2 3	169.	3790060.	1973374.
9 2 4	170.	3939914.	1968547.
9 2 5	171.	3883365.	1966938.
9 2 6	172.	3813105.	1963720.
9 2 7	173.	3649263.	1947629.
9 2 8	173.	3946251.	1957283.
9 2 9	169.	3795406.	1986247.
9 210	165.	3532216.	2018429.
9 211	162.	3341847.	2040956.
9 212	159.	3119392.	2055438.
9 213	158.	3153871.	2073138.
9 214	157.	3128918.	2082792.
9 215	156.	3126860.	2090838.
9 216	155.	3220465.	2089228.
9 217	153.	3366613.	2074747.
9 218	154.	3266043.	2065092.
9 219	156.	3265660.	2061874.
9 220	159.	3258212.	2069919.
9 221	162.	3492583.	2068310.
9 222	164.	3609627.	2055438.
9 223	166.	3692564.	2036128.
9 224	168.	3588905.	2008774.

Figure A.4 Two Day Example of Raw Columnar DOE-2 Output without Header and Footers


```
# ! /usr/local/bin/gawk -f
# %W% %G%
# ************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: OUT_PROC.AWK
# Version: %I%
# Last Update: %G%
#
# Description:
# This program extracts the multi-columns in the contiguous DOE-2 hourly
# output file in to single column data files with either the LoanSTAR date
# stamp or just the decimal date.
#
# Usage:
#  OUT_PROC.AWK <source> <stamp> <year> <begins> <ends> <column> <factor>
<output>
#
# <source> the contiguous multi-column data file
# <stamp> - 1 for the LoanSTAR date stamp - 2 for only the Decimal date
# <yr> the year that the data represents
# <begins> Julian Date for the first day of daylight savings - First Sunday in
April
# <ends> Julian Date for the last day of daylight savings - Last Sunday in
October
# <col> data column's position from the farthest right column
# <factor> conversion factor to multiple the data column by
# <output> output file name that the single data column is printed out to
#
# the extract more than one column from the contiguous multi-column data file,
# repeat <column> <factor> <output> columns for each one
#
# History:
#       Design: Doug Bronson
#       Code: Doug Bronson

# Distribution Rights
#       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#               Texas A & M University., College Station, Texas 77843-3123,
#               (409)- 845-1560
#       SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#       from being claimed by any other party. Texas A & M University intends
#       that the program be placed in the public domain and grants
#       permission for it to be used and redistributed, provided that:
#           1) the source code is distributed
#           2) this notice is retained in all copies of the source code, and
#           3) the program is not sold for profit without written approval
#              from TEES.
#       The program is distributed "as is". TEES provides no warranty or
#       support service unless special arrangements have been made to do so.
#       Certain manufacturers and trade names are mentioned in this code for
#       the purpose of describing their communications protocol. This does
#       not constitute an endorsement or recommendation of such equipment,
#       but is provided for informantional purposes only.
#
#*************************************************************************
# OUT_PROC.AWK
```

Figure A.5 Hard-copy of the OUT_PROC.AWK Routine


```
#
# input format
# gawk -f out_proc.awk
#  1    source_file
#  2    1 for loanstar date stamp - 2 only the decimal date
#  3    year
#  4    day of the year daylight savings begins, first Sunday in April
#  5    day of the year daylight savings ends, last Sunday in October
#  6    column of the data from the farthest right column
#  7    conversion factor for the data field
#  8    name of the file the data field should be printed to
#  repeat 6-8 as many times as necessary
#
BEGIN {
      errorflag = 0
      if ((ARGC < 9) || ((ARGC - 6)%3 != 0)) {
      printf(" \n");
      printf("Usage :\n");
      printf("nawk -f OUT_PROC.AWK <source> <stamp> <year> <begins>
<ends>\n");
      printf("                                    repeat - <column> <factor>
<outfile>\n");
      printf(" \n");
      errorflag = 1;
      exit 4
      }

      number_of_entries = ARGC - 6;
      print " ";
      number_of_outfiles = int(number_of_entries/3);
      print "The number of outfiles = ",number_of_outfiles;
      print "The outfiles are:";
      PRINT_FORMAT = ARGV[ARGC - (number_of_entries + 4)];
      YR = ARGV[ARGC - (number_of_entries + 3)];
      DAY_LIGHT_BEGINS = ARGV[ARGC - (number_of_entries + 2)];
      DAY_LIGHT_ENDS = ARGV[ARGC - (number_of_entries + 1)];
      i = 3;
#
      while (i <= number_of_entries) {
            p = int(i/3);
            file[p] = ARGV[ARGC - (number_of_entries + 1) + i];
            outfile[p] = file[p];
            print outfile[p];
            outfile_col[p] = ARGV[ARGC - (number_of_entries + 3) + i];
            outfile_conv[p] =  ARGV[ARGC - (number_of_entries + 2) + i];
            i += 3;
      }
ARGC = ARGC - (number_of_entries + 4);
#
DAYCODE[1] = 0;
DAYCODE[2] = 31;
DAYCODE[3] = 59;
DAYCODE[4] = 90;
DAYCODE[5] = 120;
DAYCODE[6] = 151;
DAYCODE[7] = 181;
DAYCODE[8] = 212;
DAYCODE[9] = 243;
DAYCODE[10] = 273;
DAYCODE[11] = 304;
DAYCODE[12] = 334;
```

Figure A.5 Continued

```
#
OLD_DAY = 0;
OLD_MM = 0;
OLD_YR = 0;
OLD_HR = 0;
}
#
{
#       Seperating the date string
#
      DATESET = substr($0,2,6);
#       if (substr(DATESET,6,1) == " ") {
#               HR = substr(DATESET,4,2)+0;
#               DAY = substr(DATESET,2,2)+0;
#               MM = substr(DATESET,0,1)+0;
#               }
#               else {
       HR = substr(DATESET,5,2)+0;
       DAY = substr(DATESET,3,2)+0;
       MM = substr(DATESET,1,2)+0;
#               }
#
#       Correcting the Day Light Saving Shift
#
dy = DAYCODE[MM] + DAY;
#
if (YR % 4.0 == 0.0 && MM >= 3.0) dy = dy + 1
#
if (dy >= DAY_LIGHT_BEGINS && dy < DAY_LIGHT_ENDS) {
      HR += 1
}
#
if (HR == 25) {
      HR = 1;
      if (DAY >= 28) {
            if (DAY == 31) {
                  DAY = 1;
                  if (MM != 12) {
                  MM = MM + 1;
                  }
                  else {
                        MM = 1 ;
                        YR = YR + 1;
                  }
            }
            else {
            if (DAY == 30 && (MM == 4 || MM == 6 || MM ==9 || MM == 11)) {
                        DAY = 1;
                        MM = MM + 1;
                  }
                  else {
                  if (MM == 2 && (DAY == 29 || ( DAY ==28 && YR % 4 != 0 ))) {
                              DAY = 1;
                              MM = MM + 1;
                        }
                        else {
                              DAY = DAY + 1;
                        }
                  }
            }
      }
```

Figure A.5 Continued


```
else {
      DAY = DAY + 1;
}
}
#
if (YR % 4.0 == 0.0 && MM >=3.) add_day = 1.0
else add_day = 0.0
#
DY = int(YR*1000.0 + dy + add_day);
#
DEC_TIME = int((YR-80.)*365.25+0.75) + (DAYCODE[MM]+DAY-1.+add_day) +
int((HR/24.)*10000.+0.5)/10000.
#
n = number_of_columns = NF;
#
if (HR == 24 || HR == 2400) {
      HR = 0;
      if (DAY >= 28) {
            if (DAY == 31) {
                  DAY = 1;
                  if (MM != 12) {
                        MM = MM + 1;
                        DY = DY + 1
                  }
                  else {
                        MM = 1 ;
                        YR = YR + 1;
                        DY = YR*1000 + 1
                  }
            }
      else {
            if (DAY == 30 && (MM == 4 || MM == 6 || MM ==9 || MM == 11)) {
                  DAY = 1;
                  MM = MM + 1;
                  DY = DY + 1
             }
             else {
                  if (MM == 2 && (DAY == 29 || (DAY == 28 && YR % 4 != 0 ))) {
                        DAY = 1;
                        MM = MM + 1;
                        DY = DY + 1;
                  }
                  else {
                        DAY = DAY + 1;
                        DY = DY + 1
                  }
             }
      }
}
else {
      DAY = DAY + 1;
      DY = DY + 1
}
}
```

Figure A.5 Continued


```
if (HR == OLD_HR && DAY == OLD_DAY && MM == OLD_MM && YR == OLD_YR) {
      print "DATA POINT REMOVED AT " MM, DAY, YR, HR
}
else {
      for (i = 1; i <= number_of_outfiles; i++) {
            oc = outfile_col[i];
            Value = $(n + 1 - oc)*outfile_conv[i];
#
      if (PRINT_FORMAT == 1) {
            printf "%2.0f %2.0f %2.0f %5.0f %10.4f %2.0f %-
10.4f\n",MM,DAY,YR,DY,DEC_TIME,HR,Value > outfile[i];
      }
#
      if (PRINT_FORMAT == 2) {
            printf "%10.4f %-10.4f\n",DEC_TIME, Value  > outfile[i];
      }
      }
            OLD_HR = HR;
            OLD_YR = YR;
            OLD_MM = MM;
            OLD_DAY = DAY;
      }
}
END {
      if (errorflag == 1) exit;
      i =1;
while (i <= number_of_outfiles) {
      close(outfile[i]);
      i++;
}
}
#
# END
```

Figure A.5 Continued

Decimal Date	*Cooling Load (MMBtu/hr)*
3531.0833	3.6891
3531.1250	3.6646
3531.1667	3.7728
3531.2083	3.6291
3531.2500	3.6398
3531.2917	3.5559
3531.3333	3.8915
3531.3750	3.8616
3531.4167	3.8850
3531.4583	3.8579
3531.5000	3.5296
3531.5417	3.3675
3531.5833	3.3840
3531.6250	3.4426
3531.6667	3.3817
3531.7083	3.3497
3531.7500	3.3636
3531.7917	3.3704
3531.8333	3.4025
3531.8750	3.4591
3531.9167	3.6284
3531.9583	3.6399
3532.0000	3.6613
3532.0417	3.7054
3532.0833	3.6357
3532.1250	3.6579
3532.1667	3.7901
3532.2083	3.9399
3532.2500	3.8834
3532.2917	3.8131
3532.3333	3.6493
3532.3750	3.9463
3532.4167	3.7954
3532.4583	3.5322
3532.5000	3.3418
3532.5417	3.1194
3532.5833	3.1539
3532.6250	3.1289
3532.6667	3.1269
3532.7083	3.2205
3532.7500	3.3666
3532.7917	3.2660
3532.8333	3.2657
3532.8750	3.2582
3532.9167	3.4926
3532.9583	3.6096
3533.0000	3.6926

Figure A.6 Two Day Example of Predicted Chilled Water Energy Consumption Data with Decimal Date Stamp

LoneSTAR Date Stamp						*Cooling Load (MMBtu/hr)*
9	1	89	89244	3531.0833	2	3.6891
9	1	89	89244	3531.1250	3	3.6646
9	1	89	89244	3531.1667	4	3.7728
9	1	89	89244	3531.2083	5	3.6291
9	1	89	89244	3531.2500	6	3.6398
9	1	89	89244	3531.2917	7	3.5559
9	1	89	89244	3531.3333	8	3.8915
9	1	89	89244	3531.3750	9	3.8616
9	1	89	89244	3531.4167	10	3.8850
9	1	89	89244	3531.4583	11	3.8579
9	1	89	89244	3531.5000	12	3.5296
9	1	89	89244	3531.5417	13	3.3675
9	1	89	89244	3531.5833	14	3.3840
9	1	89	89244	3531.6250	15	3.4426
9	1	89	89244	3531.6667	16	3.3817
9	1	89	89244	3531.7083	17	3.3497
9	1	89	89244	3531.7500	18	3.3636
9	1	89	89244	3531.7917	19	3.3704
9	1	89	89244	3531.8333	20	3.4025
9	1	89	89244	3531.8750	21	3.4591
9	1	89	89244	3531.9167	22	3.6284
9	1	89	89244	3531.9583	23	3.6399
9	2	89	89245	3532.0000	0	3.6613
9	2	89	89244	3532.0417	1	3.7054
9	2	89	89245	3532.0833	2	3.6357
9	2	89	89245	3532.1250	3	3.6579
9	2	89	89245	3532.1667	4	3.7901
9	2	89	89245	3532.2083	5	3.9399
9	2	89	89245	3532.2500	6	3.8834
9	2	89	89245	3532.2917	7	3.8131
9	2	89	89245	3532.3333	8	3.6493
9	2	89	89245	3532.3750	9	3.9463
9	2	89	89245	3532.4167	10	3.7954
9	2	89	89245	3532.4583	11	3.5322
9	2	89	89245	3532.5000	12	3.3418
9	2	89	89245	3532.5417	13	3.1194
9	2	89	89245	3532.5833	14	3.1539
9	2	89	89245	3532.6250	15	3.1289
9	2	89	89245	3532.6667	16	3.1269
9	2	89	89245	3532.7083	17	3.2205
9	2	89	89245	3532.7500	18	3.3666
9	2	89	89245	3532.7917	19	3.2660
9	2	89	89245	3532.8333	20	3.2657
9	2	89	89245	3532.8750	21	3.2582
9	2	89	89245	3532.9167	22	3.4926
9	2	89	89245	3532.9583	23	3.6096
9	3	89	89246	3533.0000	0	3.6926

Figure A.7 Two Day Example of Predicted Chilled Water Energy Consumption Data with LoanSTAR Date Stamp

Decimal Date	*Electricity Consumption (kWh/h)*
3531.0833	1010.300
3531.1250	1011.900
3531.1667	1011.300
3531.2083	1009.700
3531.2500	1012.600
3531.2917	1032.000
3531.3333	1105.500
3531.3750	1270.000
3531.4167	1345.100
3531.4583	1366.500
3531.5000	1364.800
3531.5417	1347.400
3531.5833	1363.200
3531.6250	1372.100
3531.6667	1361.700
3531.7083	1312.300
3531.7500	1168.800
3531.7917	1099.500
3531.8333	1073.400
3531.8750	1075.000
3531.9167	1050.500
3531.9583	1018.400
3532.0000	1007.800
3532.0417	1002.400
3532.0833	997.300
3532.1250	994.100
3532.1667	989.300
3532.2083	987.300
3532.2500	986.000
3532.2917	985.800
3532.3333	976.700
3532.3750	985.500
3532.4167	1006.100
3532.4583	1026.300
3532.5000	1034.800
3532.5417	1030.000
3532.5833	1039.700
3532.6250	1055.100
3532.6667	1061.400
3532.7083	1050.600
3532.7500	1026.500
3532.7917	1015.400
3532.8333	1004.600
3532.8750	1011.400
3532.9167	1015.900
3532.9583	981.200
3533.0000	970.500

Figure A.8 Two Day Example of Monitored Whole-building Electricity Consumption Data

Decimal Date	*Electricity Consumption (kWh/h)*
3531.0833	1101.33
3531.1250	1091.35
3531.1667	1085.35
3531.2083	1081.36
3531.2500	1084.36
3531.2917	1117.31
3531.3333	1201.19
3531.3750	1350.97
3531.4167	1417.87
3531.4583	1445.83
3531.5000	1452.82
3531.5417	1440.84
3531.5833	1451.82
3531.6250	1464.80
3531.6667	1455.82
3531.7083	1419.87
3531.7500	1299.04
3531.7917	1236.14
3531.8333	1218.16
3531.8750	1215.17
3531.9167	1198.19
3531.9583	1173.23
3532.0000	1146.27
3532.0417	1117.31
3532.0833	1073.37
3532.1250	1067.38
3532.1667	1062.39
3532.2083	1059.39
3532.2500	1058.39
3532.2917	1056.40
3532.3333	1046.41
3532.3750	1052.40
3532.4167	1070.38
3532.4583	1090.35
3532.5000	1104.33
3532.5417	1113.31
3532.5833	1124.30
3532.6250	1130.29
3532.6667	1135.28
3532.7083	1134.28
3532.7500	1125.30
3532.7917	1119.31
3532.8333	1117.31
3532.8750	1122.30
3532.9167	1121.30
3532.9583	1113.31
3533.0000	1101.33

Figure A.9 Two Day Example of Predicted Whole-building Electricity Consumption Data

```
/* ! /usr/local/bin/ */
/* %W% %G% */
/* ********************************************************************** */
/* Copyright (c) 1990, Texas Enginnering Experiment Station */
/* */
/* Program: RESIDUAL.C */
/* Version: %I% */
/* Last Update: %G% */
/* */
/* Description: */
/* This program subtracts the monitored whole-building electricity from */
/* the predicted whole-building electricity consumption to create a */
/* positive and a negative residual data files. */
/* */
/* Usage: */
/*  RESIDUAL <predicted> <monitored> <positive> <negative> */
/* */
/* <predicted> name of the predicted whole-building electricity consumption */
/* <monitored> name of the monitored whole-building electricity consumption */
/* <positive> name of the outfile for the positive residual data file */
/* <negative> name of the outfile for the negative residual data file */
/* */
/* History: */
/*       Design: Doug Bronson */
/*       Code: Doug Bronson */
/* */
/* Distribution Rights */
/*       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept., */
/*              Texas A & M University., College Station, Texas 77843-3123, */
/*              (409)- 845-1560 */
/*       SUPPORTED BY: State of Texas Governor's Energy Management Center */
/* */
/* COPYRIGHT NOTICE:This program bears a copyright notice to prevent rights */
/*    from being claimed by any other party. Texas A & M University intends */
/*    that the program be placed in the public domain and grants  */
/*    permission for it to be used and redistributed, provided that: */
/*         1) the source code is distributed */
/*         2) this notice is retained in all copies of the source code, and */
/*         3) the program is not sold for profit without written approval */
/*            from TEES. */
/*    The program is distributed "as is". TEES provides no warranty or */
/*    support service unless special arrangements have been made to do so. */
/*    Certain manufacturers and trade names are mentioned in this code for */
/*    the purpose of describing their communications protocol. This does */
/*    not constitute an endorsement or recommendation of such equipment, */
/*     but is provided for informantional purposes only. */
/*
/*********************************************************************** */
/*include <stdio.h>
/* residual.c creates positive and negative residual files from simulation */
/* and monitored data */
/* input form */
/* residual */
/* 1 simulated data file - decimal date, data column */
/* 2 monitored data file - decimal date, data column */
/* 3 positive residual data file name */
/* 4 negative residual data file name */
```

Figure A.10 Hard-copy of the RESIDUAL.C Routine


```
main (argc,argv)
      int argc;
      char *argv[];
{
FILE *simulated;
FILE *actual;
FILE *positive;
FILE *negative;
int i,j;
float diff,simval,actval,posval,negval,value,comdate;
float date[2];

simulated = fopen(argv[1],"r");
actual    = fopen(argv[2],"r");
positive  = fopen(argv[3],"w");
negative  = fopen(argv[4],"w");

/* should check each file* != NULL */
while(!feof(simulated)&&!feof(actual))
{
fscanf(simulated,"%f %f\n",&date[1],&simval);
fscanf(actual,"%f %f\n",&date[2],&actval);
/*
diff = date[1] - date[2];
if (diff < -0.02 || diff > 0.02)
{
printf("Error -- file dates mismatch. files %2.0f and %2.0f\n",i,j);
printf("Terminating operation now\n");
exit(1);
}
/*
comdate=date[1];
value = simval - actval;
/*
if (value >= 0.0) {
posval = value;
negval = 0.0;
if (actval == -99.0) posval = 0.0;
}
else {
negval = value - 2*value;
posval = 0.0;
if (actval == -99.0) negval = 0.0;
}
fprintf(positive,"%10.4f %7.2f\n",comdate,posval);
fprintf(negative,"%10.4f %7.2f\n",comdate,negval);
}
}
```

Figure A.10 Continued

Decimal Date	*Positive Residual (kWh/h)*
3531.0833	91.03
3531.1250	79.45
3531.1667	74.05
3531.2083	71.66
3531.2500	71.76
3531.2917	85.31
3531.3333	95.69
3531.3750	80.97
3531.4167	72.77
3531.4583	79.33
3531.5000	88.02
3531.5417	93.44
3531.5833	88.62
3531.6250	92.70
3531.6667	94.12
3531.7083	107.57
3531.7500	130.24
3531.7917	136.64
3531.8333	144.76
3531.8750	140.17
3531.9167	147.69
3531.9583	154.83
3532.0000	138.47
3532.0417	114.91
3532.0833	76.07
3532.1250	73.28
3532.1667	73.09
3532.2083	72.09
3532.2500	72.39
3532.2917	70.60
3532.3333	69.71
3532.3750	66.90
3532.4167	64.28
3532.4583	64.05
3532.5000	69.53
3532.5417	83.31
3532.5833	84.60
3532.6250	75.19
3532.6667	73.88
3532.7083	83.68
3532.7500	98.80
3532.7917	103.91
3532.8333	112.71
3532.8750	110.90
3532.9167	105.40
3532.9583	132.11
3533.0000	130.83

Figure A.11 Two Day Example of Positive Residual Whole-building Electricity Consumption Data

Decimal Date	*Negative Residual (kWh/h)*
3531.0833	0.00
3531.1250	0.00
3531.1667	0.00
3531.2083	0.00
3531.2500	0.00
3531.2917	0.00
3531.3333	0.00
3531.3750	0.00
3531.4167	0.00
3531.4583	0.00
3531.5000	0.00
3531.5417	0.00
3531.5833	0.00
3531.6250	0.00
3531.6667	0.00
3531.7083	0.00
3531.7500	0.00
3531.7917	0.00
3531.8333	0.00
3531.8750	0.00
3531.9167	0.00
3531.9583	0.00
3532.0000	0.00
3532.0417	0.00
3532.0833	0.00
3532.1250	0.00
3532.1667	0.00
3532.2083	0.00
3532.2500	0.00
3532.2917	0.00
3532.3333	0.00
3532.3750	0.00
3532.4167	0.00
3532.4583	0.00
3532.5000	0.00
3532.5417	0.00
3532.5833	0.00
3532.6250	0.00
3532.6667	0.00
3532.7083	0.00
3532.7500	0.00
3532.7917	0.00
3532.8333	0.00
3532.8750	0.00
3532.9167	0.00
3532.9583	0.00
3533.0000	0.00

Figure A.12 Two Day Example of Negative Residual Whole-building Electricity Consumption Data


```
@echo off
REM 3DGRAPH.BAT is the batch file which takes simulated electrical
REM load data and the positive and negative graph data files, runs
REM them through COLROW3D, and makes a 3D-graph of the simulated
REM data and positive and negative residual plots
REM
REM input format
REM call 3dgraph
REM %1 simulated data or residual graph data name, *.dat,no extension
REM %2 0 or 1, for COLROW3D, 0 for a full year of data, 1 for less
REM %3 name of the 123auto.wk1 file, *.std, no extension
REM
REM COLROW3D
REM --------------------------------------------------------------------

colrow3d %1.dat %1.3d %2
del *.log
REM
copy %1.3d c:\<123 default directory>\3dimport.std

cd <123 default directory>

copy %3.std auto123.wk1

123

copy yearview.std c:\<data directory>\%1.pic
copy worksht.std c:\<data directory>\%1.wk1

del auto123.wk1
del 3dimport.std
del yearview.std
del worksht.std

cd c:\<data directory>
```

Figure A.13 Hard-copy of the 3DGRAPH.BAT Batch File

	A	B	C	D	E	W	X	Y
1	243	0	1	2	...	21	22	23
2	244	0	0	1010.3	...	1075	1050.5	1018.4
3	245	1007.8	1002.4	997.3	...	1011.4	1015.9	981.2
4	246	970.5	0	0	...	0	0	0

Figure A.14 Two Day Example of Monitored Whole-building Electricity Consumption Data in Matrix Format

	AB	AC	AD	AE	AF
9	/fin3dimport.std~{goto}a2~				
10	/wic~(@date(89,12,31)+{right})~/rfd2~/c~.{right}{end}{down}{left}~/rv{end}{down}~~{right}/wdc~{goto}b1~				
11	{app2}rgtshxb1..y1~y{left}{down}.{end}{down}~a{down}.{end}{down}{end}{right}~				
12	otf~ts~txHour-of-the-Day~tyDay-of-the-Year~tzElectricity [kWh/h]~				
13	szml{esc}0~u{esc}1500~inqsx2~sy20~bq				
14	dr2vhayqsyearview.std~q				
15	/rnd\0~				
16	/fs{esc}worksht.std~/qy				

Figure A.15 Hard-copy of a 123AUTO.WK1 Macro - 3DELEC.STD

Date	*Dry bulb Temperature (F)*
3531.0833	82.0000
3531.1250	81.0000
3531.1667	80.0000
3531.2083	79.0000
3531.2500	79.0000
3531.2917	79.0000
3531.3333	79.0000
3531.3750	80.0000
3531.4167	83.0000
3531.4583	87.0000
3531.5000	89.0000
3531.5417	92.0000
3531.5833	93.0000
3531.6250	95.0000
3531.6667	96.0000
3531.7083	97.0000
3531.7500	97.0000
3531.7917	97.0000
3531.8333	95.0000
3531.8750	93.0000
3531.9167	90.0000
3531.9583	88.0000
3532.0000	86.0000
3532.0417	84.0000
3532.0833	83.0000
3532.1250	82.0000
3532.1667	81.0000
3532.2083	80.0000
3532.2500	80.0000
3532.2917	79.0000
3532.3333	79.0000
3532.3750	80.0000
3532.4167	84.0000
3532.4583	87.0000
3532.5000	90.0000
3532.5417	93.0000
3532.5833	95.0000
3532.6250	96.0000
3532.6667	98.0000
3532.7083	99.0000
3532.7500	99.0000
3532.7917	98.0000
3532.8333	96.0000
3532.8750	94.0000
3532.9167	91.0000
3532.9583	89.0000
3533.0000	87.0000

Figure A.16 Two Day Example of Hourly Outdoor Dry Bulb Temperature Data

Decimal Date	Outdoor Specific Humidity (lbw/lba)
3531.0833	0.0181
3531.1250	0.0184
3531.1667	0.0195
3531.2083	0.0198
3531.2500	0.0198
3531.2917	0.0198
3531.3333	0.0207
3531.3750	0.0205
3531.4167	0.0198
3531.4583	0.0179
3531.5000	0.0156
3531.5417	0.0132
3531.5833	0.0121
3531.6250	0.0117
3531.6667	0.0106
3531.7083	0.0096
3531.7500	0.0096
3531.7917	0.0096
3531.8333	0.0101
3531.8750	0.0113
3531.9167	0.0137
3531.9583	0.0158
3532.0000	0.0172
3532.0417	0.0186
3532.0833	0.0188
3532.1250	0.0200
3532.1667	0.0202
3532.2083	0.0214
3532.2500	0.0214
3532.2917	0.0217
3532.3333	0.0217
3532.3750	0.0224
3532.4167	0.0205
3532.4583	0.0179
3532.5000	0.0154
3532.5417	0.0130
3532.5833	0.0117
3532.6250	0.0106
3532.6667	0.0094
3532.7083	0.0092
3532.7500	0.0099
3532.7917	0.0094
3532.8333	0.0098
3532.8750	0.0103
3532.9167	0.0126
3532.9583	0.0156
3533.0000	0.0179

Figure A.17 Two Day Example of Hourly Outdoor Specific Humidity Data

Decimal Date	Cooling Load (MMBtu/hr)
3531.0833	8.10
3531.1250	8.04
3531.1667	8.27
3531.2083	7.97
3531.2500	7.99
3531.2917	7.85
3531.3333	8.59
3531.3750	8.55
3531.4167	8.61
3531.4583	8.54
3531.5000	7.86
3531.5417	7.52
3531.5833	7.55
3531.6250	7.67
3531.6667	7.54
3531.7083	7.46
3531.7500	7.45
3531.7917	7.43
3531.8333	7.48
3531.8750	7.59
3531.9167	7.95
3531.9583	8.00
3532.0000	8.05
3532.0417	8.14
3532.0833	7.99
3532.1250	8.04
3532.1667	8.30
3532.2083	8.61
3532.2500	8.49
3532.2917	8.35
3532.3333	8.01
3532.3750	8.62
3532.4167	8.32
3532.4583	7.77
3532.5000	7.38
3532.5417	6.90
3532.5833	6.96
3532.6250	6.89
3532.6667	6.86
3532.7083	7.04
3532.7500	7.34
3532.7917	7.12
3532.8333	7.11
3532.8750	7.10
3532.9167	7.60
3532.9583	7.90
3533.0000	8.10

Figure A.18 Two Day Example of Hourly Chilled Water Consumption Data

Decimal Date	*Heating Load (MMBtu/hr)*
3531.0833	0.08
3531.1250	0.08
3531.1667	0.09
3531.2083	0.10
3531.2500	0.11
3531.2917	0.09
3531.3333	0.04
3531.3750	0.00
3531.4167	0.00
3531.4583	0.00
3531.5000	0.00
3531.5417	0.00
3531.5833	0.00
3531.6250	0.00
3531.6667	0.00
3531.7083	0.00
3531.7500	0.00
3531.7917	0.00
3531.8333	0.00
3531.8750	0.01
3531.9167	0.04
3531.9583	0.06
3532.0000	0.07
3532.0417	0.07
3532.0833	0.08
3532.1250	0.09
3532.1667	0.09
3532.2083	0.10
3532.2500	0.10
3532.2917	0.11
3532.3333	0.13
3532.3750	0.13
3532.4167	0.09
3532.4583	0.08
3532.5000	0.08
3532.5417	0.10
3532.5833	0.11
3532.6250	0.12
3532.6667	0.12
3532.7083	0.10
3532.7500	0.09
3532.7917	0.09
3532.8333	0.10
3532.8750	0.11
3532.9167	0.11
3532.9583	0.12
3533.0000	0.12

Figure A.19 Two Day Example of Hourly Hot Water Consumption Data

```
/* ! /usr/local/bin/ */
/* %W% %G%  */
/* ********************************************************************* */
/* Copyright (c) 1990, Texas Enginnering Experiment Station */
/* */
/* Program: DOE-WEA.C */
/* Version: %I% */
/* Last Update: %G% */
/* */
/* Description: */
/* This program uses hourly dry bulb temperature (F) data and specific */
/* humidity (lbw/lba) data to create a weather data file consisting of */
/* a decimal date date stamp and hourly dry bulb temperature, relative */
/* humidity and specific humidity */
/*  */
/* Usage: */
/* DOE-WEA.C <pressure> <temperature> <humidity> <outfile> */
/* */
/* <pressure>    is the building's standard atmospheric pressure (psia) */
/* <temperature> is the hourly dry bulb temperature data file */
/* <humidity>    is the hourly specific humidity data file */
/* <outfile>     is the name of the output data file */
/*  */
/* History: */
/*      Design: Doug Bronson */
/*      Code: Doug Bronson */
/* */
/* Distribution Rights */
/*      DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept., */
/*              Texas A & M University., College Station, Texas 77843-3123, */
/*              (409)- 845-1560 */
/*      SUPPORTED BY: State of Texas Governor's Energy Management Center */
/* */
/* COPYRIGHT NOTICE: This program bears a copyright notice to prevent */
/*    rights from being claimed by any other party. Texas A & M University */
/*    intends that the program be placed in the public domain and grants */
/*    permission for it to be used and redistributed, provided that: */
/*        1) the source code is distributed  */
/*        2) this notice is retained in all copies of the source code, and */
/*        3) the program is not sold for profit without written approval */
/*           from TEES. */
/*   The program is distributed "as is". TEES provides no warranty or  */
/*   support service unless special arrangements have been made to do so. */
/*   Certain manufacturers and trade names are mentioned in this code for */
/*   the purpose of describing their communications protocol. This does */
/*   not constitute an endorsement or recommendation of such equipment, */
/*   but is provided for informantional purposes only. */
/* */
/* ********************************************************************* */
#include <stdio.h>
#include <math.h>
/* doe-wea.c uses hourly dry-bulb temperature and specific humidity */
/* and calculates the relative humidity. The output file is decimal date,*/
/* dry bulb temperature, relative humidity, and specific humidity */
/* input form */
/* doe-wea */
/* 1 station atmosphere pressure in psia */
/* 2 dry-bulb temperature - decimal date, data column */
/* 3 specific humidity - decimal date, data column */
/* 4 output file name */
```

Figure A.20 Hard-copy of the DOE-WEA.C Routine


```
main (argc,argv)
      int argc;
      char *argv[];

{
FILE *file1;
FILE *file2;
FILE *comfile;
float date1,val1,date2,val2,rh,comdate,pw,pws;
double c1,c2,c3,c4,c5,c6,c7,c8,c9,c10,c11,c12,c13,p,T,K1,K2;
double d1,d2,d3,d4,d5,d6,d7;

c1 = -1.021416462e4; c2 = -4.89350301; c3 = -5.37657944e-3;
c4 = 1.92023769e-7; c5 = 3.55758316e-10; c6 = -9.03446883e-14;
c7 = 4.1635019; c8 = -1.044039708e4; c9 = -1.12946496e1;
c10 = -2.7022355e-2; c11 = 1.2890360e-5; c12 = -2.478068e-9;
c13 = 6.5459673;

p          = atof(argv[1]);
file1      = fopen(argv[2],"r");
file2      = fopen(argv[3],"r");
comfile    = fopen(argv[4],"w");

/* should check each file* != NULL */
while(!feof(file1)&&!feof(file2))
{
fscanf(file1,"%f %f\n",&date1,&val1);
fscanf(file2,"%f %f\n",&date2,&val2);
#
if(date1!=date2)
{
printf("Error -- file dates mismatch.\n");
printf("Terminating operation now\n");
exit(1);
}
#
comdate=date1;
#
T = val1 + 459.67;
#
if (val1 < 32) {
d1 = pow(T,2.); d2 = pow(T,3.); d3 = pow(T,4.); d4 = log(T);
K1 = c1/T+c2+c3*T+c4*d1+c5*d2+c6*d3+c7*d4;
pws = exp(K1);
}
else {
d5 = pow(T,2.); d6 = pow(T,3.); d7 = log(T);
K2 = c8/T+c9+c10*T+c11*d5+c12*d6+c13*d7;
pws = exp(K2);
}
#
pw = (p*val2)/(0.61298+val2);
#
rh = (pw/pws)*100.;
if (rh >= 100.0) rh = 99.99;
#
fprintf (comfile,"%10.4f %6.2f %8.4f %6.2f\n",comdate, val1, val2, rh);
}
}
```

Figure A.20 Continued

Decimal Date	Dry Bulb Temperature (F)	Specific Humidity (lbw/lba)	Relative Humidity (%)
3531.0833	82.00	0.0181	77.07
3531.1250	81.00	0.0184	80.89
3531.1667	80.00	0.0195	88.42
3531.2083	79.00	0.0198	92.72
3531.2500	79.00	0.0198	92.72
3531.2917	79.00	0.0198	92.72
3531.3333	79.00	0.0207	96.80
3531.3750	80.00	0.0205	92.81
3531.4167	83.00	0.0198	81.40
3531.4583	87.00	0.0179	64.94
3531.5000	89.00	0.0156	53.32
3531.5417	92.00	0.0132	41.23
3531.5833	93.00	0.0121	36.70
3531.6250	95.00	0.0117	33.39
3531.6667	96.00	0.0106	29.39
3531.7083	97.00	0.0096	25.85
3531.7500	97.00	0.0096	25.85
3531.7917	97.00	0.0096	25.85
3531.8333	95.00	0.0101	28.89
3531.8750	93.00	0.0113	34.32
3531.9167	90.00	0.0137	45.51
3531.9583	88.00	0.0158	55.71
3532.0000	86.00	0.0172	64.49
3532.0417	84.00	0.0186	74.18
3532.0833	83.00	0.0188	77.41
3532.1250	82.00	0.0200	84.90
3532.1667	81.00	0.0202	88.55
3532.2083	80.00	0.0214	96.74
3532.2500	80.00	0.0214	96.74
3532.2917	79.00	0.0217	99.99
3532.3333	79.00	0.0217	99.99
3532.3750	80.00	0.0224	99.99
3532.4167	84.00	0.0205	81.51
3532.4583	87.00	0.0179	64.94
3532.5000	90.00	0.0154	51.02
3532.5417	93.00	0.0130	39.38
3532.5833	95.00	0.0117	33.39
3532.6250	96.00	0.0106	29.39
3532.6667	98.00	0.0094	24.57
3532.7083	99.00	0.0092	23.33
3532.7500	99.00	0.0099	25.08
3532.7917	98.00	0.0094	24.57
3532.8333	96.00	0.0098	27.20
3532.8750	94.00	0.0103	30.38
3532.9167	91.00	0.0126	40.64
3532.9583	89.00	0.0156	53.32
3533.0000	87.00	0.0179	64.94

Figure A.21 Two Day Example of the Weather Data File


```
/* ! /usr/local/bin */
/* %W% %G% */
/* ******************************************************************** */
/* Copyright (c) 1990, Texas Enginnering Experiment Station */
/* */
/* Program: ADD-LOAD.C  */
/* Version: %I% */
/* Last Update: %G% */
/*  */
/* Description: */
/* This routine merges hourly cooling, heating, and electricity consumption */
/* data to the weather data file created by DOE-WEA.C */
/*  */
/* Usage: */
/* add-load <weather> <cooling> <heating> <electricity> <outfile> */
/* */
/* <weather>     is the hourly weather data file generated by DOE-WEA.C */
/* <cooling>     is the hourly cooling load energy consumption data file */
/* <heating>     is the hourly heating load energy consumption data file */
/* <electricity> is the hourly whole-building electricity consumption data */
/*                file */
/* <outfile>     is the name of outfile data file */
/*  */
/* History:  */
/*        Design: Doug Bronson */
/*          Code: Doug Bronson */
/*  */
/* Distribution Rights */
/*     DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept., */
/*            Texas A & M University., College Station, Texas 77843-3123, */
/*            (409)- 845-1560 */
/*     SUPPORTED BY: State of Texas Governor's Energy Management Center */
/*  */
/* COPYRIGHT NOTICE: This program bears a copyright notice to prevent */
/*     rights from being claimed by any other party. Texas A & M University */
/*     intends that the program be placed in the public domain and grants */
/*     permission for it to be used and redistributed, provided that: */
/*         1) the source code is distributed */
/*         2) this notice is retained in all copies of the source code, and */
/*         3) the program is not sold for profit without written approval */
/*            from TEES. */
/*    The program is distributed "as is". TEES provides no warranty or */
/*    support service unless special arrangements have been made to do so. */
/*    Certain manufacturers and trade names are mentioned in this code for */
/*    the purpose of describing their communications protocol. This does */
/*    not constitute an endorsement or recommendation of such equipment, */
/*    but is provided for informantional purposes only. */
/*  */
/* ******************************************************************** */
#include <stdio.h>
/* add-load.c attachs the hourly cooling, heating, and electricity */
/* consumption to the output file from doe-wea.c */
/* input format */
/* add-loads */
/* 1 output file from doe-wea.c */
/* 2 hourly cooling load data file, decimal date, data column */
/* 3 hourly heating load data file, decimal date, data column */
/* 4 hourly electrical load data file, decimal date, data column */
/* 5 output file name */
```

Figure A.22 Hard-copy of the ADD-LOAD.C Routine


```
main (argc,argv)
      int argc;
      char *argv[];
{
FILE *file1;
FILE *file2;
FILE *file3;
FILE *file4;
FILE *comfile;
float date1,val1a,val1b,val1c,date2,val2,date3,val3,date4,val4,comdate;

file1      = fopen(argv[1],"r");
file2      = fopen(argv[2],"r");
file3      = fopen(argv[3],"r");
file4      = fopen(argv[4],"r");
comfile    = fopen(argv[5],"w");

/* should check each file* != NULL */
while(!feof(file1)&&!feof(file2)&&!feof(file3)&&!feof(file4))
{
fscanf(file1,"%f %f %f %f\n",&date1,&val1a,&val1b,&val1c);
fscanf(file2,"%f %f\n",&date2,&val2);
fscanf(file3,"%f %f\n",&date3,&val3);
fscanf(file4,"%f %f\n",&date4,&val4);
#
if(date1!=date2||date1!=date3||date2!=date3||date1!=date4||date2!=date4||date3
!=date4)
{
printf("Error -- file dates mismatch.\n");
printf("Terminating operation now\n");
exit(1);
}
#
comdate=date1;
fprintf(comfile,"%10.4f %6.2f %6.4f %6.2f %6.2f %6.2f
%6.2f\n",comdate,val1a,val1b,val1c,val2,val3,val4);
}
}
```

Figure A.22 Continued

Decimal Date	Dry Bulb Temperature (F)	Specific Humidity (lbw/lba)	Relative Humidity (%)	Cooling Load (MMBtu/hr)	Heating Load (MMBtu/hr)	Electricity Consumption (kWh/h)
3531.0833	82.00	0.0181	77.07	8.10	0.08	1101.33
3531.1250	81.00	0.0184	80.89	8.04	0.08	1091.35
3531.1667	80.00	0.0195	88.42	8.27	0.09	1085.35
3531.2083	79.00	0.0198	92.72	7.97	0.10	1081.36
3531.2500	79.00	0.0198	92.72	7.99	0.11	1084.36
3531.2917	79.00	0.0198	92.72	7.85	0.09	1117.31
3531.3333	79.00	0.0207	96.80	8.59	0.04	1201.19
3531.3750	80.00	0.0205	92.81	8.55	0.00	1350.97
3531.4167	83.00	0.0198	81.40	8.61	0.00	1417.87
3531.4583	87.00	0.0179	64.94	8.54	0.00	1445.83
3531.5000	89.00	0.0156	53.32	7.86	0.00	1452.82
3531.5417	92.00	0.0132	41.23	7.52	0.00	1440.84
3531.5833	93.00	0.0121	36.70	7.55	0.00	1451.82
3531.6250	95.00	0.0117	33.39	7.67	0.00	1464.80
3531.6667	96.00	0.0106	29.39	7.54	0.00	1455.82
3531.7083	97.00	0.0096	25.85	7.46	0.00	1419.87
3531.7500	97.00	0.0096	25.85	7.45	0.00	1299.04
3531.7917	97.00	0.0096	25.85	7.43	0.00	1236.14
3531.8333	95.00	0.0101	28.89	7.48	0.00	1218.16
3531.8750	93.00	0.0113	34.32	7.59	0.01	1215.17
3531.9167	90.00	0.0137	45.51	7.95	0.04	1198.19
3531.9583	88.00	0.0158	55.71	8.00	0.06	1173.23
3532.0000	86.00	0.0172	64.49	8.05	0.07	1146.27
3532.0417	84.00	0.0186	74.18	8.14	0.07	1117.31
3532.0833	83.00	0.0188	77.41	7.99	0.08	1073.37
3532.1250	82.00	0.0200	84.90	8.04	0.09	1067.38
3532.1667	81.00	0.0202	88.55	8.30	0.09	1062.39
3532.2083	80.00	0.0214	96.74	8.61	0.10	1059.39
3532.2500	80.00	0.0214	96.74	8.49	0.10	1058.39
3532.2917	79.00	0.0217	99.99	8.35	0.11	1056.40
3532.3333	79.00	0.0217	99.99	8.01	0.13	1046.41
3532.3750	80.00	0.0224	99.99	8.62	0.13	1052.40
3532.4167	84.00	0.0205	81.51	8.32	0.09	1070.38
3532.4583	87.00	0.0179	64.94	7.77	0.08	1090.35
3532.5000	90.00	0.0154	51.02	7.38	0.08	1104.33
3532.5417	93.00	0.0130	39.38	6.90	0.10	1113.31
3532.5833	95.00	0.0117	33.39	6.96	0.11	1124.30
3532.6250	96.00	0.0106	29.39	6.89	0.12	1130.29
3532.6667	98.00	0.0094	24.57	6.86	0.12	1135.28
3532.7083	99.00	0.0092	23.33	7.04	0.10	1134.28
3532.7500	99.00	0.0099	25.08	7.34	0.09	1125.30
3532.7917	98.00	0.0094	24.57	7.12	0.09	1119.31
3532.8333	96.00	0.0098	27.20	7.11	0.10	1117.31
3532.8750	94.00	0.0103	30.38	7.10	0.11	1122.30
3532.9167	91.00	0.0126	40.64	7.60	0.11	1121.30
3532.9583	89.00	0.0156	53.32	7.90	0.12	1113.31
3533.0000	87.00	0.0179	64.94	8.10	0.12	1101.33

Figure A.23 Two Day Example of the Weather and Energy Consumption Data File

```
# ! /usr/local/bin/nawk -f
# %W% %G%
# **************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: DAILY-AV.AWK
# Version: %I%
# Last Update: %G%
#
# Description:
# This program creates a daily average data file from hourly data
#
# Usage:
#  daily-av <source> <outfile>
#
# <source> is the hourly data file.
# <output> is the name of the daily average data file.
#
# History:
#       Design: Doug Bronson
#       Code: Doug Bronson
#
# Distribution Rights
#       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#               Texas A & M University., College Station, Texas 77843-3123,
#               (409)- 845-1560
#       SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#       from being claimed by any other party. Texas A & M University intends
#       that the program be placed in the public domain and grants
#       permission for it to be used and redistributed, provided that:
#            1) the source code is distributed
#            2) this notice is retained in all copies of the source code, and
#            3) the program is not sold for profit without written approval
#               from TEES.
#       The program is distributed "as is". TEES provides no warranty or
#       support service unless special arrangements have been made to do so.
#       Certain manufacturers and trade names are mentioned in this code for
#       the purpose of describing their communications protocol. This does
#       not constitute an endorsement or recommendation of such equipment,
#       but is provided for informantional purposes only.
#
#**************************************************************************
# DAILY-AV.AWK sums hourly data into daily average data files.
#
# input format
# gawk -f daily-av.awk
#  1 source file with extension - decimal date, data column
#  2 output file containing daily averages
#
BEGIN {
      daily_ave_file = ARGV[ARGC-1];
      start = 1;
#
      ARGC = 2;
}
#
```

Figure A.24 Hard-copy of the DAILY-AV.AWK Routine

```
{
#
if (start == 1) {
      L_DECI_TIME = int($1);
      DAY_SUM = 0;
      start = 0;
}
HH = $1 - int($1);
if (HH == 0.0) $1 -= 1;
#
if ($2 != -99.) {
#
      if (int($1) == L_DECI_TIME) {
      DAY_SUM += $2;
      }
      else {
      printf ("%-9.4f %-6.2f\n", L_DECI_TIME, DAY_SUM/24.) > daily_ave_file;
      DAY_SUM = $2;
      }
#
}
L_DECI_TIME = int($1);
F_DECI_TIME = int($1);
}
#
END {
#
if (daily_average == "y") {
      printf ("%-6.4f %-10.4f\n", F_DECI_TIME, DAY_SUM/24.) > daily_ave_file;
}
      close(daily_ave_file);
}
#
# END
```

Figure A.24 Continued

Decimal Date	Daily-Average Dry Bulb Temperature (F)
3531.0000	84.04
3532.0000	88.50

Figure A.25 Two Day Example of Daily-Averaged Outdoor Dry Bulb Temperature


```
/* ! /usr/local/bin */
/* %W% %G% */
/* ******************************************************************** */
/* Copyright (c) 1990, Texas Enginnering Experiment Station */
/* */
/* Program: LDS-PROF.C  */
/* Version: %I% */
/* Last Update: %G% */
/*  */
/* Description: */
/* This routine generates daily-average load profiles. The hourly cooling, */
/* heating, and electricity consumption data is segregated into weekend and */
/* weekday bins and bins with a daily-averaged dry bulb temperature less */
/* than and greater than 60 F. */
/*  */
/* Usage: */
/* lds-prof <daily-average> <source> <outfile> */
/* */
/* <daily-average> is the daily-averaged dry bulb temperature file */
/* <source>      is weather and energy consumption data file */
/* <outfile>     is the name of the outfile data file */
/* */
/* History:  */
/*       Design: Doug Bronson */
/*         Code: Doug Bronson */
/*  */
/* Distribution Rights */
/*     DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept., */
/*            Texas A & M University., College Station, Texas 77843-3123, */
/*            (409)- 845-1560 */
/*     SUPPORTED BY: State of Texas Governor's Energy Management Center */
/*  */
/* COPYRIGHT NOTICE: This program bears a copyright notice to prevent */
/*     rights from being claimed by any other party. Texas A & M University */
/*     intends that the program be placed in the public domain and grants */
/*     permission for it to be used and redistributed, provided that: */
/*         1) the source code is distributed */
/*         2) this notice is retained in all copies of the source code, and */
/*         3) the program is not sold for profit without written approval */
/*            from TEES. */
/*    The program is distributed "as is". TEES provides no warranty or */
/*    support service unless special arrangements have been made to do so. */
/*    Certain manufacturers and trade names are mentioned in this code for */
/*    the purpose of describing their communications protocol. This does */
/*    not constitute an endorsement or recommendation of such equipment, */
/*    but is provided for informantional purposes only. */
/*  */
/* ******************************************************************** */
#include <stdio.h>
#include <math.h>

/* LDS-PROF.C creates average daily load profiles for chilled & hot water */
/* consumption and electricity consumption for weekdays and weekends & */
/* vacation and for daily average temperature above and below 60f */

/* input form */
/* lds-prof */
/* 1 daily average temperature file */
/* 2 hourly weather and energy data file - output from add-loads.c */
/* 3 average daily load profile file */
```

Figure A.26 Hard-copy of the LDS-PROF.C Routine

```
main (argc,argv)
      int argc;
      char *argv[];
{
FILE *oa_file;
FILE *grp_file;
FILE *lds_file;

int i,j,k,hr,day,we;
float oa_date,grp_date,daily_temp,hr_temp,hr_hum,hr_rh,cw_data,hw_data,
ele_data,average_load;
float cw[25][2][3],cwcount[25][2][3],hw[25][2][3],hwcount[25][2][3],
ele[25][2],elecount[25][2];

for (i=0;i<25;i++) for (j=0;j<2;j++) {
      ele[i][j] = elecount[i][j] = 0.0;
      for (k=0;k<3;k++) cw[i][j][k] = cwcount[i][j][k] = hw[i][j][k] =
hwcount[i][j][k] = 0.0;
}

oa_file = fopen(argv[1],"r");
grp_file = fopen(argv[2],"r");
lds_file = fopen(argv[3],"w");

while(!feof(oa_file) && !feof(grp_file)) {

fscanf(oa_file,"%f %f\n",&oa_date,&daily_temp);

day = (int)oa_date % 7 + 2;

if (day > 7) day -= 7;

if ((oa_date >= 3614. && oa_date < 3616.) || (oa_date >= 3642. && oa_date <
3654.) || (day > 5)) we = 1;
else we = 0;

do {
fscanf(grp_file,"%f %f %f %f %f %f
%f\n",&grp_date,&hr_temp,&hr_hum,&hr_rh,&cw_data,&hw_data,&ele_data);

hr = ((grp_date - (int)grp_date)*24. + 0.1);
if (hr == 0) {
      hr = 24;
      grp_date -= 1.0;
}

if ((int)grp_date != oa_date) {
      printf("Error -- file dates mismatch with OA file\n");
      printf("Terminating operation now\n");
      exit(1);
}

if (cw_data != -99.) {
      if (we == 1) {
            if (daily_temp < 60.) {
                  cw[hr][we][1] += cw_data;
                  cwcount[hr][we][1] += 1.;
            }
```

Figure A.26 Continued


```
            else {
                  cw[hr][we][2] += cw_data;
                  cwcount[hr][we][2] += 1.;
            }
      }
      else {
            if (daily_temp < 60.) {
                  cw[hr][we][1] += cw_data;
                  cwcount[hr][we][1] += 1.;
            }
            else {
                  cw[hr][we][2] += cw_data;
                  cwcount[hr][we][2] += 1.;
            }
      }
}

if (hw_data != -99.) {
      if (we == 1) {
            if (daily_temp < 60.) {
                  hw[hr][we][1] += hw_data;
                  hwcount[hr][we][1] += 1.;
            }
            else {
                  hw[hr][we][2] += hw_data;
                  hwcount[hr][we][2] += 1.;
            }
      }
      else {
            if (daily_temp < 60.) {
                  hw[hr][we][1] += hw_data;
                  hwcount[hr][we][1] += 1.;
            }
            else {
                  hw[hr][we][2] += hw_data;
                  hwcount[hr][we][2] += 1.;
            }
      }
}

if (ele_data != -99.) {
      if (we == 1) {
            ele[hr][we] += ele_data;
            elecount[hr][we] += 1.;
      }
      else {
            ele[hr][we] += ele_data;
            elecount[hr][we] += 1.;
      }
}

} while ((hr < 24) && !feof(grp_file));

}

for (k = 1;k <= 24; k++) {

if (hwcount[k][0][1] == 0) average_load = 0.0;
else average_load = hw[k][0][1]/hwcount[k][0][1];
fprintf(lds_file,"%2d %5.2f",k,average_load);
```

Figure A.26 Continued


```
if (hwcount[k][0][2] == 0) average_load = 0.0;
else average_load = hw[k][0][2]/hwcount[k][0][2];
fprintf(lds_file," %5.2f",average_load);

if (hwcount[k][1][1] == 0) average_load = 0.0;
else average_load = hw[k][1][1]/hwcount[k][1][1];
fprintf(lds_file," %5.2f",average_load);

if (hwcount[k][1][2] == 0) average_load = 0.0;
else average_load = hw[k][1][2]/hwcount[k][1][2];
fprintf(lds_file," %5.2f",average_load);

if (cwcount[k][0][1] == 0) average_load = 0.0;
else average_load = cw[k][0][1]/cwcount[k][0][1];
fprintf(lds_file," %5.2f",average_load);

if (cwcount[k][0][2] == 0) average_load = 0.0;
else average_load = cw[k][0][2]/cwcount[k][0][2];
fprintf(lds_file," %5.2f",average_load);

if (cwcount[k][1][1] == 0) average_load = 0.0;
else average_load = cw[k][1][1]/cwcount[k][1][1];
fprintf(lds_file," %5.2f",average_load);

if (cwcount[k][1][2] == 0) average_load = 0.0;
else average_load = cw[k][1][2]/cwcount[k][1][2];
fprintf(lds_file," %5.2f",average_load);

if (elecount[k][0] == 0) average_load = 0.0;
else average_load = (ele[k][0]*0.0034129)/elecount[k][0];
fprintf(lds_file," %5.2f",average_load);

if (elecount[k][1] == 0) average_load = 0.0;
else average_load = (ele[k][1]*0.0034129)/elecount[k][1];
fprintf(lds_file," %5.2f\n",average_load);

}
}
```

Figure A.26 Continued

HR	Heating Profiles Weekday <60	Heating Weekday >60	Heating Weekend <60	Heating Weekend >60	Cooling Profiles Weekday <60	Cooling Weekday >60	Cooling Weekend <60	Cooling Weekend >60	Electricity WD	Electricity WE
1	3.53	1.40	4.10	1.58	4.10	5.50	3.88	5.25	3.61	3.45
2	3.64	1.45	4.16	1.66	4.07	5.49	3.86	5.25	3.56	3.41
3	3.72	1.52	4.24	1.70	4.03	5.44	3.82	5.23	3.53	3.39
4	3.81	1.58	4.31	1.77	3.99	5.41	3.78	5.21	3.51	3.38
5	3.89	1.64	4.37	1.82	3.96	5.39	3.77	5.20	3.50	3.37
6	3.98	1.67	4.43	1.85	3.92	5.38	3.75	5.19	3.51	3.37
7	3.91	1.58	4.54	1.90	3.94	5.53	3.70	5.17	3.62	3.36
8	3.74	1.42	4.60	1.94	4.01	5.73	3.68	5.12	3.89	3.33
9	3.50	1.18	4.59	1.89	4.07	5.93	3.70	5.18	4.36	3.33
10	3.31	1.00	4.51	1.73	4.15	6.10	3.75	5.29	4.60	3.40
11	3.09	0.86	4.35	1.56	4.22	6.13	3.80	5.22	4.71	3.47
12	2.93	0.76	4.23	1.44	4.27	6.08	3.83	5.22	4.73	3.51
13	2.78	0.69	4.09	1.35	4.33	6.11	3.86	5.27	4.69	3.53
14	2.64	0.63	3.95	1.27	4.37	6.16	3.89	5.36	4.73	3.56
15	2.53	0.58	3.85	1.20	4.43	6.15	3.92	5.45	4.77	3.58
16	2.49	0.56	3.78	1.15	4.44	6.18	3.94	5.50	4.74	3.58
17	2.51	0.56	3.76	1.12	4.43	6.20	3.94	5.50	4.61	3.57
18	2.74	0.70	3.79	1.13	4.32	6.05	3.94	5.53	4.21	3.54
19	2.94	0.82	3.82	1.16	4.24	5.96	3.94	5.50	4.00	3.53
20	3.08	0.90	3.86	1.22	4.20	5.87	3.95	5.46	3.94	3.53
21	3.19	0.96	3.90	1.26	4.16	5.81	3.94	5.43	3.93	3.54
22	3.31	1.04	3.93	1.30	4.13	5.74	3.94	5.40	3.88	3.53
23	3.44	1.16	3.99	1.35	4.08	5.65	3.92	5.44	3.79	3.51
24	3.61	1.26	4.04	1.40	4.01	5.60	3.91	5.42	3.71	3.48

Figure A.27 Example of the LDSPF.DAT Graph Data File


```
/* CHW.SAS creates a linear regression model of the data and creates a */
/* load versus dry-bulb temperature X-Y plot. Input is the output from */
/* add-loads.c copied to input.dat */
/* input form */
/* sas chw.sas */

  options linesize=80 pagesize=60 number;
  proc printto new print='/u/jdbronso/doeinp/zachry/chwsas.out';

  data indata ;
       infile '/u/jdbronso/doeinp/zachry/input.dat' eof=next;
       input time tdb humidity rh chw htw elec;
       cw_water = chw*1000000;
       array a tdb chw;
         do over a;
          if a = -99.0 then delete;
         end;
       output;

  next: proc reg data=indata;
         model cw_water = tdb;
         output out=cwplot p=cwp r=cwr;

  proc plot data=cwplot;
     plot cw_water*tdb / vaxis=0 to 10000000 by 2000000 haxis=0 to 120 by 20;
```

```
/* HTW.SAS creates a linear regression model and creates a load versus */
/* dry-bulb temperature X-Y plot. The input is the output from add-loads.c */
/* to input.dat */
/* input form */
/* sas htw.sas */

  options linesize=80 pagesize=60 number;
  proc printto new print='/u/jdbronso/doeinp/zachry/htwsas.out';

  data indata ;
       infile '/u/jdbronso/doeinp/zachry/input.dat' eof=next;
       input time tdb humidity rh cw htw elec;
       ht_water = htw*1000000;
       array a tdb htw;
         do over a;
          if a = -99.0 then delete;
         end;
      if tdb > 82.0 then delete;
       output;

  next: proc reg data=indata;
         model ht_water = tdb;
         output out=htplot p=htp r=htr;

  proc plot data=htplot;
     plot ht_water*tdb / vaxis=0 to 10000000 by 2000000 haxis=0 to 120 by 20;
```

Figure A.28 Hard-copies of the CHW.SAS and HTW.SAS

```
                The SAS System 19:19 Tuesday, March 3, 1992   1

Model: MODEL1
Dependent Variable: CW_WATER

                         Analysis of Variance

                           Sum of          Mean
   Source          DF     Squares        Square       F Value       Prob>F

   Model            1 4.5973719E15 4.5973719E15     11159.064       0.0001
   Error         4341 1.7884288E15 411985443158
   C Total       4342 6.3858007E15

       Root MSE  641860.92197     R-square          0.7199
       Dep Mean 4930874.97122     Adj R-sq          0.7199
       C.V.          13.01718

                         Parameter Estimates

                      Parameter       Standard    T for H0:
 Variable  DF          Estimate          Error   Parameter=0    Prob > |T|

 INTERCEP   1           1017599   38303.718917        26.567        0.0001
 TDB        1             64146   607.23126311       105.636        0.0001
```

Figure A.29 Example of the Chilled Water Consumption SAS Output


```
                    The SAS System 19:19 Tuesday, March 3, 1992   2

          Plot of CW_WATER*TDB.  Legend: A = 1 obs, B = 2 obs, etc.

CW_WATER |
         |
10000000 +
         |
         |
         |
         |
         |
         |                                            BBAB
         |                                            CBDB B
         |                                          ACBABFBA A
 8000000 +                                          DGHBJFCCAAA
         |                                          BFMECEFEAC BA
         |                                          DMEEHDDCBDJDDA
         |                                        BDGIDACEDEBDCDFB
         |                                         HNOCEDEFCBCCCCDA
         |                                       EIFMHFABDCAAADCDB
         |                                      FCDJIJBADCAA BDCA
         |                                      FGGHJCFBFJARKIA
         |                                     BJEOIFDCBFIUUAA A
 6000000 +                                     ITEWMKIGBQYJEAAC
         |                                  AEFIVLRJIFNQYGFF
         |                                 GDFGOZHOPLYSEEIF
         |                               AGF IQZZLUPYOHEFA
         |                              BDEECOYZZRZZTNUDA
         |                           ABAEHJSPQTZZSZZLG
         |                          FTVZZZZZWZZZZNQKD
         |                       AEKZZXZZZZZQZZXICAA
         |                   ABBCCZZZILJEWZZQYWLDC
 4000000 +                   DDBLBNDPKILZZZZGGEECA
         |                  AFBACFLGQZRZZNFDCIDC
         |                  JJQFEBFZWYNLMDAB
         |                EJZNJDFIQZMFA
         |         AAAACCBADCBBKRKKCA
         |        CCIKBECAA EJJIC
         |     BBHDDE     BAAFA
         |   DBA          E DA
         |                A
 2000000 +
         |
         |
         |
         |
         |
         |
         |
         |
       0 +
         |
         --+----------+----------+----------+----------+----------+-----------
           0         20         40         60         80        100

                                         TDB

NOTE: 516 obs hidden.
```

Figure A.29 Continued

```
# ! /usr/local/bin/gawk -f
# %W% %G%
# ***************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: SAS-EQU.AWK
# Version: %I%
# Last Update: %G%
#
# Description:
# This routines extracts the slope and y-intercept values from the SAS output
# file and uses them to generate a data file representing the linear
# regression models of the cooling and heating load versus outdoor
# dry bulb temperature
#
# Usage:
# sas-equ.awk <sas output> <output>
#
# <sas output>  is the output file from CHW.SAS and HTW.SAS routines.
# <output>      is the name of the output data file.
#
# History:
#       Design: Doug Bronson
#       Code: Doug Bronson
#
# Distribution Rights
#       DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#               Texas A & M University., College Station, Texas 77843-3123,
#               (409)- 845-1560
#       SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#       from being claimed by any other party. Texas A & M University intends
#       that the program be placed in the public domain and grants
#       permission for it to be used and redistributed, provided that:
#            1) the source code is distributed
#            2) this notice is retained in all copies of the source code, and
#            3) the program is not sold for profit without written approval
#               from TEES.
#       The program is distributed "as is". TEES provides no warranty or
#       support service unless special arrangements have been made to do so.
#       Certain manufacturers and trade names are mentioned in this code for
#       the purpose of describing their communications protocol. This does
#       not constitute an endorsement or recommendation of such equipment,
#       but is provided for informantional purposes only.
#
# ***************************************************************************
# SAS-EQU.AWK generates the linear regression line from the SAS runs
#
# input format
# nawk -f sas-equ.awk
#  1  source file with extension
#  2  outfile with extension
#
BEGIN {
outfile = ARGV[2];
ARGC = 2;
}
#
```

Figure A.30 Hard-copy of the SAS-EQU.AWK Routine


```
{
if ($1 == "INTERCEP") {
intercep = $3;
getline;
tdb = $3;
#
for (temp = 0; temp <= 120.0; temp++) {
load = (intercep + tdb*temp)/1000000;
printf("%5.1f %6.2f\n",temp,load) > outfile;
}
}
}
#
END {
close(outfile)
}
```

Figure A.30 Continued

Dry Bulb Temperature (F)	*Cooling Load (MMBtu/hr)*
0.0	1.02
1.0	1.08
2.0	1.15
3.0	1.21
4.0	1.27
5.0	1.34
6.0	1.40
7.0	1.47
8.0	1.53
9.0	1.59
10.0	1.66
11.0	1.72
12.0	1.79
13.0	1.85
14.0	1.92
15.0	1.98
16.0	2.04
17.0	2.11
18.0	2.17
19.0	2.24
20.0	2.30
21.0	2.36
22.0	2.43
23.0	2.49
24.0	2.56
25.0	2.62
.	
.	
.	
.	
95.0	7.11
96.0	7.18
97.0	7.24
98.0	7.30
99.0	7.37
100.0	7.43
101.0	7.50
102.0	7.56
103.0	7.62
104.0	7.69
105.0	7.75
106.0	7.82
107.0	7.88
108.0	7.95
109.0	8.01
110.0	8.07
111.0	8.14
112.0	8.20
113.0	8.27
114.0	8.33
115.0	8.39
116.0	8.46
117.0	8.52
118.0	8.59
119.0	8.65
120.0	8.72

Figure A.31 Example of the Chilled Water Consumption Linear Regression Graph Data File


```
# ! /usr/local/bin/nawk -f
# %W% %G%
# ************************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: HISGEN3
# Version: %I%
# Last Update: %G%
#
# Description:
# This routine generates the graph data files for the hourly dry bulb
# temperature and specific humidity histograms.
#
# Usage:
# hisgen3 <source>
#
# <source>      is the weather and energy consumption data file.
#
# History:
#    Design: Doug Bronson
#      Code: Doug Bronson
#
# Distribution Rights
#     DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#            Texas A & M University., College Station, Texas 77843-3123,
#            (409)- 845-1560
#     SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#    from being claimed by any other party. Texas A & M University intends
#    that the program be placed in the public domain and grants
#    permission for it to be used and redistributed, provided that:
#          1) the source code is distributed
#          2) this notice is retained in all copies of the source code, and
#          3) the program is not sold for profit without written approval
#             from TEES.
#    The program is distributed "as is". TEES provides no warranty or
#    support service unless special arrangements have been made to do so.
#    Certain manufacturers and trade names are mentioned in this code for
#    the purpose of describing their communications protocol. This does
#    not constitute an endorsement or recommendation of such equipment,
#    but is provided for informantional purposes only.
#
# ************************************************************************
#!/usr/local/bin/nawk -f
# This program read dry-bulb temperature (2 col) and specific humidity (3 col)
# and generates 5 F degree bins between 0 F and 120 F.
# It generates 0.0025 lbw/lba bins between 0.0025 and 0.025
# lbw/lba and one bin for specific humidities below 0.0025 lbw/lba. It will
# flag a error message if either the temperature or the specific humidity is
# out of range. If there if missing data marked with -99.0, for temperature
# the count will be placed in the 115-120 F temperature bin and for humidity
# the count will be placed in the 0.0200-0.0250 (lbw/lba) specific humidity
# bin.

# It then generates two data files (tmp.dat and w.dat) which when used
# along with tmp.grf and w.grf, respectively will produce two historgrams

# ver 1.0 dry-bulb temperature scale 20 - 120 F
# ver 1.1 dry-bulb temperature scale 0 - 110 F
```

Figure A.32 Hard-copy of the HISGEN3 Routine


```
# usage
# nawk -f hisgen3
# 1 source file - output from doe-wea.c or adds-loads.c

BEGIN{
       for (i=1; i<=17; i++) {
         count[i] = 0;
         wcount[i] = 0;
       }
}
{
    if (($2 >= -20.0) && ($2 <= 5))         count[1]++;

    else if (($2 > 5) && ($2 <= 10))        count[2]++;

    else if (($2 > 10) && ($2 <= 15))       count[3]++;

    else if (($2 > 15) && ($2 <= 20))       count[4]++;

    else if (($2 > 20) && ($2 <= 25))       count[5]++;

    else if (($2 > 25) && ($2 <= 30))       count[6]++;

    else if (($2 > 30) && ($2 <=35))        count[7]++;

    else if (($2 > 35) && ($2 <= 40))       count[8]++;

    else if (($2 > 40) && ($2 <= 45))       count[9]++;

    else if (($2 > 45) && ($2 <= 50))       count[10]++;

    else if (($2 > 50) && ($2 <= 55))       count[11]++;

    else if (($2 > 55) && ($2 <= 60))       count[12]++;

    else if (($2 > 60) && ($2 <= 65))       count[13]++;

    else if (($2 > 65) && ($2 <= 70))       count[14]++;

    else if (($2 > 70) && ($2 <= 75))       count[15]++;

    else if (($2 > 75) && ($2 <= 80))       count[16]++;

    else if (($2 > 80) && ($2 <= 85))       count[17]++;

    else if (($2 > 85) && ($2 <= 90))       count[18]++;

    else if (($2 > 90) && ($2 <= 95))       count[19]++;

    else if (($2 > 95) && ($2 <= 100))      count[20]++;

    else if (($2 > 100) && ($2 <= 105))     count[21]++;

    else if (($2 > 105) || ($2 == -99.0)) count[24]++;
```

Figure A.32 Continued

```
    if (($3 >= 0.0) && ($3 <= 0.0025))          wcount[1]++;
    else if (($3 > 0.0025) && ($3 <= 0.005))   wcount[2]++;
    else if (($3 > 0.005) && ($3 <= 0.0075))   wcount[3]++;
    else if (($3 > 0.0075) && ($3 <= 0.0100)) wcount[4]++;
    else if (($3 > 0.010) && ($3 <= 0.0125))   wcount[5]++;
    else if (($3 > 0.0125) && ($3 <= 0.0150)) wcount[6]++;
    else if (($3 > 0.015) && ($3 <= 0.0175))   wcount[7]++;
    else if (($3 > 0.0175) && ($3 <= 0.0200)) wcount[8]++;
    else if (($3 > 0.0200) && ($3 <= 0.0225)) wcount[9]++;
    else if ((($3 > 0.022) && ($3 <= 0.0250)) || ($3 == -99.0)) wcount[10]++;
}
END{
  print "0  " "0"                         >"tmp.dat";
  for (i=1; i<=21; i++) {
    print (0 + (i-1)*5) " " count[i] >"tmp.dat";
    print (0 + i*5) " " count[i]     >"tmp.dat";
    print (0 + i*5) " 0"             >"tmp.dat";
  }
    print "105 " count[24] >"tmp.dat";
    print "110 " count[24] >"tmp.dat";
    print "110  0"         >"tmp.dat";

  print "0  " "0" >"w.dat";
  for (i=1; i<=10; i++) {
    print (i-1)*0.0025 " " wcount[i] >"w.dat";
    print i*0.0025 " " wcount[i] >"w.dat";
    print i*0.0025 " 0" >"w.dat";
  }

}
```

Figure A.32 Continued

Dry Bulb Temperature (F)	*Frequency*
0	0
0	5
5	5
5	0
5	9
10	9
10	0
10	20
15	20
15	0
15	30
20	30
20	0
20	14
25	14
25	0
25	60
30	60
30	0
30	128
35	128
35	0
35	119
40	119
40	0
40	351
45	351
45	0
45	428
.	
.	
.	
.	
75	427
75	0
75	342
80	342
80	0
80	250
85	250
85	0
85	147
90	147
90	0
90	61
95	61
95	0
95	29
100	29
100	0
100	0
105	0
105	0
105	0
110	0
110	0

Figure A.33 Example of the Outdoor Dry Bulb Temperature Histogram Graph Data File

Specific Humidity (lbw/lba)	Frequency
0	0
0	959
0.0025	959
0.0025	0
0.0025	929
0.005	929
0.005	0
0.005	660
0.0075	660
0.0075	0
0.0075	540
0.01	540
0.01	0
0.01	487
0.0125	487
0.0125	0
0.0125	364
0.015	364
0.015	0
0.015	232
0.0175	232
0.0175	0
0.0175	140
0.02	140
0.02	0
0.02	32
0.0225	32
0.0225	0
0.0225	0
0.025	0
0.025	0

Figure A.34 Example of the Outdoor Specific Humidity Histogram Graph Data File


```
@echo off
rem ***********************************************************************
rem Copyright (c) 1990, Texas Engineering Experiment Station
rem
rem Program: DOE-LDS.BAT
rem Version: 1.0
rem Last Update: 06/20/91
rem
rem DESCRIPTION:
rem
rem HISTORY:
rem     Design: Srinivas Katipamula
rem     Code: Srinivas Katipamula
rem
rem MODIFICATIONS:
rem     NAME:                DATE:      VERSION  DESCRIPTION:
rem
rem Input : Start date and End date for the time series plots
rem Usage : DOE-LDS Start Date End Date
rem Output : DOE-LDS.OUT postscript file
rem
rem HISTORY AND DISTRIBUTION RIGHTS
rem     DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
rem         Texas A & M Univ., College Station, Texas 77843-3123,
rem         (409) 845-1560
rem     SUPPORTED BY: State of Texas Governor's Energy Management Center
rem
rem COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
rem     from being claimed by any other party.  Texas A & M University intends
rem     intends that the program be placed in the public domain and grants
rem     permission for it to be used and redistributed, provided that:
rem       1) the source code is distributed,
rem       2) this notice is retained in all copies of the source code, and
rem       3) the program is not sold for profit without written approval from
rem          TEES.
rem     The program is distributed "as is".  TEES provides no warranty or
rem     support service unless special arrangements have been made to do so.
rem     Certain manufacturers and trade names are mentioned in this code for
rem     the purpose of describing their communications protocol.  This does
rem     not constitute an endorsement or recommendation of such equipment,
rem     but is provided for informational purposes only.
rem
rem ***********************************************************************

if "%1"=="" goto error
if "%2"=="" goto error

echo Changing X-AXIS label ....

gawk -f changex tmpts.src %1 %2 >tmpts.grf
gawk -f changex wts.src %1 %2 >wts.grf
gawk -f changex cl-ts.src %1 %2 >cl-ts.grf
gawk -f changex ht-ts.src %1 %2 >ht-ts.grf

echo Generating *.plt files ....

grapher cl-tmp
grapher ht-tmp
grapher cl-ts
grapher ht-ts
grapher w
```

Figure A.35 Hard-copy of the DOE-LDS.BAT Batch File


```
grapher tmp
grapher tmpsh
grapher tmpts
grapher wts
grapher cl-w
grapher ht-w
grapher clldspf
grapher htldspf
grapher elldspf
grapher cwline
grapher hwline

copy m10.tem+htldspf.plt+elldspf.plt+m13.tem+cl-w.plt+ht-w.plt d0.plt
copy m14.tem+clldspf.plt+elldspf.plt d1.plt
copy m12.tem+tmp.plt+m11.tem+tmpsh.plt+eng-psy.plt d2.plt
copy m11.tem+cl-tmp.plt+ht-tmp.plt+cwline.plt+hwline.plt d3.plt
copy accwline.plt+achwline.plt d4.plt
copy m12.tem+tmpts.plt+wts.plt+m13.tem+w.plt d5.plt
copy m14.tem+cl-ts.plt+ht-ts.plt d6.plt
copy d0.plt+d1.plt+d2.plt+d3.plt+d4.plt+d5.plt+d6.plt doe-lds.plt

rem view doe-lds
plot /b /p=1,1 doe-lds

goto done

:error
echo                   #1            #2
echo Usage: DOE-LDS start date  end date
:done
exit
```

Figure A.35 Continued

```
# %W% %G%
# ***********************************************************************
# Copyright (c) 1990, Texas Enginnering Experiment Station
#
# Program: CHANGEX
# Version: %I%
# Last Update: %G%
#
# Description:
# This routine updates the time series graphs with the proper x-axis title
#
# Usage:
# changex <source> [begin] [end] > <output>
#
# <source>    is the source *.grf file.
# <begin>     is the Gregorian or similar date stamp that represents
#             the beginning of the data stream
# <end>       is the Gregorian or similar date stamp that represents
#             the end of the data stream
# <output>    is the output *.grf file name.
#
# History:
#    Design: Doug Bronson
#      Code: Doug Bronson
#
# Distribution Rights
#     DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
#            Texas A & M University., College Station, Texas 77843-3123,
#            (409)- 845-1560
#     SUPPORTED BY: State of Texas Governor's Energy Management Center
#
# COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
#    from being claimed by any other party. Texas A & M University intends
#    that the program be placed in the public domain and grants
#    permission for it to be used and redistributed, provided that:
#          1) the source code is distributed
#          2) this notice is retained in all copies of the source code, and
#          3) the program is not sold for profit without written approval
#             from TEES.
#    The program is distributed "as is". TEES provides no warranty or
#    support service unless special arrangements have been made to do so.
#    Certain manufacturers and trade names are mentioned in this code for
#    the purpose of describing their communications protocol. This does
#    not constitute an endorsement or recommendation of such equipment,
#    but is provided for informantional purposes only.
#
# ***********************************************************************
BEGIN{
  if (ARGC < 4) {
     print "USAGE: changex <source> [start date] [end date]";
     exit;
  }

  start_d = ARGV[ARGC-2];
  end_d = ARGV[ARGC-1];
  x_label = start_d" -- "end_d;

  ARGC = 2;
}
{ if ($1 == "X-AXIS") {
    print $0;
```

Figure A.36 Hard-copy of the CHANGEX Routine


```
    getline;  print $0;

    getline;  print $0;

    getline;  print $0;

    getline;  print $0;

    getline;  print $0;

    getline;
    printf "%s %s \"%s\" \n",$1,$2,x_label;
  }
  else print $0;
}
```

Figure A.36 Continued

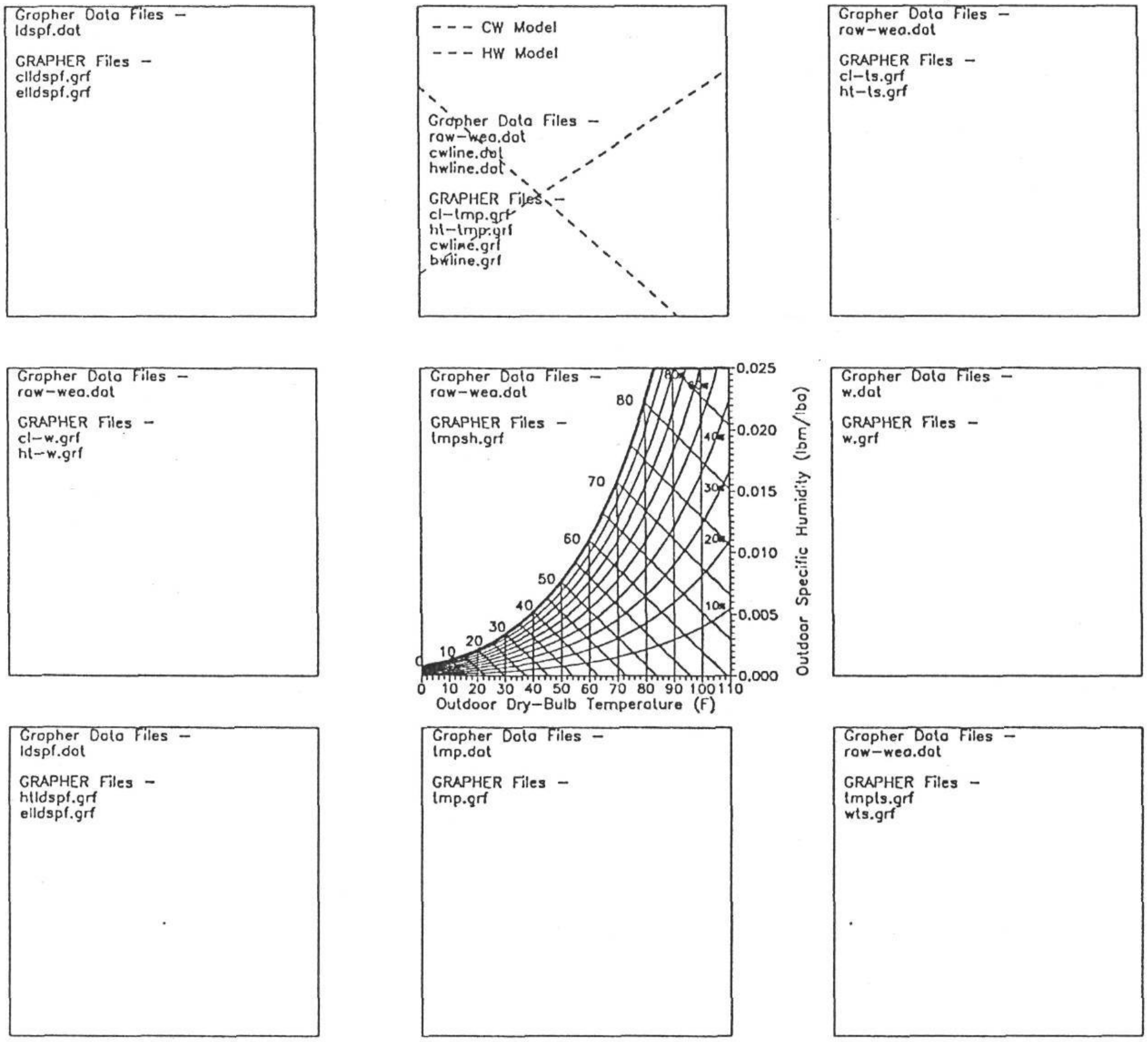

Figure A.37 Graph Data and GRAPHER Files in the Temperature-Specific Humidity Carpet Plot

APPENDIX B

Example Data Files and Program Hard-copies to Pack Site Monitored Weather Data into TRY


```
$ ! /usr/local/bin/gawk -f
$ %W% %G%
$ *********************************************************************
$ Copyright (c) 1990, Texas Enginnering Experiment Station
$
$ Program: PACKWEATHER.COM
$ Version: %I%
$ Last Update: %G%
$
$ Description:
$ This command procedure performs the necessary steps to PACK site
$ monitored weather data into a TRY (Test Reference Year) weather file.
$
$ Usage:
$ @PACKWEATHER Site Monitored Data File/ Base TRY Weather File/ LS2TRY
$ Program Instruction File/ DOE-2 Weather Packer Instruction File
$
$
$ History:
$       Design: Doug Bronson
$       Code: Doug Bronson
$
$ Distribution Rights
$ DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
$               Texas A & M University, College Station, TX 77843-3123
$               (409)- 845-1560
$ SUPPORTED BY: State of Texas Governor's Energy Management Center
$
$ COPYRIGHT NOTICE: This program bears a copyright notice to prevent
$  rights from being claimed by any other party. Texas A & M University
$  intends that the program be placed in the public domain and grants
$  permission for it to be used and redistributed, provided that:
$       1) the source code is distributed
$       2) this notice is retained in all copies of the source code, and
$       3) the program is not sold for profit without written approval
$          from TEES.
$  The program is distributed "as is". TEES provides no warranty or
$  support service unless special arrangements have been made to do so.
$  Certain manufacturers and trade names are mentioned in this code for
$  the purpose of describing their communications protocol. This does
$  not constitute an endorsement or recommendation of such equipment,
$  but is provided for informantional purposes only.
$
$*********************************************************************
$ v = f$verify(0)
$!
$! Command file to take actual weather data and pack it in to a TRY
$! format.
$!
$! @PACKWEATHER.COM P1 P2 P3 P4
$!
$! P1 is the monitored weather file
$! P2 is base TRY ASCII file
$! P3 is the program instruction file
$! P4 is the weather packer instruction file
$!
$ context = ""
$ pid = "''f$pid(context)'"
$ username = "''f$edit(f$getjpi(pid,"username"),"trim")'"
$!
$ scratch_directory = "sys$workdisk:[scratch.''username'_''pid']"
```

Figure B.1 Hard-copy of the PACKWEATHER.COM Command Procedure


```
$!
$ create/dir 'scratch_directory'
$!
$ write sys$output " "
$ write sys$output "The ASCII weather file with actual weather data laid in"
$ write sys$output "is created on ''scratch_directory'"
$ write sys$output " "
$!
$ WEATHER_DATA_FILE := "''P1'"
$ write sys$output "Data weather file -"
$ write sys$output WEATHER_DATA_FILE
$ COPY 'WEATHER_DATA_FILE' -
      'scratch_directory'WEATHER_DATA_FILE.DAT
$ write sys$output " "
$!
$ outfile = "''f$element(0,".",WEATHER_DATA_FILE)'"
$!
$ BASE_WEATHER_FILE := "''P2'"
$ write sys$output "Base weather file -"
$ write sys$output BASE_WEATHER_FILE
$ COPY 'BASE_WEATHER_FILE' -
      'scratch_directory'BASE_WEATHER_FILE.DAT
$ write sys$output " "
$!
$ PROGRAM_INSTRUCTION := "''P3'"
$ write sys$output "Program Instruction file -"
$ write sys$output PROGRAM_INSTRUCTION
$ COPY 'PROGRAM_INSTRUCTION' -
      'scratch_directory'PROGRAM_INSTRUCTION.INS
$ write sys$output " "
$!
$ WEATHER_PACKER_INSTRUCTION := "''P4'"
$ write sys$output "Weather packer instruction file -"
$ write sys$output WEATHER_PACKER_INSTRUCTION
$ COPY 'WEATHER_PACKER_INSTRUCTION' -
      'scratch_directory'WEATHER_PACKER_INSTRUCTION.INS
$ write sys$output " "
$!
$ old_directory = "''f$trnlnm("sys$disk")'''f$directory()'"
$!
$ set def 'scratch_directory'
$!
$ run sys$userdisks:[s0k3404.doe21d]LS2TRY
$!
$ @sys$userdisks:[s0k3404.doe21d]doewth2 WEATHER_TRY.SEQ
WEATHER_PACKER_INSTRUCTION.INS
$!
$ COPY WEATHER_TRY.SEQ 'old_directory''outfile'
$ COPY WEATHER_TRY.OUT 'old_directory''outfile'
$ COPY WEATHER_TRY.WTH 'old_directory''outfile'
$!
$ delete/noconfirm/nolog 'scratch_directory'*.*;*
$!
$ set def sys$workdisk:[scratch]
$ set prot=(o:rewd) sys$workdisk:[scratch]'username'_'pid'.dir
$ delete/noconfirm/nolog sys$workdisk:[scratch]'username'_'pid'.dir;*
$!
$ set def 'old_directory'
$!
```

Figure B.1 Continued


```
90.0   96.03 30.06 14.55 91.   301.  33333 180.
Y Y Y Y Y Y Y Y
```

Figure B.2 Example of a Program Instruction File

							RH (%)	TEMP (F)	SOLAR (w/m^2)	WIND SP (mph)
33333	1	1	90	90001	3653.0417	100	55.48	41.80	2.10	4.86
33333	1	1	90	90001	3653.0833	200	54.02	41.49	2.20	4.61
33333	1	1	90	90001	3653.1250	300	53.22	41.36	2.20	4.88
33333	1	1	90	90001	3653.1667	400	53.17	40.99	2.10	4.34
33333	1	1	90	90001	3653.2083	500	54.42	40.55	2.10	5.40
33333	1	1	90	90001	3653.2500	600	59.33	39.42	2.30	5.38
33333	1	1	90	90001	3653.2917	700	67.39	38.17	2.30	6.01
33333	1	1	90	90001	3653.3333	800	62.94	38.80	8.70	5.06
33333	1	1	90	90001	3653.3750	900	62.24	39.23	51.90	3.22
33333	1	1	90	90001	3653.4167	1000	60.68	40.36	122.20	1.45
33333	1	1	90	90001	3653.4583	1100	50.27	43.18	211.20	0.57
33333	1	1	90	90001	3653.5000	1200	38.05	45.93	228.30	2.10
33333	1	1	90	90001	3653.5417	1300	32.09	47.37	229.00	5.02
33333	1	1	90	90001	3653.5833	1400	31.69	47.87	221.70	3.20
33333	1	1	90	90001	3653.6250	1500	30.79	48.56	238.60	3.15
33333	1	1	90	90001	3653.6667	1600	25.85	48.94	107.00	3.41
33333	1	1	90	90001	3653.7083	1700	26.48	48.81	43.80	3.91
33333	1	1	90	90001	3653.7500	1800	29.99	47.62	6.20	6.17
33333	1	1	90	90001	3653.7917	1900	33.34	46.68	1.60	3.79
33333	1	1	90	90001	3653.8333	2000	35.39	46.06	1.70	4.42
33333	1	1	90	90001	3653.8750	2100	46.61	44.56	1.80	4.03
33333	1	1	90	90001	3653.9167	2200	55.73	43.49	2.10	4.73
33333	1	1	90	90001	3653.9583	2300	61.18	43.18	2.10	4.71
33333	1	1	90	90001	3654.0000	2400	63.39	43.49	2.10	5.00
33333	1	2	90	90002	3654.0417	100	69.15	43.43	2.00	3.73
33333	1	2	90	90002	3654.0833	200	78.66	43.30	2.00	4.27
33333	1	2	90	90002	3654.1250	300	84.37	42.80	2.10	3.22
33333	1	2	90	90002	3654.1667	400	86.47	42.30	2.20	3.98
33333	1	2	90	90002	3654.2083	500	82.27	42.74	2.10	3.94
33333	1	2	90	90002	3654.2500	600	80.97	42.80	2.10	4.41
33333	1	2	90	90002	3654.2917	700	73.20	43.68	2.10	5.06
33333	1	2	90	90002	3654.3333	800	71.00	44.05	3.70	4.46
33333	1	2	90	90002	3654.3750	900	65.19	44.62	25.00	4.99
33333	1	2	90	90002	3654.4167	1000	63.49	46.00	60.90	5.67
33333	1	2	90	90002	3654.4583	1100	60.58	48.00	97.50	5.72
33333	1	2	90	90002	3654.5000	1200	56.18	49.44	139.20	6.59
33333	1	2	90	90002	3654.5417	1300	51.07	49.44	137.00	6.25
33333	1	2	90	90002	3654.5833	1400	44.11	50.31	180.00	5.79
33333	1	2	90	90002	3654.6250	1500	42.61	51.07	81.40	5.89
33333	1	2	90	90002	3654.6667	1600	44.41	50.88	53.60	5.57
33333	1	2	90	90002	3654.7083	1700	47.86	50.69	32.10	5.74
33333	1	2	90	90002	3654.7500	1800	50.62	50.75	4.70	4.55
33333	1	2	90	90002	3654.7917	1900	54.77	50.56	1.40	4.05
33333	1	2	90	90002	3654.8333	2000	60.93	50.19	1.40	5.16
33333	1	2	90	90002	3654.8750	2100	67.04	49.63	1.60	4.57
33333	1	2	90	90002	3654.9167	2200	72.30	50.25	1.40	3.46
33333	1	2	90	90002	3654.9583	2300	76.06	49.81	1.50	5.16
33333	1	2	90	90002	3655.0000	2400	85.57	51.25	1.40	6.20

Figure B.3 Two Day Example of the Monitored Data File

```
14819008007000300008295681010302599999999999999999999999999        1974010100
14819007006-0131000529570089999999999999999999999999999999        1974010101
14819006005-0230000929580079999999999999999999999999999999        1974010102
14819003002-0531001229620060670750007770600077706000777 9999       1974010103
14819001000-0730000429640039999999999999999999999999999999        1974010104
14819000000-0730000629660009999999999999999999999999999999        1974010105
14819-01-01-0830000929680000007770007770000077700000777 9999       1974010106
14819-01-01-0930000829710009999999999999999999999999999999        1974010107
14819-02-02-1030000929730009999999999999999999999999999999        1974010108
14819000000-1128000929780000007770007770000077700000777 9999       1974010109
14819002001-1128001029810009999999999999999999999999999999        1974010110
14819003002-1028000829820009999999999999999999999999999999        1974010111
14819005003-0928000829820000039990007770000077700000777 9999       1974010112
14819006004-1028000829820009999999999999999999999999999999        1974010113
14819007005-1129000829830009999999999999999999999999999999        1974010114
14819007005-1126000829850000007770007770000077700000777 9999       1974010115
14819005003-1228000929870009999999999999999999999999999999        1974010116
14819001000-1128000729880009999999999999999999999999999999        1974010117
14819000000-1129000729890000007770007770000077700000777 9999       1974010118
14819-01-02-1228000629900009999999999999999999999999999999        1974010119
14819-01-02-1124000429920009999999999999999999999999999999        1974010120
14819000000-1023000429930000007770007770000077700000777 9999       1974010121
14819-01-01-0927000329930009999999999999999999999999999999        1974010122
14819000000-1028000329920009999999999999999999999999999999        1974010123
14819001000-0800000029920040482500077704000777040007779999        1974010200
14819001000-0800000029930029999999999999999999999999999999        1974010201
14819001000-0800000029940029999999999999999999999999999999        1974010202
14819002001-0700000029920000089990007770000077700000777 9999       1974010203
14819002001-0722000329940079999999999999999999999999999999        1974010204
14819002001-0700000029930109999999999999999999999999999999        1974010205
14819003002-0626000329940100771001092501099999999999999999        1974010206
14819004003-0600000029940109999999999999999999999999999999        1974010207
14819004003-0700000029930109999999999999999999999999999999        1974010208
14819005004-0500000029948100730300371001099999999999999999        1974010209
14819006005-0136000329938109999999999999999999999999999999        1974010210
14819008007003000000299181099999999999999999999999999999999        1974010211
14819009009006000000298781010302099999999999999999999999999        1974010212
14819011011008000000298481099999999999999999999999999999999        1974010213
14819012012009080003298381099999999999999999999999999999999        1974010214
14819014013010080004298181004301706709010999999999999999999        1974010215
14819015014011060003298181099999999999999999999999999999999        1974010216
14819015015013060003298111099999999999999999999999999999999        1974010217
14819017016012070004298001010301699999999999999999999999999        1974010218
14819017016012070005297801099999999999999999999999999999999        1974010219
14819018018015340004297801099999999999999999999999999999999        1974010220
14819017017015350005297801010301899999999999999999999999999        1974010221
14819017017014000000297801099999999999999999999999999999999        1974010222
14819017017014340003297601099999999999999999999999999999999        1974010223
```

Figure B.4 Two Day Example of the Base TRY Weather File


```
PACK
ZACHERY_WEATHER
TRY     33333  -999      6 30.06 96.0330-BITSOLAR      23 10.  0.025
0.91  0.91  0.91  0.91  0.91  0.91  0.91  0.91  0.91  0.91  0.91  0.91
-999.
END
```

Figure B.5 Example of a DOE-2 Weather Packer Instruction File


```
C ! /usr/local/bin/gawk -f
C %W% %G%
C ****************************************************************************
C Copyright (c) 1990, Texas Enginnering Experiment Station
C
C Program: LS2TRY.FOR
C Version: %I%
C Last Update: %G%
C
C Description:
C     LS2TRY.FOR is a FORTRAN program that does all the unit conversions and
C calculations to derive the necessary meteorological and insolation data
C required by the TRY weather file. The Main Program reads in the LS2TRY
C Program Instruction File and controls the order of operation of the three
C subroutines, WEATHER_PROCESS, VARIABLE_FORMAT, and LAYIN. The
C WEATHER_PROCESS subroutine does all the unit conversions and calculations to
C derive the necessary meteorological and insolation data. The
C VARIABLE_FORMAT subroutine formats the data to consist of the right number
C of columns. The LAYIN subroutine lays-in the formatted variables into the
C base TRY weather file.
C
C
C Usage:
C     RUN LS2TRY
C
C
C History:
C        Design: Doug Bronson
C        Code: Doug Bronson
C
C Distribution Rights
C        DEVELOPED BY: Energy Systems Laboratory, Mechanical Engr. Dept.,
C                Texas A & M University., College Station, Texas 77843-3123,
C                (409)- 845-1560
C        SUPPORTED BY: State of Texas Governor's Energy Management Center
C
C COPYRIGHT NOTICE: This program bears a copyright notice to prevent rights
C        from being claimed by any other party. Texas A & M University intends
C        that the program be placed in the public domain and grants
C        permission for it to be used and redistributed, provided that:
C              1) the source code is distributed
C              2) this notice is retained in all copies of the source code, and
C              3) the program is not sold for profit without written approval
C                 from TEES.
C        The program is distributed "as is". TEES provides no warranty or
C        support service unless special arrangements have been made to do so.
C        Certain manufacturers and trade names are mentioned in this code for
C        the purpose of describing their communications protocol. This does
C        not constitute an endorsement or recommendation of such equipment,
C        but is provided for informantional purposes only.
C
C*****************************************************************************
C
C  THIS FORTRAN FILE DOES UNIT CONVERSIONS AND COMPUTES ADDITIONAL
C  INSOLATION AND METEROLOGICAL WEATHER PARAMETERS FOR A TRY WEATHER
C  FILE. THE ASCII DATA FILE NEEDS TO HAVE A 1-24 TIME STAMP FORMAT.
C
C    CLEAR SKY CORRELATIONS HAVE BEEN ADDED TO MAKE COMPARISON WITH
C    THE RESULTS FROM THE PROGRAM
C
C    INPUT VARIABLES
```

Figure B.6 Hard-copy of the LS2TRY.FOR Main Program


```
      REAL TIME_ZONE_LONG,LOC_LONG,LOC_LAT,STAT_PRESSURE,WIND_DIR
      REAL DLS_BEGIN,DLS_END
      CHARACTER*5 STATION_NUMBER
      CHARACTER*1 DBL,WBTL,DPTL,WDL,WSPL,PHGL,IGL,IDIRL
C
      OPEN(6,FILE='PROGRAM_INSTRUCTION.INS',STATUS='OLD')
C
      READ(6,10)TIME_ZONE_LONG,LOC_LONG,LOC_LAT,STAT_PRESSURE,
     RDLS_BEGIN,DLS_END,STATION_NUMBER,WIND_DIR
      READ(6,11,END=20)DBL,WBTL,DPTL,WDL,WSPL,PHGL,IGL,IDIRL
C
 20   WRITE(*,*)
      WRITE(*,*) '********************************'
      WRITE(*,*) '* PROGRAM INSTRUCTION FILE DATA *'
      WRITE(*,*) '********************************'
      WRITE(*,*)
      IF (INT(TIME_ZONE_LONG/15.0).EQ.4) THEN
            WRITE(*,*)'   TIME ZONE -> ALANTIC'
            ELSE
      IF (INT(TIME_ZONE_LONG/15.0).EQ.5) THEN
            WRITE(*,*)'   TIME ZONE -> EASTERN'
            ELSE
      IF (INT(TIME_ZONE_LONG/15.0).EQ.6) THEN
            WRITE(*,*)'   TIME ZONE -> CENTRAL'
            ELSE
      IF (INT(TIME_ZONE_LONG/15.0).EQ.7) THEN
            WRITE(*,*)'   TIME ZONE -> MOUNTAIN'
            ELSE
      IF (INT(TIME_ZONE_LONG/15.0).EQ.8) THEN
            WRITE(*,*)'   TIME ZONE -> PACIFIC'
            ELSE
            WRITE(*,*)'   TIME ZONE LONGITUDE [DEGREES] -> ',
     WTIME_ZONE_LONG
      ENDIF
      ENDIF
      ENDIF
      ENDIF
      ENDIF
C
      WRITE(*,12) 'WEATHER STATION LONGITUDE [DEGREES]   -   -
     W-    -> ',LOC_LONG
      WRITE(*,12) 'WEATHER STATION LATITUDE  [DEGREES]   -   -
     W-    -> ',LOC_LAT
      WRITE(*,12) 'WEATHER STATION STANDARD PRESSURE [PSIA] -
     W-    -> ',STAT_PRESSURE
      WRITE(*,12) 'FIRST DAY OF DAYLIGHT SAVINGS [DAY OF THE YEA
     WR]  -> ',DLS_BEGIN
      WRITE(*,12) ' LAST DAY OF DAYLIGHT SAVINGS [DAY OF THE YEA
     WR]   -> ',DLS_END
      WRITE(*,13) 'WEATHER STATION NUMBER   -   -   -   -   -
     W-    -> ',STATION_NUMBER
      WRITE(*,12) 'WIND DIRECTION [DEGRESS] -   -   -   -   -
     W-    -> ',WIND_DIR
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)
C
      CLOSE (6,STATUS='DELETE')
C
      WRITE(*,*)'** RUNNING WEATHER VARIABLE PROCESSOR **'
      WRITE(*,*)
```

Figure B.6 Continued


```
      CALL WEATHER_PROCESS(TIME_ZONE_LONG,LOC_LONG,LOC_LAT,
     CSTAT_PRESSURE,DLS_BEGIN,DLS_END,WIND_DIR)
C
      WRITE(*,*)
      WRITE(*,*)
      WRITE(*,*)'** FORMATING THE VARIABLES INTO THE PROPER
     WNUMBER OF COLUMNS **'
      WRITE(*,*)
      CALL VARIABLE_FORMAT
C
      WRITE(*,*)'** LAYING IN THE WEATHER VARIABLES
     W AT THE PROPER DATE **'
      WRITE(*,*)
      WRITE(*,*)'REPLACE DRY BULB TEMPERATURE        ->',DBL
      WRITE(*,*)'REPLACE WET BULB TEMPERATURE        ->',WBTL
      WRITE(*,*)'REPLACE DEW POINT TEMPERATURE       ->',DPTL
      WRITE(*,*)'REPLACE WIND DIRECTION              ->',WDL
      WRITE(*,*)'REPLACE WIND SPEED                  ->',WSPL
      WRITE(*,*)'REPLACE STATION PRESSURE            ->',PHGL
      WRITE(*,*)'REPLACE GLOBAL HORIZONTAL RADIATION ->',IGL
      WRITE(*,*)'REPLACE DIRECT NORMAL RADIATION     ->',IDIRL
      WRITE(*,*)
C
      CALL LAYIN(STATION_NUMBER,DBL,WBTL,DPTL,WDL,WSPL,PHGL,IGL,
     CIDIRL)
C
 10     FORMAT(T1,F6.2,T7,F6.2,T13,F6.2,T19,F6.2,T25,F6.2,T31,F6.2,T37,
     FA5,T43,F6.2)
 11     FORMAT(T1,A1,T3,A1,T5,A1,T7,A1,T9,A1,T11,A1,T13,A1,T15,A1)
 12   FORMAT(T1,A55,T58,F6.2)
 13   FORMAT(T1,A55,T58,A6)
C
      WRITE(*,*)'*****************************************'
      WRITE(*,*)'* WEATHER DATA LAYIN PROCESS - COMPLETED *'
      WRITE(*,*)'*****************************************'
C
      STOP
      END
C
```

Figure B.6 Continued


```
      SUBROUTINE WEATHER_PROCESS(TZLONG,LONG,LAT,P,DLSBEGINS,DLSENDS,
     SWIND_DIR)
C
C        WEATHER_DATA_FILE.DAT SOURCE FILE WILL CONTAIN
C                MM         ~ MONTH
C                DM         ~ DAY OF THE MONTH
C                DECTIME    ~ DECIMAL TIME / PASSED THRU THE PROGRAM
C                HD         ~ HOUR OF THE OBSERVATION
C                JULDY      ~ JULIAN DATE
C                YR         ~ YEAR
C                RH         ~ RELATIVE HUMDITY
C                F          ~ DRY BULB TEMPERATURE [F]
C                IG         ~ GLOBAL HORIZONTAL RADIATION [W/M2]
C                WSP        ~ WIND SPEED [MPH]
C
C        TRY.OUT OUTFILE WILL CONTAIN
C                F          ~ DRY BULB TEMPERATURE [F]
C                WBT        ~ WET BULB TEMPERATURE [F]
C                DPT        ~ DEW POINT TEMPERATURE [F]
C                WIND_DIR   ~ WIND DIRECTION [DEGREES]
C                WSP        ~ WIND SPEED [KNOTS]
C                PHG        ~ PHG ATMOSPHERIC PRESSURE [100TH OF IN OF HG]
C                IG         ~ GLOBAL HORIZONTAL RADIATION [BTU/HR FT2]
C                IDIR       ~ DIRECT NORMAL RADIATION [BTU/HR FT2]
C                YR
C                MM
C                DM
C                CHD       ~ HOUR OF THE DAY - STANDARD TIME
C
C        INPUT VARIABLES
      REAL IG,F,WSP,DECTIME,HD,RH
      INTEGER JULDAY,YR,DM,MM,STATION
C
C        PROGRAM VARIABLES
C
      REAL DEGRAD,PIE,LAT,LONG,TZLONG,DECANG
      REAL LASTCHD,LASTMM,LASTDM,LASTYR,HEADER
C
C        TIME VARIABLES
C
      REAL DECSOLTIME,WSSET,WSRISE,H2,H1,DLS,CHD,MINUTES
      REAL W1,W2,SOLTIME,DLSBEGINS,DLSENDS,DY
      DIMENSION DYCODE(12)
C
C        SOLAR VARIABLES
C
      REAL SIGMA1,SIGMA2,SIGMA3,SIGMAMEAN,IDIFF,IDIR,IO,BEAM
      REAL B,E,KT,D,GSC,DAYFACTOR,HOURFAC1,HOURFAC2
C
C        CLEAR SKY MODEL SOLAR VARIABLES
C
      REAL TAUB,AOSTAR,A1STAR,KSTAR,CLDIR1,CLDIFF1,CLTOTAL1
      REAL RO,R1,RK,CLDIR2,CLDIFF2,CLTOTAL2,BETA,IOCOMPARE
      REAL SIGMAMIDPOINT,A0,A1,A2,A3,B1,DNIO,ALT
      DIMENSION A(12),BS(12),C(12)
C
```

Figure B.7 Hard-copy of the WEATHER_PROCESS Subroutine


```
C       METEOROLOGICAL VARIABLES
C
      REAL ALPHA,P,PW,PHG
      REAL C1,C2,C3,C4,C5,C6,C7,C8,C9,C10,C11,C12,C13
      REAL T,PWS,WS,TW,WW,DIFFW,FT,DPT,WBT,WRATIO
C
C       OPEN FILES
C
            OPEN(5,FILE='WEATHER_DATA_FILE.DAT',STATUS='OLD')
            OPEN(15,FILE='TRY.OUT',STATUS='NEW')
C           OPEN(21,FILE='TEMPS.OUT',STATUS='NEW')
C               OPEN(22,FILE='SIGMA.OUT',STATUS='NEW')
C           OPEN(25,FILE='SOLAR.OUT',STATUS='NEW')
C
C       CONSTANTS
C               GSC        ~ SOLAR RADIATION CONSTANT [BTU/HR FT2]
C               DEGRAD     ~ CONVERSION FACTOR FROM DEGREES TO RADIANS
C               LAT        ~ LATITUDE [RADIANS]
C               TZLONG     ~ LONGITUDE OF THE BEGINNING OF THE TIME ZONE
[DEGREES]
C               LONG       ~ LONGITUDE [DEGREES]
C               P          ~ ATMOSPHERIC PRESSURE [PSIA]
C               PHG        ~ ATMOSPHERIC PRESSURE [IN OF HG]
C               DLSBEGINS ~ FIRST DAY OF DAYLIGHT SAVINGS
C               DLSENDS    ~ LAST DAY OF THE DAYLIGHT SAVINGS
C               ALT        ~ THE ALTITUDE OF THE SITE - [km]
C
            GSC = 429.2
            PIE = 3.141592654
            DEGRAD = PIE/180.0
            LAT = LAT*DEGRAD
            PHG = INT(100*P*2.0418)
            ALT = 0.0841
C
C       CONSTANTS FOR CALCULATING DAY OF THE YEAR
C
      DYCODE(1)  = 0
      DYCODE(2)  = 31
      DYCODE(3)  = 59
      DYCODE(4)  = 90
      DYCODE(5)  = 120
      DYCODE(6)  = 151
      DYCODE(7)  = 181
      DYCODE(8)  = 212
      DYCODE(9)  = 243
      DYCODE(10)  = 273
      DYCODE(11)  = 304
      DYCODE(12)  = 334
C
```

Figure B.7 Continued


```
C         CONSTANTS FOR CALCULATING WETBULB
C
        C1 = -1.021416462E4
        C2 = -4.89350301
        C3 = -5.37657944E-3
        C4 = 1.92023769E-7
        C5 = 3.55758316E-10
        C6 = -9.03446883E-14
        C7 = 4.1635019
        C8 = -1.044039708E4
        C9 = -1.12946496E1
        C10 = -2.7022355E-2
        C11 = 1.2890360E-5
        C12 = -2.478068E-9
        C13 = 6.5459673
C
C         CONSTANTS FOR CLEAR SKY RADIATION FOR DUFFIE AND BECKMANN
CORRELATION
C
C       RO = 1.03
C       R1 = 1.01
C       RK = 1.00
C
C         CONSTANTS FOR CLEAR SKY RADIATION FOR ASHRAE CORRELATION
C
C       A(1) = 390.0
C       A(2) = 385.0
C       A(3) = 376.0
C       A(4) = 360.0
C       A(5) = 350.0
C       A(6) = 345.0
C       A(7) = 344.0
C       A(8) = 351.0
C       A(9) = 365.0
C       A(10) = 378.0
C       A(11) = 387.0
C       A(12) = 391.0
C
C       BS(1) = 0.142
C       BS(2) = 0.144
C       BS(3) = 0.156
C       BS(4) = 0.180
C       BS(5) = 0.196
C       BS(6) = 0.205
C       BS(7) = 0.207
C       BS(8) = 0.201
C       BS(9) = 0.177
C       BS(10) = 0.160
C       BS(11) = 0.149
C       BS(12) = 0.142
C
```

Figure B.7 Continued


```
C       C(1) = 0.058
C       C(2) = 0.060
C       C(3) = 0.071
C       C(4) = 0.097
C       C(5) = 0.121
C       C(6) = 0.134
C       C(7) = 0.136
C       C(8) = 0.122
C       C(9) = 0.092
C       C(10) = 0.073
C       C(11) = 0.063
C       C(12) = 0.053
C
C         PROGRAM
        HEADER = 0.0
C
 10       READ(5,*,END=20)STATION,MM,DM,YR,JULDY,DECTIME,HD,RH,F,IG,WSP
C
C         ***************************************************************
C         ****** TIME CALCULATIONS *******
C
C         HOUR OF THE DAY, HD
              HD = HD/100.0
C
C         DAY OF THE YEAR, DY
              DY = DYCODE(MM) + DM
C
        IF ((MOD(YR,4).EQ.0).AND.(MM.GE.3)) THEN
              DY = DY + 1.0
        ENDIF
C
C         DAYLIGHT SAVINGS, DLS
              DLS = 0.0
              IF (DY.GE.DLSBEGINS.AND.DY.LT.DLSENDS) THEN
               DLS = 1.0
              ENDIF
C
C         *****CORRECT HOUR OF DAY,YEAR,MONTH AND DAY OF YEAR AND MONTH
C                   FOR DAYLIGHT SAVINGS, CHD
C
              CHD = HD - DLS
C
              IF (DLS.EQ.1.0) THEN
                    DECTIME = DECTIME - 0.0416667
              ENDIF
C
```

Figure B.7 Continued


```
      IF (CHD.LE.0) THEN
             CHD = 24 + CHD
            IF (DM.EQ.1) THEN
             IF (MM.EQ.1) THEN
               DM = 31
               MM = 12
               YR = YR - 1
               DY = YR*1000 + 365
             ELSE
               DY = DY - 1
               MM = MM - 1
               IF (MM.EQ.1.OR.MM.EQ.3.OR.MM.EQ.5.OR.MM.EQ.7.OR.
     IMM.EQ.8.OR.MM.EQ.10) THEN
                DM = 31
               ELSE
                   IF (MM.EQ.4.OR.MM.EQ.6.OR.MM.EQ.9.OR.MM.EQ.
     I11) THEN
                   DM = 30
                   ELSE
                      IF (MM.EQ.2.AND.(MOD(YR,4).NE.0)) THEN
                        DM = 28
                      ELSE
                        DM = 29
                      ENDIF
                   ENDIF
               ENDIF
             ENDIF
            ELSE
                  DM = DM - 1
                  DY = DY - 1
            ENDIF
      ENDIF
C
C        ***** CORRECT THE DAY OF THE YEAR, DY, AFTER DAYLIGHT SAVINGS
*****
C
      DY = DYCODE(MM) + DM
C
      IF ((MOD(YR,4).EQ.0).AND.(MM.GE.3)) THEN
            DY = DY + 1.0
      ENDIF
C
C        1         2         3         4         5         6         7
C234567890123456789012345678901234567890123456789012345678901234567890l2
34567
C
      YR = 1900 + YR
C
```

Figure B.7 Continued


```
C        ****** CHECKING FOR MULTIPLE CONSECUTIVE DATE ENTRIES ******
C        ******          AND IMPROPER DATE STAMPS             ******
C
C        ******  REMOVES DATA NOT TAKEN ON THE HOUR           ******
C
      IF (AMOD(CHD,1.0).NE.0.0) THEN
C
            IF (HEADER.EQ.0.0) THEN
            WRITE(*,*)
            WRITE(*,*)
            WRITE(*,75)
            WRITE(*,79)
            WRITE(*,76)
            WRITE(*,77)
            HEADER = 1.0
            ENDIF
C
            WRITE(*,78) MM,DM,YR,INT(CHD)
            GO TO 10
      ENDIF
C
C      ******      REMOVES MULTIPLE CONSECUTIVE DATA ENTRIES      ******
C
      IF (MM.EQ.LASTMM.AND.DM.EQ.LASTDM.AND.YR.EQ.LASTYR.AND.
     ICHD.EQ.LASTCHD) THEN
C
            IF (HEADER.EQ.0.0) THEN
            WRITE(*,*)
            WRITE(*,*)
            WRITE(*,75)
            WRITE(*,79)
            WRITE(*,76)
            WRITE(*,77)
            HEADER = 1.0
            ENDIF
C
            WRITE(*,78) MM,DM,YR,INT(CHD)
            GO TO 10
      ENDIF
C
      LASTMM = MM
      LASTDM = DM
      LASTYR = YR
      LASTCHD = CHD
C
 75     FORMAT(T10,'DUPLICATE OR DATA POINT NOT TAKEN ON THE HOUR')
 79     FORMAT(T8,'OR REMOVING DAYLIGHT SAVINGS DUPLICATE DATA POINT')
 76     FORMAT(T15,'MONTH',T21,'DAY',T27,'YEAR',T33,'HOUR')
 77     FORMAT(T13,'-------------------------')
 78     FORMAT(T17,I2,T22,I2,T27,I4,T33,I4)
C
```

Figure B.7 Continued


```
C        ***** SOLAR TIME - SOLTIME  DECIMAL SOLAR TIME - DECSOLTIME
C
      B = (360.0/364.0)*(DY-81.0)*DEGRAD
      E = 9.87*SIN(2.0*B)-7.53*COS(B)-1.5*SIN(B)
      DECSOLTIME = (CHD*60.0 + 4*(TZLONG - LONG) + E)/60.0
C
      IF (DECSOLTIME.LT.0.0) THEN
            DECSOLTIME = 24.0 + DECSOLTIME
      ENDIF
C
C        ***** SOLAR HOUR ANGLES W1 & W2
C
            W2 = (DECSOLTIME*15.0-180.0)*DEGRAD
C
            IF ((DECSOLTIME - 1.0).LT.0.0) THEN
                  W1 = ((24 + DECSOLTIME - 1.0)*15.0-180.0)*DEGRAD
            ELSE
                  W1 = ((DECSOLTIME - 1.0)*15.0-180.0)*DEGRAD
            ENDIF
C
C        ***** DECLINATION ANGLE
      DECANG = (23.45*SIN((360.0/365.0)*(284+DY)*DEGRAD))*DEGRAD
C
C        ***** SUNSET HOUR ANGLE
            WSSET = ACOS(-TAN(LAT)*TAN(DECANG))
C
C        ***** SUNRISE HOUR ANGLE
            WSRISE = -WSSET
C
C        *********            SOLAR CALCULATIONS            **********
C
C        ***** FILTERING OUT BAD SOLAR DATA - SKIP SOLAR CALCULATIONS
C
      IF (IG.EQ.-99.0) THEN
            IG = 9999.0
            IDIR = 9999.0
            GOTO 12
      ENDIF
C
C        ***** CONVERTING GLOBAL HORIZONTAL FROM W/M2 TO BTU/HR FT2
C
      IG = IG * 0.317052
C
C        ***** DETERMINING IF THE RECORD'S TIME FIELD IS BETWEEN SUNRISE
AND
C                SUNSET AND SETTING VARIABLE VALUES FOR THE CONDITION
C              IF THE TIME STAMP IS NOT BETWEEN SUNRISE AND SUNSET THEN
ALL
C                SOLAR VARIABLES ARE SET EQUAL ZERO OR DEFAULT VALUES
C
      IF (W1.GE.WSSET.OR.W2.LE.WSRISE) THEN
            IG = 0.0
            IDIFF=IG
            IO = 0.0
            IOCOMPARE = 0.0
            DNIO = 0.0
            SIGMAMEAN = 1.570796
            SIGMAMIDPOINT = 1.570796
```

Figure B.7 Continued


```
          BETA = 0.0
          CLTOTAL1 = 0.0
          CLDIR1 = 0.0
          CLDIFF1 = 0.0
          CLTOTAL2 = 0.0
          CLDIR2 = 0.0
          CLDIFF2 = 0.0
          H2 = W2
          H1 = W1
          GO TO 5
      ELSE
          IF (W1.GE.WSRISE.AND.W2.LE.WSSET) THEN
                H2 = W2
                H1 = W1
          ENDIF
C
          IF (WSRISE.GT.W1.AND.WSRISE.LT.W2) THEN
                H2 = W2
                H1 = WSRISE
          ENDIF
C
          IF (WSSET.GT.W1.AND.WSSET.LT.W2) THEN
                H2 = WSSET
                H1 = W1
          ENDIF
      ENDIF
C
C       ***** MEAN OF THE ZENITH ANGLE, SIGMAMEAN,
C                FOR THE INTERVAL H1 TO H2
C
      SIGMA1 = 1/(COS(LAT)*COS(DECANG)*(COS(H2)-COS(H1)))
      SIGMA2 = (SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H2))*
     SACOS(SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H2))-
     SSQRT(1 - (SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H2))**2)
      SIGMA3 = (SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H1))*
     SACOS(SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H1))-
     SSQRT(1 - (SIN(LAT)*SIN(DECANG)+COS(LAT)*COS(DECANG)*COS(H1))**2)
C
      SIGMAMEAN = SIGMA1*(SIGMA2 - SIGMA3)
C
      IF (SIGMAMEAN.GT.1.569051) THEN
          SIGMAMEAN = 1.569051
      ENDIF
C
      BETA = 1.570796 - SIGMAMEAN
C
C       CALCULATING EXTRATERRESTRIAL RADIATION IO
C
      DAYFACTOR = 1.0+0.033*COS((360.0*DY*DEGRAD)/365.0)
      HOURFAC1 = COS(LAT)*COS(DECANG)*(SIN(H2)-SIN(H1))
      HOURFAC2 = (H2-H1)*SIN(LAT)*SIN(DECANG)
C
      IO = (1/(H2-H1))*GSC*DAYFACTOR*(HOURFAC1+HOURFAC2)
C
      DNIO = IO/COS(SIGMAMEAN)
C
```

Figure B.7 Continued


```
C         DIFFUSE RADIATION FACTOR D
C         CORRELATIONS USED ARE BY THE RECOMMENDATION OF FRED BUHL OF LBL
C         AND ARE FROM Erbs, D.G. et al,
C                 "Estimation of Diffused Radiation Fraction
C         for Hourly, Daily and Monthly-Average Global Radiation",
C         Solar Energy, Vol 28, No. 4, pp. 293-302, 1982.
C
C
        KT = IG/IO
C
        IF (KT.LE.0.22) THEN
         D=1.0-0.09*KT
        ENDIF
C
        IF (KT.GT.0.22.AND.KT.LE.0.8)THEN
         D=0.9511-0.1604*KT+4.388*(KT**2)-16.638*(KT**3)+12.336*(KT**4)
        ENDIF
C
        IF (KT.GT.0.8)THEN
              D=0.165
        ENDIF
C
        IDIFF = IG*D
C
C         CLEAR SKY RADIATION CALCULATIONS
C
C         **** DUFFIE AND BECKMANN CORRELATION ****
C
C       AOSTAR = 0.4237 - 0.00821*(6 - ALT)**2
C       A1STAR = 0.5055 + 0.00595*(6.5 - ALT)**2
C       KSTAR = 0.2711 + 0.01858*(2.5 - ALT)**2
C
C       TAUB = RO*AOSTAR + R1*A1STAR*EXP(-RK*KSTAR/COS(SIGMAMEAN))
C
C         CLEAR SKY DIRECT NORMAL
C
C       CLDIR1 = TAUB*DNIO
C
C       CLDIFF1 = IO*(0.2710 - 0.2939*TAUB)
C
C       CLTOTAL1 = CLDIFF1 + CLDIR1*COS(SIGMAMEAN)
C
C         ***** DUFFIE AND BECKMANN CORRELATION *****
C
C         ***** ASHREA CORRELATION ****
C
C         WRITE(*,*) MM,DM,IO, BS(MM), SIN(BETA),
C       WCLDIR2, MM, SIGMA, BETA
C
C       IF (BETA.LT.0.01) THEN
C             CLDIR2 = 0.0
C             GO TO 83
C       ENDIF
C
C       CLDIR2 = A(MM) / EXP(BS(MM)/SIN(BETA))
C
C 83      CLDIFF2 = C(MM)*CLDIR2
C
C       CLTOTAL2 = CLDIFF2 + CLDIR2*COS(SIGMAMEAN)
```

Figure B.7 Continued


```
C
C       ***** ASRHEA CORRELATION *****
C
 5      BEAM  = IG - IDIFF
C
      IDIR = BEAM / COS(SIGMAMEAN)
C
C       ****************** SOLAR CALCULATIONS ****************************
C
C       ***************** METEOROLOGICAL CALCULATIONS ********************
C
C       SETTING DRY BULB TO 999.0 IF F < -20.0 AND DEW POINT TEMPERATURE, DPT,
C       AND WET BULB TEMPERATURE, WBT, TO 999.0
C
 12     IF (F.LT.-20.0.OR.F.GT.120.0) THEN
            F   = 999.0
            DPT = 999.0
            WBT = 999.0
            WRATIO = -99.0
            GO TO 24
      ENDIF
C
C       CALCULATION THE WATER VAPOR SATURATION PRESSURE, PWS
C
      T = F + 459.67
C
      IF (F.LT.32.0) THEN
      PWS = EXP(C1/T + C2 + C3*T + C4*T**2 + C5*T**3 + C6*T**4 +
     PC7*ALOG(T))
      ELSE
      PWS = EXP(C8/T + C9 + C10*T + C11*T**2 + C12*T**3 + C13*ALOG(T))
      ENDIF
C
C       CALCULATION OF THE PARTIAL PRESSURE OF WATER VAPOR, PW
C
      IF (RH.LT.0.0.OR.RH.GT.100.0) THEN
            DPT = 999.0
            WBT = 999.0
            WRATIO = -99.0
            GO TO 24
      ENDIF
C
      PW = PWS*(RH/100.0)
C
C       CALCULATION OF THE HUMIDITY RATIO
C
      WRATIO = 0.62198*PW/(P-PW)
C
      IF (PW.LE.0.0) THEN
            DPT = 999.0
            WBT = 999.0
            GO TO 24
      ENDIF
C
      ALPHA = ALOG(PW)
C
```

Figure B.7 Continued


```
C        CALCULATING DEW POINT TEMPERATURE IN FAHRENHEIT, DPT
C         1         2         3         4         5         6         7
C234567890123456789012345678901234567890123456789012345678901234567890123456
C
      IF (F.GE.32.0) THEN
            DPT = 100.45 + 33.193*ALPHA + 2.319*ALPHA**2
     I+ 0.17074*ALPHA**3 + 1.2063*PW**0.1984
      ENDIF
C
      IF (F.LT.32.0) THEN
            DPT = 90.12 + 26.142*ALPHA + 0.8927*ALPHA**2
      ENDIF
C
      IF (DPT.GT.F) THEN
       DPT = F
      ENDIF
C
      IF (DPT.LT.-99.0) THEN
       DPT = 999.0
      ENDIF
C
C        CALCULATING WET BULB TEMPERATURE IN FAHRENHEIT, WBT
C
C         1         2         3         4         5         6         7
C234567890123456789012345678901234567890123456789012345678901234567890123456
C
      FT = F
      COUNTER = 1.0
C
 28      T = FT + 459.67
C
      IF (F.LT.32.0) THEN
      PWS = EXP(C1/T + C2 + C3*T  + C4*T**2  + C5*T**3  + C6*T**4 +
     PC7*ALOG(T))
      ELSE
      PWS = EXP(C8/T + C9 + C10*T + C11*T**2 + C12*T**3 + C13*ALOG(T))
      ENDIF
C
      WS = 0.62198*(PWS/(P-PWS))
      WW = ((1093. - 0.556*FT)*WS - 0.24*(F - FT))/(1093. + 0.444*F
     W- FT)
      DIFFW = WRATIO - WW
C
      IF (DIFFW.LT.-0.00005) THEN
            FT = FT - 1/COUNTER
      ELSE
            IF (DIFFW.GT.0.00005) THEN
            COUNTER = COUNTER + 1
            FT = FT + 1/(COUNTER - 1) - 1/COUNTER
            ELSE
                  WBT = FT
                  GOTO 24
            ENDIF
      ENDIF
C
```

Figure B.7 Continued


```
      IF (COUNTER.EQ.50) THEN
            WBT = 999.0
      WRITE(*,*) 'BAD WETBULB TEMPERATURE ',MM, DM, CHD
            GO TO 24
      ENDIF
C
      GO TO 28
C
C       CONVERTING WIND SPEED FROM MPH TO KNOTS
C
 24     IF (WSP.LT.0.0) THEN
            WSP = 999.0
            GO TO 25
      ENDIF
C
      WSP = WSP*0.8684
C
C CLEAR SKY MODEL AND TEMPERATURE DATA PRINTOUT
C
C 25       WRITE(21,40)'40',MM,DM,YR,INT(CHD),DECTIME,
C     WINT(F+0.5),WRATIO,INT(WBT+0.5),INT(DPT+0.5)
C       WRITE(22,41)'41',MM,DM,YR,DECSOLTIME,INT(CHD),GRAPHTIME,
C    WINT(IDIR+0.5),INT(IDIFF+0.5),WSRISE,W2,H2,(H2+H1)/2.0,TAN(DECANG),
C    WE/60.0,INT(DNIO+0.5),COS(SIGMAMEAN),COS(SIGMAMIDPOINT)
C     WRITE(25,43)'43',MM,DM,YR,DECSOLTIME,INT(CHD),GRAPHTIME,
C    WINT(IO+0.5),INT(IOCOMPARE+0.5),INT(DNIO+0.5),
C     WINT(IDIR+0.5),INT(CLDIR1+0.5),INT(CLDIR2+0.5),
C     WINT(BEAM+0.5),INT(CLDIR1*COS(SIGMAMEAN)),
C     WINT(CLDIR2*COS(SIGMAMEAN)),
C     WINT(IDIFF+0.5),INT(CLDIFF1+0.5),INT(CLDIFF2+0.5),
C     WINT(IG+0.5),INT(CLTOTAL1+0.5),INT(CLTOTAL2+0.5)
C        1         2         3         4         5         6         7
C23456789012345678901234567890123456789012345678901234567890123456789012
34567
C
C 40       FORMAT(A2,T4,I2,T7,I2,T10,I4,T15,I2,T18,F10.4,T29,
C     FI3,T33,F8.4,T42,I3,T46,I3)
C 41       FORMAT(A2,T4,I2,T7,I2,T10,I4,T15,F10.4,T26,I2,T29,F10.4,T40,
C    FI4,T45,I4,T50,F7.4,T58,F7.4,T66,F7.4,T74,F7.4,T82,F7.4,T90,F7.4,
C    FT98,I4,T103,F5.1,T109,F5.1)
C 43       FORMAT(A2,T4,I2,T7,I2,T10,I4,T15,F10.4,T26,I2,T29,F10.4,T40,
C     FI4,T45,I4,T50,I4,T55,
C     FI4,T60,I4,T65,I4,T70,
C     FI4,T75,I4,T80,I4,T85,
C     FI4,T90,I4,T95,I4,T100,
C     FI4,T105,I4,T110,I4)
C
 25   WRITE(15,16) DECTIME,INT(F+0.5),INT(WBT+0.5),INT(DPT+0.5),
     WINT(WIND_DIR),INT(WSP+0.5),INT(PHG),INT(IG+0.5),INT(IDIR+0.5),
     WYR,MM,DM,INT(CHD-1.0)
C
      GO TO 10
C
 16     FORMAT(F10.4,T12,I3,T16,I3,T20,I3,T24,I3,T28,I3,T32,I4,T37,I6,
     FT44,I6,T51,I4,T56,I2,T59,I2,T62,I2)
C
```

Figure B.7 Continued


```
 20       CLOSE(5,STATUS='DELETE')
      CLOSE(15,STATUS='KEEP')
C     CLOSE(21,STATUS='KEEP')
C        CLOSE(22,STATUS='KEEP')
C     CLOSE(25,STATUS='KEEP')
C
C
      RETURN
      END
C
C        ************  END OF THE WEATHER VARIABLE CALCULATION PROGRAM
*******
C
```

Figure B.7 Continued

Decimal DATE	DRY BULB TEMPERATURE (F)	WET BULB TEMPERATURE (F)	DEW POINT TEMPERATURE (F)	WIND DIR (deg)	WIND SPD (knots)	STATION PRES. (in Hg)	HOR. RADIATION (BTU/hr·ft²)	DIR. RADIATION (BTU/hr·ft²)	YR.	MM	DD	HR
3653.0417	42	36	27	180	4	2970	0	0	1990	1	1	0
3653.0833	41	35	26	180	4	2970	0	0	1990	1	1	1
3653.1250	41	35	25	180	4	2970	0	0	1990	1	1	2
3653.1667	41	35	25	180	4	2970	0	0	1990	1	1	3
3653.2083	41	35	25	180	5	2970	0	0	1990	1	1	4
3653.2500	39	34	26	180	5	2970	0	0	1990	1	1	5
3653.2917	38	34	28	180	5	2970	0	0	1990	1	1	6
3653.3333	39	34	27	180	4	2970	3	0	1990	1	1	7
3653.3750	39	34	27	180	3	2970	16	1	1990	1	1	8
3653.4167	40	35	28	180	1	2970	39	2	1990	1	1	9
3653.4583	43	36	26	180	0	2970	67	7	1990	1	1	10
3653.5000	46	37	22	180	2	2970	72	5	1990	1	1	11
3653.5417	47	37	19	180	4	2970	73	4	1990	1	1	12
3653.5833	48	37	19	180	3	2970	70	5	1990	1	1	13
3653.6250	49	37	19	180	3	2970	76	15	1990	1	1	14
3653.6667	49	37	15	180	3	2970	34	2	1990	1	1	15
3653.7083	49	37	15	180	3	2970	14	1	1990	1	1	16
3653.7500	48	37	17	180	5	2970	2	0	1990	1	1	17
3653.7917	47	36	19	180	3	2970	0	0	1990	1	1	18
3653.8333	46	36	20	180	4	2970	0	0	1990	1	1	19
3653.8750	45	37	25	180	3	2970	0	0	1990	1	1	20
3653.9167	43	37	29	180	4	2970	0	0	1990	1	1	21
3653.9583	43	38	31	180	4	2970	0	0	1990	1	1	22
3654.0000	43	38	32	180	4	2970	0	0	1990	1	1	23
3654.0417	43	39	34	180	3	2970	0	0	1990	1	2	0
3654.0833	43	40	37	180	4	2970	0	0	1990	1	2	1
3654.1250	43	41	38	180	3	2970	0	0	1990	1	2	2
3654.1667	42	40	39	180	3	2970	0	0	1990	1	2	3
3654.2083	43	40	38	180	3	2970	0	0	1990	1	2	4
3654.2500	43	40	37	180	4	2970	0	0	1990	1	2	5
3654.2917	44	40	36	180	4	2970	0	0	1990	1	2	6
3654.3333	44	40	35	180	4	2970	1	0	1990	1	2	7
3654.3750	45	40	34	180	4	2970	8	0	1990	1	2	8
3654.4167	46	41	34	180	5	2970	19	1	1990	1	2	9
3654.4583	48	42	35	180	5	2970	31	1	1990	1	2	10
3654.5000	49	42	34	180	6	2970	44	1	1990	1	2	11
3654.5417	49	42	32	180	5	2970	43	1	1990	1	2	12
3654.5833	50	41	29	180	5	2970	57	2	1990	1	2	13
3654.6250	51	42	29	180	5	2970	26	1	1990	1	2	14
3654.6667	51	42	30	180	5	2970	17	0	1990	1	2	15
3654.7083	51	42	32	180	5	2970	10	1	1990	1	2	16
3654.7500	51	43	33	180	4	2970	1	0	1990	1	2	17
3654.7917	51	43	35	180	4	2970	0	0	1990	1	2	18
3654.8333	50	44	37	180	4	2970	0	0	1990	1	2	19
3654.8750	50	44	39	180	4	2970	0	0	1990	1	2	20
3654.9167	50	46	42	180	3	2970	0	0	1990	1	2	21
3654.9583	50	46	43	180	4	2970	0	0	1990	1	2	22
3655.0000	51	49	47	180	5	2970	0	0	1990	1	2	23

Figure B.8 Two Day Example of TRY.OUT

```
      SUBROUTINE VARIABLE_FORMAT
C
C       THIS PROGRAM READS IN THE DATA OF THE TRY.OUT FILE AND ADDS
C       THE NECESSARY '0' TO PLACE THE DATA IN THE TRY FORMAT
C
      CHARACTER*2 DM,MM,CHD
      CHARACTER*3 F,DPT,WBT,WSP,WIND_DIR
      CHARACTER*4 PHG,YR,IDIR,IG
      REAL DECTIME
      INTEGER DMLENGTH,FLENGTH,IDIRLENGTH,IGLENTH,CHDLENGTH
      INTEGER TDLENGTH,TWLENGTH,MMLENGTH,WSPLENGTH,WIND_DIRLENGTH
C
      OPEN(5,FILE='TRY.OUT',STATUS='OLD')
      OPEN(15,FILE='TRYFORM.OUT',STATUS='NEW')
C
 10     READ(5,30,END=20)DECTIME,F,WBT,DPT,WIND_DIR,WSP,PHG,IG,IDIR,YR,
     RMM,DM,CHD
C
C
      DMLENGTH = LEN(DM)
      DO 11,T=1,DMLENGTH
            IF (DM(T:T).EQ.' ')THEN
                  DM(T:T) = '0'
            ENDIF
 11     CONTINUE
C
      MMLENGTH = LEN(MM)
      DO 12,T=1,MMLENGTH
            IF (MM(T:T).EQ.' ')THEN
                  MM(T:T) = '0'
            ENDIF
 12     CONTINUE
C
      FLENGTH = LEN(F)
      DO 13,T=1,FLENGTH
C
            IF (T.EQ.2.AND.F(T:T).EQ.'-') THEN
                  F(1:1) = '-'
                  F(2:2) = '0'
            ENDIF
C
            IF (F(T:T).EQ.' ') THEN
                  F(T:T) = '0'
            ENDIF
 13     CONTINUE
C
      IDIRLENGTH = LEN(IDIR)
      DO 14,T=1,IDIRLENGTH
            IF (IDIR(T:T).EQ.' ')THEN
                  IDIR(T:T) = '0'
            ENDIF
 14     CONTINUE
C
      WBTLENGTH = LEN(WBT)
      DO 15,T=1,WBTLENGTH
C
            IF (T.EQ.2.AND.WBT(T:T).EQ.'-') THEN
                  WBT(1:1) = '-'
                  WBT(2:2) = '0'
            ENDIF
```

Figure B.9 Hard-copy of the VARIABLE_FORMAT Subroutine

```
C
            IF (WBT(T:T).EQ.' ') THEN
                  WBT(T:T) = '0'
            ENDIF
 15     CONTINUE
C
      IGLENGTH = LEN(IG)
      DO 16,T=1,IGLENGTH
            IF (IG(T:T).EQ.' ')THEN
                  IG(T:T) = '0'
            ENDIF
 16     CONTINUE
C
      DPTLENGTH = LEN(DPT)
      DO 17,T=1,DPTLENGTH
C
            IF (T.EQ.2.AND.DPT(T:T).EQ.'-') THEN
                  DPT(1:1) = '-'
                  DPT(2:2) = '0'
            ENDIF
C
            IF (DPT(T:T).EQ.' ') THEN
                  DPT(T:T) = '0'
            ENDIF
 17     CONTINUE
C
      CHDLENGTH = LEN(CHD)
      DO 18,T=1,CHDLENGTH
            IF (CHD(T:T).EQ.' ')THEN
                  CHD(T:T) = '0'
            ENDIF
 18     CONTINUE
C
C
      WSPLENGTH = LEN(WSP)
      DO 19,T=1,WSPLENGTH
            IF (WSP(T:T).EQ.' ')THEN
                  WSP(T:T) = '0'
            ENDIF
 19     CONTINUE
C
C
      WIND_DIRLENGTH = LEN(WIND_DIR)
      DO 21,T=1,WIND_DIRLENGTH
            IF (WIND_DIR(T:T).EQ.' ')THEN
                  WIND_DIR(T:T) = '0'
            ENDIF
 21     CONTINUE
C
C
      WRITE(15,31)F,WBT,DPT,WIND_DIR,WSP,PHG,IG,IDIR,YR,MM,DM,CHD
C
      GO TO 10
C
```

Figure B.9 Continued


```
 20     CLOSE(5,STATUS='DELETE')
      CLOSE(15,STATUS='KEEP')
C
 30     FORMAT(F10.4,T12,A3,T16,A3,T20,A3,T24,A3,T28,A3,T32,A4,T37,A6,
     FT44,A6,T51,A4,T56,A2,T59,A2,T62,A2)
C
C       1         2         3         4         5         6         7
8
C23456780123456789012345678901234567890123456789012345678901234567890123
4567890
C
 31     FORMAT(A3,T5,A3,T9,A3,T13,A3,T17,A3,T21,A4,T26,A6,
     FT33,A6,T40,A4,T45,A2,T48,A2,T51,A2)
C
      RETURN
      END
C
C       ************  END OF THE CONVERT PROGRAM
***************************
C
```

Figure B.9 Continued

TEMPERATURE			WIND		STATION	RADIATION		YEAR	MM	DD	HR
(F)			DIR	SP	PRES.	HOR.	DIR.				
042	036	027	180	004	2970	0000	0000	1990	01	01	00
041	035	026	180	004	2970	0000	0000	1990	01	01	01
041	035	025	180	004	2970	0000	0000	1990	01	01	02
041	035	025	180	004	2970	0000	0000	1990	01	01	03
041	035	025	180	005	2970	0000	0000	1990	01	01	04
039	034	026	180	005	2970	0000	0000	1990	01	01	05
038	034	028	180	005	2970	0000	0000	1990	01	01	06
039	034	027	180	004	2970	0003	0000	1990	01	01	07
039	034	027	180	003	2970	0016	0001	1990	01	01	08
040	035	028	180	001	2970	0039	0002	1990	01	01	09
043	036	026	180	000	2970	0067	0007	1990	01	01	10
046	037	022	180	002	2970	0072	0005	1990	01	01	11
047	037	019	180	004	2970	0073	0004	1990	01	01	12
048	037	019	180	003	2970	0070	0005	1990	01	01	13
049	037	019	180	003	2970	0076	0015	1990	01	01	14
049	037	015	180	003	2970	0034	0002	1990	01	01	15
049	037	015	180	003	2970	0014	0001	1990	01	01	16
048	037	017	180	005	2970	0002	0000	1990	01	01	17
047	036	019	180	003	2970	0000	0000	1990	01	01	18
046	036	020	180	004	2970	0000	0000	1990	01	01	19
045	037	025	180	003	2970	0000	0000	1990	01	01	20
043	037	029	180	004	2970	0000	0000	1990	01	01	21
043	038	031	180	004	2970	0000	0000	1990	01	01	22
043	038	032	180	004	2970	0000	0000	1990	01	01	23
043	039	034	180	003	2970	0000	0000	1990	01	02	00
043	040	037	180	004	2970	0000	0000	1990	01	02	01
043	041	038	180	003	2970	0000	0000	1990	01	02	02
042	040	039	180	003	2970	0000	0000	1990	01	02	03
043	040	038	180	003	2970	0000	0000	1990	01	02	04
043	040	037	180	004	2970	0000	0000	1990	01	02	05
044	040	036	180	004	2970	0000	0000	1990	01	02	06
044	040	035	180	004	2970	0001	0000	1990	01	02	07
045	040	034	180	004	2970	0008	0000	1990	01	02	08
046	041	034	180	005	2970	0019	0001	1990	01	02	09
048	042	035	180	005	2970	0031	0001	1990	01	02	10
049	042	034	180	006	2970	0044	0001	1990	01	02	11
049	042	032	180	005	2970	0043	0001	1990	01	02	12
050	041	029	180	005	2970	0057	0002	1990	01	02	13
051	042	029	180	005	2970	0026	0001	1990	01	02	14
051	042	030	180	005	2970	0017	0000	1990	01	02	15
051	042	032	180	005	2970	0010	0001	1990	01	02	16
051	043	033	180	004	2970	0001	0000	1990	01	02	17
051	043	035	180	004	2970	0000	0000	1990	01	02	18
050	044	037	180	004	2970	0000	0000	1990	01	02	19
050	044	039	180	004	2970	0000	0000	1990	01	02	20
050	046	042	180	003	2970	0000	0000	1990	01	02	21
050	046	043	180	004	2970	0000	0000	1990	01	02	22
051	049	047	180	005	2970	0000	0000	1990	01	02	23

Figure B.10 Two Day Example of TRYFORM.OUT

```
      SUBROUTINE LAYIN(STATION_NUMBER,DBL,WBTL,DPTL,WDL,WSPL,PHGL,
     SIGL,IDIRL)
C
C       TRYFORM.OUT VARAIBLES
C
      CHARACTER*1 DBL,WBTL,DPTL,WSPL,PHGL,IGL,IDIRL,WDL
      CHARACTER*2 DM,MM,CHD
      CHARACTER*3 F,DPT,WSP,WBT,WIND_DIR
      CHARACTER*4 IDIR,IG,PHG,YR
      CHARACTER*5 STATION_NUMBER
C
C       BASE_WEATHER_FILE.DAT AND WEATHER_TRY.SEQ VARIALBES
C
      CHARACTER*80 DATA, NEWDATA
      CHARACTER*2 MMTRY,DMTRY,CHDTRY
C
C
      OPEN(5,FILE='BASE_WEATHER_FILE.DAT',STATUS='OLD')
      OPEN(6,FILE='TRYFORM.OUT',STATUS='OLD')
      OPEN(15,FILE='WEATHER_TRY.SEQ',STATUS='NEW')
C
 10     READ(6,30,END=15)F,WBT,DPT,WIND_DIR,WSP,PHG,IG,IDIR,YR,MM,DM,CHD
C
 15     READ(5,31,END=20)DATA
C
      NEWDATA = DATA
C
      MMTRY = DATA(74:75)
      DMTRY = DATA(76:77)
      CHDTRY = DATA(78:79)
C
C     WRITE(*,*)MM,' ',MMTRY,' ',DM,' ',DMTRY,' ',CHD,' ',CHDTRY
C
C       1         2         3         4         5         6         7
8
C2345678012345678901234567890123456789012345678901234567890123
4567890
C
      IF(MM.EQ.MMTRY.AND.DM.EQ.DMTRY.AND.CHD.EQ.CHDTRY) THEN
            NEWDATA(1:5) = STATION_NUMBER

      IF (DBL.EQ.'Y') THEN
            NEWDATA(6:8) = F
      ENDIF

      IF (WBTL.EQ.'Y') THEN
            NEWDATA(9:11) = WBT
      ENDIF

      IF (DPTL.EQ.'Y') THEN
            NEWDATA(12:14) = DPT
      ENDIF

      IF (WDL.EQ.'Y') THEN
            NEWDATA(15:17) = WIND_DIR
      ENDIF
```

Figure B.11 Hard-copy of the LAYIN Subroutine


```
      IF (WSPL.EQ.'Y') THEN
            NEWDATA(18:20) = WSP
      ENDIF

      IF (PHGL.EQ.'Y') THEN
            NEWDATA(21:24) = PHG
      ENDIF

            NEWDATA(25:25) = '0'
            NEWDATA(26:27) = '00'
            NEWDATA(28:35) = '99999999'
            NEWDATA(36:45) = '9999999999'
            NEWDATA(46:55) = '9999999999'

      IF (IGL.EQ.'Y') THEN
            NEWDATA(56:59) = IG
      ENDIF

      IF (IDIRL.EQ.'Y') THEN
            NEWDATA(60:63) = IDIR
      ELSE
            NEWDATA(60:63) = '9999'
      ENDIF

            NEWDATA(70:73) = YR
            NEWDATA(74:75) = MM
            NEWDATA(76:77) = DM
            NEWDATA(78:79) = CHD
C
            WRITE(15,31)NEWDATA
      ELSE
            DATA(1:5) = STATION_NUMBER
            DATA(56:59) = '0000'
            DATA(60:63) = '0000'
            WRITE(15,31)DATA
            GO TO 15
      ENDIF
C
      GO TO 10
C
 20   CLOSE(5,STATUS='DELETE')
      CLOSE(6,STATUS='DELETE')
      CLOSE(15,STATUS='KEEP')
C
 30   FORMAT(A3,T5,A3,T9,A3,T13,A3,T17,A3,T21,A4,T26,A6,
     FT33,A6,T40,A4,T45,A2,T48,A2,T51,A2)
C
 31   FORMAT(A80)
C
      RETURN
      END
```

Figure B.11 Continued

```
333330420360273000042970000999999999999999999999999999900000000      1990010100
333330410350263100042970000999999999999999999999999999900000000      1990010101
333330410350253000042970000999999999999999999999999999900000000      1990010102
333330410350253100042970000999999999999999999999999999900000000      1990010103
333330410350253000052970000999999999999999999999999999900000000      1990010104
333330390340263000052970000999999999999999999999999999900000000      1990010105
333330380340283000052970000999999999999999999999999999900000000      1990010106
333330390340273000042970000999999999999999999999999999900030000      1990010107
333330390340273000032970000999999999999999999999999999900160001      1990010108
333330400350282800012970000999999999999999999999999999900390002      1990010109
333330430360262800002970000999999999999999999999999999900670007      1990010110
333330460370222800022970000999999999999999999999999999900720005      1990010111
333330470370192800042970000999999999999999999999999999900730004      1990010112
333330480370192800032970000999999999999999999999999999900700005      1990010113
333330490370192900032970000999999999999999999999999999900760015      1990010114
333330490370152600032970000999999999999999999999999999900340002      1990010115
333330490370152800032970000999999999999999999999999999900140001      1990010116
333330480370172800052970000999999999999999999999999999900020000      1990010117
333330470360192900032970000999999999999999999999999999900000000      1990010118
333330460360202800042970000999999999999999999999999999900000000      1990010119
333330450370252400032970000999999999999999999999999999900000000      1990010120
333330430370292300042970000999999999999999999999999999900000000      1990010121
333330430380312700042970000999999999999999999999999999900000000      1990010122
333330430380322800042970000999999999999999999999999999900000000      1990010123
333330430390340000032970000999999999999999999999999999900000000      1990010200
333330430400370000042970000999999999999999999999999999900000000      1990010201
333330430410380000032970000999999999999999999999999999900000000      1990010202
333330420400390000032970000999999999999999999999999999900000000      1990010203
333330430400382200032970000999999999999999999999999999900000000      1990010204
333330430400370000042970000999999999999999999999999999900000000      1990010205
333330440400362600042970000999999999999999999999999999900000000      1990010206
333330440400350000042970000999999999999999999999999999900010000      1990010207
333330450400340000042970000999999999999999999999999999900080000      1990010208
333330460410340000052970000999999999999999999999999999900190001      1990010209
333330480420353600052970000999999999999999999999999999900310001      1990010210
333330490420340000062970000999999999999999999999999999900440001      1990010211
333330490420320000052970000999999999999999999999999999900430001      1990010212
333330500410290000052970000999999999999999999999999999900570002      1990010213
333330510420290800052970000999999999999999999999999999900260001      1990010214
333330510420300800052970000999999999999999999999999999900170000      1990010215
333330510420320600052970000999999999999999999999999999900100001      1990010216
333330510430330600042970000999999999999999999999999999900010000      1990010217
333330510430350700042970000999999999999999999999999999900000000      1990010218
333330500440370700042970000999999999999999999999999999900000000      1990010219
333330500440393400042970000999999999999999999999999999900000000      1990010220
333330500460423500032970000999999999999999999999999999900000000      1990010221
333330500460430000042970000999999999999999999999999999900000000      1990010222
333330510490473400052970000999999999999999999999999999900000000      1990010223
```

Figure B.12 Two Day Example of WEATHER_TRY.WTH

