CONSTRAINED STOCHASTIC CLIMATE SIMULATION: COMPUTER PROGRAMS AND USER MANUAL

by

DAVID CARLETON CURTIS

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Report Number 276

Prepared with the Support of the National Oceanic and Atmospheric Administration, the National Weather Service, and the National Science Foundation.

February 1983

MIT

DEPARTMENT
OF
CIVIL
ENGINEERING

SCHOOL OF ENGINEERING
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
Cambridge, Massachusetts 02139

R83-4

CONSTRAINED STOCHASTIC CLIMATE SIMULATION: COMPUTER PROGRAMS AND USER MANUAL

by

David Carleton Curtis

RALPH M. PARSONS LABORATORY
HYDROLOGY AND WATER RESOURCE SYSTEMS

Report Number 276

Prepared with the Support of the National Oceanic and Atmospheric Administration, the National Weather Service, and the National Science Foundation

February 1983

ABSTRACT

Computer programs for Constrained Stochastic Climate Simulation (CSCS) and for preliminary parameter estimation are presented. Many of the parameters required by CSCS, can be estimated by commonly available procedures. The remaining parameters can be estimated with the aid of the programs described in this report. Data input summaries and Fortran program listings are also included.

This report is a supplement to:
Curtis, David Carleton; and Peter S. Eagleson, Constrained Stochastic Climate Simulation, Report Number 274, Ralph M. Parsons Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1982.

ACKNOWLEDGEMENTS

This work was performed with the support of the National Oceanic and Atmospheric Administration, the National Weather Service, and the National Science Foundation.

The review of the manuscript by Dr. Peter S. Eagleson is deeply appreciated.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

The constrained Stochastic Climate Simulation (CSCS) Model is a stochastic, multivariate hydrometeorological data generation algorithm. Hourly values of precipitation, cloud cover, shortwave radiation, longwave radiation, temperature, dewpoint, wind speed and wind direction are jointly generated for the two-meter level.

The theoretical development of the CSCS Model is described by Curtis and Eagleson (1982) and will not be repeated here. This volume is intended as a supplement to Curtis and Eagleson (1982) and describes the use of several computer programs used to estimate parameters for the CSCS Model. A description of the data entry procedure for the CSCS Model is also provided.

Parameter estimation for some of the CSCS Model components is straightforward and commonly available procedures can be used. These cases are noted but estimation details are not presented.

Attention is focussed on those parameters requiring special or in some cases unique techniques of analysis. The programs developed to estimate these parameters and their use are described here.

The parameters required by the CSCS Model for each component include:

Precipitation

- mean time between storms
- mean storm duration
- mean storm depth

Cloud Cover

- fairweather mean
- fairweather variance
- fairweather lag-1 serial correlation
- fairweather frequency histogram
- transition decay coefficient

Temperature

- regression coefficients
- deviations variance
- deviations lag-1 serial correlation

Wind Speed

- mean
- variance
- lag-1 serial correlation
- skew

Wind Direction

- mean
- variance
- lag-1 serial correlation
- frequency histogram

Dewpoint

- mean
- variance
- lag-1 serial correlation

  or

- regression coefficients
- deviations variance
- deviations lag-1 serial correlation

The parameters needed for the wind speed, wind direction, and dewpoint components can be estimated by standard statistical procedures and will not be discussed further.

At first glance, the estimation of parameters required for the precipitation component appears to be straightforward. In fact, it is straightforward once the inter-storm periods and the intra-storm periods have been properly delineated. (See Curtis and Eagleson (1982), Chapter 9.2, pp. 112-114.) Therefore, the algorithm developed to define inter- and intra-storm periods is presented.

The definition of inter- and intra-storm periods is also a part of the determination of fairweather cloud cover sequences. Algorithms developed to identify when the fairweather periods of cloud cover occur and to select these sequences from the observed cloud cover data files are presented.

The temperature component was developed as an extension of previously unpublished work by Bryan (see Curtis and Eagleson (1982), Chapter 6, Appendices C and D). This unique approach to temperature generation required new algorithms to be developed for parameter estimation. The resulting computer programs are presented and their use is described.

Chapter 2

PRECIPITATION AND CLOUD COVER

2.1 Introduction

Parameter estimation for the precipitation and cloud cover components is combined due to a common requirement, the delineation of inter- and intra-storm periods. Obviously, the definition of inter-storm periods affects which periods of no precipitation are included in the computation of the mean time between storms and which periods of no precipitation are assumed to be a lull within a storm. Just as obviously, these definitions affect the identification of when fairweather cloud cover sequences occur.

Two computer programs are now presented that facilitate parameter estimation for the precipitation and cloud cover components. The first, PROGRAM RAIN, scans an hourly precipitation data file, defines the inter- and intra-storm periods, and estimates the mean time between storms, the mean storm duration, and the mean storm depth. PROGRAM RAIN also records the beginning and ending times of fairweather periods given an estimate of the transition time (see Curtis and Eagleson (1982) Chapter 9.3) from the end of a storm to the beginning of the fairweather period.

PROGRAM FAIR uses the times of occurrence of fairweather periods as defined by PROGRAM RAIN to extract fairweather cloud cover sequences from the observed cloud cover data file. The resulting collection of fairweather cloud cover sequences is then treated as a new time series whose parameters can be estimated by the usual techniques.

2.2 PROGRAM RAIN

Execution of PROGRAM RAIN from an interactive computer terminal requires the user to answer four questions:

1. Computer: ENTER INPUT DATA FILE NAME XXXXXX.XXX

   User: Respond by entering the name of the data file containing the hourly precipitation data

   Note: PROGRAM RAIN expects the precipitation data file to have the following format

   COL 1-10 station identifier

   COL 13-14 month (2 digits)

   COL 15-16 year (2 digits)

   COL 21-76 data (8F7.2)

2. Computer: ENTER NUMBER AND YEARS TO BE SKIPPED NN YY YY...

   User: Indicate which precipitation data are not to be included in the analysis by entering NN YY YY... where NN is the total number of YY's to follow and YY is the individual years to be skipped. YY corresponds to the 2 digit year code that appears on the data cards described in the previous paragraph.

   Note: Occasionally, a month may have enough missing data that it may be necessary to skip the entire month. If only a few data are missing, the data are read but are assumed to be zero.

3. Computer: ENTER THE NUMBER OF HOURS IN TRANSITION

   User: Enter the estimated number of hours (integer) in the transition from storm to fairweather.

4. Computer: ENTER OUTPUT FILE NAME FOR FAIRWEATHER COORDINATES

   User: Enter the filename using the form XXXXXX.XXX

   Note: The output file will have the following format

   COL 20-25 beginning hour of fairweather period

   COL 26-31 ending hour of fairweather period

Once these four requests have been answered by the user, PROGRAM RAIN scans the observed precipitation data, establishes the times between storms and storm durations, records the total precipitation, and finally

estimates the parameters. PROGRAM RAIN also uses the estimate of transition time to determine the existance of a fairweather sequence. When a valid fairweather period is found, the beginning and ending times of the period are written into the output data file for use by PROGRAM FAIR.

2.3 PROGRAM FAIR

Execution of PROGRAM FAIR from an interactive computer terminal requires one user response.

1. Computer: ENTER COORDINATE FILE, DATA FILE, OUTPUT FILE

   User: Enter the file names for the coordinate file (the output file for PROGRAM RAIN), the data file (the file containing the observed cloud cover data), and the output file for the fairweather cloud cover time series. The file names have the form XXXXXX.XXX

Once this response is made, the times of fairweather occurrence are read from the appropriate file. These times are used to locate the fairweather cloud covers in the observed data file. The fairweather cloud cover data are then extracted and subsequently stored in the output file. The required fairweather cloud cover statistics are then determined by standard techniques.

To ultimately find the proper transition period, PROGRAM RAIN and PROGRAM FAIR are executed several times using different estimates of the transition period as described in Curtis and Eagleson (1982), Chapter 9.3.

2.4 Fortran Listing for PROGRAM RAIN

```
00010   C
00020   C.....................................................................
00030   C
00040         PROGRAM RAIN
00050   C
00060   C  PROGRAMMER:  DAVID C. CURTIS
00070   C               NATIONAL WEATHER SERVICE
00080   C               NORTHEAST RIVER FORECAST CENTER
00090   C               705 BLOOMFIELD AVENUE
00100   C               BLOOMFIELD, CT 06002
00110   C
00120   C               (203) 722-2014
00130   C
00140   C
00150   C  PROGRAM RAIN IS USED TO DETERMINE PARAMETERS FOR THE RAINFALL
00160   C  COMPONENT USED IN THE CSCS MODEL.  THE OBSERVED RAINFALL DATA
00170   C  IS ALSO SCANNED TO DETERMINE THE TIME COORDINATES OF THE
00180   C  INTERSTORM PERIODS.  GIVEN AN ESTIMATE OF THE TRANSITION
00190   C  PERIOD FROM THE END OF A STORM TO THE BEGINNING OF A FAIRWEATHER
00200   C  PERIOD, THE TIME COORDINATES OF THE BEGINNING AND THE END OF
00210   C  A FAIRWEATHER PERIOD ARE OUTPUT TO ANOTHER DATA FILE FOR LATER
00220   C  STATISTICAL ANALYSIS.
00230   C
00240   C
00250   C  COMPUTE RAIN PARAMETERS:
00260   C
00270   C    TR......STORM DURATION
00280   C    TB......TIME BETWEEN STORMS
00290   C    TDEPTH..STORM DEPTH
00300   C
00310   C
00320         DIMENSION DATA(8),DATES(25)
00330         DIMENSION NFWBEG(400), NFWEND(400)
00340   C
00350   C
00360   C  NOTE:  THE DEC-10 COMPUTER USES 5 CHARACTERS/WORD
00370   C
00380         DOUBLE PRECISION DAFILE, STAID
00390         DOUBLE PRECISION FWFILE
00400         INTEGER DATES
00410   C
00420         DATA ON/'ON'/, OFF/'OFF'/
00430   C
00440   C
00450   C  INTERACTIVE INPUT SECTION **********************
00460   C
00470         WRITE (IC,100)
00480     100 FORMAT (///1X,'ENTER INPUT DATA FILE NAME  XXXXXX.XXX')
00490   C
00500         READ (5,110) DAFILE
```

```
00510     110 FORMAT (A)
00520   C
00530   C  NOTE:  PROGRAM RAIN EXPECTS THE FILE SPECIFIED BY DAFILE
00540   C         TO HAVE THE FOLLOWING FORMAT
00550   C
00560   C           COL 1-10    STATION IDENTIFIER
00570   C           COL 13-14   MONTH (2 DIGITS)
00580   C           COL 15-16   YEAR (2 DIGITS)
00590   C           COL 21-76   DATA (8F7.2)
00600   C
00610   C
00620   C
00630   C
00640   C  NOTE:  OCCASIONALLY, A MONTH MAY HAVE ENOUGH MISSING DATA THAT
00650   C         IT MAY BE NECESSARY TO SKIP THE ENTIRE MONTH.  IF ONLY
00660   C         A SMALL AMOUNT OF DATA ARE MISSING, THE DATA ARE READ BUT
00670   C         THE MISSING DATA ARE ASSUMED TO BE ZERO.
00680   C
00690         WRITE (IC,120)
00700     120 FORMAT (//1X,'ENTER NUMBER AND YEARS TO BE SKIPPED  NN YY YY...')
00710   C
00720         READ (5,130) NDA,(DATES(I),I=1,NDA)
00730     130 FORMAT (11I)
00740   C
00750         WRITE (IC,140)
00760     140 FORMAT (//1X,'ENTER NUMBER OF HOURS IN TRANSITION')
00770   C
00780         READ (5,145) NTRANS
00790     145 FORMAT (I)
00800   C
00810   C
00820         WRITE (IC,146)
00830     146 FORMAT(//1X,'ENTER OUTPUT FILE NAME FOR FAIRWEATHER COORDINATES')
00840   C
00850         READ (5,147) FWFILE
00860     147 FORMAT (A)
00870   C
00880   C  NOTE:  THE OUTPUT FILE DESIGNATED BY FWFILE WILL HAVE THE
00890   C         FOLLOWING FORMAT
00900   C
00910   C         COL 20-25  BEGINNING HOUR OF FAIRWEATHER PERIOD
00920   C         COL 26-31  ENDING HOUR OF FAIRWEATHER PERIOD
00930   C
00940   C  END INTERACTIVE INPUT ****************************
00950   C
00960   C
00970         IC = 5
00980         IWRITE = 26
00990         INPUT = 21
01000   C
01010         OPEN (UNIT=INPUT,DEVICE='DSK',ACCESS='SEQIN',FILE=DAFILE)
```

```
01020         OPEN (UNIT=IWRITE,DEVICE='DSK',ACCESS='SEQOUT',FILE=FWFILE)
01030   C
01040   C
01050   C  INITIALIZE VARIABLES
01060   C
01070         MISS = 0
01080         STORM = ' '
01090         TDEPTH = 0.0
01100         DEPTH  = 0.0
01110         DURAT  = 0.0
01120         TR     = 0.0
01130         TBS    = 0.0
01140         TBETWN = 0.0
01150         STORMS = 0.0
01160         NHOUR  = 0
01170         DRY    = 0.0
01180         NFWB   = 0
01190         NFWE   = 0
01200         IDBUG  = 0
01210   C
01220   C
01230     190 CONTINUE
01240   C
01250   C
01260   C  READ THE HOURLY RAINFALL DATA
01270   C
01280         READ (INPUT,200,END=500) STAID, KMO, KYR, (DATA(I),I=1,8)
01290     200 FORMAT (A10,2X,2I2,4X,8F7.2)
01300   C
01310   C  CHECK TO SEE IF THIS DATA SHOULD BE SKIPPED
01320   C
01330         IF (NDA.EQ.0) GO TO 221
01340         DO 220 I = 1,NDA
01350         IF (KYR .EQ. DATES(I)) GO TO 190
01360     220 CONTINUE
01370     221 CONTINUE
01380   C
01390   C  NOW ANALIZE THE DATA IN THE CURRENT RECORD
01400   C
01410         DO 301 I = 1,8
01420   C
01430         NHOUR = NHOUR + 1
01440   C
01450   C  CHECK IF RAIN OCCURRED
01460   C
01470         IF (DATA(I) .GT. 0.0005) GO TO 350
01480   C
01490   C  CHECK IF DATUM IS MISSING
01500   C
01510         IF (DATA(I) .LT. -990.0) MISS = MISS + 1
01520         IF (DATA(I) .LT. -990.0) DATA(I) = 0.0
```

```
01530   C
01540   C  RAIN EQUALS ZERO.  THEREFORE, WE ARE BETWEEN STORMS.
01550   C
01560         IF (STORM .EQ. OFF) GO TO 310
01570   C
01580   C  FIRST HOUR OF NEW INTERSTORM PERIOD. TURN STORM FLAG OFF,
01590   C  UPDATE STORM DURATION COUNTER, AND RESET TIME BETWEEN STORMS.
01600   C
01610         STORM = OFF
01620         DRY   = DRY + 1.0
01630         TR    = TR  + DURAT
01640         TDEPTH= TDEPTH + DEPTH
01650         DURAT = 0.0
01660         DEPTH = 0.0
01670   C
01680   C  RECORD THE HOUR OF THE BEGINNING OF AN INTERSTORM PERIOD
01690   C
01700         NFWB = NFWB + 1
01710         NFWBEG(NFWB) = NHOUR
01720   C
01730            IF (IDBUG .EQ. 0 ) GO TO 310
01740             WRITE (IC,309) NFWBEG(NFWB),NFWB,NHOUR
01750     309     FORMAT (/1X,'BEGINNING',3I6)
01760   C
01770   C
01780     310 CONTINUE
01790   C
01800   C UPDATE TIME BETWEEN STORMS COUNTER
01810   C
01820         TBETWN = TBETWN + 1.0
01830         GO TO 300
01840   C
01850   C
01860   C
01870   C
01880   C  RAINFALL IS GREATER THAN ZERO.  THEREFORE, WE ARE IN A STORM.
01890   C
01900     350 IF (STORM .EQ. ON) GO TO 360
01910   C
01920   C  FIRST HOUR OF NEW STORM. TURN STORM FLAG ON, UPDATE TIME BETWEEN
01930   C  STORMS COUNTER, AND RESET STORM DURATION AND DEPTH COUNTERS.
01940   C
01950         STORM = ON
01960   C
01970   C  RECORD THE BEGINNING OF A STORM PERIOD
01980   C
01990         NFWE = NFWE + 1
02000         NFWEND(NFWE) = NHOUR
02010   C
02020             IF (IDBUG .EQ. 0) GO TO 351
02030             WRITE (IC,352) NFWEND(NFWE),NFWE,NHOUR
```

```
02040     352     FORMAT (1X,3X,'ENDING',3I6)
02050     351 CONTINUE
02060   C
02070         IF( TBETWN .GT. 2.01 ) GO TO 358
02080   C
02090   C  THE CURRENT INTERSTORM PERIOD WAS LESS THAN THREE HOURS LONG.
02100   C  THEREFORE IT IS ASSUMED TO BE A LULL IN THE STORM RATHER THAN
02110   C  A PERIOD BETWEEN TWO DIFFERENT STORMS.  READJUST THE COUNTERS
02120   C  TO REFLECT THIS.
02130   C
02140         DRY = DRY - 1.0
02150         DURAT = DURAT + TBETWN
02160         TBETWN = 0.0
02170         GO TO 360
02180   C
02190     358 CONTINUE
02200   C
02210         TBS = TBS + TBETWN
02220         STORMS = STORMS + 1.0
02230         TBETWN= 0.0
02240   C
02250     360 CONTINUE
02260   C
02270   C  STORM CONTINUES
02280   C
02290         DURAT = DURAT + 1.0
02300         DEPTH = DEPTH + DATA(I)
02310   C
02320     300 CONTINUE
02330   C
02340     301 CONTINUE
02350   C
02360   C  END OF ANALYSIS OF THE CURRENT RECORD. GO READ THE NEXT RECORD.
02370   C
02380         GO TO 190
02390   C
02400   C
02410     500 CONTINUE
02420   C
02430   C  END OF DATA REACHED.  NOW FINISH ANALYSIS OF THE FINAL PERIOD
02440   C  AND ESTIMATE THE PARAMETERS.
02450   C
02460   C
02470         IF (STORM .EQ. ON) GO TO 510
02480   C
02490   C  REACHED END OF THE DATA SET DURING AN INTERSTORM PERIOD.  ASSUME
02500   C  THAT THE INTERSTORM PERIOD ENDS AT THE SAME TIME AS THE DATA SET.
02510   C  NOW UPDATE TIME BETWEEN STORMS STATISTICS.
02520   C
02530         TBS = TBS + TBETWN
02540   C
```

```
02550     510 CONTINUE
02560   C
02570   C  REACHED THE END OF THE DATA SET DURING A STORM PERIOD.  ASSUME
02580   C  THAT THE STORM PERIOD ENDS AT THE SAME TIME AS THE DATA SET.
02590   C  NOW UPDATE THE STORM STATISTICS.
02600   C
02610         TR = TR + DURAT
02620         TDEPTH = TDEPTH + DEPTH
02630   C
02640     520 CONTINUE
02650   C
02660   C  DETERMINE THE ENDING HOUR FOR THE LAST INTERSTORM PERIOD
02670   C
02680         NSTOP = NFWE
02690         IF ( NFWB .LE. NFWE ) GO TO 550
02700         NFWE = NFWE + 1
02710         NSTOP = NFWE
02720         NFWEND(NFWE) = NHOUR
02730     550 CONTINUE
02740   C
02750   C
02760   C  EXAMINE THE INTERSTORM PERIODS
02770   C
02780   C  DETERMINE IF A FAIRWEATHER PERIOD EXISTS. IF YES, WRITE THE
02790   C  BEGINNING AND ENDING HOURS OF FAIRWEATHER SEQUENCE TO THE DATA
02800   C  FILE.  IF NOT, THE INTERSTORM PERIOD IS NOT LONG ENOUGH TO
02810   C  CONSIDER. THEREFORE SKIP AND GO TO NEXT SEQUENCE.
02820   C
02830         DO 560 I = 1,NSTOP
02840   C
02850         NFW = NFWEND(I) - NFWBEG(I)
02860   C
02870             IF ( IDBUG .EQ. 0) GO TO 562
02880             WRITE (IC,563) I,NTRANS,NFW,NFWEND(I),NFWBEG(I)
02890     563      FORMAT ( 1X,5I6)
02900     562 CONTINUE
02910   C
02920         IF ( NFW .LE. 2*NTRANS) GO TO 560
02930   C
02940   C  DELETE THE HOURS IN THE TRANSITION, LEAVING ONLY THE FAIRWEATHER
02950   C  PERIOD.
02960   C
02970           NFWB = NFWBEG(I) + NTRANS -1
02980           NFWE = NFWEND(I) - NTRANS
02990   C
03000             WRITE (IWRITE,555) NFWB, NFWE
03010     555     FORMAT (2I10)
03020   C
03030             IF ( IDBUG .EQ. 0 ) GO TO 559
03040             WRITE (IC,556) NFWB, NFWE
03050     556     FORMAT (1X,18X,2I6)
```

```
03060     559 CONTINUE
03070   C
03080   C
03090   C
03100     560 CONTINUE
03110   C
03120   C  CALCULATE THE PRECIPITATION PARAMETERS
03130   C
03140   C
03150   C  COMPUTE MEAN VALUES
03160   C
03170         TR    = TR/STORMS
03180         TDEPTH= TDEPTH/STORMS
03190         TBS   = TBS/DRY
03200   C
03210   C
03220         WRITE (IC,390)
03230     390 FORMAT (1H1,2(15(5H      )//)//)
03240   C
03250         WRITE (IC,400) DAFILE
03260     400 FORMAT (1X,4(5H***** ),' RAINFALL ANALYSIS ',4(5H*****)/
03270        $        T25,A10//)
03280   C
03290         WRITE (IC,410)
03300     410 FORMAT (1X,T10,'TB',T20,'TR',T30,'H',T40,'STORMS',T50,'DRY')
03310         WRITE (IC,420) TBS,TR,TDEPTH,STORMS,DRY
03320     420 FORMAT (T8,F5.1,T17,F5.2,T27,F5.3,T41,F5.0,T49,F5.0////)
03330   C
03340         WRITE (IC,600) NHOUR,MISS
03350     600 FORMAT (1X,T10,'NHOUR',T20,'MISSING'/T9,I6,T21,I5///)
03360   C
03370         STOP
03380         END
```

2.5 Fortran Listing for PROGRAM FAIR

```
00010   C
00020   C.....................................................................
00030   C
00040         PROGRAM FAIR
00050   C
00060   C  GIVEN THE COORDINATES OF FAIR WEATHER SEQUENCES DETERMINED FROM THE
00070   C  PRECIPITATION DATA USING PROGRAM RAIN, SELECT THE SEQUENCES
00080   C  OF FAIRWEATHER CLOUD COVER FROM THE OBSERVED CLOUD COVER FILE.
00090   C
00100   C     PROGRAMMER:  DAVID C. CURTIS
00110   C                  NATIONAL WEATHER SERVICE
00120   C                  NORTHEAST RIVER FORECAST CENTER
00130   C                  705 BLOOMFIELD AVENUE
00140   C                  BLOOMFIELD, CT 06002
00150   C
00160   C                  203-722-2014
00170   C
00180   C
00190         DIMENSION NFWBEG(400), NFWEND(400), DATA(744), FWDATA(480)
00200   C
00210         DOUBLE PRECISION FWFILE, DAFILE, CLFILE
00220   C
00230   C
00240   C  SELECT I/O UNIT NUMBERS
00250   C
00260         ITERM = 5
00270         IFW   = 21
00280         IDA   = 22
00290         IOUT  = 23
00300   C
00310   C
00320   C
00330         WRITE (ITERM,100)
00340     100 FORMAT (//1X,'ENTER COORDINATE FILE, DATA FILE, OUTPUT FILE')
00350   C
00360         READ (ITERM,105) FWFILE, DAFILE, CLFILE
00370     105 FORMAT (3A)
00380   C
00390         OPEN (UNIT=IFW,DEVICE='DSK',ACCESS='SEQIN',FILE=FWFILE)
00400         OPEN (UNIT=IDA,DEVICE='DSK',ACCESS='SEQIN',FILE=DAFILE)
00410         OPEN (UNIT=IOUT,DEVICE='DSK',ACCESS='SEQOUT',FILE=CLFILE)
00420   C
00430   C  FILE: FWFILE ... CONTAINS COORDINATES OF FAIRWEATHER SEQUENCES
00440   C
00450   C                     COL 20-25  BEGINNING HOUR OF FAIRWEATHER PERIOD
00460   C                     COL 26-31  ENDING HOUR OF FAIRWEATHER PERIOD
00470   C
00480   C  FILE: DAFILE ... CONTAINS OBSERVED CLOUD COVER DATA
00490   C
00500   C                     TWO 'CARDS' PER DAY IN FORMAT (16F5.1/8F5.1)
```

```
00510   C
00520   C  FILE: CLFILE ... OUTPUT FILE FOR FAIRWEATHER CLOUD SEQUENCES.
00530   C
00540   C                    TWO 'CARDS' PER DAY IN FORMAT (16F5.1/8F5.1)
00550   C
00560   C
00570   C  INITIALIZE VARIABLES
00580   C
00590         N = 1
00600         NHOUR = 0
00610         KFW   = 0
00620         NSTOP = 0
00630   C
00640   C  GET FAIRWEATHER COORDINATES FROM FWFILE.
00650   C
00660         DO 110 J = 1,400
00670         READ (IFW,115,END=120,ERR=998) NFWBEG(J), NFWEND(J)
00680     115 FORMAT (2I)
00690     110 CONTINUE
00700   C
00710     120 CONTINUE
00720   C
00730   C
00740   C
00750   C  GET CLOUD COVER DATA FROM DAFILE.
00760   C
00770   C  NOTE:  BECAUSE THE FAIRWEATHER COORDINATES ARE RELATIVE VALUES
00780   C         REFERENCED TO THE FIRST HOUR OF DATA IN THE PRECIPITATION
00790   C         FILE, IT IS ESSENTIAL THAT THE OBSERVED CLOUD COVER FILE
00800   C         BE PRECISELY SYNCHRONIZED WITH THE OBSERVED PRECIPITATION
00810   C         FILE.  BE ESPECIALLY CAREFULL IF ANY MONTHS OF PRECIPITATION
00820   C         DATA ARE SKIPPED DURING THE EXECUTION OF PROGRAM RAIN
00830   C
00840         READ (IDA,130,END=500,ERR=999) DATA
00850     130 FORMAT (16F5.0/8F5.0)
00860   C
00870   C
00880         DO 200 I = 1,744
00890   C
00900         NHOUR = NHOUR + 1
00910   C
00920   C  IS CURRENT HOUR WITHIN A FAIRWEATHER PERIOD?
00930   C
00940         IF (NHOUR .LT. NFWBEG(N)) GO TO 200
00950         IF (NHOUR .LE. NFWEND(N)) GO TO 150
00960         N = N + 1
00970         GO TO 200
00980     150 CONTINUE
00990   C
01000   C
01010   C  NOW WE HAVE A VALID FAIRWEATHER DATUM.
```

```
01020   C
01030   C  IF DATA BUFFER IS FULL, WRITE THE FAIRWEATHER CLOUD DATA TO CLFILE
01040   C
01050         KFW = KFW + 1
01060         IF (KFW .LE. 480) GO TO 170
01070         KFW = 1
01080         WRITE (IOUT,160) (FWDATA(KJ),KJ=1,480)
01090     160 FORMAT (16F5.1/8F5.1)
01100   C
01110   C  STORE FAIRWEATHER CLOUD SEQUENCES IN THE DATA BUFFER
01120   C
01130     170 FWDATA(KFW) = DATA(I)
01140   C
01150     200 CONTINUE
01160         GO TO 120
01170     500 CONTINUE
01180   C
01190         IF (KFW .LE. 480) WRITE (IOUT,160) (FWDATA(KJ),KJ=1,KFW)
01200   C
01210         STOP
01220   C
01230     998 WRITE (ITERM,700) FWFILE
01240     700 FORMAT (//1X,'READ ERROR IN FILE: ',A10)
01250         STOP
01260   C
01270     999 WRITE (ITERM,710) DAFILE
01280     710 FORMAT (//1X,'READ ERROR IN FILE: ',A10)
01290         STOP
01300   C
01310         END
```

Chapter 3

TEMPERATURE

3.1 Introduction

Two programs are presented to help estimate the parameters for the temperature component of the CSCS Model. PROGRAM TEMPER is designed to estimate the regression coefficients of the temperature component described in Curtis and Eagleson (1982), Chapter 6 and Appendix D. PROGRAM TMPGEN uses the regression coefficients determined by PROGRAM TEMPER to create a "predicted" time series of hourly temperatures. This allows the user to compare the "predicted" temperatures with the observed temperatures. The differences between the observed and predicted temperatures are then studied using standard techniques to estimate the required temperature deviations parameters.

3.2 PROGRAM TEMPER

Data can be entered into PROGRAM TEMPER in one of two ways, through an interactive computer terminal or in batch mode. Currently, PROGRAM TEMPER is set up for use through an interactive terminal. If the user wishes to use batch mode, the statement calling the interactive input routine (CALL TRTEMP in the main program) needs to be masked or removed and change the appropriate input unit numbers. PROGRAM TEMPER was originally written for batch mode. Subroutine TRTEMP merely accepts data from an interactive terminal and writes the data to a disk file for later reading in the main program.

Using the interactive mode, from one to five responses to computer prompts are required.

1. Computer: IF AN INPUT DATA FILE EXISTS, YOU CAN DO THE FOLLOWING

ACTION	RESPONSE
1. TYPE "OLDFILE"	EXISTING INPUT FILE IS USED
2. TYPE "NEWFILE"	INPUT NEW DATA AS REQUESTED

User: Typing OLDFILE tells the program that the required data are in an existing file. (For DEC-10 systems filenames are in the OPEN and CLOSE statements) Program control returns to the main program where the data are read and execution begins.

Typing NEWFILE allows new data to be entered and the computer begins a series of prompts requesting the data

2. Computer: INPUT BEGINNING DAY, MONTH, YEAR (4 DIGITS IN YEAR) AND ENDING DAY, MONTH, YEAR

User: Enter the requested data on one line. Each entry is separated by a blank or a comma.

3. Computer: INPUT STATION LAT-LONG IN DMS AND TIME ZONE OF STATION

User: Enter latitude and longitude in degrees minutes and seconds along with the time zone of the station. For example:

42 31 00 72 19 30 EASTERN

4. Computer: INPUT EP, ET, W

User: Enter the requested values on one line separated by blanks or a comma.

Note: EP = +1 for east longitude
-1 for west longitude

ET = the difference between true solar time and mean solar time (usually neglected for heat transfer computations)

W = solar constant (see Curtis and Eagleson (1982), pp. 54-55)

5. Computer: INPUT DATA CHECK VALUES TEMP LOWER BOUND, TEMP UPPER BOUND, WIND SPEED UPPER BOUND

User: Enter the values on one line separated by blanks or a comma. PROGRAM TEMPER uses these values to try and catch any invalid data that may be in the observed data file.

Note: The new data are written to the input data file. Program control returns to main program and execution begins:

If the batch mode option is chosen for data entry, the user must format an input data file or card deck using the format described in the first section of the PROGRAM TEMPER Listing.

Five data files are used by PROGRAM TEMPER for various input/output operations.

1. File DCCTMP.DAT contains the input data described in the previous paragraphs.

2. File DCCTMP.PRM contains the observed values of hourly temperature, cloud cover, wind speed, and wind direction. For each day, 25 observations of each data type are used in the parameter estimation. The data time sequence is:

   11 PM, Midnight, 1 AM,..., 11 AM, Noon, 1 PM,..., 11 PM

   In this format, the 11 PM observation appears twice, in the 25th location for day N and in the 1st location for day N + 1. These are two data records per data type per day. Each record pair for each data type is read by the format (16F5.0/9F5.0). The first two records contain temperature data, the second pair contains cloud cover data, the third pair contains wind speed data, the fourth pair contains wind direction data, and the cycle repeats for day 2, 3,... etc.

3. File DCBCOF.DAT is the output file containing the regression coefficients. The file contains two records. The first record contains coefficients b_0 to b_3 and the second record contains b_4 to b_7. The format for these two records is ('BCOEF 0-3', 4E12.5/'BCOEF 4-7', 4E12.5)

4. File DCCTMP.OUT is used as an alternative for output normally intended for the interactive terminal. It is used only when the I/O unit number IWRITE does not equal 5, the unit number for the interactive terminal.

5. File DCCTMP.BUG is used to output any debug information when any of the internal debug flags are switched on.

3.3 Fortran Listing for PROGRAM TEMPER

```
00010  C
00020  C.....................................................
00030  C
00040        PROGRAM TEMPER
00050  C
00060  C
00070  C  PROGRAM TEMPER IS USED TO ESTIMATE THE REGRESSION COEFFICIENTS FOR
00080  C  THE DETERMINISTIC COMPONENT OF THE TEMPERATURE MODEL.  THE PROGRAM
00090  C  IS CURRENTLY SET UP FOR MONTHLY PARAMETER ESTIMATION.  IF ANY OTHER
00100  C  PERIODS ARE TO BE USED, SUBROUTINE DATE1 WILL HAVE TO BE MODIFIED
00110  C  IN ORDER THAT THE DATE COUNTERS ARE UPDATED PROPERLY.
00120  C
00130  C  PROGRAMMER:
00140  C                DAVID C. CURTIS
00150  C                NORTHEAST RIVER FORECAST CENTER
00160  C                705 BLOOMFIELD AVE
00170  C                BLOOMFIELD, CT  06002
00180  C
00190  C                (203) 722-2520
00200  C
00210  C
00220        INTEGER RANGE
00230        INTEGER SEASON(12)
00240  C
00250        REAL*8 DEBUG(20)
00260        REAL*8 TZONE,TZ(4)
00270        REAL KBAR0B
00280  C
00290        DIMENSION XY(8),XXT(8,8)
00300        DIMENSION A(3),B(3)
00310        DIMENSION TPRIME(25), THAT(25)
00320        DIMENSION KBAR0B(25), GRTEMP(25), CLOUD(25)
00330        DIMENSION WSPEED(25), WDIR(25)
00340        DIMENSION ACOEF(8), BCOEF(8)
00350  C
00360        COMMON /IO/ IREAD,IWRITE,IWBUG,IPARM, IBCOE
00370        COMMON /DBUG/ NBUG, DEBUG
00380        COMMON /SEAS/ NSEAS
00390        COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
00400        COMMON /SUN/   DELTA, DTSL, SR,SS
00410        COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
00420       $              ,JSTART, JSTOP,  JREND,  JYEAR
00430        COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
00440        COMMON /YSTAT/ YSUM, YSUMSQ, YMEAN, RSQUAR
00450  C
00460        DATA TZ/8HEASTERN  ,8HCENTRAL  ,8HMOUNTAIN  ,8HPACIFIC  /
00470  C
00480  C  SET INPUT/OUTPUT UNIT NUMBERS
00490  C
00500        IREAD = 21
```

```
00510         IWRITE=  5
00520         IWBUG = 23
00530         IPARM = 24
00540         IBCOE = 25
00550   C
00560   C
00570   C
00580   C
00590   C     CALL TRANSLATOR PROGRAM FOR INTERACTIVE INPUT
00600   C      (BY-PASS THE NEXT STATEMENT IF BATCH MODE IS USED)
00610         CALL TRTEMP (IREAD)
00620   C
00630   C
00640   C
00650   C
00660   COOOOOOOOOOOOOOOOOOOO.......OPEN........OOOOOOOOOOOOOOOOOOOO
00670   C
00680   C  DATA FILE DEFINITION:
00690   C
00700   C    DCCTMP.DAT ... INPUT DATA FILE
00710   C    DCCTMP.OUT ... OUTPUT DATA FILE
00720   C    DCCTMP.BUG ... DEBUG DATA FILE
00730   C    DCCTMP.PRM ... OBSERVED DATA FILE
00740   C    DCBCOF.DAT ... REGRESSION COEFFICIENT OUTPUT FILE
00750   C
00760   C
00770         OPEN (UNIT=IREAD,DEVICE='DSK',ACCESS='SEQIN',FILE='DCCTMP.DAT')
00780         IF (IWRITE .EQ. 5) GO TO 85
00790         OPEN (UNIT=IWRITE,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCCTMP.OUT')
00800      85 CONTINUE
00810         OPEN (UNIT=IWBUG,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCCTMP.BUG')
00820         OPEN (UNIT=IPARM,DEVICE='DSK',ACCESS='SEQIN',FILE='DCCTMP.PRM')
00830         OPEN (UNIT=IBCOE,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCBCOF.DAT')
00840   C
00850   COOOOOOOOOOOOOOOOOOOO........OPEN........OOOOOOOOOOOOOOOOOOOO
00860   C
00870   C
00880   C  READ INPUT DATA FROM CARDS OR THE INPUT DATA FILE
00890   C
00900   C
00910   C  INPUT DATA SECTION
00920   C
00930   C
00940   C  USE THE FOLLOWING READ STATEMENT WHEN OPERATING IN BATCH MODE
00950   C
00960   CXXXXXREAD (IREAD,100) IREAD, IWRITE, IWBUG
00970   CX100 FORMAT (3I5)
00980   C
00990   C
01000         GO TO 111
01010   C
```

```
C  USE THE FOLLOWING STATEMENTS TO READ SUBROUTINE NAMES FOR
C  DEBUG OUPUT
C
      READ (IREAD,110) NBUG, (DEBUG(I), I=1,NBUG)
  110 FORMAT (I5,5X,7(A8,2X)/(10X,7(A8,2X)))
C
 8000 WRITE (IWRITE,9000) NBUG
 9000 FORMAT (1H0,'NBUG=',2X,I10)
  111 CONTINUE
C
C
      READ (IREAD,120) IDAY,IMO,IYR,LDAY,LMO,LYR
  120 FORMAT (2(3X,I2,3X,I2,1X,I4))
C
C
C
C
C
C
C
      READ (IREAD,140) (A(I),I=1,3),(B(I),I=1,3),TZONE
  140 FORMAT (2(3F5.0,5X),T51,A8)
C
C
C  CONVERT DEGREES TO RADIANS
C
      PHI = DMS(A)
      THETAL = DMS(B)
C
C
C  CHECK THE TIME ZONE TO GET THE PROPER STANDARD MERIDIAN OF THE
C  OBSERVER LOCATION
C
      IF (TZONE.NE.TZ(1)) GO TO 200
      THETAS = 75.0*2.0*3.14159/360.0
      GO TO 300
C
  200 IF (TZONE.NE.TZ(2)) GO TO 210
      THETAS =  90.0*2.0*3.14159/360.
      GO TO 300
C
  210 IF (TZONE.NE.TZ(3)) GO TO 220
      THETAS = 105.0*2.0*3.14159/360.0
      GO TO 300
C
  220 IF (TZONE.NE.TZ(4)) GO TO 230
      THETAS = 120.0*2.0*3.14159/360.0
      GO TO 300
C
  230 WRITE (IWRITE,240)
  240 FORMAT (1H1,'TIME ZONE REQUESTED IS NOT VALID'///)
```

```
01530         WRITE (IWRITE,250) TZONE,(TZ(I),I=1,4)
01540     250 FORMAT (1H0,T10,'REQUESTED TIME ZONE',T36,5H*****,
01550        1A8,5H*****/T10,'AVAILABLE TIME ZONES',T36,5H*****,
01560        2A8,5H*****/T36,5H*****,A8,5H*****/T36,5H*****,
01570        3A8,5H*****/T36,5H*****,A8,5H*****)
01580   C
01590     300 CONTINUE
01600   C
01610   C
01620         READ (IREAD,260) EP,ET,W
01630     260 FORMAT (16F5.0/9F5.0)
01640   C
01650   C
01660   C        READ DATA BOUND VALUES
01670   C           TLB...TEMPERATURE LOWER BOUND
01680   C           TUB...TEMPERATURE UPPER BOUND
01690   C           WUB...WIND SPEED  UPPER BOUND
01700   C
01710         READ (IREAD,260) TLB, TUB, WUB
01720   C
01730   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
01740   C
01750   C        INITIALIZE THE ARRAYS USED IN THE REGRESSION ALGORITHM.
01760   C
01770         DO 100 I = 1,8
01780            XY(I) = 0.0
01790         DO 100 J =1,8
01800            XXT(J,I) = 0.0
01810     100 CONTINUE
01820   C
01830   C
01840            YSUM   = 0.0
01850            YSUMSQ = 0.0
01860            YMEAN  = 0.0
01870   C
01880   C
01890   C
01900   C
01910   C
01920         IRANK = 7
01930   C
01940   C
01950   C         INITIALIZE THE DATE VARIABLES
01960   C
01970         CALL DATE1
01980   C
01990   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02000   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02010   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02020   C
02030   C        EVALUATE OBSERVED DATA TO DEVELOP COEFFICIENTS FOR THE
```

```
02040  C           TEMPERATURE MODEL.
02050  C
02060    325 CONTINUE
02070  C
02080  C  FOR EACH DAY, 25 OBSERVATIONS OF EACH DATA TYPE ARE USED IN THE
02090  C  PARAMETER ESTIMATION.  THE DATA TIME SEQUENCE IS:
02100  C    11PM, MIDNIGHT, 1AM, ... , 11AM, NOON, 1PM, ... , 11PM.
02110  C  IN THIS FORMAT, THE 11PM OBSERVATION APPEARS TWICE, IN THE 25TH
02120  C  LOCATION FOR DAY N AND IN THE 1ST LOCATION FOR DAY N + 1.
02130  C
02140  C
02150         READ (IPARM,260,END=345) TPRIME
02160         READ (IPARM,260,END=345) CLOUD
02170         READ (IPARM,260,END=345) WSPEED
02180         READ (IPARM,260,END=345) WDIR
02190  C
02200  C
02210  C        CHECK INPUT DATA TO MAKE SURE DATA ARE WITHIN
02220  C           REASONABLE BOUNDS.
02230  C
02240         DO 326 LL = 1,25
02250  C
02260         L = LL
02270            IF (TPRIME(L).GE.TLB .AND. TPRIME(L).LE.TUB)GO TO 327
02280               CALL DCHECK ( JULREL, 1, TPRIME, L )
02290    327    IF (CLOUD(L).GE.0.00.AND.CLOUD(L).LE.1.00)  GO TO 328
02300               CALL DCHECK ( JULREL, 2, CLOUD , L )
02310    328    IF (WSPEED(L).GE.0.00.AND.WSPEED(L).LE.WUB) GO TO 329
02320               CALL DCHECK ( JULREL, 3, WSPEED, L )
02330    329    IF (WDIR(L).GE.0.00.AND.WDIR(L).LE.360.00)  GO TO 326
02340               CALL DCHECK ( JULREL, 4, WDIR,   L )
02350  C
02360    326 CONTINUE
02370  C
02380  C
02390  C
02400  C        ESTIMATE RADIATION ATTENUATION DUE TO CLOUD COVER
02410  C        ADJUST WIND DIRECTION DATA AS NECESSARY
02420  C
02430  C
02440            DO 330 I = 1,25
02450               KBAROB(I) = 1. - 0.65*CLOUD(I)**2.
02460               IF(WDIR(I) .GT. 180.) WDIR(I)=ABS(WDIR(I)-360.)
02470    330    CONTINUE
02480  C
02490  C
02500  C        CALL DATA ANALYSIS ROUTINES
02510  C
02520         CALL PRMEST (TPRIME,KBAROB,CLOUD,GRTEMP,WSPEED,WDIR,IRANK,
02530        $              XXT,XY)
02540  C
```

```
C        UPDATE THE DATE COUNTERS
C             ENTRY DATE ... FOR YEARLY PARAMETER ESTIMATION
C             ENTRY DATEM... FOR MONTHLY PARAMETER ESTIMATION
C
C
C-----CALL DATE
      CALL DATEM
C
C
C        CHECK TO SEE IF END OF TEST PERIOD HAS BEEN REACHED
C
C
      IF ( JULDAT .LE. JULEND ) GO TO 325
C
  345 CONTINUE
C
C
C        DETERMINE THE 'A' COEFFICIENTS
C
      CALL COEF ( IRANK, XX1, XY, ACOEF)
C
C
C        DETERMINE THE 'B' COEFFICIENTS
C
      CALL ATOB (ACOEF,BCOEF)
C
C
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
C
C
      WRITE (IWRITE,600)
  600 FORMAT (1H1/,2(1H+,100(1H )/),1H ,'TEMPERATURE MODEL PARAMETER ',
     $         'ESTIMATION PROGRAM OUTPUT'///)
C
      WRITE (IWRITE,610) (ACOEF(I),I=1,8)
  610 FORMAT (1H ,T24, 'A COEFFICIENTS'//
     $        1H ,12X,2HA0,12X,2HA1,12X,2HA2,12X,2HA3/
     $        1H ,4X,4(2X,E12.5)//
     $        1H ,12X,2HA4,12X,2HA5,12X,2HA6,12X,2HA7/
     $        1H ,4X,4(2X,E12.5)///)
C
C
      WRITE (IWRITE,620) (BCOEF(I),I=1,8)
  620 FORMAT (1H ,T24, 'B COEFFICIENTS'//
     $        1H ,12X,2HB0,12X,2HB1,12X,2HB2,12X,2HB3/
     $        1H ,4X,4(2X,E12.5)//
     $        1H ,12X,2HB4,12X,2HB5,12X,2HB6,12X,2HB7/
     $        1H ,4X,4(2X,E12.5)///)
C
```

```
03060         WRITE (IBCOE,625) (BCOEF(I),I=1,8)
03070       625 FORMAT ('BCOEF 0-3 ',4E12.5/'BCOEF 4-7 ',4E12.5)
03080     C
03090     C
03100     C
03110         WRITE (IWRITE,630) YMEAN, RSQUAR
03120       630 FORMAT (1H ,4X,7HYMEAN = ,2X,F5.2, 10X,
03130          $       10HRSQUARED =  ,2X,F5.2)
03140     C
03150     C
03160         STOP
03170         END
03180     C
03190     C
03200     C.....................................................................
03210     C
03220         SUBROUTINE TRTEMP (IREAD)
03230     C
03240     C
03250     C  ROUTINE FOR DATA ENTRY THROUGH AN INTERACTIVE TERMINAL
03260     C
03270     C
03280         INTEGER A1(6)
03290         REAL*8 FILE
03300         REAL*8 DEBUG(20),TZONE, TZ(4)
03310         DIMENSION A(3),B(3),BC(7)
03320         DIMENSION TPRIME(25)
03330     C
03340     C
03350         IC = 5
03360     C
03370     C
03380     C
03390     C
03400         WRITE (IC,905)
03410       905 FORMAT (1H0,T10, 'IF AN INPUT DATA FILE EXISTS, YOU CAN DO THE',
03420          1   ' FOLLOWING:'/
03430          2  T15, 'ACTION',T30,'RESPONSE'/T15,'......',T30,'........'/
03440          3 T10, '1. TYPE ''OLDFILE''',T30,'EXISTING INPUT FILE IS USED'/
03450          4 T10, '2. TYPE ''NEWFILE''',T30, 'INPUT NEW DATA AS REQUESTED'//)
03460     C
03470     C
03480         READ (IC,906) FILE
03490       906 FORMAT (2A)
03500     C
03510     C
03520     C
03530     C
03540         IF ( FILE .EQ. 'OLDFILE' ) RETURN
03550         IF ( FILE .EQ. 'NEWFILE' ) GO TO 14
03560     C
```

```
03570           WRITE (IC,907) FILE
03580     907   FORMAT (1H0,'INVALID INPUT FILE DESIGNATION' ,T35,'*****',A8,
03590        2          '*****'/6X,'VALID DESIGNATIONS ARE*',
03600        3          T35,'*****OLDFILE*****'/
03610        4          T35,'*****NEWFILE*****'//)
03620          STOP
03630     C
03640     C
03650     C
03660       14 CONTINUE
03670     C
03680     C
03690     C         SET UP OUTPUT FILE TO RECIEVE TRANSLATED INPUT DATA
03700     C
03710     C
03720     COOOOOOOOOOOOOOOOOOOO.....OPEN.....OOOOOOOOOOOOOOOOOOO
03730     C
03740           OPEN (UNIT=IREAD,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCCTMP.DAT')
03750     C
03760     COOOOOOOOOOOOOOOOOOOO.....OPEN.....OOOOOOOOOOOOOOOOOOO
03770     C
03780     C
03790     C
03800     C
03810     C
03820     C
03830     C
03840           GO TO 21
03850     C
03860     C  THE FOLLOWING STATEMENTS ALLOW THE USER TO INPUT SUBROUTINE
03870     C  NAMES TO OBTAIN DEBUG INFORMATION
03880     C
03890           WRITE (IC,920)
03900       920 FORMAT (1H0,'INPUT DEBUG INFO -  NBUG, (DEBUG(I),I=1,NBUG)')
03910     C
03920     C
03930           READ (IC,20) NBUG, ( DEBUG(I), I = 1,NBUG )
03940        20 FORMAT ( I, 7A/(7A))
03950     C
03960     C
03970     C
03980           WRITE ( IREAD,25 ) NBUG, ( DEBUG(I),I=1,NBUG)
03990        25 FORMAT ( I5, 5X, 7(A8,2X)/(10X ,7(A8,2X)))
04000     C
04010       21 CONTINUE
04020     C
04030     C
04040     C
04050           WRITE (IC,930)
04060       930 FORMAT (1H0,'INPUT BEGINNING DAY, MONTH, YEAR (4 DIGITS IN ',
04070          $   'YEAR AND ENDING DAY, MONTH, YEAR')
```

```
04080   C
04090   C
04100         READ (IC,30) IDAY,IMO,IYEAR,LDAY,LMO,LYEAR
04110      30 FORMAT (6I)
04120   C
04130   C
04140   C
04150         WRITE (IREAD,35) IDAY, IMO, IYEAR, LDAY, LMO, LYEAR
04160      35 FORMAT (2(3X,I2,3X,I2,1X,I4))
04170   C
04180   C
04190   C
04200   C
04210         WRITE (IC,950)
04220     950 FORMAT (1H0,'INPUT STATION LAT-LONG IN DMS AND TIME ZONE OF '
04230        1              'STATION')
04240   C
04250   C
04260         READ (IC,50)  (A(I),I=1,3), (B(I),I=1,3), TZONE
04270      50 FORMAT ( 6F, A)
04280   C
04290   C
04300   C
04310         WRITE (IREAD,55) (A(I),I=1,3), (B(I),I=1,3), TZONE
04320      55 FORMAT (2(3F5.2,5X),T51,A8)
04330   C
04340   C
04350         WRITE (IC,960)
04360     960 FORMAT (1H0,' INPUT EP, ET, W')
04370   C
04380   C
04390         READ (IC,60) EP, ET, W
04400      60 FORMAT (3F)
04410   C
04420   C
04430   C
04440         WRITE (IREAD,65) EP, ET, W
04450      65 FORMAT ( 3F5.2)
04460   C
04470   C
04480         WRITE (IC,70)
04490      70 FORMAT (1H0,' INPUT DATA CHECK VALUES'/
04500        $ ' TEMP LOWER BOUND, TEMP UPPER BOUND, WIND SPEED UPPER  BOUND')
04510   C
04520         READ (IC,60) TLB, TUB, WUB
04530   C
04540         WRITE (IREAD,970) TLB, TUB, WUB
04550     970 FORMAT  (3F5.0)
04560   C
04570   CCCCCCCCCCCCCCCCCCCCCC.....CLOSE.....CCCCCCCCCCCCCCCCCCCC
04580   C
```

```
      CLOSE (UNIT=IREAD,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCCTMP.DAT')
C
CCCCCCCCCCCCCCCCCCCCC.....CLOSE.....CCCCCCCCCCCCCCCCCCC
C
C
      RETURN
      END
C
C
C.............................................................
C
      SUBROUTINE DATE1
C
C        DATE1 INITIALIZES THE DATE COUNTERS.
C        JULIAN DATES ARE USED.
C
C
C        IYR   ... INITIAL YEAR
C        IMO   ... INITIAL MONTH
C        IDAY  ... INITIAL DAY
C        LYR   ... LAST YEAR
C        LMO   ... LAST MONTH
C        LDAY  ... LAST DAY
C
C        JULDAT... CURRENT JULIAN DATE
C        JBEGIN... JULIAN DATE AT BEGINNING OF RUN
C        JULEND... JULIAN DATE AT END OF RUN
C        JRANGE... LENGTH OF RUN
C        JULREF... JAN 1 OF INITIAL YEAR
C        JULREL... JULIAN DATE RELATIVE TO JAN 1 OF CURRENT YEAR
C        JSTART... RELATIVE JULIAN DATE TO BEGIN MONTHLY PARAMETER
C                  ESTIMATION RANGE
C        JSTOP ... RELATIVE JULIAN DATE TO END MONTHLY PARAMETER
C                  ESTIMATION RANGE
C        JYEAR ... YEAR COUNTER
C
C        NXLPYR... JULIAN DATE OF DEC 31 OF NEXT LEAP YEAR
C
C
C
      COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
      COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
     $              ,JSTART, JSTOP,  JREND,  JYEAR
      COMMON /IO/    IREAD, IWRITE. IWBUG, IPARM, IBCOE
C
      INTEGER IDBUG
C
C        SET DEBUG FLAG
C
         IDBUG = 0
C
```

```
C
C
C       DETERMINE INITIAL JULIAN DATES
C
      CALL JULIAN (IMO,IDAY,IYR,JBEGIN)
      CALL JULIAN (LMO,LDAY,LYR,JULEND)
      CALL JULIAN ( 1,   1,IYR,JULREF)
C
C
      JULREL = JBEGIN - JULREF
      JRANGE = JULEND - JBEGIN + 1
      JULDAT = JBEGIN - 1
C
C       DETERMINE THE NEXT OCCURANCE OF 12/31/(LEAP YEAR)
C             (IE. THE 366TH DAY OF THE YEAR)
C
C
      LASTLP = IYR - MOD(IYR,4)
C
      CALL JULIAN (12,31,LASTLP,NXLPYR)
C
      IF (JULDAT .GE. NXLPYR) NXLPYR = NXLPYR + 1461
C
C
C       NOTE... 1461 = 365 + 365 + 365 + 366
C
C
C     THIS SECTION DEFINES VARIABLES NEEDED FOR MONTHLY
C     PARAMETER ESTIMATION
C
        JYEAR = IYR
        JSTART = JULREL + 1
        CALL JULIAN ( LMO, LDAY, IYR, JDATE )
        JSTOP = JDATE - JULREF + 1
        JREND = JSTOP
          IF (JYEAR - MOD(JYEAR,4)) 65,70,65
   70     IF (IMO.EQ.2 .AND. IDAY.EQ.28) JREND = JSTOP + 1
   65   CONTINUE
C
C
      ENTRY DATE
C
C       THE NEXT SECTION IS USED EACH DAY TO UPDATE
C          THE JULIAN DATE COUNTERS.
C
      JULREL = JULREL + 1
      JULDAT = JULDAT + 1
C
C
C       CHECK FOR END OF YEAR
C
```

```
05610       IF (JULREL .LE. 365) GO TO 100
05620       IF (JULREL .GT. 366) GO TO 200
05630 C
05640 C        CHECK FOR LEAP YEAR
05650 C
05660       IF (JULDAT.NE.NXLPYR) GO TO 200
05670 C
05680 C        YES, THERE ARE 366 DAYS THIS YEAR.
05690 C        UPDATE NXLPYR TO NEXT LEAP YEAR.
05700 C
05710       NXLPYR = NXLPYR + 1461
05720 C
05730 C
05740       IF( IDBUG .NE. 0 ) GO TO 900
05750 C
05760 C
05770    50 CONTINUE
05780 C
05790 C
05800       RETURN
05810 C
05820 C
05830 C        RESET RELATIVE JULIAN DATE
05840 C
05850   200 JULREL = 1
05860 C
05870 C
05880   100 CONTINUE
05890          IF( IDBUG .NE. 0 ) GO TO 900
05900   110 RETURN
05910 C
05920 C
05930       ENTRY DATEM
05940 C
05950 C     THIS SECTION IS USED EACH DAY TO UPDATE THE JULIAN DATE COUNTERS
05960 C     IF MONTHLY PARAMETER ESTIMATION IS USED.
05970 C
05980         JULREL = JULREL + 1
05990         JULDAT = JULDAT + 1
06000 C
06010         IF (JULREL .LE. JREND) GO TO 400
06020 C
06030 C     UPDATE THE JULIAN COUNTERS
06040 C
06050         JULREL = JSTART
06060         JYEAR  = JYEAR + 1
06070         CALL JULIAN (IMO,IDAY,JYEAR,JULDAT)
06080         JREND = JSTOP
06090         IF(JYEAR - MOD(JYEAR,4)) 400,410,400
06100   410   IF (LMO.EQ.2 .AND. LDAY.EQ.28) JREND = JREND + 1
06110   400   CONTINUE
```

```
C
      IF ( IDBUG .NE. 0 ) GO TO 900
C
C
      RETURN
C
C
C
  900 CONTINUE
C
C       DEBUG INFORMATION FOR JULIAN DATE CALCULATIONS
C
      WRITE (IWBUG,920) JULDAT,JULREL,JBEGIN,JULEND,JRANGE,NXLPYR
     $                  ,JSTART,JSTOP,JREND,JYEAR
  920 FORMAT (1H ,'JULDAT=',I10,3X,'JULREL=',I10,3X,'JBEGIN=',I10,3X,
     $       /2X,'JULEND=',I10,3X,'JRANGE=',I10,3X,'NXLPYR=',I10,
     $       /2X,'JSTART=',I10,3X,'JSTOP =',I10,3X,'JREND =',I10,
     $        3X,'JYEAR=',I10)
C
      RETURN
C
C
      END
C
C.....
C
      SUBROUTINE DATT(IDATE,IMO,IDAY,IYR)
C
C     CONVERT JULIAN DATE TO CALENDER DATE
C
      INTEGER CAL(12,2)
      DATA CAL/0,31,59,90,120,151,181,212,243,273,304,334,
     1  0,31,60,91,121,152,182,213,244,274,305,335 /
      I1=(IDATE-1)/1461
      I2=IDATE-(I1*1461)
C
C       1<=I2<=1461
C
      IF(I2.LE.365) GO TO 10
      IF(I2.LE.730) GO TO 20
      IF(I2.LE.1095) GO TO 30
      I3=3
      I4=I2-1095
      GO TO 40
   10 I3=0
      I4=I2
      I4=I2-365
      GO TO 40
   20 I3=1
      GO TO 40
   30 I3=2
```

```
      I4=I2-730
   40 IYR=1900+I3+(4*I1)
      INDX=1
      IF(I3.EQ.3)INDX=2
      DO 100 I=2,12
      IF(I4.LE.CAL(I,INDX)) GO TO 200
  100  CONTINUE
       IMO=12
      IDAY=I4-CAL(12,INDX)
       RETURN
  200 IMO=I-1
      IDAY=I4-CAL(I-1,INDX)
      RETURN
      END
C
C
C.....................................................................
C
      SUBROUTINE JULIAN(MO,DA,YR,ANS)
      INTEGER ANS,CAL(12),DA,YR
      DATA CAL /31,28,31,30,31,30,31,31,30,31,30,31/
C
C        COMPUTE JULIAN DATE FROM JAN. 1, 1973
C
      ANS=0
      I=YR-1900
      ANS=ANS+365*I
      CAL(2)=28
      IF(MOD(YR,4).EQ.0) CAL(2)=29
      J=MO-1
      IF(J.EQ.0) GO TO 20
      DO 10 I=1,J
      ANS=ANS+CAL(I)
   10 CONTINUE
   20 CONTINUE
      ANS=ANS+DA
      RETURN
      END
C
C.....................................................................
C
      FUNCTION DMS(A)
C
C     FUNCTION DMS CONVERTS ANGLES EXPRESSED IN
C     DEGREES, MINUTES AND SECONDS TO RADIANS
C
      DIMENSION A(3)
      REAL MINUTE
C
C
      DEGREE = A(1)
```

```
07140         MINUTE = A(2)
07150         SECOND = A(3)
07160   C
07170         DMS = DEGREE*3.14159/180. + MINUTE*3.14159/180./60.
07180        1      + SECOND*3.14159/180./60./60.
07190   C
07200         RETURN
07210         END
07220   C
07230   C.....................................................................
07240   C
07250         FUNCTION TAU(ST)
07260   C
07270         COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
07280         COMMON /IO/ IREAD, IWRITE, IWBUG, IPARM, IBCOE
07290         COMMON /DBUG/ NBUG,DEBUG
07300         REAL*8    ITAU,DEBUG(20)
07310         DATA ITAU /'TAU'/
07320   C
07330   C
07340   C     THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
07350   C              75TH MERIDIAN FOR EASTERN STANDARD TIME
07360   C              90TH MERIDIAN FOR CENTRAL STANDARD TIME
07370   C              105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
07380   C              120TH MERIDIAN FOR PACIFIC STANDARD TIME
07390   C     THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
07400   C     TAU    = LOCAL HOUR ANGLE
07410   C     ST     = STANDARD TIME IN THE TIME ZONE OF THE
07420   C              OBSERVER IN HOURS COUNTED FROM
07430   C              MIDNIGHT (EG. 0.00 TO 24.00)
07440   C     EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
07450   C     ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME
07460   C              AND MEAN SOLAR TIME (USUALLY NEGLECTED
07470   C              FOR HEAT TRANSFER COMPUTATIONS)
07480   C
07490   C
07500   C     FUNCTION SUBROUTINE TAU CONVERTS THE OBSERVERS
07510   C     STANDARD TIME TO LOCAL HOUR ANGLE IN RADIANS
07520   C
07530   C     OBTAIN TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
07540   C     OBSERVERS MERIDIAN (HOURS)
07550   C
07560         DTSL = EP*(THETAS - THETAL)* 12.0/3.14159
07570   C
07580   C     COMPUTE OBSERVERS HOUR ANGLE (RADIANS). E = +1 FOR
07590   C     MORNING AND E = -1 FOR AFTERNOON (I.E. SOLARNOON)
07600   C
07610         IF (ST.GT.12. + DTSL -ET) E = -1.0000
07620         IF (ST.LE.12. + DTSL -ET) E = +1.0000
07630   C
07640         TAU = (ST + E*12. - DTSL + ET) * 3.14159/12.0
```

```
C
      IF (TAU.GT.6.283185) TAU = TAU - 6.283185
      IF (TAU.LT.0.0) TAU = TAU + 6.283185
C
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 100
      DO 200 I = 1,NBUG
C
      IF (DEBUG(I).NE.ITAU) GO TO 200
      WRITE (IWBUG,250)    ST,PHI,THETAS,THETAL,EP,ET,W,DTSL,TAU
  250 FORMAT (////1H ,'FUNCTION TAU',  2X,
     1 'ST =',F6.3,2X,'PHI =',F6.3,2X,'THETAS =',F6.3,2X,'THETAL ='
     2 F6.3,2X,'EP =',F6.3,2X,'ET =',F6.3,2X,'W =',F6.3,2X,
     3 'DTSL =',F6.3,2X,'TAU =',F6.3)
  200 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C.....
C
      SUBROUTINE PRMEST ( TPRIME, KBAROB, CLOUD, GRTEMP, WSPEED,
     $                    WDIR,  IRANK, XXT, XY)
C
C
C        PRMEST IS THE CONTROLLING SUBROUTINE FOR THE PARAMETER ESTIMATION
C           FOR THE TEMPERATURE PROJECTION ROUTINE
C           BASED ON TODAY'S OBSERVED TEMPS., CLD COVER, ETC, PRMEST
C           DETERMINES THE PARAMETERS NECESSARY TO PROJECT
C           TEMPERATURES FOR TOMORROW.
C
C
      DIMENSION TPRIME(1), XXT(8,8), XY(8)
      DIMENSION KBAROB(1), CLOUD(1), GRTEMP(1)
      DIMENSION WSPEED(1), WDIR(1)
      REAL KBAR, KBAROB
      INTEGER RANGE, IDBUG
C
C
      COMMON /SUN/    DELTA, DTSL, SR, SS
      COMMON /ORBIT/  PHI, THETAS, THETAL, EP, ET,W
      COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
     $               ,JSTART, JSTOP,  JREND,  JYEAR
      COMMON /DATES/  IYR, IMO, IDAY, LYR, LMO, LDAY
      COMMON /IO/     IREAD, IWRITE, IWBUG, IPARM, IBCOE
C
C
```

```
C        SET DEBUG FLAG
C
            IDBUG = 0
C
C
C
C
C
C        COMPUTE THE ANGLE ADJUSTMENT BETWEEN THE
C        STANDARD MERIDIAN OF THE OBSERVER'S TIME ZONE AND THE
C        OBSERVER'S LOCAL MERIDIAN.
C
      DTSL = EP*(THETAS-THETAL)*3.81972
C
C
C        COMPUTE THE DECLINATION OF THE SUN
C
      CALL DECL (JULREL,DELTA, SR,SS)
C
C
C        DETERMINE THE LIMITS OF INTEGRATION FOR THE TEMPERATURE
C           GENERATION ALGORITHM
C
      CALL LIMITS (DTSL, SR,SS,T0,RHO,T12,SIGMA,T23)
         IF ( IDBUG .EQ.0 ) GO TO 951
            WRITE(IWBUG,950)DELTA,DTSL,T0,RHO,SR,T12,SIGMA,SS,T23,PHI
  950          FORMAT (T2,'SUBROUTINE PRMEST'/T2,'DELTA=',1X,F10.4,
     $             T20,'DTSL =',1X,F10.4,T40,'T0   =',1X,F10.4,
     $             T60,'RHO  =',1X,F10.4,T80,'SR   =',1X,F10.4/
     $             T2 ,'T12  =',1X,F10.4,T20,'SIGMA=',1X,F10.4,
     $             T40,'SS   =',1X,F10.4,T60,'T23  =',1X,F10.4,
     $             T80,'PHI  =',1X,F10.4)
  951    CONTINUE
C
C
C
C
C        INITIALIZE THE STANDARD TIME COUNTER.
C
      ST = 0.0
C
C
C        BEGIN LOOP TO ANALYZE TODAY'S TEMPERATURES.  COMPUTE
C           THE SET OF COEFFICIENTS FOR THE TEMPERATURE PROJECTIONS.
C
      DO 200 I = 2,25
C
         TMPLAG = TPRIME(I-1)
         KBAR   = KBAROB(I)
         TGD    = GRTEMP(I)
         CLD    = CLOUD(I)
```

```
08670           WSP    = WSPEED(I)
08680           WDR    = WDIR(I)
08690   C
08700   C          COMPUTE THE PREDICTORS X1, X2, X3, X4, X5, X6, X7
08710   C
08720           CALL X1X2X3 (ST,TMPLAG,PHI,DELTA,T0,SR,T12,SS,T23,RHO,
08730        $      SIGMA, CLD, KBAR, TGD, WSP, WDR, X1,X2,X3,X4,X5,X6,X7,
08740        $      RANGE, I)
08750   C
08760   C
08770           IF ( IDBUG .EQ. 0 ) GO TO 961
08780              WRITE(IWBUG,960)ST,TMPLAG,PHI,X1,X2,X3,X4,X5,X6,X7,RANGE
08790     960   FORMAT (12,'ST=',F5.0,'  TMPLAG=',F5.1,'  PHI=',F10.4,
08800        $             '  X=',7(1X,F10.5),'  RANGE=',I2)
08810     961   CONTINUE
08820   C
08830   C          DETERMINE THE HOURLY TEMPERATURE CHANGE
08840   C
08850           Y = TPRIME(I) - TPRIME(I-1)
08860   C
08870   C
08880   C          UPDATE REGRESSION MATRIX AND VECTOR
08890   C
08900           CALL REGRES (Y,X1,X2,X3,X4,X5,X6,X7,XY,IRANK,XXT)
08910            GO TO 830
08920           IF(ST.LT.22.5) GO TO 830
08930         WRITE(5,850)JULDAT,((XXT(IZ,IY),IY=1,IRANK),IZ=1,IRANK)
08940     850     FORMAT(1X,7HJULDAT=,1X,I7/7(1X,7(E11.4)/))
08950     840     CONTINUE
08960     830   CONTINUE
08970   C
08980              IF ( IDBUG .EQ. 0 ) GO TO 962
08990              WRITE (IWBUG,965) ST,T,Y,TPRIME(I),TPRIME(I-1),X1,X2,X3
09000        $                    ,X4,X5,X6,X7
09010     965      FORMAT('ST=',F4.0,2X,'T=',F7.3,2X,'Y=',F7.3,' TPI=',F7.3,2X,
09020        $        'TPI1=',F7.3,2X,'X1=',F7.3,2X,'X2=',E12.5,2X,'X3=',E12.5,
09030        $        2X,'X4=',E7.2,2X,'X5=',E7.2,2X,'X6=',E7.2,2X,'X7=',F7.2)
09040     962      CONTINUE
09050   C
09060              IF ( IDBUG .EQ. 0 ) GO TO 916
09070   C
09080   C          DEBUG STATEMENTS....CHECK  MATRIX OPERATION RESULTS
09090   C
09100         DO 910 II = 1,IRANK
09110     910 WRITE (IWBUG,900) (XXT(II,J),J=1,IRANK)
09120     900 FORMAT (1X/' XXT=',1X,6(F12.5,3X))
09130         WRITE (IWBUG,903) (XY(J),J=1,IRANK)
09140     903 FORMAT (1X/'  XY=',1X,6(F12.5,3X))
09150   C
09160     916 CONTINUE
09170   C
```

```
09180   C
09190   C          UPDATE THE STANDARD TIME COUNTER
09200   C
09210           ST = ST + 1.0
09220   C
09230   C
09240     200 CONTINUE
09250   C
09260         RETURN
09270         END
09280   C
09290   C
09300   C.....................................................
09310   C
09320         SUBROUTINE COEF (IRANK, XXT, XY, ACOEF )
09330   C
09340   C
09350   C        SUBROUTINE COEF DETERMINES THE REGRESSION COEFFICIENTS
09360   C           REQUIRED FOR THE TEMPERATURE MODEL.
09370   C
09380   C
09390         DIMENSION LWORK(8), MWORK(8), A(64)
09400         DIMENSION XXT(8,8), XY(8), ACOEF(8) ,XXTINV(8,8)
09410   C
09420         INTEGER RANGE, IDBUG
09430   C
09440         COMMON /IO/  IREAD, IWRITE, IWBUG, IPARM, IBCOE
09450         COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
09460        $               ,JSTART, JSTOP,  JREND,  JYEAR
09470         COMMON /YSTAT/ YSUM, YSUMSQ, YMEAN, RSQUAR
09480   C
09490   C
09500   C        SET DEBUG FLAG
09510   C
09520         IDBUG = 0
09530   C
09540   C
09550   C
09560   C
09570   C             WHEN IRANK IS LESS THAN THE PROGRAM DIMENSIONS
09580   C             FOR XXT, PROBLEMS WILL OCCUR WHEN XXT IS INVERTED.
09590   C             THESE PROBLEMS ARISE DUE TO THE WAY DATA IS STORED
09600   C             IN MATRIX FORM.  THE SOLUTION IS TO CONVERT XXT(J,I)
09610   C             TO VECTOR FORM, A(L).
09620   C
09630   C
09640   C
09650   C
09660   C
09670   C
09680   C
```

```
09690                 DO 205 I = 1,IRANK
09700                 DO 205 J = 1,IRANK
09710  C
09720                  JJ = (I-1)*IRANK + J
09730                  A(JJ) = XXT(J,I)
09740  C
09750           IF(IDBUG.EQ.1) WRITE (IWBUG,204) J,I,JJ,A(JJ),XXT(J,I)
09760    204    FORMAT (1H0,3I5, 2(2X,E12.5))
09770    205    CONTINUE
09780  C
09790  C
09800  C
09810  C        INVERT THE REGRESSION MATRIX, XXT.
09820  C
09830        CALL MINV (A,IRANK,DXXT,LWORK, MWORK)
09840  C
09850  C
09860  C
09870  C
09880  C
09890                 DO 305 I = 1,IRANK
09900                 DO 305 J = 1,IRANK
09910    305          XXTINV(J,I) = A((I-1)*IRANK + J)
09920  C
09930  C
09940        IF ( IDBUG .EQ. 0 ) GO TO 921
09950        DO 920 I = 1,IRANK
09960    920 WRITE (IWBUG,904) (XXTINV(I,J),J=1,IRANK)
09970    904 FORMAT (1X/' XXTINV=',2X,6(F12.5,3X))
09980  C
09990    921 CONTINUE
10000  C
10010  C
10020  C        DETERMINE THE REGRESSION COEFFICIENTS
10030  C
10040        CALL MATMLT (XY,XXTINV,ACOEF,IRANK)
10050  C
10060  C
10070  C SINCE PREDICTER X5 IS NOT BEING USED, THE ELEMENTS OF ACOEF
10080  C HAVE BEEN REARRANGED SLIGHTLY (SEE SUBROUTINE X1X2X3).
10090  C NOW REORDER ACOEF.
10100  C
10110           ACOEF(8) = ACOEF(7)
10120           ACOEF(7) = ACOEF(6)
10130           ACOEF(6) = 0.0
10140  C
10150        AXY = 0
10160  C
10170  C
10180  C
10190  C
```

```
C        COMPUTE MULTIPLE COEFFICIENT OF DETERMINATION
C
      DO 400 I = 1,8
  400 AXY = ACOEF(I)*XY(I) + AXY
C
      XXTYY = XXT(1,1)*YMEAN*YMEAN
C
      RSQUAR = (AXY-XXTYY)/(YSUMSQ-XXTYY)
C
C
      IF ( IDBUG .EQ. 1 ) GO TO 919
      WRITE (IWBUG,918) JULREL, DXXT, (ACOEF(IJ),IJ=1,IRANK)
  918 FORMAT (1H ,'JULREL=',I5,5X,'DET. OF XXT=',E12.5/
     $    '  ACOEF=',8(2X,E11.4))
C
      WRITE (IWBUG,930) AXY,XXTYY,XXT(1,1),YSUMSQ
  930 FORMAT (1H ,3HAXY ,1X,E12.5,5X,5HXXTYY,1X,E12.5,5X,
     $       8HXXT(1,1) ,1X,E12.5,5X,6HYSUMSQ,1X,E12.5)
  919 CONTINUE
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE DECL (RJD,DELTA,SR,SS)
      INTEGER RJD
      COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
      COMMON /IO/ IREAD, IWRITE, IWBUG, IPARM, IBCOE
      COMMON /DBUG/ NBUG,DEBUG
      REAL*8    IDECL,DEBUG(20)
      DATA IDECL/'DECL'/
C
C     DELTA  = DECLINATION OF THE SUN (RADIANS)
C     PHI    = OBSERVERS LATITUDE (RADIANS)
C     THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
C              75TH MERIDIAN FOR EASTERN STANDARD TIME
C              90TH MERIDIAN FOR CENTRAL STANDARD TIME
C              105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
C              120TH MERIDIAN FOR PACIFIC STANDARD TIME
C     THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
C     RJD    = RELATIVE JULIAN DATE (I.E. WITH RESPECT TO JAN 1)
C     ST     = STANDARD TIME IN THE TIME ZONE OF THE OBSERVER
C              IN HOURS COUNTED FROM MIDNIGHT (E.G.0.00 TO 24.00)
C     EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
C     ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME AND
C              MEAN SOLAR TIME (USUALLY NEGLECTED FOR
C              HEAT TRANSFER COMPUTATIONS)
C
C COMPUTE TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
C OBSERVERS MERIDIAN (HOURS)
```

```
C
      DTSL   = EP*(THETAS - THETAL)* 3.81972
C
C COMPUTE DECLINATION OF THE SUN (RADIANS)
C
      DELTA  = 0.4093*COS(0.0172*(172. - FLOAT(RJD)) )
C
C COMPUTE HOUR ANGLE AT SUNSET (RADIANS)
C
C
      TSS    = ACOS(-TAN(DELTA)*TAN(PHI))
C     COMPUTE STANDARD TIME OF SUNST (HOURS)
C
      SS     = TSS*3.81972 + 12. +DTSL -ET
C
C COMPUTE HOUR ANGLE OF SUNRISE (RADIANS)
C
      TSR    = 6.283185 - TSS
C
C     COMPUTE  STANDARD TIME OF SUNRISE (HOUR)
C
      SR     = TSR*3.81972 -12. + DTSL -ET
C
C
C     CONVERT SUNRISE IN STANDARD TIME TO LOCAL TIME
C
      SR = SR - DTSL
C
C     CONVERT SUNSET IN STANDARD TIME TO LOCAL TIME
C
      SS = SS - DTSL
C
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 300
      DO 100 I = 1,NBUG
C
      IF (DEBUG(I).NE.IDECL) GO TO 100
      WRITE (IWBUG,200)  RJD,DTSL,DELTA,TSS,SS,TSR,SR
  200 FORMAT (////,1H ,'SUBROUTINE DECL ','***',' RJD =',
     1 I5,'  DTSL =',F6.3,'  DELTA =',F6.3,'  TSS =',F6.3,
     2 '  SS =',F6.3,2X,'TSR =',F6.3,2X,'SR = ',F6.3)
  100 CONTINUE
C
  300 CONTINUE
C
      RETURN
      END
C
C.....
```

```
11220   C
11230         SUBROUTINE LIMITS (DTSL,R,S,T0,RHO,T12,SIGMA,T23)
11240   C     FIND LIMITS FOR TEMPERATURE INTEGRATION
11250   C
11260   C
11270         T0  = - DTSL
11280         T23 = 23.00 - DTSL
11290   C
11300         IF (DTSL.LT.0.0) GO TO 50
11310   C
11320   C     FIND LIMITS OF INTEGRATION WHEN OBSERVER IS
11330   C     WEST OF THE STANDARD MERIDIAN
11340   C
11350   C     FOR SUNRISE
11360   C
11370         RHO = AINT(R+1.0) - DTSL
11380         IF (RHO .LT. R) RHO = RHO + 1
11390   C
11400   C     FOR SUNSET
11410   C
11420         SIGMA = AINT(S+1) - DTSL
11430         IF (SIGMA .LT. S) SIGMA = SIGMA + 1
11440   C
11450   C     FOR LOCAL NOON
11460   C
11470         T12 = 13.0 - DTSL
11480   C
11490         GO TO 75
11500   C
11510      50 CONTINUE
11520   C
11530   C      FIND LIMITS OF INTEGRATION WHEN OBSERVER
11540   C      IS EAST OF THE STANDARD MERIDIAN
11550   C
11560         RHO = AINT(R) - DTSL
11570         IF (RHO .LT. R) RHO = RHO + 1
11580   C
11590         SIGMA = AINT(S) - DTSL
11600         IF (SIGMA .LT. S) SIGMA = SIGMA + 1
11610   C
11620         T12 = 12.0 - DTSL
11630   C
11640      75 CONTINUE
11650   C
11660         RETURN
11670         END
11680   C
11690   C.....................................................................
11700   C
11710         SUBROUTINE X1X2X3 ( ST, TMPLAG, PHI, DELTA, T0, R, T12, S, T23,
11720        $      RHO, SIGMA, CLD, KBAR, TGD, WSP, WDR, X1,X2,X3,X4,X5,X6,X7,
```

```
11730       $      RANGE,T)
11740   C
11750   C
11760   C        COMPUTE THE PREDICTERS X1,...,X7.
11770   C
11780   C
11790   C        ST     ... LOCAL STANDARD TIME
11800   C        T      ... LOCAL TIME
11810   C        TMPLAG ... TEMPERATURE AT PREVIOUS TIME PERIOD
11820   C        PHI    ... STATION LATITUDE (RADIANS)
11830   C        DELTA  ... DECLINATION (RADIANS)
11840   C        R      ... LOCAL SUNRISE
11850   C        S      ... LOCAL SUNSET
11860   C        TO     ... VALUE OF T AT LOCAL STANDARD MIDNIGHT
11870   C        RHO    ... FIRST OBSERVATION HOUR AFTER SUNRISE
11880   C        T12    ... FIRST OBSERVATION HOUR AFTER LOCAL NOON
11890   C        SIGMA  ... FIRST OBSERVATION HOUR AFTER SUNSET
11900   C        T23    ... VALUE OF T AT LOCAL STANDARD 2300 HR (IE  11 PM)
11910   C
11920           INTEGER RANGE, IDBUG
11930           INTEGER SWICH1, SWICH2
11940           REAL KBAR
11950           COMMON /SWITCH/ SWICH1, SWICH2
11960           COMMON /IO/ IREAD, IWRITE, IWBUG, IPARM, IBCOE
11970   C
11980   C
11990           SWICH1 = 1
12000           SWICH2 = 0
12010   C
12020   C
12030   C
12040           A  = 0.0005
12050           PI = 3.14159
12060           IDBUG = 0
12070   C
12080   C
12090   C          CONVERT STANDARD TIME TO LOCAL TIME
12100   C
12110           T = TAU(ST)*(12.0/PI) - 12.0
12120   C
12130            IF ( T .GT. 24.0 ) T = T - 24.0
12140            IF ( T .LT.  0.0 ) T = T + 24.0
12150   C
12160   C
12170   C
12180   C          X1 IS JUST THE LAG-1 TEMPERATURE
12190   C
12200           X1 = TMPLAG
12210   C
12220   C
12230   C          DETERMINE THE APPROPRIATE RANGE FOR X2 AND X3
```

```
12240   C
12250   C
12260   C     ... BEFORE SUNRISE       ...
12270   C
12280         IF ( TO .LE. T  .AND.  T .LT. R ) GO TO 100
12290   C
12300   C
12310   C     ... SUNRISE              ...
12320   C
12330         IF ( RHO- A .LE. T  .AND.
12340        1     RHO+ A .GE. T        ) GO TO 200
12350   C
12360   C
12370   C     ... MORNING HOURS        ...
12380   C
12390         IF ( RHO+A .LE. T  .AND.  T .LE. 12 ) GO TO 300
12400   C
12410   C
12420   C     ... NOON                 ...
12430   C
12440         IF ( T12- A .LE. T  .AND.
12450        1     T12+ A .GE. T        ) GO TO 400
12460   C
12470   C
12480   C     ... AFTERNOON HOURS      ...
12490   C
12500         IF ( T12+ A .LE. T  .AND.  T .LT. S ) GO TO 500
12510   C
12520   C
12530   C     ... SUNSET               ...
12540   C
12550         IF ( SIGMA- A .LE. T  .AND.
12560        1     SIGMA+ A .GE. T      ) GO TO 600
12570   C
12580   C
12590   C     ... EVENING HOURS        ...
12600   C
12610         IF ( SIGMA+A .LE. T  .AND.  T .LE. T23 ) GO TO 700
12620   C
12630   C
12640   C---------------------------------------------------------------------
12650   C
12660     100 X2 = 0.0
12670         X3 = 0.0
12680         RANGE = 1
12690         GO TO 900
12700   C
12710   C---------------------------------------------------------------------
12720   C
12730     200 A = PI*R/12.0
12740         B = PI*RHO/12.0
```

```
12750  C
12760        X2 = (RHO-R)*SIN(PHI)*SIN(DELTA)
12770        X2 =X2  - (12.0/PI)*COS(DELTA)*COS(PHI)*(SIN(B) - SIN(A) )
12780  C
12790        X3 = COS(DELTA)*COS(PHI)*(COS(A)-COS(B))
12800        RANGE = 2
12810        GO TO 900
12820  C-------------------------------------------------------------------
12830  C
12840    300 A = PI*T/12.0
12850        B = PI*(T-1.0)/12.0
12860  C
12870        X2 = SIN(DELTA)*SIN(PHI)
12880        X2 = X2 - (12.0/PI)*COS(DELTA)*COS(PHI)*(SIN(A)-SIN(B))
12890  C
12900        X3 = COS(DELTA)*COS(PHI)*(COS(B)-COS(A))
12910        RANGE = 3
12920        GO TO 900
12930  C
12940  C-------------------------------------------------------------------
12950  C
12960    400 A = PI*T/12.0
12970        B = PI*(T-1.0)/12.0
12980        C = PI*(T12-1.0)/12.0
12990  C
13000        X2 = SIN(DELTA)*SIN(PHI)
13010        X2 = X2 - (12.0/PI)*COS(DELTA)*COS(PHI)*(SIN(A)-SIN(B))
13020  C
13030        X3 = COS(DELTA)*COS(PHI)*(COS(C)+1.0)
13040  C
13050        RANGE = 4
13060        GO TO 900
13070  C
13080  C-------------------------------------------------------------------
13090  C
13100    500 A = PI*T/12.0
13110        B = PI*(T-1.0)/12.0
13120  C
13130        X2 = SIN(DELTA)*SIN(PHI)
13140        X2 = X2 - (12.0/PI)*COS(DELTA)*COS(PHI)*(SIN(A)-SIN(B))
13150  C
13160        X3 = 0.0
13170        RANGE = 5
13180        GO TO 900
13190  C
13200  C-------------------------------------------------------------------
13210  C
13220    600 A = PI*S/12.0
13230        B = PI*(SIGMA-1.0)/12.0
13240  C
13250        X2 = (S-SIGMA+1.0)*SIN(DELTA)*SIN(PHI)
```

```
13260         X2 = X2 + (12.0/PI)*COS(DELTA)*COS(PHI)*(SIN(B)-SIN(A))
13270   C
13280         X3 = 0.0
13290         RANGE = 6
13300         GO TO 900
13310   C
13320   C-----------------------------------------------------------------
13330   C
13340     700 X2 = 0.0
13350         X3 = 0.0
13360         RANGE = 7
13370   C
13380   C-----------------------------------------------------------------
13390   C
13400     900 CONTINUE
13410   C
13420   C
13430            IF ( SWICH1 .EQ. 0 ) GO TO 905
13440   C
13450         X2 = KBAR*X2
13460         X3 = KBAR*X3
13470   C
13480     905 X4 = 1.579E-8*(1.00+0.17*CLD**2.)*(TMPLAB+460.)**6.
13490         X5 = TGD*SWICH2
13500         X6 = WSP
13510         X7 = WDR
13520   C
13530   C
13540   C
13550            IF ( IDBUG.EQ.0) RETURN
13560   C
13570            WRITE (IWBUG,909) CLD, KBAR, TGD
13580     909    FORMAT (1H0, 'CLD=',E12.5,2X, 'KBAR=',E12.5,2X, 'TGD=',
13590        $          E12.5)
13600            WRITE(IWBUG,910) ST, X1, X2, X3, X4, XIWBUG, RANGE, T, A, B
13610     910    FORMAT(1X,F5.0,1X,5(E12.5,1X),I5,F10.5,2(1X,E12.5))
13620         RETURN
13630         END
13640   C
13650   C.....................................................
13660   C
13670         SUBROUTINE REGRES ( Y, X1, X2, X3, X4, X5, X6, X7, XY, N, XXT )
13680   C
13690   C
13700   C        SET UP THE VECTOR XY AND THE MATRIX XXT THAT ARE
13710   C        NECESSARY TO ESTIMATE THE REQUIRED TEMPERATURE EQUATION
13720   C        COEFFICIENTS.  THIS SUBROUTINE IS CALLED ONCE EACH
13730   C        TIME PERIOD. (I.E. EVERY TIME THE TEMPERATURE CHANGE, Y,
13740   C        IS COMPUTED)
13750   C
13760   C        Y.....TEMPERATURE CHANGE IN LAST TIME PERIOD
```

```
C        X1.....PREDICTOR X1 IN THE TEMPERATURE MODEL
C        X2.....PREDICTOR X2 IN THE TEMPERATURE MODEL
C        X3.....PREDICTOR X3 IN THE TEMPERATURE MODEL
C        X4.....PREDICTOR X4 IN THE TEMPERATURE MODEL
C        X5.....PREDICTOR X5 IN THE TEMPERATURE MODEL
C        X6.....PREDICTOR X6 IN THE TEMPERATURE MODEL
C        X7.....PREDICTOR X7 IN THE TEMPERATURE MODEL
C        XY.....VECTOR OBTAINED BY MULTIPLYING THE PREDICTOR VALUES
C                  BY THE OBSERVED TEMPERATURE CHANGES. ( THE
C                  ELEMENTS OF XY ARE SUMMATIONS)
C        N......DIMENSION OF XY
C        XXT....MATRIX OBTAINED BY POSTMULTIPLYING THE VECTOR
C                  X BY ITS TRANSPOSE. (THE ELEMENTS OF XXT ARE
C                  SUMMATIONS)
C
      DIMENSION XY(8), XXT(8,8), X(8)
C
      COMMON /YSTAT/ YSUM, YSUMSQ, YMEAN, RSQUAR
C
C
C        PUT PREDICTOR VALUES IN PREDICTOR VECTOR
C
      X(1) = 1.0
      X(2) = X1
      X(3) = X2
      X(4) = X3
      X(5) = X4
      X(6) = X6
      X(7) = X7
      X(8) = X5
C
C8000 WRITE (5,9000) (X(J),J=1,8)
C9000 FORMAT (1H0,'REGRES X VECTOR',5X,4(E12.5,5X)/T22,4(E12.5,5X))
C
C
C
C
C        UPDATE THE XY VECTOR
C
      DO 100 I = 1,N
      XY(I) = X(I)*Y + XY(I)
  100 CONTINUE
C
C
C        UPDATE THE XXT MATRIX
C
      DO 200 I = 1,N
      DO 200 J = 1,N
      XXT(J,I) = X(J)*X(I) + XXT(J,I)
  200 CONTINUE
C
```

```
14280   C
14290   C       UPDATE Y STATISTICS
14300   C
14310         YSUM   = Y + YSUM
14320         YSUMSQ = Y*Y + YSUMSQ
14330         YMEAN  = YSUM/XXT(1,1)
14340   C
14350   C
14360         RETURN
14370         END
14380   C
14390   C.............................................................
14400   C
14410   C
14420   C     ..........................................................
14430   C
14440   C       SUBROUTINE MINV
14450   C
14460   C       PURPOSE
14470   C          INVERT A MATRIX
14480   C
14490   C       USAGE
14500   C          CALL MINV(A,N,D,L,M)
14510   C
14520   C       DESCRIPTION OF PARAMETERS
14530   C          A - INPUT MATRIX, DESTROYED IN COMPUTATION AND REPLACED BY
14540   C              RESULTANT INVERSE.
14550   C          N - ORDER OF MATRIX A
14560   C          D - RESULTANT DETERMINANT
14570   C          L - WORK VECTOR OF LENGTH N
14580   C          M - WORK VECTOR OF LENGTH N
14590   C
14600   C       REMARKS
14610   C          MATRIX A MUST BE A GENERAL MATRIX
14620   C
14630   C       SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
14640   C          NONE
14650   C
14660   C       METHOD
14670   C          THE STANDARD GAUSS-JORDAN METHOD IS USED. THE DETERMINANT
14680   C          IS ALSO CALCULATED. A DETERMINANT OF ZERO INDICATES THAT
14690   C          THE MATRIX IS SINGULAR.
14700   C
14710   C     ..........................................................
14720   C
14730         SUBROUTINE MINV(A,N,D,L,M)
14740         DIMENSION A(1),L(1),M(1)
14750   C
14760   C       ..........................................................
14770   C
14780   C       IF A DOUBLE PRECISION VERSION OF THIS ROUTINE IS DESIRED, THE
```

```
14790  C        C IN COLUMN 1 SHOULD BE REMOVED FROM THE DOUBLE PRECISION
14800  C        STATEMENT WHICH FOLLOWS.
14810  C
14820  C     DOUBLE PRECISION A,D,BIGA,HOLD
14830  C
14840  C        THE C MUST ALSO BE REMOVED FROM DOUBLE PRECISION STATEMENTS
14850  C        APPEARING IN OTHER ROUTINES USED IN CONJUNCTION WITH THIS
14860  C        ROUTINE.
14870  C
14880  C        THE DOUBLE PRECISION VERSION OF THIS SUBROUTINE MUST ALSO
14890  C        CONTAIN DOUBLE PRECISION FORTRAN FUNCTIONS.  ABS IN STATEMENT
14900  C        10 MUST BE CHANGED TO DABS.
14910  C
14920  C        ...........................................................
14930  C
14940  C        SEARCH FOR LARGEST ELEMENT
14950  C
14960         D=1.0
14970         NK=-N
14980         DO 80 k=1,N
14990         NK=NK+N
15000         L(K)=K
15010         M(K)=K
15020         KK=NK+K
15030         BIGA=A(KK)
15040         DO 20 J=K,N
15050         IZ=N*(J-1)
15060         DO 20 I=K,N
15070         IJ=IZ+I
15080     10  IF( ABS(BIGA)- ABS(A(IJ))) 15,20,20
15090     15  BIGA=A(IJ)
15100         L(K)=I
15110         M(K)=J
15120     20  CONTINUE
15130  C
15140  C        INTERCHANGE ROWS
15150  C
15160         J=L(K)
15170         IF(J-K) 35,35,25
15180     25  KI=K-N
15190         DO 30 I=1,N
15200         KI=KI+N
15210         HOLD=-A(KI)
15220         JI=KI-K+J
15230         A(KI)=A(JI)
15240     30  A(JI) =HOLD
15250  C
15260  C        INTERCHANGE COLUMNS
15270  C
15280     35  I=M(K)
15290         IF(I-K) 45,45,38
```

```
15300      38 JP=N*(I-1)
15310         DO 40 J=1,N
15320         JK=NK+J
15330         JI=JP+J
15340         HOLD=-A(JK)
15350         A(JK)=A(JI)
15360      40 A(JI) =HOLD
15370   C
15380   C        DIVIDE COLUMN BY MINUS PIVOT (VALUE OF PIVOT ELEMENT IS
15390   C        CONTAINED IN BIGA)
15400   C
15410      45 IF(BIGA) 48,46,48
15420      46 D=0.0
15430         RETURN
15440      48 DO 55 I=1,N
15450         IF(I-K) 50,55,50
15460      50 IK=NK+I
15470         A(IK)=A(IK)/(-BIGA)
15480      55 CONTINUE
15490   C
15500   C        REDUCE MATRIX
15510   C
15520         DO 65 I=1,N
15530         IK=NK+I
15540         HOLD=A(IK)
15550         IJ=I-N
15560         DO 65 J=1,N
15570         IJ=IJ+N
15580         IF(I-K) 60,65,60
15590      60 IF(J-K) 62,65,62
15600      62 KJ=IJ-I+K
15610         A(IJ)=HOLD*A(KJ)+A(IJ)
15620         IF(ABS(A(IJ)).GT.1.E 37.OR.ABS(A(IJ)).LT.1.E-37)WRITE(5,936)A(IJ)
15630     936  FORMAT(1X,'A(IJ)=',1X,E12.5)
15640      65 CONTINUE
15650   C
15660   C        DIVIDE ROW BY PIVOT
15670   C
15680         KJ=K-N
15690         DO 75 J=1,N
15700         KJ=KJ+N
15710         IF(J-K) 70,75,70
15720      70 A(KJ)=A(KJ)/BIGA
15730      75 CONTINUE
15740   C
15750   C        PRODUCT OF PIVOTS
15760            IF((ALOG10(D)+ALOG10(BIGA)).LT.37.0) GO TO 77
15770            IF (IFLAG.GT.0) GO TO 76
15780            IFLAG=1
15790   C
15800            WRITE(5,800)
```

```
15810     800   FORMAT(///,1X,'SUBROUTINE MINV: DETERMINANT SIZE EXCEEDS ',/1X,
15820        $    'MACHINE CAPACITY. CALCULATION IS GREATER THAN 1.E+37. ',/1X,
15830        $    'PROCESSING CONTINUES'///)
15840           GO TO 76
15850      77 CONTINUE
15860           D=D*BIGA
15870           IF (D.GT.1.E 30)WRITE(5,935)D
15880     935 FORMAT(1X,'D=',1X,E12.5)
15890      76 CONTINUE
15900   C
15910   C        REPLACE PIVOT BY RECIPROCAL
15920   C
15930           A(KK)=1.0/BIGA
15940      80 CONTINUE
15950   C
15960   C        FINAL ROW AND COLUMN INTERCHANGE
15970   C
15980           K=N
15990     100 K=(K-1)
16000           IF(K) 150,150,105
16010     105 I=L(K)
16020           IF(I-K) 120,120,108
16030     108 JQ=N*(K-1)
16040           JR=N*(I-1)
16050           DO 110 J=1,N
16060           JK=JQ+J
16070           HOLD=A(JK)
16080           JI=JR+J
16090           A(JK)=-A(JI)
16100     110 A(JI) =HOLD
16110     120 J=M(K)
16120           IF(J-K) 100,100,125
16130     125 KI=K-N
16140           DO 130 I=1,N
16150           KI=KI+N
16160           HOLD=A(KI)
16170           JI=KI-K+J
16180           A(KI)=-A(JI)
16190     130 A(JI) =HOLD
16200           GO TO 100
16210     150 RETURN
16220           END
16230   C
16240   C.....................................................................
16250   C
16260           SUBROUTINE MATMLT (A, B, C, N)
16270   C
16280   C
16290   C
16300   C        MATMLT POST MULTIPLIES AN  N X N  MATRIX BY A
16310   C           VECTOR, A, OF LENGTH N. THE RESULT IS A VECTOR, C,
```

```
C          OF LENGTH N.
C
C
      DIMENSION A(8), B(8,8), C(8)
C
C
      DO 50 I = 1,N
   50 C(I) = 0.0
C
C
      DO 100 J = 1,N
      DO 100 I = 1,N
C
         C(I) = B(I,J)*A(J) + C(I)
C
  100 CONTINUE
C
C
      RETURN
      END
C.....
C
      SUBROUTINE ATOB (A,B)
C
C
C       DERIVE THE TEMPERATURE MODEL COEFICIENTS (I.E.
C       ELEMENTS OF VECTOR B) FROM THE REGRESSION VECTOR A
C
C
C       A.....REGRESSION COEFFICIENT VECTOR
C       B.....VECTOR OF TEMPERATURE MODEL COEFFICIENTS
C
C
C       A(1) = A0       B(1) = B0
C       A(2) = A1       B(2) = B1
C       A(3) = A2       B(3) = B2
C       A(4) = A3       B(4) = B3
C       A(5) = A4       B(5) = B4
C       A(6) = A5       B(6) = B5
C       A(7) = A6       B(7) = B6
C       A(8) = A7       B(8) = B7
C
C
C
      DIMENSION A(1),B(1)
C
C
C
C       DETERMINE B1
C
C
```

```
      B(2) = -ALOG(1.0+A(2))
C
C
C
      BA = B(2)/(-A(2))
C
C
C        DETERMINE BI'S
C
      DO 100 I = 1,8
C
         IF ( I .EQ. 2 ) GO TO 100
C
         B(I) = A(I)*BA
C
  100 CONTINUE
C
C
C8000 WRITE (5,900) (B(J),J=1,8)
C 900 FORMAT (1H0, '8 VECTOR', 5X,4(E12.5,5X)/T15,4(E12.5,5X))
C
C
C
      RETURN
      END
C
C
C.....................................................................
C
      SUBROUTINE DCHECK (JULREL, ID, DATA, L )
C
      DIMENSION DATA(1), TYPE(4), FMT(15)
C
      DATA FMT /'(1H ,8HJULREL =,1X,I3,  ''XXXX DATA OUT OF BOUNDS'',
     $2X,E12.5,3X,''L ='',2X,I2)'/
C
      DATA TYPE(1)/'TEMP'/, TYPE(2)/'CLD'/, TYPE(3)/'WSP'/,
     $     TYPE(4)/'WDIR'/
C
C     WRITE (5,100) FMT
  100 FORMAT (1H ,10(1X,A5)/)
C
      FMT(6) = TYPE(ID)
C
      WRITE (5,FMT) JULREL, DATA(L), L
C
C
      RETURN
      END
```

3.4 PROGRAM TMPGEN

PROGRAM TMPGEN uses the regression coefficients estimated by PROGRAM TEMPER and observations of cloud cover, wind speed, wind direction, and temperature (for updating purposes) to generate a time series of "predicted" temperatures. The predicted temperatures are written to a data file for later statistical analysis and comparison with the observed temperature time series.

The user has the option of generating the deterministic component alone or generating both the deterministic component and the stochastic component of temperature. This allows the user to first generate the deterministic component of predicted temperatures which can then be compared to the observed temperatures. The resulting time series of deviations can then be analyzed by standard techniques to determine the required parameters for the deviations or stochastic component.

Next, the user can generate a "predicted" temperature time series that includes the deterministic and stochastic components. The resulting time series can be analyzed to determine if its statistical properties are acceptable.

Execution of PROGRAM TMPGEN begins after the user responds to three computer inquires at an interactive computer terminal.

1. Computer: DO YOU WISH TO ADD THE RANDOM COMPONENT? (YES/NO)

   User: Answer YES, if the stochastic component is to be added to the deterministic component.

   Answer NO if only the deterministic component is desired.

2. If the response to 1 above was YES:

   Computer: INPUT SEED FOR RANDOM NUMBER GENERATOR

   User: Enter any positive integer value that is within the machine capacity

3. Computer: MONTHLY UPDATE IS ASSUMED. ENTER 1 FOR DAILY

   User: Enter 1 for daily updating
   Enter any other number for monthly updating

   Note: At the beginning of each daily temperature simulation, the 11 PM temperature from the previous day is required as a starting point. (See Curtis and Eagleson (1982), Chapter 6.2-6.3) If the user chooses daily updating. the observed 11 PM temperature is used as the starting point. If the user choses monthly updating, the observed temperature is only used at the beginning of the month. To develop a predicted time series for estimation of the deviations sequence, monthly updating is used.

Three additional data files are used by PROGRAM TMPGEN. File DCTEMP.DAT contains the remaining input data and its format is described in the first section of the Fortran listing for PROGRAM TMPGEN. File DCTEMP.OUT is reserved for the predicted temperatures. Each day's data has the format (16F5.1/8F5.1, 5X, I5). The 24 values represent predicted temperatures for Midnight, 1 AM,..., 11 AM, Noon, 1 PM,..., 11 PM. The integer field, I5, on the second record is reserved for the Julian date. File DCTEMP.BUG is reserved for debug output when any internal debug flags are turned on.

3.5 Fortran Listing for PROGRAM TMPGEN

```
00010   C
00020   C.............................................................
00030   C
00040         PROGRAM TMPGEN
00050   C
00060   C
00070   C  THE REGRESSION COEFFICIENTS GENERATED BY PROGRAM TEMPER ARE
00080   C  USED WITH OBSERVATIONS OF CLOUD COVER, WIND SPEED, WIND DIRECTION,
00090   C  AND DEWPOINT TO CREATE A TIME SERIES OF PREDICTED TEMPERATURES.
00100   C  THE PREDICTED TEMPERATURES ARE OUTPUT FOR LATER STATISTICAL
00110   C  ANALYSIS AND FOR COMPARISON WITH THE OBSERVED TEMPERATURES.
00120   C
00130   C  THE USER HAS THE OPTION OF UPDATING THE TEMPERATURE COMPUTATIONS
00140   C  WITH AN OBSERVED TEMPERATURE VALUE ON A MONTHLY BASIS OR ON A
00150   C  DAILY BASIS.
00160   C
00170   C  THE PROGRAM IS CURRENTLY SET UP TO RUN ON A DEC-10 TIME SHARE
00180   C  SYSTEM.
00190   C
00200   C
00210   C  PROGRAMMER:  DAVID C. CURTIS
00220   C               NATIONAL WEATHER SERVICE
00230   C               NORTHEAST RIVER FORECAST CENTER
00240   C               705 BLOOMFIELD AVENUE
00250   C               BLOOMFIELD, CT 06002
00260   C
00270   C               203-722-2014
00280   C
00290   C
00300   C  INPUT DATA FILE FORMAT
00310   C
00320   C  CARD 1 ************************************************
00330   C
00340   C    COL  4-5  I2    BEGINNING DAY
00350   C         9-10 I2    BEGINNING MONTH
00360   C        12-15 I4    BEGINNING YEAR (4 DIGITS)
00370   C        19-20 I2    ENDING DAY
00380   C        24-25 I2    ENDING MONTH
00390   C        27-30 I4    ENDING YEAR (4 DIGITS)
00400   C
00410   C  CARD 2 ************************************************
00420   C
00430   C    COL  1-5  F5.0  DEGREES LATITUDE
00440   C         6-10 F5.0  MINUTES
00450   C        11-15 F5.0  SECONDS
00460   C        21-25 F5.0  DEGREES LONGITUDE
00470   C        26-30 F5.0  MINUTES
00480   C        31-35 F5.0  SECONDS
00490   C        51-58 A8    STANDARD TIME ZONE:  'EASTERN'
00500   C                                         'CENTRAL'
```

```
00510   C                                       'MOUNTAIN'
00520   C                                       'PACIFIC'
00530   C
00540   C  CARD 3 ***************************************************
00550   C
00560   C    COL  1-5  F5.0  +1 FOR EAST LONGITUDE
00570   C                     -1 FOR WEST LONGITUDE
00580   C         6-10 F5.0  DIFFERENCE BETWEEN TRUE SOLAR TIME AND MEAN
00590   C                    SOLAR TIME (USUALLY NEGLECTED FOR HEAT
00600   C                    TRANSFER COMPUTATIONS) I.E. ET = 0.0
00610   C        11-15 F5.0  SOLAR CONSTANT
00620   C
00630   C  CARD 4 ***************************************************
00640   C
00650   C    COL  1-48 4E12.5 REGRESSION COEFFICIENTS B0-B3
00660   C
00670   C  CARD 5 ***************************************************
00680   C
00690   C    COL  1-48 4E12.5 REGRESSION COEFFICIENTS B4-B7
00700   C
00710   C  CARD 6 ***************************************************
00720   C
00730   C    COL  1-10  F10.0 TEMPERATURE DEVIATIONS BIAS
00740   C         11-20  F10.0 TEMPERATURE DEVIATIONS STANDARD DEVIATION
00750   C         21-30  F10.0 TEMPERATURE DEVIATIONS LAG-1 AUTOCORRELATION
00760   C
00770   C  CARD 7-N *************************************************
00780   C
00790   C    THE REMAINING RECORDS IN THE FILE CONTAIN THE DATA OBSERVATIONS
00800   C    IN DAILY SEGMENTS IN THE FOLLOWING HOURLY ORDER:
00810   C    11PM, MIDNIGHT, 1AM, ... ,11AM, NOON, 1PM, ... , 11PM .
00820   C    THE 25 VALUES ARE DISTRIBUTED OVER TWO RECORDS (16 VALUES ON
00830   C    THE FIRST AND 9 VALUES ON THE SECOND). FOR EACH DAY THE DATA
00840   C    ORDER IS TEMPERATURE, CLOUD COVER, WIND SPEED, AND WIND DIR.
00850   C    THE READ FORMAT FOR THE OBSERVED DATA IS (16F5.0)
00860   C
00870   C ***********************************************************
00880   C
00890   C
00900   C
00910         INTEGER RANGE
00920         INTEGER SEASON(12)
00930         INTEGER RTEST
00940   C
00950   C
00960         REAL*8 DEBUG(20)
00970         REAL*8 TZONE,TZ(4)
00980         REAL KBAROB
00990   C
01000         DIMENSION A(3),B(3)
01010         DIMENSION TPRIME(25), THAT(25)
```

```
01020       DIMENSION KBAROB(25), GRTEMP(25), CLOUD(25)
01030       DIMENSION TEMDAT(25)
01040       DIMENSION WSPEED(25), WDIR(25)
01050       DIMENSION  BCOEF(8)
01060 C
01070       COMMON /IO/ IREAD,IWRITE,IWBUG
01080       COMMON /DBUG/ NBUG, DEBUG
01090       COMMON /RTESTR/ RTEST
01100       COMMON /SEAS/ NSEAS
01110       COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
01120       COMMON /SUN/   DELTA, DTSL, SR,SS
01130       COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
01140      $               ,JSTART, JSTOP,  JREND,  JYEAR
01150       COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
01160       COMMON /TBEGIN/ TPR, BCOEF
01170       COMMON /SEED/ ISEED
01180       COMMON /TDSTAT/ TDBIAS, TDSDEV, TDRHO
01190 C
01200       DATA TZ/8HEASTERN  ,8HCENTRAL  ,8HMOUNTAIN  ,8HPACIFIC  /
01210 C
01220 C
01230 C
01240 C
01250 C  INTERACTIVE TERMINAL INPUT
01260 C
01270       CALL TRTEMP (IREAD,IWRITE,IWBUG,IUPDAT)
01280 C
01290 C
01300 C
01310 C
01320 COOOOOOOOOOOOOOOOOOOOO.......OPEN........OOOOOOOOOOOOOOOOOOOO
01330 C
01340       OPEN (UNIT=IREAD,DEVICE='DSK',ACCESS='SEQIN',FILE='DCTEMP.DAT')
01350       OPEN (UNIT=IWRITE,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCTEMP.OUT')
01360       OPEN (UNIT=IWBUG,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCTEMP.BUG')
01370 C
01380 COOOOOOOOOOOOOOOOOOOOO.......OPEN........OOOOOOOOOOOOOOOOOOOO
01390 C
01400 C
01410 C
01420 C       READ FIRST SECTION OF DATA FROM INPUT DATA FILE
01430 C
01440       CALL INPUT
01450 C
01460 C
01470       IRANK = 7
01480       TDPREV = 0.0
01490 C
01500 C
01510 C       INITIALIZE DATE VARIABLES
01520 C
```

```
01530         CALL DATE1
01540   C
01550   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
01560   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
01570   C:::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
01580   C
01590   C        FORECAST TEMPERATURES FOR THE REMAINDER OF THE PERIOD
01600   C
01610   C
01620   C
01630     400 CONTINUE
01640   C
01650   C
01660   C
01670               IDBUG = 0
01680               IF(IDBUG.EQ.1)WRITE (IWBUG,805) (CLOUD(K),KBAROB(K),K=1,16)
01690     805       FORMAT (1H ,16F6.3)
01700   C
01710   C
01720   C   READ DATA TO FORECAST TEMPERATURES
01730   C     (USE ENTRY POINT IN SUBROUTINE INPUT)
01740   C
01750   C
01760         CALL DATA ( TEMDAT, CLOUD, GRTEMP, WSPEED, WDIR )
01770   C
01780   C
01790   C    PERFORM NECESSARY DATA CONVERSIONS
01800   C
01810            DO 810 IC = 1,25
01820   C
01830               KBAROB(IC) = 1.00 - 0.65*CLOUD(IC)**2.
01840               IF ( WDIR(IC) .LE. 180. ) GO TO 810
01850               WDIR(IC) = ABS( WDIR(IC) - 360. )
01860   C
01870               IF ( IDBUG .EQ. 0 ) GO TO 810
01880               WRITE (IWBUG,806) IC, CLOUD(IC), KBAROB(IC), WDIR(IC)
01890     806       FORMAT (1H ,'CLOUD AND WIND SPEED TRANSFORMATIONS',
01900        $              I5,3(2X,E12.5))
01910   C
01920     810 CONTINUE
01930   C
01940   C
01950   C   UPDATE ORBIT PARAMETERS
01960   C
01970         CALL DECL (JULREL,DELTA,SR,SS)
01980   C
01990         DTSL = EP*(THETAS-THETAL)*3.81972
02000   C
02010   C
02020   C  INITIALIZE PREVIOUS TEMPERATURE AND PREVIOUS DEVIATION
02030   C
```

```
02040           IF(JULREL.EQ.JSTART) TPRIME(1)=TEMDAT(1)
02050           IF(JULREL.EQ.JSTART) TDPREV = 0.0
02060   C
02070   C
02080   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02090   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02100   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02110   C
02120   C
02130   C  GENERATE TODAY'S TEMPERATURES
02140   C
02150         CALL TMPSIM (TPRIME, KBAROB, CLOUD, GRTEMP, WSPEED, WDIR,
02160        $             BCOEF, THAT)
02170   C
02180           TPRIME(1) = THAT(24)
02190           IF ( IUPDAT .EQ. 1 ) TPRIME(1) = TEMDAT(24)
02200   C
02210   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02220   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02230   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02240   C
02250   C
02260   C        ADD RANDOM COMPONENT TO TEMPERATURE
02270   C
02280           IF ( RTEST .EQ. 0 ) GO TO 501
02290   C
02300           DO 500 I = 1,24
02310   C
02320             CALL NORMAL (ARV)
02330             TDEV = TDBIAS + TDRHO*(TDPREV-TDBIAS) +
02340        $           ARV*TDSDEV*(1.00-TDRHO*TDRHO)**0.50
02350   C
02360             TDPREV = TDEV
02370   C
02380             THAT(I) = THAT(I) + TDEV
02390   C
02400     500   CONTINUE
02410     501 CONTINUE
02420   C
02430   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02440   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02450   C::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
02460   C
02470   C
02480   C
02490   C        OUTPUT ESTIMATED TEMPERATURES
02500   C
02510              WRITE (IWRITE,270) (THAT(I),I=1,24),JULREL
02520     270      FORMAT (16F5.1/8F5.1,5X,I5)
02530   C
02540   C
```

```
02550   C
02560   C        UPDATE JULIAN DATE COUNTERS
02570   C
02580   C         USE ENTRY DATE IF REGRESSION COEFFICIENTS USED FOR THE
02590   C         ENTIRE YEAR
02600   C
02610   C         USE ENTRY DATEM IF REGRESSION COEFFICIENTS USED FOR
02620   C         ONE MONTH PER YEAR
02630   C
02640   C-----CALL DATE
02650         CALL DATEM
02660   C
02670   C
02680         IF (JULDAT .LE. JULEND ) GO TO 400
02690   C
02700   C
02710   C
02720   C
02730   CCCCCCCCCCCCCCCCCCCCC.....CLOSE.....CCCCCCCCCCCCCCCCCCC
02740   C
02750         CLOSE (UNIT=IREAD,DEVICE='DSK',ACCESS='SEQIN',FILE='DCTEMP.DAT')
02760         CLOSE (UNIT=IWRITE,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCTEMP.OUT')
02770         CLOSE (UNIT=IWBUG,DEVICE='DSK',ACCESS='SEQOUT',FILE='DCTEMP.BUG')
02780   C
02790   CCCCCCCCCCCCCCCCCCCCC.....CLOSE.....CCCCCCCCCCCCCCCCCCC
02800   C
02810         WRITE (5,9998) IREAD, IWRITE, IWBUG
02820    9998 FORMAT (///1H0,T2, 'FILES USED',T30, 'UNIT', T40,'CONTENTS'/
02830        1              T2,'.........',T30,  '....', T40.'........'/
02840        2 T2, 'DCTEMP.DAT', T30,I4, T40, 'INPUT DATA'/
02850        3 T2, 'DCTEMP.OUT', T30,I4, T40, 'OUTPUT DATA SET'/
02860        4 T2, 'DCTEMP.BUG', T30,I4, T40, 'DEBUG OUPUT (IF ANY)')
02870   C
02880         WRITE (5,9999)
02890    9999 FORMAT (////1H0,'TEMPERATURE PROGRAM SUCCESSFULLY EXECUTED'///)
02900   C
02910   C
02920   C
02930         RETURN
02940         END
02950   C
02960   C.....
02970   C
02980   C
02990         SUBROUTINE NORMAL (X)
03000   C
03010   C  GENERATE A NORMALLY DISTRIBUTED RANDOM DEVIATE FROM N(0,1).
03020   C  REFERENCE:  FISHMAN,GEORGE S.:CONCEPTS AND METHODS IN DISCRETE EVENT
03030   C              DIGITAL SIMULATION, WILEY AND SONS, 1973, PP 211.
03040   C
03050      10 U1 = RAND(0)
```

```
03060         IF (U1 .LT. 0.00001) GO TO 10
03070         U2 = RAND(0)
03080         X = SQRT (-2.0*ALOG(U1))*COS(6.28319*U2)
03090   C
03100         RETURN
03110         END
03120   C.....................................................
03130   C
03140         SUBROUTINE TRTEMP (IREAD,IWRITE,IWBUG,IUPDAT)
03150   C
03160   C
03170         INTEGER A1(6)
03180         INTEGER RTEST
03190         REAL*8 FILE
03200         REAL*8 DEBUG(20),TZONE,TZ(4)
03210         DIMENSION A(3),B(3),BC(7)
03220         DIMENSION TPRIME(25)
03230   C
03240         COMMON /TTEST/ ZTEST
03250         COMMON /SEED/ ISEED
03260         COMMON /RTESTR/ RTEST
03270   C
03280   C
03290         IREAD = 21
03300         IWRITE = 22
03310         IWBUG = 23
03320   C
03330   C
03340         WRITE (5,710)
03350     710 FORMAT(//1H0,'DO YOU WISH TO ADD THE RANDOM COMPONENT? (YES/NO)')
03360   C
03370         RTEST = 0
03380         READ (5,720) RT
03390     720 FORMAT (A)
03400   C
03410         IF ( RT .EQ. 'YES' ) RTEST = 1
03420         IF ( RT .NE. 'YES' ) GO TO 702
03430   C
03440   C
03450   C
03460   C
03470         WRITE (5,700)
03480     700 FORMAT (1H0, 'INPUT SEED FOR RANDOM NUMBER GENERATOR')
03490   C
03500   C
03510         READ (5,701) ISEED
03520     701 FORMAT (I)
03530         CALL START (ISEED)
03540   C
03550     702 CONTINUE
03560   C
```

```
C
      IUPDAT = 0
C
      WRITE (5,951)
  951 FORMAT (///1X,'MONTHLY UPDATE ASSUMED.  ENTER 1 FOR DAILY.')
C
      READ (5,955) IUPD
  955 FORMAT (I)
C
      IF ( IUP .EQ. 1 ) IUPDAT = 1
C
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE INPUT
C
C
      INTEGER SEASON(12)
C
      REAL*8 DEBUG(20)
      REAL*8 TZONE,TZ(4)
C
      DIMENSION A(3), B(3)
      DIMENSION CLOUD(25), GRTEMP(25), WSPEED(25), WDIR(25)
      DIMENSION BCOEF(8)
      DIMENSION TEMDAT(25)
C
C
      COMMON /IO/ IREAD,IWRITE,IWBUG
      COMMON /DBUG/ NBUG, DEBUG
      COMMON /SEAS/ NSEAS
      COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
      COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
      COMMON /TBEGIN/ TPR, BCOEF
      COMMON /TDSTAT/ TDBIAS, TDSDEV, TDRHO
C
      DATA TZ/8HEASTERN  ,8HCENTRAL  ,8HMOUNTAIN  ,8HPACIFIC  /
C
C
C
C
C
C
C
CXXXXXREAD (IREAD,100) IREAD, IWRITE, IWBUG
CX100 FORMAT (3I5)
C
      GO TO 111
```

```
C
      READ (IREAD,110) NBUG, (DEBUG(I), I=1,NBUG)
  110 FORMAT (I5,5X,7(A8,2X)/(10X,7(A8,2X)))
C
 8000 WRITE (5,9000) NBUG
 9000 FORMAT (1H0,'NBUG=',2X,I10)
  111 CONTINUE
C
C
      READ (IREAD,120) IDAY,IMO,IYR,LDAY,LMO,LYR
  120 FORMAT (2(3X,I2,3X,I2,1X,I4))
C
C
C
C
C
C
C
      READ (IREAD,140) (A(I),I=1,3),(B(I),I=1,3),TZONE
  140 FORMAT (2(3F5.0,5X),T51,A8)
C
      PHI = DMS(A)
      THETAL = DMS(B)
C
      IF (TZONE.NE.TZ(1)) GO TO 200
      THETAS = 75.0*2.0*3.14159/360.0
      GO TO 300
C
  200 IF (TZONE.NE.TZ(2)) GO TO 210
      THETAS =  90.0*2.0*3.14159/360.
      GO TO 300
C
  210 IF (TZONE.NE.TZ(3)) GO TO 220
      THETAS = 105.0*2.0*3.14159/360.0
      GO TO 300
C
  220 IF (TZONE.NE.TZ(4)) GO TO 230
      THETAS = 120.0*2.0*3.14159/360.0
      GO TO 300
C
  230 WRITE (IWRITE,240)
  240 FORMAT (1H1,'TIME ZONE REQUESTED IS NOT VALID'///)
      WRITE (IWRITE,250) TZONE,(TZ(I),I=1,4)
  250 FORMAT (1H0,T10,'REQUESTED TIME ZONE',T36,5H*****,
     1A8,5H*****/T10,'AVAILABLE TIME ZONES',T36,5H*****,
     2A8,5H*****/T36,5H*****,A8,5H*****/T36,5H*****,
     3A8,5H*****/T36,5H*****,A8,5H*****)
C
  300 CONTINUE
C
C
```

```
04590         READ (IREAD,260) EP,ET,W
04600     260 FORMAT (16F5.0)
04610   C
04620   C
04630   C
04640         READ (IREAD,270) (BCOEF(I),I=1,8)
04650     270 FORMAT (4E12.5/4E12.5)
04660   C
04670   C
04680         READ (IREAD,280) TDBIAS,TDSDEV, TDRHO
04690     280 FORMAT (3F10.0)
04700   C
04710   C
04720   C
04730         RETURN
04740   C
04750     310 CONTINUE
04760   C
04770         ENTRY DATA (TEMDAT,CLOUD,GRTEMP,WSPEED,WDIR)
04780   C
04790   C
04800            READ (IREAD,260) TEMDAT
04810            READ (IREAD,260) CLOUD
04820            READ (IREAD,260) WSPEED
04830            READ (IREAD,260) WDIR
04840   C
04850   C
04860   C           GROUND TEMPERATURE IS NOT USED AS A PREDICTOR YET
04870   C
04880         RETURN
04890         END
04900   C
04910   C
04920   C
04930   C
04940   C.....................................................
04950   C
04960         SUBROUTINE DATE1
04970   C
04980   C        DATE1 INITIALIZES THE DATE COUNTERS.
04990   C        JULIAN DATES ARE USED.
05000   C
05010   C
05020   C        IYR   ... INITIAL YEAR
05030   C        IMO   ... INITIAL MONTH
05040   C        IDAY  ... INITIAL DAY
05050   C        LYR   ... LAST YEAR
05060   C        LMO   ... LAST MONTH
05070   C        LDAY  ... LAST DAY
05080   C
05090   C        JULDAT... CURRENT JULIAN DATE
```

```
05100   C       JBEGIN... JULIAN DATE AT BEGINNING OF RUN
05110   C       JULEND... JULIAN DATE AT END OF RUN
05120   C       JRANGE... LENGTH OF RUN
05130   C       JULREF... JAN 1 OF INITIAL YEAR
05140   C       JULREL... JULIAN DATE RELATIVE TO JAN 1 OF CURRENT YEAR
05150   C       JSTART... RELATIVE JULIAN DATE TO BEGIN MONTHLY PARAMETER
05160   C                 ESTIMATION RANGE
05170   C       JSTOP ... RELATIVE JULIAN DATE TO END MONTHLY PARAMETER
05180   C                 ESTIMATION RANGE
05190   C       JYEAR ... YEAR COUNTER
05200   C
05210   C       NXLPYR... JULIAN DATE OF DEC 31 OF NEXT LEAP YEAR
05220   C
05230   C
05240   C
05250         COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
05260         COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
05270        $              ,JSTART, JSTOP,  JREND,  JYEAR
05280         COMMON /IO/    IREAD, IWRITE, IWBUG
05290   C
05300         INTEGER IDBUG
05310   C
05320   C       SET DEBUG FLAG
05330   C
05340           IDBUG = 0
05350   C
05360   C
05370   C
05380   C       DETERMINE INITIAL JULIAN DATES
05390   C
05400         CALL JULIAN (IMO,IDAY,IYR,JBEGIN)
05410         CALL JULIAN (LMO,LDAY,LYR,JULEND)
05420         CALL JULIAN (  1,   1,IYR,JULREF)
05430   C
05440   C
05450         JULREL = JBEGIN - JULREF
05460         JRANGE = JULEND - JBEGIN + 1
05470         JULDAT = JBEGIN - 1
05480   C
05490   C       DETERMINE THE NEXT OCCURANCE OF 12/31/(LEAP YEAR)
05500   C             (IE. THE 366TH DAY OF THE YEAR)
05510   C
05520   C
05530         LASTLP = IYR - MOD(IYR,4)
05540   C
05550         CALL JULIAN (12,31,LASTLP,NXLPYR)
05560   C
05570         IF (JULDAT .GE. NXLPYR) NXLPYR = NXLPYR + 1461
05580   C
05590   C
05600   C       NOTE... 1461 = 365 + 365 + 365 + 366
```

```
C
C
C     THIS SECTION DEFINES VARIABLES NEEDED FOR MONTHLY
C     PARAMETER ESTIMATION
C
        JYEAR = IYR
        JSTART = JULREL + 1
        CALL JULIAN ( LMO, LDAY, IYR, JDATE )
        JSTOP = JDATE - JULREF + 1
        JREND = JSTOP
          IF (JYEAR - MOD(JYEAR,4)) 65,70,65
   70     IF (IMO.EQ.2 .AND. IDAY.EQ.28) JREND = JSTOP + 1
   65   CONTINUE
C
C
      ENTRY DATE
C
C       THE NEXT SECTION IS USED EACH DAY TO UPDATE
C          THE JULIAN DATE COUNTERS.
C
      JULREL = JULREL + 1
      JULDAT = JULDAT + 1
C
C
C       CHECK FOR END OF YEAR
C
      IF (JULREL .LE. 365) GO TO 100
      IF (JULREL .GT. 366) GO TO 200
C
C       CHECK FOR LEAP YEAR
C
      IF (JULDAT.NE.NXLPYR) GO TO 200
C
C       YES, THERE ARE 366 DAYS THIS YEAR.
C       UPDATE NXLPYR TO NEXT LEAP YEAR.
C
      NXLPYR = NXLPYR + 1461
C
C
      IF( IDBUG .NE. 0 ) GO TO 900
C
C
   50 CONTINUE
C
C
      RETURN
C
C
C       RESET RELATIVE JULIAN DATE
C
  200 JULREL = 1
```

```
06120   C
06130   C
06140     100 CONTINUE
06150           IF( IDBUG .NE. 0 ) GO TO 900
06160     110 RETURN
06170   C
06180   C
06190         ENTRY DATEM
06200   C
06210   C     THIS SECTION IS USED EACH DAY TO UPDATE THE JULIAN DATE COUNTERS
06220   C     IF MONTHLY PARAMETER ESTIMATION IS USED.
06230   C
06240           JULREL = JULREL + 1
06250           JULDAT = JULDAT + 1
06260   C
06270           IF (JULREL .LE. JREND) GO TO 400
06280   C
06290   C     UPDATE THE JULIAN COUNTERS
06300   C
06310           JULREL = JSTART
06320           JYEAR  = JYEAR + 1
06330           CALL JULIAN (IMO,IDAY,JYEAR,JULDAT)
06340           JREND = JSTOP
06350           IF(JYEAR - MOD(JYEAR,4)) 400,410,400
06360     410   IF (LMO.EQ.2 .AND. LDAY.EQ.28) JREND = JREND + 1
06370     400   CONTINUE
06380   C
06390           IF ( IDBUG .NE. 0 ) GO TO 900
06400   C
06410   C
06420           RETURN
06430   C
06440   C
06450   C
06460     900 CONTINUE
06470   C
06480   C        DEBUG INFORMATION FOR JULIAN DATE CALCULATIONS
06490   C
06500         WRITE (IWBUG,920) JULDAT,JULREL,JBEGIN,JULEND,JRANGE,NXLPYR
06510        $                    ,JSTART,JSTOP,JREND,JYEAR
06520     920 FORMAT (1H ,'JULDAT=',I10,3X,'JULREL=',I10,3X,'JBEGIN=',I10,3X,
06530        $        /2X,'JULEND=',I10,3X,'JRANGE=',I10,3X,'NXLPYR=',I10,
06540        $        /2X,'JSTART=',I10,3X,'JSTOP =',I10,3X,'JREND =',I10,
06550        $         3X,'JYEAR=',I10)
06560   C
06570         RETURN
06580   C
06590   C
06600         END
06610   C
06620   C.............................................................
```

```
06630   C
06640         SUBROUTINE DATT(IDATE,IMO,IDAY,IYR)
06650   C
06660   C     CONVERT JULIAN DATE TO CALENDER DATE
06670   C
06680         INTEGER CAL(12,2)
06690         DATA CAL/0,31,59,90,120,151,181,212,243,273,304,334,
06700        1  0,31,60,91,121,152,182,213,244,274,305,335 /
06710         I1=(IDATE-1)/1461
06720         I2=IDATE-(I1*1461)
06730   C
06740   C       1<=I2<=1461
06750   C
06760         IF(I2.LE.365) GO TO 10
06770         IF(I2.LE.730) GO TO 20
06780         IF(I2.LE.1095) GO TO 30
06790         I3=3
06800         I4=I2-1095
06810         GO TO 40
06820      10 I3=0
06830         I4=I2
06840         I4=I2-365
06850         GO TO 40
06860      20 I3=1
06870         GO TO 40
06880      30 I3=2
06890         I4=I2-730
06900      40 IYR=1900+I3+(4*I1)
06910         INDX=1
06920         IF(I3.EQ.3)INDX=2
06930         DO 100 I=2,12
06940         IF(I4.LE.CAL(I,INDX)) GO TO 200
06950     100  CONTINUE
06960          IMO=12
06970         IDAY=I4-CAL(12,INDX)
06980          RETURN
06990     200 IMO=I-1
07000         IDAY=I4-CAL(I-1,INDX)
07010         RETURN
07020         END
07030   C
07040   C
07050   C.............................................................
07060   C
07070         SUBROUTINE JULIAN(MO,DA,YR,ANS)
07080         INTEGER ANS,CAL(12),DA,YR
07090         DATA CAL /31,28,31,30,31,30,31,31,30,31,30,31/
07100   C
07110   C        COMPUTE JULIAN DATE FROM JAN. 1, 1973
07120   C
07130         ANS=0
```

```
07140         I=YR-1900
07150         ANS=ANS+365*I
07160         CAL(2)=28
07170         IF(MOD(YR,4).EQ.0) CAL(2)=29
07180         J=MO-1
07190         IF(J.EQ.0) GO TO 20
07200         DO 10 I=1,J
07210         ANS=ANS+CAL(I)
07220      10 CONTINUE
07230      20 CONTINUE
07240         ANS=ANS+DA
07250         RETURN
07260         END
07270   C
07280   C.....................................................................
07290   C
07300         FUNCTION DMS(A)
07310   C
07320   C     FUNCTION DMS CONVERTS ANGLES EXPRESSED IN
07330   C     DEGREES, MINUTES AND SECONDS TO RADIANS
07340   C
07350         DIMENSION A(3)
07360         REAL MINUTE
07370   C
07380   C
07390         DEGREE = A(1)
07400         MINUTE = A(2)
07410         SECOND = A(3)
07420   C
07430         DMS = DEGREE*3.14159/180. + MINUTE*3.14159/180./60.
07440        1      + SECOND*3.14159/180./60./60.
07450   C
07460         RETURN
07470         END
07480   C
07490   C.....................................................................
07500   C
07510         FUNCTION TAU(ST)
07520   C
07530         COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
07540         COMMON /IO/ IREAD, IWRITE, IWBUG
07550         COMMON /DBUG/ NBUG,DEBUG
07560         REAL*8    ITAU,DEBUG(20)
07570         DATA ITAU /'TAU'/
07580   C
07590   C
07600   C     THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
07610   C              75TH MERIDIAN FOR EASTERN STANDARD TIME
07620   C              90TH MERIDIAN FOR CENTRAL STANDARD TIME
07630   C              105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
07640   C              120TH MERIDIAN FOR PACIFIC STANDARD TIME
```

```
C     THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
C     TAU    = LOCAL HOUR ANGLE
C     ST     = STANDARD TIME IN THE TIME ZONE OF THE
C              OBSERVER IN HOURS COUNTED FROM
C              MIDNIGHT (EG. 0.00 TO 24.00)
C     EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
C     ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME
C              AND MEAN SOLAR TIME (USUALLY NEGLECTED
C              FOR HEAT TRANSFER COMPUTATIONS)
C
C
C     FUNCTION SUBROUTINE TAU CONVERTS THE OBSERVERS
C     STANDARD TIME TO LOCAL HOUR ANGLE IN RADIANS
C
C     OBTAIN TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
C     OBSERVERS MERIDIAN (HOURS)
C
      DTSL = EP*(THETAS - THETAL)* 12.0/3.14159
C
C     COMPUTE OBSERVERS HOUR ANGLE (RADIANS). E = +1 FOR
C     MORNING AND E = -1 FOR AFTERNOON (I.E. SOLARNOON)
C
      IF (ST.GT.12. + DTSL -ET) E = -1.0000
      IF (ST.LE.12. + DTSL -ET) E = +1.0000
C
      TAU = (ST + E*12. - DTSL + ET) * 3.14159/12.0
C
      IF (TAU.GT.6.283185) TAU = TAU - 6.283185
      IF (TAU.LT.0.0) TAU = TAU + 6.283185
C
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 100
      DO 200 I = 1,NBUG
C
      IF (DEBUG(I).NE.ITAU) GO TO 200
      WRITE (IWBUG,250)    ST,PHI,THETAS,THETAL,EP,ET,W,DTSL,TAU
  250 FORMAT (////1H ,'FUNCTION TAU',  2X,
     1 'ST =',F6.3,2X,'PHI =',F6.3,2X,'THETAS =',F6.3,2X,'THETAL ='
     2 F6.3,2X,'EP =',F6.3,2X,'ET =',F6.3,2X,'W =',F6.3,2X,
     3 'DTSL =',F6.3,2X,'TAU =',F6.3)
  200 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C.....
C
```

```
08160       SUBROUTINE DECL (RJD,DELTA,SR,SS)
08170       INTEGER RJD
08180       COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
08190       COMMON /IO/ IREAD, IWRITE, IWBUG
08200       COMMON /DBUG/ NBUG,DEBUG
08210       REAL*8    IDECL,DEBUG(20)
08220       DATA IDECL/'DECL'/
08230  C
08240  C    DELTA  = DECLINATION OF THE SUN (RADIANS)
08250  C    PHI    = OBSERVERS LATITUDE (RADIANS)
08260  C    THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
08270  C             75TH MERIDIAN FOR EASTERN STANDARD TIME
08280  C             90TH MERIDIAN FOR CENTRAL STANDARD TIME
08290  C             105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
08300  C             120TH MERIDIAN FOR PACIFIC STANDARD TIME
08310  C    THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
08320  C    RJD    = RELATIVE JULIAN DATE (I.E. WITH RESPECT TO JAN 1)
08330  C    ST     = STANDARD TIME IN THE TIME ZONE OF THE OBSERVER
08340  C             IN HOURS COUNTED FROM MIDNIGHT (E.G.0.00 TO 24.00)
08350  C    EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
08360  C    ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME AND
08370  C             MEAN SOLAR TIME (USUALLY NEGLECTED FOR
08380  C             HEAT TRANSFER COMPUTATIONS)
08390  C
08400  C COMPUTE TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
08410  C OBSERVERS MERIDIAN (HOURS)
08420  C
08430       DTSL   = EP*(THETAS - THETAL)* 3.81972
08440  C
08450  C COMPUTE DECLINATION OF THE SUN (RADIANS)
08460  C
08470       DELTA  = 0.4093*COS(0.0172*(172. - FLOAT(RJD)) )
08480  C
08490  C COMPUTE HOUR ANGLE AT SUNSET (RADIANS)
08500  C
08510  C
08520       TSS    = ACOS(-TAN(DELTA)*TAN(PHI))
08530  C    COMPUTE STANDARD TIME OF SUNST (HOURS)
08540  C
08550       SS     = TSS*3.81972 + 12. +DTSL -ET
08560  C
08570  C COMPUTE HOUR ANGLE OF SUNRISE (RADIANS)
08580  C
08590       TSR    = 6.283185 - TSS
08600  C
08610  C    COMPUTE  STANDARD TIME OF SUNRISE (HOUR)
08620  C
08630       SR     = TSR*3.81972 -12. + DTSL -ET
08640  C
08650  C
08660  C    CONVERT SUNRISE IN STANDARD TIME TO LOCAL TIME
```

```
08670   C
08680         SR = SR - DTSL
08690   C
08700   C     CONVERT SUNSET IN STANDARD TIME TO LOCAL TIME
08710   C
08720         SS = SS - DTSL
08730   C
08740   C
08750   C     DEBUG OPTION
08760   C
08770         IF (NBUG.EQ.0) GO TO 300
08780         DO 100 I = 1,NBUG
08790   C
08800         IF (DEBUG(I).NE.IDECL) GO TO 100
08810         WRITE (IWBUG,200)  RJD,DTSL,DELTA,TSS,SS,TSR,SR
08820     200 FORMAT (////.1H ,'SUBROUTINE DECL ','***',' RJD =',
08830        1 I5,'  DTSL =',F6.3,'  DELTA =',F6.3,'  TSS =',F6.3,
08840        2 '  SS =',F6.3,2X,'TSR =',F6.3,2X,'SR = ',F6.3)
08850     100 CONTINUE
08860   C
08870     300 CONTINUE
08880   C
08890         RETURN
08900         END
08910   C
08920   C.....................................................
08930   C
08940         SUBROUTINE LIMITS (DTSL,R,S,T0,RHO,T12,SIGMA,T23)
08950   C     FIND LIMITS FOR TEMPERATURE INTEGRATION
08960   C
08970   C
08980         T0  = - DTSL
08990         T23 = 23.00 - DTSL
09000   C
09010         IF (DTSL.LT.0.0) GO TO 50
09020   C
09030   C     FIND LIMITS OF INTEGRATION WHEN OBSERVER IS
09040   C     WEST OF THE STANDARD MERIDIAN
09050   C
09060   C     FOR SUNRISE
09070   C
09080         RHO = AINT(R+1.0) - DTSL
09090         IF (RHO .LT. R) RHO = RHO + 1
09100   C
09110   C     FOR SUNSET
09120   C
09130         SIGMA = AINT(S+1) - DTSL
09140         IF (SIGMA .LT. S) SIGMA = SIGMA + 1
09150   C
09160   C     FOR LOCAL NOON
09170   C
```

```
09180         T12 = 13.0 - DTSL
09190   C
09200         GO TO 75
09210   C
09220      50 CONTINUE
09230   C
09240   C      FIND LIMITS OF INTEGRATION WHEN OBSERVER
09250   C      IS EAST OF THE STANDARD MERIDIAN
09260   C
09270         RHO = AINT(R) - DTSL
09280         IF (RHO .LT. R) RHO = RHO + 1
09290   C
09300         SIGMA = AINT(S) - DTSL
09310         IF (SIGMA .LT. S) SIGMA = SIGMA + 1
09320   C
09330         T12 = 12.0 - DTSL
09340   C
09350      75 CONTINUE
09360   C
09370         RETURN
09380         END
09390   C
09400   C.....................................................
09410   C
09420   C
09430         SUBROUTINE TMPSIM (TPRIME, KBAROB, CLOUD, GRTEMP,WSPEED,WDIR,
09440        $                     BCOEF,THAT)
09450   C
09460   C
09470   C        TMPSIM DETERMINES THE PROJECTED TEMPERATURES
09480   C           FOR THE NEXT DAY
09490   C
09500   C
09510         DIMENSION TPRIME(1), THAT(1),  BCOEF(8)
09520         DIMENSION KBAROB(1), CLOUD(1), GRTEMP(1)
09530         DIMENSION WSPEED(1), WDIR(1)
09540         INTEGER  IDBUG
09550         REAL KBAR, KBAROB
09560         REAL I0, I1, I2, I3, I4, I5, I6, I7
09570   C
09580         COMMON /SUN/   DELTA, DTSL, SR, SS
09590         COMMON /ORBIT/ PHI, THETAS, THETAL, EP, ET, W
09600         COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
09610        $               ,JSTART, JSTOP,  JREND,  JYEAR
09620         COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
09630         COMMON /IO/    IREAD, IWRITE, IWBUG
09640         COMMON /INTEG/ I0,I1,I2,I3,I4,I5,I6,I7
09650   C
09660   C
09670   C        SET DEBUG FLAG
09680   C
```

```
09690                IDBUG = 1
09700   C
09710   C
09720            TPR = TPRIME(1)
09730   C
09740         CALL TEMPK (DELTA, PHI, BCOEF, TPR,
09750        $              C0, C1, C2, C3, C4, C5, C6)
09760   C
09770   C
09780            IF ( IDBUG .EQ. 0 ) GO TO 910
09790            WRITE(IWBUG,909)TPRIME(25),C0,C1,C2,C3,C4,C5,C6
09800     909    FORMAT ('  TMPSIM  TPRIME(25)=',F10.5,2X,'C= ',
09810        $              6(F8.4,1X))
09820     910    CONTINUE
09830   C
09840   C        BEGIN TEMPERATURE PROJECTION LOOP.
09850   C
09860   C        INITIALIZE THE STANDARD TIME COUNTER
09870   C
09880         ST = 0.0
09890   C
09900   C
09910   C        INITIALIZE VARIABLES FOR THE DAILY TEMPERATURE
09920   C        PROJECTION SUBROUTINE TEMPSN
09930   C
09940   C
09950               I0 = 0.0
09960               I1 = 0.0
09970               I2 = 0.0
09980               I3 = 0.0
09990               I4 = 0.0
10000               I5 = 0.0
10010               I6 = 0.0
10020               I7 = 0.0
10030   C
10040   C
10050   C
10060               TMPLAG = TPR
10070   C
10080         DO 100 I = 1,24
10090   C
10100   C
10110            CLD = CLOUD(I+1)
10120            KBAR = KBAROB(I+1)
10130            TGD = GRTEMP(I+1)
10140            WSP = WSPEED(I+1)
10150            WDR = WDIR(I+1)
10160   C
10170   C
10180            CALL TEMPSN (ST,DTSL,SR,SS,BCOEF,
10190        $              C0,C1,C2,C3,C4,C5,C6,CLD,KBAR,GTD,
```

```
10200       $              WSP,WDR,TMPLAG,THT,T)
10210   C
10220           THAT(I) = THT
10230           TMPLAG  = THT
10240   C
10250   C
10260   C          INCREMENT THE STANDARD TIME COUNTER
10270   C
10280           ST = ST + 1.0
10290   C
10300     100 CONTINUE
10310   C
10320   C
10330         RETURN
10340         END
10350   C
10360   C.....................................................................
10370   C
10380         SUBROUTINE TEMPK (DELTA,PHI,B, TPRIME, K0, K1, K2, K3, K4, K5, K6)
10390   C
10400   C     SUBROUTINE TEMPK COMPUTES THE COEFFICIENTS
10410   C     FOR THE TEMPERATURE EQUATION
10420   C
10430   C
10440   C
10450   C     DELTA  = DECLINATION OF SUN IN RADIANS
10460   C     PHI    = LATITUDE IN RADIANS
10470   C     B      = VECTOR OF REGRESSION COEFFICIENTS
10480   C     TPRIME = YESTERDAY'S TEMPERATURE AT 11 PM
10490   C     K0-K6  = COEFFICIENTS IN TEMPERATURE EQUATION
10500   C     B0-B6  = EQUIVALENCED VARIABLES WITH B VECTOR ELEMENTS
10510   C     P      = CONSTANT = 2*PI/24
10520   C     B2P2   = INTERMEDIATE VARIABLE USED FREQUENTLY
10530   C
10540   C
10550   C
10560         REAL K0,K1,K2,K3,K4,K5,K6
10570         DIMENSION B(1)
10580   C
10590         B0 = B(1)
10600         B1 = B(2)
10610         B2 = B(3)
10620         B3 = B(4)
10630   C
10640         P      = 3.14159/12.0
10650         B2P2   = B1*B1 + P*P
10660   C
10670         K0     = TPRIME
10680   C
10690         K1     = B0/B1
10700   C
```

```
10710         K2     = B2*SIN(DELTA)*SIN(PHI)/B1
10720   C
10730         K3     = B1*B2*COS(DELTA)*COS(PHI)/B2P2
10740   C
10750         K4     = P*B2*COS(DELTA)*COS(PHI)/B2P2
10760   C
10770         K5     = P*P*B3*COS(DELTA)*COS(PHI)/B2P2
10780   C
10790         K6     = P*B1*B3*COS(DELTA)*COS(PHI)/B2P2
10800   C
10810         RETURN
10820         END
10830   C
10840   C
10850   C.....................................................................
10860   C
10870         SUBROUTINE TEMPSN ( ST, DTSL, R,S,B, K0,K1,K2,K3,K4,K5,K6,
10880        $                    CLD, KBAR, GTO. WSP, WDR, TMPLAG, THAT, T )
10890   C
10900   C
10910         INTEGER IDBUG
10920         REAL*8 DEBUG(20), DTEMPS
10930         REAL K0,K1,K2,K3,K4,K5,K6
10940         REAL I1,I2,I3,I4,I5,I6,I7, KBAR
10950   C
10960         DIMENSION B(1)
10970   C
10980         COMMON /INTEG/ I1,I2,I3,I4,I5,I6,I7
10990         COMMON /IO/    IREAD, IWRITE, IWBUG
11000         COMMON /DBUG/  NBUG, DEBUG
11010         COMMON /SWITCH/ SWICH1, SWICH2
11020   C
11030         DATA DTEMPS / 'TEMPS   ' /
11040   C
11050   C
11060   C        STATEMENT FUNCTIONS FOR INTEGRALS I2 AND I3
11070   C
11080   C
11090         FUNC1(A,B) = K2*(EXP(B1*A) - EXP(B1*B))
11100         FUNC2(A)   = EXP(B1*A)*(K3*COS(P*A) + K4*SIN(P*A))
11110         FUNC3(B)   = EXP(B1*B)*(K6*SIN(P*B) - K5*COS(P*B))
11120   C
11130   C
11140   C
11150   C
11160   C        SET DEBUG FLAG
11170   C
11180              IDBUG = 1
11190   C
11200   C
11210   C
```

```
C       SET SWITCHES THAT DETERMINE WHICH PREDICTORES ARE USED
C
          SWICH1 = 1
          SWICH2 = 0
C
C
      B1 = B(2)
      B4 = B(5)
      B5 = B(6)
      B6 = B(7)
      B7 = B(8)
C
C
C8000 WRITE (5,9000) (B(J),J=1,8)
C9000 FORMAT (1H0,'TEMPS B VECTOR',5X, 4(E12.5,5X)/T20,4(E12.5,5X))
C
C
          IF ( SWICH1 .EQ. 0 ) KBAR = 1.000000
C
C
      P = 3.14159/12.00000
C
C
C       CONVERT STANDARD  TIME TO LOCAL TIME
C
      T = TAU(ST)* 12.0/3.14159   -  12.0
C
      IF ( ST .LT. 0.5 .AND. T .LT. 0.0 ) GO TO 5
      IF ( ST .GT.22.5 .AND. T .GT. 24. ) GO TO 5
      IF ( T .LT. 0.0 ) T = T + 24.00
      IF ( T.GT. 24.00 ) T = T - 24.00
    5 CONTINUE
C
C
C     IN ADDITION TO SUNRISE AND SUNSET DETERMINE THE LIMITS
C     OF THE RANGES OF THE TEMPERATURE EQUATIONS
C
      T0 = -DTSL
      T12 = 12.0 - DTSL
      T23 = 23.0 - DTSL
C
      TP = T0 - 1.0
C
      IF(IDBUG.EQ.0) WRITE (IWRITE,10) T0,R,T12,S,T23,TP
   10 FORMAT (1H ,T40,6(2X,E10.3)/)
C
C
C       THE FORM OF INTEGRALS I1, I4, I5, I6, AND I7 ARE THE SAME
C       FOR ALL TIMES OF THE DAY.  I2 AND I3 WILL VARY IN
C       FORM DEPENDING ON THE TIME OF DAY.
C
```

```
C
C        COMPUTE I1, I4, I5, I6, I7
C
C
      I1 = K1*(EXP(B1*T) - EXP(B1*TP))
C
C
         IF ( SWICH1 .EQ. 0 ) GO TO 40
C
C
C
      PP = (1.0-EXP(-B1))*EXP(B1*T)/B1
      QB = 1.579E-8*(1.00+0.17*CLD**2.)*(TMPLAG+460.)**6.
      I4 = B4*QB*PP + I4
C
   40 CONTINUE
C
         IF ( SWICH2 .EQ. 0 ) GO TO 50
C
      I5 = B5*GTO*PP + I5
C
C
   50 CONTINUE
C
      I6 = B6*WSP*PP + I6
      I7 = B7*WDR*PP + I7
C
C
C
C        CALCULATE SUBTOTAL
C
      SUBTOT = I1 + I4 + I5 + I6 + I7
C
C
C
      IF ( T .GT. R ) GO TO 100
C
C
C******** RANGE 1 -- AFTER MIDNIGHT AND BEFORE SUNRISE
C
C
C
      GTT = SUBTOT
C
      GO TO 900
C
C
  100 IF ( T .GT. R + 1.00 ) GO TO 200
C
C
C******** RANGE 2 --FIRST HOUR OR FRACTION AFTER SUNRISE
```

```
12240   C
12250   C
12260   C
12270   C
12280         I2 = FUNC1(T,R) - FUNC2(T) + FUNC2(R)
12290         I2 = I2*KBAR
12300   C
12310         I3 = FUNC3(T) - FUNC3(R)
12320         I3 = KBAR*I3
12330   C
12340         GTT = I2 + I3 + SUBTOT
12350   C
12360         GO TO 900
12370   C
12380   C
12390   C
12400     200 IF( T .GT. TI2 ) GO TO 250
12410   C
12420   C
12430   C******** RANGE 3 -- AFTER SUNRISE AND BEFORE NOON
12440   C
12450            Q1 = FUNC1(T,T-1.0)
12460            Q2 = FUNC2(T)
12470            Q3 = FUNC2(T-1.0)
12480   C
12490         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
12500         I2 = TI2*KBAR + I2
12510   C
12520            IF(IDBUG.EQ.0)WRITE(IWBUG,991)KBAR, I2
12530     991     FORMAT (2X, 'KBAR ', 2(E12.5,3X))
12540   C
12550         TI3 = FUNC3(T) - FUNC3(T-1.0)
12560         I3  = TI3*KBAR + I3
12570   C
12580   C
12590         GTT = I2 + I3 + SUBTOT
12600   C
12610         GO TO 900
12620   C
12630   C
12640   C
12650     250 IF ( T .GT. TI2 + 1.0 ) GO TO 300
12660   C
12670   C
12680   C
12690   C********* RANGE 4 -- FIRST HOUR AFTER LOCAL NOON
12700   C
12710         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
12720         I2  = TI2*KBAR + I2
12730   C
12740         TI3 = FUNC3(12.0) - FUNC3(T-1.0)
```

```
12750         I3  = TI3*KBAR + I3
12760   C
12770         GTT = I2 + I3 + SUBTOT
12780         GO TO 900
12790   C
12800     300 IF ( T .GT. S ) GO TO 400
12810   C
12820   C
12830   C******** RANGE 5 -- AFTER LOCAL NOON BUT BEFORE SUNSET
12840   C
12850   C
12860         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
12870         I2  = TI2*KBAR + I2
12880   C
12890   C
12900         GTT = I2 + I3 + SUBTOT
12910   C
12920         GO TO 900
12930   C
12940   C
12950   C
12960     400 IF (T .GT. S+1.0 ) GO TO 500
12970   C
12980   C
12990   C******** RANGE 6 -- FIRST HOUR AFTER SUNSET
13000   C
13010   C
13020         TI2 = FUNC1(S,T-1.0) - FUNC2(S) + FUNC2(T-1.0)
13030         I2  = TI2*KBAR + I2
13040   C
13050   C
13060         GTT = I2 + I3 + SUBTOT
13070   C
13080         GO TO 900
13090   C
13100   C
13110   C
13120     500 CONTINUE
13130   C
13140   C
13150   C
13160   C******** RANGE 7 -- AFTER SUNSET
13170   C
13180   C
13190   C        NOTE: I2 DOES NOT CHANGE WHEN T .GT. S
13200   C              I3 DOES NOT CHANGE WHEN T .GT. 12.0
13210   C        THUS USE THE PREVIOUSLY COMPUTED VALUES FOR I2 AND I3
13220   C        THAT HAVE BEEN STORED IN THE COMMON /INTEG/
13230   C
13240   C
13250         GTT = I2 + I3 + SUBTOT
```

```
C
C
C
  900 CONTINUE
C
C
C        NOW THAT THE FUNCTION GTT HAS BEEN EVALUATED,
C        COMPUTE THE TEMPERATURE AT TIME T.
C
C
         IF(IDBUG.EQ.0)WRITE(IWBUG,990) GTT,I1,I2,I3,I4,I5,I6,I7
  990    FORMAT ( 2X, 'GTT ',8(E12.5,3X))
C
C
C
      THAT = K0*EXP(-B1*(T-TP)) + GTT*EXP(-B1*T)
C
C
C8010 WRITE (5,9010) THAT,K0,B1,T,TP,GTT
 9010 FORMAT (1H0, 'TEMPSN', 6(E12.5,5X))
C
C
C
C*************************************************************************
      RETURN
 5990 CONTINUE
C*************************************************************************
C
C
C        DEBUG INFORMATION FOR TEMPSN
C
C
      WRITE (5,6000)
 6000 FORMAT (1H1/2(1H+,100(1H )/),1H+,25(4H****))
C
      WRITE (5,6005)
 6005 FORMAT (1H ,10(4H****),T44,'DEBUG TEMPSN',T61,10(4H****)//)
C
      WRITE (5,6010) T, TP, R, S
 6010 FORMAT (1H ,'TIME PARAMETERS ',7X,1HT,12X,2HTP,13X,1HR,
     $           13X,1HS/15X,4(3X,F11.4))
C
      WRITE (5,6020)
 6020 FORMAT (//1H ,7X,'K0',12X,'K1',12X,'K2',12X,'K3',12X,
     $         'K4',12X,'K5',12X,'K6')
C
      WRITE (5,6030)K0,K1,K2,K3,K4,K5,K6
 6030 FORMAT (1H ,8(3X,E11.4))
C
      WRITE (5,6040)
 6040 FORMAT (//,1H ,7X,2HB0,12X,2HB1,12X,2HB2,12X,2HB3,12X,2HB4,
```

```
13770       $         12X,2HB5,12X,2HB6,12X,2HB7)
13780   C
13790         WRITE (5,6030) (B(M),M=1,8)
13800   C
13810         WRITE (5,6035)
13820    6035 FORMAT (//1H ,6X,3HCLD,10X,6HTMPLAG,9X,3HWSP,12X,3HWDR)
13830         WRITE (5,6030) CLD, TMPLAG, WSP, WDR
13840   C
13850         WRITE (5,6050)
13860    6050 FORMAT (//1H ,6X,3HGTT,12X,2HI1,12X,2HI2,12X,2HI3,12X,2HI4,
13870       $           12X,2HI5,12X,2HI6,12X,2HI7)
13880   C
13890         WRITE (5,6030) GTT,I1,I2,I3,I4,I5,I6,I7
13900         WRITE (5,6060)
13910    6060 FORMAT (//1H ,5X,4HTIME,11X,4HTEMP)
13920   C
13930         WRITE (5,6030) T,THAT
13940   C
13950         WRITE (5,6070)
13960    6070 FORMAT (//1H ,25(4H****))
13970   C
13980   C
13990   C
14000   C*************************************************************************
14010   C*************************************************************************
14020   C
14030   C
14040         RETURN
14050         END
14060   C
14070   C
14080   C.........................................................
14090   C
14100   C
14110   C
14120   C.........................................................
14130   C
14140         FUNCTION ARVA (IDUMMY)
14150   C
14160   C     FUNCTION ARVA SELECTS A RANDOM VARIABLE FROM AN
14170   C     ARBITRARY DISCRETE PROBABILITY MASS FUNCTION
14180   C
14190   C       IN TEMPERATURE GENERATION, THE RANDOM VARIATE IS TO BE
14200   C       DRAWN FROM THE DISTRIBUTION N(0,1). THUS, PDF AND COORD
14210   C       ARE DEFINED BY DATA STATEMENTS WITH VALUES DERIVED FROM
14220   C       A DISTRIBUTION N(0,1).
14230   C
14240   C
14250   C     PDF    = DISCRETE PROBABILITY DENSITY FUNCTION
14260   C     N      = NUMBER OF INTERVALS
14270   C     A      = LOWER LIMIT OF U(A,B)
```

```
C     B      = UPPER LIMIT OF U(A,B)
C     ISEED  = SEED FOR RANDU
C     COORD  = CONTAINS COORDINATES OF THE INTERVALS
C              OF PDF. (COORD(I-1).LT.X.AND.X.LE.COORD(I))
      REAL PEAK, PDF(30), COORD(30)
C
      COMMON /SEED/ ISEED
      COMMON /DBUG/ NBUG,DEBUG
C
      REAL*8    IARVA,DEBUG(20)
C
      DATA IARVA /'ARVA'/
      DATA PDF/0.00016,0.00183,0.00815,0.0161,0.0218,0.0254,
     $  0.0278,0.0301,0.0323,0.0342,0.0361,0.0375,0.0386,0.0395,
     $ 0.0398,0.0398,0.0395,0.0386,0.0375,0.0361,0.0342,0.0323,0.0301,
     $ 0.0278,0.0254,0.0218,0.0161,0.00815,0.00183,0.00016/
      DATA COORD/-3.0,-2.1,-1.5,-1.2,-1.0,-0.9,-0.8,-0.7,-0.6,-0.5,
     $  -0.4,-0.3,-0.2,-0.1,0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,
     $ 1.0,1.2,1.5,2.1,3.0,3.9/
      DATA A/-3.9/, B/+3.9/, N/30/, ISEAS/1/
C
C
C     FIND THE PEAK OF THE DISTRIBUTION
C
      PEAK = 0.0
      DO 100 I = 1,N
      IF (PDF((ISEAS-1)*N +I).GT.PEAK) PEAK = PDF((ISEAS-1)*N +I)
  100 CONTINUE
C
C     SELECT THE FIRST RANDOM NUMBER FROM U(A,B)
C
  150 IX = ISEED
CIBM  CALL RANDU (IX,ISEED,R)
      CALL RAND1 (IX,ISEED,R)
      U1 = A + (B-A) * R
C
C     FIND WHICH INTERVAL U1 BELONGS TO
C
      DO 200 I = 1,N
      IF (U1 .GT. COORD((ISEAS-1)*N + I )) GO TO 200
      J= (ISEAS-1)*N +I
      GO TO 300
  200 CONTINUE
C
C
      WRITE (IWRITE,250)
  250 FORMAT (1H1, ' SUBROUTINE ARVA -- U1 IS GREATER ',
     1 'THAN THE MAXIMUM INTERVAL FOR THE DISCRETE PDF')
C
C
      STOP
```

```
  300 CONTINUE
C
C     CALCULATE THE SELECTION CRITERION
C
      F= PDF(J)/PEAK
      IX = ISEED
CIBM  CALL RANDU (IX,ISEED,U2)
      CALL RAND1 (IX,ISEED,U2)
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 600
C
      DO 500 I = 1,NBUG
      IF (DEBUG(I).NE.IARVA) GO TO 500
C
C
      WRITE (IWBUG,550)PEAK,U1,U2,F
  550 FORMAT (///1H ,'FUNCTION ARVA',2X,'PEAK =',F5.3,
     1 'U1 =',F5.3,2X,'U2 =',F5.3,2X,'F =',F5.3)
C
C
  500 CONTINUE
C
  600 CONTINUE
C
C     ACCEPT OR REJECT U1
C
      IF (U2.GT.F) GO TO 150
      ARVA=U1
      RETURN
      END
C
C.....................................................................
C
C
C
C.....................................................................
C
      SUBROUTINE RANDU (IX,IY,YFL)
C
C     GENERATES A UNIFORM DISTRIBUTION
C
      IY = IX*65539
      IF (IY) 10,20,20
   10 IY = IY + 2147483647 + 1
   20 YFL = IY
      YFL = YFL*.4656613E-9
      RETURN
C
C
```

```
15300         ENTRY RAND1 (IX,IY,YFL)
15310   C
15320            YFL = RAND(0)
15330   C
15340         RETURN
15350         END
15360   C
15370   C.....................................................................
15380   C
15390   C
15400   C.....................................................................
15410   C
15420         SUBROUTINE START (ISEED)
15430   C
15440   C
15450   C        ACTIVATE THE RANDOM NUMBER GENERATOR ISEED TIMES
15460   C        IN ORDER TO GET A DIFFERENT SEQUENCE OF
15470   C        NUMBERS WITH EACH RUN OF THE PROGRAM
15480   C
15490   C
15500         DO 100 I = 1,ISEED
15510   C
15520            YFL = RAND(0)
15530   C
15540     100 CONTINUE
15550   C
15560         RETURN
15570         END
```

Chapter 4

CONSTRAINED STOCHASTIC CLIMATE SIMULATION

4.1 Introduction

Once all of the necessary parameters have been estimated, PROGRAM CSCS can be used to generate hourly values of precipitation, cloud cover, shortwave radiation, longwave radiation, temperature, dewpoint, wind speed, and wind direction. Run-time options are entered through an interactive terminal and parameter values are read from a separate data file. The input data file structure is documented in the beginning section of the PROGRAM CSCS Fortran listing. PROGRAM CSCS is extensively documented throughout and its theoretical development is described in Curtis and Eagleson (1982).

4.2 PROGRAM CSCS

To begin execution, the user must respond to several run-time prompts.

1. Computer: THE CURRENT DATA FILE IS DCCMOD.DAT. DO YOU WISH TO READ A DIFFERENT ONE? (Y/N)

   User: Answer Y if the current parameter information is stored in file DCCMOD.DAT.

   Answer N if a different one is to be used.

2. If the answer to 1 above is Y

   Computer: WHAT DATA FILE CONTAINS THE INPUT DATA? ENTER FILE NAME IN THE FORM XXXXXX.YYY

   User: Enter the desired file name.

3. Computer: DO YOU WANT TO PRINT THE INPUT SUMMARY TO THE CONSOLE? (Y/N)

   User: Answer Y and the input data summary will be directed to the interactive console.

   Answer N will direct the input summary to the file specified on Card 5 in the input data file.

4. Computer: INPUT SEED FOR RANDOM NUMBER GENERATOR

   User: Enter any positive integer value

5. Computer: DO YOU WANT A PLOT? (Y/N)

   User: Enter a Y if you want to plot the generated data.
   Enter N otherwise.

6. If the answer to 5 above is Y:

   Computer: WHICH PLOT? 1 for six-variable
   2 for four-variable

   User: Enter 1 for plot of temperature, dewpoint, longwave radiation, shortwave radiation, cloud cover and precipitation.

   Enter 2 for plot of longwave radiation, shortwave radiation, cloud cover, and precipitation.

7. Computer: CREATE LAND SURFACE MODEL DATA FILE? (Y/N)

   User: Enter Y or Y

   Note: See Curtis and Eagleson (1982), Chapter 12, and Milly and Eagleson (1982).

8. If answer to 7 above is Y:

   Computer: ENTER DATA FILE NAME...XXXXXX.YYY

   User: Enter the output file name

9. If the answer to 7 above is Y

   Computer: ENTER COMMENTS TO IDENTIFY OUTPUT DATA (80 CHAR.MAX)

   User: Enter one line of identifying comments for the output file.

4.3 Fortran Listing for PROGRAM CSCS

```
00010  C=====================================================================
00020  C=====================================================================
00030  C
00040  C  CARD INPUT SUMMARY FOR THE CSCS MODEL
00050  C
00060  C
00070  C  CARD  COLUMN   FORMAT   DESCRIPTION
00080  C  ----  ------   ------   -----------
00090  C
00100  C     1    1-80     --      USER INFORMATION CARD USED ONLY TO SEPARATE
00110  C                           CARD GROUPS IN THE DECK (OR FILE)
00120  C
00130  C   2-4    1-80    15A5     THREE TITLE CARDS. THE TEXT ON THESE CARDS
00140  C                           WILL BE PRINTED OUT AT THE BEGINNING OF
00150  C                           THE INPUT DATA SUMMARY.
00160  C
00170  C     5    1-10     10X     SPACE FOR CARD LABEL. NOT READ BY CSCS.
00180  C          11-20     A10     OUTPUT FILE NAME FOR INPUT DATA SUMMARY AND
00190  C                           OUTPUT DATA ANALYSIS. FILE NAME HAS THE
00200  C                           FORM  XXXXXX.YYY
00210  C          21-30     A10     OUTPUT FILE NAME FOR GENERATED DEWPOINT
00220  C                           DEPRESSION.  XXXXXX.YYY
00230  C          31-40     A10     OUTPUT FILE NAME FOR DEBUG INFORMATION
00240  C                           XXXXXX.YYY
00250  C
00260  C     6    1-10     10X     CARD LABEL
00270  C          11-12      I2     INITIAL MONTH - MM
00280  C          14-15      I2     INITIAL DAY   - DD
00290  C          17-20      I4     INITIAL YEAR  - YYYY
00300  C          22-23      I2     ENDING MONTH  - MM
00310  C          25-26      I2     ENDING DAY    - DD
00320  C          28-31      I4     ENDING YEAR   - YYYY
00330  C          33-35    F3.0     LATITUDE      - DEGREES
00340  C          37-38    F2.0     LATITUDE      - MINUTES
00350  C          40-41    F2.0     LATITUDE      - SECONDS
00360  C          43-45    F3.0     LONGITUDE     - DEGREES
00370  C          47-48    F2.0     LONGITUDE     - MINUTES
00380  C          50-51    F2.0     LONGITUDE     - SECONDS
00390  C          53-62     A10     TIME ZONE (U.S. 'EASTERN', 'CENTRAL',
00400  C                           'MOUNTAIN', OR 'PACIFIC')  LEFT JUSTIFY.
00410  C
00420  C     7    1-80     --      USER INFORMATION CARD
00430  C
00440  C     8    1-10     10X     CARD LABEL
00450  C          11-20   F10.0     MEAN FAIRWEATHER CLOUD COVER
00460  C          21-30   F10.0     ST. DEV. OF FAIRWEATHER CLOUD COVER
00470  C          31-40   F10.0     LAG-1 CORRELATION COEF. OF FAIRWEATHER
00480  C                           CLOUD COVER.
00490  C          41-50   F10.0     CLOUD COVER TRANSITION DECAY COEFFICIENT
00500  C          51-60   F10.0     ATMOSPHERIC TURBIDITY FACTOR
```

```
00510  C
00520  C     9    1-10      10X    CARD LABEL
00530  C          11-12      I2    NUMBER OF FAIRWEATHER CLOUD COVER HISTOGRAM
00540  C                            ELEMENTS
00550  C          21-30    F10.0   LOWER BOUND OF FAIRWEATHER CLOUD COVER
00560  C                            HISTOGRAM
00570  C          31-40    F10.0   UPPER BOUND OF FAIRWEATHER CLOUD COVER
00580  C                            HISTOGRAM
00590  C
00600  C    10    1-10      10X    CARD LABEL
00610  C          11-70   6F10.0   HISTOGRAM ELEMENTS. USE AS MANY CARDS AS
00620  C                            NEEDED. REMEMBER THAT THE FIRST 10 SPACES
00630  C                            ON EACH CARD ARE RESERVED FOR THE CARD LABEL
00640  C
00650  C    11    1-10      10X    CARD LABEL
00660  C          11-70   6F10.0   RIGHT HAND COORDINATE OF EACH HISTOGRAM
00670  C                            ELEMENT FROM LOWEST TO HIGHEST.
00680  C
00690  C    12    1-80      --     USER INFORMATION CARD
00700  C
00710  C    13    1-10      10X    CARD LABEL
00720  C          11-20    F10.0   MEAN TIME BETWEEN STORMS
00730  C          21-30    F10.0   MEAN STORM DURATION
00740  C          31-40    F10.0   MEAN STORM DEPTH
00750  C
00760  C    14    1-80      --     USER INFORMATION CARD
00770  C
00780  C    15    1-10      10X    CARD LABEL
00790  C          11-58   4E12.5   REGRESSION COEFFICIENTS FOR THE DETER-
00800  C                            MINISTIC COMPONENT OF THE TEMPERATURE
00810  C                            MODEL (B0-B7). USE TWO CARDS.
00820  C
00830  C    16    1-10      10X    CARD LABEL
00840  C          11-20    F10.0   TEMPERATURE BIAS FOR THE STOCHASTIC
00850  C                            COMPONENT OF TEMPERATURE
00860  C          21-30    F10.0   ST. DEVIATION FOR THE STOCHASTIC COMPONENT
00870  C                            OF TEMPERATURE
00880  C          31-40    F10.0   LAG-1 SERIAL CORRELATION COEFFICIENT FOR
00890  C                            THE STOCHASTIC COMPONENT OF TEMPERATURE
00900  C
00910  C    17    1-80      --     USER INFORMATION CARD
00920  C
00930  C    18    1-10      10X    CARD LABEL
00940  C          11-20    F10.0   MINIMUM HOURLY WIND SPEED
00950  C          21-30    F10.0   TIME OF MINIMUM HOURLY WIND SPEED
00960  C          31-40    F10.0   MAXIMUM HOURLY WIND SPEED
00970  C          41-50    F10.0   TIME OF MAXIMUM HOURLY WIND SPEED
00980  C
00990  C    19    1-10      10X    CARD LABEL
01000  C          11-20    F10.0   MINIMUM HOURLY ST. DEVIATION OF WIND SPEED
01010  C          21-30    F10.0   TIME OF MINIMUM HOURLY ST. DEVIATION OF
```

```
C                                    WIND SPEED
C          31-40    F10.0    MAXIMUM HOURLY ST. DEVIATION OF WIND SPEED
C          41-50    F10.0    TIME OF MAXIMUM HOURLY ST. DEVIATION OF
C                             WIND SPEED
C
C     20    1-10      10X    CARD LABEL
C          11-20    F10.0    WIND SPEED SKEW COEFFICIENT
C          21-30    F10.0    LAG-1 SERIAL CORRELATION COEFFICIENT OF
C                             WIND SPEED
C
C     21    1-80       --    USER INFORMATION
C
C     22    1-10      10X    CARD LABEL
C          11-20    F10.0    MEAN TRANSFORMED WIND DIRECTION
C          21-30    F10.0    ST. DEVIATION OF TRANSFORMED WIND DIRECTION
C          31-40    F10.0    LAG-1 SERIAL CORRELATION COEFFICIENT OF
C                             TRANSFORMED WIND DIRECTION
C
C     23    1-10    F10.0    CARD LABEL
C          11-12       I2    NUMBER OF ELEMENTS IN TRANSFORMED WIND
C                             DIRECTION HISTOGRAM
C          21-30    F10.0    LOWER BOUND OF HISTOGRAM (USUALLY 0.00)
C          31-40    F10.0    UPPER BOUND OF HISTOGRAM (USUALLY 180.)
C
C     24    1-10      10X    CARD LABEL
C          11-70   6F10.0    HISTOGRAM ELEMENTS. USE AS MANY CARDS AS
C                             NEEDED
C
C     25    1-10      10X    CARD LABEL
C          11-70   6F10.0    RIGHT HAND COORDINATE OF EACH HISTOGRAM
C                             ELEMENT, LOWEST TO HIGHEST
C
C     26    1-80       --    USER INFORMATION CARD
C
C     27    1- 5       A5    DEWPOINT MODEL TYPE
C                             'REGRS' = REGRESSION MODEL
C                             'INDEP' = INDEPENDENT MODEL
C
C     *** FOR INDEPENDENT DEWPOINT GENERATION ONLY ***
C
C     28    1-10      10X    CARD LABEL
C          11-20    F10.0    MEAN DEWPOINT TEMPERATURE
C          21-30    F10.0    ST. DEVIATION OF DEWPOINT TEMPERATURE
C          31-40    F10.0    LAG-1 SERIAL CORRELATION COEFFICIENT OF
C                             DEWPOINT TEMPERATURE
C
C     *** FOR REGRESSION DEWPOINT GENERATION ONLY  ***
C
C     29    1-10      10X    CARD LABEL
C          11-58   4E12.5    REGRESSION COEFFICIENTS FOR THE DETER-
C                             MINISTIC COMPONENT OF DEWPOINTS (D0-D5)
```

```
C
C
C     30   1-10      10X    CARD LABEL
C          11-20    F10.0   BIAS OF STOCHASTIC COMPONENT OF DEWPOINTS
C          21-30    F10.0   ST. DEVIATION OF STOCHASTIC COMPONENT OF
C                           DEWPOINTS
C          31-40    F10.0   LAG-1 SERIAL CORRELATION COEFFICIENT OF
C                           STOCHASTIC COMPONENT OF DEWPOINTS
C
C
C=======================================================================
C=======================================================================
C
C
C.....................................................................
C
      PROGRAM CSCS
C
C
C  CONSTRAINED STOCHASTIC CLIMATE SIMULATION
C
C  PROGRAMMER: DAVID C. CURTIS
C              NORTHEAST RIVER FORECAST CENTER
C              705 BLOOMFIELD AVENUE
C              BLOOMFIELD, CT 06002-2478
C
C              TELEPHONE: (203) 722-2014
C
C  THE CSCS MODEL GENERATES HOURLY VALUES OF PRECIPITATION, CLOUD
C  COVER, SHORTWAVE RADIATION, LONGWAVE RADIATION, TEMPERATURE,
C  DEWPOINT, WIND SPEED AND WIND DIRECTION.  THE PROGRAM CODE IS
C  FORTRAN AND HAS BEEN DEVELOPED ON A DEC-10 TIME-SHARE COMPUTER
C  SYSTEM.  STANDARD FORTRAN CODE WAS USED AS MUCH AS POSSIBLE TO
C  AVOID TOO MANY PROBLEMS WHEN TRANSFERRING THIS CODE TO OTHER
C  MACHINES.  HOWEVER SOME MACHINE DEPENDENT CODE IS INEVITABLE,
C  SUCH AS:
C
C           -- 'OPEN' STATEMENTS FOR DATA FILE ACCESS
C           -- 5 CHARACTER WORDS FOR ALPHANUMERIC DATA MANIPULATION
C           -- INPUT/OUTPUT UNIT NUMBERS
C           -- RANDOM NUMBER GENERATION (SEE SUBROUTINE RANDU)
C
C  DATA INPUT AND INTERNAL COMPUTATIONS HAVE BEEN CARRIED OUT IN
C  ENGLISH UNITS.  DATA OUTPUT CAN BE IN ENGLISH OR METRIC UNITS. (SEE
C  THE METRIC CONVERSION SECTION IN THE MAIN PROGRAM) THE PLOT SUB-
C  ROUTINE IS SCALED FOR METRIC OUTPUT.
C
C  THE PROGRAM IS CURRENTLY SET UP FOR GENERATING ANY NUMBER OF SETS
C  OF DATA FOR A PARTICULAR MONTH.  IN OTHER WORDS, 20 JULYS, 30 APRILS,
C  15 JANUARYS ETC. CAN BE GENERATED.  IF THE INPUT PARAMETERS
C  REPRESENT OTHER PERIODS SUCH AS BIMONTHLY, SEASONALLY, ETC., THE
```

```
C  DATE COUNTERS MUST BE ADJUSTED ACCORDINGLY (SEE SUBROUTINE DATE1)
C  JULIAN DATES ARE USED INTERNALLY.  THE PROGRAM HAS BEEN FULLY
C  TESTED FOR JANUARY AND JULY ONLY.
C
C  TO ALL USERS:  GOOD LUCK!!!
C
      DIMENSION TITLE(16,3), BCOEF(8), ACOEF(8)
      DIMENSION CCPDF(30), CCORD(30), TTPDF(30), TTORD(30)
      DIMENSION DRPDF(30),DRORD(30)
      DIMENSION DWPDF(30), DWORD(30)
      DIMENSION ZERO(10), SPB(24),SPSD(24)
      DIMENSION RAWSUM(5), XXT(5,5), MEAN(5)
      DIMENSION COVMAT(5,5), CORMAT(5,5)
      DIMENSION TCDATA(24),DWDATA(24),CLDATA(24),WSDATA(24),WDDATA(24)
      DIMENSION TCTITL(5) ,DWTITL(5) ,CLTITL(5) ,WSTITL(5) ,WDTITL(5)
      DIMENSION TCRHO(24) ,DPRHO(24) ,CLRHO(24) ,WSRHO(24) ,WDRHO(24)
      DIMENSION TCHIST(50),DPHIST(50),CLHIST(11),WSHIST(40),WDHIST(9)
      DIMENSION ASWRS(24),ACLDS(24),ALWRS(24),ATMPS(24),ADEWS(24)
      DIMENSION AWSPS(24),AWDRS(24),ASWRB(24),ACLDB(24),ALWRB(24)
      DIMENSION ATMPB(24),ADEWB(24),AWSPB(24),AWDRB(24)
      DIMENSION ASWRSQ(24),ACLDSQ(24),ALWRSQ(24),ATMPSQ(24),ADEWSQ(24)
      DIMENSION ASWRSD(24),ACLDSD(24),ALWRSD(24),ATMPSD(24),ADEWSD(24)
      DIMENSION AWSPSQ(24),AWDRSQ(24)
      DIMENSION AWSPSD(24),AWDRSD(24)
      DIMENSION DEF(24), PTEXT(16)
C
C
      DOUBLE PRECISION  WRITEF, BUGOFF, OUTPUT, TZONE, TZ(4), DEBUG(7)
      DOUBLE PRECISION  DAFILE, RADTYP, PNFILE
C
C
      REAL KBAR, LW, MEAN
      REAL I0, I1, I2, I3, I4, I5, I6, I7
      REAL LAT(3), LONG(3)
C
C
      INTEGER TCHIST, DPHIST, CLHIST, WSHIST, WDHIST
      INTEGER TCHDIM, DPHDIM, CLHDIM, WSHDIM, WDHDIM
C
C
      EQUIVALENCE (ZERO(1),I0),(ZERO(2),I1),(ZERO(3),I2),(ZERO(4),I3),
     $           (ZERO(5),I4),(ZERO(6),I5),(ZERO(7),I6),(ZERO(8),I7)
C
      COMMON /TITLES/ TITLE
      COMMON /FILES/  WRITEF, OUTPUT, BUGOFF
      COMMON /DATES/  IYR,IMO,IDAY, LYR,LMO,LDAY
      COMMON /LOCATE/ LAT, LONG, TZONE
      COMMON /DBUG/   NBUG,DEBUG
      COMMON /CLOUDS/ CCBAR,CCSD,CCRHO,BETA,GAM
      COMMON /ATMOS/  EN
      COMMON /PDFCLD/ NUMCC,CCPDF,CCORD,CCA,CCB
```

```
      COMMON /RAINS/  TBBAR, TRBAR, DBAR
      COMMON /TEMPAR/ TDBIAS, TDSDEV, TDRHO, BCOEF, TEMBAR
      COMMON /PDFTEM/ NUMTT, TTPDF, TTORD, TTA, TTB
      COMMON /WINDSP/ SPBAR1,SPBAR2,SPBT1,SPBT2,SPSDV1,SPSDV2,
     $                SPSDT1,SPSDT2,SPSKEW,SPRHO
      COMMON /WINDIR/ DRBAR,DRDEV,DRRHO
      COMMON /PDFDIR/ NUMDR,DRPDF,DRORD,DRA,DRB
      COMMON /DEWONE/ TYPE, ACOEF
      COMMON /DEWTWO/ DWBAR,DWSDEV,DWSKEW,DWRHO
      COMMON /DEWDVS/ DWBIAS, DWDEV, DWDRHO
      COMMON /ORBIT/  PHI,THETAS,THETAL,EP,ET,W
      COMMON /SUN/    DELTA, DTSL, SR,SS
      COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
     $              ,JSTART, JSTOP,  JREND,  JYEAR
      COMMON /IO/     IN,IS,IB
      COMMON /RAINI/  ITR,ITB
      COMMON /SEED/   ISEED
      COMMON /CLDCOV/ CCLAG1
      COMMON /SEAS/   NSEAS
      COMMON /RTYPE/  RADTYP
      COMMON /STORMS/ STORM
      COMMON /INTEG/  I0,I1,I2,I3,I4,I5,I6,I7
      COMMON /LINES/  NLINES
      COMMON /VAPORP/ VP
      COMMON /PUNCHD/ PTEXT,PNFILE,IPUNCH
C
      DATA TZ/8HEASTERN  ,8HCENTRAL  ,8HMOUNTAIN  ,8HPACIFIC  /
      DATA TCTITL /5HHOURL, 5HY TEM, 5HPERAT, 5HURE  , 5H      /
      DATA DWTITL /5HHOURL, 5HY DEW, 5HPOINT, 5H     , 5H      /
      DATA CLTITL /5HHOURL, 5HY CLO, 5HUD CO, 5HVER  , 5H      /
      DATA WSTITL /5HHOURL, 5HY WIN, 5HD SPE, 5HED   , 5H      /
      DATA WDTITL /5HHOURL, 5HY WIN, 5HD DIR, 5HECTIO, 5HN     /
      DATA ON /2HON/, OFF /3HOFF /
C
C
C  OUTPUT VARIABLE DEFINITION
C
C  VARIABLE     DIMENSION       DESCRIPTION
C  --------     ---------       -----------
C    SWR          LY/HR         SHORTWAVE RADIATION
C    LW           LY/HR         LONGWAVE RADIATION
C    WDIR         DEGREES       WIND DIRECTION
C    CLD          -----         CLOUD COVER
C
C   ******  ENGLISH UNITS  ******
C
C    RAIN         IN/HR         PRECIPITAION
C    WSP          MI/HR         WIND SPEED
C    TEMP         DEG F         TEMPERATURE
C    DEW          DEG F         DEWPOINT
C
```

```
C   ******  METRIC  UNITS  ******
C
C    RAINM        MM/HR         PRECIPITAION
C    WSPM         M/S           WIND SPEED
C    TEMPM        DEG C         TEMPERATURE
C    DEWM         DEG C         DEWPOINT
C
C
C
C
C  CALL INTERACTIVE INPUT SUBROUTINE TO GET UNIT NUMBER AND
C  DATA FILE INFORMATION NEEDED TO BEGIN OPERATION
C
      CALL INTER (DAFILE,IS,DPLOT,IPL)
      CALL START (ISEED)
C
C=====================================================================
C=====================================================================
C
C
C  ESTABLISH THE INPUT DATE FILE UNIT NUMBER AND OPEN FILE FOR READ
C
      IN = 21
      OPEN (UNIT=IN,DEVICE='DSK',ACCESS='SEQIN',FILE=DAFILE)
C
      IF (IPUNCH .LE. 0) GO TO 100
      OPEN (UNIT=IPUNCH,DEVICE='DSK',ACCESS='SEQOUT',FILE=PNFILE)
C
      WRITE (IPUNCH,50) PTEXT
   50 FORMAT (16A5)
C
  100 CONTINUE
C
      IU = 26
C
      OPEN (UNIT=IU,DEVICE='DSK',ACCESS='SEQOUT',FILE=OUTPUT)
C
C
C=====================================================================
C=====================================================================
C
C
C
C  READ INPUT DATA FILE
C
      CALL READF (IN,IS,IB)
C
C
C=====================================================================
C=====================================================================
C
```

```
C
C  CONVERT LATITUDE AND LONGITUDE DEGREES:MINUTES:SECONDS TO
C  THEIR DECIMAL EQUIVALENTS.
C
      PHI = DMS(LAT)
      THETAL = DMS(LONG)
C
C  CHECK IF VALID TIME ZONE HAS BEEN REQUESTED
C
      IF (TZONE.NE.TZ(1)) GO TO 200
      THETAS = 75.0*2.0*3.14159/360.0
      GO TO 300
C
  200 IF (TZONE.NE.TZ(2)) GO TO 210
      THETAS =  90.0*2.0*3.14159/360.
      GO TO 300
C
  210 IF (TZONE.NE.TZ(3)) GO TO 220
      THETAS = 105.0*2.0*3.14159/360.0
      GO TO 300
C
  220 IF (TZONE.NE.TZ(4)) GO TO 230
      THETAS = 120.0*2.0*3.14159/360.0
      GO TO 300
C
  230 WRITE (IS,240)
  240 FORMAT (1H1,'TIME ZONE REQUESTED IS NOT VALID'///)
      WRITE (IS,250) TZONE,(TZ(I),I=1,4)
  250 FORMAT (1H0,T10,'REQUESTED TIME ZONE',T36,5H*****,
     1A8,5H*****/T10,'AVAILABLE TIME ZONES',T36,5H*****,
     2A8,5H*****/T36,5H*****,A8,5H*****/T36,5H*****,
     3A8,5H*****/T36,5H*****,A8,5H*****)
C
C
C=====================================================================
C=====================================================================
C
  300 CONTINUE
C
C
C  INITIALIZE DATE COUNTERS
C
      CALL DATE1
C
C  INITIALIZE RAINFALL MODEL PARAMETERS
C
      CALL RAINST (TB,TR,D,JSINCE,STORM,JHREQS,JHNEXT)
C
C
C
C=====================================================================
```

```
04080  C=====================================================================
04090  C
04100  C
04110  C  VARIABLE INITIALIZATION SECTION
04120  C
04130  C    STATISTICAL VARIABLES
04140  C
04150        NDC   = 0
04160        NDATA = 0
04170        NRDATA = 0
04180        TCSUM = 0.0
04190        DWSUM = 0.0
04200        CLSUM = 0.0
04210        WSSUM = 0.0
04220        WDSUM = 0.0
04230        TRSUM = 0.0
04240        TBSUM = 0.0
04250        DHSUM = 0.0
04260        DRY   = 0.0
04270        STORMS = 0.0
04280        TCSMSQ = 0.0
04290        DWSMSQ = 0.0
04300        CLSMSQ = 0.0
04310        WSSMSQ = 0.0
04320        WDSMSQ = 0.0
04330        TTSUM3 = 0.0
04340        DWSUM3 = 0.0
04350        CLSUM3 = 0.0
04360        SPSUM3 = 0.0
04370        WDSUM3 = 0.0
04380  C
04390        DO 375 IA = 1,24
04400          TCRHO(IA) = 0.0
04410          DPRHO(IA) = 0.0
04420          CLRHO(IA) = 0.0
04430          WSRHO(IA) = 0.0
04440          WDRHO(IA) = 0.0
04450          ASWRS(IA)  = 0.0
04460          ASWRSQ(IA) = 0.0
04470          ACLDS(IA)  = 0.0
04480          ACLDSQ(IA) = 0.0
04490          ALWRS(IA)  = 0.0
04500          ALWRSQ(IA) = 0.0
04510          ATMPS(IA)  = 0.0
04520          ATMPSQ(IA) = 0.0
04530          ADEWS(IA)  = 0.0
04540          ADEWSQ(IA) = 0.0
04550          AWSPS(IA)  = 0.0
04560          AWSPSQ(IA) = 0.0
04570          AWDRS(IA)  = 0.0
04580          AWDRSQ(IA) = 0.0
```

```
04590     375 CONTINUE
04600   C
04610   C    HISTOGRAM VARIABLES
04620   C
04630         TCHDIM = 50
04640         DPHDIM = 50
04650         CLHDIM = 11
04660         WSHDIM = 40
04670         WDHDIM = 9
04680         TCDT   = 2.0
04690         DPDT   = 2.0
04700         CLDT   = 0.10
04710         WSDT   = 1.0
04720         WDDT   = 20.0
04730         TCBASE = -30.0
04740         DPBASE = -30.0
04750         CLBASE = -.05
04760         WSBASE = 00.0
04770         SUMSW  = 00.0
04780         SUMLW  = 00.0
04790         WDBASE = 00.0
04800         DO 270 I = 1,TCHDIM
04810     270 TCHIST(I) = 0
04820         DO 271 I = 1,DPHDIM
04830     271 DPHIST(I) = 0
04840         DO 272 I = 1,CLHDIM
04850     272 CLHIST(I) = 0
04860         DO 273 I = 1,WSHDIM
04870     273 WSHIST(I) = 0
04880         DO 274 I = 1,WDHDIM
04890     274 WDHIST(I) = 0
04900   C
04910   C    MISCELLANEOUS VARIABLES
04920   C
04930         TRACE = OFF
04940         NLINES = 0
04950         IDY = 1
04960         MLAG = 24
04970         JHOUR = 0
04980         NSEAS = 1
04990         NMAX = 4
05000         RADTYP = 'CLOUDYSKY'
05010         EP = -1.00
05020         BETA = GAM
05030         ET =  0.00
05040         W  =  2.00
05050   C
05060         IF (TRACE .EQ. ON) WRITE (IS,9000)
05070    9000 FORMAT (' M1')
05080   C
05090   C=====================================================================
```

```
C=====================================================================
C
C  SET UP VARIABLE MEAN AND STANDARD DEVIATION ARRAYS FOR
C   WIND SPEED.
C
      ST = 0.0
C
      DO 330 IV = 1,24
C
      CALL VARYX (SPBAR1,SPBAR2,SPBT1,SPBT2,ST,SPB(IV))
      CALL VARYX (SPSDV1,SPSDV2,SPSDT1,SPSDT2,ST,SPSD(IV))
      ST = ST + 1.0
C
  330 CONTINUE
C
C
C=====================================================================
C=====================================================================
C
C
C  BEGIN CYCLES FOR DATA GENERATION
C
C  THE '400' LOOP REPRESENTS THE DAY CYCLE
C
  400 CONTINUE
C
C
C       UPDATE ORBIT PARAMETERS
C
      CALL DECL (JULREL,DELTA,SR,SS)
C
      DTSL = EP*(THETAS-THETAL)*3.81972
C
C
C=====================================================================
C=====================================================================
C
C  STARTING VALUE SELECTION
C
C
C  FOR THE FIRST TIME PERIOD OF EACH MONTH,
C  GET INITIAL VALUE FOR THE VARIOUS NOISE TERMS BY SELECTING A
C  RANDOM VARIATE FROM THE APPROPRIATE PDF.
C
      IF (IDAY .GT. 1) GO TO 310
      IF (JHOUR.GT. 0) GO TO 310
C
C  GET CLOUD COVER STARTING VALUE
C
      CCLAG1 = ARVA (CCPDF,NUMCC,CCA,CCB,CCORD,NSEAS)
C
```

```
05610         CALL NORMAL (VN)
05620         TTLAG1 = TDSDEV*VN
05630   C
05640   C  GET WIND SPEED STARTING VALUE
05650   C
05660         CALL MARGAM (NUMTT,TTPDF,TTORD,TTA,TTB,SPBAR,SPSDEV,0.0,
05670        $               SPSKEW,0.0,SPLAG1,WNOISE)
05680   C
05690   C  GET WIND DIRECTION STARTING VALUE
05700   C
05710         DRLAG1 = ARVA (DRPDF,NUMDR,DRA,DRB,DRORD,NSEAS)
05720   C
05730   C  GET DEWPOINT STARTING VALUE
05740   C
05750         IF (TYPE .EQ. 'REGRS') GO TO 350
05760         CALL MARGAM (NUMTT,TTPDF,TTORD,TTA,TTB,DWBAR,DWSDEV,0.0,
05770        $              DWSKEW,0.0,DWLAG1,DNOISE)
05780         GO TO 355
05790     350 CONTINUE
05800   C
05810         CALL NORMAL (DWX)
05820         DWLAG1 = 0.85*DWBAR + DWX*DWSDEV
05830   C
05840         CALL NORMAL(DWX)
05850         DWDLAG = DWBIAS + DWX*DWDEV
05860   C
05870     355 CONTINUE
05880   C
05890   C
05900   C  COMPUTE INITIAL TEMPERATURE AND CONSTRAIN DEWPOINT IF NECESSARY
05910   C
05920         TPR = TEMBAR + TTLAG1
05930         IF (DWLAG1 .GE. TPR) DWLAG1 = 0.99*TPR
05940   C
05950         IF (TRACE .EQ. ON) WRITE (IS,9001)
05960    9001 FORMAT (' M2')
05970   C
05980     310 CONTINUE
05990   C
06000   C
06010   C=====================================================================
06020   C=====================================================================
06030   C
06040   C  ESTABLISH THE LAG-1 TEMPERATURES FOR THE TEMPERATURE AND LONG
06050   C  WAVE RADIATION MODELS.
06060   C
06070         TEMP1 = TPR
06080         TMPLAG = TPR
06090   C
06100   C  COMPUTE TODAY'S COEFFICIENTS FOR THE TEMPERATURE MODEL.
06110   C
```

```
06120        CALL TEMPK (DELTA, PHI, BCOEF, TPR,
06130       $             C0, C1, C2, C3, C4, C5, C6)
06140  C
06150  C  INITIALIZE THE INTEGRATION VARIABLES FOR THE TEMPERATURE MODELS.
06160  C    (SEE THE EQUIVALENCE STATEMENT AT THE BEGINNING OF THE PROGRAM)
06170  C
06180        DO 320 K = 1,10
06190        ZERO(K)  = 0.00
06200      320 CONTINUE
06210  C
06220  C
06230  C=====================================================================
06240  C=====================================================================
06250  C
06260  C
06270  C  THE '500' LOOP REPRESENTS THE HOUR BY HOUR DATA GENERATION
06280  C
06290        ST1 = 0.0
06300  C
06310  C
06320        DO 500 I = 0.23
06330  C
06340        IF (TRACE .EQ. ON) WRITE (IS,9002)
06350     9002 FORMAT (' M3')
06360  C
06370        JP = I + 1
06380        ST2 = FLOAT(I)
06390  C
06400  C=====================================================================
06410  C=====================================================================
06420  C
06430  C  RAINFALL SECTION
06440  C
06450        CALL PCPN (TB,TR,D,STORM,JHOUR,JHREOS,JHNEXT,JSINCE,RAIN)
06460  C
06470        TSINCE = FLOAT(JSINCE)
06480  C
06490        IF (TRACE .EQ. ON) WRITE (IS,9003)
06500     9003 FORMAT (' M4')
06510  C
06520  C=====================================================================
06530  C=====================================================================
06540  C
06550  C  SHORTWAVE RADIATION SECTION
06560  C
06570        CALL SOLRAD (JULREL,ST1,ST2,TSINCE,TB,NMAX,CCA,CCB,CCPDF,NUMCC,
06580       $      CCORD,SWR,CLD,BETA,GAM,CCBAR,CCSD,CCRHO,SEASON)
06590  C
06600        IF (TRACE .EQ. ON) WRITE (IS,9004)
06610     9004 FORMAT (' M5')
06620  C
```

```
C=====================================================================
C=====================================================================
C
C  WIND SPEED SECTION
C
      SPBAR = SPB(JP)
      SPSDEV = SPSD(JP)
C
      CALL MARGAM (NUMTT,TTPDF,TTORD,TTA,TTB,SPBAR,SPSDEV,SPRHO,
     $             SPSKEW,SPLAG1,WSP,WNOISE)
C
      IF (WSP .LT. 0.0) WSP = 0.0
      SPLAG1 = WSP
C
      IF (TRACE .EQ. ON) WRITE (IS,9005)
 9005 FORMAT (' M6')
C
C=====================================================================
C=====================================================================
C
C  WIND DIRECTION SECTION
C
      CALL MARKOV (NUMDR,DRPDF,DRORD,DRA,DRB,DRBAR,DRDEV,DRRHO,
     $             DRLAG1,1,WDIR)
  505 CONTINUE
C
  510 IF (WDIR .GT. 180.0) WDIR = 360.0 - WDIR
      IF (WDIR .GT. 180.0) GO TO 510
  520 IF (WDIR .LT.   0.0) WDIR = ABS(WDIR)
      IF (WDIR .LT.   0.0) GO TO 520
      DRLAG1 = WDIR
C
      IF (TRACE .EQ. ON) WRITE (IS,9006)
 9006 FORMAT (' M7')
C
C=====================================================================
C=====================================================================
C
C  TEMPERATURE SECTION
C
C  COMPUTE THE SHORTWAVE RADIATION ATTENUATION DUE TO CLOUD COVER.
C
      KBAR = 1.00 - 0.65*CLD*CLD
C
C  COMPUTE HOURLY TEMPERATURES
C
C
         CALL TEMPSN (ST2,DTSL,SR,SS,BCOEF,
     $               C0,C1,C2,C3,C4,C5,C6,CLD,KBAR,GTO,
     $               WSP,WDIR,TMPLAG,THT,T)
C
```

```
07140   C
07150   C  NOTE THAT TEMP1 AND TMPLAG ARE DIFFERENT VARAIBLES!! TMPLAG
07160   C  DOES NOT HAVE THE DEVIATIONS TERM ADDED IN AND IS USED ONLY
07170   C  IN THE 'REGRESSION' PORTION OF THE TEMPERATURE GENERATION
07180   C  COMPONENT.  TEMP1 IS USED WHEN THE ACTUAL LAG-1 TEMPERATURE
07190   C  IS REQUIRED.
07200   C
07210         TMPLAG = THT
07220         TPR = TMPLAG
07230   C
07240   C  ADD THE RANDOM COMPONENT TO THE TEMPERATURE JUST COMPUTED.
07250   C
07260         CALL NORMAL (ARV)
07270   C
07280         TDEV = TDBIAS + TDRHO*(TTLAG1 - TDBIAS) +
07290        $       ARV*TDSDEV*SQRT(1.00 - TDRHO*TDRHO)
07300   C
07310   C
07320   C
07330         TTLAG1 = TDEV
07340         TEMP = THT + TDEV
07350   C
07360   C
07370   C
07380         IF (TRACE .EQ. ON) WRITE (IS,9007)
07390    9007 FORMAT (' M8')
07400   C
07410   C=====================================================================
07420   C=====================================================================
07430   C
07440   C  DEWPOINT TEMPERATURE SECTION
07450   C
07460          IF (TYPE .EQ. 'REGRS') GO TO 560
07470          IF (TYPE .EQ. 'INDEP') GO TO 570
07480   C
07490   C
07500             WRITE (IS,80) TYPE
07510      80     FORMAT (//1X, 'INVALID DEWPOINT MODEL TYPE....',A5)
07520   C
07530               STOP
07540   C
07550   C
07560   C                                      ****************************
07570   C                                      *** REGRESSION DEWPOINTS ***
07580   C                                      ****************************
07590   C
07600   C
07610     560 CONTINUE
07620   C
07630         CALL DEWSIM (ACOEF,DWLAG1,TEMP,CLD,WDIR,WSP,DEWR)
07640   C
```

```
C  ADD DEVIATIONS TO GENERATED DEW POINTS
C
      CALL NORMAL(ARV)
      DEWDEV = DWBIAS + DWDRHO*(DWDLAG - DWBIAS) +
     $          ARV*DWDEV*SQRT(1.00 - DWDRHO*DWDRHO)
C
      DWDL   = DWDLAG
      DWDLAG = DEWDEV
      DEW = DEWR + DEWDEV
C
      IF (DEW .GE. TEMP) DEW = 0.99*TEMP
      DWL = DWLAG1 + DWDL
      DWLAG1 = DEWR
C
      GO TO 580
C
  570 CONTINUE
C
C                                      ****************************
C                                      ** INDEPENDENT DEWPOINTS **
C                                      ****************************
C
C
      CALL NORMAL (DWX)
      DWL = DWLAG1
      DEW = DWBAR + DWRHO*(DWLAG1-DWBAR) + DWX*DWSDEV*SQRT(1.-DWRHO**2)
      DWLAG1 = DEW
C
      IF (DEW .GE. TEMP) DEW = 0.99*TEMP
C
  580 CONTINUE
C
      IF (TRACE .EQ. ON) WRITE (IS,9009)
 9009 FORMAT (' M9')
C
C
C=======================================================================
C=======================================================================
C
C  LONGWAVE RADIATION SECTION
C
      CALL LONGWV (TEMP1,TEMP,DEW,DWL,CCLAG1,CLD,LW)
C
      TEMP1 = TEMP
C
C
      IF (TRACE .EQ. ON) WRITE (IS,9008)
 9008 FORMAT (' M10')
C
C=======================================================================
C=======================================================================
```

```
08160   C
08170   C  METRIC CONVERSION SECTION
08180   C
08190         TEMPM = (TEMP - 32.00)*(5.00/9.00)
08200         DEWM  = (DEW  - 32.00)*(5.00/9.00)
08210         WSPM  =  WSP*0.4470
08220         RAINM =  RAIN*25.4
08230   C
08240   C  STORE DATA IN THE HOURLY ARRAYS FOR AUTOCORRELATION ANALYSIS
08250   C
08260         CLDATA(JP) = CLD
08270         WSDATA(JP) = WSPM
08280         WDDATA(JP) = WDIR
08290         DWDATA(JP) = DEWM
08300         TCDATA(JP) = TEMPM
08310   C
08320   C=====================================================================
08330   C=====================================================================
08340   C
08350   C  COMPUTE DEWPOINT DEPRESSION AND OUTPUT FOR LATER ANALYSIS
08360   C
08370         DEP(JP) = (TEMPM - DEWM)*WSPM
08380   C
08390         IF ( JP .EQ. 24 ) WRITE (10,582) DEP
08400     582 FORMAT (16F5.1/8F5.1)
08410   C
08420   C
08430   C=====================================================================
08440   C=====================================================================
08450   C
08460   C
08470   C  OUTPUT DATA FOR LAND SURFACE MODEL
08480   C
08490         IF(IPUNCH.GT.0)CALL PUNCH (IPUNCH, RAINM,VP, WSPM, SWR, LW, TEMPM)
08500   C
08510   C=====================================================================
08520   C=====================================================================
08530   C
08540   C
08550         GO TO 506
08560   C
08570   C  DEBUG STATEMENT
08580   C
08590         WRITE (IS,600)JHOUR,JHNEXT,I,RAINM,CLD,SWR,WSPM,WDIR,TEMPM,LW,DEWM
08600     600 FORMAT (1H ,I5,4X,I6,3X,I2,3X,F5.2,3X,F4.2,2X,F4.1,2X,F4.0,
08610        $        2X,F5.0,2X,F5.0,2X,F5.1,2X,F4.0)
08620     506 CONTINUE
08630   C
08640         IH = I
08650   C
08660   C
```

```
08670   C=====================================================================
08680   C=====================================================================
08690   C
08700   C  DATA PLOT SECTION
08710   C
08720         IF (DPLOT .NE. 'Y') GO TO 507
08730   C
08740         PLOTL = -20.
08750         IF (IMO.GE.4 .AND. IMO.LE.10) PLOTL = 0.
08760         PLOTU = PLOTL + 40.
08770   C
08780         CALL PLOT (IMO,IDY,IH,80,TEMPM,DEWM,SWR,LW,CLD,RAINM,WSPM,WDIR,
08790        $           STORM,IPL,PLOTL,PLOTU)
08800   C
08810         IF (TRACE .EQ. ON) WRITE (IS,9010)
08820    9010 FORMAT (' M11')
08830   C
08840   C=====================================================================
08850   C=====================================================================
08860   C
08870     507 CONTINUE
08880   C
08890         IF(I .EQ. 23) IDY = IDY + 1
08900         IF (IDY .GT.31) IDY = 1
08910   C
08920         JHOUR = JHOUR + 1
08930         ST1 = ST2
08940         NDATA = NDATA + 1
08950   C
08960   C  UPDATE THE STATISTICAL ANALYSIS
08970   C
08980         CALL MSTAT (1,TEMPM,DEWM,CLD,WSPM,WDIR,RAWSUM,XXT)
08990   C
09000         IF (TRACE .EQ. ON) WRITE (IS,9020)
09010    9020 FORMAT (' 11A')
09020   C
09030   C
09040   C=====================================================================
09050   C=====================================================================
09060   C
09070   C  UPDATE AUTOCORRELATION ANALYSIS
09080   C
09090   C  THE FIRST 24 HOURS OF THE MONTH ARE NEEDED TO FILL UP THE DATA
09100   C  ARRAYS TO BEGIN THE AUTOCORRELATION ANALYSIS.
09110   C
09120         IF (JHOUR .LE. 24) GO TO 550
09130           NRDATA = NRDATA + 1
09140         CALL RAWLAG(MLAG,JP,TCDATA,TCSUM,TCSMSQ,TCSUM3,TCB,TCV,TCK,NRDATA,
09150        $     TCRHO)
09160         CALL RAWLAG(MLAG,JP,DWDATA,DWSUM,DWSMSQ,DWSUM3,DPB,DPV,DPK,NRDATA,
09170        $     DPRHO)
```

```
09180         CALL RAWLAG(MLAG,JP,CLDATA,CLSUM,CLSMSQ,CLSUM3,CLB,CLV,CLK,NRDATA,
09190        $     CLRHO)
09200         CALL RAWLAG(MLAG,JP,WSDATA,WSSUM,WSSMSQ,WSSUM3,WSB,WSV,WSK,NRDATA,
09210        $     WSRHO)
09220         CALL RAWLAG(MLAG,JP,WDDATA,WDSUM,WDSMSQ,WDSUM3,WDB,WDV,WDK,NRDATA,
09230        $     WDRHO)
09240   C
09250         IF (TRACE .EQ. ON) WRITE (IS,9021)
09260    9021 FORMAT (' 11B')
09270   C
09280     550 CONTINUE
09290   C
09300   C=====================================================================
09310   C=====================================================================
09320   C
09330   C  DIURNAL CURVE SECTION
09340   C
09350   C
09360   C  COMPUTE MEAN AND STANDARD DEVIATIONS FOR EACH HOUR OF THE DAY.
09370   C  SKEWS ARE NOT COMPUTED.
09380   C
09390   C
09400         IF ( I .EQ. 0 ) NDC = NDC + 1
09410   C
09420         CALL STAT (SWR,ASWRS(JP),ASWRSQ(JP),ZZZ,ASWRB(JP),ASWRSD(JP),
09430        $           -999.0,NDC)
09440         CALL STAT (CLD,ACLDS(JP),ACLDSQ(JP),ZZZ,ACLDB(JP),ACLDSD(JP),
09450        $           -999.0,NDC)
09460         CALL STAT (WSPM,AWSPS(JP),AWSPSQ(JP),ZZZ,AWSPB(JP),AWSPSD(JP),
09470        $           -999.0,NDC)
09480         CALL STAT(WDIR,AWDRS(JP),AWDRSQ(JP),ZZZ,AWDRB(JP),AWDRSD(JP),
09490        $           -999.0,NDC)
09500         CALL STAT(TEMPM,ATMPS(JP),ATMPSQ(JP),ZZZ,ATMPB(JP),ATMPSD(JP),
09510        $           -999.0,NDC)
09520         CALL STAT (DEWM,ADEWS(JP),ADEWSQ(JP),ZZZ,ADEWB(JP),ADEWSD(JP),
09530        $           -999.0,NDC)
09540         CALL STAT ( LW,ALWRS(JP),ALWRSQ(JP),ZZZ,ALWRB(JP),ALWRSD(JP),
09550        $           -999.0,NDC)
09560   C
09570         IF (TRACE .EQ. ON) WRITE (IS,9022)
09580    9022 FORMAT (' 11C')
09590   C
09600   C
09610   C=====================================================================
09620   C=====================================================================
09630   C
09640   C
09650   C  UPDATE RAINFALL STATISTICS
09660   C
09670         CALL RSTAT (TRSUM,TBSUM,DHSUM,TRSBAR,TBSBAR,DHBAR,RAINM,STORMS,
09680        $            DRY,  STORM)
```

```
C
      IF (TRACE .EQ. ON) WRITE (IS,9023)
 9023 FORMAT (' 11D')
C
C
C=====================================================================
C=====================================================================
C
C
C  UPDATE THE HISTOGRAMS
C
        CALL HGRAM (TCHIST,TCHDIM,TEMPM,TCDT,TCBASE)
        CALL HGRAM (DPHIST,DPHDIM,DEWM,DPDT,DPBASE)
        CALL HGRAM (CLHIST,CLHDIM,CLD ,CLDT,CLBASE)
        CALL HGRAM (WSHIST,WSHDIM,WSPM,WSDT,WSBASE)
        CALL HGRAM (WDHIS),WDHDIM,WDIR,WDDT,WDBASE)
C
      IF (TRACE .EQ. ON) WRITE (IS,9011)
 9011 FORMAT (' M12')
C
  500 CONTINUE
C
C
C=====================================================================
C=====================================================================
C
C
      IF (JULREL .LT. JREND) GO TO 390
C
C  RESET MONTHLY COUNTERS
C
      JHOUR = 0
C
C
C  RESTART STORM SEQUENCE
C
      CALL RAINST (TB,TR,D,JSINCE,STORM,JHREOS,JHNEXT)
C
  390 CONTINUE
C
C  UPDATE THE DAY COUNTERS
C
      CALL DATEM
C
C  CHECK FOR END OF RUN
C
      IF (TRACE .EQ. ON) WRITE (IS,9012)
 9012 FORMAT (' M13')
C
      IF (JULDAT .LE. JULEND) GO TO 400
C
```

```
C
C=====================================================================
C=====================================================================
C
C  CALL THE FINAL STATISTICAL ANALYSIS SUBROUTINE
C
      CALL FSTAT (5,RAWSUM,XXT,MEAN,COVMAT,CORMAT,NDATA)
C
C  CALL THE AUTOCORRELATION  SUBROUTINE
C
      CALL AUTOCO (MLAG,TCRHO,TCB,TCV,NRDATA,TCTITL)
      CALL AUTOCO (MLAG,DPRHO,DPB,DPV,NRDATA,DWTITL)
      CALL AUTOCO (MLAG,CLRHO,CLB,CLV,NRDATA,CLTITL)
      CALL AUTOCO (MLAG,WSRHO,WSB,WSV,NRDATA,WSTITL)
      CALL AUTOCO (MLAG,WDRHO,WDB,WDV,NRDATA,WDTITL)
C
      IF (TRACE .EQ. ON) WRITE (IS,9013)
 9013 FORMAT (' M14')
C
C
C=====================================================================
C=====================================================================
C
C
C       OUTPUT RESULTS
C
C
      WRITE(IS,715) (MEAN(I),I=1,5)
  715 FORMAT(1H1////T28,'MEAN VALUES'/T9,'TCBAR',T21,'DWBAR',
     $ T33,'CLBAR',T45,'WSBAR',T57,'WDBAR'/1X,5F12.2/)
C
      WRITE(IS,720) ((COVMAT(I,J),J=1,5),I=1,5)
  720 FORMAT(1X,T24,'COVARIANCE MATRIX',/(5(1X,5F12.2/)))
C
      WRITE(IS,740) ((CORMAT(I,J),J=1,5),I=1,5)
  740 FORMAT(1X,T24,'CORRELATION MATRIX',/(5(1X,5F12.2/)))
C
      WRITE (IS,745) TCK,DPK,CLK,WSK,WDK
  745 FORMAT (/T25,'SKEW COEFFICIENTS'/T8,'TCSKEW',T20,'DWSKEW',T32,
     $ 'CLSKEW',T44,'WSSKEW',T56,'WDSKEW'/1X,5F12.2)
C
      WRITE (IS,747) TBSBAR, TRSBAR, DHBAR
  747 FORMAT (///T20,'RAINFALL OUTPUT STATISTICS'//
     $  T24,'TB',T32,'TR',T40,'D'//T22,F6.2,T31,F5.2,T37,F5.2///)
C
C
C=====================================================================
C=====================================================================
C
C
C  PRINT HISTOGRAMS OF THE GENERATED DATA
```

```
C
      CALL PRINTH (TCHIST,TCHDIM,TCDT,TCBASE,TCTITL,NDATA)
      CALL PRINTH (DPHIST,DPHDIM,DPDT,DPBASE,DWTITL,NDATA)
      CALL PRINTH (CLHIST,CLHDIM,CLDT,CLBASE,CLTITL,NDATA)
      CALL PRINTH (WSHIST,WSHDIM,WSDT,WSBASE,WSTITL,NDATA)
      CALL PRINTH (WDHIST,WDHDIM,WDDT,WDBASE,WDTITL,NDATA)
C
C
C=====================================================================
C=====================================================================
C
C
C  CONVERT VARIANCES TO STANDARD DEVIATIONS.
C
      DO 800 IG = 1,24
C
      ASWRSD(IG) = SQRT(ASWRSD(IG))
      ACLDSD(IG) = SQRT(ACLDSD(IG))
      AWSPSD(IG) = SQRT(AWSPSD(IG))
      AWDRSD(IG) = SQRT(AWDRSD(IG))
      ATMPSD(IG) = SQRT(ATMPSD(IG))
      ADEWSD(IG) = SQRT(ADEWSD(IG))
      ALWRSD(IG) = SQRT(ALWRSD(IG))
C
  800 CONTINUE
C
C
C=====================================================================
C=====================================================================
C
C
C  COMPUTE TOTAL DAILY SHORT AND LONGWAVE RADIATION
C
      DO 810 IG = 1,24
C
      SUMSW = SUMSW + ASWRB(IG)
      SUMLW = SUMLW + ALWRB(IG)
C
  810 CONTINUE
C
C
C=====================================================================
C=====================================================================
C
C
C  PRINT HOURLY MEANS AND STANDARD DEVIATIONS.
C
      CALL HOUR (ASWRB,ASWRSD,'SWR')
        WRITE (IS,811) SUMSW
  811   FORMAT (T29,'TOTAL=',T37,F7.2)
      CALL HOUR (ALWRB,ALWRSD,'LWR')
```

```
11220           WRITE (IS,811) SUMLW
11230         CALL HOUR (ACLDB,ACLDSD,'CLD')
11240         CALL HOUR (AWSPB,AWSPSD,'WSP')
11250         CALL HOUR (AWDRB,AWDRSD,'WDR')
11260         CALL HOUR (ATMPB,ATMPSD,'TMP')
11270         CALL HOUR (ADEWB,ADEWSD,'DEW')
11280   C
11290         IF (TRACE .EQ. ON) WRITE (IS,9014)
11300    9014 FORMAT (' M15')
11310   C
11320         WRITE (IS,760)
11330     760 FORMAT (1H1,15(15(5H      )/))
11340         STOP
11350         END
11360   C
11370   C.....................................................................
11380   C
11390         SUBROUTINE VARYX (X1,X2,T1,T2,ST,X)
11400   C
11410   C  ROUTINE TO LINEARLY INTERPOLATE A VALUE OF X
11420   C
11430         RANGE1 = T2 - T1
11440         RANGE2 = 24.0 - RANGE1
11450   C
11460         IF (ST .GT. T1) GO TO 100
11470         X = X2 - (X2 - X1)*((24.0 - T2 + ST)/RANGE2)
11480         RETURN
11490   C
11500     100 IF (ST .GT. T2) GO TO 200
11510         X = X1 + (X2 - X1)*((ST - T1)/RANGE1)
11520         RETURN
11530   C
11540     200 CONTINUE
11550         X = X2 - (X2 - X1)*((ST - T2)/RANGE2)
11560         RETURN
11570   C
11580         END
11590   C
11600   C.....................................................................
11610   C
11620         SUBROUTINE HOUR ( BAR, DEV, TITLE )
11630   C
11640   C  PRINT HOURLY MEANS AND STANDARD DEVIATIONS.
11650   C
11660         DIMENSION BAR(1), DEV(1)
11670   C
11680         COMMON /IO/     IN,IS,IB
11690   C
11700   C
11710         WRITE (IS,100)
11720     100 FORMAT (1H1,15(5H      )/1H ,15(5H      ))
```

```
11730   C
11740         WRITE (IS,200) TITLE
11750     200 FORMAT (////T41,A3//T30,'HOUR',T40,'MEAN',T49,'ST.DEV.'/)
11760   C
11770         DO 300 I = 1,24
11780   C
11790         II = I - 1
11800         WRITE (IS,250) II, BAR(I), DEV(I)
11810     250 FORMAT (T31,I2,T38,F6.2,T47,F6.2)
11820   C
11830     300 CONTINUE
11840   C
11850         WRITE (IS,400)
11860     400 FORMAT (////)
11870   C
11880   C
11890         RETURN
11900         END
11910   C
11920   C.....................................................................
11930   C
11940         SUBROUTINE RSTAT (TRSUM,TBSUM,DHSUM,TR,TB,DH,RAIN,STORMS,DR
11950        $                  DRY,STFLAG)
11960   C
11970   C
11980   C  COMPUTE STATISTICS FOR RAINFALL MODEL
11990   C    TR......MEAN STORM DURATION
12000   C    TB......MEAN TIME BETWEEN STORMS
12010   C    DH......MEAN STORM DEPTH
12020   C
12030   C
12040         COMMON /SFLAG/ RSTORM
12050   C
12060         DATA  ON/'ON'/, OFF/'OFF'/
12070   C
12080   C
12090         IF (STFLAG .EQ. ON) GO TO 350
12100   C
12110   C  STORM FLAG IS OFF.  THEREFORE WE ARE BETWEEN STORMS.
12120   C
12130         IF (RSTORM .EQ. OFF) GO TO 310
12140   C
12150   C  FIRST HOUR OF INTERSTORM PERIOD.
12160   C
12170         RSTORM = OFF
12180         DRY    = DRY + 1.0
12190   C
12200     310 CONTINUE
12210   C
12220         TBSUM = TBSUM + 1.0
12230         GO TO 300
```

```
12240   C
12250     350 CONTINUE
12260   C
12270   C  STORM FLAG IS ON.  THEREFORE, WE ARE IN A STORM.
12280   C
12290         IF (RSTORM .EQ. ON) GO TO 360
12300   C
12310   C  FIRST HOUR OF NEW STORM.
12320   C
12330         RSTORM = ON
12340   C
12350   C  PUT STATEMENTS HERE IF THERE IS SOME MINIMUM TIME BETWEEN STORMS
12360   C  CRITERION THAT MUST BE CHECKED.
12370   C
12380         STORMS = STORMS + 1.0
12390   C
12400     360 CONTINUE
12410   C
12420         TRSUM = TRSUM + 1.0
12430         DHSUM = DHSUM + RAIN
12440   C
12450     300 CONTINUE
12460   C
12470         IF (STORMS .LT. 0.01) GO TO 400
12480         TR = TRSUM/STORMS
12490         DH = DHSUM/STORMS
12500     400 IF (DRY    .LT. 0.01) GO TO 500
12510         TB = TBSUM/DRY
12520   C
12530     500 CONTINUE
12540   C
12550         RETURN
12560         END
12570   C
12580   C.....................................................................
12590   C
12600   C
12610         SUBROUTINE NORMAL (X)
12620   C
12630   C  GENERATE A NORMALLY DISTRIBUTED RANDOM DEVIATE FROM N(0,1).
12640   C  REFERENCE:  FISHMAN,GEORGE S.:CONCEPTS AND METHODS IN DISCRETE EVENT
12650   C              DIGITAL SIMULATION, WILEY AND SONS, 1973, PP 211.
12660   C
12670      10 U1 = RAND(0)
12680         IF (U1 .LT. 0.00001) GO TO 10
12690         U2 = RAND(0)
12700         X = SQRT (-2.0*ALOG(U1))*COS(6.28319*U2)
12710   C
12720         RETURN
12730         END
12740   C
```

```
12750  C.....................................................................
12760  C
12770        SUBROUTINE VAPOR (T,TD,E,ES)
12780  C
12790  C
12800  C  ROUTINE TO COMPUTE ATMOSPHERIC VAPOR PRESSURE GIVEN
12810  C  TWO METER TEMPERATURE AND DEWPOINT.
12820  C
12830  C     T    ... TEMPERATURE - DEG C
12840  C     TD   ... DEWPOINT TEMPERATURE - DEG C
12850  C     E    ... VAPOR PRESSURE - MILLIBAR
12860  C     ES   ... SATURATED VAPOR PRESSURE - MILLIBAR
12870  C     C0-C5... COEFFICIENTS IN SAT. VAPOR PRESS. APPROX.
12880  C     R    ... RELATIVE HUMIDITY
12890  C
12900  C
12910  C
12920        DOUBLE PRECISION C0,C1,C2,C3,C4,C5
12930  C
12940        DATA C0/6.0689226    /
12950        DATA C1/4.4358312E-01/
12960        DATA C2/1.4590816E-02/
12970        DATA C3/2.7619554E-04/
12980        DATA C4/2.9952590E-06/
12990        DATA C5/1.4398885E-08/
13000  C
13010  C
13020  C
13030  C  COMPUTE RELATIVE HUMIDITY
13040  C
13050        R = ((112. - 0.1*T + TD)/(112. + 0.9*T))**8.0
13060  C
13070  C
13080  C   COMPUTE SATURATION VAPOR PRESSURE
13090  C
13100        X = C4 + T*C5
13110        X = C3 + T*X
13120        X = C2 + T*X
13130        X = C1 + T*X
13140        ES= C0 + T*X
13150  C
13160  C
13170  C  COMPUTE ATMOSPHERIC VAPOR PRESSURE
13180  C
13190        E = R*ES
13200  C
13210  C
13220        RETURN
13230        END
13240  C
13250  C.....................................................................
```

```
C
      SUBROUTINE INTER (DAFILE,IS,DPLOT,IPL)
C
C  ROUTINE TO READ THE NECCESSARY RUNTIME INFORMATION FROM THE
C  CONSOLE.
C
      DIMENSION PTEXT(16)
C
      COMMON /SEED/  ISEED
      COMMON /PUNCHD/ PTEXT, PNFILE, IPUNCH
C
      DOUBLE PRECISION  DAFILE, PNFILE
C
C
C  SET THE CONSOLE UNIT NUMBER FOR THIS MACHINE.
C
      IC = 5
C
      DAFILE = 'DCCMOD.DAT'
      WRITE (IC,90) DAFILE
   90 FORMAT (//1X,'THE CURRENT DATA FILE IS ',A10/
     $  1X,'DO YOU WISH TO READ A DIFFERENT ONE? (Y/N)')
C
      READ (IC,110) ANS
      IF (ANS .NE. 'Y') GO TO 115
C
C
      WRITE (IC,100)
  100 FORMAT (//1X,'WHAT DATA FILE CONTAINS THE INPUT DATA?'/
     $ 1X,'   ENTER FILE NAME IN THE FORM  XXXXXX.YYY ')
C
      READ (IC,110) DAFILE
  110 FORMAT (A)
C
  115 CONTINUE
C
      WRITE (IC,120)
  120 FORMAT (/1X,'DO YOU WANT TO PRINT THE INPUT SUMMARY TO THE ',
     $ 'CONSOLE? (Y/N)')
C
      READ (IC,130)ANS
  130 FORMAT (A)
C
      IF (ANS .EQ.'Y') IS = IC
C
      WRITE (IC,140)
  140 FORMAT (/1X,'INPUT SEED FOR THE RANDOM NUMBER GENERATOR')
      READ (IC,150) ISEED
  150 FORMAT (I)
C
      WRITE (IC,160)
```

```
13770   160 FORMAT (/1X,'DO YOU WANT A PLOT? (Y/N)')
13780   C
13790       READ (IC,130) DPLOT
13800       IF (DPLOT .NE. 'Y') GO TO 185
13810   C
13820       WRITE (IC,170)
13830   170 FORMAT (/1X,'WHICH PLOT?  1 FOR 6-VARIABLE'/
13840      $        1X,'            2 FOR 4-VARIABLE'/)
13850   C
13860       READ (IC,180) IPL
13870   180 FORMAT (I)
13880   C
13890   C
13900   185 CONTINUE
13910   C
13920       IPUNCH = 0
13930       WRITE (IC,190)
13940   190 FORMAT (/1X,'CREATE LAND SURFACE MODEL DATA FILE? (Y/N)')
13950   C
13960       READ (IC,130) ANS
13970       IF (ANS .NE. 'Y') GO TO 230
13980   C
13990       IPUNCH = 27
14000       WRITE (IC,200)
14010   200 FORMAT (/1X,'ENTER DATA FILE NAME ... XXXXXX.YYY')
14020   C
14030       READ (IC,110) PNFILE
14040   C
14050       WRITE (IC,210)
14060   210 FORMAT (/1X, 'ENTER COMMENTS TO IDENTIFY OUTPUT DATA ',
14070      $            '(80 CHAR. MAX)')
14080       READ (IC,220) (PTEXT(I),I=1,16)
14090   220 FORMAT (16A5)
14100   C
14110       WRITE (IC,240) IPUNCH,PNFILE,(PTEXT(I),I=1,16)
14120   240 FORMAT (1X,I5/1X,A10/1X,16A5)
14130   230 CONTINUE
14140   C
14150       RETURN
14160       END
14170   C
14180   C.....................................................................
14190   C
14200       SUBROUTINE START (ISEED)
14210   C
14220   C       ACTIVATE RAND ISEED TIMES TO PROVIDE A DIFFERENT
14230   C          STARTING POINT IN THE GENERATION OF RANDOM NUMBERS
14240   C          WITH EACH INPUT OF ISEED
14250   C
14260   C
14270       DO 100 I = 1,ISEED
```

```
14280   C
14290           X = RAND(0)
14300   C
14310     100 CONTINUE
14320   C
14330   C
14340         RETURN
14350         END
14360   C
14370   C.....................................................................
14380   C
14390         SUBROUTINE READF (IN,IS,IB)
14400   C
14410   C
14420   C  THIS SUBPROGRAM READS THE INPUT DATA FOR THE STOCHASTIC
14430   C  HYDROMETEOROLOGICAL MODEL.
14440   C
14450   C
14460         DIMENSION TITLE(16,3), BCOEF(8), ACOEF(8)
14470         DIMENSION CCPDF(30), CCORD(30)
14480         DIMENSION DRPDF(30), DRORD(30)
14490   C
14500   C
14510         DOUBLE PRECISION  WRITEF, BUGOFF, TZONE, TZ(4), DEBUG(7)
14520         DOUBLE PRECISION  OUTPUT
14530   C
14540   C
14550         REAL LAT(3), LONG(3)
14560   C
14570         COMMON /TITLES/ TITLE
14580         COMMON /FILES/  WRITEF, OUTPUT, BUGOFF
14590         COMMON /DATES/  IYR,IMO,IDAY, LYR,LMO,LDAY
14600         COMMON /LOCATE/ LAT, LONG, TZONE
14610         COMMON /DBUG/   NBUG,DEBUG
14620         COMMON /CLOUDS/ CCBAR,CCSD,CCRHO,BETA,GAM
14630         COMMON /ATMOS/  EN
14640         COMMON /PDFCLD/ NUMCC,CCPDF,CCORD,CCA,CCB
14650         COMMON /RAINS/  TBBAR, TRBAR, DBAR
14660         COMMON /TEMPAR/ TDBIAS, TDSDEV, TDRHO, BCOEF, TEMBAR
14670         COMMON /WINDSP/ SPBAR1,SPBAR2,SPBT1,SPBT2,SPSDV1,SPSDV2,
14680        $                SPSDT1,SPSDT2,SPSKEW,SPRHO
14690         COMMON /WINDIR/ DRBAR,DRDEV,DRRHO
14700         COMMON /PDFDIR/ NUMDR,DRPDF,DRORD,DRA,DRB
14710         COMMON /DEWONE/ TYPE, ACOEF
14720         COMMON /DEWTWO/ DWBAR,DWSDEV,DWSKEW,DWRHO
14730         COMMON /DEWDVS/ DWBIAS, DWDEV, DWDRHO
14740   C
14750   C  READ THE GENERAL DATA SECTION
14760   C
14770   C    NOTE: 'DUMMY' READS ARE INSERTED TO READ THE 'CARDS' THAT SEPERATE
14780   C          THE MAJOR SECTIONS OF THE INPUT DATA.  IT IS DESIGNED TO
```

```
C         MAKE HANDLING THE DATA DECK EASIER AND AS A MEANS TO MAKE
C         EXAMINATION OF THE DATA DECK EASIER.
C
C
C
      READ (IN,10) DUMMY
   10 FORMAT (A)
C
C  READ THE TITLE CARDS (3)
C
      DO 15 I = 1,3
         READ (IN,20) (TITLE(J,I),J=1,15)
   20    FORMAT (15A5)
   15    CONTINUE
C
C  READ THE DATA FILE NAMES FOR THE GENERAL OUTPUT AND DEBUG INFO
C
      READ (IN,30) WRITEF, OUTPUT, BUGOFF
   30 FORMAT (10X,3A10)
C
C=============================================================================
C
C OPEN FILES FOR OUTPUT
C
      IB = 22
      OPEN (UNIT=IB,DEVICE='DSK',ACCESS='SEQOUT',FILE=BUGOFF)
      IF (IS .EQ. 5) GO TO 35
      IS = 23
      OPEN (UNIT=IS,DEVICE='DSK',ACCESS='SEQOUT',FILE=WRITEF)
   35 CONTINUE
C
C=========================================================================
C
C  READ DATES, LATITUDE, LONGITUDE, AND TIME ZONE
C
      READ (IN,40) IMO,IDAY,IYR,LMO,LDAY,LYR,
     $     (LAT(I),I=1,3),(LONG(I),I=1,3),TZONE
   40 FORMAT (10X,I2,1X,I2,1X,I4,1X,I2,1X,I2,1X,I4,1X,
     $     2(F3.0,1X,F2.0,1X,F2.0,1X),A10)
C
C  READ DEBUG INFO
C
C   DEBUG INFORMATION CAN BE OUTPUT FROM SEVERAL SUBROUTINES BY SIMPLY
C   READING IN THE APPROPRIATE SUBROUTINE NAME. THESE SUBROUTINES
C   INCLUDE:  TAU, DECL, SOLRAD, CLRSKY, COVER, ARVA, AND TEMPSN.
C   NAMES ARE LEFT JUSTIFIED.
C
C   THIS FEATURE IS CURRENTLY DISABLED.
C
      NBUG = 0
      GO TO 51
```

```
15300   C
15310         READ (IN,50) NBUG,(DEBUG(I),I=1,NBUG)
15320      50 FORMAT (10X,I2,6A10)
15330      51 CONTINUE
15340   C
15350   C  READ CLOUD AND RADIATION DATA
15360   C
15370         READ (IN,10) DUMMY
15380   C
15390   C  READ PARAMETER CARD
15400   C
15410         READ (IN,60) CCBAR, CCSD, CCRHO, GAM, EN
15420      60 FORMAT (10X,6F10.0)
15430   C
15440         READ (IN,62) NUMCC, CCA, CCB
15450      62 FORMAT (10X,I2,8X,2F10.0)
15460   C
15470         READ (IN,64) (CCPDF(I),I=1,NUMCC)
15480         READ (IN,64) (CCORD(I),I=1,NUMCC)
15490      64 FORMAT (10X,6F10.0)
15500   C
15510   C  READ RAIN MODEL PARAMETERS
15520   C
15530         READ (IN,10) DUMMY
15540         READ (IN,60) TBBAR, TRBAR, DBAR
15550   C
15560   C  READ TEMPERATURE DATA
15570   C
15580         READ (IN,10) DUMMY
15590         READ (IN,70) (BCOEF(I),I=1,8)
15600      70 FORMAT(10X,4E12.5)
15610         READ (IN,60)  TDBIAS, TDSDEV, TDRHO
15620   C
15630   C  READ WIND SPEED PARAMETER DATA
15640   C
15650         READ (IN,10) DUMMY
15660         READ (IN,60) SPBAR1, SPBAR2, SPBT1, SPBT2
15670         READ (IN,60) SPSDV1, SPSDV2, SPSDT1, SPSDT2
15680         READ (IN,60) SPSKEW, SPRHO
15690   C
15700   C  READ WIND DIRECTION DATA
15710   C
15720         READ (IN,10) DUMMY
15730         READ (IN,60) DRBAR, DRDEV, DRRHO
15740         READ (IN,62) NUMDR,DRA,DRB
15750         READ (IN,64) (DRPDF(I),I=1,NUMDR)
15760         READ (IN,64) (DRORD(I),I=1,NUMDR)
15770   C
15780   C  READ DEWPOINT MODEL PARAMETERS
15790   C
15800         READ (IN,10) DUMMY
```

```
      READ (IN,10) TYPE
C
      IF (TYPE.EQ.'REGRS' .OR. TYPE.EQ.'INDEP') GO TO 100
C
      WRITE (IS,95) TYPE
   95 FORMAT (///,T5,'***** DEWPOINT MODEL TYPE --',A5,'-- IS INVALID.'/
     $          T11,      'ONLY ''INDEP'' OF ''REGRS'' ARE ACCEPTABLE'))))')')
      STOP
C
C
  100 IF ( TYPE .EQ. 'INDEP' ) READ (IN,60) DWBAR, DWSDEV,  DWRHO
      IF ( TYPE .EQ. 'REGRS' ) READ (IN,70) (ACOEF(I),I=1,6)
      IF ( TYPE .EQ. 'REGRS' ) READ (IN,60) DWBIAS, DWDEV, DWDRHO
C
C
C
C
C
C ::::: INPUT DATA SUMMARY  ::::::::::::::::::::::::::::::::
C
C
C  PRINT GENERAL DATA
C
C
      WRITE (IS,490)
  490 FORMAT (1H1,3(15(5H      )/))
C
      WRITE (IS,491)
  491 FORMAT (1X,79(1H*)/)
C
      WRITE (IS,492)
  492 FORMAT (1X,120,'CONSTRAINED STOCHASTIC CLIMATE SIMULATION'/
     $          T33,'INPUT SUMMARY'/)
C
      WRITE (IS,491)
C
C
      DO 510 J = 1,3
      WRITE (IS,500) (TITLE(I,J),I=1,15)
  500 FORMAT (1H ,15A5)
  510 CONTINUE
C
      WRITE (IS,491)
C
      IF (IS.EQ.5) WRITEF = 'CONSOLE'
      WRITE (IS,515) WRITEF, OUTPUT, BUGOFF
  515 FORMAT (//1X,T31,'OUTPUT FILE NAMES'//T31,'WRITEF: ',A10/
     $                                       T31,'OUTPUT: ',A10/
     $                                       T31,'BUGOFF: ',A10/)
C
      WRITE (IS,491)
```

```
C
      WRITE (IS,520) IMO,IDAY,IYR,LMO,LDAY,LYR
  520 FORMAT (//T12,'BEGINNING DATE ',2X,I2,'/',I2,'/',I4,5X,
     $ 'ENDING DATE ',2X,I2,'/',I2,'/'I4)
C
      WRITE (IS,530) (LAT(I),I=1,3),(LONG(I),I=1,3),TZONE
  530 FORMAT(//1X,'LATITUDE = ',2X,F4.0,F3.0,F3.0,5X,
     $ 'LONGITUDE = ',2X,F4.0,F3.0,F3.0,5X,'TIME ZONE =  ',A10/)
C
      WRITE (IS,491)
C
      IF (NBUG.EQ.0) GO TO 545
      WRITE (IS,540)(DEBUG(I),I=1,NBUG)
  540 FORMAT (//1X,'DEBUG SUBROUTINES - ',7A10)
      WRITE (IS,491)
  545 CONTINUE
C
C
C  PRINT CLOUD AND SKY PARAMETERS
C
      WRITE (IS,550)
  550 FORMAT (1X,T28,'CLOUD AND SKY PARAMETERS'/)
C
      WRITE (IS,551) CCBAR,CCSD,CCRHO
  551 FORMAT (1X,T28,'FAIRWEATHER CLOUD COVER'/
     $          T31,'MEAN',T42,F6.2/
     $          T31,'ST. DEV.',T42,F6.2/
     $          T31,'LAG-1 COEF. ',T42,F6.2//)
C
      WRITE (IS,552)
  552 FORMAT (1X,T23,'FAIRWEATHER CLOUD COVER HISTOGRAM'/)
C
      CALL PRDIST ( CCPDF, CCORD,  NUMCC )
C
C
C
      WRITE (IS,562) CCA,CCB
  562 FORMAT (//T5,'LEFT BOUND OF HIST. = ',F10.4,5X,
     $            'RIGHT BOUND OF HIST. = ',F10.4)
C
      WRITE (IS,553) GAM, EN
  553 FORMAT (//1X,T23,'CLOUD COVER DECAY COEFFICENT = ',F5.3/
     $            T23,'ATMOSPHERIC TURBIDITY FACTOR = ',F4.1/)
C
      WRITE (IS,491)
      WRITE (IS,490)
      WRITE (IS,491)
      WRITE (IS,492)
      WRITE (IS,491)
C
C  PRINT PRECIPITATION MODEL PARAMETERS
```

```
C
      WRITE (IS,565) TBBAR,TRBAR,DBAR
  565 FORMAT (/1X,T25,'PRECIPITATION MODEL PARAMETERS'//
     $           T23,'MEAN TIME BETWEEN STORMS',T50,F7.2/
     $           T23,'MEAN STORM DURATION',T50,F7.2/
     $           T23,'MEAN STORM DEPTH',T50,F7.2/)
C
      WRITE (IS,491)
C
C
C
C  PRINT TEMPERATURE MODEL PARAMETERS
C
      WRITE (IS,570)(BCOEF(I),I=1,8)
  570 FORMAT (//1X,T26,'TEMPERATURE MODEL PARAMETERS'//
     $  T22,'B0',2X,E12.5,4X,'B1',2X,E12.5/
     $  T22,'B2',2X,E12.5,4X,'B3',2X,E12.5/
     $  T22,'B4',2X,E12.5,4X,'B5',2X,E12.5/
     $  T22,'B6',2X,E12.5,4X,'B7',2X,E12.5//)
C
C
      WRITE (IS,580) TDBIAS, TDSDEV, TDRHO
  580 FORMAT (1X, T30,'STOCHASTIC COMPONENT'//
     $   T28,'TEMPERATURE BIAS',2X,F6.2/
     $   T28,'ST. DEVIATION  ',2X,F6.2/
     $   T28,'LAG-1 COR. COEF.',2X,F6.2/)
C
C
      WRITE (IS,491)
      WRITE (IS,490)
      WRITE (IS,491)
      WRITE (IS,492)
      WRITE (IS,491)
C
C
C  PRINT WIND SPEED MODEL PARAMETERS
C
      WRITE (IS,600) SPBAR1,SPBT1,SPBAR2,SPBT2,
     $              SPSDV1,SPSDT1,SPSDV2,SPSDT2
  600 FORMAT (/T29,'WIND SPEED PARAMETERS'//
     $   T21,'MIN HOURLY MEAN  = ',F4.1,' AT ',F5.2,' HOURS'/
     $   T21,'MAX HOURLY MEAN  = ',F4.1,' AT ',F5.2,' HOURS'//
     $   T21,'MIN HOURLY ST DEV= ',F4.1,' AT ',F5.2,' HOURS'/
     $   T21,'MAX HOURLY ST DEV= ',F4.1,' AT ',F5.2,' HOURS'//)
C
      WRITE (IS,601) SPSKEW, SPRHO
  601 FORMAT (//T29,'SKEW COEFFICIENT ',F5.2/
     $         T29,'LAG-1 COEFFICIENT',F5.2/)
C
      WRITE (IS,491)
C
```

```
C
C  PRINT WIND DIRECTION MODEL PARAMETERS
C
      WRITE (IS,620) DRBAR,DRDEV,DRRHO
  620 FORMAT (//T27,'WIND DIRECTION PARAMETERS'//
     $          T31,'MEAN',T42,F6.2/
     $          T31,'ST. DEV.',T42,F6.2/
     $          T31,'LAG-1 COEF',T42,F6.2//)
C
C
      WRITE (IS,630)
  630 FORMAT (//T28,'WIND DIRECTION HISTOGRAM'/)
C
      CALL PRDIST ( DRPDF, DRORD, NUMDR )
C
      WRITE (IS,632) DRA,DRB
  632 FORMAT (//T5,'LEFT BOUND OF HIST. = ',F10.4,5X,
     $           'RIGHT BOUND OF HIST. = ',F10.4/)
C
      WRITE (IS,491)
C
      IF (TYPE .EQ. 'REGRS') GO TO 665
C
C
C  PRINT DEWPOINT MODEL PARAMETERS ... 'INDEPENDENT GENERATION'
C
      WRITE (IS,640) DWBAR, DWSDEV, DWRHO
  640 FORMAT (//T27,'DEWPOINT MODEL PARAMETERS'//
     $          T31,'MEAN',T42,F6.2/
     $          T31,'ST DEV',T42,F6.2/
     $          T31,'LAG-1 COEF',T42,F6.2//)
C
      WRITE (IS,491)
C
      GO TO 700
C
  665 CONTINUE
C
C
C  PRINT DEW POINT MODEL PARAMETERS ... 'REGRESSION TYPE '
C
      WRITE (IS,670)(ACOEF(I),I=1,6)
  670 FORMAT (/T21,'DEWPOINT MODEL REGRESSION COEFFICIENTS'//
     $  T22,'A0',2X,E12.5,4X,'A1',2X,E12.5/
     $  T22,'A2',2X,E12.5,4X,'A3',2X,E12.5/
     $  T22,'A4',2X,E12.5,4X,'A5',2X,E12.5//)
C
      WRITE (IS,680) DWBIAS, DWDEV, DWDRHO
  680 FORMAT (//T24,'STOCHASTIC COMPONENT PARAMETERS'/
     $          T28,'DEWPOINT BIAS  ',2X,F6.2/
     $          T28,'ST DEVIATION   ',2X,F6.2/
```

```
17850       $          T28,'LAG-1 COR COEF ',2X,F6.2//)
17860   C
17870       WRITE (IS,491)
17880   C
17890     700 CONTINUE
17900   C
17910   C
17920       RETURN
17930       END
17940   C
17950   C.....................................................................
17960   C
17970       SUBROUTINE FRDIST ( H, ORD, NMAX)
17980   C
17990   C  PRINT OUT THE INPUT PROBABILITY MASS FUNCTION
18000   C
18010   C
18020   C
18030       COMMON /IO/     IN,IS,IB
18040   C
18050   C
18060       DIMENSION ORD(NMAX)
18070       DIMENSION H(NMAX)
18080       DIMENSION FMT(6),FMT1(10)
18090   C
18100       DATA FMT  /'(T8 ,',',',',   10','(7H--','-----',','))'/
18110       DATA FMT1 /'    1','    2','    3','    4','    5',
18120      $           '    6','    7','    8','    9','   10'/
18130   C
18140       DO 100 J = 1,NMAX,10
18150   C
18160       IMAX = 10
18170       IF( J+10 .GT. NMAX ) IMAX = NMAX - J + 1
18180   C
18190   C
18200       WRITE (IS,200)(H(I+J-1),I=1,IMAX)
18210     200 FORMAT (1H ,1X,'HIST ',10(1X,F6.2))
18220   C
18230       FMT(3) = FMT1(IMAX)
18240   C
18250   C
18260       WRITE (IS,FMT)
18270   C
18280   C
18290       WRITE (IS,400) (ORD(I+J-1),I=1,IMAX)
18300     400 FORMAT(1H ,1X,'ORD  ',10(1X,F6.2)//)
18310   C
18320   C
18330     100 CONTINUE
18340   C
18350   C
```

```
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE DATE1
C
C     DATE1 INITIALIZES THE DATE COUNTERS.
C     JULIAN DATES ARE USED.
C
C
C     IYR   ... INITIAL YEAR
C     IMO   ... INITIAL MONTH
C     IDAY  ... INITIAL DAY
C     LYR   ... LAST YEAR
C     LMO   ... LAST MONTH
C     LDAY  ... LAST DAY
C
C     JULDAT... CURRENT JULIAN DATE
C     JBEGIN... JULIAN DATE AT BEGINNING OF RUN
C     JULEND... JULIAN DATE AT END OF RUN
C     JRANGE... LENGTH OF RUN
C     JULREF... JAN 1 OF INITIAL YEAR
C     JULREL... JULIAN DATE RELATIVE TO JAN 1 OF CURRENT YEAR
C     JSTART... RELATIVE JULIAN DATE TO BEGIN MONTHLY PARAMETER
C               ESTIMATION RANGE
C     JSTOP ... RELATIVE JULIAN DATE TO END MONTHLY PARAMETER
C               ESTIMATION RANGE
C     JYEAR ... YEAR COUNTER
C
C     NXLPYR... JULIAN DATE OF DEC 31 OF NEXT LEAP YEAR
C
C
C
      COMMON /DATES/ IYR, IMO, IDAY, LYR, LMO, LDAY
      COMMON /JDATES/ JULDAT, JULREL, JBEGIN, JULEND, JRANGE, NXLPYR
     $              ,JSTART, JSTOP,  JREND,  JYEAR
      COMMON /IO/    IREAD, IWRITE, IWBUG
C
      INTEGER IDBUG, CAL(12)
C
      DATA CAL /31,28,31,30,31,30,31,31,30,31,30,31/
C
C     SET DEBUG FLAG
C
      IDBUG = 0
C
C
C
C     DETERMINE INITIAL JULIAN DATES
C
```

```
      CALL JULIAN (IMO,IDAY,IYR,JBEGIN)
      CALL JULIAN (LMO,LDAY,LYR,JULEND)
      CALL JULIAN (  1,   1,IYR,JULREF)
C
C
      JULREL = JBEGIN - JULREF
      JRANGE = JULEND - JBEGIN + 1
      JULDAT = JBEGIN - 1
C
C        DETERMINE THE NEXT OCCURANCE OF 12/31/(LEAP YEAR)
C              (IE. THE 366TH DAY OF THE YEAR)
C
C
      LASTLP = IYR - MOD(IYR,4)
C
      CALL JULIAN (12.31,LASTLP,NXLPYR)
C
      IF (JULDAT .GE. NXLPYR) NXLPYR = NXLPYR + 1461
C
C
C        NOTE... 1461 = 365 + 365 + 365 + 366
C
C
C     THIS SECTION DEFINES VARIABLES NEEDED FOR MONTHLY
C     PARAMETER ESTIMATION
C
        JYEAR = IYR
        JSTART = JULREL + 1
        LD = CAL(LMO)
C
        CALL JULIAN ( LMO, LD, IYR, JDATE )
C
        JSTOP = JDATE - JULREF + 1
        JREND = JSTOP
C
          IF (JYEAR - MOD(JYEAR,4)) 65,70,65
   70     IF (IMO.EQ.2 .AND. IDAY.EQ.28) JREND = JSTOP + 1
   65   CONTINUE
C
C
      ENTRY DATE
C
C        THE NEXT SECTION IS USED EACH DAY TO UPDATE
C        THE JULIAN DATE COUNTERS IF ANNUAL PARAMETERS ARE USED.
C
      JULREL = JULREL + 1
      JULDAT = JULDAT + 1
C
C
C        CHECK FOR END OF YEAR
C
```

```
      IF (JULREL .LE. 365) GO TO 100
      IF (JULREL .GT. 366) GO TO 200
C
C        CHECK FOR LEAP YEAR
C
      IF (JULDAT.NE.NXLPYR) GO TO 200
C
C        YES, THERE ARE 366 DAYS THIS YEAR.
C        UPDATE NXLPYR TO NEXT LEAP YEAR.
C
      NXLPYR = NXLPYR + 1461
C
C
      IF( IDBUG .NE. 0 ) GO TO 900
C
C
   50 CONTINUE
C
C
      RETURN
C
C
C        RESET RELATIVE JULIAN DATE
C
  200 JULREL = 1
C
C
  100 CONTINUE
         IF( IDBUG .NE. 0 ) GO TO 900
  110 RETURN
C
C
      ENTRY DATEM
C
C     THIS SECTION IS USED EACH DAY TO UPDATE THE JULIAN DATE COUNTERS
C     IF MONTHLY PARAMETER ESTIMATION IS USED.
C
        JULREL = JULREL + 1
        JULDAT = JULDAT + 1
C
        IF (JULREL .LE. JREND) GO TO 400
C
C     UPDATE THE JULIAN COUNTERS
C
        JYEAR = JYEAR + 1
C
        CALL JULIAN (IMO,01,JYEAR,JBEGIN)
        CALL JULIAN (01,01,JYEAR,JULREF)
C
        JULREL = JBEGIN - JULREF + 1
        JSTART = JULREL
```

```
C
      CALL JULIAN (IMO,01,JYEAR,JULDAT)
C
      LD = CAL(LMO)
C
      CALL JULIAN ( LMO, LD, JYEAR, JDATE )
C
      JSTOP = JDATE - JULREF + 1
      JREND = JSTOP
C
      IF(JYEAR - MOD(JYEAR,4)) 400,410,400
  410 IF (LMO.EQ.2 .AND. LDAY.EQ.28) JREND = JREND + 1
  400 CONTINUE
C
      IF ( IDBUG .NE. 0 ) GO TO 900
C
C
      RETURN
C
C
C
  900 CONTINUE
C
C       DEBUG INFORMATION FOR JULIAN DATE CALCULATIONS
C
      WRITE (IWRITE,920) JULDAT,JULREL,JBEGIN,JULEND,JRANGE,NXLPYR
     $                  ,JSTART,JSTOP,JREND,JYEAR
  920 FORMAT (1H ,'JULDAT=',I10,3X,'JULREL=',I10,3X,'JBEGIN=',I10,3X,
     $       /2X,'JULEND=',I10,3X,'JRANGE=',I10,3X,'NXLPYR=',I10,
     $       /2X,'JSTART=',I10,3X,'JSTOP =',I10,3X,'JREND =',I10,
     $        3X,'JYEAR=',I10)
C
      RETURN
C
C
      END
C
C.....................................................................
C
      SUBROUTINE DATT(IDATE,IMO,IDAY,IYR)
C
C     CONVERT JULIAN DATE TO CALENDER DATE
C
      INTEGER CAL(12,2)
      DATA CAL/0,31,59,90,120,151,181,212,243,273,304,334,
     1  0,31,60,91,121,152,182,213,244,274,305,335 /
      I1=(IDATE-1)/1461
      I2=IDATE-(I1*1461)
C
C       1<=I2<=1461
C
```

```
20400         IF(I2.LE.365) GO TO 10
20410         IF(I2.LE.730) GO TO 20
20420         IF(I2.LE.1095) GO TO 30
20430         I3=3
20440         I4=I2-1095
20450         GO TO 40
20460      10 I3=0
20470         I4=I2
20480         I4=I2-365
20490         GO TO 40
20500      20 I3=1
20510         GO TO 40
20520      30 I3=2
20530         I4=I2-730
20540      40 IYR=1900+I3+(4*I1)
20550         INDX=1
20560         IF(I3.EQ.3)INDX=2
20570         DO 100 I=2,12
20580         IF(I4.LE.CAL(I,INDX)) GO TO 200
20590     100  CONTINUE
20600          IMO=12
20610         IDAY=I4-CAL(12,INDX)
20620          RETURN
20630     200 IMO=I-1
20640         IDAY=I4-CAL(I-1,INDX)
20650         RETURN
20660         END
20670   C
20680   C
20690   C.....................................................................
20700   C
20710         SUBROUTINE JULIAN(MO,DA,YR,ANS)
20720         INTEGER ANS,CAL(12),DA,YR
20730         DATA CAL /31,28,31,30,31,30,31,31,30,31,30,31/
20740   C
20750   C        COMPUTE JULIAN DATE FROM JAN. 1, 1973
20760   C
20770         ANS=0
20780         I=YR-1900
20790         ANS=ANS+365*I
20800         CAL(2)=28
20810         IF(MOD(YR,4).EQ.0) CAL(2)=29
20820         J=MO-1
20830         IF(J.EQ.0) GO TO 20
20840         DO 10 I=1,J
20850         ANS=ANS+CAL(I)
20860      10 CONTINUE
20870      20 CONTINUE
20880         ANS=ANS+DA
20890         RETURN
20900         END
```

```
20910   C
20920   C.....................................................................
20930   C
20940         FUNCTION DMS(A)
20950   C
20960   C     FUNCTION DMS CONVERTS ANGLES EXPRESSED IN
20970   C     DEGREES, MINUTES AND SECONDS TO RADIANS
20980   C
20990         DIMENSION A(3)
21000         REAL MINUTE
21010   C
21020   C
21030         DEGREE = A(1)
21040         MINUTE = A(2)
21050         SECOND = A(3)
21060   C
21070         DMS = DEGREE*3.14159/180. + MINUTE*3.14159/180./60.
21080        1      + SECOND*3.14159/180./60./60.
21090   C
21100         RETURN
21110         END
21120   C
21130   C.....................................................................
21140   C
21150         FUNCTION TAU(ST)
21160   C
21170         COMMON /ORBIT/ PHI.THETAS,THETAL,EP,ET,W
21180         COMMON /IO/ IREAD, IWRITE, IWBUG
21190         COMMON /DBUG/ NBUG.DEBUG
21200         DOUBLE PRECISION    ITAU,DEBUG(1)
21210         DATA ITAU /'TAU'/
21220   C
21230   C
21240   C     THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
21250   C              75TH MERIDIAN FOR EASTERN STANDARD TIME
21260   C              90TH MERIDIAN FOR CENTRAL STANDARD TIME
21270   C              105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
21280   C              120TH MERIDIAN FOR PACIFIC STANDARD TIME
21290   C     THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
21300   C     TAU    = LOCAL HOUR ANGLE
21310   C     ST     = STANDARD TIME IN THE TIME ZONE OF THE
21320   C              OBSERVER IN HOURS COUNTED FROM
21330   C              MIDNIGHT (EG. 0.00 TO 24.00)
21340   C     EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
21350   C     ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME
21360   C              AND MEAN SOLAR TIME (USUALLY NEGLECTED
21370   C              FOR HEAT TRANSFER COMPUTATIONS)
21380   C
21390   C
21400   C     FUNCTION SUBROUTINE TAU CONVERTS THE OBSERVERS
21410   C     STANDARD TIME TO LOCAL HOUR ANGLE IN RADIANS
```

```
21420   C
21430   C     OBTAIN TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
21440   C     OBSERVERS MERIDIAN (HOURS)
21450   C
21460         DTSL = EP*(THETAS - THETAL)* 12.0/3.14159
21470   C
21480   C     COMPUTE OBSERVERS HOUR ANGLE (RADIANS). E = +1 FOR
21490   C     MORNING AND E = -1 FOR AFTERNOON (I.E. SOLARNOON)
21500   C
21510         IF (ST.GT.12..+ DTSL -ET) E = -1.0000
21520         IF (ST.LE.12. + DTSL -ET) E = +1.0000
21530   C
21540         TAU = (ST + E*12. - DTSL + ET) * 3.14159/12.0
21550   C
21560         IF (TAU.GT.6.283185) TAU = TAU - 6.283185
21570         IF (TAU.LT.0.0) TAU = TAU + 6.283185
21580   C
21590   C
21600   C     DEBUG OPTION
21610   C
21620         IF (NBUG.EQ.0) GO TO 100
21630         DO 200 I = 1,NBUG
21640   C
21650         IF (DEBUG(I).NE.ITAU) GO TO 200
21660         WRITE (IWBUG,250)    ST,PHI,THETAS,THETAL,EP,ET,W,DTSL,TAU
21670     250 FORMAT (////1H ,'FUNCTION TAU',  2X,
21680        1 'ST =',F6.3,2X,'PHI =',F6.3,2X,'THETAS =',F6.3,2X,'THETAL ='
21690        2 F6.3,2X,'EP =',F6.3,2X,'ET =',F6.3,2X,'W =',F6.3,2X,
21700        3 'DTSL =',F6.3,2X,'TAU =',F6.3)
21710     200 CONTINUE
21720   C
21730     100 CONTINUE
21740   C
21750         RETURN
21760         END
21770   C
21780   C.....................................................................
21790   C
21800         SUBROUTINE DECL (RJD,DELTA,SR,SS)
21810         INTEGER RJD
21820         COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
21830         COMMON /IO/ IREAD, IWRITE, IWBUG
21840         COMMON /DBUG/ NBUG,DEBUG
21850         DOUBLE PRECISION    IDECL,DEBUG(1)
21860         DATA IDECL/'DECL'/
21870   C
21880   C     DELTA  = DECLINATION OF THE SUN (RADIANS)
21890   C     PHI    = OBSERVERS LATITUDE (RADIANS)
21900   C     THETAS = LONGITUDE OF STANDARD MERIDIAN (RADIANS)
21910   C              75TH MERIDIAN FOR EASTERN STANDARD TIME
21920   C              90TH MERIDIAN FOR CENTRAL STANDARD TIME
```

```
C             105TH MERIDIAN FOR MOUNTAIN STANDARD TIME
C             120TH MERIDIAN FOR PACIFIC STANDARD TIME
C     THETAL = LONGITUDE OF OBSERVERS MERIDIAN (RADIANS)
C     RJD    = RELATIVE JULIAN DATE (I.E. WITH RESPECT TO JAN 1)
C     ST     = STANDARD TIME IN THE TIME ZONE OF THE OBSERVER
C              IN HOURS COUNTED FROM MIDNIGHT (E.G.0.00 TO 24.00)
C     EP     = +1 FOR EAST LONGITUDE, -1 FOR WEST LONGITUDE
C     ET     = DIFFERENCE BETWEEN TRUE SOLAR TIME AND
C              MEAN SOLAR TIME (USUALLY NEGLECTED FOR
C              HEAT TRANSFER COMPUTATIONS)
C
C COMPUTE TIME DIFFERENCE BETWEEN STANDARD MERIDIAN AND
C OBSERVERS MERIDIAN (HOURS)
C
      DTSL   = EP*(THETAS - THETAL)* 3.81972
C
C COMPUTE DECLINATION OF THE SUN (RADIANS)
C
      DELTA  = 0.4093*COS(0.0172*(172. - FLOAT(RJD)) )
C
C COMPUTE HOUR ANGLE AT SUNSET (RADIANS)
C
C
      TSS    = ACOS(-TAN(DELTA)*TAN(PHI))
C     COMPUTE STANDARD TIME OF SUNST (HOURS)
C
      SS     = TSS*3.81972 + 12. +DTSL -ET
C
C COMPUTE HOUR ANGLE OF SUNRISE (RADIANS)
C
      TSR    = 6.283185 - TSS
C
C     COMPUTE  STANDARD TIME OF SUNRISE (HOUR)
C
      SR     = TSR*3.81972 -12. + DTSL -ET
C
C
C     CONVERT SUNRISE IN STANDARD TIME TO LOCAL TIME
C
      SR = SR - DTSL
C
C     CONVERT SUNSET IN STANDARD TIME TO LOCAL TIME
C
      SS = SS - DTSL
C
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 300
      DO 100 I = 1,NBUG
C
```

```
22440         IF (DEBUG(I).NE.IDECL) GO TO 100
22450         WRITE (IWBUG,200)  RJD,DTSL,DELTA,TSS,SS,TSR,SR
22460       200 FORMAT (////,1H ,'SUBROUTINE DECL ','***',' RJD =',
22470          1 I5,'  DTSL =',F6.3,'  DELTA =',F6.3,'  TSS =',F6.3,
22480          2 '  SS =',F6.3,2X,'TSR =',F6.3,2X,'SR = ',F6.3)
22490       100 CONTINUE
22500     C
22510       300 CONTINUE
22520     C
22530         RETURN
22540         END
22550     C
22560     C.....................................................................
22570     C
22580         SUBROUTINE RAINST (TB,TR,D,JSINCE,STORM,JHREOS,JHNEXT)
22590     C
22600     C
22610     C  ROUTINE TO INITIALIZE THE RAINFALL MODEL. THIS ROUTINE INSURES
22620     C  THAT THE BEGINNING OF THE MONTH OCCURS RANDOMLY DURING EITHER
22630     C  AN INTRA- OR AN INTER-STORM PERIOD ACCORDING TO THE APPROPRIATE
22640     C  PROBABILITY DISTRIBUTION.
22650     C
22660          COMMON /RAINS/  TBMEAN, TRMEAN, DMEAN
22670         COMMON /RAIN1/  ITR,ITB
22680          DATA   ON/'ON'/, OFF/'OFF'/
22690     C
22700     C
22710          TSUM = 0.0
22720          DEBUG = OFF
22730     C
22740     C  GENERATE THE TIME SINCE THE LAST STORM.
22750     C
22760          CALL EXPO (TBMEAN,TSINCE)
22770     C
22780     C  NOW BEGIN TO GENERATE A SEQUENCE OF STORMS THAT WILL BRING US UP
22790     C  TO THE BEGINNING OF THE MONTH.
22800     C
22810       100  CALL EXPO (TBMEAN,TB)
22820          TSUM = TSUM + TB
22830     C
22840     C  ARE WE UP TO THE STARTING POINT YET?
22850     C
22860          IF (TSUM .GE. TSINCE) GO TO 200
22870     C
22880     C  IF NOT, GENERATE A STORM DURATION.
22890     C
22900          CALL EXPO (TRMEAN,TR)
22910          TSUM = TSUM + TR
22920     C
22930     C  ARE WE UP TO THE STARTING POINT YET?  IF NOT, GO BACK AND
22940     C  GENERATE THE NEXT INTERSTORM PERIOD.
```

```
C
      IF (TSUM .LT. TSINCE) GO TO 100
C
C  IN THIS CASE, THE MONTH BEGINS DURING A STORM.  DETERMINE TIME
C  TILL END OF STORM (TTEOS) AND TURN STORM FLAG ON.
C
      TTEOS = TSUM - TSINCE
      STORM = ON
      JSINCE = 0
      TB = 0.0
C
C  COMPUTE THE STORM DEPTH GIVEN STORM DURATION.
C
      ALPHA = TR/TRMEAN
      BETA  = DMEAN
      CALL GAMMAD (ALPHA,BETA,D)
C
C  ADJUST STORM DEPTH TO REFLECT ONLY THE PORTION DURING THE
C  CURRENT MONTH.
C
      D = D*(TTEOS/TR)
      TR = TTEOS
C
C  CONVERT TR TO NEAREST INTEGER VALUE
C
      CALL ROUND (TR,ITR)
      IF (ITR .EQ. 0) ITR = 1
      JHREOS = ITR
C
C
      RETURN
C
C
  200  CONTINUE
C
C  IN THIS CASE, THE MONTH BEGINS DURING AN INTER-STORM PERIOD.
C  DETERMINE TIME TILL NEXT STORM.  TURN STORM FLAG OFF.
C
      TTNEXT = TSUM - TSINCE
      STORM  = OFF
      TR     = 0.0
      D      = 0.0
C
C
C  CLOUD COVER MODEL WILL ALSO NEED THE TIME SINCE THE LATEST STORM
C  ENDED.
C
      TSINCE = TB - TTNEXT
C
C  CONVERT TTNEXT TO NEAREST INTEGER
C
```

```
23460         CALL ROUND (TTNEXT,ITB)
23470         IF (ITB .EQ. 0) ITB = 1
23480         JHNEXT = ITB
23490         CALL ROUND(TSINCE,JSINCE)
23500   C
23510   C
23520   C
23530   C
23540         RETURN
23550         END
23560   C
23570   C.....................................................................
23580   C
23590         SUBROUTINE ROUND (X,IX)
23600   C
23610   C  ROUND IS A ROUTINE THAT CONVERTS A REAL VALUE .X, TO THE NEAREST
23620   C  INTEGER VALUE. IN OTHER WORDS, IX IS ROUNDED UP WHEN NECESSARY.
23630   C
23640         IX = INT(X)
23650         RX = AINT(X)
23660   C
23670   C  CHECK IF X IS NEGATIVE OR POSITIVE.
23680   C
23690         IF (X) 100,200,300
23700     100 IF (ABS(X-RX) .GT. 0.50) IX = IX - 1
23710     200 RETURN
23720     300 IF (ABS(X-RX) .GT. 0.50) IX = IX + 1
23730   C
23740   C
23750         RETURN
23760         END
23770   C
23780   C.....................................................................
23790   C
23800         SUBROUTINE PCPN (ITB,TR,D,STORM,JHOUR,JHREOS,JHNEXT,JSINCE,RAIN)
23810   C
23820   C  PCPN CHECKS TO SEE IF WE ARE CURRENTLY IN A STORM OR BETWEEN
23830   C  STORMS AND COMPUTES THE HOURLY RAINFALL TOTAL ACCORDINGLY.
23840   C  WHEN NECESSARY, PCPN SELECTS NEW TIMES BETWEEN STORMS, STORM
23850   C  DURATIONS, AND STORM DEPTHS.  THE HOURLY COUNTERS ARE ALSO
23860   C  UPDATED FOR TIME TILL NEXT STORM AND TIME TILL END OF CURRENT
23870   C  STORM.
23880   C
23890         COMMON /RAINI/  ITR,ITB
23900         COMMON /RAINS/  TBMEAN,TRMEAN,DMEAN
23910   C
23920         DATA    ON/'ON'/, OFF/'OFF'/
23930   C
23940   C
23950   C  CHECK IF STORM FLAG IS ON OR OFF. IF STORM FLAG IS ON, GO TO THE
23960   C  STORM SECTION.
```

```
C
      IF ( STORM .EQ. ON ) GO TO 200
C
C  STORM FLAG IS OFF.  NOW CHECK IF WE HAVE ENDED THE LATEST INTER-
C  STORM PERIOD.
C
      IF ( JHOUR .GT. JHNEXT ) GO TO 100
C
C  STILL IN BETWEEN STORMS.  THEREFORE SET RAIN = 0.0 AND RETURN.
C  ALSO INCREMENT THE COUNTER FOR TIME SINCE LAST STORM.
       JSINCE = JSINCE + 1
C
      D = 0.0
      RAIN = 0.0
      RETURN
C
C
C
  100 CONTINUE
C
C  GENERATE A NEW STORM. FIRST, TURN STORM FLAG ON. SECOND, SELECT A
C  STORM DURATION. THEN SELECT A STORM DEPTH
C
      STORM = ON
      CALL EXPO (TRMEAN,TR)
      ALPHA = TR/TRMEAN
      BETA  = DMEAN
      CALL GAMMAD (ALPHA,BETA,D)
C
C  CONVERT STORM DURATION TO THE NEAREST INTEGER VALUE.
C
      CALL ROUND (TR,ITR)
C
C  MINIMUM STORM DURATION IS ONE HOUR.
C
      IF (ITR .EQ. 0) ITR = 1
C
C  UPDATE THE TIME TILL END OF STORM.
C
      JHREOS = JHOUR + ITR - 1
      JSINCE = 0
C
C  COMPUTE THE HOURLY RAINFALL DEPTH
C
      RAIN = D/FLOAT(ITR)
C
      RETURN
C
C
C
  200 CONTINUE
```

```
C
C
C  STORM FLAG IS ON. NOW CHECK TO SEE IF THE STORM ENDED.
C
      IF (JHOUR .GT. JHREOS) GO TO 300
C
C  THE STORM IS STILL GOING ON.  THEREFORE, COMPUTE RAIN AND RETURN.
C
      RAIN = D/FLOAT(ITR)
      JSINCE = 0
C
      RETURN
C
C
  300 CONTINUE
C
      STORM = OFF
C
C
C  STORM ENDED. SELECT THE NEXT TIME BETWEEN STORMS.
C
      CALL EXPO (TBMEAN,TB)
C
C  CONVERT TIME BETWEEN STORMS TO NEAREST INTEGER VALUE.
C
      CALL ROUND (TB,ITB)
C
C  MINIMUM TIME BETWEEN STORMS IS ONE HOUR.
C
      IF (ITB .EQ. 0) ITB = 1
C
C  UPDATE THE TIME TILL NEXT STORM.
C
      JHNEXT = JHOUR + ITB - 1
      JSINCE = 1
C
      RAIN = 0.0
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE EXPO (EM,T)
C
      COMMON /SEED/ ISEED
C
C     SUBROUTINE TO GENERATE EXPONENTIALLY DISTRIBUTED RANDOM NUMBERS
C     EM = MEAN OF THE DISTRIBUTION
C     T  = RANDOM VARIABLE
C
```

```
C     GENERATE U(0,1)
C
      IX = ISEED
CXXXXXCALL RANDU (IX,ISEED,R)
      CALL RAND1 (IX,ISEED,R)
C
C     TAKE THE INVERSE OF THE EXPONENTIAL PDF
C
      T = -EM*ALOG(R)
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE GAMMAD (ALPHA,BETA,X)
C
      COMMON /WARN/ IWARN
      COMMON /SEED/ ISEED
      COMMON /IO/     IN,IS,IB
C
      U = 1.0
      X = 0.0
      K = IFIX(ALPHA)
      GAM = ALPHA - FLOAT(K)
C
C WRITE (5,900) U,X,K,GAM,ALPHA,BETA
  900    FORMAT (1H0,'U=',E12.5,2X, 'X=',E12.5,2X, 'K=',I5,2X,
     1              'GAM=',E12.5,2X, 'ALPHA=',E12.5,2X, 'BETA=',
     2               E12.5)
C
C
C
      IF (K.EQ. 0) GO TO 100
C
      DO 50 I = 1,K
      IX = ISEED
CXXXX CALL RANDU (IX,ISEED,R)
      CALL RAND1 (IX,ISEED,R)
      U = R*U
C
C WRITE (5,920) I, R, U
  920    FORMAT (1H0, 'TRACE 1 ',' I= ',I5,2X, 'R= ',E12.5,2X,
     1              'U= ',E12.5)
C
   50 CONTINUE
C
      X = -ALOG(U)
C
      IF (GAM.GE. 0.000001 ) GO TO 100
C
      X = BETA*X
```

```
25500   C
25510   C WRITE (5,930) X
25520     930    FORMAT (1HO,'TRACE 2 ',2X,'X= ',E12.5)
25530   C
25540          RETURN
25550   C
25560     100 CONTINUE
25570          IX = ISEED
25580   CXXXX CALL RANDU (IX,ISEED,R)
25590          CALL RAND1 (IX,ISEED,R)
25600          Z = -ALOG (R)
25610   C
25620   C WRITE (5,940) R,Z
25630     940    FORMAT (1HO,' TRACE 3 ',' R= ',E12.5,'  Z= '.E12.5)
25640   C
25650   C
25660   C
25670          DO 200 J = 1,100
25680          IX = ISEED
25690   CXXXX CALL RANDU (IX,ISEED,U1)
25700          CALL RAND1 (IX,ISEED,U1)
25710          IX = ISEED
25720   CXXXX CALL RANDU (IX,ISEED,U2)
25730          CALL RAND1 (IX,ISEED,U2)
25740   C
25750   C
25760   C      COMPUTE THE VALUES OF EM AND EN
25770   C
25780   C      IF EM AND EN ARE COMPUTED DIRECTLY AS:
25790   C
25800   C         EM = U1**(1.0/GAM)
25810   C         EN = U2**(1.0/(1.0-GAM))
25820   C
25830   C      A MACHINE UNDERFLOW OR OVERFLOW CAN EASILY OCCUR.  THESE
25840   C      CONDITIONS CAN BE ANTICIPATED BY FIRST CALCULATING THE
25850   C      LOG (BASE 10) OF EM AND EN.  THE VALID RANGE OF LOG(EM)
25860   C      AND LOG(EN) IS MACHINE DEPENDENT BUT HAS NEVERTHELESS
25870   C      BEEN SET TO BETWEEN -37.0 AND +37.0 IN THIS PROGRAM.
25880   C      IF A VALUE OF HAS BEEN FOUND BELOW THIS RANGE, A DEFAULT
25890   C      OF LOG(EM OR EN) = -37.0 IS USED. IF A VALUE OHAS BEEN
25900   C      FOUND ABOVE THIS RANGE, THEN LOG (EM OR EN) = +37.0.
25910   C      EM AND EN ARE THEN FOUND BY TAKING THE APPROPRIATE ANTILOGS.
25920   C
25930   C
25940          EML10 = (1.0/GAM)*ALOG10(U1)
25950   C
25960             IF ( EML10 .GE. -37.0 .AND.
25970          1       EML10 .LE. +37.0      ) GO TO 110
25980   C
25990                IWARN = IWARN + 1
26000                IF ( EML10 .GT. +37.0 ) EML10 = +37.0
```

```
26010            IF ( EML10 .LT. -37.0 ) EML10 = -37.0
26020   C
26030     110 EM = 10.0**EML10
26040   C
26050   C
26060         ENL10 = (1.0/(1.0 - GAM))*ALOG10(U2)
26070   C
26080            IF ( ENL10 .GE. -37.0  .AND.
26090        1        ENL10 .LE. +37.0       ) GO TO 120
26100   C
26110               IWARN = IWARN + 1
26120               IF ( ENL10 .GT. +37.0 ) ENL10 = +37.0
26130               IF ( ENL10 .LT. -37.0 ) ENL10 = -37.0
26140   C
26150     120 EN = 10.0**ENL10
26160   C
26170   C
26180   C
26190   C WRITE (5,950) J,U1,U2,EM,EN,EML10,ENL10
26200     950    FORMAT (1HO,'TRACE 4 ',' J= ',I5,' U1= ',E12.5,
26210        1               ' U2= ',E12.5, ' EM= ',E12.5,' EN= ',E12.5,
26220        2               ' EML10=',E12.5,' ENL10=',E12.5)
26230   C
26240            IF (EM +EN .LE. 1.0) GO TO 300
26250     200 CONTINUE
26260   C
26270   C
26280          WRITE (IS,500)
26290     500 FORMAT('   END OF DO LOOP IN BETA SUBSECTION OF GAMMAD ')
26300         STOP
26310   C
26320   C
26330     300 Y = EM/(EM + EN)
26340   C
26350         X = BETA*(X + Y*Z)
26360   C
26370         RETURN
26380         END
26390   C
26400   C.....................................................................
26410   C
26420         SUBROUTINE RANDU (IX,IY,YFL)
26430   C
26440   C     GENERATES A UNIFORM DISTRIBUTION
26450   C
26460         IY = IX*65539
26470         IF (IY) 10,20,20
26480      10 IY = IY + 2147483647 + 1
26490      20 YFL = IY
26500         YFL = YFL*.4656613E-9
26510   C
```

```
      RETURN
C
C
      ENTRY RAND1 (IX,IY,YFL)
C
         YFL = RAND10)
C
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE STAT(X,SUM,SUMSQ,SUM3,XBAR,XVAR,XSKEW,N)
C
C  ROUTINE TO COMPUTE THE FIRST THREE MOMENTS OF INTEREST
C  --- MEAN --- VARIANCE --- SKEW COEFFICIENT -----------
C      XBAR       XVAR            XSKEW
C
          TRACE = 'OFF'
          IF (TRACE .EQ. 'ON') WRITE (5.901)
  901     FORMAT (' STAT1')
C
      SUM = X + SUM
      SUMSQ = X**X + SUMSQ
      SUM3 = X**3.0 + SUM3
C
          IF (TRACE .EQ. 'ON') WRITE (5,902)
  902     FORMAT (' STAT2')
C
C
C  UPDATE THE MEAN AND VARIANCE COMPUTATION
C
      XBAR = SUM/ N
      XVAR = SUMSQ/N - XBAR*XBAR
      IF (XSKEW .LT. -990.0) RETURN
      XM3 = SUM3/N - 3.0*XBAR*SUMSQ/N + 2.0*XBAR**3.0
C
          IF (TRACE .EQ. 'ON') WRITE (5,903)
  903     FORMAT (' STAT3')
C
      IF (N .LE. 2) RETURN
C
C  COMPUTE SKEW COEFFICIENT
C
      IF (XVAR .GT. 0.00001) GO TO 100
      XSKEW = 0.0
      GO TO 999
  100 CONTINUE
C
      FACTOR = FLOAT(N*N)/FLOAT((N-1)*(N-2))
```

```
27030       XSKEW = FACTOR*XM3/(XVAR*SQRT(XVAR))
27040 C
27050           IF (TRACE .EQ. 'ON') WRITE (5,904)
27060   904     FORMAT (' STAT4')
27070           IF (TRACE .EQ. 'ON') WRITE (5,905) N, XSKEW, FACTOR, XVAR
27080   905     FORMAT (I10,3(E12.5,2X))
27090   999 RETURN
27100       END
27110 C
27120 C.....................................................................
27130 C
27140       SUBROUTINE HGRAM (H,IA,X,DT,BASE)
27150 C
27160 C  SUBROUTINE TO UPDATE THE FREQUENCY HISTOGRAMS
27170 C
27180       COMMON /IO/     IN,IS,IB
27190       INTEGER H
27200       DIMENSION H(1A)
27210 C
27220       DO 100 I = 1,1A
27230       IF(X.GT.BASE+I*DT) GO TO 100
27240       H(I) = H(I) + 1
27250       RETURN
27260   100 CONTINUE
27270 C
27280       H(I) = H(I) + 1
27290       AMAX = BASE + IA*DT
27300 C
27310       WRITE (IS,900)AMAX,X
27320   900 FORMAT (1H ,'A VALUE GREATER THAN ',E12.5,'  WAS FOUND. X = ',
27330      1 E12.5)
27340 C
27350       RETURN
27360       END
27370 C
27380 C.....................................................................
27390 C
27400       SUBROUTINE PRINTH (H,NMAX,DT,BASE,TITLE,NDATA)
27410 C
27420 C  PRINT OUT NORMALIZED HISTOGRAMS OF GENERATED DATA
27430 C
27440 C
27450 C
27460       COMMON /IO/     IN,IS,IB
27470 C
27480 C
27490       INTEGER H
27500       DIMENSION H(NMAX)
27510       DIMENSION TITLE(1)
27520       DIMENSION TA(10)
27530 C
```

```
C  NORMALIZE THE HISTOGRAM ELEMENTS
C
C      GO TO 60
       DO 50 I = 1,NMAX
       N = H(I)
        X = 100.0*(FLOAT(N)/FLOAT(NDATA))
       CALL ROUND (X,IH)
C      IF(NMAX.EQ.11)WRITE(5,501)I,NMAX,N,X,IH
  501 FORMAT(I5,1X,I5,1X,I15,1X,E12.5,1X,I15)
       H(I) = IH
    50 CONTINUE
    60 CONTINUE
C
       WRITE (IS,910)
  910 FORMAT (1H1,15(5H      )/1H+,15(5H      ))
       WRITE (IS,900) (TITLE(I),I=1,5)
  900 FORMAT (1H ,14X,'HISTOGRAM OF ',5A5,' (PERCENT)'/)
       DO 100 J = 1,NMAX,10
       IMAX = 10
       IF( J+10 .GT. NMAX ) IMAX = NMAX - J + 1
C
C
        GO TO 199
  196 DO 198 II=1,IMAX
       IZ = II + J - 1
       WRITE (5,197)IZ,II,J,IMAX,NMAX,H(IZ)
  197 FORMAT(5I5,I20)
  198 CONTINUE
  199 CONTINUE
       WRITE (IS,200)(H(I+J-1),I=1,IMAX)
  200 FORMAT (1H ,10(2X,I5))
C
C
       WRITE (IS,300)
  300 FORMAT (1H ,10(7H------- ))
C
C
       DO 350 K = 1,10
  350 TA(K) = (K-1+J)*DT + BASE
C
C
       WRITE (IS,400)(TA(K),K=1,10)
  400 FORMAT(1H ,10(1X,F6.2)//)
C
C
  100 CONTINUE
C
C
       RETURN
       END
C
```

```
C.....................................................................
C
C
      SUBROUTINE SOLRAD (RJD,ST1,ST2,T,TB,NMAX,CCA,CCB,PDF,N,COORD,
     1        SWR,CLD,BETA,GAM,CCBAR,CCSD,RHO,SEASON)
C
C     SUBROUTINE SOLRAD COMPUTES INCIDENT SOLAR
C     RADIATION ON THE GROUND OR ON THE TOP OF A
C     VEGETAL CANOPY DURING A SPECIFIED INTERVAL OF TIME
C
C     ST1    = BEGINNING OF INTERVAL - STANDARD TIME
C     ST2    = END OF INTERVAL - STANDARD TIME
C     CSKY   = CLEAR SKY RADIATION - LANGLY
C     CLD    = CLOUD COVER (0.0 - CLD   - 1.0)
C     SWR    = TOTAL INCIDENT SOLAR RADIATION - LANGLY
C     SR     = SUNRISE
C     SS     = SUNSET
C     T1     = BEGINNING OF INTERVAL OF INTEGRATION - LOCAL HOUR ANGLE
C     T2     = END OF INTERVAL OF INTEGRATION - LOCAL HOUR ANGLE
C     RJD    = RELATIVE JULIAN DATE
C     SIALPH = SIN(ALPHA)
C     PDF    = PROBABILITY DENSITY FUNCTION (DISCRETE)
C              FOR NOISE TERM IN CLOUD COVER MODEL
C     RADTYP = INDICATES IF USER WANTS CLRSKY CALCULATIONS ONLY
C     COORD  = COORDINATES OF THE INTERVALS OF PDF
C
      DOUBLE PRECISION    CLEAR,RADTYP
      DOUBLE PRECISION    ISOLRD,DEBUG(1)
      DIMENSION PDF(1), COORD(1)
      DIMENSION RHO(1),CCBAR(1),CCSD(1),BETA(1),GAM(1)
      INTEGER SEASON(1)
      INTEGER RJD
      COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
      COMMON /RTYPE/ RADTYP
      COMMON /SEED/ ISEED
      COMMON /DBUG/ NBUG,DEBUG
      COMMON /IO/    IN,IS,IB
      DATA ISOLRD /'SOLRAD'/
      DATA CLEAR /'CLEARSKY'/
C
C
C     COMPUTE DECLINATION, SUNRISE AND SUNSET
C
      CALL DECL (RJD,DELTA,SR,SS)
C
C
C     SCREENING TO DETERMINE THE PROPER INTERVAL OF INTEGRATION
C
      IF (ST2.LE.ST1) GO TO 100
      IF (ST1.LE.SR.AND.ST2.LE.SR) GO TO 120
      IF (ST1.LE.SR.AND.ST2.GE.SR.AND.ST2.LE.SS) GO TO 130
```

```
28560         IF (ST1.LE.SR.AND.ST2.GE.SS) GO TO 140
28570         IF (ST1.GE.SR.AND.ST2.LE.SS) GO TO 150
28580         IF (ST1.LE.SS.AND.ST2.GE.SS) GO TO 160
28590         IF (ST1.GE.SS.AND.ST2.GE.SS) GO TO 120
28600   C
28610   C     ST2 IS IN THE AM WHILE ST1 IS STILL IN PM
28620   C
28630     100 CONTINUE
28640         IF (ST2.GT.SR) GO TO 130
28650   C
28660   C     NO SHORTWAVE RADIATION IN THIS INTERVAL
28670   C
28680     120 CSKY = 0.0
28690         T1 = TAU(ST1)
28700         T2 = TAU(ST2)
28710         SIALPH  = 999.
28720         GO TO 800
28730   C
28740   C     PART OF INTERVAL COMES AFTER SUNRISE. SET BEGINNING
28750   C     OF INTERVAL EQUAL TO THE LOCAL HOUR ANGLE OF SUNRISE.
28760   C     THEN CONVERT ENDING TIME TO LOCAL HOUR ANGLE.
28770   C
28780     130 T1 = TAU(SR)
28790         T2 = TAU(ST2)
28800         GO TO 500
28810   C
28820   C     INTEGRATION INTERVAL INCLUDES ENTIRE INTERVAL FROM SUNRISE
28830   C     TO SUNSET
28840   C
28850     140 T1 = TAU(SR)
28860         T2 = TAU(SS)
28870         GO TO 500
28880   C
28890   C     INTEGRATION INTERVAL IS ENTIRELY WITHIN SUNSHINE PERIOD
28900   C
28910     150 T1 = TAU(ST1)
28920         T2 = TAU(ST2)
28930         GO TO 500
28940   C
28950   C     ENDING TIME OCCURS AFTER SUNSET
28960   C
28970     160 T1 = TAU(ST1)
28980         T2 = TAU(SS)
28990   C
29000   C     COMPUTE CLEAR SKY SOLAR RADIATION FOR THE
29010   C     INTERVAL T1 TO T2
29020   C
29030     500 CONTINUE
29040         CALL CLRSKY (RJD,T1,T2,NMAX,CSKY,SIALPH,DELTA)
29050   C
29060   C     DETERMINE CLOUD COVER
```

```
C
  800 CONTINUE
      IF (RADTYP.EQ.CLEAR) GO TO  900
C     GO TO 801
      CALL COVER (RJD,CCA,CCB,PDF,N,COORD,SEASON,TB,T,BETA,GAM,CCBAR,
     1       CCSD,RHO,CLD)
  801 CONTINUE
C
C     COMPUTE CLOUDY SKY SOLAR RADIATION
C
      SWR = CSKY*(1.0 - 0.65*CLD*CLD)
      GO TO 950
C
  900 SWR = CSKY
C
  950 CONTINUE
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 1100
      DO 1000 I = 1,NBUG
C
      IF (DEBUG(I).NE.ISOLRD) GO TO 1000
      WRITE (I8,1050)RJD,ST1,ST2,SR,SS,T1,T2,CSKY,SIALPH,CLD
 1050 FORMAT (///,1H ,'SUBROUTINE SOLRAD',2X,'RJD ='
     1 I4,2X,'ST1 =',F7.3,2X,'ST2 =',F7.3,2X,'SR ='
     2 F7.3,2X,'SS =',F7.3,2X,'T1 =',F7.3,2X,'T2 =',
     3 F7.3/T20,'CSKY =',F12.2,2X,'SIALPH =',F8.3,
     4 2X,'CLD =',F7.3)
 1000 CONTINUE
C
 1100 CONTINUE
      RETURN
      END
C
C.....................................................................
C
C
      SUBROUTINE CLRSKY (RJD,T1,T2,NMAX,CSKY,SIALPH,DELTA)
C
C     SUBROUTINE TO NUMERICALLY INTEGRATE THE
C     EQUATION FOR CLEAR SKY RADIATION. SIMPSONS
C     RULE IS USED.
C
C     DELTA  = DECLINATION OF THE SUN (RADIANS)
C     PHI    = OBSERVERS LATITUDE (RADIANS)
C     EN     = TURBIDITY FACTOR
C            = 2.0 FOR CLEAR MOUNTAIN AIR
C            = 4-5 FOR SMOGGY URBAN AREAS
C     W      = SOLAR CONSTANT = 120. LANGLY/HR
C              W IS READ IN AS A VARIABLE TO ALLOW THE USER  TO CHOOSE
```

```
C              WHICH VALUE OF W IS APPROPRIATE.
C     RJD    = RELATIVE JULIAN DATE
C     T1     = HOUR ANGLE AT BEGINNING OF INTERVAL
C     T2     = HOUR ANGLE AT END OF INTERVAL
C     NMAX   = NUMBER OF SUBINTERVALS = 2,4,6...
C     CSKY   = FINAL VALUE OF F IS CLEAR SKY RADIATION
C     SIALPH = SIN (ALPHA), WHERE ALPHA IS THE ANGLE
C              OF RADIATION WITH THE HORIZONTAL (RADIANS)
C     ALPHA =  ANGLE OF RADIATION (RADIANS)
C
C     REFERENCE FOR SIMPSONS RULE
C           TI PROGRAMMABLE 58/59 MASTER LIBRARY
C           TEXAS INSTRUMENTS INCORPORATED, 1977 P29-31
C
      COMMON /DBUG/ NBUG,DEBUG
      COMMON /ORBIT/ PHI,THETAS,THETAL,EP,ET,W
      COMMON /ATMOS/ EN
      COMMON /IO/    IN,IS,IB
      INTEGER RJD
      DOUBLE PRECISION    ICSKY,DEBUG(1)
      DATA ICSKY /'CLRSKY'/
C
C
C     IS DEBUG REQUESTED FOR SUBROUTINE CLRSKY?
C
      IBUG = 0
      IF (NBUG.EQ.0) GO TO 910
      DO 900 I = 1,NBUG
C
      IF (DEBUG(I).NE.ICSKY) GO TO 900
      IBUG = 1
      GO TO 910
  900 CONTINUE
C
  910 CONTINUE
C
      IF (IBUG.EQ.0) GO TO 10
      WRITE (IB,930) RJD,T1,T2,NMAX
  930 FORMAT (////,1HO,'SUBROUTINE CLRSKY',2X,'RJD =',
     1 I5,2X,'T1 =',F6.3,'T2 =',F6.3,2X,'NMAX =',I5)
C
   10 CONTINUE
C
C
C     DO LOOP PERFORMS INTEGRATION BY SIMPSON'S RULE
C
      X = 0.0
      F=0.000
      IMAX = NMAX + 1
C
C
```

```
30090         D = (T2 - T1)/NMAX
30100         IF (D.GE.0.0) GO TO  70
30110         D = ( 6.28318 - T1 + T2 )/NMAX
30120      70 CONTINUE
30130   C
30140   C
30150         DO 100 NN = 1,IMAX
30160   C
30170         N=NN-1
30180   C
30190   C     COMPUTE CURRENT HOUR ANGLE
30200   C
30210         T = T1 + N*D
30220   C
30230   C     COMPUTE SIN(ALPHA)
30240   C
30250         SIALPH = SIN(DELTA)*SIN(PHI) + COS(DELTA)*COS(PHI)*COS(T)
30260   C
30270   C     CHECK TO PREVENT DIVISION BY ZERO OR USING ZERO
30280   C     AS THE ARGUMENT OF A LOG FUNCTION
30290   C
30300   C
30310   C
30320   C     CONSIDER THE TERM
30330   C
30340   C
30350   C        Y = ( 0.128 - 0.054*ALOG10(1./SIALPH))
30360   C     WHEN ALPHA APPROACHES ZERO, THE DECAY FUNCTION STARTS TO GROW.
30370   C     THIS OCCURS DUE TO POLES THAT EXIST AT THE ENDS OF THE INTERVAL
30380   C     OF INTEGRATION. AN APPROXIMATION TO THE DECAY FUNCTION WAS
30390   C     MADE THAT CONSISTED OF A STRAIGHT LINE EXTRAPOLATION OF THE
30400   C     DECAY FUNCTION FROM ALPHA = 0.016 TO ZERO.
30410   C
30420   C
30430         ALPHA = ASIN(SIALPH)
30440         IF (ALPHA .GT. 0.016) GO TO 40
30450         IF ( ALPHA .LT. 0.0 ) ALPHA = 0.0
30460         IF ( SIALPH .LT. 0.0 ) SIALPH = 0.0
30470   C
30480         X = 1.293454*ALPHA*SIALPH
30490   C
30500         GO TO 45
30510   C
30520      40 CONTINUE
30530   C
30540         X=(EXP(-EN*(0.128 - 0.054*ALOG10(1./SIALPH))/SIALPH))*SIALPH
30550   C
30560      45 CONTINUE
30570   C
30580         IF (MOD(N,2).NE.0) GO TO 200
30590         M=2
```

```
30600         IF (N.EQ.0) M=1
30610         IF (N.EQ.NMAX) M=1
30620         F=F + M*X
30630         GO TO 50
30640   C
30650     200 F=F + 4*X
30660   C
30670   C     DEBUG OPTION
30680   C
30690      50 IF (IBUG.EQ.0) GO TO 100
30700         WRITE (IB,920) ,N,T,SIALPH,X,F
30710     920 FORMAT (1H ,T25,'N =',I4,2X,'T =',F6.3,2X,'SIALPH =', F6.2,
30720        1 2X,'X =',E12.3,2X,'F =',E12.3)
30730   C
30740     100 CONTINUE
30750   C
30760         F = F*D/3.0
30770   C
30780   C     COMPUTE CORRECTION FACTOR FOR ELLIPTICAL ORBIT
30790   C
30800   C
30810         R =  1.000 + 0.017*COS(6.2832*FLOAT(186 - RJD)/365.)
30820   C
30830         CSKY = (12.0*60.0/3.1416)*F*W/(R*R)
30840   C
30850   C     DEBUG OPTION
30860   C
30870         IF (IBUG.EQ.0) GO TO 300
30880         WRITE (IB,940) F,R,W,CSKY
30890     940 FORMAT (1H ,T25,'F =',E12.3,2X,'R =',F6.3,
30900        1 'W =',F8.3,2X,'CSKY =',E12.3)
30910   C
30920     300 CONTINUE
30930         RETURN
30940         END
30950   C
30960   C.....................................................................
30970   C
30980   C
30990         SUBROUTINE COVER (RJD,A,B,PDF,N,COORD,SEASON,TB,T,BETA,GAM,CCBAR,
31000        1CCSD,RHO,CLD)
31010   C
31020         INTEGER  RJD,SEASON(1)
31030         DIMENSION PDF(1),COORD(1)
31040         DIMENSION RHO(1),CCBAR(1),CCSD(1),BETA(1),GAM(1)
31050         COMMON /CLDCOV/ C1
31060         COMMON /LEAP/ LCHECK
31070         COMMON /SEAS/ NSEAS
31080         COMMON /DBUG/ NBUG,DEBUG
31090         COMMON /IO/    IN,IS,IB
31100         COMMON /STORMS/ STORM
```

```
      DOUBLE PRECISION    ICOVER,DEBUG(1)
      DATA ICOVER /'COVER'/
      DATA ON/'ON'/, OFF/'OFF'/
C
C     SEASON = ARRAY CONTAINING RELATIVE JULIAN DATES OF THE FIRST DAY
C              OF EACH SEASON
C     PDF    = DISCRETE PROBABILITY DENSITY FUNCTION OF CLOUD COVER
C     COORD  = COORDINATES OF PDF (I.E. INTERVALS)
C     N      = NUMBER OF INTERVALS IN PDF. DIMENSION OF PDF
C              AND COORD IS N* (NUMBER OF SEASONS OF CLOUD
C              COVER PARAMETERS)
C     ISEAS  = CURRENT SEASON
C     TB     = TIME BETWEEN STORMS (HOURS)
C     T      = TIME SINCE LAST STORM (HOURS)
C     CCBAR  = MEAN CLOUD COVER
C     CCSD   = STANDARD DEVIATION OF CLOUD COVER
C     RHO    = LAG-1 AUTOCORRELATION COEFFICIENT
C     BETA   = TRANSITION DECAY PARAMETER
C     GAM    = TRANSITION DECAY PARAMETER
C     NSEAS  = NUMBER OF SEASONS PER YEAR
C     ARV    = RANDOM VARIATE FOR THE NOISE TERM IN THE CLOUD COVER
C              MODEL
C     C1     = PREVIOUS VALUE OF THE AR(1) PROCESS
C     C2     = CURRENT VALUE OF THE AR(1) PROCESS
C     P      = VALUE OF THE MODULATION FUNCTION
C     CLD    = CLOUD COVER
C
C     DETERMINE THE CURRENT SEASON
C
      IF (NSEAS.GT.1) GO TO 50
         ISEAS = 1
         GO TO 150
C
   50 CONTINUE
      IF (RJD .LT. SEASON(NSEAS+LCHECK*NSEAS)) GO TO 60
         ISEAS = NSEAS
         GO TO 150
   60 CONTINUE
C
C
      DO 100 I=1,NSEAS
      IF (RJD.GE.SEASON(I + LCHECK*NSEAS)) GO TO 100
      ISEAS = I-1
      GO TO 150
  100 CONTINUE
      WRITE (IS,160)
  160 FORMAT (1H1,///,'SEASON SELECTION FAILED IN SUBROUTINE COVER')
      STOP
  150 CONTINUE
C
C
```

```
C
C     COMPUTE STOCHASTIC COMPONENT
C
      ARV = ARVA(PDF,N,A,B,COORD,ISEAS)
C
      C2 = RHO(ISEAS)*C1 + SQRT(1.-RHO(ISEAS)*RHO(ISEAS))*
     1 (ARV - CCBAR(ISEAS))
C    2 + CCBAR(ISEAS)*(1. - RHO(ISEAS))
C
C  CHECK TO SEE IF A STORM IS GOING ON.  IF NO STORM, COMPUTE THE
C  MODULATION FUNCTION.  IF STORM IS ON, SET CLD = 1.0 AND BY-PASS
C  THE MODULATION FUNCTION.
C
      IF (STORM .EQ. OFF) GO TO 200
      CLD = 1.0
      GO TO 300
  200 CONTINUE
C
C     COMPUTE MODULATION FUNCTION
C
      BEXP = BETA(ISEAS)*T
      GEXP = GAM(ISEAS)*(TB-T)
C
C  CHECK TO SEE IF BEXP OR GEXP WILL CAUSE A MACHINE
C  UNDERFLOW WHEN USED AS THE ARGUMENT IN THE EXP FUNCTION.
C
      IF (BEXP .GT. 37.0*ALOG(10.)) BEXP =37.0*ALOG(10.0)
      IF (GEXP .GT. 37.0*ALOG(10.)) GEXP =37.0*ALOG(10.0)
C
      P = (1.0 -EXP(-BEXP))*(1.0 -EXP(-GEXP))
C
C     COMPUTE AVERAGE CLOUD COVER FOR INTERVAL
C
C
      CLD = CCBAR(ISEAS) + (1.0 - CCBAR(ISEAS))*(1.0 -P) + C2*P
      IF (CLD.GT.1.0) CLD = 1.00
      IF (CLD.LT.0.0) CLD = 0.00
  300 CONTINUE
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 910
      DO 900 I = 1,NBUG
      IF (DEBUG(I).NE.ICOVER) GO TO 900
      WRITE (IB,920) RJD,ISEAS,C1,C2,ARV ,P,CLD
  920 FORMAT (///1H ,'SUBROUTINE COVER',2X,'RJD =',I5,2X,
     1 'ISEAS =',I4,2X,'C1 =',F7.3,2X,'C2 =',F7.3,
     2 'ARVA =',F6.3,2X,'P =',F6.3,2X,'CLD =',F5.3)
C
      WRITE (IB,930) BETA(ISEAS),GAM(ISEAS),TB,T
  930 FORMAT (1H ,'BETA= ',E12.5,2X,'GAM= ',E12.5,2X,
```

```
32130       $  'TB= ',E12.5,2X,'T= ',E12.5)
32140     900 CONTINUE
32150 C
32160     910 CONTINUE
32170 C
32180 C     SAVE CURRENT VALUE OF THE STOCHASTIC COMPONENT FOR
32190 C     USE IN THE NEXT TIME PERIOD
32200 C
32210       C1 = C2
32220 C
32230       RETURN
32240       END
32250 C
32260 C.....................................................................
32270 C
32280 C
32290       FUNCTION ARVA (PDF,N,A,B,COORD,ISEAS)
32300 C
32310 C     FUNCTION ARVA SELECTS A RANDOM VARIABLE FROM AN
32320 C     ARBITRARY DISCRETE PROBABILITY MASS FUNCTION
32330 C
32340 C     PDF    = DISCRETE PROBABILITY DENSITY FUNCTION
32350 C     N      = NUMBER OF INTERVALS
32360 C     A      = LOWER LIMIT OF U(A,B)
32370 C     B      = UPPER LIMIT OF U(A,B)
32380 C     ISEED  = SEED FOR RANDU
32390 C     COORD  = CONTAINS COORDINATES OF THE INTERVALS
32400 C              OF PDF. (COORD(I-1).LT.X.AND.X.LE.COORD(I))
32410 C
32420 C
32430       REAL PDF(1), PEAK, COORD(1)
32440       COMMON /SEED/ ISEED
32450       COMMON /DBUG/ NBUG,DEBUG
32460       COMMON /IO/    IN,IS,IB
32470       DOUBLE PRECISION    IARVA,DEBUG(1)
32480       DATA IARVA /'ARVA'/
32490 C
32500 C     FIND THE PEAK OF THE DISTRIBUTION
32510 C
32520       PEAK = 0.0
32530       DO 100 I = 1,N
32540       IF (PDF((ISEAS-1)*N +I).GT.PEAK) PEAK = PDF((ISEAS-1)*N +I)
32550     100 CONTINUE
32560 C
32570 C     SELECT THE FIRST RANDOM NUMBER FROM U(A,B)
32580 C
32590     150 IX = ISEED
32600       CALL RAND1 (IX,ISEED,R)
32610 C     CALL RANDU (IX,ISEED,R)
32620       U1 = A + (B-A) * R
32630 C
```

```
C     FIND WHICH INTERVAL U1 BELONGS TO
C
      DO 200 I = 1,N
      IF (U1 .GT. COORD((ISEAS-1)*N + I )) GO TO 200
      J= (ISEAS-1)*N +I
      GO TO 300
  200 CONTINUE
      WRITE (IS,250)
  250 FORMAT (1H1, ' SUBROUTINE ARVA -- U1 IS GREATER ',
     1'THAN THE MAXIMUM INTERVAL FOR THE DISCRETE PDF')
      STOP
  300 CONTINUE
C
C     CALCULATE THE SELECTION CRITERION
C
      F= PDF(J)/PEAK
      IX = ISEED
      CALL RAND1 (IX,ISEED,U2)
C     CALL RANDU (IX,ISEED,U2)
C
C     DEBUG OPTION
C
      IF (NBUG.EQ.0) GO TO 600
      DO 500 I = 1,NBUG
      IF (DEBUG(1).NE.IARVA) GO TO 500
      WRITE (IB,550) PEAK,U1,U2,F
  550 FORMAT (///1H ,'FUNCTION ARVA',2X,'PEAK =',F5.3,
     1 'U1 =',F6.3,2X,'U2 =',F6.3,2X,'F =',F6.3)
  500 CONTINUE
C
  600 CONTINUE
C
C     ACCEPT OR REJECT U1
C
      IF (U2.GT.F) GO TO 150
      ARVA=U1
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE MARKOV (N,PDF,COORD,A,B,XBAR,XDEV,XRHO,XLAG1,K,X)
C
C  MARKOV IS A GENERAL ROUTINE TO COMPUTE A STOCHASTIC VARIATE
C  GENERATED BY A FIRST ORDER MARKOV PROCESS.
C
      DIMENSION PDF(1),COORD(1),XBAR(1),XDEV(1),XRHO(1)
C
C
      NSEAS = 1
C
```

```
33150         GO TO  (200,300) K
33160   C
33170     200 CONTINUE
33180   C
33190   C  SECTION 1 -- USE THIS SECTION WHEN ARV IS SELECTED FROM
33200   C               AN ARBITRARY PDF WITH MEAN = XBAR AND
33210   C               STANDARD DEVIATION = XDEV
33220   C
33230   C  DETERMINE THE RANDOM VARIATE
33240   C
33250         ARV = ARVA (PDF,N,A,B,COORD,NSEAS)
33260   C
33270         X= XBAR(NSEAS) + XRHO(NSEAS)*(XLAG1 - XBAR(NSEAS)) +
33280        $    SQRT(1.0-XRHO(NSEAS)*XRHO(NSEAS))*(ARV-XBAR(NSEAS))
33290   C
33300   C
33310         GO TO 800
33320     300 CONTINUE
33330   C
33340   C  SECTION 2 -- USE THIS SECTION WHEN ARV IS FRAM A
33350   C               STANDARDIZED NORMAL DISTRIBUTION ( N(0,1) )
33360   C     ARV = ARVA (PDF,N,A,B,COORD,NSEAS)
33370         CALL NORMAL (ARV)
33380         X= XBAR(NSEAS) + XRHO(NSEAS)*(XLAG1 - XBAR(NSEAS)) +
33390        $    SQRT(1.0-XRHO(NSEAS)*XRHO(NSEAS))*(ARV*XDEV(NSEAS))
33400   C
33410   C
33420     800 CONTINUE
33430   C
33440   C
33450   C
33460   C     XLAG1 COULD BE SET EQUAL TO X AT THIS POINT OR CHECKED FOR
33470   C     NEGATIVE NUMBERS.  HOWEVER THE NATURE OF THESE CHECKS DEPENDS
33480   C     ON THE VARIATE BEING GENERATED.  THEREFORE, THESE CHECKS ARE
33490   C     MADE IN THE CALLING ROUTINE WHERE THE IDENTITY OF THE VARIATE
33500   C     IS KNOWN ALONG WITH THE PECULIARITIES ASSOCIATED WITH IT.
33510   C
33520         IDEBUG = 0
33530         IF (IDEBUG .EQ.0) RETURN
33540         WRITE (5,100) XBAR(NSEAS),XDEV(NSEAS),XRHO(NSEAS),ARV,XLAG1,X
33550     100 FORMAT (/1X,6(E11.4,1X))
33560         RETURN
33570         END
33580   C
33590   C.....................................................................
33600   C
33610         SUBROUTINE LIMITS (DTSL,R,S,TO,RHO,T12,SIGMA,T23)
33620   C
33630   C     FIND LIMITS FOR TEMPERATURE INTEGRATION
33640   C
33650   C
```

```
      T0  = - DTSL
      T23 = 23.00 - DTSL
C
      IF (DTSL.LT.0.0) GO TO 50
C
C     FIND LIMITS OF INTEGRATION WHEN OBSERVER IS
C     WEST OF THE STANDARD MERIDIAN
C
C     FOR SUNRISE
C
      RHO = AINT(R+1.0) - DTSL
      IF (RHO .LT. R) RHO = RHO + 1
C
C     FOR SUNSET
C
      SIGMA = AINT(S+1) - DTSL
      IF (SIGMA .LT. S) SIGMA = SIGMA + 1
C
C     FOR LOCAL NOON
C
      T12 = 13.0 - DTSL
C
      GO TO 75
C
   50 CONTINUE
C
C      FIND LIMITS OF INTEGRATION WHEN OBSERVER
C      IS EAST OF THE STANDARD MERIDIAN
C
      RHO = AINT(R) - DTSL
      IF (RHO .LT. R) RHO = RHO + 1
C
      SIGMA = AINT(S) - DTSL
      IF (SIGMA .LT. S) SIGMA = SIGMA + 1
C
      T12 = 12.0 - DTSL
C
   75 CONTINUE
C
      RETURN
      END
C
C.....
C
      SUBROUTINE TEMPK (DELTA,PHI,B, TPRIME, K0, K1, K2, K3, K4, K5, K6)
C
C     SUBROUTINE TEMPK COMPUTES THE COEFFICIENTS
C     FOR THE TEMPERATURE EQUATION
C
C
C
```

```
34170  C     DELTA  = DECLINATION OF SUN IN RADIANS
34180  C     PHI    = LATITUDE IN RADIANS
34190  C     B      = VECTOR OF REGRESSION COEFFICIENTS
34200  C     TPRIME = YESTERDAY'S TEMPERATURE AT 11 PM
34210  C     K0-K6  = COEFFICIENTS IN TEMPERATURE EQUATION
34220  C     B0-B6  = EQUIVALENCED VARIABLES WITH B VECTOR ELEMENTS
34230  C     P      = CONSTANT = 2*PI/24
34240  C     B2P2   = INTERMEDIATE VARIABLE USED FREQUENTLY
34250  C
34260  C
34270  C
34280        REAL K0,K1,K2,K3,K4,K5,K6
34290        DIMENSION B(1)
34300  C
34310        B0 = B(1)
34320        B1 = B(2)
34330        B2 = B(3)
34340        B3 = B(4)
34350  C
34360        P      = 3.14159/12.0
34370        B2P2   = B1*B1 + P*P
34380  C
34390        K0     = TPRIME
34400  C
34410        K1     = B0/B1
34420  C
34430        K2     = B2*SIN(DELTA)*SIN(PHI)/B1
34440  C
34450        K3     = B1*B2*COS(DELTA)*COS(PHI)/B2P2
34460  C
34470        K4     = P*B2*COS(DELTA)*COS(PHI)/B2P2
34480  C
34490        K5     = P*P*B3*COS(DELTA)*COS(PHI)/B2P2
34500  C
34510        K6     = P*B1*B3*COS(DELTA)*COS(PHI)/B2P2
34520  C
34530        RETURN
34540        END
34550  C
34560  C
34570  C.....................................................................
34580  C
34590        SUBROUTINE TEMPSN ( ST, DTSL, R,S,B, K0,K1,K2,K3,K4,K5,K6,
34600       $                    CLD, KBAR, GTO, WSP, WDR, TMPLAG, THAT, T )
34610  C
34620  C
34630        INTEGER IDBUG
34640        DOUBLE PRECISION DEBUG(1), DTEMPS
34650        REAL K0,K1,K2,K3,K4,K5,K6
34660        REAL I0,I1,I2,I3,I4,I5,I6,I7, KBAR
34670  C
```

```
      DIMENSION B(1)
C
      COMMON /INTEG/ I0,I1,I2,I3,I4,I5,I6,I7
      COMMON /IO/    IREAD, IWRITE, IWBUG
      COMMON /DBUG/  NBUG, DEBUG
      COMMON /SWITCH/ SWICH1, SWICH2
C
      DATA DTEMPS / 'TEMPSN  ' /
C
C
C        STATEMENT FUNCTIONS FOR INTEGRALS I2 AND I3
C
C
      FUNC1(A,B) = K2*(EXP(B1*A) - EXP(B1*B))
      FUNC2(A)   = EXP(B1*A)*(K3*COS(P*A) + K4*SIN(P*A))
      FUNC3(B)   = EXP(B1*B)*(K6*SIN(P*B) - K5*COS(P*B))
C
C
C
C
C        SET DEBUG FLAG
C
            IDBUG = 1
C
C
C
C        SET SWITCHES THAT DETERMINE WHICH PREDICTORES ARE USED
C
            SWICH1 = 1
            SWICH2 = 0
C
C
      B1 = B(2)
      B4 = B(5)
      B5 = B(6)
      B6 = B(7)
      B7 = B(8)
C
C
C8000 WRITE (5,9000) (B(J),J=1,8)
C9000 FORMAT (1H0,'TEMPS B VECTOR',5X, 4(E12.5,5X)/T20,4(E12.5,5X))
C
C
         IF ( SWICH1 .EQ. 0 ) KBAR = 1.000000
C
C
      P = 3.14159/12.00000
C
C
C        CONVERT STANDARD  TIME TO LOCAL TIME
C
```

```
      T = TAU(ST)* 12.0/3.14159   -  12.0
C
      IF ( ST .LT. 0.5 .AND. T .LT. 0.0 ) GO TO 5
      IF ( ST .GT.22.5 .AND. T .GT. 24. ) GO TO 5
      IF ( T .LT. 0.0 ) T = T + 24.00
      IF ( T.GT. 24.00 ) T = T - 24.00
    5 CONTINUE
C
C
C     IN ADDITION TO SUNRISE AND SUNSET DETERMINE THE LIMITS
C     OF THE RANGES OF THE TEMPERATURE EQUATIONS
C
      T0 = -DTSL
      T12 = 12.0 - DTSL
      T23 = 23.0 - DTSL
C
      TP = T0 - 1.0
C
      IF(IDBUG.EQ.0) WRITE (IWRITE,10) T0,R,T12,S,T23,TP
   10 FORMAT (1H ,T40,6(2X,E10.3)/)
C
C
C        THE FORM OF INTEGRALS I1, I4, I5, I6, AND I7 ARE THE SAME
C        FOR ALL TIMES OF THE DAY.  I2 AND I3 WILL VARY IN
C        FORM DEPENDING ON THE TIME OF DAY.
C
C
C        COMPUTE I1, I4, I5, I6, I7
C
C
      I1 = K1*(EXP(B1*T) - EXP(B1*TP))
C
C
         IF ( SWICH1 .EQ. 0 ) GO TO 40
C
C
C
      PP = (1.0-EXP(-B1))*EXP(B1*T)/B1
      QB = 1.579E-8*(1.00+0.17*CLD**2.)*(TMPLAG+460.)**6.
      I4 = B4*QB*PP + I4
C
   40 CONTINUE
C
         IF ( SWICH2 .EQ. 0 ) GO TO 50
C
      I5 = B5*GTO*PP + I5
C
C
   50 CONTINUE
C
      I6 = B6*WSP*PP + I6
```

```
35700         I7 = B7*WDR*PP + I7
35710   C
35720   C
35730   C
35740   C        CALCULATE SUBTOTAL
35750   C
35760         SUBTOT = I1 + I4 + I5 + I6 + I7
35770   C
35780   C
35790   C
35800         IF ( T .GT. R ) GO TO 100
35810   C
35820   C
35830   C******** RANGE 1 -- AFTER MIDNIGHT AND BEFORE SUNRISE
35840   C
35850   C
35860   C
35870         GTT = SUBTOT
35880   C
35890         GO TO 900
35900   C
35910   C
35920     100 IF ( T .GT. R + 1.00 ) GO TO 200
35930   C
35940   C
35950   C******** RANGE 2 --FIRST HOUR OR FRACTION AFTER SUNRISE
35960   C
35970   C
35980   C
35990   C
36000         I2 = FUNC1(T,R) - FUNC2(T) + FUNC2(R)
36010         I2 = I2*KBAR
36020   C
36030         I3 = FUNC3(T) - FUNC3(R)
36040         I3 = KBAR*I3
36050   C
36060         GTT = I2 + I3 + SUBTOT
36070   C
36080         GO TO 900
36090   C
36100   C
36110   C
36120     200 IF( T .GT. T12 ) GO TO 250
36130   C
36140   C
36150   C******** RANGE 3 -- AFTER SUNRISE AND BEFORE NOON
36160   C
36170            Q1 = FUNC1(T,T-1.0)
36180            Q2 = FUNC2(T)
36190            Q3 = FUNC2(T-1.0)
36200   C
```

```
36210         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
36220         I2 = TI2*KBAR + I2
36230   C
36240            IF(IDBUG.EQ.0)WRITE(IWBUG,991)KBAR, I2
36250     991     FORMAT (2X, 'KBAR ', 2(E12.5,3X))
36260   C
36270         TI3 = FUNC3(T) - FUNC3(T-1.0)
36280         I3  = TI3*KBAR + I3
36290   C
36300   C
36310         GTT = I2 + I3 + SUBTOT
36320   C
36330         GO TO 900
36340   C
36350   C
36360   C
36370     250 IF ( T .GT. T12 + 1.0 ) GO TO 300
36380   C
36390   C
36400   C
36410   C********* RANGE 4 -- FIRST HOUR AFTER LOCAL NOON
36420   C
36430         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
36440         I2  = TI2*KBAR + I2
36450   C
36460         TI3 = FUNC3(12.0) - FUNC3(T-1.0)
36470         I3  = TI3*KBAR + I3
36480   C
36490         GTT = I2 + I3 + SUBTOT
36500         GO TO 900
36510   C
36520     300 IF ( T .GT. S ) GO TO 400
36530   C
36540   C
36550   C******** RANGE 5 -- AFTER LOCAL NOON BUT BEFORE SUNSET
36560   C
36570   C
36580         TI2 = FUNC1(T,T-1.0) - FUNC2(T) + FUNC2(T-1.0)
36590         I2  = TI2*KBAR + I2
36600   C
36610   C
36620         GTT = I2 + I3 + SUBTOT
36630   C
36640         GO TO 900
36650   C
36660   C
36670   C
36680     400 IF (T .GT. S+1.0 ) GO TO 500
36690   C
36700   C
36710   C******** RANGE 6 -- FIRST HOUR AFTER SUNSET
```

```
C
C
      TI2 = FUNC1(S,T-1.0) - FUNC2(S) + FUNC2(T-1.0)
      I2  = TI2*KBAR + I2
C
C
      GTT = I2 + I3 + SUBTOT
C
      GO TO 900
C
C
C
  500 CONTINUE
C
C
C
C******** RANGE 7 -- AFTER SUNSET
C
C
C       NOTE: I2 DOES NOT CHANGE WHEN T .GT. S
C             I3 DOES NOT CHANGE WHEN T .GT. 12.0
C       THUS USE THE PREVIOUSLY COMPUTED VALUES FOR I2 AND I3
C       THAT HAVE BEEN STORED IN THE COMMON /INTEG/
C
C
      GTT = I2 + I3 + SUBTOT
C
C
C
  900 CONTINUE
C
C
C       NOW THAT THE FUNCTION GTT HAS BEEN EVALUATED,
C       COMPUTE THE TEMPERATURE AT TIME T.
C
C
        IF(IDBUG.EQ.0)WRITE(5,990) GTT,I1,I2,I3,I4,I5,I6,I7
  990   FORMAT ( 2X, 'GTT ',8(E12.5,3X))
C
C
C
      THAT = K0*EXP(-B1*(T-TP)) + GTT*EXP(-B1*T)
C
C
CB010 WRITE (5,9010) THAT,K0,B1,T,TP,GTT
C9010 FORMAT (1H0, 'TEMPSN', 6(E12.5,5X))
C
C
C
C*************************************************************************
      RETURN
```

```
 5990 CONTINUE
C*********************************************************************
C
C
C        DEBUG INFORMATION FOR TEMPSN
C
C
      WRITE (5,6000)
 6000 FORMAT (1H1/2(1H+,100(1H )/),1H+,25(4H****))
C
      WRITE (5,6005)
 6005 FORMAT (1H ,10(4H****),T44,'DEBUG TEMPSN',T61,10(4H****)//)
C
      WRITE (5,6010) T, TP, R, S
 6010 FORMAT (1H ,'TIME PARAMETERS ',7X,1HT,12X,2HTP,13X,1HR,
     $          13X,1HS/15X,4(3X,F11.4))
C
      WRITE (5,6020)
 6020 FORMAT (//1H ,7X,'K0',12X,'K1',12X,'K2',12X,'K3',12X,
     $        'K4',12X,'K5',12X,'K6')
C
      WRITE (5,6030)K0,K1,K2,K3,K4,K5,K6
 6030 FORMAT (1H ,8(3X,E11.4))
C
      WRITE (5,6040)
 6040 FORMAT (//,1H ,7X,2HB0,12X,2HB1,12X,2HB2,12X,2HB3,12X,2HB4,
     $        12X,2HB5,12X,2HB6,12X,2HB7)
C
      WRITE (5,6030) (B(M),M=1,8)
C
      WRITE (5,6035)
 6035 FORMAT (//1H ,6X,3HCLD,10X,6HTMPLAG,9X,3HWSP,12X,3HWDR)
      WRITE (5,6030) CLD, TMPLAG, WSP, WDR
C
      WRITE (5,6050)
 6050 FORMAT (//1H ,6X,3HGTT,12X,2HI1,12X,2HI2,12X,2HI3,12X,2HI4,
     $          12X,2HI5,12X,2HI6,12X,2HI7)
C
      WRITE (5,6030) GTT,I1,I2,I3,I4,I5,I6,I7
      WRITE (5,6060)
 6060 FORMAT (//1H ,5X,4HTIME,11X,4HTEMP)
C
      WRITE (5,6030) T,THAT
C
      WRITE (5,6070)
 6070 FORMAT (//1H ,25(4H****))
C
C
C
C*********************************************************************
C*********************************************************************
```

```
37740   C
37750   C
37760         RETURN
37770         END
37780   C
37790   C.....................................................................
37800   C
37810         SUBROUTINE LONGWV (TF1,TF2,TDF1,TDF2,CLD1,CLD2,LW)
37820   C
37830   C
37840   C        COMPUTE LONGWAVE RADIATION
37850   C
37860   C        TC.....TEMPERATURE IN DEGREES CELSIUS
37870   C        CLD....CLOUD COVER (0<= CLD >=1)
37880   C        LW.....COMPUTED LONGWAVE RADIATION
37890   C        TDC....DEWPOINT TEMPERATURE IN DEG C
37900   C        VP.....VAPOR PRESSURE IN MILLIBARS
37910   C        SVP....SATURATED VAPOR PRESSURE IN MILLIBARS
37920   C
37930         REAL LW
37940   C
37950         COMMON /VAPORP/ VP
37960   C
37970   C
37980   C
37990   C  CONVERT DEG F TO DEG C
38000   C
38010         TDAVE = (TDF1+TDF2)/2.0
38020         TAVE  = (TF1+TF2)/2.0
38030   C
38040         TC = (TAVE - 32.0)*(5.0/9.0)
38050         TDC = (TDAVE - 32.0)*(5.0/9.0)
38060   C
38070   C
38080   C        CONVERT CELSIUS TO KELVIN
38090   C
38100         TK = TC + 273.16
38110   C
38120   C
38130   C        DEFINE THE VALUE OF THE STEPHEN-BOLTZMAN CONSTANT
38140   C           (CAL/(CM**2 * MIN * K**4))
38150   C
38160         S = 0.826E-10
38170   C
38180   C
38190   C        DEFINE THE VALUE OF ATMOSPHERIC EMISSIVITY
38200   C
38210         CALL VAPOR (TC,TDC,VP,SVP)
38220   C
38230         E = 0.70 + 5.95E-05*VP*EXP(1500./TK)
38240   C
```

```
C
C       COMPUTE LONGWAVE RADIATION
C
      LW = E*S*TK**4.0
C
C
C       ACCOUNT FOR CLOUDINESS
C
      C = ( 1.0 + 0.17*CLD**2.0 )
C
      LW = C*LW
C
C
C  COMPUTE TOTAL LONGWAVE FOR ONE HOUR (IE. 60 MINUTES)
C
      LW =  LW*60.0
C
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE DEWSIM (ACOEF,DEWLAG,TEMP,CLD,WDIR,WSP,DEW)
C
C
C  DEWSIM USES THE FOLLOWING MODEL TO GENERATE DEWPOINTS
C
C  TD(T) = A0 + A1*TD(T-1) + A2*TMP + A3*CLD + A4*WDIR + A5*WSP
C
C
      DIMENSION ACOEF(1)
C
C  GENERATE TODAY'S DEWPOINTS
C
C
C
      DEW = ACOEF(1) +ACOEF(2)*DEWLAG +ACOEF(3)*TEMP +
     $     ACOEF(4)*CLD + ACOEF(5)*WDIR +ACOEF(6)*WSP
C
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE MSTAT (N,A,B,C,D,E,RAWSUM,XXT)
C
C
C       ACCUMULATE RAW SUMS AND RAW SUMS OF SQUARES AND CROSS PRODUCTS
C
```

```
C
      DIMENSION A(N),B(N),C(N),D(N),E(N),RAWSUM(5),XXT(5,5),X(5)
C
C
      DO 100 I = 1,N
C
C       LOAD DATA INTO WORK ARRAY
C
      X(1) = A(I)
      X(2) = B(I)
      X(3) = C(I)
      X(4) = D(I)
      X(5) = E(I)
C
C
C       COMPUTE RAW SUMS
C
      DO 200 J = 1,5
C
         RAWSUM(J) = RAWSUM(J) + X(J)
C
      DO 200 K = 1,5
C
         XXT(K,J) = XXT(K,J) + X(K)*X(J)
C
  200 CONTINUE
C
  100 CONTINUE
C
      RETURN
      END
C
C.....................................................................
C
C
      SUBROUTINE FSTAT (IDIM,RAWSUM,XXT,MEAN,COVMAT,CORMAT,NDATA)
C
C
C       COMPUTE THE MEAN VECTOR, THE COVARIANCE MATRIX, AND THE
C          CORRELATION MATRIX
C
      DIMENSION RAWSUM(IDIM), XXT(IDIM,IDIM), MEAN(IDIM)
      DIMENSION COVMAT(IDIM,IDIM), CORMAT(IDIM,IDIM)
      DIMENSION MMT(5,5)
      REAL MEAN,MMT
C
C
C     COMPUTE MEANS AND AVERAGE CROSS PRODUCTS
C
      DO 100 I = 1,IDIM
        MEAN(I) = RAWSUM(I)/NDATA
```

```
39270         DO 100 J = 1,IDIM
39280           XXT(J,I) = XXT(J,I)/NDATA
39290     100 CONTINUE
39300   C
39310   C
39320   C     MULTIPLY THE MEAN VECTOR BY ITS TRANSPOSE
39330   C
39340         DO 200 I = 1,IDIM
39350         DO 200 J = 1,IDIM
39360           MMT(J,I) = MEAN(J)*MEAN(I)
39370     200 CONTINUE
39380   C
39390   C
39400   C     COMPUTE COVARIANCE MATRIX
39410   C
39420         DO 300 I = 1,IDIM
39430         DO 300 J = 1,IDIM
39440           COVMAT(J,I) = XXT(J,I) - MMT(J,I)
39450     300 CONTINUE
39460   C
39470   C
39480   C     COMPUTE THE CORRELATION MATRIX
39490   C
39500         DO 400 I = 1,IDIM
39510         DO 400 J = 1,IDIM
39520           CORMAT(J,I) = COVMAT(J,I)/SQRT(COVMAT(J,J)*COVMAT(I,I))
39530     400 CONTINUE
39540   C
39550   C
39560         RETURN
39570         END
39580   C
39590   C.....................................................................
39600   C
39610         SUBROUTINE RAWLAG (IDIM,JP,DATA,SUM,SUMSQ,SUM3,XBAR,XVAR,XSKEW,
39620        $                   NR,R)
39630   C
39640   C
39650   C  UPDATE ARRAY FOR AUTOCORRELATION ANALYSIS
39660   C
39670   C  IDIM.....DIMINSION OF DATA ARRAY AND MAX LAG
39680   C  JP.......POINTER FOR CURRENT OR LATEST DATUM
39690   C  DATA.....DATA ARRAY ( A 'CIRCULAR' DATA ARRAY )
39700   C  R........SUM OF SQUARES AND CROSS-PRODUCT ARRAY
39710   C
39720   C
39730         DIMENSION DATA(IDIM), R(IDIM)
39740         COMMON /IO/ IN,IS,IB
39750   C
39760              BUG = 'OFF'
39770         IK = JP
```

```
39780   C
39790   C
39800   C  REFERENCE FOR THE EQUATIONS TO COMPUTE AUTOCORRELATION:
39810   C
39820   C           HAAN, CHARLES T.;STATISTICAL METHODS IN HYDROLOGY,
39830   C           IOWA STATE UNIVERSITY PRESS,1977, PAGE 228, EQ (11.13)
39840   C
39850   C
39860           IF (BUG .EQ. 'ON') WRITE (5,910)
39870     910   FORMAT (' RAWLAG1')
39880   C
39890   C
39900           X = DATA(JP)
39910           CALL STAT (X,SUM,SUMSQ,SUM3,XBAR,XVAR,XSKEW,NR)
39920   C
39930           DO 100 K = 1,IDIM
39940   C
39950           IF (BUG .EQ. 'ON') WRITE (5,920)
39960     920 FORMAT (' RAWLAG2')
39970   C
39980   C
39990                IF (BUG .EQ. 'ON' ) WRITE (IB,900) K,JP,IK
40000   C
40010   C
40020           R(K) = R(K) + DATA(JP)*DATA(IK)
40030   C
40040           IK = IK - 1
40050           IF (IK .LE. 0) IK = IK + IDIM
40060   C
40070     100 CONTINUE
40080   C
40090   C
40100     900 FORMAT (1X,'K= ',I2,2X,'JP=',I2,2X,'IK=',I2)
40110   C
40120   C
40130           RETURN
40140           END
40150   C
40160   C.....................................................................
40170   C
40180           SUBROUTINE AUTOCO (MLAG, RHO, XBAR, XVAR, NN, TITLE)
40190   C
40200   C  DETERMINE THE AURTOCORRELATION FUNCTION.  THE MAXIMUM LAG IS MLAG.
40210   C
40220   C
40230   C  REFERENCE FOR THE EQUATIONS TO COMPUTE AUTOCORRELATION :
40240   C
40250   C           HAAN, CHARLES T.;STATISTICAL METHODS IN HYDROLOGY,
40260   C           IOWA STATE UNIVERSITY PRESS,1977, PAGE 228, EQ (11.13)
40270   C
40280   C
```

```
C  MLAG.....MAXIMUM LAG
C  RHO......RAW DATA IN --- AUTOCORRELATION OUT
C  XBAR.....MEAN OF CURRENT DATA TYPE
C  XVAR.....VARIANCE OF CURRENT DATA TYPE
C  NN.......NUMBER OF DATA POINTS IN MONTH
C
C
      DIMENSION RHO(MLAG)
      DIMENSION TITLE(1)
      COMMON /IO/ IN, IS, IB
C
          BUG = 'OFF'
          IF (BUG .EQ. 'ON') WRITE (IB,9900) (TITLE(M),M=1,5)
          IF (BUG .EQ. 'ON') WRITE (IB,9910) (RHO(K),K=1,MLAG),NN,XBAR,
     $                                                        XVAR
C
      DO 100 K = 1,MLAG
C
      RHO(K) = (RHO(K) - NN*XBAR*XBAR)/((NN-1)*XVAR)
C
  100 CONTINUE
C
          IF (BUG .EQ. 'ON') WRITE (IB,9910) (RHO(K),K=1,MLAG)
C
      WRITE (IS,900)
  900 FORMAT (1H1,15(5H     ))
C
      WRITE (IS,910) (TITLE(M),M=1,5)
  910 FORMAT (1H ,15X, 'AUTOCORRELATION FUNCTION FOR ',5A5/)
C
      WRITE (IS,920) (K,K=0,11),(RHO(K),k=1,12)
  920 FORMAT (7X,'LAG ',12I5/6X,13(5H-----)/7X,'RHO ',12F5.2/)
      WRITE (IS,920) (K,K=12,23),(RHO(K),K=13,24)
C
 9900 FORMAT (1X,16A5)
 9910 FORMAT (1X,4(6F10.2/)/1X,'NN= ',I6,5X,'XBAR= ',F10.2,
     $                                    'XVAR= ',F10.2//)
C
C
      RETURN
      END
C
C.....................................................................
C
      SUBROUTINE MARGAM(N,PDF,COORD,A,B,XBAR,XDEV,XRHO,SKEW,XLAG1,X,ARV)
C
C  GENERATE THE NEXT DATUM OF A FIRST ORDER MARKOV PROCESS WHOSE
C  VARIATES ARE GAMMA DISTRIBUTED.
C
C  N........NUMBER OF ORDINATES IN PDF
C  PDF......ARRAY CONTAINING ELEMENTS OF PROBABILITY DISTRIBUTION
```

```
C          FUNCTION (HISTOGRAM FORM) WHICH IS N(0,1).
C  COORD....COORDINATES OF PDF
C  A........LEFT BOUND OF PDF
C  B........RIGHT BOUND OF PDF
C  XBAR.....PROCESS MEAN
C  XDEV.....PROCESS STANDARD DEVIATION
C  XRHO.....PROCESS LAG-1 AUTOCORRELATION COEEFICIENT
C  XLAG1....PREVIOUS VALUE OF PROCESS
C  SKEW.....GAMMA DISTRIBUTION SKEW COEFFICIENT
C  X........CURRENT VALUE OF THE PROCESS
C
C
C  REFERENCE:
C          HAAN, CHARLES T.;STATISTICAL METHODS IN HYDROLOGY,
C          IOWA STATE UNIVERSITY PRESS, 1977
C
C
      DIMENSION PDF(1),COORD(1),XBAR(1),XDEV(1),XRHO(1),SKEW(1)
C
C  SET NUMBER OF SEASONS TO ONE
C
      NSEAS = 1
C
C  EVALUATE RANDOM COMPONENT DISTRIBUTED ACCORDING TO PDF
C
C  10 ARV = ARVA (PDF,N,A,B,COORD,NSEAS)
   10 CALL NORMAL (ARV)
C
C  TO CONTERACT THE PROBLEM OF SUDDEN SHIFTS IN A GENERATED TIME SERIES
C  WHOSE VARIATE IS SKEWED AND HAS A HIGH (EG. GREATER THAN .8) LAG-1
C  AUTOCORRELATION COEFFICIENT, RESTRICT THE USAGE OF THE TAIL OF THE
C  N(0,1) THAT CAUSES THE PROBLEM.
C
C  BY RESTRICTING EXCURSIONS INTO THE OFFENDING TAIL TO ABSOLUTE VALUES
C  BELOW 2.8. ONLY 0.26 PERCENT OF THE DISTRIBUTION IS RESTRICTED.
C
C     1. IF THE SKEW IS NEGATIVE, RESTRICT THE NEGATIVE TAIL OF N(0,1)
C     2. IF THE SKEW IS POSITIVE, RESTRICT THE POSITIVE TAIL OF N(0,1)
C
      IF (ABS(ARV) .LE. 2.8) GO TO 40
C
      IF (ARV) 20, 40, 30
   20 IF (SKEW(NSEAS)) 10, 40, 40
   30 IF (SKEW(NSEAS)) 40, 40, 10
C
   40 CONTINUE
C
C
C
C  EVALUATE RANDOM COMPONENT
C
```

```
41310         CSE = (1.0 -XRHO(NSEAS)**3.0)*SKEW(NSEAS)/
41320        $          (1.0 - XRHO(NSEAS)**2.0)**1.5
41330   C
41340         E   = (2.0/CSE)*(1.0+CSE*ARV/6.0 - CSE*CSE/36.)**3.0
41350        $          - 2.0/CSE
41360   C
41370   C  GENERATE THE NEXT VALUE OF THE PROCESS
41380   C
41390         X = XBAR(NSEAS) + XRHO(NSEAS)*(XLAG1 - XBAR(NSEAS)) +
41400        $         E*XDEV(NSEAS)*SQRT(1.0-XRHO(NSEAS)*XRHO(NSEAS))
41410   C
41420   C
41430   C     XLAG1 COULD BE SET EQUAL TO X AT THIS POINT OR CHECKED FOR
41440   C     NEGATIVE NUMBERS.  HOWEVER THE NATURE OF THESE CHECKS DEPENDS
41450   C     ON THE VARIATE BEING GENERATED.  THEREFORE, THESE CHECKS ARE
41460   C     MADE IN THE CALLING ROUTINE WHERE THE IDENTITY OF THE VARIATE
41470   C     IS KNOWN ALONG WITH THE PECULIARITIES ASSOCIATED WITH IT.
41480   C
41490         IDEBUG = 0
41500         IF (IDEBUG .EQ.0) RETURN
41510         WRITE (5,100) XBAR(NSEAS),XDEV(NSEAS),XRHO(NSEAS),ARV,XLAG1,X
41520        $              ,SKEW,CSE,E
41530     100 FORMAT (/1X,6(E11.4,1X))
41540         RETURN
41550         END
41560   C
41570   C.....................................................................
41580   C
41590         SUBROUTINE PUNCH (IPUNCH, RAINM, VP, WSPM, SWR, LW, TEMPM)
41600   C
41610   C  CONVERT THE DATA GENERATED BY THE CSCS MODEL TO DATA WITH UNITS
41620   C  COMPATABLE WITH MILLY'S LAND SURFACE MODEL.
41630   C
41640   C*************************************************************************
41650   C******  INPUT VARIABLES   ***********************************************
41660   C*************************************************************************
41670   C
41680   C  IPUNCH ... UNIT NUMBER FOR OUTPUT DATA FILE
41690   C  RAINM  ... PRECIPITATION IN MM/HR
41700   C  VP     ... VAPOR PRESSURE IN MILLIBARS
41710   C  WSPM   ... WIND SPEED IN M/SEC
41720   C  SWR    ... SOLAR RADIATION IN LANGLEYS/HR
41730   C  LW     ... LONGWAVE RADIATION IN LANGLEYS/HR
41740   C  TEMPM  ... TEMPERATURE IN DEG C
41750   C
41760   C*************************************************************************
41770   C******  OUTPUT VARIABLES  ***********************************************
41780   C*************************************************************************
41790   C
41800   C  PRECIP ... PRECIPITATION IN CM/SEC
41810   C  RHOVA  ... WATER VAPOR DENSITY IN GRAMS/CM**3
```

```
41820   C  UA     ... WIND SPEED IN CM/SEC
41830   C  RADS   ... SHORTWAVE RADIATION IN LANGLEY/SEC
41840   C  RADLD  ... LONGWAVE RADIATION IN LANGLEY/SEC
41850   C  TEMPM  ... TEMPERATURE IN DEG C
41860   C
41870   C*************************************************************************
41880   C
41890         REAL LW
41900   C
41910   C
41920   C*****  PRECIPITATION CONVERSION
41930   C       CM/SEC = (MM/HR)*(CM/10MM)*(HR/3600SEC)
41940   C
41950         PRECIP = RAINM/36000.00
41960   C
41970   C
41980   C*****  VAPOR DENSITY CONVERSION
41990   C
42000         RHOVA = (0.622/2.876E+06) * VP / (273.16 + TEMPM)
42010   C
42020   C
42030   C
42040   C*****  WIND SPEED CONVERSION
42050   C       CM/SEC = (M/SEC)*(100CM/M)
42060   C
42070         UA = WSPM*100.00
42080   C
42090   C
42100   C*****  RADIATION CONVERSION
42110   C       LANGLEY/SEC = (LANGLEY/HR)*(HR/3600.)
42120   C
42130         RADS = SWR/3600.
42140         RADLD = LW/3600.
42150   C
42160   C
42170   C*****  DATA OUTPUT SECTION
42180   C
42190   C     WRITE (IPUNCH,900 ) PRECIP, RHOVA, UA, RADS, RADLD, TEMPM
42200         WRITE (IPUNCH,900) RAINM, VP, WSPM, SWR, LW, TEMPM
42210     900 FORMAT (6E10.3)
42220   C
42230   C*************************************************************************
42240   C
42250         RETURN
42260         END
42270   C
42280   C.....................................................................
42290   C
42300         SUBROUTINE PLOT (IM,ID,IH,NMAX,TEMP,DEW,SWR,WRL,CLD,RAIN,
42310        $                 WSP,WDIR,STORM,IPL,TB,TE)
42320   C
```

```
C  DATA PLOTTING SUBROUTINE
C
C  IM.......CURRENT MONTH
C  ID.......CURRENT DAY
C  IH.......CURRENT HOUR
C  NMAX.....MAXIMUM NUMBER OF LINES BEFORE NEW TITLE AND HEADING
C  TEMP.....TEMPERATURE
C  DEW......DEWPOINT TEMPERATURE
C  SWR......SHORT WAVE RADIATION
C  WRL......LONG WAVE RADIATION
C  CLD......CLOUD COVER
C  RAIN.....RAINFALL
C  WSP......WIND SPEED
C  WDIR.....WIND DIRECTION
C  IPL......FLAG FOR WHICH COMBINATION OF DATA IS PLOTTED
C           IPL = 1  TEMP, DEW, SWR, WRL, CLD, AND RAIN PLOTTED
C           IPL = 2  SWR, WRL, CLD, RAIN PLOTTED
C           IPL = 3  WSP, WDIR PLOTTED
C  STORM....ON/OFF FLAG TO DETERMINE IF IT IS RAINING
C  TB.......BEGINNING OF TEMPERATURE RANGE FOR ORDINATE SCALE
C  TE.......END OF TEMPERATURE RAND FOR ORDINATE SCALE
C
C
      DIMENSION SYMBOL(135)
      INTEGER CAL(12)
      INTEGER PPTEMP, PPDEW, PPOFF, PPSWR. PPLWR, PPCLD
C
      COMMON /LINES/  NLINES
      COMMON /IO/     IN,IS,IB
      DATA CAL /31,28,31,30,31,30,31,31,30,31,30,31/
C
C
      IC = 5
C
C
C  SET UP PAGE HEADINGS
C
      IF (NLINES .GT. 0) GO TO 200
C
      WRITE (IC,870)
  870 FORMAT(1H1,110(1H )/1H+ ,110(1H ))
C
      GO TO (100,120,140) IPL
C
  100 CONTINUE
      CALL ROUND (TE,ITE)
      CALL ROUND (TB,ITB)
C
C*************************************************************************
C
C
```

```
C  HEADINGS FOR PLOTTING 6 DATA POINTS
C
      WRITE (IC,880)
  880 FORMAT (1H ,112(1H:))
      WRITE (IC,881)
  881 FORMAT (1H ,35X,' CONSTRAINED STOCHASTIC CLIMATE SIMULATION',
     $                '  (CSCS)  ',8X)
      WRITE (IC,882)
  882 FORMAT (1H ,112(1H:)/)
      WRITE (IC,901)
  901 FORMAT (1H ,T15,'HOURLY TEMPERATURES (DEG C)', T71, 'RADIATION ',
     $       '(LANGLY/HOUR)', T105,'CLOUD (*)')
      WRITE (IC,902)
  902 FORMAT (1H ,T18,'(T = TEMP, D = DEW PT)',T68,'(S = SHORT WAVE, '
     $      ,'L = LONG WAVE)',T104,'***RAIN***')
      INCR = 10
      WRITE (IC,903) (I,I=ITB,ITE,INCR), (I,I=20,80,20)
  903 FORMAT (1H ,T6,5I10,T60,4I10,T103,'0   .5    1')
      WRITE (IC,904)
  904 FORMAT (1H ,'MM/DD:HR.',9(5H----+),'---.',8(5H----+),
     $       4H---.,2(5H----+))
C
      NLINES = NLINES + 8
C
      GO TO 200
C
C*****************************************************************************
C
C
  120 CONTINUE
C
C  WRITE HEADINGS FOR 4 VARIABLE PLOT
C
      WRITE (IC,890)
  890 FORMAT (1H ,63(1H:))
  900 FORMAT (1H ,8X,'  CONCEPTUAL STOCHASTIC CLIMATE SIMULATION'
     $                ,'  (CSCS)  ',8X)
      WRITE (IC,900)
      WRITE (IC,891)
  891 FORMAT (1H ,63(1H:)/)
      WRITE (IC,910)
  910 FORMAT (1H .T18,'RADIATION (LANGLY/HOUR)', T55,'CLOUD (*)')
      WRITE (IC,911)
  911 FORMAT(1H ,T15,'(S = SHORTWAVE, L = LONGWAVE)',T54,
     '       '*** RAIN **')
      WRITE (IC,912) (I,I=10,70,20)
  912 FORMAT (1H ,T6,4I10,T54,'0   .5    1')
      WRITE (IC,913)
  913 FORMAT (1H ,'MM/DD:HR.',8(5H----+),4H---.,2(5H----+))
C
      NLINES = NLINES + 8
```

```
43350   C
43360         GO TO 200
43370   C
43380   C*************************************************************************
43390   C
43400     140 CONTINUE
43410   C
43420   C  RESERVED FOR HEADINGS FOR WIND AND WIND DIRECTION
43430   C
43440   C
43450   C*************************************************************************
43460   C
43470     200 CONTINUE
43480   C
43490   C  DETERMINE PLOTTING POSITIONS
43500   C
43510   C  INITIALIZE THE PLOTTING POINT OFFSET
43520   C
43530         PPOFF = 1
43540         IRMIN = 1
43550   C
43560         GO TO (210,220,230) IPL
43570   C
43580   C
43590   C  DETERMINE PLOTTING POSITIONS
43600   C
43610     210 TT = TEMP
43620         CALL ROUND (TT,IT)
43630         DD = DEW
43640         CALL ROUND (DD,IDW)
43650   C
43660   C  ADD PLOTTING POSITION OFFSET
43670   C
43680   C  FIRST, CONVERT TB TO UNITS OF 2 DEGREES
43690   C
43700         TB1 = TB
43710         CALL ROUND (TB1,ITB1)
43720   C
43730   C  ACCOUNT FOR THE OFFSET FROM THE LEFT SIDE OF THE GRAPH TO TB.
43740   C
43750         ITB0 = ITB1 - 5
43760   C
43770   C  ADD PLOTTING POSITION OFFSET
43780   C
43790         PPTEMP = IT - ITB0
43800         PPDEW  = IDW - ITB0
43810         PPOFF  = 49 + PPOFF
43820   C
43830         IRMIN =   PPOFF
43840   C
43850         IF (PPTEMP .GT. PPOFF) PPTEMP = PPOFF
```

```
      IF (PPDEW  .GT. PPOFF) PPDEW  = PPOFF
C
  220 CONTINUE
C
      SW = SWR/2.0
      CALL ROUND (SW,ISW)
      WL = WRL/2.0
      CALL ROUND (WL,IWL)
C
      PPSWR = ISW + PPOFF
      PPLWR = IWL + PPOFF
      PPOFF =  44 + PPOFF
C
      IF (PPSWR .GT. PPOFF) PPSWR = PPOFF
      IF (PPLWR .GT. PPOFF) PPLWR = PPOFF
C
      CLDY = CLD*10.
      CALL ROUND (CLDY,ICLD)
C
      IF (ICLD .LT. 0) ICLD = 0
      IF (ICLD .GT. 10) ICLD = 10
C
      PPCLD = ICLD + PPOFF
      PPOFF = PPOFF + 10
C
      IF (PPCLD .GT. PPOFF) PPCLD = PPOFF
      NPMAX = PPOFF
      GO TO 250
C
C
C*****************************************************************************
C
C
  230 CONTINUE
C
C  RESERVED FOR SETTING PLOTTING POSITIONS FOR WIND AND WIND DIRECTION
C
  250 CONTINUE
C
C*****************************************************************************
C
C
C  SET UP SYMBOL ARRAY
C
      DO 300 I = 1,135
      SYMBOL(I) = ' '
      IF (IH .NE. 23) GO TO 300
      IF (NLINES .GE. NMAX-1) GO TO 300
      SYMBOL(I) = '_'
  300 CONTINUE
C
```

```
      GO TO (305,320,600) IPL
C
  305 CONTINUE
      GO TO 315
      IF (IDW .LT. 10) GO TO 315
      DO 310 I = 10,IDW,10
      SYMBOL(I) = '!'
  310 CONTINUE
  315 CONTINUE
C
      DO 330 I = PPDEW,PPTEMP
      SYMBOL(I) = '-'
  330 CONTINUE
C
      SYMBOL(PPDEW) = 'D'
      SYMBOL(PPTEMP)= 'T'
  320 CONTINUE
C
C
      DO 360 I = IRMIN,PPSWR
      SYMBOL(I) = '^'
  360 CONTINUE
      SYMBOL(PPSWR) = 'S'
      SYMBOL(PPLWR) = 'L'
C
      SYMBOL(NPMAX) = '1'
      SYMBOL(NPMAX-10) = 'I'
      SYMBOL(PPCLD) = '*'
      SYMBOL(1)     = 'I'
      SYMBOL(NPMAX-54) = 'I'
C
C*************************************************************************
C
C
      IF (STORM .EQ. 'OFF') GO TO 370
C
      NPMAX = NPMAX - 10
C
C  PLOT DATA IF STORM IS 'ON'
C
      GO TO (400,410) IPL
C
  400 WRITE (IC,950) IM,ID,IH,(SYMBOL(I),I=1,NPMAX),RAIN
  950 FORMAT (1H ,I2,'/',I2,':',I2,94A1,'***',F4.1,1X,'**')
      GO TO 500
C
  410 WRITE (IC,960) IM,ID,IH,(SYMBOL(I),I=1,NPMAX),RAIN
  960 FORMAT (1H ,I2,'/',I2,':',I2,45A1,'***',F4.1,1X,'**')
      GO TO 500
C
  370 CONTINUE
```

```
44880   C
44890         WRITE (IC,970) IM,ID,IH,(SYMBOL(I),I=1,NPMAX)
44900     970 FORMAT (1H ,I2,'/',I2,':',I2,120A1)
44910   C
44920     500 CONTINUE
44930   C
44940         NLINES = NLINES + 1
44950         IF (ID.EQ.CAL(IM) .AND. IH.EQ.23) NLINES = 0
44960         IF (NLINES .GE. NMAX) NLINES = 0
44970   C
44980         IF (NLINES .GT. 0) GO TO 600
44990         GO TO (560,570) IPL
45000     560 WRITE (IC,904)
45010         WRITE (IC,903) (I,I=ITB,ITE,INCR), (I,I=20,80,20)
45020         GO TO 600
45030     570 WRITE (IC,913)
45040         WRITE (IC,912) (I,I=10,70,20)
45050     600 CONTINUE
45060         RETURN
45070         END
```

REFERENCES

1. Curtis, David C. and P. S. Eagleson, "Constrained Stochastic Climate Simulation," MIT Department of Civil Engineering, R. M. Parsons Laboratory, Report No. 274, May 1982.

2. Milly, P. C. D. and P. S. Eagleson, "Parameterization of Moisture and Heat Fluxes Across the Land Surface for Use in Atmospheric General Circulation Models," MIT Department of Civil Engineering, R. M. Parsons Laboratory, Report No. 279, June 1982.