Degree-Day Computer Routines

```
      SUBROUTINE DACALC (LTHRES,CTHRES,CM,COFF,CI,MAX,MIN,
     *                   NDAYS,DDAYS,ACCUM,ERROR)
C
C     PURPOSE:
C     CALCULATE HEAT DEGREE-DAYS FOR NDAYS FROM MAX & MIN TEMPERATURES.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  DACALC.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  DOSINE, DOTRIA, DOVRCT, HUBERM, SISINE, SITRIA, VERTCUT
C
C    ************************************************************
C
```



```
C
      LOGICAL*2 ABOVE                  ! TRUE FOR HEAT UNITS
C
      INTEGER*2 CI,                    ! COMPUTATION INTERVAL
     *          CM,                    ! COMPUTATION METHOD
     *          COFF,                  ! CUTOFF METHOD
     *          ERROR                  ! RETURNED ERROR SIGNAL
C
      REAL*4 ACCUM(1001),              ! ACCUMULATED DEGREE-DAYS
     *       CTHRES,                   ! CUTOFF THRESHOLD
     *       DDAYS(1000),              ! DEGREE-DAYS RETURNED
     *       LTHRES,                   ! LOWER THRESHOLD
     *       MAX(1001),                ! MAXIMUM TEMPERATURES
     *       MIN(1001),                ! MINIMUM TEMPERATURES
     *       THRESH(2),                ! BOTH THRESHOLDS FOR VERTICAL CUTOFF
     *       TEMP1,                    ! DD ABOVE CUTOFF THRESHOLD
     *       TEMP2                     ! ADDITIONAL DD FOR VERTICAL CUTOFF
C
C                  INTIALIZATIONS
C
      ABOVE = .TRUE.
      ERROR = 0
      IF (COFF.GE.2) THEN
         IF (CM.LE.2) THEN
            J = 1
         ELSE ! CM.LE.4
            J = 2
         END IF
      END IF
      THRESH(1) = LTHRES
      THRESH(2) = CTHRES
C
C                  COMPUTE DEGREE-DAYS
C
      IF (CM.EQ.1) THEN             ! SINGLE SINE
```

```
      DO 100 I = 1,NDAYS
         CALL SISINE (ABOVE,MIN(I),MAX(I+1),LTHRES,DDAYS(I),ERROR)
         IF (ERROR.NE.0) RETURN
         IF (COFF.GT.0) THEN        ! CUTOFF THRESHOLD ACTIVE
            CALL SISINE (ABOVE,MIN(I),MAX(I+1),CTHRES,TEMP1,ERROR)
            DDAYS(I) = DDAYS(I) - TEMP1  ! REMOVE DD ABOVE CUTOFF
            IF (COFF.GE.2) THEN          ! CHECK FOR INT. OR VERT. CUTOFF
               CALL VERTCUT (ABOVE,J,MIN(I),MAX(I+1),THRESH,TEMP2)
               IF (TEMP1.GT.TEMP2) TEMP1 = TEMP2   ! IC MUST BE <= VC
               IF (COFF.EQ.3) TEMP1 = TEMP2        ! VERTICAL CUTOFF
               DDAYS(I) = DDAYS(I) - TEMP1
            END IF
            IF (DDAYS(I).LT.0.0) DDAYS(I) = 0.0
         END IF
         ACCUM(I+1) = ACCUM(I) + DDAYS(I)
100   CONTINUE
   ELSE IF (CM.EQ.2) THEN        ! DOUBLE SINE
      DO 200 I = 1,NDAYS
         CALL DOSINE (ABOVE,CI,MAX(I),MIN(I),LTHRES,DDAYS(I),ERROR)
         IF (ERROR.NE.0) RETURN
         IF (COFF.GT.0) THEN        ! CUTOFF THRESHOLD ACTIVE
            CALL DOSINE (ABOVE,CI,MAX(I),MIN(I),CTHRES,TEMP1,ERROR)
            DDAYS(I) = DDAYS(I) - TEMP1  ! REMOVE DD ABOVE CUTOFF
            IF (COFF.GE.2) THEN          ! CHECK FOR INT. OR VERT. CUTOFF
               CALL DOVRCT (ABOVE,J,CI,MAX(I),MIN(I),THRESH,TEMP2)
               IF (TEMP1.GT.TEMP2) TEMP1 = TEMP2   ! IC MUST BE <= VC
               IF (COFF.EQ.3) TEMP1 = TEMP2        ! VERTICAL CUTOFF
               DDAYS(I) = DDAYS(I) - TEMP1
            END IF
            IF (DDAYS(I).LT.0.0) DDAYS(I) = 0.0
         END IF
         ACCUM(I+1) = ACCUM(I) + DDAYS(I)
200   CONTINUE
   ELSE IF (CM.EQ.3) THEN        ! SINGLE TRIANGLE
      DO 300 I = 1,NDAYS
         CALL SITRIA (ABOVE,MIN(I),MAX(I+1),LTHRES,DDAYS(I),ERROR)
         IF (ERROR.NE.0) RETURN
         IF (COFF.GT.0) THEN        ! CUTOFF THRESHOLD ACTIVE
            CALL SITRIA (ABOVE,MIN(I),MAX(I+1),CTHRES,TEMP1,ERROR)
            DDAYS(I) = DDAYS(I) - TEMP1  ! REMOVE DD ABOVE CUTOFF
            IF (COFF.GE.2) THEN          ! CHECK FOR INT. OR VERT. CUTOFF
               CALL VERTCUT (ABOVE,J,MIN(I),MAX(I+1),THRESH,TEMP2)
               IF (TEMP1.GT.TEMP2) TEMP1 = TEMP2   ! IC MUST BE <= VC
               IF (COFF.EQ.3) TEMP1 = TEMP2        ! VERTICAL CUTOFF
               DDAYS(I) = DDAYS(I) - TEMP1
            END IF
            IF (DDAYS(I).LT.0.0) DDAYS(I) = 0.0
         END IF
         ACCUM(I+1) = ACCUM(I) + DDAYS(I)
300   CONTINUE
   ELSE IF (CM.EQ.4) THEN        ! DOUBLE TRIANGLE
      DO 400 I = 1,NDAYS
         CALL DOTRIA (ABOVE,CI,MAX(I),MIN(I),LTHRES,DDAYS(I),ERROR)
         IF (ERROR.NE.0) RETURN
         IF (COFF.GT.0) THEN        ! CUTOFF THRESHOLD ACTIVE
            CALL DOTRIA (ABOVE,CI,MAX(I),MIN(I),CTHRES,TEMP1,ERROR)
            DDAYS(I) = DDAYS(I) - TEMP1  ! REMOVE DD ABOVE CUTOFF
            IF (COFF.GE.2) THEN          ! CHECK FOR INT. OR VERT. CUTOFF
               CALL DOVRCT (ABOVE,J,CI,MAX(I),MIN(I),THRESH,TEMP2)
```

```
                  IF (TEMP1.GT.TEMP2) TEMP1 = TEMP2   ! IC MUST BE <= VC
                  IF (COFF.EQ.3) TEMP1 = TEMP2        ! VERTICAL CUTOFF
                  DDAYS(I) = DDAYS(I) - TEMP1
               END IF
               IF (DDAYS(I).LT.0.0) DDAYS(I) = 0.0
            END IF
            ACCUM(I+1) = ACCUM(I) + DDAYS(I)
  400    CONTINUE
      ELSE ! CM.EQ.5             ! HUBER'S METHOD
         DO 500 I = 1,NDAYS
            CALL HUBERM (MIN(I),MAX(I+1),LTHRES,CTHRES,DDAYS(I),ERROR)
            IF (ERROR.NE.0) RETURN
            ACCUM(I+1) = ACCUM(I) + DDAYS(I)
  500    CONTINUE
      END IF
C
      RETURN
      END
C
      SUBROUTINE DOSINE (AB,CI,MAX,MIN,THRESH,DDAY,ERROR)
C
C     PURPOSE:
C      TO CALCULATE DEGREE-DAYS ACCUMULATED ABOVE or BELOW A THRESHOLD
C     USING A SINE WAVE ESTIMATION OF AREA UNDER THE CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  DOSINE.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  SISINE
C
C    ****************************************************************
C

Copyright 1985 - Regents of the University of California.  All rights reserved.

C
      LOGICAL*2 AB                     ! T=ABOVE; F=BELOW
C
      INTEGER*2 CI,                    ! COMPUTATIONAL INTERVAL
     *          ERROR                  !
C
      REAL*4 DDAY,                     ! DEGREE-DAYS
     *       MAX(2),                   ! YESTERDAY'S AND TODAY'S MAXIMUM
     *       MIN(2),                   ! TODAY'S AND TOMORROW'S MINIMUM
     *       THRESH                    ! THRESHOLD
C
C                 COMPUTE DEGREE-DAYS
C
      IF (CI.EQ.1) THEN                ! MINIMUM TO MINIMUM
         CALL SISINE (AB,MIN(1),MAX(2),THRESH,DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         CALL SISINE (AB,MIN(2),MAX(2),THRESH,TEMP,ERROR)
      ELSE                             ! MAXIMUM TO MAXIMUM
         CALL SISINE (AB,MIN(1),MAX(1),THRESH,DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         CALL SISINE (AB,MIN(1),MAX(2),THRESH,TEMP,ERROR)
      END IF
      IF (ERROR.NE.0) RETURN
C
```

```
      DDAY = (DDAY + TEMP) / 2.0
C
      RETURN
      END
C
      SUBROUTINE DOTRIA (AB,CI,MAX,MIN,THRESH,DDAY,ERROR)
C
C     PURPOSE:
C      TO CALCULATE DEGREE-DAYS ACCUMULATED ABOVE or BELOW A THRESHOLD
C     USING A TRIANGULAR ESTIMATION OF AREA UNDER THE CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  DOTRIA.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  SITRIA
C
C    ****************************************************************
C
```



```
C
      LOGICAL*2 AB                    ! T=ABOVE; F=BELOW
C
      INTEGER*2 CI,                   ! COMPUTATIONAL INTERVAL
     *          ERROR                 !
C
      REAL*4 DDAY,                    ! DEGREE-DAYS
     *       MAX(2),                  ! YESTERDAY'S AND TODAY'S MAXIMUM
     *       MIN(2),                  ! TODAY'S AND TOMORROW'S MINIMUM
     *       THRESH                   ! THRESHOLD
C
C                  COMPUTE DEGREE-DAYS
C
      IF (CI.EQ.1) THEN               ! MINIMUM TO MINIMUM
         CALL SITRIA (AB,MIN(1),MAX(2),THRESH,DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         CALL SITRIA (AB,MIN(2),MAX(2),THRESH,TEMP,ERROR)
      ELSE                            ! MAXIMUM TO MAXIMUM
         CALL SITRIA (AB,MIN(1),MAX(1),THRESH,DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         CALL SITRIA (AB,MIN(1),MAX(2),THRESH,TEMP,ERROR)
      END IF
      IF (ERROR.NE.0) RETURN
C
      DDAY = (DDAY + TEMP) / 2.0
C
      RETURN
      END
C
      SUBROUTINE DOVRCT (AB,METHOD,CI,MAX,MIN,THRESH,AREA)
C
C     PURPOSE:
C        PRODUCE VERTICAL CUTOFF AREAS FOR DOUBLE SINE & DOUBLE
C     TRIANGULAR METHODS.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  DOVRCT.FOR
C                  MICROSOFT FORTRAN 5.0
C
```

```
C     CALLS:  VERTCUT
C
C    ******************************************************************
```



```
C
      LOGICAL*2 AB                     ! T=ABOVE; F=BELOW
C
      INTEGER*2 CI,                    ! COMPUTATIONAL INTEVAL
     *          METHOD                 ! COMPUTATIONAL METHOD
C
      REAL*4 AREA,                     !
     *       MAX(2),                   ! YESTERDAY'S & TODAY'S MAX
     *       MIN(2),                   ! TODAY'S & TOMORROW'S MIN
     *       TEMP,                     !
     *       THRESH(2)                 ! THRESHOLDS
C
C                 COMPUTE FOR PEAK : MIN,MAX,MIN
C
      IF (CI.EQ.1) THEN
         CALL VERTCUT (AB,METHOD,MIN(1),MAX(2),THRESH,AREA)
         CALL VERTCUT (AB,METHOD,MIN(2),MAX(2),THRESH,TEMP)
C
C                 COMPUTE FOR TROUGH : MAX,MIN,MAX
C
      ELSE ! CI.EQ.2
         CALL VERTCUT (AB,METHOD,MIN(1),MAX(1),THRESH,AREA)
         CALL VERTCUT (AB,METHOD,MIN(1),MAX(2),THRESH,TEMP)
      END IF
C
C                 AVERAGE AREAS
C
      AREA = (AREA + TEMP) / 2.0
C
      RETURN
      END
C
      SUBROUTINE HEATU (CM,CI,MAX,MIN,THRESH,DDAY,ERROR)
C
C     PURPOSE:
C     CALCULATE DEGREE-DAYS FOR ONE DAY FROM MAX & MIN.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  HEATU.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  DOSINE, DOTRIA, DOVRCT, HUBERM, SISINE, SITRIA, VERTCUT
C
C    ******************************************************************
C
```



```
C
      LOGICAL*2 ABOVE                  ! TRUE FOR HEAT UNITS
C
      INTEGER*2 CI,                    ! COMPUTATION INTERVAL
     *          CM(2),                 ! COMPUTATION METHOD
     *          ERROR                  ! RETURNED ERROR SIGNAL
```

```
C
      REAL*4 DDAY,                      ! DEGREE-DAYS RETURNED
     *       THRESH(2),                 ! THRESHOLDS
     *       MAX(2),                    ! YESTERDAY'S & TODAY'S MAX TEMPS.
     *       MIN(2),                    ! TODAY'S & TOMORROW'S MIN TEMPS.
     *       TEMP1,                     ! DD ABOVE CUTOFF THRESHOLD
     *       TEMP2                      ! ADDITIONAL DD FOR VERTICAL CUTOFF
C
C                  INITIALIZATIONS
C
      ABOVE = .TRUE.
      ERROR = 0
      TEMP1 = 0.0
C
C                  COMPUTE DEGREE-DAYS
C
      IF (CM(1).EQ.1) THEN              ! SINGLE SINE WAVE
         CALL SISINE (ABOVE,MIN(1),MAX(2),THRESH(1),DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         IF (CM(2).GT.0)                ! CUTOFF THRESHOLD ACTIVE
     *      CALL SISINE (ABOVE,MIN(1),MAX(2),THRESH(2),TEMP1,ERROR)
      ELSE IF (CM(1).EQ.2) THEN         ! DOUBLE SINE WAVE
         CALL DOSINE (ABOVE,CI,MAX(1),MIN(1),THRESH(1),DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         IF (CM(2).GT.0)                ! CUTOFF THRESHOLD ACTIVE
     *      CALL DOSINE (ABOVE,CI,MAX(1),MIN(1),THRESH(2),TEMP1,ERROR)
      ELSE IF (CM(1).EQ.3) THEN         ! SINGLE TRIANGULAR
         CALL SITRIA (ABOVE,MIN(1),MAX(2),THRESH(1),DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         IF (CM(2).GT.0)                ! CUTOFF THRESHOLD ACTIVE
     *      CALL SITRIA (ABOVE,MIN(1),MAX(2),THRESH(2),TEMP1,ERROR)
      ELSE IF (CM(1).EQ.4) THEN         ! DOUBLE TRIANGULAR
         CALL DOTRIA (ABOVE,CI,MAX(1),MIN(1),THRESH(1),DDAY,ERROR)
         IF (ERROR.NE.0) RETURN
         IF (CM(2).GT.0)                ! CUTOFF THRESHOLD ACTIVE
     *      CALL DOTRIA (ABOVE,CI,MAX(1),MIN(1),THRESH(2),TEMP1,ERROR)
      ELSE ! CM(1).EQ.5                 ! HUBER'S METHOD
         CALL HUBERM (MIN(1),MAX(2),THRESH(1),THRESH(2),DDAY,ERROR)
      END IF
      IF (ERROR.NE.0) RETURN
      DDAY = DDAY - TEMP1               ! REMOVE DD ABOVE CUTOFF, IF ANY
C
C                  GENERATE INTERMEDIATE OR VERTICAL CUTOFF, IF ANY
C
      IF (CM(2).GE.2) THEN
C
C                  SET METHOD TYPE
C
         IF (CM(1).EQ.1 .OR. CM(1).EQ.2) THEN
            I = 1                                ! SINE METHODS
         ELSE ! CM(1).EQ.3 .OR. CM(1).EQ.4
            I = 2                                ! TRIANGULAR METHODS
         END IF
C
C                  COMPUTE INTERMEDIATE OR VERTICAL CUTOFF
C
         IF (CM(1).EQ.1 .OR. CM(1).EQ.3) THEN    ! SINGLE METHODS
            CALL VERTCUT (ABOVE,I,MIN(1),MAX(2),THRESH,TEMP2)
         ELSE ! CM(1).EQ.2 .OR. CM(1).EQ.4       ! DOUBLE METHODS
```

```
            CALL DOVRCT (ABOVE,I,CI,MAX(1),MIN(1),THRESH,TEMP2)
         END IF
         IF (TEMP1.GT.TEMP2) TEMP1 = TEMP2   ! INTERM. CUTOFF MUST BE <= VERT.
         IF (CM(2).EQ.3) TEMP1 = TEMP2       ! VERTICAL CUTOFF
         DDAY = DDAY - TEMP1
         IF (DDAY.LT.0.0) DDAY = 0.0
      END IF
C
      RETURN
      END
C
      SUBROUTINE HTEMPS (CI,MAX,MIN,HOURLY,ERROR)
C
C     PURPOSE:
C        COMPUTE THE TEMPERATURES FOR EACH HOUR IN A DAY, GIVEN
C     THE MAXIMUM AND MINIMUM TEMPERATURES.
C
C     NOTE: ASSUMES TEMPERATURES ON A DOUBLE SINE CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  HTEMPS.FOR
C                  MICROSOFT FORTRAN 5.0
C
C    ****************************************************************
C
      INTEGER*2 CI,                     ! COMPUTATIONAL INTERVAL
     *          ERROR
C
      REAL*4 A,                         ! AMPLITUDE OF SINE WAVE
     *       B,                         ! AVERAGE TEMPERATURE
     *       C,                         ! PERIOD OF SINE WAVE
     *       HOURLY(24),                ! HOURLY TEMPERATURES
     *       MAX(2),                    ! YESTERDAY'S & TODAY'S MAXIMUMS
     *       MIN(2),                    ! TODAY'S & TOMMORROW'S MINIMUMS
     *       MN,                        ! LOCAL MINIMUM
     *       MX,                        ! LOCAL MAXIMUM
     *       PI,
     *       X                          ! POSITION ON SINE WAVE
C
      PARAMETER  (PI = 3.141592654)
C
C                  SET CHARACTERISTICS FOR EACH HALF DAY
C
      C = PI / 12.0
      ERROR = 0
      K = 0
      IF (CI.EQ.1) THEN
         X = 18.0
      ELSE ! CI.EQ.2
         X = 6.0
      END IF
      DO 120 I = 1,2
         IF (CI.EQ.1) THEN              ! MIN TO MIN
            MN = MIN(I)
            MX = MAX(2)
         ELSE ! CI.EQ.2              ! MAX TO MAX
            MN = MIN(1)
            MX = MAX(I)
         END IF
         IF (MN.GT.MX) THEN
```

```
            ERROR = 1
            RETURN
         END IF
C
         A = (MX - MN) / 2.0
         B = (MX + MN) / 2.0
C
C                  COMPUTE THE HOURLY TEMPERATURES
C
         DO 100 J = 1,12
            K = K + 1
            HOURLY(K) = A * SIN(C*X) + B
            X = X + 1.0
  100    CONTINUE
  120 CONTINUE
C
      RETURN
      END
C
      SUBROUTINE HTEMPST (CI,MAX,MIN,HOURLY,ERROR)
C
C     PURPOSE:
C        COMPUTE THE TEMPERATURES FOR EACH HOUR IN A DAY, GIVEN
C     THE MAXIMUM AND MINIMUM TEMPERATURES.
C
C     NOTE: ASSUMES TEMPERATURES ON A DOUBLE TRIANGLE (SAWTOOTH) CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  HTEMPST.FOR
C                  MICROSOFT FORTRAN 5.0
C
C    ****************************************************************
C
      INTEGER*2 CI,                    ! COMPUTATIONAL INTERVAL
     *          ERROR
C
      REAL*4 HOURLY(24),               ! HOURLY TEMPERATURES
     *       INCREMENT,                ! HOURLY INCREMENT
     *       MAX(2),                   ! YESTERDAY'S & TODAY'S MAXIMUMS
     *       MIN(2),                   ! TODAY'S & TOMMORROW'S MINIMUMS
     *       X                         ! POSITION ON SAWTOOTH
C
C                  INITIALIZATION
C
      ERROR = 0
C
C                  SET CHARACTERISTICS FOR FIRST HALF DAY
C
      IF (CI .EQ. 1) THEN              ! MIN TO MIN
         X = MIN(1)
         INCREMENT = (MAX(2) - MIN(1))/12.0
      ELSE ! CI .EQ. 2              ! MAX TO MAX
         X = MAX(1)
         INCREMENT = (MIN(1) - MAX(1))/12.0
      END IF
C
C                  COMPUTE THE HOURLY TEMPERATURES
C
      DO 100 J = 1,12
         HOURLY(J) = X
```

```
         X = X + INCREMENT
  100 CONTINUE
C
C                  SET CHARACTERISTICS FOR SECOND HALF DAY
C
      IF (CI .EQ. 1) THEN              ! MIN TO MIN
         X = MAX(2)
         INCREMENT = (MIN(2) - MAX(2))/12.0
      ELSE ! CI .EQ. 2              ! MAX TO MAX
         X = MIN(1)
         INCREMENT = (MAX(2) - MIN(1))/12.0
      END IF
C
C                  COMPUTE THE HOURLY TEMPERATURES
C
      DO 200 J = 13,24
         HOURLY(J) = X
         X = X + INCREMENT
  200 CONTINUE
C
      RETURN
      END
C
SUBROUTINE HUBERM (MIN,MAX,LTHRES,UTHRES,TU,HU,ERROR)
C
C     PURPOSE:
C      TO CALCULATE DEGREE-DAYS ACCUMULATED ABOVE A THRESHOLD
C     USING A SINE WAVE ESTIMATION OF AREA UNDER THE CURVE WITH
C     REDUCED CONTRIBUTIONS ABOVE THE UPPER THRESHOLD.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  HUBERM.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  NO EXTERNAL ROUTINES
C
C    ****************************************************************
C
```



```
      INTEGER*2 ERROR                  ! RETURNED ERROR FLAG
C
      REAL*4 HALFPI,                   ! A CONSTANT
     *       HU,                       ! HEAT UNITS
     *       LTHRES,                   ! LOWER THRESHOLD
     *       MAX,                      ! MAXIMUM TEMPERATURE
     *       MEAN,                     ! MEAN TEMPERATURE
     *       MIN,                      ! MINIMUM TEMPERATURE
     *       PI,                       ! A CONSTANT
     *       THETA1,                   !
     *       THETA2,                   !
     *       UTHRES,                   ! UPPER THRESHOLD
     *       W                         !
C
      CHARACTER*1 TU                   ! TEMPERATURE UNITS
C
      DATA HALFPI / 1.570795/
      DATA PI     / 3.14159 /
C
```

```
      ERROR = 0
      HU = 0.0
      IF (MIN.GT.MAX) THEN
         ERROR = 1
         RETURN
      END IF
C
      MEAN = (MAX + MIN)/2.0
C
      IF (MIN.LT.LTHRES) THEN
         IF (MAX.LE.LTHRES) THEN
            RETURN
         ELSE IF (MAX.LE.UTHRES) THEN
            W = MAX - MEAN
            THETA1 = ASIN((LTHRES - MEAN)/W)
            HU = (W*COS(THETA1) - (LTHRES-MEAN)*(HALFPI-THETA1))/PI
         ELSE ! MAX.GT.UTHRES
            W = MAX - MEAN
            THETA1 = ASIN((LTHRES - MEAN)/W)
            THETA2 = ASIN((UTHRES - MEAN)/W)
            HU =   W * (COS(THETA1) - COS(THETA2))
     *           - (LTHRES - MEAN) * (HALFPI - THETA1)
     *           + (UTHRES - MEAN) * (HALFPI - THETA2)
            HU = HU / PI
         END IF
C
      ELSE ! MIN.GE.LTHRES
         IF (MIN.LT.UTHRES) THEN
            IF (MAX.LE.UTHRES) THEN
               IF (TU.EQ.'F') THEN
                  HU = MEAN - LTHRES - 0.3
               ELSE  ! (TU.EQ.'C')
                  HU = MEAN - LTHRES - .166667
               END IF
               IF (HU.LT.0.0) HU = 0.0
            ELSE ! MAX.GT.UTHRES
               W = MAX - MEAN
               THETA2 = ASIN((UTHRES-MEAN)/W)
               HU = (MEAN-LTHRES) - (W*COS(THETA2)-(UTHRES-MEAN)*
     *              (HALFPI-THETA2))/PI
            END IF
         ELSE ! MIN.GE.UTHRES
            HU = UTHRES - LTHRES
         END IF
      END IF
C
      RETURN
      END
C
      SUBROUTINE SISINE (AB,MIN,MAX,THRESH,DDAY,ERROR)
C     PURPOSE:
C      TO CALCULATE DEGREE-DAYS ACCUMULATED ABOVE or BELOW A
C     THRESHOLD USING A SINE WAVE ESTIMATION OF AREA UNDER THE CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  SISINE.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  NO EXTERNAL ROUTINES
C
```

```
C     ****************************************************************
C
```



```
      LOGICAL*2 AB                     ! T=ABOVE; F=BELOW
C
      INTEGER*2 ERROR                  !
C
      REAL*4 ARG,                      !
     *       DDAY,                     ! DEGREE-DAYS
     *       MAX,                      ! MAXIMUM TEMPERATURE
     *       MIN,                      ! MINIMUM TEMPERATURE
     *       MN,                       ! LOCAL COMPUTATIONAL MIN
     *       MX,                       ! LOCAL COMPUTATIONAL MAX
     *       PI,                       !
     *       Q,                        !
     *       TA,                       !
     *       TM,                       !
     *       THRESH,                   ! THRESHOLD
     *       XA                        !
C
      DATA PI /3.14159265/
C
C                  INITIALIZE VARIABLES
C
      DDAY = 0.0
      ERROR = 0
      IF (AB) THEN                     ! DEGREE-DAYS ABOVE
         MN = MIN
         MX = MAX
      ELSE                             ! DEGREE-DAYS BELOW (USE MIRROR IMAGE)
         MN = THRESH - (MAX - THRESH)
         MX = THRESH + (THRESH - MIN)
      END IF
C
C                  COMPUTE DEGREE-DAYS
C
      IF (MN.LT.MX) THEN
         IF (MX.LE.THRESH) RETURN
         TM = 0.5 * (MX + MN)
         TA = 0.5 * (MX - MN)
C
         ARG = (THRESH - TM)/TA
         IF (ARG.GT.1.0) ARG = 1.0
         IF (ARG.LT.-1.0) ARG = -1.0
C
C            APPROXIMATE VALUE OF THETA AT ARG
C
         XA = ABS(ARG)
         Q = 1.57079632 - SQRT(1.0-XA) * (1.5707288 + XA*
     *        (-0.2121144 + XA * (0.0745610 - XA * 0.0187293)))
         Q = ABS(Q)
         IF (ARG.LT.0) Q = -Q
         THETA = Q                     ! THETA = ARCSIN (ARG)
         THETA = THETA + 1.57079632    ! THETA = ARCCOS (ARG)
C
         DDAY = ((TM - THRESH) * (PI-THETA) + TA * SIN (THETA))/PI
      ELSE IF (MN.EQ.MX) THEN
```

```
         IF (MX.GT.THRESH) THEN
            DDAY = MX - THRESH
         END IF
      ELSE  ! MN.GT.MX
         ERROR = 1
      END IF
C
      RETURN
      END
C
      SUBROUTINE SITRIA (AB,MIN,MAX,THRESH,DDAY,ERROR)
C
C     PURPOSE:
C      TO CALCULATE DEGREE-DAYS ACCUMULATED ABOVE or BELOW A THRESHOLD
C     USING TRIANGULAR ESTIMATION OF AREA UNDER THE CURVE.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  SITRIA.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  NO EXTERNAL ROUTINES
C
C    ***************************************************************
C
```



```
C
      LOGICAL*2 AB                     ! T=ABOVE; F=BELOW
C
      INTEGER*2 ERROR                  !
C
      REAL*4 DDAY,                     ! DEGREE-DAYS
     *       MAX,                      ! MAXIMUM TEMPERATURE
     *       MIN,                      ! MINIMUM TEMPERATURE
     *       MN,                       ! LOCAL COMPUTATIONAL MIN
     *       MX,                       ! LOCAL COMPUTATIONAL MAX
     *       THRESH                    ! THRESHOLD
C
C                  INITIALIZE VARIABLES
C
      DDAY = 0.0
      ERROR = 0
      IF (AB) THEN                     ! DEGREE-DAYS ABOVE
         MN = MIN
         MX = MAX
      ELSE                             ! DEGREE-DAYS BELOW (USE MIRROR IMAGE)
         MN = THRESH - (MAX - THRESH)
         MX = THRESH + (THRESH - MIN)
      END IF
C
C                  COMPUTE DEGREE-DAYS
C
      IF (MN.LT.MX) THEN
         IF (MN.LT.THRESH) THEN
            IF (MX.GT.THRESH) THEN
               DDAY = (0.5 * (MX-THRESH)*(MX-THRESH))/(MX-MN)
            END IF
         ELSE
            DDAY = 0.5 * (MX + MN - (2 * THRESH))
```

```
         END IF
      ELSE IF (MN.EQ.MX) THEN
         IF (MX.GT.THRESH) THEN
            DDAY = MX - THRESH
         END IF
      ELSE ! MN.GT.MX
         ERROR = 1
      END IF
C
      RETURN
      END
C
      SUBROUTINE VERTCUT (AB,METHOD,MIN,MAX,THRESH,AREA)
C
C     PURPOSE:
C        CALCULATE THE AREA TO BE SUBTRACTED FROM DEGREE-DAYS
C     COMPUTED WITH A HORIZONTAL CUT OFF TO PRODUCE A VERTICAL
C     CUT OFF.
C
C     COMPILE AS:  FL %4I2 %4Nt %FPi  VERTCUT.FOR
C                  MICROSOFT FORTRAN 5.0
C
C     CALLS:  NO EXTERNAL ROUTINES
C
C    ****************************************************************
```



```
C
      LOGICAL*2 AB                     ! T=ABOVE; F=BELOW
C
      INTEGER*2 METHOD                 ! 1=SINE WAVE; 2=TRIANGULAR
C
      REAL*4 AREA,                     ! AREA TO BE REMOVED
     *       LT,                       ! LOCAL LOWER THRESHOLD
     *       MAX,                      ! MAXIMUM TEMPERATURE
     *       MIN,                      ! MINIMUM TEMPERATURE
     *       MN,                       ! LOCAL COMPUTATIONAL MIN
     *       MX,                       ! LOCAL COMPUTATIONAL MAX
     *       Q,                        !
     *       TA,                       ! TEMPERATURE AMPLITUDE
     *       THRESH(2),                ! THRESHOLDS
     *       TM,                       ! AVERAGE TEMPERATURE
     *       UT                        ! LOCAL UPPER THRESHOLD
C
C                  INITIALIZATIONS
C
      AREA = 0.0
      PI = 3.14159                     ! PI
      LT = THRESH(1)
      IF (AB) THEN                     ! DEGREE-DAYS ABOVE
         MN = MIN
         MX = MAX
         UT = THRESH(2)
      ELSE                             ! DEGREE-DAYS BELOW (USE MIRROR IMAGE)
         MN = LT - (MAX - LT)
         MX = LT + (LT - MIN)
         UT = LT + (LT - THRESH(2))
      END IF
```

```
C
C                  CHECK FOR APPROPRIATE CONDITIONS
C
      IF (MX.LE.UT) RETURN
      IF (MN.EQ.MX .OR. MN.GE.UT) THEN
         AREA = UT - LT
         RETURN
      END IF
C
C                  COMPUTE AREA FOR SINE WAVE
C
      IF (METHOD.EQ.1) THEN
         TM = 0.5 * (MX + MN)
         TA = 0.5 * (MX - MN)
C
         THETA = ASIN ((UT-TM)/TA)
         Q = PI - 2.0 * THETA
         AREA = Q * (UT-LT)/(PI * 2.0)
C
C                  COMPUTE AREA FOR TRIANGULAR METHOD
C
      ELSE ! METHOD.EQ.2
         Q = (MX - UT) / (MX - MN)
         AREA = Q * (UT - LT)
      END IF
C
      RETURN
      END
```

 Statewide IPM Project, University of California. /WEATHER/ddroutines.html revised: October 4, 2001.
Webmaster: ipmig@ucdavis.edu