APPENDIX
B

SKY TEMPERATURE ESTIMATION CODE

B.1 TYPE69.F Code

The algorithm developed by Martin and Berdahl [1984] is used in a TRNSYS subroutine to estimate the radiative sky temperature from TMY weather data. Two files are necessary to execute this subroutine, the original full TMY file and the corresponding TMY file from the Solar Energy Lab at the University of Wisconsin-Madison. The SEL TMY file contains the TMY data which is most commonly used by TRNSYS, corrected for known problems in the original data. The component outputs the radiative sky temperature, atmospheric pressure, and snow cover indicator. This subroutine is in the TRNSYS component library, TRNLIB, which is available from the Solar Energy Lab at the University of Wisconsin - Madison via anonymous ftp and the World Wide Web.

```
*======================================================================

      subroutine type69 (time, xin, out, t, dtdt, par, info, icntrl, *)

*----------------------------------------------------------------------
* Radiative sky temperature estimation subroutine.
*
* Reference:  Martin & Berdahl, "Characteristics of Infrared Sky
*             Radiation in the United States," Solar Energy, Vol. 33,
*             p. 321, 1984.
*
* Parameters: none
*
* Inputs from the original TMY file:
*   1. Ceiling(202): Filled with 9's 2 out of 3 hours. (m * 0.1)
* 2-5. Sky condition(203): Filled with 9's 2 out of 3 hours.
*   6. Station pressure(206): No known missing values. (kPa * 100)
*   7. Dewpoint temp(207): No known missing values. (C * 10)
* 8-9. Cloud cover(209): Filled with 9's 2 out of 3 hours. (* 10)
*  10. Snow cover(210): Filled with 9's at night.
```

```
*
* Inputs from the SEL TMY file:
*  11. Hour of the month
*  12. Ambient temperature (C)
*
* Outputs:
*   1. Radiative sky temperature (C)
*   2. Atmospheric pressure (kPa)
*   3. Snow cover indicator
*
* For some TRNSYS components, including the unglazed transpired
* collector system (Type 71), the sky temperature is needed to
* calculate the amount of radiation from the surface to the sky.
* However, the sky temperature is not included in the TMY weather
* files.
*
* This component calculates the sky temperature from data in a full TMY
* weather file.  Some of these data are missing, in which case they
* are replaced with the nearest previously recorded values.  In the
* TRNSYS deck, it is important to have reasonable initial values for
* the inputs.  The following are suggested:
*
* 7777.0  0.0  0.0  0.0  0.0  10125.0  0.0  0.0  0.0  0.0  0.0  0.0
*
* A ceiling value of 7777 corresponds to a clear sky.
*
* In addition, the atmospheric pressure is output.  The snow cover
* indicator is also output so that the ground reflectance can be more
* accurately calculated in the TRNSYS deck.
*
* Written by David Summers, February 1995.
*----------------------------------------------------------------------

      implicit none

* type69 variables

*     inputs
      integer iceiling, sc(4), ipatm, itdp, intot, inopaq, snow
      real mhr
      real*8 tamb

*     other variables
      integer i, istr, nscat, nbrok, layers
      real*8 hr, ceiling, patm, tdp, ntot, nopaq
      real*8 nlow, nthin, nceil, ho, cloud, pi, emiso, emis, tsky

* TRNSYS variables

      character*3 ycheck, ocheck
      real time, t, dtdt, par, s, time0, tfinal, delt
      double precision xin, out
      integer*4 info, unit, type
      integer np, ni, no, icntrl, iwarn, nstore, iav, iunit
      integer lur, luw, iform, luk
```

```
      parameter ( np=0, ni=12, no=3 )
      dimension xin(ni), out(no), info(15)
      dimension ycheck(ni), ocheck(no)

* TRNSYS common blocks

      common /sim/ time0, tfinal, delt, iwarn
      common /store/ nstore, iav, s(5000)
      common /lunits/ lur, luw, iform, luk

      data iunit/0/

*-----------------------------------------------------------------------
* first call of simulation

      if ( info(7) .gt. -1 ) go to 10
      info(10) = 10

* check parameters

      info(6) = no
      call typeck( 1, info, ni, np, 0 )

* setup initial storage values

      unit = info(1)
      type = info(2)
      istr = info(10)
      do i = 1,10
         s(istr+i-1) = 0.0
      enddo
      s(istr) = 7777.0   ! default iceiling if first value is missing
      s(istr+5) = 10125.0 ! default ipatm if first value is missing

* set variable types

      data ycheck/'DM1','DM1','DM1','DM1','DM1','DM1','DM1','DM1','DM1',
     &            'DM1','TD1','TE1'/
      data ocheck/'TE1','PR2','DM1'/
      call rcheck( info, ycheck, ocheck )

*-----------------------------------------------------------------------
* if its a different unit, set parameters

 10   continue
      if ( info(1) .eq. iunit ) go to 20

      iunit = info(1)

      pi = 3.14159265359

*-----------------------------------------------------------------------
* set inputs
* if missing values exist, replace with value at previous timestep

 20   continue
```

```
      iceiling = nint( xin(1) )
      if ( iceiling .eq. 9999 ) iceiling = nint( s(istr) )

      do i = 1,4
         sc(i) = nint( xin(1+i) )
         if ( sc(i) .eq. 9 ) sc(i) = nint( s(istr+i) )
      enddo

      ipatm = nint( xin(6) )
      if ( ipatm .eq. 99999 ) ipatm = nint( s(istr+5) )

      itdp = nint( xin(7) )
      if ( itdp .eq. 9999 ) itdp = nint( s(istr+6) )

      intot = nint( xin(8) )
      if ( intot .eq. 99 ) intot = nint( s(istr+7) )

      inopaq = nint( xin(9) )
      if ( inopaq .eq. 99 ) inopaq = nint( s(istr+8) )

      snow = nint( xin(10) )
      if ( snow .eq. 9 ) snow = nint( s(istr+9) )

      mhr  = xin(11)
      tamb = xin(12)

*----------------------------------------------------------------------
* store current inputs for next time step

      s(istr) = real( iceiling )
      do i = 1,4
         s(istr+i) = real( sc(i) )
      enddo
      s(istr+5) = real( ipatm )
      s(istr+6) = real( itdp )
      s(istr+7) = real( intot )
      s(istr+8) = real( inopaq )
      s(istr+9) = real( snow )

*----------------------------------------------------------------------

      hr      = amod( mhr, 24.0 )
      ceiling = iceiling * 10.0
      patm    = ipatm / 100.0
      tdp     = itdp / 10.0
      ntot    = intot / 10.0
      nopaq   = inopaq / 10.0

*----------------------------------------------------------------------
* calculate nlow, the fraction of opaque clouds at 2 km

      nlow = 0.0
      if ( iceiling .eq. 8888 ) then

         ceiling = 8000.0
```

```
          layers = 4
          do i = 4,1,-1
             if ( sc(i) .eq. 0 ) layers = i - 1
          enddo

          nscat = 0
          nbrok = 0
          do i = 1,layers-1
             if ( sc(i) .eq. 2 ) nscat = nscat + 1
             if ( sc(i) .eq. 4 ) nbrok = nbrok + 1
          enddo

*         the fraction of low opaque clouds are estimated based on the
*         number of broken and scattered opaque cloud layers before
*         the ceiling.  these numbers are my best guesses.

          if ( nscat .eq. 0 ) then
             if ( nbrok .eq. 1 ) nlow = 0.6
             if ( nbrok .eq. 2 ) nlow = 0.9
             if ( nbrok .eq. 3 ) nlow = 1.0
          endif

          if ( nscat .eq. 1 ) then
             if ( nbrok .eq. 0 ) nlow = 0.2
             if ( nbrok .eq. 1 ) nlow = 0.7
             if ( nbrok .eq. 2 ) nlow = 1.0
          endif

          if ( nscat .eq. 2 ) then
             if ( nbrok .eq. 0 ) nlow = 0.3
             if ( nbrok .eq. 1 ) nlow = 0.8
          endif

          if ( nscat .eq. 3 ) nlow = 1.0

       endif

*----------------------------------------------------------------------
* calculate tsky

       if ( iceiling .eq. 7777 ) ceiling = 2000.0

       nthin = ntot - nopaq
       nceil = nopaq - nlow

       ho = 8200.0
       cloud = nthin * 0.4 * exp( - 8000.0  / ho )
     &       + nceil * 1.0 * exp( - ceiling / ho )
     &       + nlow  * 1.0 * exp( - 2000.0  / ho )

       emiso = 0.711 + 0.0056*tdp + 7.3e-5*tdp*tdp
     &       + 0.013 * cos( hr*pi/12.0 )
     &       + 0.0012 * ( patm - 100.0 )
       emis = emiso + ( 1 - emiso ) * cloud
       tsky = - 273.15 + (tamb+273.15) * (emis)**(0.25)
```

```
*-----------------------------------------------------------------------
* set outputs

      out(1) = tsky
      out(2) = patm
      out(3) = real( snow )

*-----------------------------------------------------------------------

      return 1
      end

*=======================================================================
```

B.2 TRNSYS Manual Page

TYPE 69: RADIATIVE SKY TEMPERATURE ESTIMATOR

General Description

For some TRNSYS components, including the unglazed transpired collector system (Type 71), the sky temperature is needed to calculate the amount of radiation from the surface to the sky. However, the sky temperature is not included in the TMY weather files.

This component calculates the sky temperature from data in a full TMY weather file. Some of these data are missing, in which case they are replaced with the nearest previously recorded values. In the TRNSYS deck, it is important to have reasonable initial values for the inputs. The following are suggested:

7777.0 0.0 0.0 0.0 0.0 10125.0 0.0 0.0 0.0 0.0 0.0 0.0

A ceiling value of 7777 corresponds to a clear sky. It is also important to make sure that the inputs to the Type 69 subroutine from the TRNSYS deck are not interpolated. So, the data reader (Type 9) which reads the full TMY data file should have the parameters set such that interpolation is off.

The atmospheric pressure and snow cover indicator are also output. The snow cover indicator is output so that the ground reflectance can be more accurately calculated in the TRNSYS deck.

Mathematical Description

See the article by Martin and Berdahl [1] for a detailed mathematical description.

TRNSYS Component Configuration

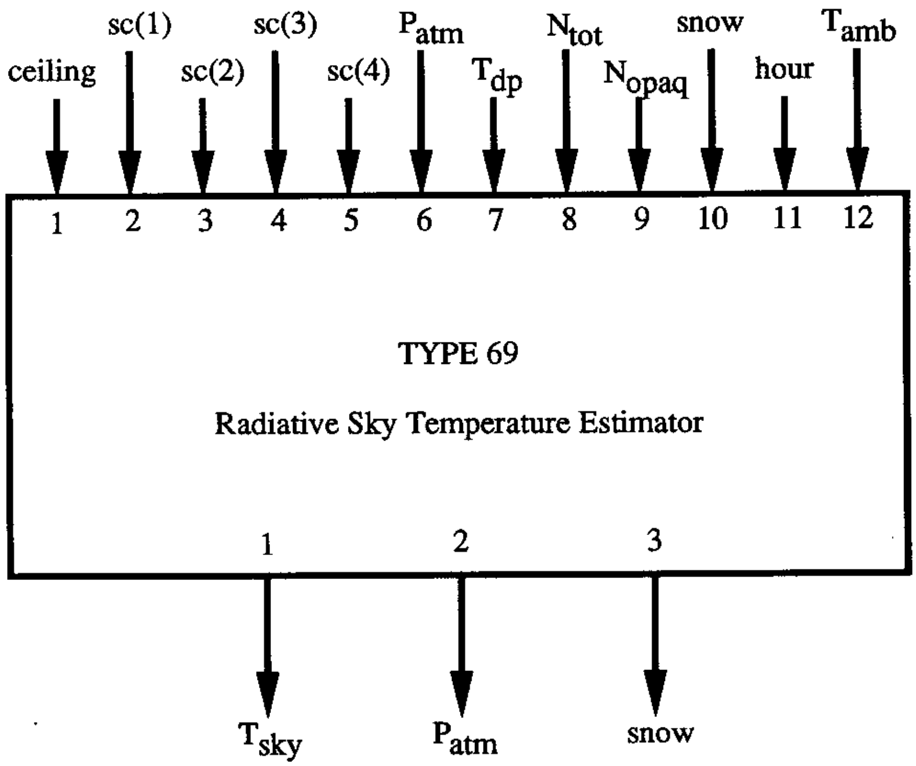
PARAMETER NO.		DESCRIPTION
none		

INPUT NUMBER		DESCRIPTION
1	ceiling	- ceiling height (m) (x 0.1)

2	sc(1)	- sky cover indicator 1
3	sc(2)	- sky cover indicator 2
4	sc(3)	- sky cover indicator 3
5	sc(4)	- sky cover indicator 4
6	P_{atm}	- atmospheric pressure (kPa) (x 100)
7	T_{dp}	- dew point temperature (C) (x 10)
8	N_{tot}	- fraction of the sky covered by clouds (x 10)
9	N_{opaq}	- fraction of the sky covered by opaque clouds (x 10)
10	snow	- snow cover indicator
11	hour	- hour of the month
12	T_{amb}	- ambient temperature (C)

OUTPUT NUMBER		DESCRIPTION
1	T_{sky}	- radiative sky temperature (°C)
2	P_{atm}	- atmospheric pressure (kPa)
3	snow	- snow cover indicator

Information Flow Diagram



Parameters: none

References

1. Martin, M. and Berdahl, P., "Characteristics of Infrared Sky Radiation in the United States," Solar Energy, Vol. 33, p. 321, 1984.