APPENDIX
A

TRNSYS MODEL CODE

A.1 TYPE71.F Code

The unglazed transpired collector subroutine is written to simulate the performance of UTC systems with TRNSYS. See Chapters 2 and 3 for a description of the UTC system theory and model. 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 type71 (time, xin, out, t, dtdt, par, info, icntrl, *)

*------------------------------------------------------------------------
* Unglazed transpired collector (UTC) subroutine.
*
* Reference:  Kutscher, C.F., "Heat Exchanger Effectiveness and
*    Pressure Drop for Air Flow Through Perforated Plates With and
*    Without Crosswind,"  J. Heat Transfer, vol 116, p 391, 1994.
*
* Parameters:
*   1. Collector area (m2)
*   2. Collector height (m)
*   3. Collector hole diameter (m)
*   4. Collector hole pitch, distance between centers of holes (m)
*   5. Collector emissivity
*   6. Collector absorptivity
*   7. Plenum depth (m)
*   8. Emissivity of the wall behind the collector
*   9. R-value of the wall behind the collector (C-m2-hr/kJ)
*  10. Total UA-value of the building walls and roof (kJ/hr-C)
*  11. Room air temperature (C)
*  12. Ambient air temperature above which the summer bypass damper
*       is opened (C)
*  13. Maximum auxiliary heat rate available (kJ/hr)
*  14. Night bypass = 0 if bypass not automatically opened at night
*                   = 1 if bypass automatically opened at night
*
```

```
      a(1,1) = - effhx
      a(1,2) = 1.0
      a(2,1) = - emisc * sb * surfa
     &          * (tcol**2 + tsur**2) * (tcol + tsur)
      a(2,3) = 1.0
      a(3,1) = sb * area * (tcol**2 + twall**2) * (tcol + twall)
     &          / ( 1.0/emisw + 1.0/emisc - 1.0 )
      a(3,4) = 1.0
      a(3,5) = - sb * area * (tcol**2 + twall**2) * (tcol + twall)
     &          / ( 1.0/emisw + 1.0/emisc - 1.0 )
      a(4,2) = hvwall * area
      a(4,5) = - hvwall * area
      a(4,6) = 1.0
      a(5,2) = - gamma * mflow1 * aircp
      a(5,7) = 1.0
      a(6,3) = 1.0
      a(6,4) = - 1.0
      a(6,7) = 1.0
      a(7,4) = - 1.0
      a(7,6) = - 1.0
      a(7,8) = 1.0
      a(8,5) = udwall * area
      a(8,8) = 1.0

      b(1) = tamb * ( 1.0 - effhx )
      b(2) = - emisc * sb * surfa
     &        * (tcol**2 + tsur**2) * (tcol + tsur) * tsur
      b(5) = - gamma * mflow1 * aircp * tamb
      b(6) = qabs
      b(8) = udwall * area * troom

* start iterations

      k = 0
 110  continue
      k = k + 1

* solve for unknowns using lapack matrix solver
* see http://www.netlib.org -> lapack -> lapack/double

      call dgesv( neq, one, a, neq, ipiv, b, neq, ifail )

      if ( ifail .ne. 0  ) then
         write (luw,813) unit, type, ifail, nint(month), nint(hour)
 813     format (//1x,'***** ERROR *****',/1x,'UNIT ',i3,
     &       ' TYPE ',i3,' UNGLAZED TRANSPIRED COLLECTOR',/1x,
     &       'MATRIX SOLVER ERROR, IFAIL = ',i3,/1x,
     &       'MONTH = ',i2,'  HOUR = ',i3)
         stop
      endif

      do i = 1,neq
         x(i) = b(i)
      enddo
      tcol  = x(1)
```

```
* Inputs:
*    1. Month of year
*    2. Hour of month
*    3. Radiation incident on the collector (kJ/m2)
*    4. Ambient temperature (C)
*    5. Sky temperature (C)
*    6. Atmospheric pressure (kPa)
*    7. Internal gains due to people, equipment, etc. (kJ/hr)
*    8. Supply air flow rate from collector air-handling units (m3/hr)
*    9. Minimum outdoor air flow rate through collector/summer bypass
*       damper (m3/hr)
*   10. Supply air flow rate from no-collector air-handling units
*       (m3/hr)
*   11. Outdoor air flow rate through no collector (m3/hr)
*
* Outputs:
*    1. Plenum air temperature (C)
*    2. Collector outlet air temperature (C)
*    3. Mixed air temperature (C)
*    4. Supply air temperature (C)
*    5. Collector surface temperature (C)
*    6. Energy savings rate (kJ/hr)
*    7. Convection from collector (kJ/hr)
*    8. Convection from wall (kJ/hr)
*    9. Radiation from collector (kJ/hr)
*   10. Radiation from wall (kJ/hr)
*   11. Conduction through wall (kJ/hr)
*   12. Reduced conduction through wall because of collector (kJ/hr)
*   13. Absorbed energy rate (kJ/hr)
*   14. Auxiliary heating rate (kJ/hr)
*   15. Outdoor air flow rate through collector/summer bypass damper
*       (m3/hr)
*   16. Heat exchanger effectiveness of collector (C)
*   17. Solar efficiency of the collector
*   18. Pressure drop across collector plate (kPa)
*   19. Bypass damper position = 0.0 if open
*                              = 1.0 if closed
*   20. Heat rate supplied by a traditional heating system (kJ/hr)
*   21. Additional fan power required (kJ/hr)
*
* For this model to correctly calculate the performance of transpired
* collectors, the approach velocity (appvel) should be greater than
* 72 m/hr.  Otherwise, there will be convection losses from the
* collector between the holes, and the collector's performance will be
* reduced.  Also, the collector pressure drop (pcol) should be at least
* 0.025 Pa to ensure uniform flow through the collector.  Otherwise,
* sections of the collector will become hotter than others, and
* radiation losses from the collector will increase.  Again, this will
* reduce the collector's performance.  To achieve a sufficient pressure
* drop, the porosity (por) should be about 0.005 to 0.01 for the given
* approach velocities.  If these approach velocity and pressure drop
* conditions are not met, this subroutine will write a warning to a
* file.  It is important to emphasize that a collector _can_ be operated
* at approach velocities and pressure drops below these values, but this
* model will just over predict the performance.
*
```

```
* Written by David Summers, Solar Energy Lab, U. Wisconsin - Madison
* for M.S. thesis, December 1995.
*---------------------------------------------------------------------

      implicit none

* type71 variables

*     parameters
      real*8 area, ht, diam, pitch, emisc, absor, depth, emisw, rwall
      real*8 ua, troom, tbypass, qmax, nitebp

*     inputs
      real*8 month, hour, rad, tamb, tsky, patm, gain, flow1, minout1
      real*8 flow2, out2

*     values calculated directly from parameters and inputs
      real*8 por, surfa, udwall, qabs, tgnd, tsur, qbldg
      real*8 aircp, aircond, airvisc, airden, a0, a1, a2
      real*8 mflow1, mflow2, mflow, gmin, beta, qtrad

*     outputs and other variables
      real*8 gamma, tout, tcol, tplen, twall, qvcol, qrcol, qvwall
      real*8 qrwall, qdwall, red, effhx, appvel
      real*8 tmix, tsup, qaux1, lowg, hig, oldg, dif, diflim, small, g
      real*8 zeta, pcol, plenden, f, dh, plenvel, pfric, pbuoy, pacc
      real*8 delp, fanpw
      real*8 qaux2, qaux, film, tsolair, qpot, qred, qu, qsave, soleff
      integer istr, i, j, jmax
      real warn
      character*10 warnfile

* 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=14, ni=11, no=21 )
      dimension par(np), xin(ni), out(no), info(15)
      dimension ycheck(ni), ocheck(no)

* UTC common block (used in utcsolve subroutine)

      common /utc/ area, ht, diam, pitch, emisc, depth, emisw, troom,
     &             flow1, mflow1, por, surfa, udwall, tamb, qabs, tsur,
     &             aircp, aircond, airvisc, airden, unit, type,
     &             month, hour, g

* TRNSYS common blocks

      common /sim/ time0, tfinal, delt, iwarn
      common /store/ nstore, iav, s(5000)
```

```
      common /lunits/ lur, luw, iform, luk

      data iunit/0/

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

      diflim   = 1.0e-4
      small    = 1.0e-4
      jmax     = 100
      warnfile = 'WARN.UTC'
      g        = 9.8 * 3600**2    ! acceleration of gravity, m/hr2

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

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

* check parameters

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

* setup for warnings

      unit = info(1)
      type = info(2)
      istr = info(10)
      s(istr) = -1.0
      open( unit = 71, file = warnfile, status = 'unknown' )
      write (71,800) type
 800  format ('Type',i3,' warnings:')

* set variable types

      data ycheck/'MN1','TD1','IR1','TE1','TE1','PR2','PW1','VF1','VF1',
     &            'VF1','VF1'/
      data ocheck/'TE1','TE1','TE1','TE1','TE1','PW1','PW1','PW1','PW1',
     &            'PW1','PW1','PW1','PW1','PW1','VF1','DM1','DM1','PR2',
     &            'DM1','PW1','PW1'/
      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)
      area    = par(1)
      ht      = par(2)
      diam    = par(3)
      pitch   = par(4)
      emisc   = par(5)
      absor   = par(6)
      depth   = par(7)
```

```
      emisw   = par(8)
      rwall   = par(9)
      ua      = par(10)
      troom   = par(11)
      tbypass = par(12)
      qmax    = par(13)
      nitebp  = par(14)

      por    = 0.907 * ( diam / pitch )**2
      surfa  = ( 1 - por ) * area
      udwall = 1.0 / rwall

*-----------------------------------------------------------------------
* set inputs

 20   continue
      month   = xin(1)
      hour    = xin(2)
      rad     = xin(3)
      tamb    = xin(4)
      tsky    = xin(5)
      patm    = xin(6)
      gain    = xin(7)
      flow1   = xin(8)
      minout1 = xin(9)
      flow2   = xin(10)
      out2    = xin(11)

      qabs = rad * surfa * absor
      tgnd = tamb
      tsur = ( 0.5 * ((tsky+273.15)**4 + (tgnd+273.15)**4) )**0.25
     &       - 273.15

* calculate building loss, assuming no reduced wall loss

      qbldg = ua * (troom - tamb) - gain

* calculate ambient air properties (curve-fits from EES)

      aircp   = 1.0062 + 3.6028e-5*tamb
     &        - 1.0885e-6*tamb*tamb + 1.3791e-8*tamb**3
      aircond = 0.08659 + 2.6993e-4*tamb
      airvisc = 0.06201 + 1.7428e-4*tamb

      a0 = -5.1936e-5 + 1.2758e-2*patm
      a1 = -5.7037e-7 - 4.6865e-5*patm
      a2 = -5.6746e-9 + 1.5511e-7*patm
      airden = a0 + a1*tamb + a2*tamb*tamb

      mflow1 = flow1 * airden
      mflow2 = flow2 * airden
      mflow = mflow1 + mflow2

      qtrad = (minout1 + out2) * airden * aircp * (troom - tamb)
     &      + qbldg
      if ( qtrad .lt. small ) qtrad = 0.0
```

```
*---------------------------------------------------------------------
* check if summer bypass damper is open

      if ( tamb .gt. tbypass ) then
         do i = 1,no
            out(i) = 0.0
         enddo
         out(2)  = tamb                       ! tout = tamb
         out(3)  = tamb                       ! tmix = tamb
         out(4)  = tamb                       ! tsup = tamb
         out(15) = flow1                      ! 100% fresh air
         return 1
      endif

*---------------------------------------------------------------------
* check air flow rates

      if ( minout1 .gt. flow1  .or.  out2 .gt. flow2 ) then
         write (luw,810) unit, type
 810     format (//1x,'***** ERROR *****',/1x,'UNIT ',i3,
     &      ' TYPE ',i3,' UNGLAZED TRANSPIRED COLLECTOR',/1x,
     &      'CHECK AIR FLOW RATES IN DECK')
         stop
      endif

      if ( flow2 .gt. small ) then
         beta = out2 / flow2
      else
         beta = 0.0
      endif

      if ( flow1 .gt. small ) then
         gmin = minout1 / flow1
      else
         qaux = qtrad
         do i = 1,no
            out(i) = 0.0
         enddo
         out(2)  = tamb                       ! tout = tamb
         out(3)  = tamb                       ! tmix = tamb
         out(4)  = tamb                       ! tsup = tamb
         out(14) = qaux
         out(20) = qtrad
         return 1
      endif

*---------------------------------------------------------------------
* at night, open bypass damper if nitebp = 1

      if ( qabs .lt. small  .and.  nitebp .gt. 0.5 ) then
         gamma = gmin
         tout = tamb
         tmix = gamma*tout + (1.0 - gamma)*troom
         tsup = troom + qbldg / (mflow * aircp)
         qaux = qtrad
```

```
          do i = 1,no
             out(i) = 0.0
          enddo
          out(2)  = tout
          out(3)  = tmix
          out(4)  = tsup
          out(14) = qaux
          out(15) = minout1
          out(20) = qtrad
          return 1
       endif

*-----------------------------------------------------------------------
* find gamma such that qaux1 is minimized.  in the winter, this will
* usually be when gamma = gmin.  if the air is too hot for this
* condition, then no auxiliary heat is needed (i.e. qaux1 = 0) and
* gamma > gmin.  if gamma is between gmin and 1.0, then use the
* bisection method to determine gamma.
*-----------------------------------------------------------------------

* calculate necessary supply air temperature to meet load

      tsup = troom + qbldg / (mflow * aircp)

* solve equations for gamma = gmin

      gamma = gmin
      call utcsolve( gamma, tout, tcol, tplen, twall,
     &               qvcol, qrcol, qvwall, qrwall, qdwall,
     &               red, effhx, appvel )
      tmix = gamma*tout + (1.0 - gamma)*troom

      if ( tmix .lt. tsup ) then
*        calculate auxiliary heat
         qaux1 = mflow1 * aircp * ( tsup - tmix )
      else
*        solve equations for gamma = 1
         qaux1 = 0.0
         gamma = 1.0
         call utcsolve( gamma, tout, tcol, tplen, twall,
     &                  qvcol, qrcol, qvwall, qrwall, qdwall,
     &                  red, effhx, appvel )
         tmix = tout
         if ( tmix .lt. tsup ) then
*           use bisection method to find gamma
            j = 0
            lowg  = gmin
            hig = 1.0
            gamma = (lowg + hig) / 2.0
 100        continue
               j = j + 1
               call utcsolve( gamma, tout, tcol, tplen, twall,
     &                        qvcol, qrcol, qvwall, qrwall, qdwall,
     &                        red, effhx, appvel )
               tmix = gamma*tout + (1.0 - gamma)*troom
```

```
            if ( tmix .lt. tsup ) then
               hig = gamma
            else
               lowg = gamma
            endif
            oldg = gamma
            gamma = (lowg + hig) / 2.0
            dif = abs( gamma - oldg )

 109       if ( dif .gt. diflim  .and.  j .lt. jmax ) go to 100
           if ( j .ge. jmax ) then
              write (luw,811) unit, type, dif, nint(month), nint(hour)
 811          format (//1x,'***** ERROR *****',/1x,'UNIT ',i3,
     &           ' TYPE ',i3,' UNGLAZED TRANSPIRED COLLECTOR',/1x,
     &           'NO CONVERGENCE IN J LOOP, DIF = ',e7.2,/1x,
     &           'MONTH = ',i2,'  HOUR = ',i3)
              stop
           endif
        endif
      endif

*-----------------------------------------------------------------------
* calculate additional fan power

* calculate pressure drop across collector

      zeta  = 6.82 * ( (1-por)/por )**2 * ( red )**( -0.236 )
      pcol  = 0.5 * airden * appvel * appvel * zeta * 7.71605e-11
                          ! factor of 7.7e-11 to convert units

* calculate air density in plenum

      plenden = a0 + a1*tplen + a2*tplen*tplen

* calculate friction pressure drop through plenum

      f = 0.05    ! estimate
      dh = 4.0 * ( depth * area/ht ) / ( 2.0 * ( area/ht + depth ) )
      plenvel = 0.5 * appvel * ht / depth
      pfric = f * (ht / dh) * plenden * plenvel**2 / 2.0
*     convert units to kPa
      pfric = pfric / ( 3600.0**2 * 1000.0 )

* calculate buoyancy pressure term

      pbuoy = (airden - plenden) * g * ht
*     convert units to kPa
      pbuoy = pbuoy / ( 3600.0**2 * 1000.0 )

* calculate acceleration pressure drop

      pacc = plenden * (2.0*plenvel)**2 / 2.0
*     convert units to kPa
      pacc = pacc / ( 3600.0**2 * 1000.0 )

* calculate total pressure drop and fan power
```

```
      delp = pcol + pfric - pbuoy + pacc
      fanpw = gamma * flow1 * delp

*------------------------------------------------------------------------
* calculate reduced conduction through wall

      film = 54.0         ! (kJ/hr-m2-C) film coefficient for air
                          !              against original wall
      tsolair = tamb + (emisw * rad / film)     ! sol-air temp for
                                                ! absor = emis

      qpot = udwall * area * (troom - tsolair) ! potential conduction
      qred = qpot - qdwall                     ! reduced conduction

* calculate total auxiliary heat required, subtracting reduced wall
* loss since it was assumed to be zero when qbldg was calculated

      qaux2 = mflow2 * aircp * ( tsup - beta*tamb - (1.0-beta)*troom )
      qaux = qaux1 + qaux2 - qred
      if ( qaux .lt. small ) qaux = 0.0

* calculate energy savings

      qu    = qvcol + qvwall             ! useful energy gained by air
      qsave = qtrad - qaux               ! energy saved

* calculate solar efficiency

      if ( rad .gt. small ) then
         soleff = qvcol / ( rad * area )
         if ( soleff .gt. 1.0 ) soleff = 1.0
         if ( soleff .lt. 0.0 ) soleff = 0.0
      else
         soleff = 0.0
      endif

*------------------------------------------------------------------------
* write warnings to file

      if ( info(7) .lt. 0 ) go to 30
      warn = -1.0

      if ( appvel .lt. 72.0  ) then
         write (71,801) month, hour
 801     format (/,'month = ',f3.0,'  hour = ',f5.1)
         write (71,802) appvel
 802     format ('* approach velocity = ',f5.2,' m/hr',/,'  For appvel',
     &      ' < 72 m/hr, model overpredicts performance')
         warn = 1.0
      endif

      if ( pcol .lt. 0.025 ) then
         if ( warn .lt. 0.0 ) then
            write (71,801) month, hour
         endif
```

```
        write (71,803) pcol
 803    format ('* pressure drop = ',f6.4,' kPa',/,'  For pcol',
     &       ' < 0.025 kPa, model overpredicts performance')
        warn = 1.0
      endif

      if ( qaux .gt. qmax ) then
        if ( warn .lt. 0.0 ) then
          write (71,801) month, hour
        endif
        write (71,804) qaux, qmax
 804    format ('* qaux = ',e8.2,' kJ/hr;  qmax = ',e8.2,' kJ/hr',/,
     &       '  auxiliary heater(s) too small to meet load')
        warn = 1.0
      endif

      if ( tmix .gt. tsup ) then
        if ( warn .lt. 0.0 ) then
          write (71,801) month, hour
        endif
        write (71,805) tmix, tsup
 805    format ('* actual tsup = ',f5.1,' C;  desired tsup = ',f5.1,
     &       ' C',/,'  tsup is too hot, summer bypass damper should',
     &       ' have been opened')
        tsup = tmix
        warn = 1.0
      endif

      if ( warn .gt. 0.0 ) then
        istr = info(10)
        if ( s(istr) .lt. 0.0 ) then
          write (luw,812) unit, type, warnfile
 812      format (//2x,'***** WARNING ***** UNIT',i3,' TYPE',i3/4x,
     &         'CHECK ', a10,
     &         ' FOR UNGLAZED TRANSPIRED COLLECTOR WARNINGS')
          s(istr) = 1.0
          iwarn = iwarn + 1
        endif
      endif

 30   continue

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

      out(1)  = tplen
      out(2)  = tout
      out(3)  = tmix
      out(4)  = tsup
      out(5)  = tcol
      out(6)  = qsave
      out(7)  = qvcol
      out(8)  = qvwall
      out(9)  = qrcol
      out(10) = qrwall
      out(11) = qdwall
```

```
      out(12) = qred
      out(13) = qabs
      out(14) = qaux
      out(15) = gamma * flow1
      out(16) = effhx
      out(17) = soleff
      out(18) = pcol
      out(19) = 1.0
      out(20) = qtrad
      out(21) = fanpw

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

      return 1
      end

*=======================================================================

      subroutine utcsolve( gamma, tout, tcol, tplen, twall,
     &                     qvcol, qrcol, qvwall, qrwall, qdwall,
     &                     red, effhx, appvel )

*-----------------------------------------------------------------------
* This subroutine solves the energy balances on the collector, air, and
* outside wall surface.  The temperatures and heat flows are output.
*
* The temperatures are:
*
*   tout  = air at the outlet from the collector
*   tcol  = collector surface
*   tplen = air in the plenum
*   twall = outside wall surface
*   troom = air in the room
*   tamb  = ambient air
*   tsur  = surroundings (sky & ground) for radiation calculation
*
* All temperatures in this subroutine are converted from Celsius to
* Kelvin at the beginning, and then they are converted back to Celsius
* at the end for the main UTC subroutine.
* The heat flows are labelled 'qXsource', where X = (r, v, d) for
* radiation, convection, conduction.  The source of the heat flow is
* defined for the usual direction of heat flow for winter operation,
* from the inside of the building to the outside.  So, qvwall is the
* convection from the wall to the plenum (not wall to the room).
* Similarly, qrcol is the radiation from the collector to the
* surroundings (not collector to wall).  The only exceptions are qabs,
* the absorbed solar energy, and qdwall, the conduction through the
* wall.  The heat transfer coefficients are labelled the same way.
*
*-----------------------------------------------------------------------

      implicit none

* TRNSYS variables

      integer lur, luw, iform, luk
```

```
* utcsolve variables

*     arguments
      real*8 gamma, tout, tcol, tplen, twall, qvcol, qrcol, qvwall
      real*8 qrwall, qdwall, red, effhx, appvel

*     UTC common block variables
      real*8 area, ht, diam, pitch, emisc, depth, emisw, troom
      real*8 flowl, mflowl, por, surfa, udwall, tamb, qabs, tsur
      real*8 aircp, aircond, airvisc, airden, month, hour, g
      integer*4 unit, type

*     subroutine internal variables
      real*8 pi, sb, diflim, small
      real*8 holevel, plenvel, nud, hvcol, pr, reht, nuht, hvwall
      double precision a, b
      real*8 x, res, dif
      integer i, j, k, kmax, neq, one, ipiv, ifail

      parameter ( neq = 8 )
      dimension a(neq,neq), b(neq), x(neq), ipiv(neq), res(neq)

* TRNSYS common block

      common /lunits/ lur, luw, iform, luk

* UTC common block

      common /utc/ area, ht, diam, pitch, emisc, depth, emisw, troom,
     &             flowl, mflowl, por, surfa, udwall, tamb, qabs, tsur,
     &             aircp, aircond, airvisc, airden, unit, type,
     &             month, hour, g

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

      pi     = 3.14159265359
      sb     = 2.0412e-7        ! Stefan-Boltzmann, kJ/hr-m2-K4
      diflim = 1.0e-4
      small  = 1.0e-4
      kmax   = 100
      one    = 1
      ifail  = 0

* check for no flow through collector

      if ( gamma .lt. small ) then
         tout   = tamb
         tcol   = tamb
         tplen  = tamb
         twall  = tamb
         qvcol  = 0.0
         qrcol  = 0.0
         qvwall = 0.0
         qrwall = 0.0
         qdwall = udwall * area * ( troom - tplen )
```

```
         red    = 0.0
         effhx  = 0.0
         appvel = 0.0
         return
      endif

* convert celsius to kelvin

      troom = troom + 273.15
      tamb  = tamb + 273.15
      tsur  = tsur + 273.15

* calculate approach, hole, and plenum velocities

      appvel  = gamma * flow1 / area
      holevel = gamma * flow1 / ( area * por )
      plenvel = 0.5 * appvel * ht / depth

* calculate heat exchanger effectiveness for collector

      red = airden * holevel * diam / airvisc
      nud = 2.75 * ( pitch/diam )**( -1.2 ) * ( red )**( 0.43 )
      hvcol = nud * aircond / diam
      effhx = 1 - exp( - hvcol * surfa / (gamma * mflow1 * aircp) )

* calculate heat transfer coefficient for wall to air convection

      pr = 0.71
      reht = airden * plenvel * ht / airvisc
      if ( reht .gt. 500000 ) then
*        turbulent
         nuht = (0.037*reht**0.8 - 871.0) * pr**(1.0/3.0)
      else
*        laminar
         nuht = 0.664 * reht**0.5 * pr**(1.0/3.0)
      endif
      hvwall = nuht * aircond / ht

*-----------------------------------------------------------------------
* solve simultaneous equations for unknown temperatures and heat flows
* [a][x] = [b]
*-----------------------------------------------------------------------

* initial guesses for collector and wall temperatures

      tcol  = tamb
      twall = tamb

* calculate [a] matrix and [b] array

      do i = 1,neq
         do j = 1,neq
            a(i,j) = 0.0
         enddo
         b(i) = 0.0
      enddo
```

```
      twall = x(5)

*  calculate [a] matrix and [b] array

      do i = 1,neq
         do j = 1,neq
            a(i,j) = 0.0
         enddo
         b(i) = 0.0
      enddo

      a(1,1) = - effhx
      a(1,2) = 1.0
      a(2,1) = - emisc * sb * surfa
     &         * (tcol**2 + tsur**2) * (tcol + tsur)
      a(2,3) = 1.0
      a(3,1) = sb * area * (tcol**2 + twall**2) * (tcol + twall)
     &         / ( 1.0/emisw + 1.0/emisc - 1.0 )
      a(3,4) = 1.0
      a(3,5) = - sb * area * (tcol**2 + twall**2) * (tcol + twall)
     &         / ( 1.0/emisw + 1.0/emisc - 1.0 )
      a(4,2) = hvwall * area
      a(4,5) = - hvwall * area
      a(4,6) = 1.0
      a(5,2) = - gamma * mflow1 * aircp
      a(5,7) = 1.0
      a(6,3) = 1.0
      a(6,4) = - 1.0
      a(6,7) = 1.0
      a(7,4) = - 1.0
      a(7,6) = - 1.0
      a(7,8) = 1.0
      a(8,5) = udwall * area
      a(8,8) = 1.0

      b(1) = tamb * ( 1.0 - effhx )
      b(2) = - emisc * sb * surfa
     &       * (tcol**2 + tsur**2) * (tcol + tsur) * tsur
      b(5) = - gamma * mflow1 * aircp * tamb
      b(6) = qabs
      b(8) = udwall * area * troom

*  calculate residuals

      do i = 1,neq
         res(i) = 0.0
         do j = 1,neq
            res(i) = res(i) + a(i,j) * x(j)
         enddo
         res(i) = res(i) - b(i)
      enddo

*  check for convergence

      dif = 0.0
      do i = 1,neq
```

```
         dif = sqrt( dif**2 + res(i)**2 )
      enddo

 119  if ( dif .gt. diflim  .and.  k .lt. kmax ) go to 110
      if ( k .ge. kmax ) then
         write (luw,815) unit, type, dif, nint(month), nint(hour)
 815     format (//1x,'***** ERROR *****',/1x,'UNIT ',i3,
     &      ' TYPE ',i3,' UNGLAZED TRANSPIRED COLLECTOR',/1x,
     &      'NO CONVERGENCE IN K LOOP, DIF = ',e7.2,/1x,
     &      'MONTH = ',i2,'  HOUR = ',i3)
         stop
      endif

*-----------------------------------------------------------------------
* solution has been found

      tcol   = x(1)
      tplen  = x(2)
      qrcol  = x(3)
      qrwall = x(4)
      twall  = x(5)
      qvwall = x(6)
      qvcol  = x(7)
      qdwall = x(8)

* calculate tout from energy balance

      tout = tplen + qvwall / ( gamma * mflow1 * aircp )

* convert kelvin to celsius

      tout  = tout - 273.15
      tcol  = tcol - 273.15
      tplen = tplen - 273.15
      twall = twall - 273.15
      troom = troom - 273.15
      tamb  = tamb - 273.15
      tsur  = tsur - 273.15

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

      return
      end

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

A.2 TRNSYS Manual Page

TYPE 71: UNGLAZED TRANSPIRED COLLECTOR SYSTEM

General Description

Unglazed transpired collectors (UTCs) consist of a perforated, solar-absorbing plate mounted on a large south-facing wall. Air is drawn through the holes in the plate, into the plenum, and finally into the building, as shown in Figure 1.

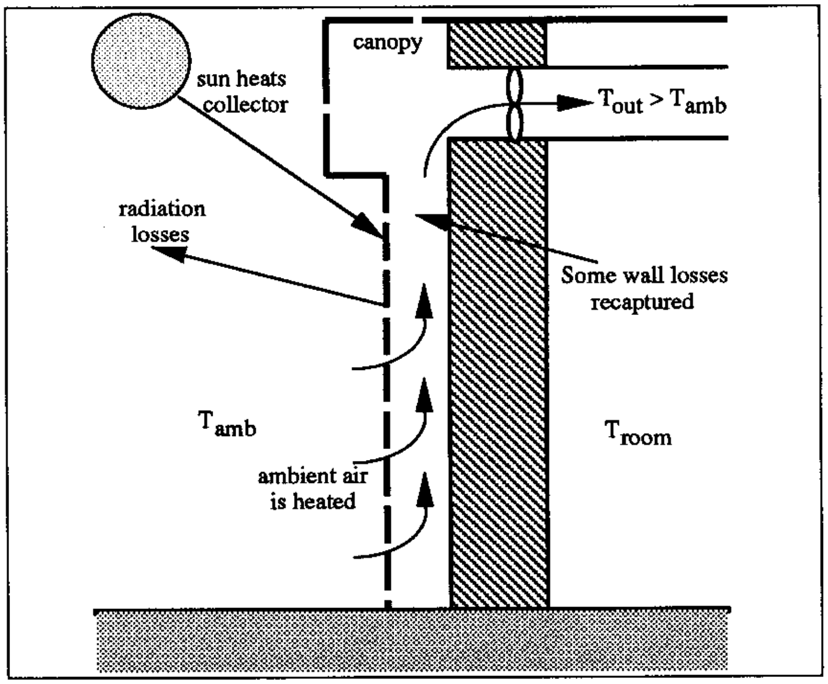


Figure 1: Schematic diagram of an unglazed transpired collector system.

This component models a UTC system, shown in Figure 2. The entire system includes the UTC plate and the building on which it is mounted. The basic energy balances are solved and the energy savings is calculated every time step for which the UTC system is operating (i.e. the bypass damper is closed). The bypass damper is opened when the ambient temperature is above the summer bypass set temperature. The bypass damper is also opened at night when the automatic night bypass is on (parameter 14).

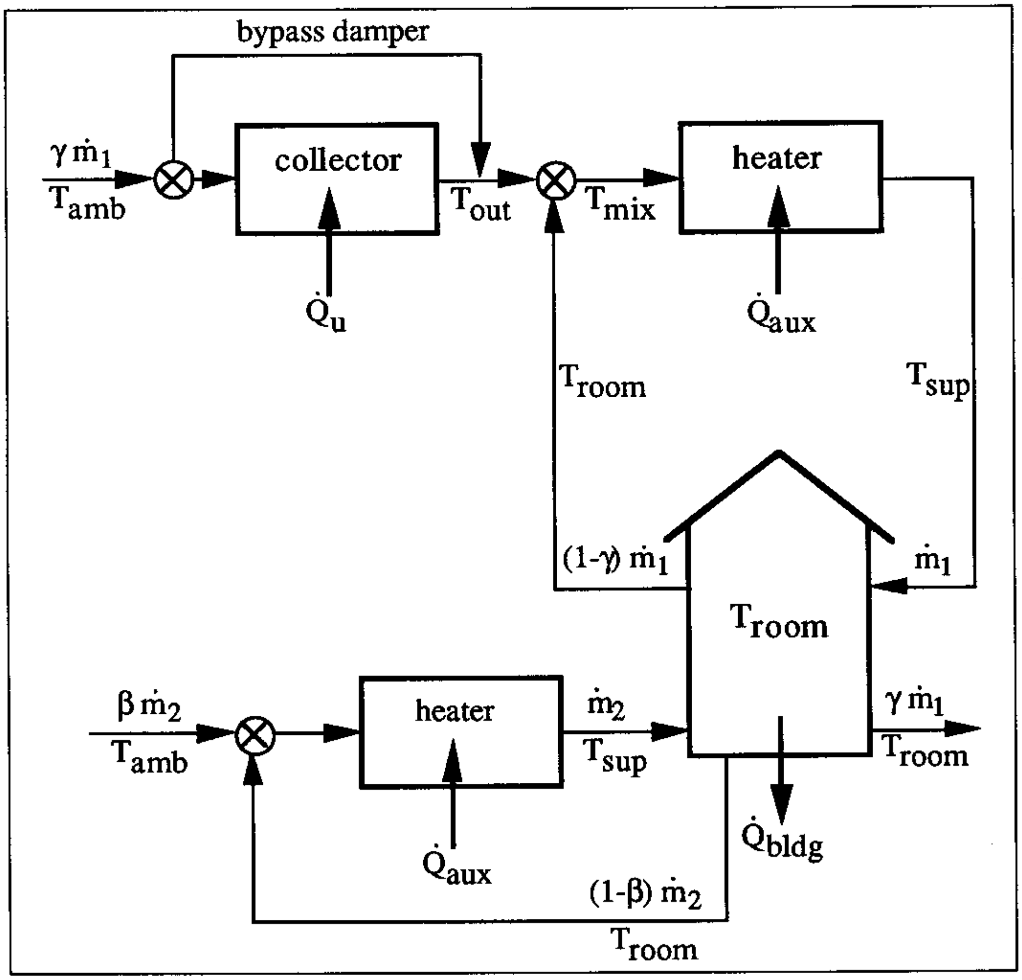


Figure 2: Complete overview of the UTC system model.

Nomenclature

A	-	total collector area (m^2)
A_s	-	collector surface area (m^2) = $(1-\sigma)$ A
c_p	-	specific heat (kJ/kg-C)
D	-	hole diameter (m)
$h_{cond,wall}$	-	coefficient for conduction through the wall (kJ/hr-m^2-C)
$h_{conv,col\text{-}air}$	-	coefficient for convection from the collector to the air (kJ/hr-m^2-C)
$h_{conv,wall\text{-}air}$	-	coefficient for convection from the outside wall surface to the air (kJ/hr-m^2-C)
I_T	-	incident solar radiation on the collector surface (kJ/hr-m^2)
$\dot{m}_{out}$	-	outdoor air mass flow rate through UTC (kg/hr) = $\gamma \dot{m}_1$
$\dot{m}_1$	-	mass flow rate of air supply from UTC system (kg/hr)
$\dot{m}_2$	-	mass flow rate of air supply from conventional system (kg/hr)

Nu_D	-	Nusselt number where hole diameter is the characteristic length
P	-	hole pitch (m)
$\dot{Q}_{abs}$	-	absorbed solar heat rate (kJ/hr)
$\dot{Q}_{aux}$	-	auxiliary heat rate (kJ/hr)
$\dot{Q}_{bldg}$	-	building heat loss rate (kJ/hr)
$\dot{Q}_{cond,wall}$	-	conduction rate through the wall (kJ/hr)
$\dot{Q}_{conv,col\text{-}air}$	-	convection rate from the collector to the air (kJ/hr)
$\dot{Q}_{conv,wall\text{-}air}$	-	convection rate from the outside wall surface to the air (kJ/hr)
$\dot{Q}_{rad,col\text{-}sur}$	-	radiation rate from the collector to the surroundings (kJ/hr)
$\dot{Q}_{rad,wall\text{-}col}$	-	radiation rate from the outside wall surface to the back of the collector (kJ/hr)
$\dot{Q}_{save}$	-	saved energy rate (kJ/hr)
$\dot{Q}_{trad}$	-	traditional heating system heat rate (kJ/hr)
$\dot{Q}_u$	-	useful energy rate (kJ/hr)
Re_D	-	Reynolds number where hole diameter is the characteristic length
T_{amb}	-	ambient air temperature (C)
T_{col}	-	collector plate temperature (C)
T_{mix}	-	mixed air temperature (C)
T_{out}	-	collector outlet air temperature (C)
T_{plen}	-	plenum air temperature (C)
T_{room}	-	room air temperature (C)
T_{sup}	-	supply air temperature (C)
T_{sur}	-	radiative surroundings temperature (C)
T_{wall}	-	outside wall surface temperature (C)
α_{col}	-	collector plate absorptivity
β	-	fraction of conventional system supply air that is outdoor air
ε_{col}	-	collector plate emissivity
ε_{HX}	-	heat exchanger effectiveness of collector
ε_{wall}	-	outside wall surface emissivity
γ	-	fraction of UTC system supply air that is outdoor air
σ	-	collector porosity
σ_{sb}	-	Stefan-Boltzmann constant ($kJ/hr\text{-}m^2\text{-}K^4$)

Mathematical Description

The first step in predicting the thermal performance of the UTC system is to calculate the outlet air temperature from the collector, T_{out}. There are four fundamental energy balance equations that are solved to find T_{out}.

$$\dot{m}_{out}\, c_p\, (T_{plen} - T_{amb}) = \dot{Q}_{conv,col\text{-}air}$$
$$\dot{m}_{out}\, c_p\, (T_{out} - T_{plen}) = \dot{Q}_{conv,wall\text{-}air}$$
$$\dot{Q}_{cond,wall} = \dot{Q}_{conv,wall\text{-}air} + \dot{Q}_{rad,wall\text{-}col}$$
$$\dot{Q}_{abs} + \dot{Q}_{rad,wall\text{-}col} = \dot{Q}_{conv,col\text{-}air} + \dot{Q}_{rad,col\text{-}sur}$$

The labelling convention that is used for heat flows is $\dot{Q}_{mode,from\text{-}to}$. So $\dot{Q}_{conv,col\text{-}air}$ is convection from the collector to the air. The useful energy from the UTC system is the sum of the convection to the air from the collector and from the outside wall.

$$\dot{Q}_u = \dot{Q}_{conv,col\text{-}air} + \dot{Q}_{conv,wall\text{-}air}$$

The rate equations for the energy flows are necessary to solve the energy balance equations. For convection from the collector to the air, an empirical heat transfer correlation is used [Kutscher, 1992].

$$Nu_D = 2.75\ (P / D)^{-1.2}\ Re_D^{0.43}$$

This correlation determines the Nusselt number based on hole diameter that is used to find $h_{conv,col\text{-}air}$. The heat exchanger effectiveness of the collector is calculated.

$$\varepsilon_{HX} = 1 - \exp(\ (h_{conv,col\text{-}air}\, A_s) / (\dot{m}_{out}\, c_p)\ )$$

This effectiveness is used in the relation between the plenum air temperature and the collector temperature.

$$\varepsilon_{HX} = (\ T_{plen} - T_{amb}\ ) / (\ T_{col} - T_{amb}\ )$$

This equation is effectively a rate equation for $\dot{Q}_{conv,col\text{-}air}$. The following rate equations are also used with the energy balances.

$$\dot{Q}_{conv,wall\text{-}air} = h_{conv,wall\text{-}air}\, A\ (\ T_{wall} - T_{plen}\ )$$
$$\dot{Q}_{cond,wall} = h_{cond,wall}\, A\ (\ T_{room} - T_{wall}\ )$$
$$\dot{Q}_{rad,wall\text{-}col} = \sigma_{sb}\, A\ (\ T_{wall}^4 - T_{col}^4\ ) / (\ 1/\varepsilon_{wall} + 1/\varepsilon_{col} - 1\ )$$
$$\dot{Q}_{abs} = \alpha_{col}\, I_T\, A_s$$
$$\dot{Q}_{rad,col\text{-}sur} = \varepsilon_{col}\, \sigma_{sb}\, A_s\ (\ T_{col}^4 - T_{sur}^4\ )$$

The outlet air from the collector is mixed with recirculated air from the building.

$$T_{mix} = \gamma\, T_{out} + (1\text{-}\gamma)\, T_{room}$$

The mixed air is heated to the necessary supply temperature to meet the heating load.

$$\dot{Q}_{aux} = \dot{m}_l\, c_p\, (T_{sup} - T_{mix})$$

The recirculation damper varies γ, the fraction of the supply air that is drawn from the outside through the collector, such that the auxiliary energy is minimized.

There are three energy savings mechanisms for a UTC system: active solar gain, recaptured wall loss, and reduced wall loss. However, the energy savings of the UTC system is not simply the sum of these three components. Fundamentally, the energy savings is the reduction in the heat required from a traditional system, which translates into a reduction of the heating bill. The heat required from an auxiliary unit of a UTC system is less than the heat required from a traditional heating system.

$$\dot{Q}_{save} = \dot{Q}_{trad} - \dot{Q}_{aux}$$

The energy savings never exceeds the heating requirements of the building with a traditional system.

TRNSYS Component Configuration

PARAMETER NO.		DESCRIPTION
1	A	- collector area (m^2)
2	ht	- collector height (m)
3	D	- collector hole diameter (m)
4	P	- collector hole pitch (m)
5	ε_{col}	- collector emissivity
6	α_{col}	- collector absorptivity
7	depth	- plenum depth (m)
8	ε_{wall}	- emissivity of wall behind collector
9	R_{wall}	- R-value of the wall behind the collector (°C-m-hr/kJ)
10	UA	- total UA-value of the building walls and roof (kJ/hr-°C)
11	T_{room}	- room air temperature (°C)
12	T_{bypass}	- ambient air temperature above which the summer bypass damper is opened (°C)
13	$Q_{aux,max}$	- maximum auxiliary heat rate available (kJ/hr)
14	Night bypass mode:	0 - bypass not automatically opened at night 1 - bypass automatically opened at night

INPUT NUMBER		DESCRIPTION
1	month	- month of the year
2	hour	- hour of the month

3	I_T	- solar radiation incident on the collector (kJ/hr-m^2)
4	T_{amb}	- ambient air temperature (°C)
5	T_{sky}	- radiative sky temperature (°C)
6	P_{atm}	- atmospheric pressure (kPa)
7	Gain	- internal building gains (kJ/hr)
8	$Flow_1$	- supply air flow rate from collector air-handling units (m^3/hr)
9	$MinOut_1$	- minimum outdoor air flow rate through collector / summer bypass damper (m^3/hr)
10	$Flow_2$	- supply air flow rate from no-collector air-handling units (m^3/hr)
11	Out_2	- outdoor air flow rate through no collector (m^3/hr)

OUTPUT NUMBER		DESCRIPTION
1	T_{plen}	- plenum air temperature (°C)
2	T_{out}	- collector outlet air temperature (°C)
3	T_{mix}	- mixed air temperature (°C)
4	T_{sup}	- supply air temperature (°C)
5	T_{col}	- collector surface temperature (°C)
6	$\dot{Q}_{save}$	- energy savings rate (kJ/hr)
7	$\dot{Q}_{conv,col-air}$	- convection from collector to air (kJ/hr)
8	$\dot{Q}_{conv,wall-air}$	- convection from wall to air (kJ/hr)
9	$\dot{Q}_{rad,col-sur}$	- radiation from collector to surroundings (kJ/hr)
10	$\dot{Q}_{rad,wall-col}$	- radiation from wall to collector (kJ/hr)
11	$\dot{Q}_{cond,wall}$	- conduction through wall (kJ/hr)
12	$\dot{Q}_{red,wall}$	- reduced conduction through wall because of collector (kJ/hr)
13	$\dot{Q}_{abs}$	- absorbed energy rate (kJ/hr)
14	$\dot{Q}_{aux}$	- auxiliary energy rate (kJ/hr)
15	$\gamma Flow_1$	- outdoor air flow rate through collector / summer bypass damper (m^3/hr)
16	ε_{HX}	- heat exchanger effectiveness of collector
17	η_{solar}	- solar efficiency of collector
18	ΔP_{col}	- pressure drop across collector plate (kPa)

19 Bypass damper position:

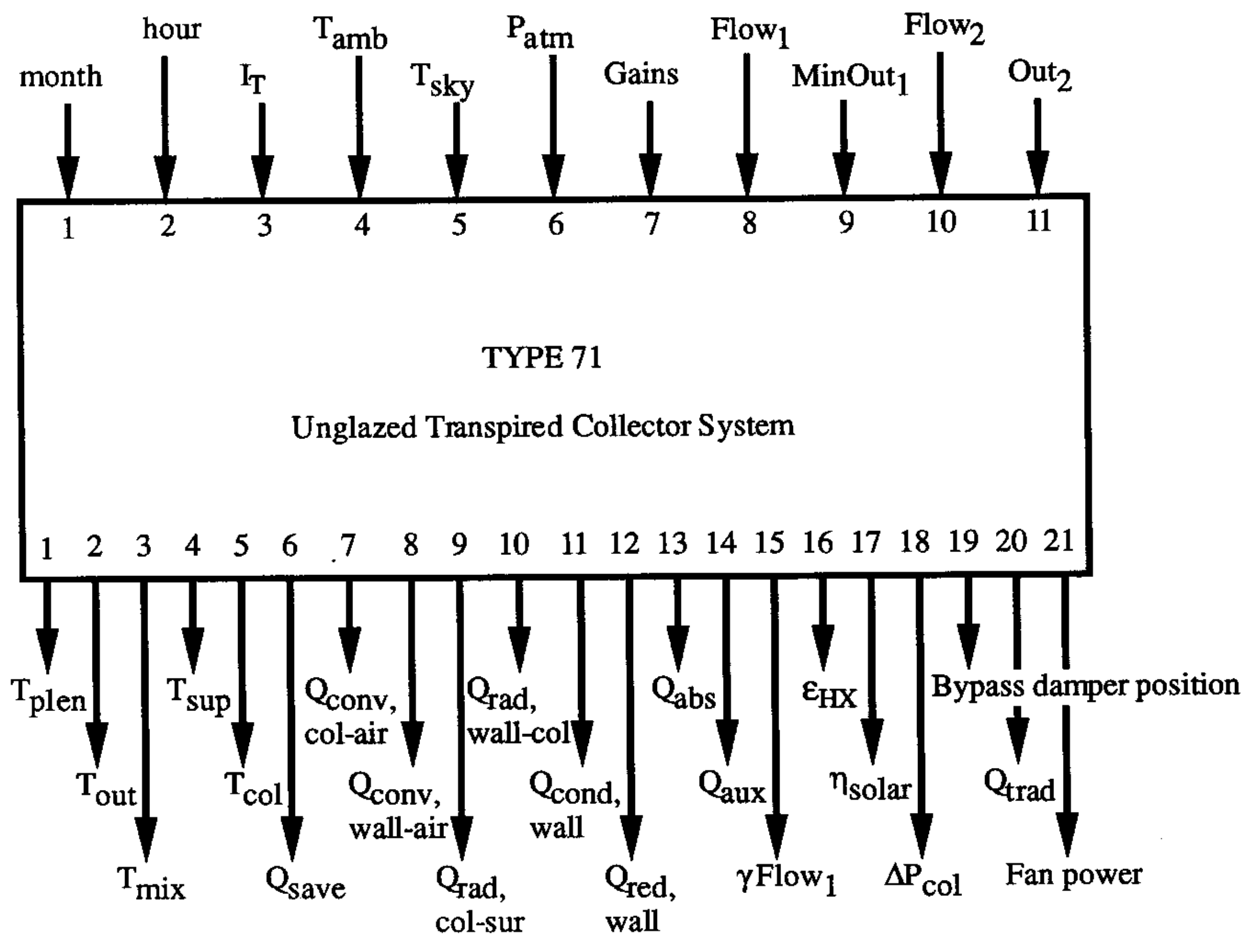
0 - bypass damper is open

1 - bypass damper is closed

20 $\dot{Q}_{trad}$ - heat rate supplied by a traditional heating system (kJ/hr)

21 Fan power - additional fan power required (kJ/hr)

Information Flow Diagram



Parameters:

1. A
2. ht

8. ε_{wall}
9. R_{wall}

3. D
4. P
5. ε_{col}
6. α_{col}
7. depth
10. UA
11. T_{room}
12. T_{bypass}
13. $Q_{aux,max}$
14. Night bypass mode

References

1. Summers, David N., Thermal Simulation and Economic Assessment of Unglazed Transpired Collectors, M.S. Thesis in Mechanical Engineering, University of Wisconsin-Madison, 1995.
2. Kutscher, Charles F., An Investigation of Heat Transfer for Air Flow Through Low Porosity Perforated Plates, Ph.D. Thesis in Mechanical Engineering, University of Colorado, 1992.