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, dt, tpar, info, icntr1, *)

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

* Unglazed transpired collector (UTC) subroutine.
*
* Reference: Kutscher, C.F., "Heat Exchanger Effectiveness and
*             Pressure Drop for Air Flow Through Perforated Plates With and
*
* 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,013) unit, type, ifail, nint(month), nint(hour)

713 format ('/lx,'****** ERROR ******',/lx,'UNIT ',i3,
  & ' TYPE ',i3,' UNGLIZED TRANSPRED COLLECTOR',/lx,
  & 'MATRIX SOLVER ERROR, IFAIL = ',i3,/lx,
  & '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.
implicit none

* type71 variables

* parameters
real*8 area, ht, diam, pitch, emisc, absor, depth, emisw, rwall
real*8 ua, troom, thbypass, 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, tgrid, tsur, qblend
real*8 aircp, aircond, airvisc, airden, a0, a1, a2
real*8 mflow1, mflow2, mflow, qmin, beta, qrad

* outputs and other variables
real*8 gamma, tout, tcol, tplen, twall, qvcol, qrcol, qvwall
real*8 qrwall, qdwall, red, effhx, appvel
real*8 tmix, tsep, qa ux1, lowg, hig, oldg, dif, diflim, small, g
real*8 zeta, pcoll, plenden, f, dh, plenvel, pfri c, pbuoy, pacc
real*8 delp, fanpw
real*8 qaux2, qua x, film, tso al, qpot, qred, qu, qsave, sol efl
integer istr, i, j, jmax
real warn
character*10 warnfile

* TRNSYS variables

character*3 ycheck, ocheck
real time, t, dt dt, par, s, time0, tfinal, delt
double precision xin, out
integer*4 info, unit, type
integer np, ni, no, icntr1, iwarn, nstore, iav, iunit
integer iur, iuw, iform, iuk

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 utc solve 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 /units/ lur, luw, ifrom, 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','PW1','PW1','PW1','PW1','VF1','VF1','DM1','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)
h = par(2)
diam = par(3)
pitch = par(4)
emisc = par(5)
absor = par(6)
depth = par(7)
emisw = par(8) 
rmall = par(9) 
ua = par(10) 
troom = par(11) 
th bypass = par(12) 
gmax = 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 (1uw,810) unit, type
        format (/1x, '***** ERROR *****',/1x, 'UNIT ',i3,
        & ' TYPE ',i3,' UNGLARED TRANSPARED 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
endo
do = tout
out(15) = minout
out(20) = qrad
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, qwall, grwall, 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, qwall, grwall, 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, qwall, grwall, qdwall,
&                red, effhx, appvel )
tmix = gamma*tout + (1.0 - gamma)*troom
            go to 100
        end
if ( tmix .lt. tsep ) 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)
    format (//1x,'***** ERROR *****',/1x,'UNIT ',i3,
    & ' TYPE ',i3,' UNGLARED TRANSPURRED 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 )
pool  = 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 + pfrc - 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 gbldg 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 + qwwall    ! useful energy gained by air
qsave = qrad - 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
format (/,'month = ',f3.0,' hour = ',f5.1)
write (71,802) appvel
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
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
  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
  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
    format ('//2x,'**** WARNING ***** UNIT',i3,' TYPE',i3/4x,
    & 'CHECK ',a10,
    & ' FOR UNGLAZED TRANSPIERED COLLECTOR WARNINGS')
    s(istr) = 1.0
    iwarn = iwarn + 1
  endif
endif
30 continue

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

out(1) = tlen
out(2) = tout
out(3) = tmix
out(4) = tsup
out(5) = tcol
out(6) = qsave
out(7) = qvcol
out(8) = qwall
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, qwall, 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, qwall
  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, gabs, tsur
  real*8 aircp, aircond, airvisc, aireden, 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, gabs, tsur,
  &  aircp, aircond, airvisc, aireden, 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
    qwall = 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)
pivel = 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 * sufa / (gamma * mflow1 * aircp) )

* calculate heat transfer coefficient for wall to air convection

pr = 0.71
reht = airden * pivel * 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
endo
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(1uw,815) unit, type, dif, nint(month), nint(hour)
        format(//1x, '***** ERROR *****',/lx, 'UNIT ',i3,
        & ' TYPE ',i3, ' UNGL AZED TRANSPR E COLLECTOR',/lx,
        & 'NO CONVERGENCE IN K LOOP, DIF = ',e7.2,/lx,
        & 'MONTH = ',i2,' HOUR = ',i3)
        stop
    endif

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

tcol = x(1)
tplen = x(2)
qrcol = x(3)
qwall = 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 TRANSPIERED 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.

![Diagram of an unglazed transpired collector system](image)

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$^2$) = $(1-\alpha)A$
- $c_p$ - specific heat (kJ/kg-C)
- $D$ - hole diameter (m)
- $h_{\text{cond,wall}}$ - coefficient for conduction through the wall (kJ/hr-m$^2$-C)
- $h_{\text{conv,collect-air}}$ - coefficient for convection from the collector to the air (kJ/hr-m$^2$-C)
- $h_{\text{conv,wall-air}}$ - 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$^2$)
- $\dot{m}_{\text{out}}$ - 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)
\[ \text{Nu}_{D} \quad \text{P} \quad \dot{Q}_{\text{abs}} \quad \dot{Q}_{\text{aux}} \quad \dot{Q}_{\text{bldg}} \quad \dot{Q}_{\text{cond,wall}} \quad \dot{Q}_{\text{conv, col-air}} \quad \dot{Q}_{\text{conv, wall-air}} \quad \dot{Q}_{\text{rad, col-sur}} \quad \dot{Q}_{\text{rad, wall-col}} \quad \dot{Q}_{\text{save}} \quad \dot{Q}_{\text{trad}} \quad \dot{Q}_{u} \quad Re_{D} \quad T_{\text{amb}} \quad T_{\text{col}} \quad T_{\text{mix}} \quad T_{\text{out}} \quad T_{\text{plen}} \quad T_{\text{room}} \quad T_{\text{sup}} \quad T_{\text{sur}} \quad T_{\text{wall}} \quad \alpha_{\text{col}} \quad \beta \quad \varepsilon_{\text{col}} \quad \varepsilon_{HXX} \quad \varepsilon_{\text{wall}} \quad \gamma \quad \sigma \quad \sigma_{\text{Stefan-Boltzmann}} \]

**Mathematical Description**

The first step in predicting the thermal performance of the UTC system is to calculate the outlet air temperature from the collector, \( T_{\text{out}} \). There are four fundamental energy balance equations that are solved to find \( T_{\text{out}} \):
\[ \dot{m}_{\text{out}} \, c_p \, (T_{\text{pl}} - T_{\text{amb}}) = \dot{Q}_{\text{conv, col-air}} \]
\[ \dot{m}_{\text{out}} \, c_p \, (T_{\text{out}} - T_{\text{pl}}) = \dot{Q}_{\text{conv, wall-air}} \]
\[ \dot{Q}_{\text{cond, wall}} = \dot{Q}_{\text{conv, wall-air}} + \dot{Q}_{\text{rad, wall-col}} \]
\[ \dot{Q}_{\text{abs}} + \dot{Q}_{\text{rad, wall-col}} = \dot{Q}_{\text{conv, col-air}} + \dot{Q}_{\text{rad, col-sur}} \]

The labelling convention that is used for heat flows is \text{Q}_{\text{mode, from-to}}. So \dot{Q}_{\text{conv, col-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.

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

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].

\[ \text{Nu}_D = 2.75 \, (P / D)^{-1.2} \, \text{Re}_D^{0.43} \]

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

\[ \varepsilon_{\text{HX}} = 1 - \exp \left( \frac{(h_{\text{conv, col-air}} \, A_s)}{(\dot{m}_{\text{out}} \, c_p)} \right) \]

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

\[ \varepsilon_{\text{HX}} = \frac{(T_{\text{pl}} - T_{\text{amb}})}{(T_{\text{col}} - T_{\text{amb}})} \]

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

\[ \dot{Q}_{\text{conv, wall-air}} = h_{\text{conv, wall-air}} \, A \, (T_{\text{wall}} - T_{\text{pl}}) \]

\[ \dot{Q}_{\text{cond, wall}} = h_{\text{cond, wall}} \, A \, (T_{\text{room}} - T_{\text{wall}}) \]

\[ \dot{Q}_{\text{rad, wall-col}} = \sigma_{\text{sb}} \, A \, (T_{\text{wall}}^4 - T_{\text{col}}^4) / (1 / \varepsilon_{\text{wall}} + 1 / \varepsilon_{\text{col}} - 1) \]

\[ \dot{Q}_{\text{abs}} = \alpha_{\text{col}} \, I_T \, A_s \]

\[ \dot{Q}_{\text{rad, col-sur}} = \varepsilon_{\text{col}} \, \sigma_{\text{sb}} \, A_s \, (T_{\text{col}}^4 - T_{\text{sur}}^4) \]

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

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

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

\[ \dot{Q}_{\text{aux}} = \dot{m}_1 \, c_p \, (T_{\text{sup}} - T_{\text{mix}}) \]

The recirculation damper varies \( \gamma \), 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.

\[ Q_{\text{save}} = Q_{\text{trad}} - Q_{\text{aux}} \]

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

---

**TRNSYS Component Configuration**

<table>
<thead>
<tr>
<th>PARAMETER NO.</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A - collector area (m²)</td>
</tr>
<tr>
<td>2</td>
<td>ht - collector height (m)</td>
</tr>
<tr>
<td>3</td>
<td>D - collector hole diameter (m)</td>
</tr>
<tr>
<td>4</td>
<td>P - collector hole pitch (m)</td>
</tr>
<tr>
<td>5</td>
<td>( \varepsilon_{\text{col}} ) - collector emissivity</td>
</tr>
<tr>
<td>6</td>
<td>( \alpha_{\text{col}} ) - collector absorptivity</td>
</tr>
<tr>
<td>7</td>
<td>depth - plenum depth (m)</td>
</tr>
<tr>
<td>8</td>
<td>( \varepsilon_{\text{wall}} ) - emissivity of wall behind collector</td>
</tr>
<tr>
<td>9</td>
<td>( R_{\text{wall}} ) - R-value of the wall behind the collector (°C-m-hr/kJ)</td>
</tr>
<tr>
<td>10</td>
<td>UA - total UA-value of the building walls and roof (kJ/hr-°C)</td>
</tr>
<tr>
<td>11</td>
<td>T_{\text{room}} - room air temperature (°C)</td>
</tr>
<tr>
<td>12</td>
<td>T_{\text{bypass}} - ambient air temperature above which the summer bypass damper is opened (°C)</td>
</tr>
<tr>
<td>13</td>
<td>Q_{\text{aux, max}} - maximum auxiliary heat rate available (kJ/hr)</td>
</tr>
<tr>
<td>14</td>
<td>Night bypass mode:</td>
</tr>
<tr>
<td></td>
<td>0 - bypass not automatically opened at night</td>
</tr>
<tr>
<td></td>
<td>1 - bypass automatically opened at night</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>INPUT NUMBER</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>month - month of the year</td>
</tr>
<tr>
<td>2</td>
<td>hour - hour of the month</td>
</tr>
<tr>
<td>OUTPUT NUMBER</td>
<td>DESCRIPTION</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>1</td>
<td>$T_{plen}$ - plenum air temperature (°C)</td>
</tr>
<tr>
<td>2</td>
<td>$T_{out}$ - collector outlet air temperature (°C)</td>
</tr>
<tr>
<td>3</td>
<td>$T_{mix}$ - mixed air temperature (°C)</td>
</tr>
<tr>
<td>4</td>
<td>$T_{sup}$ - supply air temperature (°C)</td>
</tr>
<tr>
<td>5</td>
<td>$T_{col}$ - collector surface temperature (°C)</td>
</tr>
<tr>
<td>6</td>
<td>$\dot{Q}_{save}$ - energy savings rate (kJ/hr)</td>
</tr>
<tr>
<td>7</td>
<td>$\dot{Q}_{conv, col-air}$ - convection from collector to air (kJ/hr)</td>
</tr>
<tr>
<td>8</td>
<td>$\dot{Q}_{conv, wall-air}$ - convection from wall to air (kJ/hr)</td>
</tr>
<tr>
<td>9</td>
<td>$\dot{Q}_{rad, col-surf}$ - radiation from collector to surroundings (kJ/hr)</td>
</tr>
<tr>
<td>10</td>
<td>$\dot{Q}_{rad, wall-col}$ - radiation from wall to collector (kJ/hr)</td>
</tr>
<tr>
<td>11</td>
<td>$\dot{Q}_{cond, wall}$ - conduction through wall (kJ/hr)</td>
</tr>
<tr>
<td>12</td>
<td>$\dot{Q}_{red, wall}$ - reduced conduction through wall because of collector (kJ/hr)</td>
</tr>
<tr>
<td>13</td>
<td>$\dot{Q}_{abs}$ - absorbed energy rate (kJ/hr)</td>
</tr>
<tr>
<td>14</td>
<td>$\dot{Q}_{aux}$ - auxiliary energy rate (kJ/hr)</td>
</tr>
<tr>
<td>15</td>
<td>$\gamma F_{\text{Flow}_1}$ - outdoor air flow rate through collector / summer bypass damper (m³/hr)</td>
</tr>
<tr>
<td>16</td>
<td>$\varepsilon_{HX}$ - heat exchanger effectiveness of collector</td>
</tr>
<tr>
<td>17</td>
<td>$\eta_{\text{solar}}$ - solar efficiency of collector</td>
</tr>
<tr>
<td>18</td>
<td>$\Delta P_{\text{col}}$ - pressure drop across collector plate (kPa)</td>
</tr>
</tbody>
</table>
Bypass damper position:
0 - bypass damper is open
1 - bypass damper is closed

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

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

Information Flow Diagram

TYPE 71
Unglazed Transpired Collector System

Parameters:
1. A
2. ht
3. T_{\text{plen}}
4. T_{\text{sup}}
5. Q_{\text{conv, col-air}}
6. Q_{\text{rad, wall-col}}
7. Q_{\text{save}}
8. \epsilon_{\text{wall}}
9. R_{\text{wall}}
3. D
4. P
5. $\varepsilon_{\text{col}}$
6. $\alpha_{\text{col}}$
7. depth
8. $\Delta T$
9. $I_{\text{out}}$
10. UA
11. Troom
12. Tbypass
13. Qaux,max
14. Night bypass mode

References
