APPENDIX D

TYPE 69 (Detailed Model): Description, TRNSYS Decks and Code

DESCRIPTION:

TYPE 69 models a NCHE water loop based upon geometric parameters of the heat exchanger. TYPE 69 takes into account both the heat transfer and pressure effects in order to predict the water flow rate, the HX outlet water temperatures, and the heat transfer in the heat exchanger. TYPE 69 was written for manufacturers and researchers, so that given the geometric specifications on a particular shell and coil or concentric tube counterflow NCHE, a SDHW system's performance can be predicted. TYPE 69 is limited to shell and coil and concentric tube counterflow NCHE geometries.

Requirements:

In order to run TYPE 69, the following inputs must be specified:

1) Type and percent glycol solution used in collector loop (either propylene glycol or ethylene glycol solutions may be used.)

2) Dimensions of all piping in water loop: length, diameter, change in elevation.

3) Good estimate of minor losses for piping fittings, valves and entrance, exit conditions in water loop.

4) Geometric Specifications of shell and coil or concentric tube counterflow NCHE.

1) Dimensions of all piping in water loop

In describing the physical aspects of the SDHW system, the following diagram should make clear the various parameter inputs:

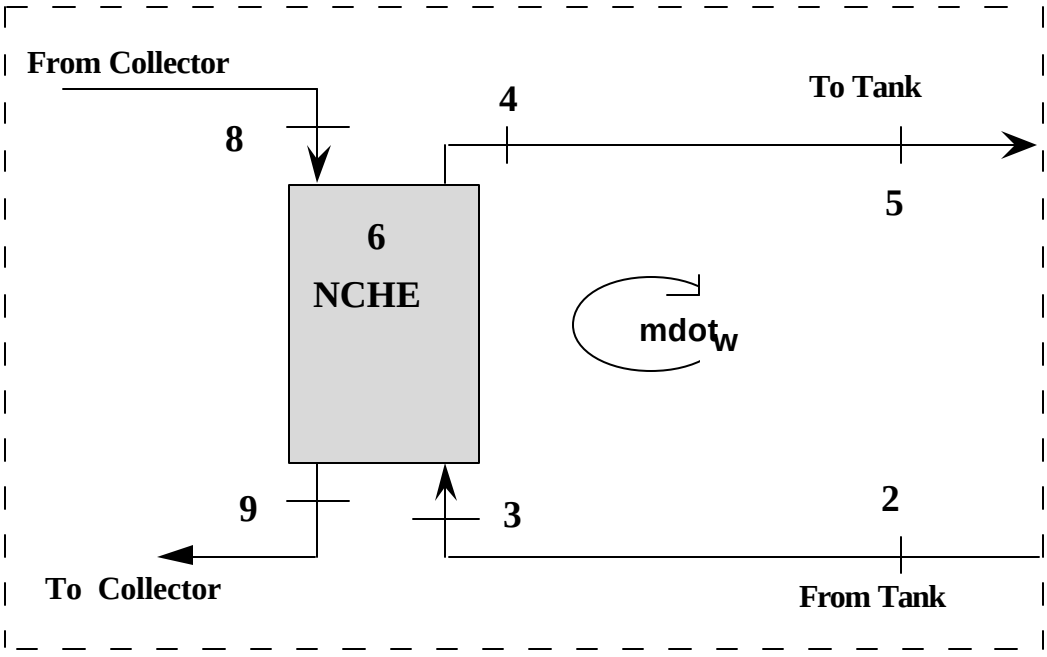

Figure D.1 Diagram of NCHE and associated piping.

TYPE 69 allows for two different diameter pipes in the piping runs from tank to NCHE and from NCHE to the tank. If only one diameter pipe is used from the tank to the NCHE, then input 0's into the TRNSYS deck for the respective positions.

Care must be taken when specifying *Dz* values as parameters. If $\sum_{water\ loop} \Delta z \neq 0$ then TRNSYS will deliver strange answers. That is, the following equation should hold:

$$\Delta z_2 + \Delta z_3 + \Delta z_4 + \Delta z_5 + \Delta z_{NCHE} + \Delta z_{Tank} = 0 \qquad \text{(D.1)}$$

where Δz_{Tank} is the distance from the tank inlet piping to the tank outlet piping.

2) Minor loss coefficients for water loop fittings

The minor losses can be found either experimentally, with Hooper's correlations for fitting pressure losses (Kakic 1987), or from table values.

3) Geometric specifications for NCHE--Shell and Coil NCHE

In order to use the crossflow correlations in TYPE 69, the following assumptions are made, and are used for every ensemble of geometric parameters:

1) The shell of the NCHE will always be placed a distance of $S_T/2$ from the outermost coil's outer diameter (see Figure D.2). This spacing is necessary for the crossflow heat transfer correlations to be valid.
2) The core of the NCHE will always be placed a distance of $S_T/2$ from the innermost coil's inner diameter. This spacing is necessary for the crossflow heat transfer correlations to be valid.
3) The specified pitch is constant for all coils in the NCHE.

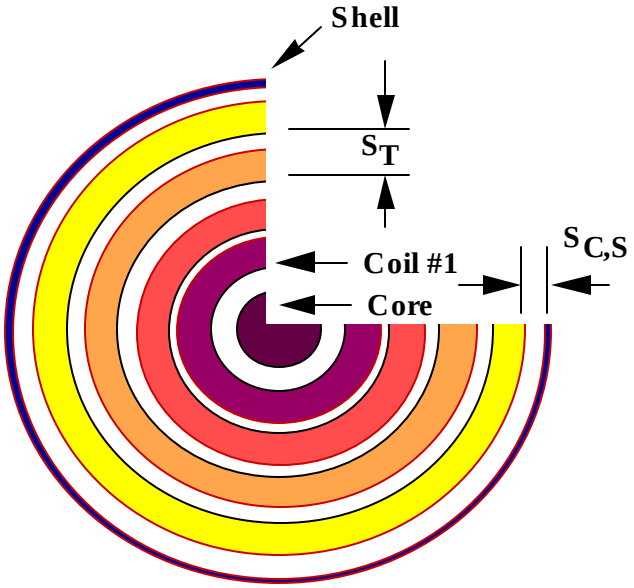

Figure D.2 Diagram of some NCHE geometric parameters. S_T is the

average distance between coils. $S_{C,S}$ ($S_{C,S} = S_T/2$) is the distance between the outermost coil's outer diameter and the inner diameter of the heat exchanger shell.

As models of NCHEs often have exterior water inlet and outlet piping associated with them, it is necessary to account for the pressure loss in these piping sections. As the lengths typically are small, only the minor loss coefficients are considered. Figure D.3 presents typical minor loss conditions associated with a NCHE. Parameters 20 and 21 account for the entrance and exit conditions for the NCHE. Figure D.4 presents some typical minor loss coefficients.

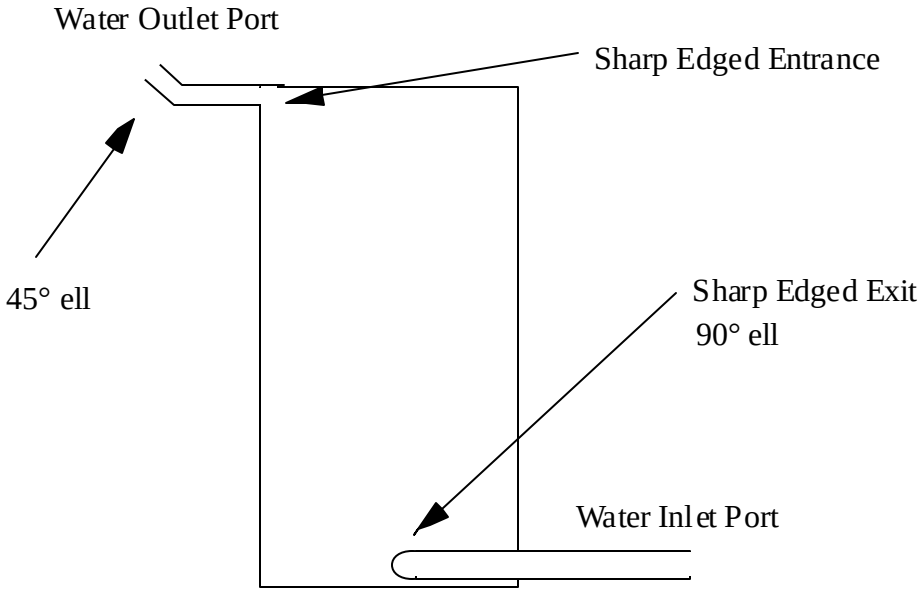

Figure D.3 Minor losses associated with NCHE.

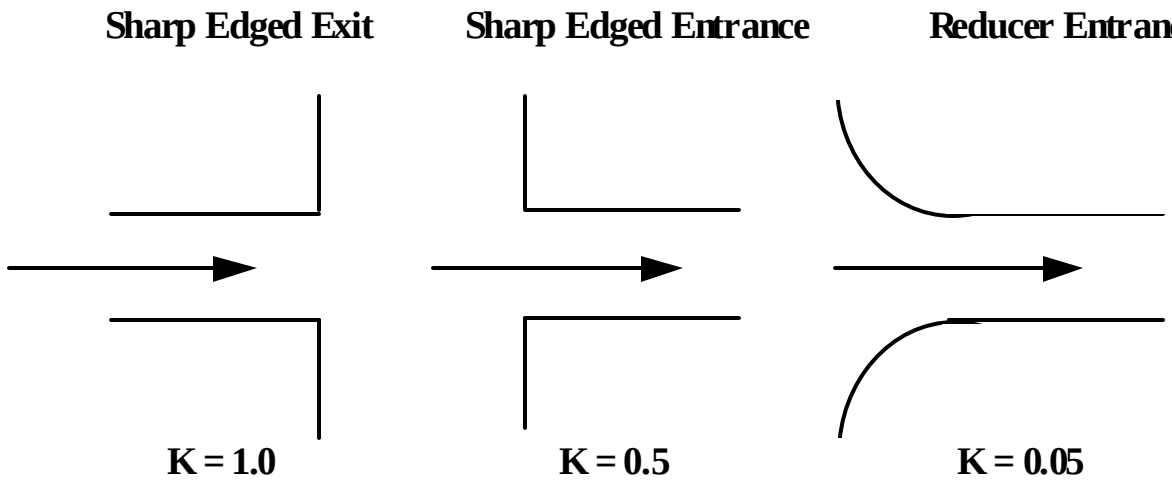

Figure D.4 Entrance and exit minor loss K values found in tables. Minor loss coefficient tables assume turbulent flow (Cheremisinoff, 1981).

The ells shown in Figure D.3 are evaluated using Hooper's correlation (Kakic 1987). Parameters K_1 and K_8 for the ells can be found using Table D.1.

Table D.1 Hooper's Correlation K Constants.

Bend	Fitting Radius	R/a	Type	K1	Kinf
90	standard	2	screwed	800	0.4
90	standard	2	flanged, welded	800	0.25
90	long radius	3	all types	800	0.2
45	standard	2	all types	500	0.2
45	long radius	3	all types	500	0.15

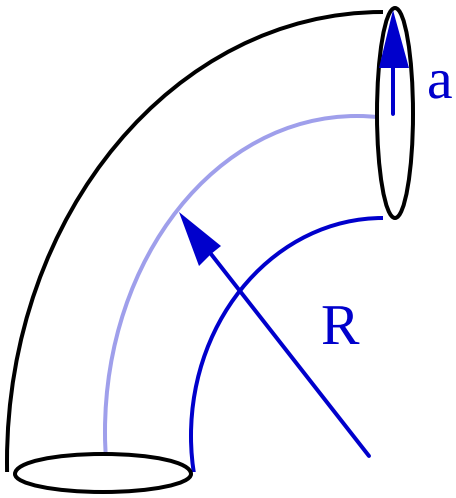

Figure D.5 R and a values for pipe ells

When using TYPE 69, it should be kept in mind that there is a limitation to the minimum pitch that can be specified. The pitch cannot be smaller than the tube diameter, as this would be physically impossible.

4) Regarding the Tank Model

As TYPE 67 requires as an input the static pressure gain in the tank, the TYPE 4 tank model must be ammended or an alternate tank model (that outputs the tank static pressure gain) must be used in

place of TYPE 4. The changes that must be made to the TYPE 4 code are listed below. Changes or additions to the code are in boldface.

```
310   DTDT(K) = (S(KF) - S(KI))/DELT
      TF = TF/HIGH

320   QTANK = FLWL*(S(AVG+1) - TL)
      QIN=FLWS*(TIN-S(AVG+NEQ))
      DELAU=(TF-S(IS+1))*MSCP
C-----------------------------------------------
C  COMPUTE STATIC HEAD LOSS
C-----------------------------------------------
        G=9.806
      if (n.lt.2) then
         Rho=(999.8396 + 18.224944*TF
     @              - 0.00792221*TF**2
     @              -55.44846e-6*TF**3
     @              +149.7562e-9*TF**4
     @              -393.2952e-12*TF**5)
     @              /(1.d0+18.159725e-3*TF)

        else
           rhot=0.d0
           DO  I=2,N
              Rhot=rhot+(999.8396 + 18.224944*s(avg+i)
     @              - 0.00792221*s(avg+i)**2
     @              -55.44846e-6*s(avg+i)**3
     @              +149.7562e-9*s(avg+i)**4
     @              -393.2952e-12*s(avg+i)**5)
     @              /(1.d0+18.159725e-3*s(avg+i))

           enddo
           rho=rhot/(n-1)
        endif
           STHEAD=RHO*G*HIGH*.001
C-----------------------------------------------

C  SET OUTPUTS
 330  OUT(1)=S(AVG+NEQ)
      OUT(2)=FLWSS
      OUT(3)=S(AVG+1)
      OUT(4)=FLWLL
      OUT(5)=QENV+QVENT
      OUT(6)=QTANK
      OUT(7)=DELAU
      OUT(8)=QBTOT(1)+QBTOT(2)
      OUT(9)=QBTOT(1)
      OUT(10)=QBTOT(2)
      OUT(11)=QIN
```

```
      OUT(12)=TF
     OUT(13)=STHEAD


      N=NEQ-1
      IF(N.LT.2) RETURN 1
      DO 350 I=2,N
350   OUT(12+I)=S(AVG+I)
      RETURN 1

1001  FORMAT(//2X,'***** WARNING *****',/2X,'THE SET POINT TEMERATURE OF
     1 THE UNIT ',I2,' TYPE 4 STORAGE TANK IS HIGHER THAN'/2X,'THE BOILI
     1NG TEMPERATURE.  THE BOILING TEMPERATURE WILL BE USED AS THE SET',
     1/2X,'POINT.')

      END
```

4) The Convergence Promoter

TYPE 69 requires a TYPE 44 convergence promoter in order to reach convergence at some time steps. The convergence promoter should be set to vary the glycol inlet temperature of the NCHE, (the first input to NCHE).

TRNSYS Component Configuration

PARAMETER NO.		DESCRIPTION
1	mode	Type geometry of NCHE: shell and coil = 1 concentric tube counterflow = 2
2	m_g	Flow rate in collector loop [kg/s]
3	H_{hx}	Height of HX shell [m]
Length of Pipe [m]:		
4	L(2)	of pipe section 2
5	L(3)	of pipe section 3
6	L(4)	of pipe section 4
7	L(5)	of pipe section 5
ΔZ of Pipe [m]:		
8	dz(2)	of pipe section 2
9	dz(3)	of pipe section 3
10	dz(4)	of pipe section 4
11	dz(5)	of pipe section 5
Diameter of Pipe [m]:		
12	D(2)	of pipe section 2
13	D(3)	of pipe section 3
14	D(4)	of pipe section 4
15	D(5)	of pipe section 5
Associated Minor Loss K Values [-]:		
16	K(2)	of pipe section 2
17	K(3)	of pipe section 3
18	K(4)	of pipe section 4
19	K(5)	of pipe section 5
Minor Loss K Values for Inlet and Exit Conditions of NCHE [-]:		
20	K_{exit}	of water inlet to NCHE
21	K_{entr}	of water exit from NCHE
Glycol Type, Percentage:		
22	Glycol	Type of glycol solution [-]:

		1 = propylene glycol solution 2 = ethylene glycol solution
23	%	Percentage glycol of solution [%]

For Mode 1 (Shell and Coil NCHE):

24	D_{tubes}	Outer diameter of tubes in the NCHE [m]
25	N_{coils}	Number of helices in the NCHE [-]
26	δ_w	thickness of tube wall in the NCHE [m]
27	Pitch	Pitch of helices in NCHE [m]
28	D_{exit}	Inner diameter of pipe corresponding to exit condition minor loss [m]
29	D_{inlet}	Inner diameter of pipe corresponding to inlet condition minor loss [m]
30	N_{bends}	# of ells associated with piping that is attached to the shell of the NCHE [-]

Inner Diameters of Coils in NCHE [m]
List in ascending order. (For a n-coil NCHE, $D_{coil,n+1}$ - $D_{coil,10}$ do not affect calculations)

31	$D_{coil,1}$	Coil #1
32	$D_{coil,2}$	Coil #2
33	$D_{coil,3}$	Coil #3
34	$D_{coil,4}$	Coil #4
35	$D_{coil,5}$	Coil #5
36	$D_{coil,6}$	Coil #6
37	$D_{coil,7}$	Coil #7
38	$D_{coil,8}$	Coil #8
39	$D_{coil,9}$	Coil #9
40	$D_{coil,10}$	Coil #10

Parameters Associated with Minor Loss Conditions for Exterior Piping of NCHE:

Diameter of piping associated with each exterior NCHE bend

40+1	$D_{bend,1}$	Inner diameter of piping associated with 1st bend
40+2	$D_{bend,2}$	Inner diameter of piping associated with 2nd bend
...		
40+N_{bends}	$D_{bend,N}$	Inner diameter of piping associated with Nth bend

Position of piping associated with each exterior bend: 3 represents inlet piping
4 represents outlet piping

$40+N_{bends}+1$	$Pos_{bend,1}$	Position of piping associated with 1st bend
$40+N_{bends}+2$	$Pos_{bend,2}$	Position of piping associated with 2nd bend
...		
$40+2 \times N_{bends}$	$Pos_{bend,N}$	Position of piping associated with Nth bend

K_1 value associated with each exterior bend [-]:

$40+2 \times N_{bends}+1$	$K_{1,1}$	K_1 value associated with 1st bend
$40+2 \times N_{bends}+2$	$K_{1,2}$	K_1 value associated with 2nd bend
...		
$40+3 \times N_{bends}$	$K_{1,N}$	K_1 value associated with Nth bend

K_8 value associated with each exterior bend [-]:

$40+3 \times N_{bends}+1$	$K_{8,1}$	K_8 value associated with 1st bend
$40+3 \times N_{bends}+2$	$K_{8,2}$	K_8 value associated with 2nd bend
...		
$40+4 \times N_{bends}$	$K_{8,N}$	K_8 value associated with Nth bend

For Mode 2 (Concentric Tube Counterflow NCHE):

24	D_{inner}	Inner diameter of inner tube in the NCHE [m]
25	D_{outer}	Inner diameter of outer tube in the NCHE [m]
26	δ_w	thickness of inner tube wall in the NCHE [m]

INPUT NUMBER		**DESCRIPTION**
1	$T_{w,i}$	Temperature of inlet water stream to HX [°C]
2	$T_{g,i}$	Temperature of inlet glycol stream to HX [°C]
3	$\Delta P_{st,tank}$	Static pressure head in tank [kPa]
4	γ	Control Function from controller

OUTPUT NUMBER		**DESCRIPTION**
1	$T_{w,o}$	Water outlet temperature [°C]

2	$T_{g,o}$	Glycol outlet temperature [°C]
3	Q_{hx}	Energy transfer rate in HX [kJ/hr]
4	e'	HX modified effectiveness
5	m_w	Water flow rate [kg/hr]
6	$\sum \Delta P_{static}$	Total static presure drop around water loop [kPa]
7	$\sum \Delta P_{shear}$	Total shear pressure drop around water loop [kPa]
8	Error	Error = $\sum \Delta P_{static} - \sum \Delta P_{shear}$
9	UA	UA product in NCHE [W/°C]
10	$\Delta P_{sh,HX}$	Shear pressure drop in NCHE [kPa]

Information Flow Diagram

mode 1:

Inputs	-	4
Outputs	-	10
Parameters	-	26
Derivatives	-	0

mode 2:

Inputs	-	4
Outputs	-	10
Parameters	-	40 + 4 x N_{bends}
Derivatives	-	0

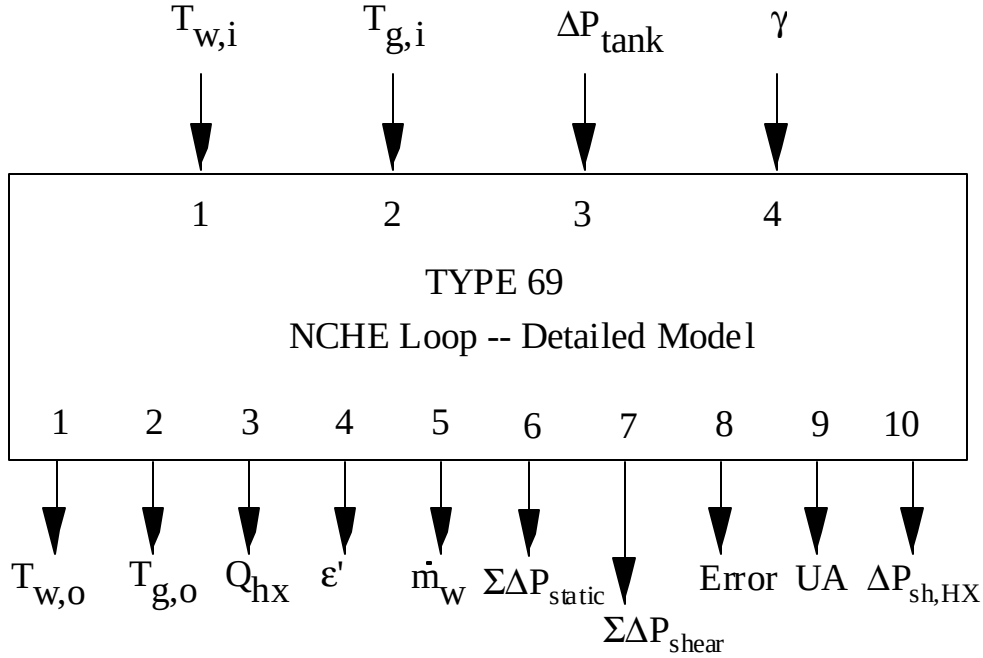

Figure D.6 TRNSYS component diagram for detailed model.

DECKS:

1) Steady State Deck (Mode 1)

```
ASSIGN STEADYSTATE.OUT   14
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                             *
*                            steady state                                     *
*               NATURAL CONVECTION HEAT EXCHANGER SETUP                       *
*                            DECEMBER 1994                                    *
*                      Detailed model, mode 1                                 *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
limits 100 100
SIMULATION 25. 90. 1
WIDTH 72
tol -0.0001 0.0001

equations 3
THXWI=19
THXGI=TIME
MWS = [68,2]/3600

equation 1
* NCHE mode: 1) shell and coil HX
*            2) concentric tube counterflow HX
mode69=1

Equations 14 --------Variable Geometric HX Parameters-----------------
dtube=.25*2.54/100
pitch=.35*2.54/100
hthx=16*2.54/100
Ncoils=2
*
* coil diameters
*
d1=1.*2.54/100
d2=.04445
d3=.0635
d4=.08255
d5=.1016
d6=.12065
d7=.1397
d8=.15875
d9=.1778
```

```
d10=.19685

equations 4 --------- PARAMETERS COMMON TO BOTH MODES-----------------
*
* type of glycol solution: 1=propylene glycol
*                          2=ethylene glycol
glycol=1
*
* percent propylene glycol solution
*
percent=40
*
* HX inlet, outlet minor loss coefficients
*
Kinlet=0.4
Kexit=1

equations 9 -----GEOMETRIC PARAMETERS OF HX INLET OUTLET TUBING--------
*
* diameters of ells which are part of NCHE
*
DLin=.5*2.54/100
DLout=.75*2.54/100
*
* diameter of exit and entrance pipes into HX, (part of HX)
*
Dkentr=.75*2.54/100
Dkexit=.5*2.54/100
*
* thickness of copper tubing
*
deltaw=0.035*2.54/100
*
* Hooper's Correlation inputs for ells which are part of NCHE
*
k11=800
k12=500
kinf1=.25
kinf2=.2

Equations 21 ------------System Parameters-----------------------------
mdotg=.03*3600
*
* vertical rise in piping loop
*
dz2=0
dz3=0.064
dz4=0
* as hthx is variable, dz5 must compensate in height in order that
* the total of all vertical rise = 0.
dz5=0.965+16*2.54/100-hthx
httank=-(dz2+dz3+dz4+dz5+hthx)
```

```
*
* length of pipes in piping loop
*
L=0.5
L2=L
L3=L
L4=L
L5=L
*
* diameter of pipes in piping loop
*
D=.75*2.54/100
DIA2=D
DIA3=D
DIA4=D
DIA5=D
*
* minor loss coefficients in piping loop
*
K=3/2
K2=K
K3=K
K4=K
K5=K

UNIT 69  TYPE 69 ----- NATURAL CONVECTION HEAT EXCHANGER -------------
PARAMETERS 48
 mode69 mdotg hthx
 L2     L3     L4    L5
 dz2    dz3    dz4   dz5
 D2     D3     D4    D5
 K2     K3     K4    K5
 kinlet kexit glycol percent
 dtube ncoils deltaw pitch dLout dLin 2
 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10
 dkentr dkexit 3 4  k11 k12 kinf1 kinf2
INPUTS 4
*T2    T8     head    control
THXWI  THXGI  68,13   0,0
25     25     14.     1
*outputs: 1)Two 2)Tgo 3)Qhx 4)eff 5)mdotw 6)DPSTAT 7)-DPSHEAR
* 8)error  9)ua  10)dphxsh

EQUATIONS 1
qhx=[69,3]/3.6

UNIT 68 TYPE 68 ----------  TANK -----------------------------------
PARAMETERS 20
*mode vol    cp     roe   U_t    ht of node
1  0.454    4.18   993     1.44    httank        1
1 1 55 2 0
1 1 55 2 0
```

```
*          bp
0.0  20.0  100.0
INPUTS 5
*T5   mdotw  Tmains mdotload   Tenv
0,0   69,5  0,0     0,0       0,0
19     72    19       0       19.0
DERIVATIVES 1
19
*output: 1)Tout 2)mdot out 3)Tload 4)mdotLoad 5)Qenv ...12)Tave

UNIT 25 TYPE 25 ---- PRINTER ----------------------------------------
PARAMETERS 5
*DTIME    t_START    t_END   LOGICAL UNIT#
1.        25.        90.         14           1
INPUTS 6
  THXGI  MWS   69,1   69,2   69,4     qhx
  Tgin  MDOTW  THXWO  THXGO  HXEFF    QHX
    C     KG/S   C      C      -      J/S

end
************************************************************************
```

2) Transient Deck (Mode 1)

```
assign WDATA.DAT    24
assign out.month 32
assign out.year 35
assign out.timestep 51
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                      *
*                            TRANSIENT                                 *
*            NATURAL CONVECTION HOT WATER HEATING SYSTEM               *
*                            Dec   1994                                *
*                                                                      *
*                 set for mode 1, shell and coil HX                    *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

limits 100 100
WIDTH 72
tol -.001 .001

equation 1
* NCHE mode: 1) shell and coil HX
*            2) concentric tube counterflow HX
mode69=1

Equations 14 --------Variable Geometric HX Parameters-----------------
dtube=.25*2.54/100
pitch=.35*2.54/100
hthx=16*2.54/100
Ncoils=2
```

```
*
* coil diameters
*
d1=1.*2.54/100
d2=.04445
d3=.0635
d4=.08255
d5=.1016
d6=.12065
d7=.1397
d8=.15875
d9=.1778
d10=.19685

equations 4 --------- PARAMETERS COMMON TO BOTH MODES-----------------
*
* type of glycol solution: 1=propylene glycol
*                          2=ethylene glycol
glycol=1
*
* percent propylene glycol solution
*
percent=40
*
* HX inlet, outlet minor loss coefficients
*
Kinlet=0.4
Kexit=1

equations 9 -----GEOMETRIC PARAMETERS OF HX INLET OUTLET TUBING-------
*
* diameters of ells which are part of NCHE
*
DLin=.5*2.54/100
DLout=.75*2.54/100
*
* diameter of exit and entrance pipes into HX, (part of HX)
*
Dkentr=.75*2.54/100
Dkexit=.5*2.54/100
*
* thickness of copper tubing
*
deltaw=0.035*2.54/100
*
* Hooper's Correlation inputs for ells which are part of NCHE
*
k11=800
k12=500
kinf1=.25
kinf2=.2
```

```
constants 5 --------------------- TIME -------------------------------
tbegin=90*24
tend=120*24
*tbegin=0
*tend = 8670
timest=1
firstday=int(tbegin/24)
pstop=91*24

SIMULATION tbegin tend timest

Equations 27 ------------System Parameters----------------------------
city=127
lat=43.14
slope=lat
roeg=0.2
cpg=3.82
mdotg=.03*3600
areac=4.5
*
* vertical rise in piping loop
*
dz2=0
dz3=0.064
dz4=0
* as hthx is variable, dz5 must compensate in height in order that
* the total of all vertical rise = 0.
dz5=0.965+16*2.54/100-hthx
httank=-(dz2+dz3+dz4+dz5+hthx)
*
* length of pipes in piping loop
*
L=0.5
L2=L
L3=L
L4=L
L5=L
*
* diameter of pipes in piping loop
*
D=.75*2.54/100
DIA2=D
DIA3=D
DIA4=D
DIA5=D
*
* minor loss coefficients in piping loop
*
K=3/2
K2=K
K3=K
K4=K
```

```
K5=K

unit 54 type 54 ---  WEATHER DATA GENERATER -127 madison  -------------
parameters 6
*si units LU city#
  1       24  city     1 1 1
*outputs:1)month,(2)dayofmonth,(3)hourofday,(4)T_db,(5)T_dewpoint
*        (6)humidity,(7)total radiation on horizontal
*        (8)Direct normal radiation,(9)Diffuse radiation,(10)Windspeed

UNIT 16 TYPE 16 ------   RADIATION PROCESSOR  ------------------------
PARAMETERS 9
*MODE  TRACK-MODE  SURF-MODE  DAY       LAT    SC  SHFT  SMOOTH  IE
  8        1           1      firstday  lat   4871  0       1    -1
INPUTS 9
* I  Idiff   TD1    TD2   RHOG  SLOPE  AZIMUTH  Inext   Idiff_next
54,7  54,9  54,19  54,20  0,0    0,0   0,0      54,27  54,29
0.0   0      0     1      roeg   slope  0        0      0

equation 1   fcol is the glycol flow rate
fcol=[2,1]*mdotg

UNIT 1 TYPE 1 ------------  SOLAR COLLECTOR  -------------------------
* this mode allows for a constant effectiveness HX attached
* as this is not appropriate in this deck:
*  1) set parameter 11 = parameter 4
*  2) set input 2 = input 3
PARAMETERS 14
*mode #series Area Cp_g  effmode mtest[kg/hr-m^2]
1      1      Areac  cpg    1        11.86
*FRTalpha  FRUL-1  FRUL-2  NO HX
0.634      14.49  0.026    -1
*Cp_g  opticalmode  bo-1  bo-2
cpg     1           0.448 -0.234
INPUTS 10
*Tin  mdotg mdotg  Tamb  I_T   I_horiz I_d  roeg inc.ang. slope
69,2  fcol   fcol  54,4  16,6  16,4    16,5  0,0  16,9    16,10
20.0   72     72   15.6  0.0   0.0     0.0   roeg  0.0     40.0
* outputs: 1)Tout  2)mdot_g  3)Qu[kJ/hr]


unit 2 type 2 -------------  CONTROLLER  ----------------------------
param 4
*   HDB  LDB  cutoff
7    0    0    100    96
input 4
*hi T  lo T   cutoff T  inputcontrolfunction
1,1    68,1     0,0       2,1
1      30        0         1

unit 44 type 44 -------convergence promoter (varies Tgi into NCHE)----
parameters 1
```

```
3
inputs 1
1,1
25

UNIT 69  TYPE 69 ----- NATURAL CONVECTION HEAT EXCHANGER -------------
PARAMETERS 48
 mode69 mdotg hthx
 L2     L3     L4    L5
 dz2    dz3    dz4   dz5
 D2     D3     D4    D5
 K2     K3     K4    K5
 kinlet kexit glycol percent
 dtube ncoils deltaw pitch dLout dLin 2
 d1 d2 d3 d4 d5 d6 d7 d8 d9 d10
 dkentr dkexit 3 4  k11 k12 kinf1 kinf2
INPUTS 4
*T2     T8   head    control
68,1   44,1  68,13      2,1
25     25    14.         1
*outputs: 1)Two 2)Tgo 3)Qhx 4)eff 5)mdotw 6)DPSTAT 7)-DPSHEAR
* 8)error  9)ua  10)dphxsh

nocheck 4
2,1 2,2 2,4  69,4

UNIT 14 TYPE 14 ------------LOAD---------------------------------------
PARAMETERS 72
0.0, 0.0       7.0, 0.0
7.0, 85.86   8.0, 85.86
8.0,  4.64   9.0,  4.64
9.0,  4.18  10.0,  4.18
10.0, 4.08  11.0,  4.08
11.0, 0.0   12.0,  0.0
12.0, 4.18  13.0,  4.18
13.0, 2.23   14.0,  2.23
14.0, 3.25  15.0,  3.25
15.0, 4.64  16.0,  4.64
16.0, 1.39  17.0,  1.39
17.0, 0.0   18.0,  0.0
18.0, 142.77 19.0, 142.77
19.0, 0.0   20.0, 0.0
20.0, 1.39  21.0, 1.39
21.0, 0.0   22.0, 0.0
22.0, 1.39  23.0, 1.39
23.0, 0.0   24.0, 0.0

unit 6 type 6 -----------AUXILLIARY HEATER----------------------------
parameters 5
*max heating rate, Tset, Cp, UAheater, efficiency of heater
1000000             60  4.19    0           1
inputs 4
```

```
*Tin  mdoti control Tamb for loss calcs
11,1  11,2     0,0      0,0
60    1        1       20
*output:(1)Tout,(2)mdotout,(3)required heating rate[kj/hr]

unit 10 type 11 --tempering valve located immed. upstream of tank-----
par 2
5  5
inputs 4
*Tinlet  minlet  Theatsource  Tset
0,0      14,1     68,3        0,0
8.0      0.0      15.0        60.0
*output: (1)(3)Toutlet,(2)mw to tank,(4)mw to heater

equations 3 ----------------TEMPERING VALVE FLOW EQNS-----------------
* mdraw is the mass flow rate of draw
* mtempr is the mass flow rate of mains diverted to heater
* mtk is the mass flow rate of mains that flows into tank
mdraw=[14,1]
mtempr=[10,4]
mtk=mdraw-mtempr

unit 11 type 11 ------T Piece, located in front of aux heater--------
par 1
1
inputs 4
*Tinlet1  minlet1  Tinlet2    minlet2
68,3     mtk     10,3        10,4
8.0      1.0      8.0        0.0
*output: (1)Toutlet to aux heater,(2)mw to aux heater
*inlet 1 is tank, inlet 2 is from tempering valve

UNIT 68 TYPE 68---------------------TANK-----------------------------
PARAMETERS 20
*mode vol    cp     roe   U_t    ht of node
1  0.454    4.18   993     1.44    httank        1
1 1 55 2 0
1 1 55 2 0
*          bp
0.0  20.0  100.0
INPUTS 7
*T5   mdotw    Tmains mdotload  Tamb
69,1  69,5       0,0      mtk      0,0   0,0  0,0
25     72         8        0        20     0.0  0.0
DeRIVATIVES 5
19 19 19 19 19
*output: 1)Tout 2)mdot out 3)Tload 4)mdotLoad 5)Qenv 6)rate energy to load
*         7)DUtank...12)tave 13)tnode2 14)tnode3 15)tnode4

equations 5  ---------------EQNS FOR INTEGRATER----------------------
month=8670/12
week=180
```

```
year=8670
*[Qtoload (from tank) + Qaux = Qrequired]
*[Q aux = heat needed to supplement solar heating (aux heater)]
Qreq=max([68,6]+[6,3],.0001)
Qaux=[6,3]

UNIT 30 TYPE 24------------------- INTEGRATOR-------------------------
PARAMETERS 1
month
INPUTS 5
*           Qtoenv  Qtoload QintoTK
Qaux Qreq   68,5     68,6    68,11
0      0     0          0       0

UNIT 24 TYPE 24------------------- INTEGRATOR-------------------------
PARAMETERS 1
8670
INPUTS 5
*            Qtoenv Qtoload Qintotk
 Qaux Qreq   68,5   68,6    68,11
   0         0      0     0   0

equations 9 -----------------EQNS FOR PRINTERS-----------------------
mws=[69,5]/(3600)
Qin=[30,5]
Qout=[30,4]+[30,3]+[68,7]
enbal=200*(Qin-Qout)/(max(0.001,Qin+Qout))
SF=([30,2]-[30,1])/max(0.001,[30,2])
Qinyr=[24,5]
Qoutyr=[24,4]+[24,3]+[68,7]
enbalyr=200*(Qinyr-Qoutyr)/(max(0.001,Qinyr+Qoutyr))
SFyr=([24,2]-[24,1])/max(0.001,[24,2])

UNIT 49 TYPE 25 ----------------------PRINTER-------------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
month        tbegin     tend       32
inputs 6
  Qin Qout 30,1  30,2 sf enbal
  Qin Qout Qaux  Qreq sf enbal%


UNIT 48 TYPE 25 ----------------------PRINTER-------------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
8670       tbegin     tend       35
inputs 6
 Qinyr Qoutyr 24,1 24,2 sfyr enbalyr
 Qinyr Qoutyr Qaux Qreq sf   enbal%
```

```
UNIT 51 TYPE 25 ----------------------PRINTER-----------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
timest    tbegin    pstop       51
inputs 4
69,3    mws  69,1 69,4
Q       mw   two  eff


end
************************************************************************
```

3) Transient Deck (Mode 2)

```
assign WDATA.DAT    24
assign out.month 32
assign out.year 35
assign out.timestep 51
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
*                                                                       *
*                            TRANSIENT                                  *
*            NATURAL CONVECTION HOT WATER HEATING SYSTEM                *
*                            Dec   1994                                 *
*                                                                       *
*                 set for mode 2, concentric counterflow HX             *
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

limits 100 100
WIDTH 72
tol -.001 .001

equation 1
* NCHE mode: 1) shell and coil HX
*            2) concentric tube counterflow HX
mode69=2


equation 4 ----------GEOMETRIC PARAMETERS FOR MODE 2 HX----------------
*
* heat exchanger length
*
Lhx=1
*
* inner diameters of inner and outer diameter tubes
*
di=0.75*2.54/100
do=1.50*2.54/100
*
* wall thickness of inner tube
*
```

```
thick=0.002

equations 4 ---------GEOMETRIC PARAMETERS COMMON TO BOTH MODES---------
*
* type of glycol solution: 1=propylene glycol
*                          2=ethylene glycol
glycol=1
*
* percent propylene glycol solution
*
percent=40
*
* HX inlet, outlet minor loss coefficients
*
Kinlet=0.4
Kexit=1

constants 5 --------------------- TIME ------------------------------
tbegin=90*24
tend=120*24
*tbegin=0
*tend = 8670
timest=1/4
firstday=int(tbegin/24)
pstop=91*24

SIMULATION tbegin tend timest

Equations 27 ------------System Parameters--------------------------
lat=43.14
slope=lat
roeg=0.2
cpg=3.82
mdotg=.03*3600
areac=4.5
*
* vertical rise in piping loop
*
dz2=0
dz3=0.064
dz4=0
* as Lhx is variable, dz5 must compensate in height in order that
* the total of all vertical rise = 0.
dz5=0.965+16*2.54/100-Lhx
httank=-(dz2+dz3+dz4+dz5+Lhx)
*
* length of pipes in piping loop
*
L=0.5
L2=L
L3=L
L4=L
```

```
L5=L
*
* diameter of pipes in piping loop
*
D=.75*2.54/100
DIA2=D
DIA3=D
DIA4=D
DIA5=D
*
* minor loss coefficients in piping loop
*
K=3/2
K2=K
K3=K
K4=K
K5=K

unit 54 type 54 ---  WEATHER DATA GENERATER -127 madison  ------------
parameters 6
*si units LU city#
  1       24  city     1 1 1
*outputs:1)month,(2)dayofmonth,(3)hourofday,(4)T_db,(5)T_dewpoint
*        (6)humidity,(7)total radiation on horizontal
*        (8)Direct normal radiation,(9)Diffuse radiation,(10)Windspeed

UNIT 16 TYPE 16 ------   RADIATION PROCESSOR  ------------------------
PARAMETERS 9
*MODE  TRACK-MODE  SURF-MODE  DAY       LAT    SC  SHFT  SMOOTH  IE
  8        1           1      firstday  lat   4871  0       1    -1
INPUTS 9
* I  Idiff   TD1    TD2   RHOG  SLOPE  AZIMUTH  Inext   Idiff_next
54,7  54,9  54,19  54,20  0,0    0,0   0,0      54,27  54,29
0.0   0      0     1      roeg   slope  0        0      0

equation 1   fcol is the glycol flow rate
fcol=[2,1]*mdotg

UNIT 1 TYPE 1 ------------  SOLAR COLLECTOR  -----------------------
* this mode allows for a constant effectiveness HX attached
* as this is not appropriate in this deck:
*  1) set parameter 11 = parameter 4
*  2) set input 2 = input 3
PARAMETERS 14
*mode #series Area Cp_g  effmode mtest[kg/hr-m^2]
1      1      Areac  cpg    1        11.86
*FRTalpha  FRUL-1  FRUL-2  NO HX
0.634      14.49  0.026    -1
*Cp_g  opticalmode  bo-1  bo-2
cpg     1           0.448 -0.234
INPUTS 10
*Tin  mdotg mdotg  Tamb  I_T   I_horiz I_d  roeg inc.ang. slope
```

```
69,2  fcol   fcol  54,4  16,6  16,4    16,5  0,0  16,9    16,10
20.0   72     72   15.6  0.0   0.0     0.0   roeg  0.0     40.0
* outputs: 1)Tout  2)mdot_g  3)Qu[kJ/hr]

unit 2 type 2 ------------  CONTROLLER  ---------------------------
param 4
*   HDB  LDB  cutoff
7    0    0    100    96
input 4
*hi T  lo T   cutoff T  inputcontrolfunction
1,1    68,1     0,0       2,1
1      30        0         1

unit 44 type 44 -------convergence promoter (varies Tgi into NCHE)---
parameters 1
3
inputs 1
1,1
25

UNIT 69  TYPE 69 --- NATURAL CONVECTION HEAT EXCHANGER --------------
PARAMETERS 26
 mode69 mdotg Lhx
 L2     L3     L4    L5
 dz2    dz3    dz4   dz5
 D2     D3     D4    D5
 K2     K3     K4    K5
kinlet kexit glycol percent di do  thick
INPUTS 4
*T2     T8   head  control
68,1   44,1  68,13    2,1
25     25    14.       1
*outputs: 1)Two 2)Tgo 3)Qhx 4)eff 5)mdotw 6)DPSTAT 7)-DPSHEAR
* 8)error 9)ua 10)dphxsh

nocheck 4
2,1 2,2 2,4  69,4

UNIT 14 TYPE 14 ------------LOAD-------------------------------------
PARAMETERS 72
0.0, 0.0       7.0, 0.0
7.0, 85.86   8.0, 85.86
8.0,  4.64   9.0,  4.64
9.0,  4.18  10.0,  4.18
10.0, 4.08  11.0,  4.08
11.0, 0.0   12.0,  0.0
12.0, 4.18  13.0,  4.18
13.0, 2.23   14.0,  2.23
14.0, 3.25  15.0,  3.25
15.0, 4.64  16.0,  4.64
16.0, 1.39  17.0,  1.39
17.0, 0.0   18.0,  0.0
```

```
18.0, 142.77 19.0, 142.77
19.0, 0.0   20.0, 0.0
20.0, 1.39  21.0, 1.39
21.0, 0.0   22.0, 0.0
22.0, 1.39  23.0, 1.39
23.0, 0.0   24.0, 0.0

unit 6 type 6 -----------AUXILLIARY HEATER----------------------------
parameters 5
*max heating rate, Tset, Cp, UAheater, efficiency of heater
1000000             60  4.19    0           1
inputs 4
*Tin  mdoti control Tamb for loss calcs
11,1  11,2     0,0      0,0
60    1        1       20
*output:(1)Tout,(2)mdotout,(3)required heating rate[kj/hr]

unit 10 type 11 ---tempering valve located immed. upstream of tank----
par 2
5  5
inputs 4
*Tinlet  minlet  Theatsource  Tset
0,0      14,1     68,3        0,0
8.0      0.0      15.0        60.0
*output: (1)(3)Toutlet,(2)mw to tank,(4)mw to heater

equations 3 ------TEMPERING VALVE FLOW EQNS---------------------------
* mdraw is the mass flow rate of draw
* mtempr is the mass flow rate of mains diverted to heater
* mtk is the mass flow rate of mains that flows into tank
mdraw=[14,1]
mtempr=[10,4]
mtk=mdraw-mtempr

unit 11 type 11 ---T Piece, located in front of aux heater------------
par 1
1
inputs 4
*Tinlet1  minlet1  Tinlet2    minlet2
68,3     mtk     10,3        10,4
8.0      1.0      8.0        0.0
*output: (1)Toutlet to aux heater,(2)mw to aux heater
*inlet 1 is tank, inlet 2 is from tempering valve

UNIT 68 TYPE 68---------------------TANK-----------------------------
PARAMETERS 20
*mode vol    cp     roe   U_t    ht of node
1  0.454    4.18   993     1.44    httank        1
1 1 55 2 0
1 1 55 2 0
*          bp
0.0  20.0  100.0
```

```
INPUTS 7
*T5   mdotw    Tmains mdotload  Tamb
69,1  69,5       0,0      mtk      0,0   0,0  0,0
25     72         8        0        20     0.0  0.0
DeRIVATIVES 5
19 19 19 19 19
*output: 1)Tout 2)mdot out 3)Tload 4)mdotLoad 5)Qenv 6)rate energy to load
*         7)DUtank...12)tave 13)tnode2 14)tnode3 15)tnode4

equations 5  ---------------EQNS FOR INTEGRATER-----------------------
month=8670/12
week=180
year=8670
*[Qtoload (from tank) + Qaux = Qrequired]
*[Q aux = heat needed to supplement solar heating (aux heater)]
Qreq=max([68,6]+[6,3],.0001)
Qaux=[6,3]

UNIT 30 TYPE 24------------------- INTEGRATOR-------------------------
PARAMETERS 1
month
INPUTS 5
*           Qtoenv  Qtoload QintoTK
Qaux Qreq   68,5     68,6    68,11
0      0     0         0       0

UNIT 24 TYPE 24------------------- INTEGRATOR-------------------------
PARAMETERS 1
8670
INPUTS 5
*            Qtoenv Qtoload Qintotk
 Qaux Qreq   68,5   68,6    68,11
   0         0      0     0   0

equations 9 -----------------EQNS FOR PRINTERS-----------------------
mws=[69,5]/(3600)
Qin=[30,5]
Qout=[30,4]+[30,3]+[68,7]
enbal=200*(Qin-Qout)/(max(0.001,Qin+Qout))
SF=([30,2]-[30,1])/max(0.001,[30,2])
Qinyr=[24,5]
Qoutyr=[24,4]+[24,3]+[68,7]
enbalyr=200*(Qinyr-Qoutyr)/(max(0.001,Qinyr+Qoutyr))
SFyr=([24,2]-[24,1])/max(0.001,[24,2])

UNIT 49 TYPE 25 ----------------------PRINTER-------------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
month        tbegin     tend       32
inputs 6
  Qin Qout 30,1  30,2 sf enbal
```

```
  Qin Qout Qaux  Qreq sf enbal%

UNIT 48 TYPE 25 ----------------------PRINTER-----------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
8670       tbegin     tend       35
inputs 6
 Qinyr Qoutyr 24,1 24,2 sfyr enbalyr
 Qinyr Qoutyr Qaux Qreq sf   enbal%

UNIT 51 TYPE 25 ----------------------PRINTER-----------------------
*PRINT ENERGY TERMS
PARAMETERS 4
*DTIME    t_START    t_END   LOGICAL UNIT#
timest    tbegin    pstop       51
inputs 4
69,3   mws  69,1 69,4
Q      mw   two  eff


end
************************************************************************
```

CODE:

```
      SUBROUTINE TYPE69(time,xin,out,t,dtdt,par,info,icntrl,*)
************************************************************
* This subroutine calls the NCHE subroutine, and uses the Regula
* Falsi solver to solve for water flow rate.
*
* This subroutine is called the detailed model.  It has two
* modes:
*            Mode 1:  shell and coil HX
*            Mode 2:  concentric tube counterflow HX
* Last revised: Jan 16, 1994
************************************************************
      implicit none

      integer iter,fix,seed
      real*8 mhi,mlo,errhi,errlo,dm,de,m,err,toler
      real*8 m11,m22,two1,two2,tgo0,tgo1,tgo2
      real*8 two3,tgo3,m33
      real*8 two4,tgo4,m44
      real*8 two5,two6,tgo5,tgo6,m55,m66
      integer*4 info(15)
      integer i,j,icount,ni,nd,np,icntrl(2)
      character*3 ycheck(4),ocheck(10)
      real*4 t,dtdt,par(60),time
```

```
      real*8 xin(4),out(10)
      real*8 toler2,factor
      real*8 eeeold,eeealt,deeold,deesum,deealt,eee
      integer control
      real*8 tgocp,twocp,mcp
      real*4 timeold
      real*8 fac
      integer ct,Nconv
      integer isum,jsum,fiter
*--------------------------------
*  first call, info array stuff
*--------------------------------
      if (info(7).ge.0) go to 1

       np=info(4)            ! # of parameters
       info(6)=10            ! # of outputs
       info(9)=1             ! call type 69 every timestep
       icount=0
       ni=4
       nd=0

       call typeck(1,info,ni,np,nd)
*--------------------------------
*   set variable types
*--------------------------------
      data ycheck/'TE1','TE1','PR2','DM1'/
      data ocheck/'TE1','TE1','PW1','DM1','MF1',
     @            'PR2','PR2','PR2','DM1','PR2'/
      call rcheck(info,ycheck,ocheck)

      fac=0.d0
      call NCHE(fac,fac,TIME,XIN,OUT,PAR,INFO)

      tgocp=0
      twocp=0
      mcp=0
      isum=0
      jsum=0
      toler2=0.005d0
      Nconv=0
      timeold=time

      return1
*-------------------------------------------------------------
* Program beginning
*-------------------------------------------------------------
 1    if ((time-timeold).gt.0.01) then
         fiter=0
         mcp=0
         twocp=0
         tgocp=0
         isum=0
```

```
         jsum=0
         ct=0
         fix=0

       endif
*---------------------
      ct=ct+1
      icount=icount+1
*----------------------------------------
* if controller is off, or T_g,i < T_w,i
* then set outlet temps to inlet temps, exit subroutine.
*----------------------------------------
      control=xin(4)
      if (control.lt.0.5.or.xin(1).gt.xin(2)) then
         out(2)=xin(2)                    !Tgo=Tgi
         out(1)=xin(1)                    !Two=Twi
         out(3)=0.d0                      !Q_hx=0
         out(4)=0.d0                      !eff=0
         out(5)=0.d0                      !mdotw=0
         out(6)=0.d0                      !dpstto=0
         out(7)=0.d0                      !dpshto=0
         out(8)=out(6)-out(7)             !error=0
         out(9)=0.d0                      !UA=0
         out(10)=0.d0                     !dphx=0

         m=out(5)
         err=out(8)
         timeold=time
         iter=1
         return1
      endif

      iter=1
      toler=0.001d0
*----------
* this finds a good starting mhi point, cuts down on iterations
*----------
 3    mhi=0.16d0*3600.d0

      j=1
      errhi=1
      do while (errhi.gt.0.d0)
         mhi=0.5d0*mhi
         call NCHE(mhi,errhi,TIME,XIN,OUT,PAR,INFO)

         j=j+1
         if (j.gt.9) then
            mlo=0.d0
            goto 4
         endif
       enddo
       mlo=mhi               !as the last mhi was negative
```

```
 4     mhi=2.d0*mhi
*-------------------------------------
 5    call NCHE(mhi,errhi,TIME,XIN,OUT,PAR,INFO)
      call NCHE(mlo,errlo,TIME,XIN,OUT,PAR,INFO)

 10   dm=mhi-mlo
      de=errhi-errlo
*--------------------
* If, as happens once every 600,000 timesteps or so
* (I've seen it a few times), the process is
* forced out of ncheloop again and again, and the dm is allowed
* to converge to zero without a solution, then the program will
* die.  To avoid this, the following statement kicks it back out
* to TRNSYS to change the inputs, so hopefully this won't
* reoccur.
*--------------------
      if (abs(dm).lt.0.000001d0.and.abs(de).gt.0.005d0) then
         goto 2000
      endif
*--------------------
* I've also seen it iterate within TYPE69 for 12000 iterations
* before being sent out.  Solution: send it out at 500 iterations,
* let TRNSYS return with better inputs. Also rare ocurrence.
*--------------------
      if (iter.gt.500) then
         goto 2000
      endif
*-----------------------
* this is the line function used to find new guess point
*-----------------------
      m=dm/de*(mlo*(de/dm)-errlo)

      call NCHE(m,err,TIME,XIN,OUT,PAR,INFO)
*---------------------
* The following says if m<0, call NCHE with m=0.
* If err>0, the solution for mdotw is negative,
*  => so send the solution out for mdotw=0.
* If err<0, then continue on as before.
*--------------------
 48   if (m.lt.0.d0) then
             m=0.0000d0
             call NCHE(m,err,TIME,XIN,OUT,PAR,INFO)
             if (err.gt.0.d0) then
                goto 2000
             elseif (err.le.0.d0) then
                 goto 1000
             endif
      endif
*--------
* This is for if NCHE keeps returning the same error value
* and keeps getting sent the same m value.
```

```
* The following code will sent it out of this type for TRNSYS to
* give it new values to work with.
*-----------------------
      eeealt=eeeold             !eee,eeeold,eeealt are error values
      eeeold=eee                !eee at current timestep,
      eee=err                   !eeeold at previous
      deeold=abs(eeeold-err)    !eeealt at timestep before eeeold
      deealt=abs(eeealt-eeeold)
      deesum=deeold+deealt
      if (deesum.lt.0.0000001d0) then
         goto 2000
      endif
*---------------------------------
*  The meat of the regula falsi, if err<>=toler then do this or that.
*---------------------------------
 1000 if (abs(err).lt.toler) then
         goto 2000
      endif
      if (err.lt.0.d0) then
         mlo=m
         errlo=err
      elseif (err.gt.0.d0) then
         mhi=m
         errhi=err
      endif
      iter=iter+1
      goto 10
*---------------------------
* convergence promotion: If there is excessive cycling between
* TRNSYS and this TYPE, if TYPE 44 won't resolve the cycling,
* this internal convergence promoter will.  Most likely this is
* not needed, but to insure convergence at every step, I left it
* in.  It's function is to guess a good starting point for
* the successive substitution iterating process, as at times,
* successive substution cannot converge.  It also helps the
* secant method.
*---------------------------
 2000 if (ct.gt.3) then
         Two6=Two5
         Two5=Two4
         Two4=Two3
         Two3=Two2
         Two2=Two1
         Two1=out(1)
         Tgo6=Tgo5
         Tgo5=Tgo4
         Tgo4=Tgo3
         Tgo3=Tgo2
         Tgo2=Tgo1
         Tgo1=out(2)
         m66=m55
         m55=m44
```

```
         m44=m33
         m33=m22
         m22=m11
         m11=out(5)
*----------
* The following are different cases of cycling--that is, when the
* outputs cycle between two sets of outputs, as do the inputs.
*---------
         if (abs(Two3-Two1) .lt.toler2.
     @   and.abs(Tgo3-Tgo1) .lt.toler2.
     @   and.abs(m33-m11)   .lt.toler2.

                           !ct is #times in 69 at timestep
                           !fiter is ct at last convergence promotion

     @   or.abs(twocp-two2) .lt.0.1d0.
     @   and.abs(tgocp-tgo2).lt.0.1d0.
     @   and.abs(mcp-m22)   .lt.0.1d0.
     @   and.(ct-fiter).gt.5.

     @   or.abs(m44-m22).lt.0.0000001d0.
     @   and.abs(m33-m11).lt.20.d0.
     @   and.abs(m33-m11).gt.0.0000001d0.
     @   and.abs(two3-two1).lt.1.d0.
     @   and.abs(tgo3-tgo1).lt.1.d0.
     @   and.(ct-fiter).gt.5.

     @   or.abs(m55-m33-m11).lt.0.0000001d0.
     @   and.abs(m66-m22).lt.20.d0.
     @   and.abs(m66-m22).gt.0.0000001d0.
     @   and.abs(two6-two2).lt.1.d0.
     @   and.abs(tgo6-tgo2).lt.1.d0
     @   .and.(ct-fiter).gt.5
     @   ) then

            Twocp=Two3
            Tgocp=Tgo3
            mcp=m33
            fix=fix+1
            Nconv=Nconv+1
            if (fix.eq.1) then
               tgo0=(tgo2+tgo3)/2.d0
                fiter=ct
            elseif (fix.eq.2) then
               tgo0=(tgo2+tgo3)/2.d0+abs(tgo2-tgo3)/4.d0
                fiter=ct
            elseif (fix.eq.3) then
               tgo0=(tgo2+tgo3)/2.d0-abs(tgo2-tgo3)/4.d0
                fiter=ct
            elseif (fix.ge.4.and.fix.lt.13) then
               if (fix.eq.5) fix=6
               tgo0=min(tgo2,tgo3)+(fix-3)*abs(tgo2-tgo3)/10.d0
```

```
                fiter=ct
              elseif (fix.ge.13) then
                    if (fix.eq.13) seed=13
                    seed=2045*seed+1
                    seed=seed-(seed/1048576)*1048576
                    factor=dble(seed+1)/1048577.d0
                 tgo0=min(tgo2,tgo3)+
     @                  factor*abs(tgo2-tgo3)
                 fiter=ct
              endif

              out(2)=tgo0
           endif
        endif

 2001 isum=isum+iter
      jsum=jsum+j
      timeold=time
*-----------------------------
      return1
      end
*****************************************************************
      SUBROUTINE NCHE(m,error,TIME,XIN,OUT,PAR,INFO)
*---------------------------------------------------------------
* This subroutine models the NCHE loop.
*
*  LOCATIONS:
*     (1)=HX water average
*     (2)=h2o out of storage tank
*     (3)=cold h2o into NCHE
*     (4)=hot water out of NCHE
*     (5)=hot water into storage tank
*     (6)=storage tank average
*     (7)=hot solution out of collector
*     (8)=hot solution into NCHE
*     (9)=cool solution out of NCHE
*     (10)=cool solution into pump
*     (11)=cool solution out of pump
*     (12)=cool solution into collector
*     (13)=
*     (14)=HX tube wall surface
*     (15)=HX glycol average
*-------------------------------------------------------------
      implicit none

* TRNSYS VARIABLES

      integer*4 info(15)
      integer i
      real*4 par(60),time
      real*8 xin(4),out(10)
```

```
* TYPE 69 VARIABLES

      integer hxnx                ! # of nodes for HX temp. distribution
      real*8 percent              ! % of propylene glycol solution
      real*4 mdotg1
      real*8 temp(16),roe(16)     ! property arrays  [C],[kg/m^3]
      real*8 cp(16),mu(16)        ! property arrays  [J/kg-K],[Pa-s]
      real*8 hthx                 ! height (length) of HX  [m]
      real*8 dia(6),L(6)          ! diameter, length of pipe sections  [m]
      real*8 dz(6)                ! change in elev. of pipe sections  [m]
      real*8 k(6)              ! minor loss k values assoc.w. pipe sections
      real*8 cpgave               ! ave cp for glycol in HX  [J/kg-K]
      real*8 cw,cg                ! cw = mdotw * cpw, cg = ...  [W/K]
      real*8 mdotw,mdotg          ! flow rates of water, glycol  [kg/hr]
      real*8 mdotws,mdotgs        ! flow rates of water, glycol  [kg/s]
      real*8 dphx                 ! shear pressure head in HX  [kPa]
      real*8 dptank               ! static pressure head in tank  [kPa]
      real*8 mpos,cwpos           ! abs. values of mdotw, cw
      real*8 effmod               ! modified effectiveness
      real*8 qhx                  ! heat transfer in HX  [W]
      real*8 dpstto            ! total static head around H20 loop  [kPa]
      real*8 dpshto            ! total shear pressure loss around loop[kPa]

*  VARIABLES NECESSARY TO GENERATE PRESSURE DROP AND MODIFIED EFFECTIVENESS
*  WITHOUT EXPERIMENTAL CURVES

      real*8 pitch                ! pitch of coils in HX
      real*8 Dtubes,Dtubei        ! outside,inside dia of tubes in HX
      integer Ncoils              ! # of coils in HX
      real*8 Kentr                ! minor loss K value for entrance in HX
      real*8 Kexit                ! minor loss K value for exit it HX
      integer Nbends              ! # of elbows attached to HX:
                                  ! not considered part of the piping loop
      real*8 Dcoil(15)            ! Dia of coils in HX
      real*8 Dbend(5)             ! Dia of elbows attached to HX
      integer Posben(5)           ! position of elbow:
                                  !   3 signifies water inlet
                                  !   4 signifies water outlet
      real*8 K1(5)                ! K1 value of bend taken from Hooper's
chart
      real*8 Kinf(5)              ! Kinf value of bend, Hooper's chart

      real*8 ST                   ! ave distance betw tubes across flow
      real*8 SL                   ! ave distance betw tubes in flow direx
      real*8 NL                   ! # tubes in flow direx
      real*8 N                    ! total # of tubes
      real*8 z                    ! ave length of tubes
      real*8 As,Asi               ! outer,inner surf area of tubes
      real*8 A1                   ! frontal area of crossflow tube bundle
      real*8 AT                   ! transverse flow area
      real*8 P                    ! wetted perimeter in HX
      real*8 Amin                 ! minimum flow area
```

```
      real*8 sigma                ! contraction ratio
      real*8 Dh                   ! hydraulic diameter of HX
      real*8 yawave               ! average yaw angle of tubes
      real*8 Dcave                ! average coil diameter

      character*3 fluid           ! indicator for props subroutine
      real*8 tem,dens,visc,cond,pr,cpp     ! temporary variables
      real*8 Prw                 ! ave HX water Prantl #
      real*8 Kg,Kw                ! glycol and water conductivities
      real*8 hi                   ! glycol heat transfer coefficient
      real*8 ReHX                 ! max Reynold's # in HX
      real*8 deltaw               ! HX tube wall thickness
      integer glycol              ! propylene glycol = 1
                                  ! ethylene glycol = 2
      real*8 m                    ! m is input into NCHE, mdotw
      real*8 error                ! error is output from NCHE,=(DPst-DPsh)
      real*8 errold               ! last iterations value of error
      real*8 erralt               ! last iterations value of errold
      real*8 derr                 ! derr=abs(error-errold)
                                        +abs(error-erralt)
      integer mode                ! tells type of nche: 1) shell and coil
                                  !                     2) concentric tube
      real*8 ho                   ! water-side convection coef. for nche
      real*8 ff                   ! Jakob's friction factor for nche
      real*8 velmax               ! max velocity in nche
      integer zzzz,zzz            ! counters
      real*8 Di,Do,Lhx,thick      ! geometric parameters for mode 2 nche
                                  ! Di, Do=IDs of inner and outer tubes
                                ! Lhx,thick=HX length,thickness of inner tube
      real*8 ua                   ! ua value for nche
      real*8 pi
      real*8 prg,mugly            ! average glycol props for nche
      real*8 eff1st,cw1st         ! effectiveness corresponding to a low
                                  ! cw, this effectiveness and cw value are
                                ! used to interpolate near zero effectiveness
*---------------------------------
*  first call
*---------------------------------
      if (info(7).ge.0) go to 1
*---------------------------------
* get parameters and inputs from arrays
*---------------------------------
      hxnx=10

      mode=par(1)
      mdotg1=par(2)
      mdotg=dble(int(10000.d0*mdotg1))/10000.d0

      hthx=par(3)
      L(2)=par(4)
      L(3)=par(5)
      L(4)=par(6)
```

```
      L(5)=par(7)

      dz(1)=par(3)
      dz(2)=par(8)
      dz(3)=par(9)
      dz(4)=par(10)
      dz(5)=par(11)
      dz(6)=dz(1)+dz(2)+dz(3)+dz(4)+dz(5)

      dia(2)=par(12)
      dia(3)=par(13)
      dia(4)=par(14)
      dia(5)=par(15)

      k(2)=par(16)
      k(3)=par(17)
      k(4)=par(18)
      k(5)=par(19)

      kentr=par(20)
      kexit=par(21)
      glycol=par(22)
      percent=par(23)

      if (mode.eq.1) then
      Dtubes=par(24)
      Ncoils=par(25)
      deltaw=par(26)
      pitch=par(27)
      Dbend(Nbends+2)=par(28)
      Dbend(Nbends+1)=par(29)
      Nbends=par(30)

      do i=1,10
         Dcoil(i)=par(30+i)
      enddo

      do i=1,Nbends
         Dbend(i)=par(30+10 +i)
         Posben(i)=par(30+10+Nbends+i)
         K1(i)=par(30+10+2*Nbends+i)
         Kinf(i)=par(30+10+3*Nbends+i)
      enddo

      elseif (mode.eq.2) then
         Di=par(24)
         Do=par(25)
         thick=par(26)
         Lhx=hthx
      endif
*---------------------------------------
* determine type of glycol solution to be used
```

```
*-------------------------------------
        if (glycol.eq.1) then
           fluid='pro'
        else if (glycol.eq.2) then
           fluid='eth'
        else
           if (mode.eq.2) print*,'error in parameter 26'
           if (mode.eq.1) print*,'error in parameter 22'
           print*,'must be 1 or 2'
           stop
        endif
*-------------------------------------
* analyze geometry of HX
*-----------------------------------
        pi=3.14159d0

        if (mode.eq.1) then
           call geometry(pitch,dtubes,dcoil,ncoils,st,sl,hthx,
     @             nl,z,n,as,a1,at,p,amin,sigma,dh,
     @             yawave,dcave,deltaw,asi,Dtubei)
        elseif (mode.eq.2) then
           Asi=pi*Di*Lhx
           As=pi*(Di+2.d0*thick)*Lhx
        endif
*---------------------------------------
*  guess initial output, ave temps at first iteration
*-----------------
         temp(4)=1.d0/2.d0*(temp(7)-temp(2))+temp(2)
         temp(9)=temp(7)-1.d0/2.d0*(temp(7)-temp(2))

           temp(3)=temp(2)
           temp(8)=temp(7)

         temp(1)=1.d0/2.d0*(temp(4)+temp(2))
           temp(15)=(temp(8)+temp(9))/2.d0
           temp(14)=(temp(15)+temp(1))/2.d0

      return
*=====================================================================
* Program Beginning
*=====================================================================
 1    zzz=0
      zzzz=0

 2    temp(2)=xin(1)
      temp(7)=xin(2)
      dptank=xin(3)
      mdotw=m
*---------------------------------
* Once I noticed a glycol inlet temperature > 1000,
* therefore, if this happens, to prevent TRNSYS from
* crashing, I put in the following.
```

```
*--------------------------------
      if (temp(7).gt.400) then
         temp(7)=100
      endif
*--------------------------------
*  change mdot from [kg/hr] to [kg/s]
*--------------------------------
       mdotgs=mdotg/3600.d0
       mdotws=mdotw/3600.d0
*--------------------------------
*  Assuming no heat losses in pipes:
*--------------------------------
       temp(3)=temp(2)
       temp(8)=temp(7)
*--------------------------------
*  get properties
*--------------------------------
* for water loop:
*----------------
 5     call propsw(roe,mu,cp,temp)
*----------------
*  ave glycol in HX #15:
*----------------
       tem=temp(15)
       call props(fluid,tem,percent,cpp,dens,visc,cond,pr)
       Kg=cond
       cp(15)=cpp
       cpgave=cpp
       mugly=visc
       prg=pr
*----------------
*  for ave water in HX #1:
*----------------
       tem=temp(1)
       call props(fluid,tem,percent,cpp,dens,visc,cond,pr)
       mu(1)=visc
       cp(1)=cpp
       roe(1)=dens
       prw=abs(pr)
       Kw=cond
*-------------------------------------
*  find c_w, c_g
*-------------------------------------
       cw = abs(mdotws) * cp(1)
       cg = mdotgs * cpgave
*---------------------------------
* find pressure drop in HX
*---------------------------------
       mpos=abs(mdotws)

       if (mpos.lt.0.0000001d0) then
          DPHX=0.d0
```

```
         goto 20
       endif

       if (mode.eq.1) then
          call Jakob(Nbends,Kentr,Kexit,Dbend,posben,
     @           K1,Kinf,mu,roe,mpos,ST,SL,Dtubes,Amin,
     @           NL,DPHX,ReHX,ff,velmax)

       elseif (mode.eq.2) then
          call DP2(kentr,kexit,mu,dia,di,do,roe,
     @            mdotws,Lhx,dphx)

       endif
*---------------------------------
* find modified effectiveness
*-----------------------------------------------
 20    cwpos=abs(cw)

       call HT(As,Asi,Dtubes,Dtubei,Prw,
     @         Kg,Kw,Cg,Cwpos,ReHX,effmod,hi,
     @         Di,Do,mode,ua,ho,prg,mdotgs,
     @         mugly,ncoils,dcave,eff1st,cw1st)
*-------------------------------------
*  find Temperature of water out of NCHE from eff_mod
*  energy balances
*-------------------------------------
      if (cw.gt.0.4d0) then
         temp(4)=temp(3)+
     @           effmod*cg*(temp(8)-temp(3))/cw
      else
         temp(4)=temp(3)+
     @           eff1st*cg*(temp(8)-temp(3))/cw1st
      endif
*-------
* added to ensure smooth curve for temp(4), to eliminate possible
* fluctuations due to interpolation of effectiveness mdotw relationship
* at very low flowrates.
*--------
      if (mdotws.lt.0.000003) then
         temp(4)=temp(8)
      endif
*-------------------------------------
*  find heat transfer in NCHE from t(4),mdot_w
*-------------------------------------
      qhx=cw*(temp(4)-temp(3))
*-------------------------------------
*  find temperature of antifreeze out ot NCHE from q_hx
*-------------------------------------
       temp(9)=temp(8)-qhx/cg
       temp(5)=temp(4)
       temp(15)=(temp(8)+temp(9))/2.d0
*-------------------------------------
```

```
*  find properties of water loop again
*-------------------------------------
       call propsw(roe,mu,cp,temp)
*-------------------------------------
*  find hx average water temperature, ave hx density
*-------------------------------------
       call xtdist(temp,hthx,hxnx,cwpos,cg,roe)
*-------------------------------------
* find average tube surface temperature in HX
*-------------------------------------
       Temp(14)=Temp(1)+Qhx/(hi*Asi)
*-------------------------------------
*  find static pressure drop from densities around water loop
*-------------------------------------
 30    call statdp(dptank,roe,dz,dpstto)
*-------------------------------------
* find dpshto from old mdot value
*-------------------------------------
       call findshear(k,L,dia,roe,mu,dpshto,mdotws,dphx)
*-------------------------------------
*  change mdot from [kg/s] to [kg/hr]
*-------------------------------------
 900  mdotg=mdotgs*3600.d0
      mdotw=mdotws*3600.d0
*-------------------------------------
*  write output variables to arrays
*-------------------------------------
        out(1)=temp(5)             ![C]
        out(2)=temp(9)             ![C]
        out(3)=qhx*3.6d0           ![kJ/hr]
        out(4)=effmod              ![-]
        out(5)=mdotw               ![kg/hr]
        out(6)=-dpstto             ![kPa]
        out(7)=dpshto              ![kPa]
        out(8)=out(7)-out(6)       ![kPa]
        out(9)=ua                  ![W/k]
        out(10)=dphx               ![kPa]
*-------------------------------------
* Counters, old m,error,time values
*-------------------------------------
        erralt=errold
        errold=error
        error=out(8)
        derr=abs(error-errold)+abs(error-erralt)
*------------------------------
        if (derr.le.1.0d-6) then
           return
*------------------------------
* if it has not yet converged then,
* increment number of iterations in NCHE
* for this call from Regula Falsi
*------------------------------
```

```
        elseif (derr.gt.1.0d-6) then
           zzz=zzz+1
*-----------------------------
* If it hasn't converged in 10 iterations, it is cycling within NCHE.
* After 10 iterations in NCHE, shift Tgi over by 1%, try again.
*-----------------------------
           if (zzz.gt.10) then               !if 10 iter in nche
               xin(2)=1.01d0*xin(2)
               zzz=0
               zzzz=zzzz+1                   !keep track of # shifts
           endif
*--------------------------
* If it still won't converge after 5 shifts, send it out to Regula Falsi
* solver as it is.
*--------------------------
           if (zzzz.gt.5) then
                 return
           endif

           goto 2                            !shift or no shift, try again
        endif
*-------------------------
       return
       end
*=======================================================
      Subroutine DP2(kentr,kexit,mu,D,di,do,roe,
     @               mdotw,Lhx,dphx)
*-------------------------------------------------------
*  This subroutine will find the shear pressure drop in a
*  concentric tube heat exchanger using correlations.
*
*     The Fanning friction factor comes from Kakic, "Handbook of
*     Single Phase Heat Transfer", pp. 3.91-3.
*     The DPtube eqn from Bejan, "Heat Transfer",  p. 479.
*-------------------------------------------------------
      implicit none

      real*8 di,do,af,roe(16),mu(16),mdotw,Lhx,dphx
      real*8 kentr,kexit,pi,dh,rstar,rmstar,veloc,dppipe
      real*8 D(6),f,dpentr,dpexit,red,lentr,lexit

      pi=3.14159d0

         dh=do-di
         af=0.25d0*pi*(do**2-di**2)
         rstar=di/do
         rmstar=sqrt((1-rstar)/(2.d0*dlog(1/rstar)))
         veloc=mdotw/(roe(1)*af)
         red=roe(1)*veloc*dh/mu(1)
         f=(16.d0*(1.d0-rstar)**2)/(red*(1+rstar**2-2.d0*rmstar**2))
         dppipe=(f*4.d0*Lhx*roe(1)*veloc**2)/(2.d0*dh)
         Lentr=kentr*mdotw/(16.d0*pi*mu(4))
```

```
         Lexit=kexit*mdotw/(16.d0*pi*mu(3))
         dpentr=128.d0*mu(4)*mdotw*lentr/(pi*d(4)**4*roe(4))
         dpexit=128.d0*mu(3)*mdotw*lexit/(pi*d(3)**4*roe(3))
         dphx=(dppipe+dpentr+dpexit)*1/1000
         return

      end
*=========================================================
       Subroutine findshear(k,L,d,roe,mu,dpshto,mdotws,dphx)
*---------------------------------------------------------
*  This subroutine will find the shear pressure drop in the
*  water loop  using the given water flow rate.
*---------------------------------------------------------
      implicit none
      integer i,j
      real*8 k(6),L(6),d(6),roe(16),mu(16)
      real*8 dpshto,mdotws,pi
      real*8 var1,var2,var3,var4,dphx
*-----------------
      pi=3.14159d0
*-----------------
      var1=(16.d0*pi*mu(2)*L(2) + K(2)*abs(mdotws))/(D(2)**4*roe(2))
      var2=(16.d0*pi*mu(3)*L(3) + K(3)*abs(mdotws))/(D(3)**4*roe(3))
      var3=(16.d0*pi*mu(4)*L(4) + K(4)*abs(mdotws))/(D(4)**4*roe(4))
      var4=(16.d0*pi*mu(5)*L(5) + K(5)*abs(mdotws))/(D(5)**4*roe(5))

      dpshto= 1.d0/1000.d0*
     @          ( 8.d0*abs(mdotws)/(pi*pi)
     @                            *(var1+var2+var3+var4))
     @           +dphx
*-------------------
*  if flow is negative, then set dpshto to opposite sign.
*-------------------
      if (mdotws.lt.0.d0) dpshto=-dpshto
      return
      end
*=========================================================
      Subroutine statdp(dptank,roe,dz,dpstto)
*---------------------------------------------------------
*  This subroutine will use densities around the water
*  loop to find static pressure drop.
*---------------------------------------------------------
      integer i
      real*8 dptank,roe(16),g
      real*8 dz(6),dpstto,dpstat(16)

      g=9.806d0
      dpstat(6)=dptank

      dpstto=0.d0
      do i=1,5
       dpstat(i)=roe(i)*g*dz(i)*.001
```

```
       dpstto=dpstto+dpstat(i)
      enddo

      dpstto=dpstto-dpstat(6)
      return
      end
*=============================================================
      Subroutine xtdist(temp,hthx,hxnx,cw,cgly,roe)
*-------------------------------------------------------------
*  This subroutine will use the trapezoid rule to find the
*  temperature distribution of the heat exchanger.
*  This subroutine assumes a perfectly counterflow temperature
*  distribution.
*-------------------------------------------------------------
*  SYMBOLS:
*     hthx=height of hx
*     hxnx=number of points used in integration
*     ax,bx,cx,dx,ex=variables of no great significance
*     dtw,dthot,dto=interim variables of little significance
*-------------------------------------------------------------
      implicit none

      integer i,hxnx
      real*8 temp(16),var1,dtw,dthot,dto,cw,cgly
      real*8 ax,bx,cx,dx,ex,thx(20),tinteg
      real*8 thxave,hthx,avdens,rhohx(20),roe(16)
*----------------------------
      if (abs(temp(3)-temp(4)).lt.0.0010d0) then
        temp(4)=temp(3)+0.0001d0
      endif
*--------------------------
      var1=hthx/(2*hxnx)
      dtw=temp(4)-temp(3)
      dthot=temp(8)-temp(4)
*---------------
* negative dthot kills program
*---------------
      if (temp(4).gt.temp(8)) dthot=0.d0
*---------------
      dto=temp(8)-dtw*cw/cgly-temp(3)
*------------------------------
      ax=temp(3)
      dx=2*var1
*------------------------------
      if (dthot.eq.dto) then
       bx=0.d0
       cx=0.d0
       ex=dtw/hthx
      else
       bx=dto*dtw/(dthot-dto)
         cx=dthot/dto
       ex=0.d0
```

```
      endif

       if (cx.lt.0.d0) cx=0.0001d0
*----------------------------
*  find temperature at each point in the NCHE
*----------------------------
      thx(0)=ax + bx*(cx**(0*dx/hthx)-1)

      do i=1,hxnx
       thx(i)=ax + bx*(cx**(i*dx/hthx)-1)
     @              +ex*i*dx

      end do
*----------------------------
*  use Trapezoid Rule to integrate
*----------------------------
      tinteg=0.d0
      do i=1,hxnx-1
       tinteg=tinteg+2.d0*thx(i)
      end do
      tinteg=var1*(tinteg+thx(0)+thx(hxnx))
*----------------------------
*  find the average temperature of hx, assign it to temp(1)
*----------------------------
      thxave=tinteg/hthx
      temp(1)=thxave
*----------------------------
*  find the density at each point corresponding to the temp
*----------------------------
             rhohx(0)=(999.8396 + 18.224944*thx(0)
     @              - 0.00792221*thx(0)**2
     @              -55.44846e-6*thx(0)**3
     @              +149.7562e-9*thx(0)**4
     @              -393.2952e-12*thx(0)**5)
     @              /(1.d0+18.159725e-3*thx(0))
      do i=1,hxnx
       rhohx(i)=(999.8396 + 18.224944*thx(i)
     @              - 0.00792221*thx(i)**2
     @              -55.44846e-6*thx(i)**3
     @              +149.7562e-9*thx(i)**4
     @              -393.2952e-12*thx(i)**5)
     @              /(1.d0+18.159725e-3*thx(i))

      enddo
*----------------------------
*  use Trapezoid Rule to integrate
*----------------------------
      tinteg=0.d0
      do i=1,hxnx-1
       tinteg=tinteg+2.d0*rhohx(i)
      end do
      tinteg=var1*(tinteg+rhohx(0)+rhohx(hxnx))
```

```
*-----------------------------
*  find the average temperature of hx, assign it to temp(1)
*-----------------------------
      avdens=tinteg/hthx
      roe(1)=avdens
      return

      end
*=========================================================
      Subroutine propsw(roe,mu,cp,temp)
*---------------------------------------------------------
*  This subrountine will find:
*     1) the density of water [kg/m^3]
*     2) the viscosity of water [kg/m-s]
*     3) the specific heat of water [J/kg-C]
*  at the different locations specified using equations given
*  in Gebhart [1988].
*---------------------------------------------------------
      integer  i,j
      real*8  roe(16),mu(16),cp(16),temp(16)
*-------------------------
      do i=1,6

       roe(i)=(999.8396 + 18.224944*temp(i)
     @              - 0.00792221*temp(i)**2
     @              -55.44846e-6*temp(i)**3
     @              +149.7562e-9*temp(i)**4
     @              -393.2952e-12*temp(i)**5)
     @              /(1.d0+18.159725e-3*temp(i))

       mu(i)=2.414e-5*10**(247.8/(temp(i)+133.15))

       cp(i) = 4193 - 0.79411* temp(i) + 0.012545*temp(i)**2
     @            -3.9665e-5*temp(i)**3 + 1.7441e-7*temp(i)**4

      end do

      return
      end
*-----------------------------------------------------------------------
       Subroutine Props(fluid,Tem,percent,cp,dens,visc,cond,prantl)

C   This routine calculates the properties of propylene glycol given
C   the temperature in Kelvin and the percent by volume of propylene
C   glycol.
C   This routine also calculates the properties of ethylene glycol
C   given the temperature in Kelvin and the percent by volume of
C   ethylene glycol.  The ethylene glycol curves only good between
C   55 and 85 %.
C   This routine will also calculate the properties of water
C   given the temperature in Celsius, from equations given in
C   Gebhart [1988].  The conductivity of water was found by
```

```
C   putting tabulated data in Incropera into equation form.

C   Be aware all inputs to this subroutine must be in Celsius.

C   The properties are:    1) specific heat (cp) in J/kg-K
C                          2) density (dens) in kg/m^3
C                          3) thermal conductivity (cond) in W/m-K
C                          4) viscosity (visc) in Pa-s.
C   This routine will also calculate the Prandtl number.
C-----------------------------------------------------------------------
      implicit none
      real*8 Tem,percent,cp,dens,cond,visc,prantl
      real*8 a,b,c,d
      character*3 fluid

      if (fluid.eq.'pro') then
      tem=tem+273.15d0

      cp      = ((3.8649883866 - 0.023691954902*percent -
     *      0.00011278222908*percent**2)+(0.001023655712+
     *        5.6633876714e-5*percent)*Tem)*1000

      dens     = (875.54696219 + 2.151387542*percent) +
     *        (1.1191046068 - 0.0007599907262*percent -
     *        4.9236799989e-5*percent**2)*Tem + (-0.002377960199 -
     *        9.1377252136e-6*percent + 1.0872237562e-7*percent**2)
     *        * Tem**2

      cond    = (-0.78595253278 + 0.015561899561*percent -
     *        4.8933521576e-5*percent**2) + (0.0076866167254 -
     *        0.0001155974176*percent + 3.660336083e-7*percent**2)*Tem
     *        + (-9.9976810237e-6 + 1.4560615474e-7*percent -
     *        4.5879383578e-10*percent**2)*Tem**2

      visc    = exp((71.639163222 - 0.66981698459*Tem +
     *        0.0019150513174*Tem**2 - 1.8587687783e-6*Tem**3) +
     *        (0.27019804611 - 0.0012299975866*Tem +
     *        1.5045427918e-6*Tem**2)*percent)
*-----------------------------------------------------------
      elseif (fluid.eq.'wat') then

        dens        =(999.8396 + 18.224944*tem
     @              - 0.00792221*tem**2
     @              -55.44846e-6*tem**3
     @              +149.7562e-9*tem**4
     @              -393.2952e-12*tem**5)
     @              /(1.d0+18.159725e-3*tem)

      visc       =2.414e-5*10**(247.8/(tem+133.15))

      cp     = 4193 - 0.79411* tem + 0.012545*tem**2
     @            -3.9665e-5*tem**3 + 1.7441e-7*tem**4
```

```
*        cond   = 0.56974 + 1.8e-3*tem -6.236e-6*tem**2
*     @           -7.3505e-9*tem**3

         cond    = 0.56619d0 + 1.5915d-3*tem

*-----------------------------------------------
         elseif (fluid.eq.'eth') then
            if (percent.gt.85.d0.or.percent.lt.55.d0) then
               print*,'ethylene properties only calculated'
               print*,' for 55% to 85% by volume.'
               pause
               return
            endif

       Tem=tem+273.15

       cp=3.9189-.035267*percent+(.0014555+4.8423d-5*percent)*Tem

       B=-.84402+.016948*percent-6.99691d-5*percent**2.
       C=.0079877-.00012444*percent+5.00412d-7*percent**2.
       D=-1.0647d-5+1.708955d-7*percent-7.065844d-10*percent**2.
       cond=B+C*Tem+D*Tem**2.

       B=970.43146598-10.001392253*Tem+.034056662648*Tem**2.
      &     -3.8613683343d-5*Tem**3.
       C=-27.036068044+.27995557712*Tem-.00096062280174*Tem**2.
      &     +1.0941819338d-6*Tem**3.
       D=.19624504556-.0020225892738*Tem+6.9220560583d-6*Tem**2.
      &     -7.8710335530e-9*Tem**3.
       visc=exp(B+C*percent+D*percent**2.)

       B=884.53+2.1741*percent
       C=1.1613-0.0033403*percent
       D=-0.0024393+2.994d-8*percent
       dens=B+C*Tem+D*Tem**2

       endif
*--------------------------
       prantl  = cp*visc/cond
*--------------------------
       return
       end
*****************************************************************
        Subroutine Geometry(pitch,dtube,dcoil,ncoils,st,sl,hthx,
     @                      nl,z,n,as,a1,at,p,amin,sigma,dh,
     @                      yawave,dcave,deltaw,asi,dtubei)
*---------------------------------------------------------------
*  This subroutine will find analytical geometric representations
*  of the NCHE.
*  Although this is a coil we are analyzing, we treat it as inline
*  tubes in crossflow for the hydraulic and Heat Transfer analysis.
```

```
*-------------------------------------------------------------
       implicit none

       integer i,j
       integer ncoils              !number of coils
       real*8  n                   !total number of tubes in crossflow
       real*8  nl                  !tubes in flow direx
       real*8  nt                  !tubes across flow direx
       real*8 dtube,dtubei         !outside, inside tube diameter
       real*8 pitch,pi,hthx
       real*8 deltaw               !tube wall thickness
       real*8 sl                   !distance betw tubes in flow direx
       real*8 st                   !distance betw tubes across flow
       real*8 sttot,stt(15)
       real*8 l                    !length of tube bundle in flow direx
       real*8 w                    !width of tube bundle across flow
       real*8 z                    !length of tubes
       real*8 As                   !total surface area for heat transfer
       real*8 Asi                  !inside tube surface area for HT
       real*8 A1                   !frontal area
       real*8 At                   !transverse flow area
       real*8 P                    !ht surface perimeter, wetted perimeter
       real*8 Amin                 !minimum flow area
       real*8 sigma                !contraction ratio
       real*8 Dh                   !hydraulic diameter
       real*8 Dcoil(15)            !array of coil diameters
       real*8 Dcave                !average coil diameter
       real*8 Dcoilt               !Sum of coil diameters
       real*8 yawave               !ave yaw angle to be used elsewhere
       real*8 beta,volume
*------------------------
       pi=3.14159d0
       nt=dble(ncoils)
*-------------------
* find inside tube diameter
*------------------
       dtubei=dtube-deltaw*2.d0
*--------------------
* find SL
*--------------------
       SL=pitch
*-------------------
* find ST
*-------------------
       sttot=0.d0
       do i=1,Ncoils-1
          stt(i)=abs(dcoil(i+1)/2.d0-dcoil(i)/2.d0)
          sttot=sttot+stt(i)
       enddo
       ST=sttot/dble(Ncoils-1)
*---------------------
* find number of rows of tubes
```

```
*---------------------
       nl=(hthx)/pitch
*-----------------------
*  find average coil diameter
*  (assuming that the input coil diameter is the inside diameter)
*----------------------
       Dcave=0.d0
       do i=1,Ncoils
          Dcave=dcoil(i)+Dcave
       enddo
       Dcave=Dcave/dble(Ncoils)
*----------------------
*  find average tube length,z
*-----------------------
       dcoilt=0.d0
       do i=1,Ncoils
          dcoilt=dcoilt+dcoil(i)
       enddo
       z=sqrt((pi*dcave)**2+(pitch)**2)
*----------------------
* find number of tubes
*-----------------------
       n=nl*nt
*--------------------
* find length of tube bundle in flow direction
*--------------------
       L=NL*SL
*-------------------
* find width of tube bundle across flow
*---------------------
       w=nt*st
*---------------------
* find total heat transfer surface area
*---------------------
       as=pi*dtube*z*n
       asi=pi*dtubei*z*n
*---------------------
* find frontal area
*---------------------
       a1=nt*st*z
*--------------------
* find transverse area
*--------------------
       at=(st-dtube)*nt*z
*--------------------
* find perimeter
*--------------------
       p=as/l
*--------------------
* for inline tube bundles,
*--------------------
       amin=at
```

```
*--------------------
* find contraction ratio
*--------------------
       sigma=amin/a1
*------------------
* find volume
*------------------
       volume=hthx*a1
*-------------------
* find surface area density
*-------------------
       beta=As/volume
*--------------------
* find hydraulic diameter
*--------------------
       dh=4*sigma/beta
*--------------------
* find average yaw angle
*-------------------
       yawave=datan((2.d0*dcave/pitch))*360.d0/(2.d0*pi)

       return
       end
*************************************************************************
      Subroutine Jakob(Nbends,Kentr,Kexit,Dbend,posben,
     @                 K1,Kinf,mu,roe,mdotw,ST,SL,Dtube,Amin,
     @                 NL,DPHX,Remax,
     @                 f,velmax)
*-----------------------------------------------------------------
*        This subroutine will find the pressure drop
*     across a shell and coil or in-line tube cross flow heat
*     exchanger.  This subroutine uses Jakob's correlation for
*     in line tube bundles in cross flow as presented in
*     Chapman p. 353.
*        The minor losses for ells are found using Hooper's
*     correlation for laminar and turbulent
*     flow [Kakic pp 10.8,10.9].
*        The K values for the entrance and exit conditions, and
*     the K1, Kinf values are chosen by the user, and submitted
*     as parameters.
*-----------------------------------------------------------------
      implicit none

      integer i,j
      real*8 pi
*----------- minor losses----------------------
      integer Nbends         !number of ells attached to HX for minor
     @                       !loss calculation
      integer posben(5)      !position in SDHW setup where bend is
     @                       !ie. if at beginning of HX, posben=3
     @                       !    if at end of HX posben=4
      real*8 Kentr           !minor loss coef. for HX to pipe, entrance
```

```
      real*8 Kexit           !minor loss coef. for pipe to HX, exit
      real*8 Dbend(5)        !diameter of pipe at bend
      real*8 K1(5),Kinf(5)   !k1, Kinf  value chosen by user for ell
      real*8 astar(5)        !radius of fitting in inches
      real*8 K(10)           !minor loss coeff
      real*8 Le(10)          !effective length
*-------------------Pressure Loss-------------------------------

      real*8 DPm(10)         !pressure loss due to fittings
      real*8 DPmtot          !total minor losses
      real*8 DP              !pressure loss in HX, without minor losses
      real*8 DPHX            !total pressure loss in HX
*------------------Losses in HX--------------------------------
      real*8 NL              !number of tubes in flow direx (not # coils)
      real*8 Remax           !max Reynold's # at Amin in HX, uses velmax
      real*8 f               !Jakob's friction factor
      real*8 velmax          !max velocity in HX
      real*8 mu(16),roe(16),mdotw,ST,SL,Dtube,Amin
      real*8 C1,C2,m,nn      !parameters for Jakob correlation
*-------------------------------------------------------------
      pi=3.14159
*-------------------------------------------------------------
* MINOR LOSSES (for situations that are part of HX only)
*-------------------------------------------------------------
*     Bends, HOOPER'S CORRELATION
*------------------
       do i=1,Nbends
         astar(i)  =  Dbend(i)/2.d0 * 100.d0/2.54d0
         le(i)=(k1(i)*pi*Dbend(i)*mu(posben(i))/4.d0
     @        + kinf(i)*(1.d0+0.5d0*astar(i))*mdotw)
     @        *1.d0/(16.d0*pi*mu(posben(i)))
       enddo
*---------------
* exit and entrance conditions are placed after the bends
*---------------
      posben(Nbends+1)=3
      posben(Nbends+2)=4
      K(Nbends+1)  =  Kexit
      K(Nbends+2)  =  Kentr
      do i=Nbends+1,Nbends+2
         Le(i)     =  K(i)*mdotw/(16.d0*pi*mu(posben(i)))
      enddo
*--------------
* find and sum minor losses for each condition
*--------------
      DPmtot=0.d0

      do i=1,(2+Nbends)
         DPm(i)     =  128d0*mu(posben(i))*mdotw*Le(i)
     @             /  (pi*Dbend(i)**4*roe(posben(i)))
         DPmtot       =  DPmtot+DPm(i)
      enddo
```

```
*-----------------------------------------------------------------
*  PRESSURE LOSSES IN HX; MAJOR LOSSES
*  JAKOB'S CORRELATION
*-----------------------------------------------------------------
*  constants for Jakobs correlation
*--------------------
      C1           =  0.176d0
      C2           =  0.34d0*SL/Dtube
      nn           =  0.43d0+1.13d0*Dtube/ST
      m            =  0.15d0
*------------------
* find max velocity, max Re, friction factor, yaw angle correx factor
*-----------------
      velmax       =  mdotw/(Amin*roe(1))
      Remax        =  velmax*Dtube*roe(1)/mu(1)
      f            =  (C1+C2/((ST/Dtube-1)**nn))*Remax**(-m)
*----------------
* find DP for tubes, DP total
*----------------
      DP           =  f*NL*(0.5d0*roe(3)*velmax**2)
      DPHX         =  1.d0/1000.d0*(DP+DPmtot)             ! [kPa]

      Return
      end
*******************************************************************
      Subroutine HT(As,Asi,Dt,Dti,Prw,
     @              Kg,Kw,Cg,Cw,Re,modeff,hi,
     @              di,do,mode,ua,ho,prg,mdotg,mugly,
     @              ncoils,dcave,eff1st,cw1st)
*-----------------------------------------------------------------
*  This subroutine finds the modified effectiveness of a helical tube
*  heat exchanger.   This subroutine uses
*  Zukauskas Crossflow correlation for the external heat transfer and
*  Manlapanz and Churchill's correlation for internal flow through helices.
*  The modified effectiveness of the HX is calculated.
*
*  note on positions:  14--wall ave in HX
*                      15--gly ave in HX
*                       1--water ave in HX
*-----------------------------------------------------------------
      implicit none

      integer i,mode
      real*8 As,Asi           !Outer, inner surface area of tubes in HX
      real*8 Dt,Dti           !Outer, inner diameter of glycol tubes
      real*8 Prw              !Ave HX Prandtl # of water
      real*8 Kg,Kw            !Ave conductivity of glycol, water
      real*8 Cg,Cw,Cstar      !Cg=cpg*mdotg, Cw=cpw*mdotw, Cstar=cg/cw
      real*8 Re               !max Re # of water
      real*8 UA,NTU
      real*8 modeff           !modified effectiveness
      real*8 hi,ho            !inside, outside heat transfer coefficients
```

```
      real*8 NUi,NUo          !inside, outside Nusselt #'s
      real*8 di,do
      real*8 x1,x2,dean,prg,regly,mdotg,mugly,pi
      real*8 dcave,cstar2,eff1st,cw1st,modef2
      integer ncoils
*------------------
* find inside heat transfer coefficient
*-----------------
      pi=3.14159d0

      if (mode.eq.1) then
         regly=4.d0*mdotg/(ncoils*pi*mugly*dti)
         dean=regly*sqrt(dti/dcave)
         x1=(1+1352.d0/(dean**2*prg))**2
         x2=1.d0+1.15d0/prg
         nui=((4.364d0+4.636d0/x1)**3
     @        + 1.816d0*(dean/x2)**(1.5d0))**.33333
         hi=NUi*kg/Dti
      elseif (mode.eq.2) then
         nui=3.66d0
         hi=NUi*kg/Di
      endif
*-----------------
* find outside heat transfer coefficient
*----------------
       if (mode.eq.1) then
         NUo=0.9d0*Re**0.4*Prw**0.36
         ho=NUo*kw/Dt
       elseif (mode.eq.2) then
         NUo= 31.2562d0              -    401.2468d0*(di/do)
     @       + 3107.4528d0*(di/do)**2  -  13741.8820d0*(di/do)**3
     @       +36503.6833d0*(di/do)**4  -  59189.5378d0*(di/do)**5
     @       +57261.5851d0*(di/do)**6  -  30318.9480d0*(di/do)**7
     @       + 6752.4973d0*(di/do)**8
         ho=NUo*kw/(Do-Di)
       else
          print*,'mode must be 1 or 2, you have specified: ',mode
       endif
*----------------
* find UA, NTU
*----------------
       if (ho.lt.0.000001d0) then
            ua=(hi*Asi)
       else
            UA= 1 /    (   1/(hi*Asi)   +   1/(ho*As)    )
       endif

       NTU=UA/cg
*----------------
* for zero flow, set modeff=0
* for low flow, modeff eqn blows up, so take a point near the origin
* and extrapolate back to get the appropriate modeff.
```

```
* In NCHE, for low flow, cw1st and eff1st are used rather than modeff.
*----------------
 400   if (cw.lt.0.4d0) then
         Cstar2=20.d0
         modef2=   ( 1.d0 -     exp(-NTU*(1.d0-Cstar2)))
     @        / ( 1.d0 - Cstar2*exp(-NTU*(1.d0-Cstar2)))
         cw1st=cg/cstar2
         eff1st=modef2

           if (cw.le.0.0000001d0) then
              modeff=0.d0
              return
           endif

         modeff=eff1st*cw/cw1st

       else

        cstar=cg/cw
         modeff=   ( 1.d0 -    exp(-NTU*(1.d0-Cstar)))
     @        / ( 1.d0 - Cstar*exp(-NTU*(1.d0-Cstar)))

       endif
*----------------
       return
       end
*****************************************************************
```