APPENDIX E

EES Models: Simple and Detailed Models

Simple Model:

"Steady State System Model
Analysis of Shell and Coil NCHE-----SIMPLE MODEL
written by John Avina, Solar Energy Lab, University of Wisconsin-Madison, January 1995

This EES deck covers Fraser's 'steady state case' in which:
symbols: g represents glycol (collector fluid), w represents water
INPUTS: Tgi (in parametric table as T[7])
PARAMETERS: Twi, T_tank, Density_tank, mdotg
OUTPUTS: Qhx, Tgo, Two, mdotw
Physically this represents: 1) a NCHE, with glycol flow in and out
2) a tank connected to the NCHE inlet, but held at a constant temperature, 19C.
3) the piping connecting the water outflow of the NCHE and the tank.
positions:
[1] NCHE
[2] cold water out of tank
[3] cold water out of tank, into HX (connected to [2])
[4] hot water out of HX
[5] hot water out of HX, into tank (connected to [4])
[6] tank
[7] hot glycol into HX, out of collector
[8] warm glycol out of HX"

Procedure tempdist(dx,posit,height_hx,t[3],t[4],T[7],cw,cgly:rho_posit,thx_posit)

```
dtw:=t[4]-t[3]
dt_hot:=T[7]-t[4]
dto:=T[7]-(dtw*cw/cgly)-t[3]
ax:=t[3]
If (abs(t[3]-t[4])<0.0001) Then
t[4]=t[3]+0.0001
EndIf
If (t[4]>T[7]) Then
dthot=0
EndIf
If (dt_hot=dto) Then
  bx:=0
  cx:=0
  ex:=dtw/height_hx
  goto 20
Else bx:=(dto*dtw)/(dt_hot-dto)
     cx:=dt_hot/dto
     ex:=0
20:
If (cx<0) Then
cx=0.0001
EndIf
thx_posit:=ax+bx*(cx**(posit*dx/height_hx)-1)+ex*posit*dx
rho_posit=(999.8396+18.224944*thx_posit-.00792221*thx_posit^2-55.44846e-
6*thx_posit^3+149.7562e-9*thx_posit^4-393.2952e-12*thx_posit^5)/(1+18.159725e-
3*thx_posit)
end

"  DRIVING FORCES:  INPUTS  "
mdotgly=.02
T[2]=19

"GEOMETRY------"
heighttank=1.435
dz[1]=0.406;dz[2]=0.064;dz[3]=0;dz[4]=0;dz[5]=0.965;dz[6]=-heighttank
dx[2]=1.535;dx[3]=.281;dx[4]=1.384;dx[5]=.246
pipedia[2]=0.0328;pipedia[3]=.0254;pipedia[4]=.0191;pipedia[5]=.0253
k[2]=6.7;k[3]=2.4;k[5]=2.9;k[4]=0.33 "minor loss coefficients"
g=9.806
height_hx=dz[1]

T[2]=t[3]"Assume no heat loss in pipes"
T[4]=T[5]
cpgly=(3636.2+3.325*(T[7]+t[8])/2)"J/kgK"
```

```
cpw=(4209.4-1.824*T_hx_ave+3.0525e-2*T_hx_ave^2-1.2388e-4*T_hx_ave^3+1.2774e-
7*T_hx_ave^4)
Duplicate i=2,5
rho[i]=(999.8396+18.224944*t[i]-.00792221*t[i]^2-55.44846e-6*t[i]^3+149.7562e-9*t[i]^4-
393.2952e-12*t[i]^5)/(1+18.159725e-3*t[i])
mu[i]=2.414e-5*10^(247.8/(t[i]+133.15))
End
rho[1]=rhohx
rho[6]=rho[2]

"find temperature distribution in hx in order to get rho hx ave, T hx ave"
Duplicate i=0,10
Call tempdist(dx,i,height_hx,t[3],T[4],T[7],cw,cgly:rho_hx[i],t_hx[i])
End

"trapezoidal approximation of HX density, HX water temperature"
dx=2*h
n=10
h=height_hx/(2*n)
rho_integ=h*(rho_hx[0]+rho_hx[10]+2*sum(rho_hx[i],i=1,9))
rhohx=rho_integ/(height_hx)

T_integ=h*(t_hx[0]+t_hx[10]+2*sum(t_hx[i],i=1,9))
T_hx_ave=T_integ/(height_hx)

"energy balance, modified effectiveness"
cw=cpw*mdotw
cgly=cpgly*mdotgly
"Fraser's experimental modified effectiveness curve:"
effmod=-.0014009+49.756*mdotw+1783.1*mdotw^2-
2.0541e5*mdotw^3+5.5063e6*mdotw^4-4.841e7*mdotw^5
effmod=mdotw*cpw*(T[4]-t[3])/(mdotgly*cpgly*(T[7]-t[3]))
qhx=mdotgly*cpgly*(T[7]-t[8])
qhx=mdotw*cpw*(T[4]-t[3])

"Static Pressure Losses Around Water Loop"
Duplicate i=1,6
DP_st[i]=rho[i]*g*dz[i]*1e-3
End
Delta_P_static_tot=sum(DP_st[i],i=1,6)

"Fraser's experimental HX pressure loss curve"
DP_sh[1]=(47e3*mdotw^2+270*mdotw)*1e-3
DP_sh[6]=0
```

```
"Minor Losses Around Water Loop"
Duplicate i=2,5
"Le represents effective length of minor loss K value"
Le[i]=k[i]*mdotw/(16*pi*mu[i])
 "Lc represents the corrected length after the minor losses have been accounted for"
Lc[i]=dx[i]+Le[i]
DP_sh[i]=128*mu[i]*1e-3*mdotw*(Lc[i]/(pi*pipedia[i]^4*rho[i]))"Pipe pressure loss--shear"
End

"Total Shear Pressure Loss Around Water Loop"
Delta_P_shear_tot=sum(DP_sh[i],i=1,6)

"Requirement for Closed Loops"
Delta_P_shear_tot+Delta_P_static_tot=0

"-----Fraser's ncheloop model------equations taken manually from Fraser's results"
ncheloopshear=3.2634e-5+.18162*mdotw+55.857*mdotw^2+53.582*mdotw^3-
20745*mdotw^4+4.1024e5*mdotw^5
ncheloopT[4]=-34.603+4.6907*T[7]-.14017*T[7]^2+.0022888*T[7]^3-1.8763e-
5*T[7]^4+6.0846e-8*T[7]^5
ncheloopqhx=-1106.2+48.432*T[7]+.31548*T[7]^2-.0018182*T[7]^3
ncheloopmdotw=-.078964+.0077423*T[7]-.0002637*T[7]^2+5.0414e-6*T[7]^3-5.4198e-
8*T[7]^4+3.0873e-10*T[7]^5-7.2915e-13*T[7]^6

"--------Percent error-------based on Fraser's model"
T_pcterr[4]=100*abs(T[4]-ncheloopT[4])/ncheloopT[4]
qhx_pcterr=100*abs((qhx-ncheloopqhx)/ncheloopqhx)
mdotw_pcterr=100*abs(mdotw-ncheloopmdotw)/ncheloopmdotw
```

Detailed Model:

```
"Steady State System Model Analysis of Shell and Coil NCHE-----DETAILED MODEL
written by John Avina,  Solar Energy Lab, University of Wisconsin-Madison, January 1995

This EES deck covers Fraser's 'steady state case' in which:
symbols:  g represents glycol (collector fluid), w represents water
INPUTS:      Tgi (in parametric table as T[7])
PARAMETERS:     Twi, T_tank, Density_tank, mdotg
OUTPUTS:     Qhx, Tgo, Two, mdotw
Physically this represents:       1) a NCHE, with glycol flow in and out
```

2) a tank connected to the NCHE inlet, but held at a constant temperature, 19C.
3) the piping connecting the water outflow of the NCHE and the tank.

positions:
[1] NCHE
[2] cold water out of tank
[3] cold water out of tank, into HX (connected to [2])
[4] hot water out of HX
[5] hot water out of HX, into tank (connected to [4])
[6] tank
[7] hot glycol into HX, out of collector
[8] warm glycol out of HX"

```
"************************* FUNCTIONS **********************************"

Procedure tempdist(dx,posit,height_hx,t[3],t[4],T[7],cw,cgly:rho_posit,thx_posit)
dtw:=t[4]-t[3]
dt_hot:=T[7]-t[4]
dto:=T[7]-(dtw*cw/cgly)-t[3]
ax:=t[3]
If (abs(t[3]-t[4])<0.0001) Then
t[4]=t[3]+0.0001
EndIf
If (t[4]>T[7]) Then
dthot=0
EndIf
If (dt_hot=dto) Then
  bx:=0
  cx:=0
  ex:=dtw/height_hx
  goto 20
Else bx:=(dto*dtw)/(dt_hot-dto)
     cx:=dt_hot/dto
     ex:=0
20:
If (cx<0) Then
cx=0.0001
EndIf
thx_posit:=ax+bx*(cx**(posit*dx/height_hx)-1)+ex*posit*dx
rho_posit=(999.8396+18.224944*thx_posit-.00792221*thx_posit^2-55.44846e-6*thx_posit^3+149.7562e-9*thx_posit^4-393.2952e-12*thx_posit^5)/(1+18.159725e-3*thx_posit)
end
```

```
Function kpg(percent,T)
{calculates the thermal conductivity of Dowfrost, a propylene glycol solution, given
percent composition of Dowfrost and the temperature in degrees Kelvin}
B = -0.78595253278 + 0.015561899561*percent - 4.8933521576e-5*percent^2
C = 0.0076866167254 - 0.0001155974176*percent + 3.6603360830e-7*percent^2
D = -9.9976810237e-6 + 1.4560615474e-7*percent - 4.5879383578e-10*percent^2
kpg = B + C*T + D*T^2  {W/m-C}
END

Function mupg(percent,T)
{calculates the viscosity of Dowfrost, a propylene glycol solution, given percent
composition of Dowfrost and the temperature in degrees Kelvin}
B = 71.639163222 - 0.66981698459*T + 0.0019150513174*T^2 - 1.8587687783e-6*T^3
C = 0.27019804611 - 0.0012299975866*T + 1.5045427918e-6*T^2
mupg = exp(B+C*percent)    {Pa s}
END

"************    DRIVING FORCES:  INPUTS  ***************************"
mdotgly=.02
T[2]=19

"GEOMETRY"
heighttank=1.435
dz[1]=0.406;dz[2]=0.064;dz[3]=0;dz[4]=0;dz[5]=0.965;dz[6]=-heighttank
k[2]=6.7;k[3]=2.4;k[4]=0.33;k[5]=2.9
pipedia[2]=0.0328;pipedia[3]=0.0254;pipedia[4]=0.0191;pipedia[5]=0.0253
dx[2]=1.535;dx[3]=0.281;dx[4]=1.384;dx[5]=0.246

g=9.806
height_hx=dz[1]

"geometry inputs"
d_tube=.00635
S_T=.009525
N_T=4
delta_wall=.00076
D_coil_i[1]=0.0254
D_coil_i[2]=0.04445
D_coil_i[3]=0.0635
D_coil_i[4]=0.08255
pitch=0.35*2.54/100

S_L=pitch
```

```
N_L=height_hx/pitch
Duplicate i=1,4
D_coil_o[i]=D_coil_i[i]+0.5*2.54/100
D_coil_ctr[i]=D_coil_i[i]+0.25*2.54/100
End
D_coil_ave=sum(D_coil_ctr[i],i=1,4)/N_T
Z=sqrt((D_coil_ave*pi)^2+pitch^2)
N_TUBES=N_L*N_T
L=N_L*S_L
A_1=N_T*S_T*Z
A_T=(S_T-d_tube)*N_T*Z
A_MIN=A_T
length_tubes_tot=N_TUBES*Z
d_tube_i=d_tube-delta_wall*2
Ac=.25*pi*d_tube_i^2
a_gly=(d_tube-2*delta_wall)/2 "a_gly is the inside tube radius"
R_gly=1.2675*2.54/100
A_S_i=pi*d_tube_i*length_tubes_tot
A_S=pi*d_tube*length_tubes_tot

"
****************************PROPERTIES***************************
"
"Temperatures"
T[2]=T[3]"Assume no heat loss in pipes"
T[4]=T[5]
T[6]=T[2]
T_hx_gly=(T[7]+T[8])/2

"GLYCOL PROPERTIES"
cpgly=(3636.2+3.325*T_hx_gly)"J/kgK"
K_gly=kpg(40,T_hx_gly+273.15)
mu_gly=mupg(40,T_hx_gly+273.15)
rho_gly=1046.3-.40137*T_hx_gly-.24001e-2*T_hx_gly^2-.10304e-5*T_hx_gly^3
mu_gly_wall=mupg(40,Ts+273.15)

cgly=cpgly*mdotgly
Pr_gly=cpgly*mu_gly/K_gly

Re_gly=rho_gly*vel_gly*d_tube_i/mu_gly
De_gly=Re_gly*sqrt(a_gly/R_gly)"Dean number [kakic 5.4]"

"WATER PROPERTIES"
Duplicate i=2,6
```

```
rho[i]=(999.8396+18.224944*t[i]-.00792221*t[i]^2-55.44846e-6*t[i]^3+149.7562e-9*t[i]^4-
393.2952e-12*t[i]^5)/(1+18.159725e-3*t[i])
mu[i]=2.414e-5*10^(247.8/(t[i]+133.15))
End
rho[1]=rhohx
cpw=(4209.4-1.824*T_hx_ave+3.0525e-2*T_hx_ave^2-1.2388e-4*T_hx_ave^3+1.2774e-
7*T_hx_ave^4)
k=0.56619+1.5915e-3*(T_hx_ave+273.15)
mu_hx_ave=2.414e-5*10^(247.8/(T_hx_ave+133.15))

cw=cpw*mdotw
Pr=cpw*mu_hx_ave/k

"WATER WALL PROPERTIES"
k_wall=0.56619+1.5915e-3*(Ts+273.15)
mu_hx_wall=2.414e-5*10^(247.8/(Ts+133.15))
cp_hx_wall=(4209.4-1.824*Ts+3.0525e-2*Ts^2-1.2388e-4*Ts^3+1.2774e-7*Ts^4)
Pr_tubewall=cp_hx_wall*mu_hx_wall/k_wall

"FLOWRATES, VELOCITIES, RE #"
RE_Z=d_tube*rhohx*VELOCITY_BULK/mu_hx_ave
VELOCITY_BULK=mdotw/(A_MIN*rhohx)

vel_gly=mdotgly_coil/(rho_gly*Ac)
mdotgly_coil=mdotgly/N_T

"find temperature distribution in hx in order to get rho hx ave and T hx ave"
Duplicate i=0,10
Call tempdist(dx,i,height_hx,t[3],T[4],T[7],cw,cgly:rho_hx[i],temp_hx[i])
End

"trapezoidal approximation of HX density and temperature"
dx=2*h
n=10
h=height_hx/(2*n)
rho_integ=h*(rho_hx[0]+rho_hx[n]+2*sum(rho_hx[i],i=1,n-1))
Thx_integ=h*(temp_hx[0]+temp_hx[n]+2*sum(temp_hx[i],i=1,n-1))
rhohx=rho_integ/(height_hx)
T_hx_ave=Thx_integ/(height_hx)

"
****************** HEAT TRANSFER CORRELATIONS *********************
"
eff=mdotw*cpw*(T[4]-T[3])/(mdotgly*cpgly*(T[7]-T[3]))
```

```
qhx=mdotgly*cpgly*(T[7]-T[8])
qhx=mdotw*cpw*(T[4]-T[3])

ua=1/(1/(h_i*A_S_i)+1/(h_bar*A_S))
qhx=h_i*A_S_i*(Ts-T_hx_ave)

"
**************EFFECTIVENESS NTU METHOD*************************
"
CSTAR=(mdotgly*cpgly)/(mdotw*cpw)
NTU=ua/(mdotgly*cpgly)
eff=(1-exp(-NTU*(1-CSTAR)))/(1-CSTAR*exp(-NTU*(1-CSTAR)))
"
************ZUKAUSKAS HEAT TRANSFER CORRELATIONS ***********
"
Nus_d=.9*RE_Z^.4*Pr^.36*(Pr/Pr_tubewall)^.25*(mu_hx_ave/mu_hx_wall)^.14
h_bar=Nus_d*k/d_tube

"
*********FIND hi FOR INSIDE OF GLYCOL TUBES--Manlapaz and Churchill's correlation
for helical flow heat transfer
"
h_i=Nus_gly*K_gly/d_tube_i
x3=(1+1342/(De_gly^2*Pr_gly))^2;x4=1.0+1.15/Pr_gly
Nus_gly=((4.364+4.636/x3)^3+1.816*(De_gly/x4)^(3/2))^(1/3){*(mu_hx_ave/mu_hx_wall)^.14
}{4.36}

"
*******************PRESSURE LOSSES ***************************
"
"Jakob's Correlation"
c_1=.176;c_2=.34*S_L/d_tube;n_j=.43+1.13*d_tube/S_T;m_j=.15
f_j=(c_1+c_2/((S_T/d_tube-1)^n_j))*RE_Z^(-m_j)
dpjakob=f_j*N_L*(1/2*rho[3]*VELOCITY_BULK^2)*(mu_hx_wall/mu[3])^.14*1e-3

"Minor losses in HX"
"1) OUTLET PIPE ENTRANCE"
K3=.4
DIA3=.75*2.54/100
LE3=K3*mdotw/(16*pi*mu[4])
MINORLOSSoutlet=128*mu[4]*mdotw*LE3/(pi*DIA3^4*rho[4])*1e-3

"2) INLET PIPE ENTRANCE"
K4=1
```

```
DIA4=.5*2.54/100
LE4=K4*mdotw/(16*pi*mu[3])
MINORLOSSinlet=128*mu[3]*mdotw*LE4/(pi*DIA4^4*rho[3])*1e-3

"3) 90' bend--HOOPER'S CORRELATION"
k_1_90=800;k_inf_90=.25
a_star_90=pipedia[3]/2*100/2.54"must be in inches"
A_c[3]=1/4*pi*pipedia[3]^2
vel[3]=mdotw/(A_c[3]*rho[3])
Re_90=rho[3]*vel[3]*pipedia[3]/mu[3]
k_star_90=k_1_90*Re_90^(-1)+k_inf_90*(1+1/2*a_star_90)
LE1_star=k_star_90*mdotw/(16*pi*mu[3])
MINORLOSS_90=128*mu[3]*mdotw*LE1_star/(pi*pipedia[3]^4*rho[3])*1e-3

"4) 45' BEND--HOOPER'S CORRELATION"
k_1_45=500;k_inf_45=.2
a_star_45=pipedia[4]/2*100/2.54"must be in inches"
A_c[4]=1/4*pi*pipedia[4]^2
vel[4]=mdotw/(rho[4]*A_c[4])
Re_45=rho[4]*vel[4]*pipedia[4]/mu[4]
k_star_45=k_1_45*Re_45^(-1)+k_inf_45*(1+1/2*a_star_45)
LE2_star=k_star_45*mdotw/(16*pi*mu[4])
MINORLOSS_45=128*mu[4]*mdotw*LE2_star/(pi*pipedia[4]^4*rho[4])*1e-3

minorlosses_tot=MINORLOSS_45+MINORLOSS_90+MINORLOSSinlet+MINORLOSSoutle
t
dp_j_tot=dpjakob+minorlosses_tot

"Static Pressure Losses Around Water Loop"
Duplicate i=1,6
DP_st[i]=rho[i]*g*dz[i]*1e-3
End
Delta_P_static_tot=sum(DP_st[i],i=1,6)


"Shear Pressure Losses Around Water Loop"
DP_sh[1]=dp_j_tot
DP_sh[6]=0

"Minor Losses Around Water Loop"
Duplicate i=2,5
"Le represents effective length of minor loss K value"
Le[i]=k[i]*mdotw/(16*pi*mu[i])
"Lc represents the corrected length after the minor losses have been accounted for"
```

```
Lc[i]=dx[i]+Le[i]
DP_sh[i]=128*mu[i]*1e-3*mdotw*(Lc[i]/(pi*pipedia[i]^4*rho[i]))
End

"Total Shear Pressure Drop in Water Loop"
Delta_P_shear_tot=sum(DP_sh[i],i=1,6)

"Requirement in All Closed Loops"
Delta_P_shear_tot+Delta_P_static_tot=0
```