Simulink Report: Static_HX_

Christian Müller

2008-03-06

Model - Static_HX_

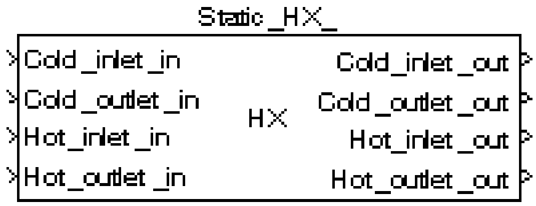


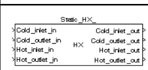
Tabelle 1.1. Static_HX_ Simulation Parameters

Solver ode14x	*ZeroCross* on	*StartTime* 0.0*StopTime* 10.0
RelTol 1e-3	*AbsTol* auto	*Refine* 1
InitialStep auto	*FixedStep* auto	*MaxStep* auto

Tabelle 1.2. Static_HX_ Summary Information

NumModelInputs	N/A	*NumModelOutputs*	N/A
NumVirtualSubsystems	N/A	*NumNonvirtSubsystems*	N/A
NumNonVirtBlocksInModel	N/A	*NumBlockTypeCounts*	N/A
NumBlockSignals	N/A	*NumBlockParams*	N/A
NumZCEvents	N/A	*NumNonsampledZCs*	N/A

Systems

Name	**Parent**	**Snapshot**	**Blocks**	**Signals**
Static_HX_	<root>		Static_HX_	Static_HX_<1> Static_HX_<2> Static_HX_<3> Static_HX_<4>

Blocks

Tabelle 1.3. Block Type Count

BlockType	**Count**	**Block Names**
Inport	31	Cold_inlet_in, Cold_outlet_in, Hot_inlet_in, Hot_outlet_in, K_1_hot, m_1_hot, rho_hot_inlet, K_1_cold, m_1_cold, rho_cold_inlet, p_cold_inlet, p_cold_outlet, p_hot_inlet, p_hot_outlet, x_H2O_gas_cold, x_CO2_cold, x_H2O_gas_hot, x_CO2_hot, p_0, T_0, eff, m_dot_air_cold, m_dot_air_hot, T_cold_inlet, T_hot_inlet, p_cold_outlet,

BlockType	Count	Block Names
		p_hot_outlet, x_H2O_gas_cold, x_CO2_cold, x_H2O_gas_hot, x_CO2_hot
Outport	14	m_dot_air_cold, m_dot_air_cold_inlet, m_dot_air_cold_outlet, m_dot_air_hot, m_dot_air_hot_inlet, m_dot_air_hot_outlet, rho_cold_outlet, rho_hot_outlet, T_cold_outlet, T_hot_outlet, Cold_inlet_out, Cold_outlet_out, Hot_inlet_out, Hot_outlet_out
Constant	7	K_1_cold, K_1_hot, T_0, m_1_cold, m_1_hot, m_air, p_0
BusSelector	4	Bus Selector1, Bus Selector2, Bus Selector3, Bus Selector4
BusCreator	4	Bus Creator1, Bus Creator2, Bus Creator3, Bus Creator4
Terminator	2	Terminator , Terminator
Stateflow (m)	2	Embedded MATLA Function, Embedded MATLA Function1
S-Function	2	SFunction , SFunction
Demux	2	Demux , Demux
SubSystem	1	Static_HX_
Saturate	1	Saturation1
Lookup2D	1	Lookup Table (2-D)

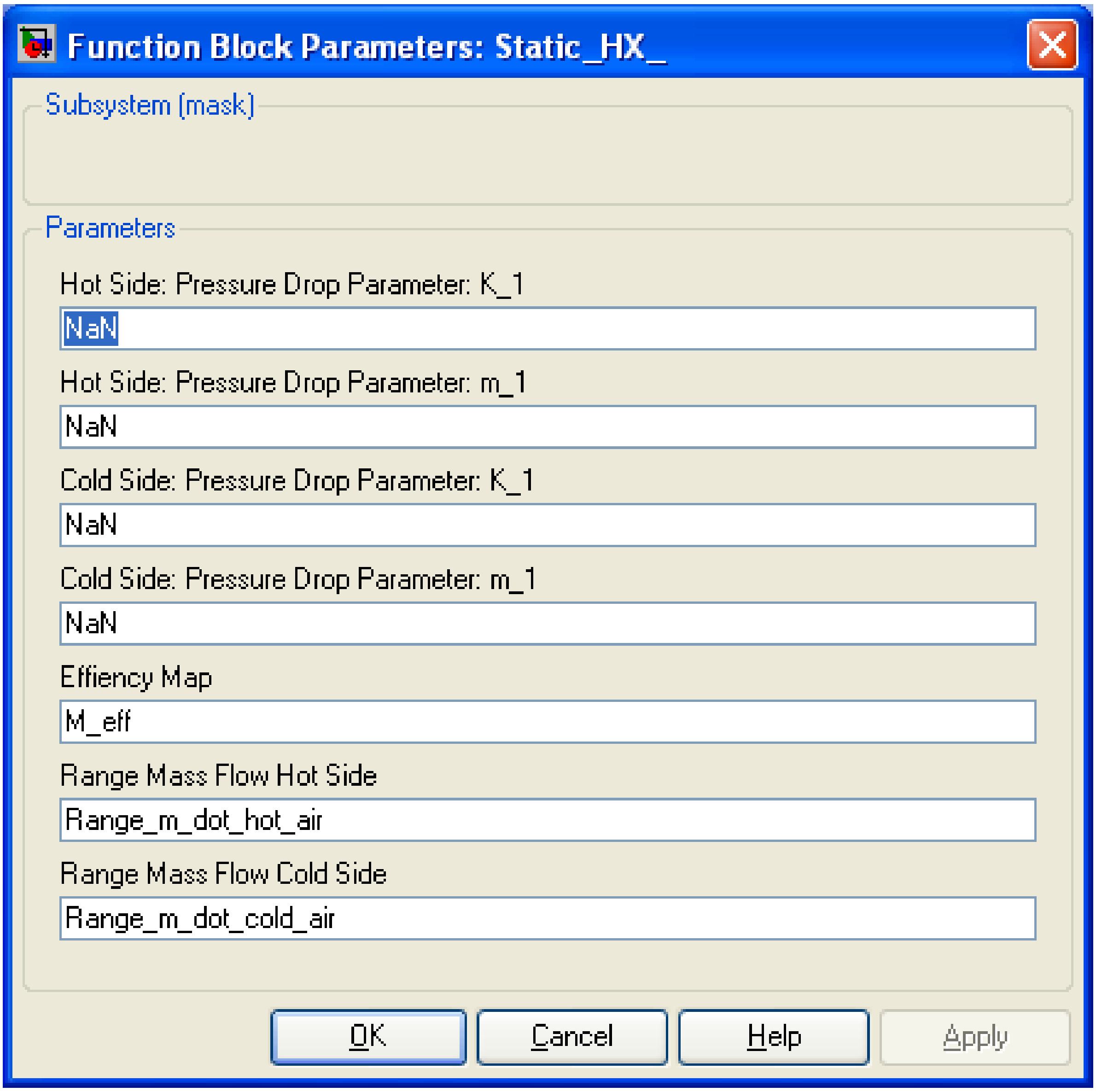
Data and Functions

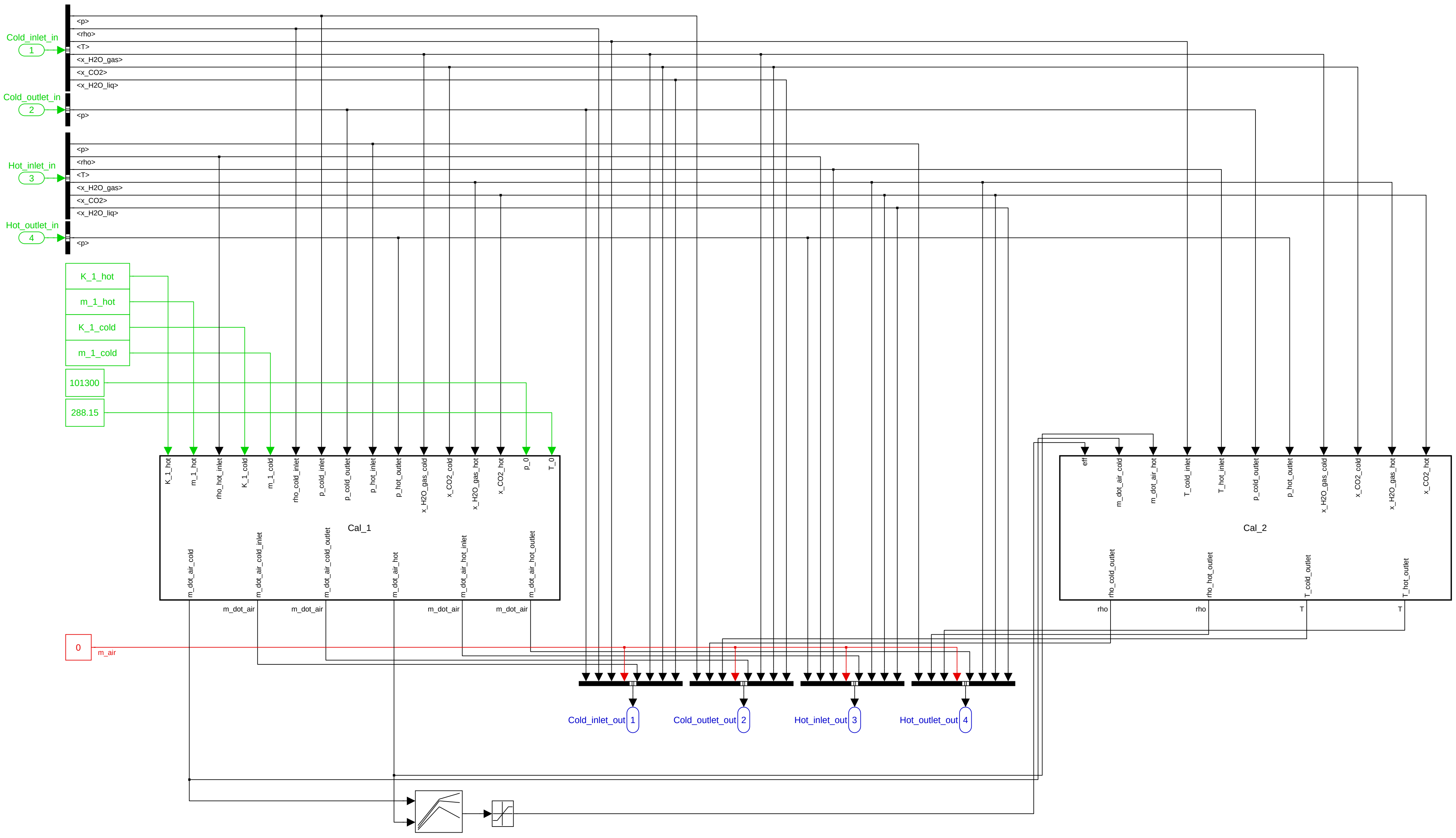
Tabelle 1.4. Model Variables

Variable Name	Parent Blocks	Calling string	Value
M_eff	Static_HX_	`M_eff`	[]

Tabelle 1.5. Model Functions

Function Name	Parent Blocks	Calling string
NaN	Static_HX_ Static_HX_ Static_HX_ Static_HX_	`NaN` `NaN` `NaN` `NaN`


Function Block Parameters: Static_HX_
Subsystem (mask)
Parameters
Hot Side: Pressure Drop Parameter: K_1
NaN
Hot Side: Pressure Drop Parameter: m_1
NaN
Cold Side: Pressure Drop Parameter: K_1
NaN
Cold Side: Pressure Drop Parameter: m_1
NaN
Effiency Map
M_eff
Range Mass Flow Hot Side
Range_m_dot_hot_air
Range Mass Flow Cold Side
Range_m_dot_cold_air
OK
Cancel
Help
Apply



Cold_inlet_in
1
<p>
<rho>
<T>
<x_H2O_gas>
<x_CO2>
<x_H2O_liq>
Cold_outlet_in
2
<p>
Hot_inlet_in
3
<p>
<rho>
<T>
<x_H2O_gas>
<x_CO2>
<x_H2O_liq>
Hot_outlet_in
4
<p>
K_1_hot
m_1_hot
K_1_cold
m_1_cold
101300
288.15
K_1_hot
m_1_hot
rho_hot_inlet
K_1_cold
m_1_cold
rho_cold_inlet
p_cold_inlet
p_cold_outlet
p_hot_inlet
p_hot_outlet
x_H2O_gas_cold
x_CO2_cold
x_H2O_gas_hot
x_CO2_hot
p_0
T_0
Cal_1
m_dot_air_cold
m_dot_air_cold_inlet
m_dot_air_cold_outlet
m_dot_air_hot
m_dot_air_hot_inlet
m_dot_air_hot_outlet
m_dot_air
m_dot_air
m_dot_air
m_dot_air
eff
m_dot_air_cold
m_dot_air_hot
T_cold_inlet
T_hot_inlet
p_cold_outlet
p_hot_outlet
x_H2O_gas_cold
x_CO2_cold
x_H2O_gas_hot
x_CO2_hot
Cal_2
rho_cold_outlet
rho_hot_outlet
T_cold_outlet
T_hot_outlet
rho
rho
T
T
0
m_air
Cold_inlet_out
1
Cold_outlet_out
2
Hot_inlet_out
3
Hot_outlet_out
4

```
function [m_dot_air_cold,m_dot_air_cold_inlet,m_dot_air_cold_outlet,m_dot_air_hot,↙
m_dot_air_hot_inlet,m_dot_air_hot_outlet] = Cal_1(K_1_hot,m_1_hot,rho_hot_inlet,↙
K_1_cold,m_1_cold,rho_cold_inlet,p_cold_inlet,p_cold_outlet,p_hot_inlet,p_hot_outlet,↙
x_H2O_gas_cold,x_CO2_cold,x_H2O_gas_hot,x_CO2_hot,p_0,T_0)

% *************************************************************************
% * Definition of a static heat exchanger
% *
% * Number of inputs :                4
% *
% * Parameter: Characteristic Map:     Efficiency
% *
% *
% * Relevant input variables of Static_HX
% *
% * Pressure:                         p_in
% * Density:                          rho_in
% * Temperature:                      T_in
% * Content water vapor:              x_H2O_gas_in
% * Content CO2:                      x_CO2_in
% * Content water:                    x_H2O_liq_in
% *
% *
% * Relevant output variables of Static_HX
% *
% * Temperature:                      T
% * Mass flow dry air:                m_dot_air_in
% * Content water vapor:              x_H2O_gas
% * Content CO2:                      x_CO2
% * Content water:                    x_H2O_liq
% *
% ************************************************************************
% * Embedded Matlab Function Cal_1:
% *
% * Calculations:
% * 1. Calculation mass flow.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% ************************************************************************

% * 1. Calculation mass flow
R_air                = 287.058;
rho_0                = p_0/(R_air*T_0);
rho_air_cold_inlet   = rho_cold_inlet/(1+x_H2O_gas_cold+x_CO2_cold);
rho_air_hot_inlet    = rho_hot_inlet/(1+x_H2O_gas_hot+x_CO2_hot);
sigma_1_cold         = rho_air_cold_inlet/rho_0;
sigma_1_hot          = rho_air_hot_inlet/rho_0;
m_dot_air_cold       = (abs(p_cold_inlet-p_cold_outlet)*sigma_1_cold/K_1_cold)^↙
(1/m_1_cold);

if p_cold_inlet < p_cold_outlet
   m_dot_air_cold    = 0;
end
```

```
m_dot_air_hot          = (abs(p_hot_inlet-p_hot_outlet)*sigma_1_hot/K_1_hot)^↙
(1/m_1_hot);

if p_hot_inlet < p_hot_outlet
   m_dot_air_hot       = 0;
end

m_dot_air_cold_inlet  = -m_dot_air_cold;
m_dot_air_cold_outlet =  m_dot_air_cold;
m_dot_air_hot_inlet   = -m_dot_air_hot;
m_dot_air_hot_outlet  =  m_dot_air_hot;
% ************************************************************************
```

```
function [rho_cold_outlet,rho_hot_outlet,T_cold_outlet,T_hot_outlet] = Cal_2(eff,↙
m_dot_air_cold,m_dot_air_hot,T_cold_inlet,T_hot_inlet,p_cold_outlet,p_hot_outlet,↙
x_H2O_gas_cold,x_CO2_cold,x_H2O_gas_hot,x_CO2_hot)

% **************************************************************************
% * Definition of a static heat exchanger
% *
% * Number of inputs :               4
% *
% * Parameter: Characteristic Map:     Efficiency
% *
% *
% * Relevant input variables of Static_HX
% *
% * Pressure:                        p_in
% * Density:                         rho_in
% * Temperature:                     T_in
% * Content water vapor:             x_H2O_gas_in
% * Content CO2:                     x_CO2_in
% * Content water:                   x_H2O_liq_in
% *
% *
% * Relevant output variables of Static_HX
% *
% * Temperature:                     T
% * Mass flow dry air:               m_dot_air_in
% * Content water vapor:             x_H2O_gas
% * Content CO2:                     x_CO2
% * Content water:                   x_H2O_liq
% *
% *************************************************************************
% * Embedded Matlab Function Cal_2:
% *
% * Calculations:
% * 1. Definition specific gas constants.
% * 2. Calculation heat transfer.
% *
% *
% * Last modification : 15.03.2008
% * Author : Christian Müller(HAW)
% *
% *************************************************************************

rho_cold_outlet        = 0;
rho_hot_outlet         = 0;
T_cold_outlet          = 0;
T_hot_outlet           = 0;
check                  = 0;

% * 1. Definition specific gas constants.
R_air                  = 287.058;
R_H2O_gas              = 461.523;
R_CO2                  = 188.924;
c_p_air                = 1005;
c_p_H2O_gas            = 1870;
c_p_CO2                = 830;
```

```
R_cold                  = (R_air+x_H2O_gas_cold*R_H2O_gas+x_CO2_cold*R_CO2)/↙
(1+x_H2O_gas_cold+x_CO2_cold);
R_hot                   = (R_air+x_H2O_gas_hot*R_H2O_gas+x_CO2_hot*R_CO2)/↙
(1+x_H2O_gas_hot+x_CO2_hot);
c_p_cold                = (c_p_air+x_H2O_gas_cold*c_p_H2O_gas+x_CO2_cold*c_p_CO2)/↙
(1+x_H2O_gas_cold+x_CO2_cold);
c_p_hot                 = (c_p_air+x_H2O_gas_hot*c_p_H2O_gas+x_CO2_hot*c_p_CO2)/↙
(1+x_H2O_gas_hot+x_CO2_hot);
% ************************************************************************

% * 2. Calculation heat transfer
m_dot_cold              = m_dot_air_cold*(1+x_H2O_gas_cold+x_CO2_cold);
m_dot_hot               = m_dot_air_hot*(1+x_H2O_gas_hot+x_CO2_hot);

if m_dot_cold > 0.01
   if m_dot_hot > 0.01
      check             = 1;
   end
end

if check == 0
   T_cold_outlet        = T_cold_inlet;
   T_hot_outlet         = T_hot_inlet;
   rho_cold_outlet      = p_cold_outlet/(R_cold*T_cold_inlet);
   rho_hot_outlet       = p_hot_outlet/(R_hot*T_hot_inlet);
end

if check > 0
   H_dot_hot_inlet      = m_dot_hot*c_p_hot*T_hot_inlet;
   H_dot_hot_outlet_min = m_dot_hot*c_p_hot*T_cold_inlet;
   H_dot_hot_outlet     = H_dot_hot_inlet-eff*(H_dot_hot_inlet-H_dot_hot_outlet_min);
   T_hot_outlet         = H_dot_hot_outlet/(m_dot_hot*c_p_hot);

   H_dot_cold_inlet     = m_dot_cold*c_p_cold*T_cold_inlet;
   H_dot_cold_outlet    = H_dot_cold_inlet+(H_dot_hot_inlet-H_dot_hot_outlet);
   T_cold_outlet        = H_dot_cold_outlet/(m_dot_cold*c_p_cold);

   rho_cold_outlet      = p_cold_outlet/(R_cold*T_cold_outlet);
   rho_hot_outlet       = p_hot_outlet/(R_hot*T_hot_outlet);
end
% ************************************************************************
```