

---

## Table of Contents

ER468 Combined Cycle code .....	1
Prepare the environment .....	1
Establish working fluid and unit system .....	1
Initialize state-point variables .....	1
Compute state point properties (incorporating given data) .....	1
Carry out first-law analysis and answer questions .....	4

## ER468 Combined Cycle code

```
clear
clc
close 'all'
```

### Prepare the environment

```
EasyProp_path = ' '; %<- Path if EasyProp.py is in your current directory
if count(py.sys.path,EasyProp_path) == 0 % <-- see if desired directory is on
    path
    insert(py.sys.path,int32(0),EasyProp_path); %<-- if not; add it.
end
```

### Establish working fluid and unit system

```
fluid = 'Air';
unitSystem = 'SI';
air = py.EasyProp.simpleFluid(fluid,unitSystem);
water = py.EasyProp.simpleFluid('Water',unitSystem);
```

### Initialize state-point variables

```
numSp = 11;

h = nan(numSp,1);
h_s = nan(numSp,1);
s = nan(numSp,1);
s_s = nan(numSp,1);
P = nan(numSp,1);
T = nan(numSp,1);
x = nan(numSp,1);
```

### Compute state point properties (incorporating given data)

```
Patm = 101.3; % kPa
Tatm = 15; % C
```

---

```

Tmax = 770;
Texh = 250; % C

rp = 17;
eta_c = 0.89;

r_el = 4;
eta_t = 0.92;

% dP parameters

noDP = 0;
switch noDP
    case 1
        % ignore hydraulic losses
        dpInlet = 0; % kPa
        dP_hx = 0; % kPa
        dP_exh = 0; % kPa
    case 0

        dpInlet = 1; % kPa
        dP_hx = 10; % kPa
        dP_exh = 5; % kPa
end

% compute pressures first
P(1) = Patm - dpInlet;
P(2) = rp*P(1);
P(3) = P(2)-dP_hx;
P(4) = P(3)/r_el;
P(5) = P(4) - dP_hx;
P(7) = Patm+dP_exh;
P(6) = P(7) + dP_hx;

% compute remainin state point properties
T(1) = Tatm;
h(1) = air.h_pT(P(1),T(1));
s(1) = air.s_pT(P(1),T(1));

s_s(2) = s(1);
h_s(2) = air.h_ps(P(2),s_s(2));
h(2) = h(1) - (h(1) - h_s(2))/eta_c;
s(2) = air.s_ph(P(2),h(2));
T(2) = air.T_ph(P(2),h(2));

T(3) = Tmax; % C, given
h(3) = air.h_pT(P(3),T(3));
s(3) = air.s_pT(P(3),T(3));

s_s(4) = s(3);
h_s(4) = air.h_ps(P(4),s_s(4));
h(4) = h(3) - (h(3)-h_s(4))*eta_t;
s(4) = air.s_ph(P(4),h(4));
T(4) = air.T_ph(P(4),h(4));

```

---

---

```

T(5) = Tmax;
h(5) = air.h_pT(P(5),T(5));
s(5) = air.s_pT(P(5),T(5));

s_s(6) = s(5);
h_s(6) = air.h_ps(P(6),s_s(6));
h(6) = h(5) - (h(5)-h_s(6))*eta_t;
s(6) = air.s_ph(P(6),h(6));
T(6) = air.T_ph(P(6),h(6));

T(7) = Texh;
h(7) = air.h_pT(P(7),T(7));
s(7) = air.s_pT(P(7),T(7));

% skip to state point 8;
x(8) = 0;
P(8) = 8; % kPa
T(8) = water.Tsat_p(P(8));
h(8) = water.hL_p(P(8));
s(8) = water.sL_p(P(8));

eta_p = 0.8;
P(9) = 3500; % kPa, given
s_s(9) = s(8);
h_s(9) = water.h_ps(P(9),s_s(9));
h(9) = h(8) - (h(8) - h_s(9))./eta_p;
s(9) = water.s_ph(P(9),h(9));
T(9) = water.T_ph(P(9),h(9));

P(10) = P(9); % isobaric heat addition
T(10) = water.Tsat_p(P(10));
h(10) = water.hV_p(P(10));
s(10) = water.sV_p(P(10));

P(11) = P(8);
s_s(11) = s(10);
h_s(11) = water.h_ps(P(11),s_s(11));
h(11) = h(10) - (h(10)-h_s(11))*eta_t;%<- assume same turbine efficiency
s(11) = water.s_ph(P(11),h(11));
x(11) = water.x_ph(P(11),h(11));
T(11) = water.Tsat_p(P(11));

SP = {'1','2','3','4','5','6','7','8','9','10','11',};
Tb = table(T,P,h,s,'RowName',SP);
fprintf('FHR Combined Cycle First-law analysis\n\n');
fprintf('State Point Property Data:\n');
disp(Tb);

```

*FHR Combined Cycle First-law analysis*

*State Point Property Data:*

---

	$T$	$P$	$h$	$s$
	<hr/>	<hr/>	<hr/>	<hr/>
1	15	100.3	414.38	3.8491
2	407.56	1705.1	819.2	3.9166
3	770	1695.1	1222.9	4.3926
4	482.65	423.77	900.16	4.4304
5	770	413.77	1222.1	4.7977
6	503.72	116.3	923.06	4.8317
7	250	106.3	653.29	4.4379
8	41.509	8	173.84	0.59249
9	41.825	3500	178.24	0.59529
10	242.56	3500	2802.6	6.1243
11	41.509	8	1985.5	6.3501

## Carry out first-law analysis and answer questions

```

m_a = 1;

m_s = m_a*(h(6)-h(7))/(h(10)-h(9));

q_s = h(3)-h(2) + h(5)-h(4);
q_r = h(1)-h(7) + (m_s/m_a)*(h(8)-h(11));
q_net = q_s + q_r;
fprintf('Net specific heat: %g kJ/kg \n',q_net);

w_c = h(1)-h(2);
w_t1 = h(3)-h(4);
w_t2 = h(5)-h(6);
w_t3 = (m_s/m_a)*(h(10)-h(11));
w_p = (m_s/m_a)*(h(8)-h(9));
w_net = w_c+w_t1+w_t2+w_t3+w_p;
fprintf('Net specific work: %g kJ/kg \n',w_net);

eta_th = w_net/q_s;
fprintf('Thermal Efficiency: %5.2f percent \n',eta_th*100);

eta_th_noCoGen = (w_c+w_t1+w_t2)/q_s;
fprintf('Thermal Efficiency without cogen: %5.2f percent \n',...
    eta_th_noCoGen*100);

Net specific heat: 300.518 kJ/kg
Net specific work: 300.518 kJ/kg
Thermal Efficiency: 41.41 percent
Thermal Efficiency without cogen: 29.90 percent

```

Published with MATLAB® R2022a