Table of Contents

SCO-2 energy conversion example

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

Add current directory to the Python Path

```
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
```

Initialize Fluid Property object

```
fluid = 'CO2';
units = 'SI';
gas = py.EasyProp.EasyProp(fluid,units);

To_C = 25; % C
To = To_C + 273; % K
Po = 101.3; % kPa
ho = gas.h_pT(Po,To_C);
so = gas.s_pT(Po,To_C);


% function for specific flow exergy neglecting kinetic and potential energy
ef_fun = @(h_val,s_val) h_val - ho - To*(s_val - so);
```

Initialize State Point Data Arrays

```
numSP = 6;
h = nan(numSP,1);
h_s = nan(numSP,1);
s = nan(numSP,1);
s_s = nan(numSP,1);
T = nan(numSP,1);
P = nan(numSP,1);
ef = nan(numSP,1);
```

```
eta_t = 1.0;
eta_c = 1.0;

Pmax = 20000; % kPa
Pmin = 7700; % kPa

xi_regen = 0.9; % let's be optimistic
```

Statepoint Properties

```
P(1) = Pmin;
T(1) = 32; % C
h(1) = gas.h_pT(P(1),T(1));
s(1) = gas.s_pT(P(1),T(1));
ef(1) = ef_fun(h(1),s(1));

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

w_comp = h(1) - h(2);

% come back to state point 3
P(3) = P(2);

% state point 4
P(4) = P(3);
T(4) = 550; % C
h(4) = gas.h_pT(P(4),T(4));
s(4) = gas.s_pT(P(4),T(4));
ef(4) = ef_fun(h(4),s(4));

% state point 5
P(5) = Pmin;
s_s(5) = s(4);
h_s(5) = gas.h_ps(P(5),s_s(5));
h(5) = h(4) - eta_t*(h(4) - h_s(5));
s(5) = gas.s_ph(P(5),h(5));
ef(5) = ef_fun(h(5),s(5));
T(5) = gas.T_ph(P(5),h(5));
w_turb = h(4) - h(5);

% back to state point 3:
h(3) = h(2) + xi_regen*(h(5) - h(2));


T(3) = gas.T_ph(P(3),h(3));
s(3) = gas.s_ph(P(3),h(3));
```

```
ef(3) = ef_fun(h(3),s(3));

q_s = h(4) - h(3);

% state point 6
P(6) = P(5);
h(6) = h(5) - (h(3) - h(2)); % conservation of energy in the regenerator
T(6) = gas.T_ph(P(6),h(6));
s(6) = gas.s_ph(P(6),h(6));
ef(6) = ef_fun(h(6),s(6));

ef_in = ef(4) - ef(3);




q_r = h(1) - h(6);
ef_out = ef(6) - ef(1);

w_net = w_comp + w_turb;
q_net = q_s + q_r;

Ex_d_regen = ef(5) + ef(2) - ef(3) - ef(6);

Flow_ex_bal = (ef_in - ef_out) - (w_net + Ex_d_regen);

fprintf('Exergy balance: %g kJ/kg \n',Flow_ex_bal);



eta_th = w_net/q_s;

fprintf('Thermal efficiency = %g percent \n',eta_th*100);

bwr = abs(w_comp)/w_turb;

fprintf('Net specific work = %g kJ/kg \n',w_net);
fprintf('Net specific heat = %g kJ/kg \n',q_net);
fprintf('Back work ratio = %g percent \n',bwr*100);

fprintf('Flow exergy in: %g kJ/kg \n',ef(4) - ef(3));
fprintf('Flow exergy out: %g kJ/kg \n',ef(6) - ef(1));
fprintf('Specific exergy destroyed in the regen: %g kJ/kg \n',Ex_d_regen);
```

```
Exergy balance: 2.84217e-12 kJ/kg
Thermal efficiency = 61.5029 percent
Net specific work = 120.534 kJ/kg
Net specific heat = 120.534 kJ/kg
Back work ratio = 13.1581 percent
Flow exergy in: 117.147 kJ/kg
Flow exergy out: 2.02829 kJ/kg
Specific exergy destroyed in the regen: -5.41487 kJ/kg
```

```
fprintf('\n\nState point data: \n\n');

% display state point data neatly
SP = {'1','2','3','4','5','6'};
SP_table = table(P,T,h,s,ef,'RowName',SP);
disp(SP_table);
```

State point data:

	P	T	h	s	ef
1	7700	32	306.23	1.3463	214.78
2	20000	59.999	324.5	1.3463	233.04
3	20000	390.7	839.15	2.4765	410.89
4	20000	550	1035.1	2.7411	528.04
5	7700	424.55	896.34	2.7411	389.24
6	7700	34.578	381.68	1.5927	216.8

Published with MATLAB® R2022a