ME 140

Advanced Thermal Systems

Project 5 and 6:
PEM Fuel Cell System Evaluation and Hydrogen Production Analysis

Ian Gunady
Hayden Hall
Minh Ngo Duc
Nick Wilson

May 11, 2018

Part A, Problem 1

Raw Data Plots:

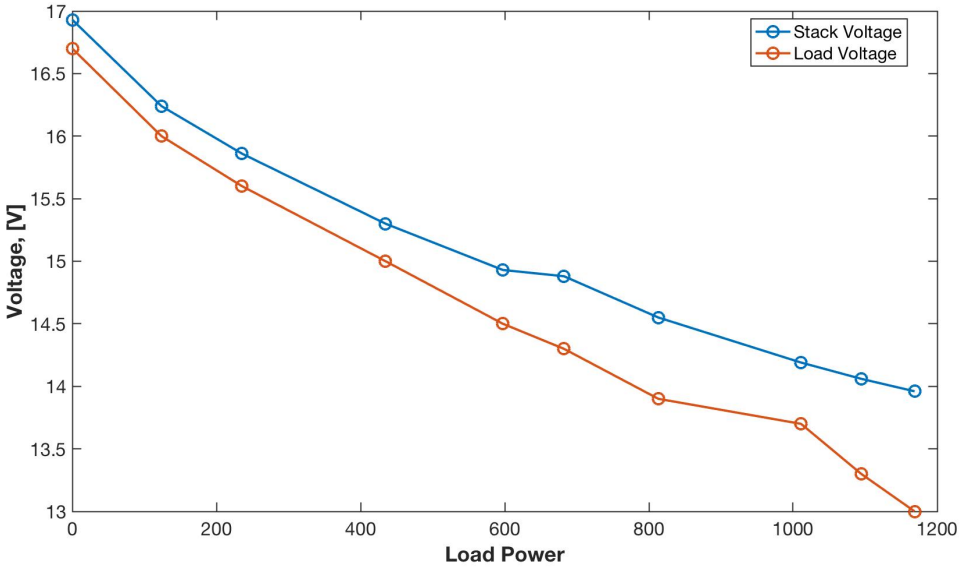

Figure 1: Load and Stack Voltage as a function of Load Power

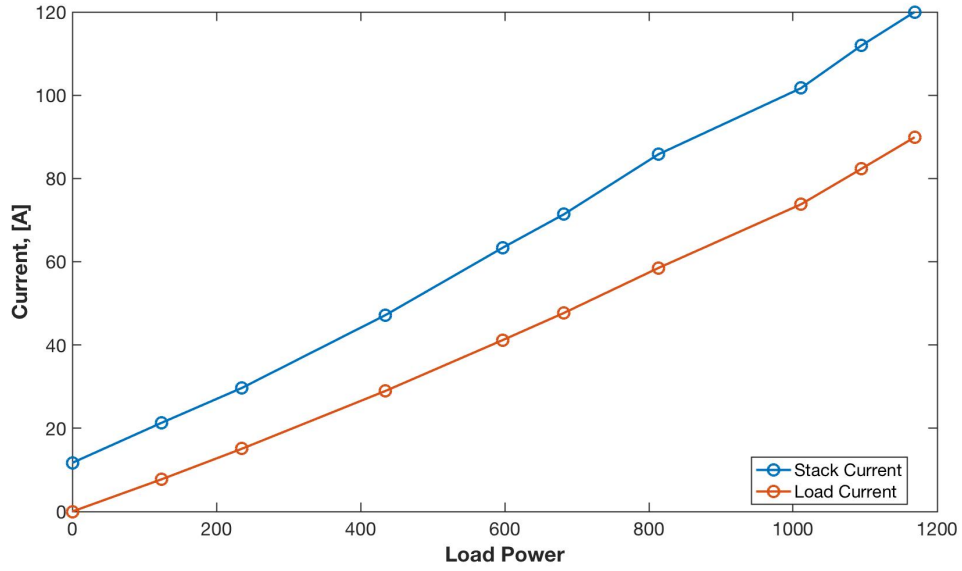

Figure 2: Stack and Load Current as a function of Load Power

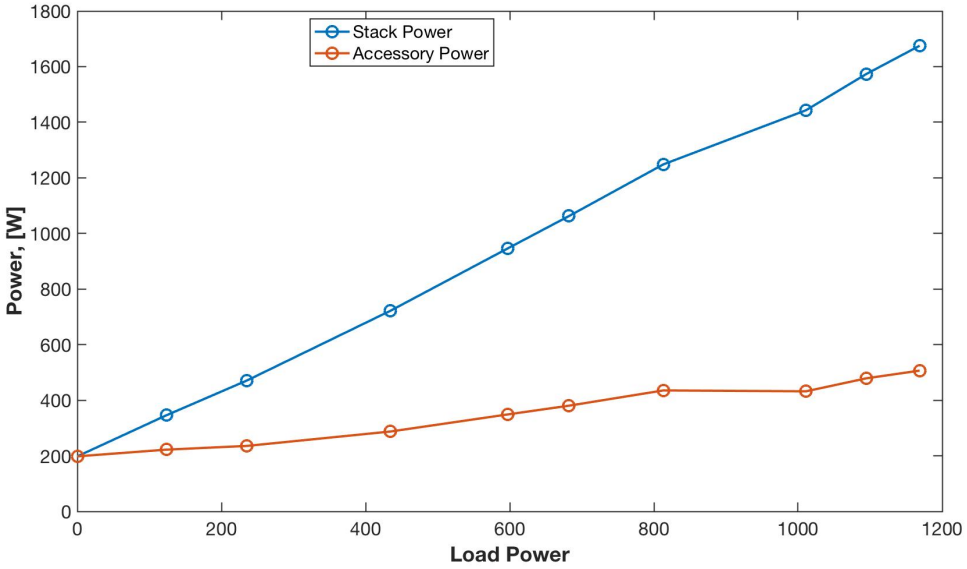

Figure 3: Stack and Accessory Power as a function of Load Power

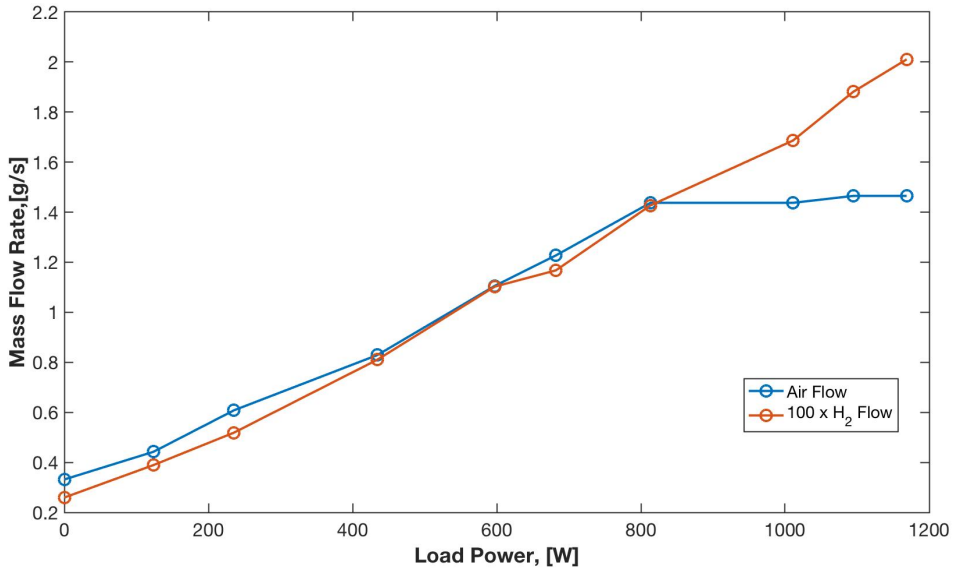

Figure 4: Mass Flow Rate of Air and H_2 as a function of Load Power

Part A Problem 2

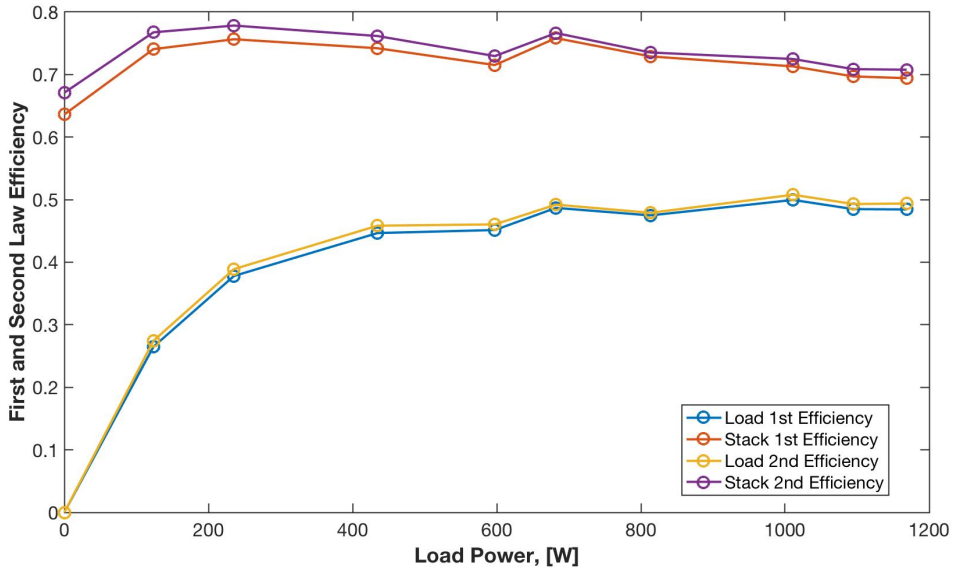

Figure 5: First and Second Law Efficiencies η_I and η_{II} of Load and Stack as functions Load Power

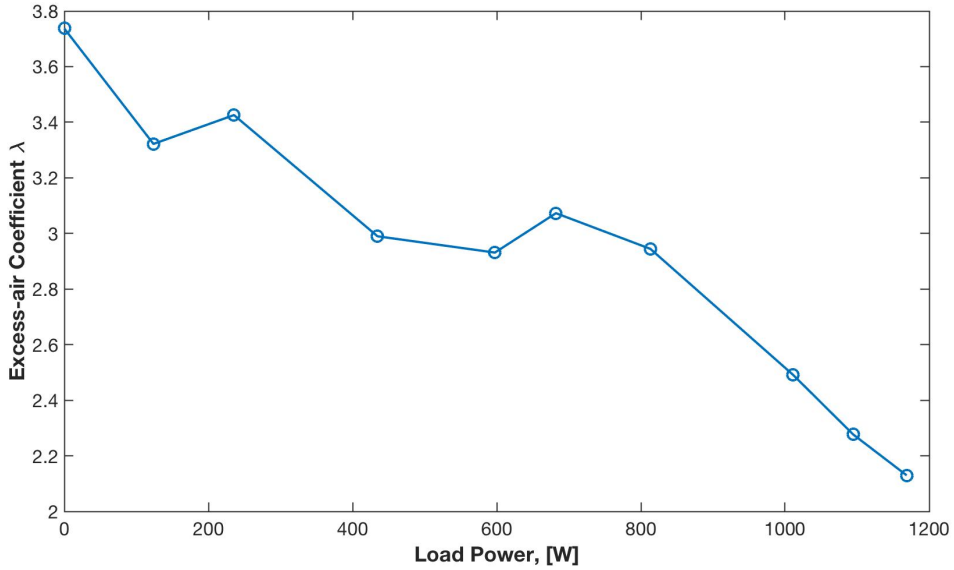

Figure 6: Excess Air Coefficient λ as a function of Load Power

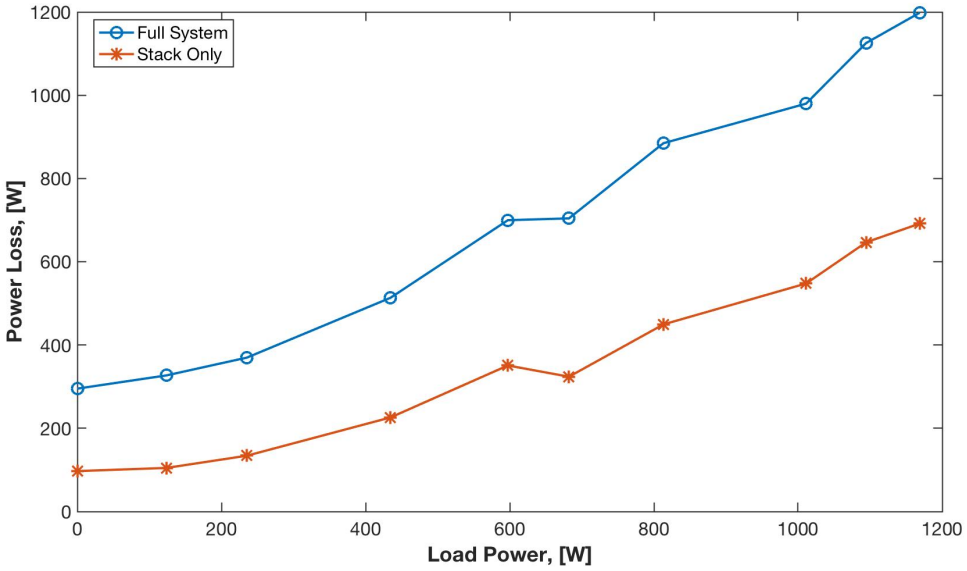

Figure 7: Lost power (stack-only and full system) as a function of load power

Part A, Problem 3

Type of Power Plant	Max First Law Efficiency (%)
Diesel	40-45
Gasoline-Hybrid	35-40

The performance of the H-Power fuel cell that we analyzed produced about 597 Watts at its peak efficiency of 45.2% first law efficiency and 46.1% second law efficiency. The hybrid system net power output of the 2018 Prius is 90 kW (approximately 121 hp). Therefore, this fuel cell would need to be scaled up by a factor of at least 151 in order to attain the same power of 90 kW and implemented in a typical automobile such as the Prius.

For the 20 cell stack we analyzed, the assumed a coolant mass flow of water to be about 22.1 g/s. The heat rejection rate for our fuel cell was found by multiplying the spefici heat of water, c_p, with the product of the mass flow of water and the temperature difference between the inlet and outlet water streams.

$$\dot{Q} = \dot{m} c_p (T_{out} - T_{in}) \tag{1}$$

The heat rejection rate was 55.38 W. Assuming that we scaled up this fuel cell linearly (which we understand is an oversimplification), we would need to reject heat on the order of about 8.35 kJ/s, which would require a coolant (water) flow rate of 3.3289 kg/s, or about 3.3289 L/s recirculating in the engine. Since the current Prius model has a total coolant capacity of 4.9 Liters, this is a feasible volumetric rate to accommodate.

Furthermore, when considering the requirements for practical use in an automobile, we would also need to consider the scaling of accessories required to run the fuel cell. For example, we would need to consider the fact that the membrane that makes fuel cells function must be kept properly hydrated in order for the cell to run. PEM fuel cells operate at temperatures above ambient (to avoid flooding) but below 100^oC (to avoid drying). To maintain this temperature and humidity, a host of accessories are required, from pumps to controllers. All of these devices were relatively large and complex for even just our 20 cell stack in the lab, let alone for a scale up of 151 times the original size. The power needed to run these accessory systems can use up a significant portion of the power produced by the cell, thereby diminishing a fuel cell's feasibility as an automobile's power plant.

All of these devices were relatively large and complex for even just our fuel cell stack in the lab, let alone for a scale up of 151 times the original size. The accessory power needed to run these accessory systems make up a significant portion of total stack power produced. Scaling this up 151 times seem , thereby diminishing a fuel cell's feasibility as an automobile's power plant.

Since it is not an abundant resource, using hydrogen as a fuel is incredibly difficult to use as a fuel. It needs to be produced from other substances such as water and methane through electrolysis processes or Natural Gas (Methane) Reforming. These processes used to produce hydrogen can be energy intensive, and are likely to be powered by fossil fuels, leading to a reduction in the wells-to-wheels efficiency of the fuel cell. On a final note, we would also need to increase the availability of refueling stations for fuel cell cars, which are currently quite limited.

Part B, Problem 1

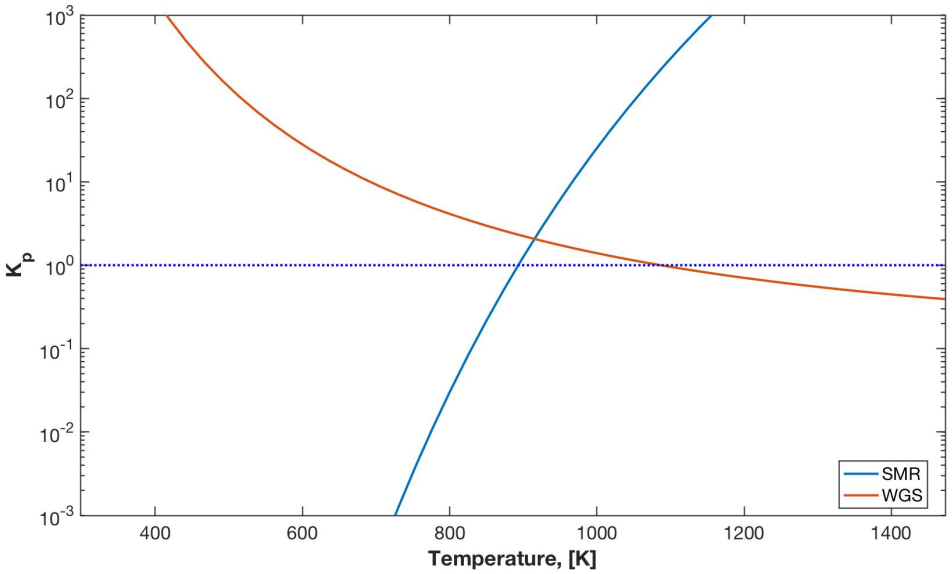

Figure 8: Equilibrium constant, K_p, of Steam-Methane Reformation and Water-Gas Shift as a Function of Temperature

Part B, Problem 2

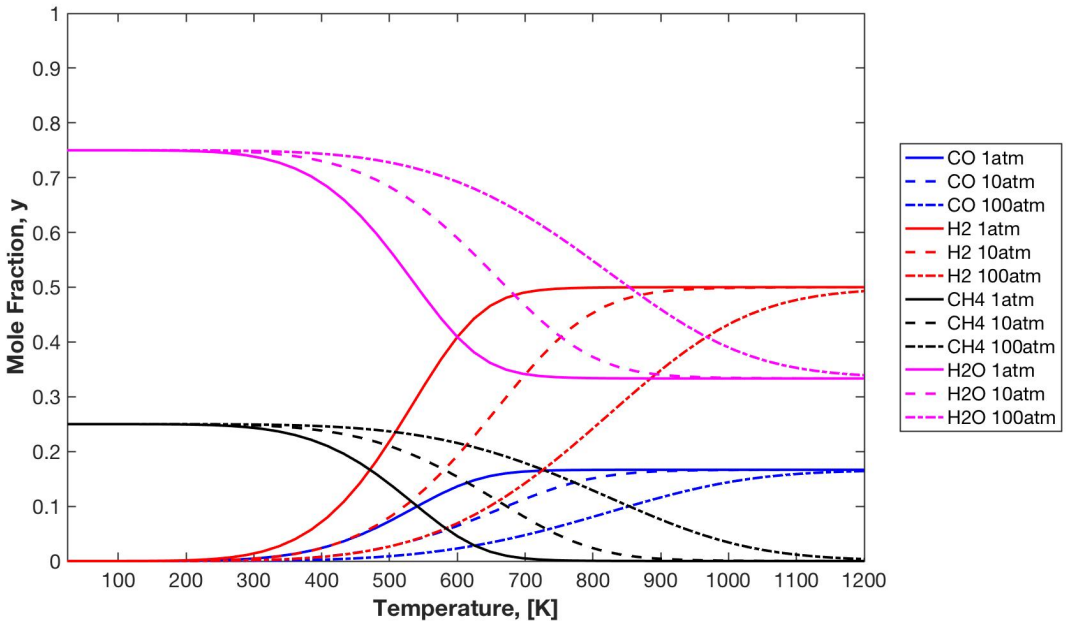

Figure 9: Equilibrium Mole Fractions of the SMR Reaction as Function of Temperature

Part B, Problem 3

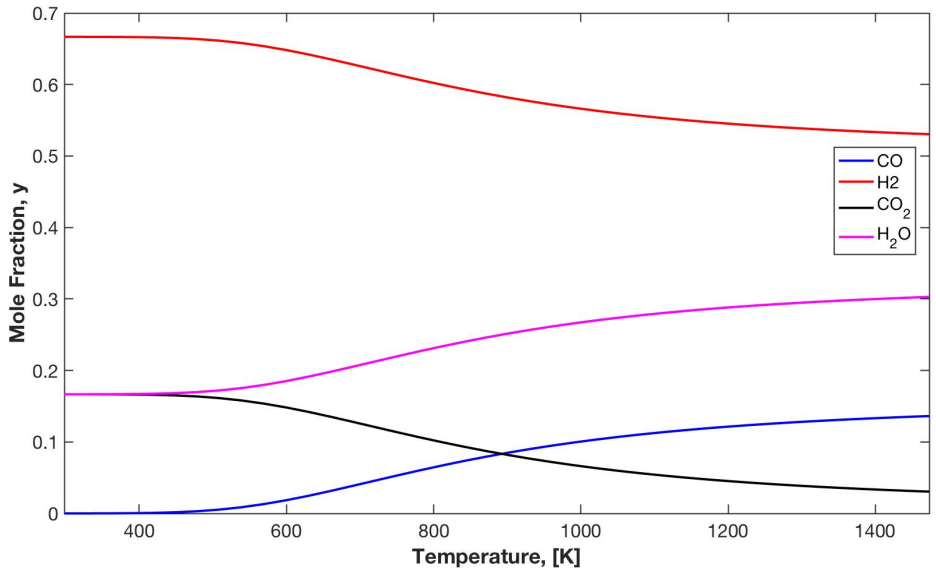

Figure 10: Equilibrium Mole Fractions of the WGS Reaction as Function of Temperature

Part B, Problem 4

	Reformer	**1st Shift Reactor**		**2nd Shift Reactor**	
	Isothermal	Isothermal	Adiabatic	Isothermal	Adiabatic
Inlet Temperature °C	800°C	400°C	400°C	250°C	250°C
Exit Temperature °C	800°C	400°C	467.3°C	250°C	295.7°C
Exit Composition					
mole-% CO_2	5.7	13.21	11.6	15.99	15.37
mole-% H_2O	27.63	20.12	21.74	17.35	17.96
mole-% CO	10.96	3.45	5.07	0.68	1.3
mole-% H_2	55.7	63.21	61.6	65.99	65.37
Heat Addition for Isothermal Reaction (MJ per kg reactants)	3.06	-0.2469	NA	-0.0945	NA
Methane Burned to Heat Reformer (%)	21.09	NA	NA	NA	NA
Efficiency: LHV° H_2 per LHV° CH_4 used (%)	NA	NA	NA	94.14	93.26

Table 1: Table of final metrics

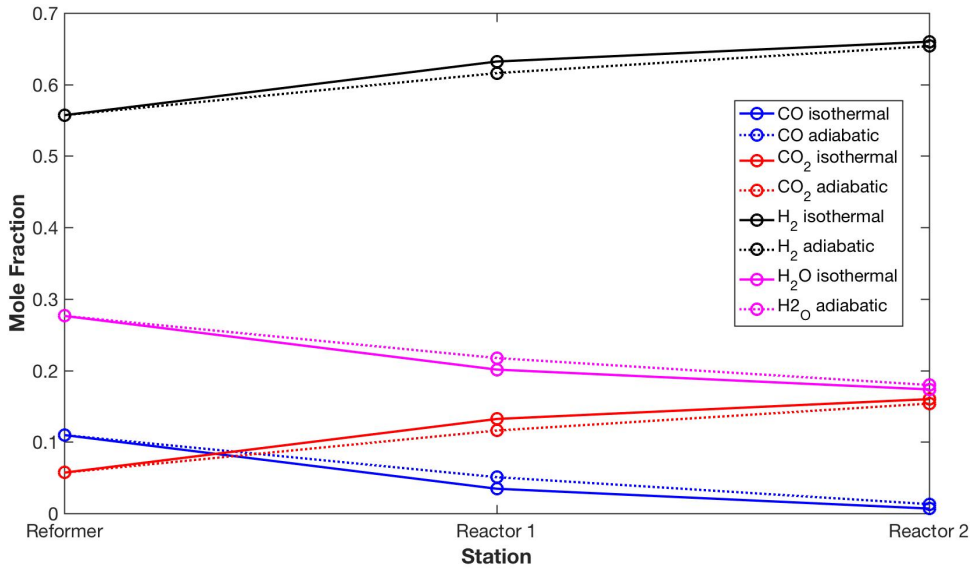

Figure 11: Exit Composition versus System Station

Team Write-Up

Ian: This was a challenging project but I learned a lot. We ran into a lot of bugs and typo errors because a lot of the code is similar or repeated with slight differences. I don't understand how some of the atom balances for the coefficients of each species. After going back and adding comments to the code, most of it makes sense but coming up with everything the first time through was confusingI spent about 20 hours on this project.

Hayden: I found this project to be quite satisfying as it required many concepts from this and previous classes and gave us considerable experience with PEM fuel cells. It was interesting to learn about forming Hydrogen by reforming methane via the three flow reactors and use of SMR and WGS reactions. I put in about 10 hours of work primarily on the part A plot deliverables and populating the table in part B. Fortunately I also learned lots of helpful MATLAB tips and some of its particular quirks. I am still a bit unclear about the accumulation of products in the reformer and two shift reactors. I was thinking that if the mole percentages at the end of the 2nd shift reactor is cumulative, then the accumulation of products would drive the reaction back toward the reactants.

Minh: This was a really difficult project, which took more time than other projects have in previous weeks, but it was a fun learning experience. The majority of our problems came from small typography errors that went unchecked. I think the most useful part of this assignment was actually problem 4 on part B, which made us understand and combine what we had done previously under different conditions to get the desired values. I think this problem was also frustrating in that it took a long time for the problem to run using the iterative methods to find the outlet temperatures knowing that the change in enthalpy was 0 and K_p was a function of temperature. I still don't understand why there is no built in function in MATLAB that could speed up the iterative solving process. In total, I think we spent about 7 hours on project 5, and upwards of 20 hours on project 6.

Nick: I really enjoyed this project despite my skepticism in the beginning due to the extensive chemistry involved in the project. In the end, I found the chemistry very useful and even interesting. This project took more time than some of the other projects because it basically brought together many of the concepts we previously covered in other projects. I worked a total of about 10 hours on coding/debugging all of the problems in part B, with problem 4 taking a good chunk of that time. I also spent about an hour and a half working on problem 3 from part A. I learned a lot about MATLAB from working with this group too that I will take with me in the future.

Matlab Functions and Code

```
1 %% Main Script
```

```
% ME 140 — Reginald Mitchell — Spring 2018
% Projects 5&6 — PEM Fuel Cell System Evaluation and Hydrogen Production Analysis
% Ian Gunady, Hayden Hall, Minh Ngo, Nick Wilson

%% Set up
close all; clear all; clc;
filename = 'PEMData.xlsx';
data = xlsread(filename);

% Constants
C.C2K = 273.15;                     % [K]
C.atm2Pa = 101400;                  % [Pa]
C.ft2m = 0.3048;                    % [m/ft]
C.R_u = 8.314;                      % [J/ mol K]
C.psi2Pa = 6894.76;
C.Patm = 1 *C.atm2Pa;
C.T_stand = 25 + C.C2K;
C.M_H2 = 2.01588e-3;                % [kg/mol]
C.M_H2O = 18.01528e-3;              % [kg/mol]
C.M_CO = 28.01e-3;                  % [kg/mol]
C.M_CO2 = 44.01e-3;                 % [kg/mol]
C.M_air = 28.9647e-3;               % [kg/mol]
C.stack_mult = 3;                   % some multiplier

% LHV, HHV of H2 from Table A-27
C.LHV_H2 = 120e6;               % [J/kg]
C.LHV_H2_bar = 120e6 * C.M_H2;  % [J/mol]

% Coefficients for specific heats
C.coef_H2Ov = [32.24, 0.1923e-2, 1.055e-5, -3.595e-9];      %  specific heats in [J/mol]
C.coef_H2 = [29.11, -0.1916e-2, 0.4003e-5, -0.8704e-9];
C.coef_O2 = [25.48, 1.520e-2, -0.7155e-5, 1.312e-9];
C.coef_N2 = [28.90, -0.1571e-2, 0.8081e-5, -2.873e-9];
C.coef_CO = [28.16, 0.1675e-2, 0.5372e-5, -2.222e-9];
C.coef_CO2 = [22.26, 5.981e-2, -3.501e-5, 7.469e-9];
C.coef_CH4 = [19.89, 5.024e-2, 1.269e-5, -11.01e-9];

% Heat of formation and entropy of formation at STP

% N2
C.h_fo_N2 = 0;                                              % [J/mol]
C.s_o_N2 = 191.61;                                          % [J/mol K]

% O2
C.h_fo_O2 = 0;                                              % [J/mol]
C.s_o_O2 = 205.04;                                          % [J/mol K]

% H2
C.h_fo_H2 = 0;                                              % [J/mol]
C.s_o_H2 = 130.68;                                          % [J/mol K]

% H2O vapor
C.h_fo_H2Ov = -241820;                                      % [J/mol]
C.s_o_H2Ov = 188.83;                                        % [J/mol K]

% H2O liquid
C.h_fo_H2Ol = -285830;                                      % [J/mol]
C.s_o_H2Ol = 69.92;                                         % [J/mol K]
C.cp_H2Ol = 4.18e3;                                         % [J/kg]
C.cp_bar_H2Ol = 4.18e3*C.M_H2O;                             % [J/mol K]

% CO (SMR, WGS)
C.h_fo_CO = -110530;                                        % [J/mol]
C.s_o_CO = 197.65;                                          % [J/mol K]

% CO2 (WGS)
C.h_fo_CO2 = -393520;                                       % [J/mol]
C.s_o_CO2 = 213.8;                                          % [J/mol K]

% CH4 (WGS)
C.h_fo_CH4 = -74850;                                        % [J/mol]
C.s_o_CH4 = 186.16;                                         % [J/mol K]

% Number of mols of fuel per reaction
C.N_fuel = 1;                                               % H2

```

```
% Measured Data
D.NumR = data(1,:);
D.Resistor = data(2,:);
D.V_stack = data(3,:);                                          % [V]
D.I_stack = data(4,:) * C.stack_mult ;                          % [A]
D.V_load = data(5,:);                                           % [V]
D.I_load = data(6,:);                                           % [A]
D.V_dot_H2 =  data(7,:) * C.ft2m^3;                             % [SCFPH] --> [SCMPH]
D.V_dot_air = data(8,:) * C.ft2m^3;                             % [SCFPM] --> [SCMPM]
D.T1 = data(9,:) + C.C2K;                                       % [K]
D.T2 = data(10,:) + C.C2K;
D.T3 = data(11,:) + C.C2K;
D.T4 = data(12,:) + C.C2K;
D.T5 = data(13,:) + C.C2K;
D.T6 = data(14,:) + C.C2K;
D.P_air = data(15,:) * C.psi2Pa +  C.Patm;  % [Pa]
D.P_H2 = data(16,:) *C.psi2Pa +  C.Patm;    % [Pa]

% Total resistance
D.R_tot(1:2) = D.Resistor(1:2);
for i = 3:length(D.Resistor)
    D.R_tot(i) = (1/D.R_tot(i-1) + 1/D.Resistor(i))^-1;
end

%% Part A Problem 1
% Power_stack = everything, all the power
P1.P_stack = D.V_stack .* D.I_stack;                % [W]
% Power_load = "useful work", through the resistors
P1.P_load = D.V_load .* D.I_load;                   % [W]
% Power_accessory = P_stack - P_load
P1.P_acc = P1.P_stack  - P1.P_load;                 % [W]

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P1.plot1 = plot(P1.P_load, D.V_stack, 'o-', P1.P_load, D.V_load, 'o-');
xlabel('Load Power'); ylabel('Voltage, [V]');
plotfixer;
legend('Stack Voltage', 'Load Voltage', 'Location', 'best');

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P1.plot2 = plot(P1.P_load, D.I_stack, 'o-', P1.P_load, D.I_load, 'o-');
xlabel('Load Power'); ylabel('Current, [A]');
plotfixer;
legend('Stack Current', 'Load Current', 'Location', 'best');

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P1.plot3 = plot(P1.P_load, P1.P_stack, 'o-', P1.P_load, P1.P_acc, 'o-');
xlabel('Load Power'); ylabel('Power, [W]');
plotfixer;
legend('Stack Power', 'Accessory Power', 'Location', 'best');

% [SCMPM] -> [mol/sec] convert Standard Cubic Meters Per Minute to mol/sec
P1.Psat_std = T2P_sat(C.T_stand);
P1.P_dryAir = C.Patm - 0.36 * P1.Psat_std;
P1.n_dot_air = P1.P_dryAir * D.V_dot_air /(C.R_u * C.T_stand)/60;
P1.m_dot_air = P1.n_dot_air *C.M_air;

% [SCMFH] -> [mol/sec]  convert Standard Cubic Meters Per Hour to mol/sec
P1.n_dot_H2 = C.Patm * D.V_dot_H2 /(C.R_u * C.T_stand)/3600;
P1.m_dot_H2 = P1.n_dot_H2 *C.M_H2;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P1.plot3 = plot(P1.P_load, P1.m_dot_air*1e3 , 'o-', P1.P_load, P1.m_dot_H2*1e5, 'o-');
xlabel('Load Power, [W]'); ylabel('Mass Flow Rate,[g/s]');
plotfixer;
legend('Air Flow', '100 x H_2 Flow', 'Location', 'best')

%% Part A, Problem 2
% Reactants: H2 + 0.5*L*(O2+3.76*N2) + alpha*H2Ov
% Products: beta*H2Ov + gamma*H2Ol + 0.5*(L-1)*O2 + 0.5*L*3.76*N2

P2.Pm = C.Patm;                             % [Pa]
P2.n_dot_O2 = 1/(1+3.76) * P1.n_dot_air;    % [mol/sec]
P2.L = 2*P2.n_dot_O2 ./ P1.n_dot_H2;        % [unitless]

P2.Psat_react = T2P_sat(D.T1);              % [Pa]
P2.Psat_prod = T2P_sat(D.T2);               % [Pa]
```

```
154
155 P2.y_max_react = P2.Psat_react./P2.Pm;      % [unitless]
156 P2.y_max_prod = P2.Psat_prod./P2.Pm;        % [unitless]
157
158 % 100% RH, saturation at the inlet
159 P2.alpha = P2.y_max_react./(1-P2.y_max_react).*(0.5*P2.L*(1+ 3.76));
160
161 P2.N_H2O_prod = 1 + P2.alpha;
162 P2.N_a_prod =  0.5*(P2.L - 1) + 0.5*P2.L*(3.76);
163
164 P2.y_test_prod = P2.N_H2O_prod ./ (P2.N_H2O_prod + 0.5*(P2.L-1) + 0.5*P2.L*3.76);
165
166 [P2.beta, P2.gamma] = vaporLiquidBalance (D.T2, P2.y_test_prod, ...
167                         P2.y_max_prod, P2.N_a_prod, P2.N_H2O_prod, P2.alpha);
168
169 P2.N_prod = P2.beta + P2.N_a_prod;
170 P2.N_react = 0.5*(P2.L)*(1+3.76) + P2.alpha;
171
172 % N2 REACTANT
173 P2.N_N2_react = (.5*P2.L*3.76);
174 P2.y_N2_react = P2.N_N2_react./(P2.N_react);                          % mol fraction of N2 on reactants side
175 P2.prat_N2_react = P2.y_N2_react.*D.P_air./C.Patm;
176 % pressure ratio
177 P2.ds_o_N2_react = Δ_s (C.coef_N2, D.T1, P2.prat_N2_react);
178 % integral term and log term of entropy
179 P2.dh_bar_N2_react = integral_h (C.coef_N2, D.T1);                    % [J/mol]
180
181 P2.g_bar_N2react = C.h_fo_N2 + P2.dh_bar_N2_react ...
182                     - D.T1.*(C.s_o_N2 + P2.ds_o_N2_react);            % [J/mol]
183
184 % N2 PRODUCT
185 P2.N_N2_prod = (.5*P2.L*3.76);
186 P2.y_N2_prod = P2.N_N2_prod./P2.N_prod;
187 % mole fraction of N2 on PRODUCT side
188 P2.ds_o_N2_prod = Δ_s (C.coef_N2, D.T2, P2.y_N2_prod);
189 % integral term and log term of entropy
190 P2.dh_bar_N2_prod = integral_h (C.coef_N2, D.T2);                     % [J/mol]
191
192 P2.g_bar_N2_prod = C.h_fo_N2 + P2.dh_bar_N2_prod ...
193                     - D.T2.*(C.s_o_N2 + P2.ds_o_N2_prod);
194
195 % O2 REACTANT
196 P2.N_O2_react = (.5*P2.L);
197 P2.y_O2_react = P2.N_O2_react ./(P2.N_react);
198 % mole fraction of O2 on reactants side
199 P2.prat_O2_react = P2.y_O2_react.*D.P_air./C.Patm;
200 % pressure ratio
201 P2.ds_o_O2_react = Δ_s (C.coef_O2, D.T1, P2.prat_O2_react);
202 % integral term and log term of entropy
203 P2.dh_bar_O2_react = integral_h (C.coef_O2, D.T1);       % [J/mol]
204
205 P2.g_bar_O2react = C.h_fo_O2 + P2.dh_bar_O2_react ...
206                     - D.T1.*(C.s_o_O2 + P2.ds_o_O2_react);
207
208 % O2 PRODUCT
209 P2.N_O2_prod = (0.5*(P2.L - 1));
210 P2.y_O2_prod = P2.N_O2_prod ./P2.N_prod;
211 % mole fraction of O2 on reactants side
212 P2.ds_o_O2_prod = Δ_s (C.coef_O2, D.T2, P2.y_O2_prod);
213 % integral term and log term of entropy
214 P2.dh_bar_O2_prod = integral_h (C.coef_O2, D.T2);       % [J/mol]
215
216 P2.g_bar_O2_prod = C.h_fo_O2 + P2.dh_bar_O2_prod ...
217                     - D.T2.*(C.s_o_O2 + P2.ds_o_O2_prod);
218
219 % H2 REACTANT
220 P2.prat_H2_react = D.P_H2./C.Patm;
221 P2.ds_o_H2 = Δ_s (C.coef_H2, D.T1, P2.prat_H2_react);
222 % integral term and log term of entropy
223 P2.dh_bar_H2 = integral_h (C.coef_H2, D.T1);            % [J/mol]
224
225 P2.g_bar_H2 = C.h_fo_H2 + P2.dh_bar_H2 ...
226                 - D.T1.*(C.s_o_H2 + P2.ds_o_H2);
227
228 % H2O vapor REACTANT
229 P2.y_H2Ov_react = P2.alpha./P2.N_react;
```

```
P2.prat_H2Ov_react = P2.y_H2Ov_react.*D.P_air./C.Patm;
P2.ds_o_H2Ov_react = Δ_s (C.coef_H2Ov, D.T1, P2.prat_H2Ov_react);
% integral term and log term of entropy
P2.dh_bar_H2Ov_react = integral_h (C.coef_H2Ov, D.T1);    % [J/mol]

P2.g_bar_H2Ov_react = C.h_fo_H2Ov + P2.dh_bar_H2Ov_react ...
                        - D.T1.*(C.s_o_H2Ov + P2.ds_o_H2Ov_react);

% H2O vapor PRODUCT
P2.y_H2Ov_prod = P2.beta./P2.N_prod;
P2.ds_o_H2Ov_prod = Δ_s (C.coef_H2Ov, D.T2, P2.y_H2Ov_prod);
% integral term and log term of entropy
P2.dh_bar_H2Ov_prod = integral_h (C.coef_H2Ov, D.T2);       % [J/mol]

P2.g_bar_H2Ov_prod = C.h_fo_H2Ov + P2.dh_bar_H2Ov_prod ...
                    - D.T2.*(C.s_o_H2Ov + P2.ds_o_H2Ov_prod);

% H2O Liquid PRODUCT
P2.dh_bar_H2Ol_prod = C.cp_bar_H2Ol.*(D.T2 - C.T_stand);    % [J/mol]
P2.ds_o_H2Ol_prod = C.cp_bar_H2Ol.*log(D.T2/C.T_stand);

P2.g_bar_H2Ol_prod = C.h_fo_H2Ol + P2.dh_bar_H2Ol_prod ...
                        - D.T2.*(C.s_o_H2Ol + P2.ds_o_H2Ol_prod);

% Delta G [J/mol H2]
P2.DG = P2.beta .* P2.g_bar_H2Ov_prod ...
        + P2.gamma .* P2.g_bar_H2Ol_prod ...
        + 0.5 *(P2.L - 1).* P2.g_bar_O2_prod ...
        + (.5*P2.L*3.76).* P2.g_bar_N2_prod...
        - P2.g_bar_H2...                                    % N_H2 = 1
        - 0.5*P2.L.*P2.g_bar_O2react...                     % N_O2_react = 1 = 1/2 * 2
        - (.5*P2.L.*3.76).* P2.g_bar_N2react...
        - P2.alpha.*P2.g_bar_H2Ov_react;

P2.P_H2O_v = P2.alpha./(P2.alpha+.5.*P2.L.*(1+3.76)).*P2.Pm;
P2.RH_inlet = P2.y_H2Ov_react.*P2.Pm./P2.Psat_react;

P2.LHV = C.LHV_H2_bar * C.N_fuel;                           % [J]

% First Law Efficiency
P2.E_I_load = P1.P_load./(C.LHV_H2.*P1.m_dot_H2);
P2.E_I_stack = P1.P_stack./(C.LHV_H2.*P1.m_dot_H2);

% Second Law Efficiency
P2.E_II_load = P1.P_load./(-P2.DG .* P1.n_dot_H2);
P2.E_II_stack = P1.P_stack./(-P2.DG .* P1.n_dot_H2);

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P2.plot1 = plot(P1.P_load, P2.L , 'o-');
xlabel('Load Power, [W]'); ylabel('Excess-air Coefficient \lambda');
plotfixer;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P2.plot2 = plot(P1.P_load, P2.E_I_load , 'o-', P1.P_load, P2.E_I_stack , 'o-');
hold on;
P2.plot2 = plot(P1.P_load, P2.E_II_load , 'o-', ...
                P1.P_load, P2.E_II_stack , 'o-');
legend('Load 1st Efficiency','Stack 1st Efficiency ',...
    'Load 2nd Efficiency','Stack 2nd Efficiency ',...
    'Location', 'best');
xlabel('Load Power, [W]'); ylabel('First and Second Law Efficiency');
plotfixer;

% Calculate Lost Power in Stack Only and Full System
P2.power_loss_load = -P2.DG.* P1.n_dot_H2 - P1.P_load;
P2.power_loss_stack = -P2.DG.* P1.n_dot_H2 - P1.P_stack;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P2.plot4 = plot(P1.P_load, P2.power_loss_load, 'o-',...
                P1.P_load, P2.power_loss_stack, '*-');
xlabel('Load Power, [W]'); ylabel('Power Loss, [W]');
legend('Full System','Stack Only', 'Location', 'best')
plotfixer;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P2.plot4 = plot(P1.P_stack, P2.power_loss_load, 'o-', ...
```

```
                P1.P_stack, P2.power_loss_stack, '*-');
xlabel('Stack Power, [W]'); ylabel('Power Loss, [W]');
legend('Full System','Stack Only', 'Location', 'best')
plotfixer;

%% Part A, Problem 3
C.cp_H2Ol = 4.18e3;                             % [J/kg]
C.gpm2Ls = 0.06309;                             % [gpm/(L/s)]
P3.m_dot_cool = 0.35 * C.gpm2Ls * 1;            % [gpm] --> [kg/s]
P3.Q_dot_cool1 = P3.m_dot_cool * C.cp_H2Ol *(D.T1(5) - D.T2(5));
P3.P_peak = 597;                                %[W]
P3.P_net = 90e3;                                %[W]
P3.scale = P3.P_net/P3.P_peak;
P3.Q_dot_cool_scale = P3.scale * P3.Q_dot_cool1;
P3.m_dot_cool_scale = P3.m_dot_cool * P3.scale;

%% Part B, Problem 1
% Kp for SMR and WGS reaction
P4.T = linspace(25, 1200, 10) + C.C2K;
P4.Kp_SMR = Kp_SMR_function(P4.T);
P4.Kp_WGS = Kp_WGS_function(P4.T);

%% A semi-log plot (i.e., plot log10 KP vs. T).
figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
P4.plot1 = semilogy(P4.T, P4.Kp_SMR , '-', ...
                        P4.T, P4.Kp_WGS, '-');
hold on;
P4.plot1 = semilogy(P4.T, ones(length(P4.T)) ,'b:');
xlabel('Temperature, [K]'); ylabel('K_p');
axis([P4.T(1), P4.T(end), 1e-3, 1e3]);
legend('SMR','WGS', 'Location', 'best')
plotfixer;

%% Part B, Problem 2
% plot mol fractions of SMR reaction
P5.Prat = [1,10,100];
P5.T = P4.T;

% CH4 + 3*H2O -> x*CO + y*H2 + z*CH4 + w*H2O
% CH4 + H2O -> CO +3yH2
P5.Kp = P4.Kp_SMR;
P5.dx = 1e-5;
P5.x = zeros(length(P5.Prat), length(P5.Kp));
P5.Kp_est = zeros(length(P5.Prat), length(P5.Kp));

for j = 1:length(P5.Prat)
    for i = 1:length(P5.Kp)
        while P5.Kp(i) > P5.Kp_est(j,i)
            P5.y(j,i) = 3 * P5.x(j,i);          % N_H2
            P5.z(j,i) = 1 - P5.x(j,i);          % N_CH4
            P5.w(j,i) = 3 - P5.x(j,i);          % N_H2O

            if (P5.x(j,i) ≥ 0 && P5.y(j,i) ≥ 0 && P5.w(j,i)≥0 && P5.z(j,i) ≥ 0)
                P5.Kp_est(j,i) = (P5.x(j,i) * P5.y(j,i)^3)/...
                            (P5.w(j,i) * P5.z(j,i)) ...
                    * (P5.Prat(j)/(P5.x(j,i)+ P5.y(j,i) + P5.z(j,i) + P5.w(j,i)))^2;
            else
                break
            end
            P5.x(j,i) = P5.x(j,i) + P5.dx;
        end
    end
end
P5.y_total = P5.x + P5.y + P5.z + P5.w;
P5.y_CO = P5.x ./P5.y_total;
P5.y_H2 = P5.y ./P5.y_total;
P5.y_CH4 = P5.z ./P5.y_total;
P5.y_H2O = P5.w ./P5.y_total;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
plot(P5.T-C.C2K, P5.y_CO(1,:),'-b' ,...
        P5.T-C.C2K, P5.y_CO(2,:),'--b',...
        P5.T-C.C2K, P5.y_CO(3,:),'-.b')
hold on;
plot( P5.T-C.C2K, P5.y_H2(1,:),'-r',...
        P5.T-C.C2K, P5.y_H2(2,:),'--r',...
```

```
        P5.T-C.C2K, P5.y_H2(3,:),'-.r')

plot( P5.T-C.C2K, P5.y_CH4(1,:),'-k',...
        P5.T-C.C2K, P5.y_CH4(2,:),'--k',...
        P5.T-C.C2K, P5.y_CH4(3,:),'-.k')

plot( P5.T-C.C2K, P5.y_H2O(1,:),'-m',...
        P5.T-C.C2K, P5.y_H2O(2,:),'--m',...
        P5.T-C.C2K, P5.y_H2O(3,:),'-.m');

axis([P5.T(1)-C.C2K, P5.T(end)-C.C2K, 0, 1]);
legend('CO 1atm', 'CO 10atm', 'CO 100atm',...
    'H2 1atm', 'H2 10atm', 'H2 100atm',...
    'CH4 1atm', 'CH4 10atm', 'CH4 100atm',...
    'H2O 1atm', 'H2O 10atm', 'H2O 100atm',...
    'Location', 'eastoutside');
ylabel('Mole Fraction, y'); xlabel('Temperature, [K]');
plotfixer;

%% Part B Problem 3
% Stoichiometric: H2O + CO -> CO2 + H2
% WGS Reaction: H2O + CO -> xCO2 + yH2 + zH2O + wCO;
P6.T = P4.T;
P6.Kp = P4.Kp_WGS;
P6.dx = 1e-6;
P6.x = zeros(1,length(P6.Kp));
P6.Kp_est = zeros(1, length(P6.Kp));

for i = 1:length(P6.Kp)
    while P6.Kp(i) > P6.Kp_est(i)
        P6.y(i) = P6.x(i) + 3;          % N_H2
        P6.z(i) = 2 - P6.x(i);          % N_CH4
        P6.w(i) = 1 - P6.x(i);          % N_H2O

        if (P6.x(i) ≥ 0 && P6.y(i) ≥ 0 && P6.w(i)≥0 && P6.z(i) ≥ 0)
            P6.Kp_est(i) = (P6.x(i) * P6.y(i))/(P6.w(i) * P6.z(i));
        else
            break
        end
        P6.x(i) = P6.x(i) + P6.dx;
    end
end

P6.y_total = 6;
P6.y_CO2 = P6.x ./ P6.y_total;
P6.y_H2 = P6.y ./ P6.y_total;
P6.y_CO = P6.z ./ P6.y_total;
P6.y_H2O = P6.w ./ P6.y_total;

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot
plot(P6.T, P6.y_CO,'-b')
hold on;
plot( P6.T, P6.y_H2,'-r')

plot( P6.T, P6.y_CO2,'-k')

plot( P6.T, P6.y_H2O,'-m')

axis([P6.T(1), P6.T(end), 0, 0.7]);
legend('CO','H2','CO_2','H_2O',...
    'Location', 'best')
ylabel('Mole Fraction, y'); xlabel('Temperature, [K]');
plotfixer;

%% Part B Problem 4
% Station a
P7a.T_reform = 800 + C.C2K;
P7a.Kp_reform = Kp_WGS_function(P7a.T_reform);

% WGS Reaction: H2O + CO -> xCO2 + yH2 + zH2O + wCO;

% Find composition of mixture from Kp
P7a.func = @(x) x*(x+3)/((1-x)*(2-x)) - P7a.Kp_reform;
P7a.x = fzero(P7a.func, 0);
P7a.y = P7a.x + 3;
P7a.z = 2 - P7a.x;
```

```
458 P7a.w = 1 - P7a.x;
459
460 % Find mol fractions
461 P7a.y_total = P7a.x + P7a.y + P7a.z + P7a.w;
462 P7a.y_CO2 = P7a.x ./P7a.y_total;
463 P7a.y_H2 = P7a.y ./P7a.y_total;
464 P7a.y_CO = P7a.w ./P7a.y_total;
465 P7a.y_H2O = P7a.z ./P7a.y_total;
466
467 % Change in enthalpy from STP
468 P7a.dh_bar_H2Ov = integral_h (C.coef_H2Ov, P7a.T_reform);       % [J/mol]
469 P7a.dh_bar_CH4 = integral_h (C.coef_CH4, P7a.T_reform);         % [J/mol]
470 P7a.dh_bar_CO2 = integral_h (C.coef_CO2, P7a.T_reform);         % [J/mol]
471 P7a.dh_bar_H2 = integral_h (C.coef_H2, P7a.T_reform);           % [J/mol]
472 P7a.dh_bar_CO = integral_h (C.coef_CO, P7a.T_reform);           % [J/mol]
473
474 % Consider everything coming in and then everything coming out.
475 P7a.DH_mol = P7a.y*(C.h_fo_H2 + P7a.dh_bar_H2) ...
476         + P7a.x*(C.h_fo_CO2 + P7a.dh_bar_CO2)...
477         + P7a.w*(C.h_fo_CO + P7a.dh_bar_CO)...
478         + (P7a.z-3)*(C.h_fo_H2Ov + P7a.dh_bar_H2Ov)...
479         - 1*(C.h_fo_CH4 + P7a.dh_bar_CH4);
480
481 P7a.m_react = (16.042 + 3*18.016)*1e-3;
482 P7a.DH_kg = P7a.DH_mol /P7a.m_react /1e6;
483
484 C.MCH4 = 16.04e-3;      %[kg/mol]
485 C.LHV_CH4 = 50.05e6;        %[J/kg]
486 C.LHV_bar_CH4 = C.LHV_CH4 * C.MCH4;
487 P7a.burnedCH4_pc = P7a.DH_mol/(P7a.DH_mol + C.LHV_bar_CH4);
488
489 %% Station b
490 P7b.T_shift = 400 + C.C2K;
491 P7b.Kp_shift = Kp_WGS_function(P7b.T_shift);
492
493 % WGS Reaction: H2O + CO -> xCO2 + yH2 + zH2O + wCO;
494
495 % Find composition of mixture from Kp
496 P7b.func = @(x) x*(x+3)/((1-x)*(2-x)) - P7b.Kp_shift;
497 P7b.x = fzero(P7b.func, 0.5);
498 P7b.y = P7b.x + 3;
499 P7b.z = 2 - P7b.x;
500 P7b.w = 1 - P7b.x;
501
502 % Find mol fractions
503 P7b.y_total = P7b.x + P7b.y + P7b.z + P7b.w;
504 P7b.y_CO2 = P7b.x ./ P7b.y_total;
505 P7b.y_H2 = P7b.y ./ P7b.y_total;
506 P7b.y_CO = P7b.w ./ P7b.y_total;
507 P7b.y_H2O = P7b.z ./ P7b.y_total;
508
509 % Change in enthalpy from STP
510 P7b.dh_bar_H2Ov = integral_h (C.coef_H2Ov, P7b.T_shift);% [J/mol]
511 P7b.dh_bar_CO2 = integral_h (C.coef_CO2, P7b.T_shift);  % [J/mol]
512 P7b.dh_bar_H2 = integral_h (C.coef_H2, P7b.T_shift);    % [J/mol]
513 P7b.dh_bar_CO = integral_h (C.coef_CO, P7b.T_shift);    % [J/mol]
514
515 % Consider everything coming in and then everything coming out
516 P7b.DH_mol = (P7b.z-P7a.z)*(C.h_fo_H2Ov + P7b.dh_bar_H2Ov) ...
517             + (P7b.w-P7a.w)*(C.h_fo_CO + P7b.dh_bar_CO)...
518             + (P7b.x-P7a.x)*(C.h_fo_CO2 + P7b.dh_bar_CO2)...
519             + (P7b.y-P7a.y)*(C.h_fo_H2 + P7b.dh_bar_H2);
520
521 P7b.m_react = P7a.z*C.M_H2O + P7a.w*C.M_CO + P7a.x*C.M_CO2 + P7a.y*C.M_H2;
522 % [kg/mol(reaction)]
523 P7b.DH_kg = P7b.DH_mol/P7b.m_react/1e6;                 % [MJ/kg(reactant)]
524
525 %% Station c
526
527 P7c.dT = 1e-1;
528 P7c.T = P7b.T_shift;
529 P7c.DH_mol = P7b.DH_mol;
530 P7c.H_mol_react = (P7a.z)*(C.h_fo_H2Ov + P7b.dh_bar_H2Ov) ...
531             + (P7a.w)*(C.h_fo_CO + P7b.dh_bar_CO)...
532             + (P7a.x)*(C.h_fo_CO2 + P7b.dh_bar_CO2)...
533             + (P7a.y)*(C.h_fo_H2 + P7b.dh_bar_H2);
```

```
534
535 while P7c.DH_mol < 0
536     P7c.T = P7c.T + P7c.dT;
537     P7c.Kp = Kp_WGS_function(P7c.T);
538     P7c.func = @(x) x*(x+3)/((1-x)*(2-x)) - P7c.Kp;
539     P7c.x = fzero(P7c.func, 0.5);
540
541     % WGS Reaction: H2O + CO -> xCO2 + yH2 + zH2O + wCO;
542
543     P7c.y = P7c.x + 3;
544     P7c.z = 2 - P7c.x;
545     P7c.w = 1 - P7c.x;
546
547     P7c.y_total = 6;
548     P7c.y_CO2 = P7c.x ./ P7c.y_total;
549     P7c.y_H2 = P7c.y ./ P7c.y_total;
550     P7c.y_CO = P7c.w ./ P7c.y_total;
551     P7c.y_H2O = P7c.z ./ P7c.y_total;
552
553     P7c.dh_bar_H2Ov = integral_h (C.coef_H2Ov, P7c.T);  % [J/mol]
554     P7c.dh_bar_CO2 = integral_h (C.coef_CO2, P7c.T);    % [J/mol]
555     P7c.dh_bar_H2 = integral_h (C.coef_H2, P7c.T);      % [J/mol]
556     P7c.dh_bar_CO = integral_h (C.coef_CO, P7c.T);      % [J/mol]
557
558     % Enthalpy at outlet - enthalpy at inlet
559     P7c.DH_mol = (P7c.z)*(C.h_fo_H2Ov + P7c.dh_bar_H2Ov) ...
560                 + (P7c.w)*(C.h_fo_CO + P7c.dh_bar_CO)...
561                 + (P7c.x)*(C.h_fo_CO2 + P7c.dh_bar_CO2)...
562                 + (P7c.y)*(C.h_fo_H2 + P7c.dh_bar_H2)...
563                 - P7c.H_mol_react;
564 end
565
566 %% Station d
567 P7d.T_shift = 250 + C.C2K;
568 P7d.Kp_shift = Kp_WGS_function(P7d.T_shift);
569
570 P7d.dx = 1e-5;
571 P7d.x = 0;
572 P7d.Kp_est = 0;
573
574 % WGS Reaction: H2O + CO -> x CO2 + y H2 +  z H2O + w CO;
575 while P7d.Kp_shift > P7d.Kp_est
576     P7d.y = P7d.x +3;
577     P7d.z = 2 - P7d.x;
578     P7d.w = 1 - P7d.x;
579
580     if (P7d.x ≥ 0 && P7d.y ≥ 0 && P7d.w≥0 && P7d.z ≥ 0)
581         P7d.Kp_est = (P7d.x * P7d.y)/(P7d.w * P7d.z);
582     else
583         break
584     end
585     P7d.x = P7d.x + P7d.dx;
586 end
587
588 P7d.y_total = P7d.x + P7d.y + P7d.z + P7d.w;
589 P7d.y_CO2 = P7d.x ./P7d.y_total;
590 P7d.y_H2 = P7d.y ./P7d.y_total;
591 P7d.y_CO = P7d.w ./P7d.y_total;
592 P7d.y_H2O = P7d.z ./P7d.y_total;
593
594 P7d.dh_bar_H2Ov = integral_h (C.coef_H2Ov, P7d.T_shift);    % [J/mol]
595 P7d.dh_bar_CO2 = integral_h (C.coef_CO2, P7d.T_shift);      % [J/mol]
596 P7d.dh_bar_H2 = integral_h (C.coef_H2, P7d.T_shift);        % [J/mol]
597 P7d.dh_bar_CO = integral_h (C.coef_CO, P7d.T_shift);        % [J/mol]
598
599 % Consider everything coming in and then everything coming out
600 P7d.DH_mol = (P7d.z-P7b.z)*(C.h_fo_H2Ov + P7d.dh_bar_H2Ov) ...
601             + (P7d.w-P7b.w)*(C.h_fo_CO + P7d.dh_bar_CO)...
602             + (P7d.x-P7b.x)*(C.h_fo_CO2 + P7d.dh_bar_CO2)...
603             + (P7d.y-P7b.y)*(C.h_fo_H2 + P7d.dh_bar_H2);
604
605 P7d.m_react = P7b.z*C.M_H2O + P7b.w*C.M_CO + P7b.x*C.M_CO2 + P7b.y*C.M_H2;  % [kg/mol(reaction)]
606 P7d.DH_kg = P7d.DH_mol/P7d.m_react/1e6;                % [MJ/kg(reactant)]
607
608 P7d.N_H2 = P7d.y;
609 P7d.Eff = P7d.N_H2 * C.LHV_H2_bar /(1*C.LHV_bar_CH4 + P7a.DH_mol);
```

```
%% Station e

P7e.dT = [1, 1e-1, 1e-2, 1e-3];
P7e.T = P7d.T_shift;
P7e.DH_mol = P7d.DH_mol;
P7e.DH_mol_react = (P7c.z)*(C.h_fo_H2Ov + P7d.dh_bar_H2Ov) ...
            + (P7c.w)*(C.h_fo_CO + P7d.dh_bar_CO)...
            + (P7c.x)*(C.h_fo_CO2 + P7d.dh_bar_CO2)...
            + (P7c.y)*(C.h_fo_H2 + P7d.dh_bar_H2);

while (P7e.DH_mol < 0 && P7e.T < 750)
    P7e.T = P7e.T + P7e.dT(2);
    P7e.Kp = Kp_WGS_function(P7e.T);
    P7e.x = find_x_from_KpWGS(P7e.Kp);

    % WGS Reaction: H2O + CO -> xCO2 + yH2 + zH2O + wCO;

    P7e.y = P7e.x + 3;
    P7e.z = 2 - P7e.x;
    P7e.w = 1 - P7e.x;

    P7e.y_total = 6;
    P7e.y_CO2 = P7e.x ./ P7e.y_total;
    P7e.y_H2 = P7e.y ./ P7e.y_total;
    P7e.y_CO = P7e.w ./ P7e.y_total;
    P7e.y_H2O = P7e.z ./ P7e.y_total;

    P7e.dh_bar_H2Ov = integral_h (C.coef_H2Ov, P7e.T);      % [J/mol]
    P7e.dh_bar_CO2 = integral_h (C.coef_CO2, P7e.T);        % [J/mol]
    P7e.dh_bar_H2 = integral_h (C.coef_H2, P7e.T);          % [J/mol]
    P7e.dh_bar_CO = integral_h (C.coef_CO, P7e.T);          % [J/mol]

    % Consider everything coming in and then everything coming out
    P7e.DH_mol = (P7e.z)*(C.h_fo_H2Ov + P7e.dh_bar_H2Ov) ...
                + (P7e.w)*(C.h_fo_CO + P7e.dh_bar_CO)...
                + (P7e.x)*(C.h_fo_CO2 + P7e.dh_bar_CO2)...
                + (P7e.y)*(C.h_fo_H2 + P7e.dh_bar_H2)...
                - P7e.DH_mol_react;
end

P7e.N_H2 = P7e.y;
P7e.Eff = P7e.N_H2 * C.LHV_H2_bar /(1*C.LHV_bar_CH4 + P7a.DH_mol);

%% plotting section

P7.stations = [100, 200, 300];
P7.y_CO_isotherm = [P7a.y_CO, P7b.y_CO, P7d.y_CO];
P7.y_CO2_isotherm = [P7a.y_CO2, P7b.y_CO2, P7d.y_CO2];
P7.y_H2O_isotherm = [P7a.y_H2O, P7b.y_H2O, P7d.y_H2O];
P7.y_H2_isotherm = [P7a.y_H2, P7b.y_H2, P7d.y_H2];

P7.y_CO_adiab = [P7a.y_CO, P7c.y_CO, P7e.y_CO];
P7.y_CO2_adiab = [P7a.y_CO2, P7c.y_CO2, P7e.y_CO2];
P7.y_H2O_adiab = [P7a.y_H2O, P7c.y_H2O, P7e.y_H2O];
P7.y_H2_adiab = [P7a.y_H2, P7c.y_H2, P7e.y_H2];

figure('units','normalized','outerposition',[0 0 .75 .75]); % for larger plot

plot(P7.stations, P7.y_CO_isotherm, 'b-o', ...
        P7.stations, P7.y_CO_adiab, 'b:o',...
        P7.stations, P7.y_CO2_isotherm,'r-o', ...
        P7.stations, P7.y_CO2_adiab, 'r:o', ...
        P7.stations, P7.y_H2_isotherm, 'k-o', ...
        P7.stations, P7.y_H2_adiab, 'k:o' ,...
        P7.stations, P7.y_H2O_isotherm,'m-o', ...
        P7.stations, P7.y_H2O_adiab, 'm:o');
legend('CO isothermal','CO adiabatic','CO_2 isothermal','CO_2 adiabatic',...
    'H_2 isothermal','H_2 adiabatic','H_2O isothermal','H2_O adiabatic')
set(gca,'XTickLabel',{'Reformer', 'Reactor 1', 'Reactor 2'})
xlabel('Station');
ylabel('Mole Fraction');
xticks(P7.stations);
plotfixer();
```

```
function Δ_h_bar = integral_h (coef, T2)

% coef = [a,b,c,d]
a = coef(1);
b = coef(2);
c = coef(3);
d = coef(4);

T1 = 298;                                %[K]

Δ_h_bar = a.*(T2 - T1) ...
            + b/2.*(T2.^2 - T1^2)...
            + c/3.*(T2.^3 - T1^3)...
            + d/4.*(T2.^4 - T1^4);       %[J/mol]
end
```

```
function ds = Δ_s (coef, T2, p_ratio)
a = coef(1);
b = coef(2);
c = coef(3);
d = coef(4);

T1 = 298;               % [K]

R_u = 8.314;

% the integral term in the entropy equation
I = a.*log(T2./T1)...
    + b.*(T2 - T1)...
    + c.*(T2.^2 - T1.^2)...
    + d.*(T2.^3 - T1.^3);

% in problems 1 and 2
% log term in equation, mole fraction = pressure ratio
% since P_i = y_i * P_m = y_i * P_o b/c P_m = P_o
% R ln(P_i/P_o)
L = R_u.*log(p_ratio);

ds = I - L;
end
```

```
function Psat = T2P_sat(T)
% temp in Kelvin
Psat = exp(-1.2914e8.*T.^-3 + 8.2048e5.*T.^-2 - 6522.8./T + 25.5887);
end
```

```
function [beta, gamma] = vaporLiquidBalance (T, y_test, y_max, N_a, N_H2O, alpha)

    if (length(y_test) == 1)
        for i = 1:max(length(T),length(y_max))
            if y_test > y_max(i)
                y_actual(i) = y_max(i);
                N_v(i) = y_max(i)*N_a/(1-y_max(i));
                N_l(i) = N_H2O - N_v(i);
            else % y_test ≤ y_max
                y_actual(i) = y_test;
                N_v(i) = N_H2O ;
                N_l(i) = 0;
            end
        end
    elseif (length(y_max) == 1)
        for i = 1:length(y_test)
            if y_test(i) > y_max
                y_actual(i) = P2.y_max;
                N_v(i) = y_max * N_a./(1-y_max);
                N_l(i) = N_H2O - N_v(i);
            else % y_test(i) ≤ y_max
                y_actual(i) = y_test(i);
                N_v(i) = N_H2O ;
```

```
                N_l(i) = 0;
            end
        end
    else
        for i = 1:length(y_test)
            if y_test(i) > y_max(i)
                y_actual(i) = y_max(i);
                N_v(i) = y_max(i) * N_a/(1-y_max(i));
                N_l(i) = N_H2O(i) - N_v(i);
            else % y_test ≤ y_max
                y_actual(i) = y_test(i);
                N_v(i) = N_H2O(i);
                N_l(i) = 0;
            end
        end
    end
% beta = moles of water vapor
% gamma = moles of liquid water
beta = N_v;
gamma = N_l;
end
```

```
function x = find_x_from_KpWGS(Kp)
dx = 1e-6;
x = 0;
Kp_est = 0;

while Kp > Kp_est
    y = x + 3;          % N_H2
    z = 2 - x;          % N_CH4
    w = 1 - x;          % N_H2O

    if (x ≥ 0 && y ≥ 0 && w ≥0 && z ≥ 0)
        Kp_est = (x * y)/(w * z);
    else
        break
    end
    x = x + dx;
end
end
```

```
function Kp_SMR = Kp_SMR_function(T)

coef_H2Ov = [32.24, 0.1923e-2, 1.055e-5, -3.595e-9];      % should be [J/mol]
coef_H2 = [29.11, -0.1916e-2, 0.4003e-5, -0.8704e-9];
coef_CO = [28.16, 0.1675e-2, 0.5372e-5, -2.222e-9];
coef_CH4 = [19.89, 5.024e-2, 1.269e-5, -11.01e-9];

% heat of formation @ STP, H2
h_fo_H2 = 0;                                              % [J/mol]
% entropy of formation @ STP, H2
s_o_H2 = 130.68;                                          % [J/mol K]
% H2O vapor
h_fo_H2Ov = -241820;                                      % [J/mol]
s_o_H2Ov = 188.83;                                        % [J/mol K]
% CO (SMR, WGS)
h_fo_CO = -110530;                                        % [J/mol]
s_o_CO = 197.65;                                          % [J/mol K]
% CH4 (WGS)
h_fo_CH4 = -74850;                                       % [J/mol]
s_o_CH4 = 186.16;                                         % [J/mol K]

% SMR : CH4 + H2O -> CO + 3 H2
R_u = 8.314;

% CH4,  SMR REACTANT
N_CH4 = 1;
%  y_CH4 =  N_CH4 ./ N_SMR_R;                             % mole fraction; assume Pm = 1 atm
dh_bar_CH4 = integral_h ( coef_CH4,  T);                  % [J/mol]
ds_o_CH4 = Δ_s ( coef_CH4,  T, 1);% y_CH4);          % integral term and log term of entropy
g_bar_CH4 =  h_fo_CH4 +  dh_bar_CH4 ...
                -  T.*( s_o_CH4 +  ds_o_CH4);
```

```
32
33 % Steam, H2Ov, SMR REACTANT
34 N_H2Ov_SMR = 1;
35 %  y_H2Ov_SMR =  N_H2Ov_SMR ./ N_SMR_R;                          % mole fraction; assume Pm = 1 atm
36 dh_bar_H2Ov_SMR = integral_h ( coef_H2Ov,  T);                  % [J/mol]
37 ds_o_H2Ov_SMR = Δ_s ( coef_H2Ov,  T, 1);% y_H2Ov_SMR);  % integral term and log term of entropy
38 g_bar_H2Ov_SMR =  h_fo_H2Ov +  dh_bar_H2Ov_SMR ...
39                 -  T.*( s_o_H2Ov +  ds_o_H2Ov_SMR);
40
41 % CO, SMR PRODUCT
42 N_CO_SMR = 1;
43 %  y_CO_SMR =  N_CO_SMR ./ N_SMR_P;                             % mole fraction; assume Pm = 1 atm
44 dh_bar_CO_SMR = integral_h ( coef_CO,  T);                     % [J/mol]
45 ds_o_CO_SMR = Δ_s ( coef_CO,  T, 1);% y_CO_SMR);          % integral term and log term of entropy
46 g_bar_CO_SMR =  h_fo_CO +  dh_bar_CO_SMR ...
47                 -  T.*( s_o_CO+  ds_o_CO_SMR);
48
49 % H2, SMR PRODUCT
50 N_H2_SMR = 3;
51 %  y_H2_SMR =  N_H2_SMR ./ N_SMR_P;                             % mole fraction; assume Pm = 1 atm
52 dh_bar_H2_SMR = integral_h ( coef_H2,  T);                     % [J/mol]
53 ds_o_H2_SMR = Δ_s ( coef_H2,  T, 1);% y_H2_SMR);          % integral term and log term of entropy
54 g_bar_H2_SMR =  h_fo_H2 +  dh_bar_H2_SMR ...
55                 -  T.*( s_o_H2 +  ds_o_H2_SMR);
56
57 % GATHER TERMS FOR SMR
58 DGT_SMR =  N_H2_SMR *  g_bar_H2_SMR...
59             +  N_CO_SMR *  g_bar_CO_SMR ...
60             -  N_H2Ov_SMR *  g_bar_H2Ov_SMR...
61             -  N_CH4 *  g_bar_CH4;
62
63 Kp_SMR = exp(-DGT_SMR ./ (R_u* T));
64
65 end
```

```
1  function Kp_WGS = Kp_WGS_function(T)
2
3  % WGS: H2O + CO -> CO2 + H2
4  R_u = 8.314;                          % [J/ mol K]
5  coef_H2Ov = [32.24, 0.1923e-2, 1.055e-5, -3.595e-9];      % should be [J/mol]
6  coef_H2 = [29.11, -0.1916e-2, 0.4003e-5, -0.8704e-9];
7  coef_CO = [28.16, 0.1675e-2, 0.5372e-5, -2.222e-9];
8  coef_CO2 = [22.26, 5.981e-2, -3.501e-5, 7.469e-9];
9
10 % heat of formation @ STP, H2
11 h_fo_H2 = 0;                                                % [J/mol]
12 % entropy of formation @ STP, H2
13 s_o_H2 = 130.68;                                           % [J/mol K]
14 % heat of formation @ STP, H2O vapor
15 h_fo_H2Ov = -241820;                                       % [J/mol]
16 % entropy of formation @ STP, H2O vapor
17 s_o_H2Ov = 188.83;                                         % [J/mol K]
18 % heat of formation @ STP, CO
19 h_fo_CO = -110530;                                         % [J/mol]
20 % entropy of formation @ STP, CO
21 s_o_CO = 197.65;                                           % [J/mol K]
22 % heat of formation @ STP, CO2
23 h_fo_CO2 = -393520;                                        % [J/mol]
24 % entropy of formation @ STP, CO2
25 s_o_CO2 = 213.8;                                           % [J/mol K]
26
27 % H2Ov WGS REACTANT
28 N_H2Ov_WGS = 1;                                            % number of mols
29 % y_H2Ov_WGS =  N_H2Ov_WGS ./ N_WGS_R;
30 % mole fraction; assume Pm = 1 atm
31 dh_bar_H2Ov_WGS = integral_h ( coef_H2Ov,  T);             % [J/mol]
32 ds_o_H2Ov_WGS = Δ_s ( coef_H2Ov,  T, 1);               % y_H2Ov_WGS);
33 % integral term and log term of entropy
34 g_bar_H2Ov_WGS =  h_fo_H2Ov +  dh_bar_H2Ov_WGS ...
35                 -  T.*( s_o_H2Ov +  ds_o_H2Ov_WGS);
36
37 % CO WGS REACTANT
38 N_CO_WGS = 1;
39 %  y_CO_WGS =  N_CO_WGS ./ N_WGS_R;
```

```
% mole fraction; assume Pm = 1 atm
dh_bar_CO_WGS = integral_h ( coef_CO,  T);
ds_o_CO_WGS = Δ_s ( coef_CO,  T, 1);% y_CO_WGS);
% integral term and log term of entropy
g_bar_CO_WGS =  h_fo_CO +  dh_bar_CO_WGS ...
                -  T.*( s_o_CO+  ds_o_CO_WGS);              % [J/mol]

% CO2 WGS PRODUCT
N_CO2 = 1;
%  y_CO2 =  N_CO2 ./ N_WGS_P;
% mole fraction; assume Pm = 1 atm
dh_bar_CO2 = integral_h ( coef_CO2,  T);                    % [J/mol]
ds_o_CO2 = Δ_s ( coef_CO2,  T, 1);% y_CO2);
% integral term and log term of entropy
g_bar_CO2 =  h_fo_CO2 +  dh_bar_CO2 ...
                -  T.*( s_o_CO2 +  ds_o_CO2);               % [J/mol]

% H2 WGS PRODUCT
N_H2_WGS = 1;
%  y_H2_WGS =  N_H2_WGS ./ N_WGS_P;
% mole fraction; assume Pm = 1 atm
dh_bar_H2_WGS = integral_h ( coef_H2,  T);
ds_o_H2_WGS = Δ_s ( coef_H2,  T, 1);% y_H2_WGS);
% integral term and log term of entropy
g_bar_H2_WGS =  h_fo_H2 +  dh_bar_H2_WGS ...
                -  T.*( s_o_H2 +  ds_o_H2_WGS);             % [J/mol]

% GATHER TERMS FOR WGS
DGT_WGS =  N_H2_WGS *  g_bar_H2_WGS ...
            +  N_CO2 *  g_bar_CO2 ...
            -  N_CO_WGS *  g_bar_CO_WGS ...
            -  N_H2Ov_WGS *  g_bar_H2Ov_WGS;                % [J] per reaction

Kp_WGS = exp(- DGT_WGS ./ ( R_u.* T));

end
```