Table of Contents

```
clear all;
clc;
close all;
```

Part A1

```
% measured values (that is, TA data measured values!)
P_atm = 101325; % Pa
TA_data = xlsread('TA_data.xlsx');
I_load = TA_data(:,1)'; %amps
I_stack = TA_data(:,2)';
V_load = TA_data(:,3)'; %volts
V_stack = TA_data(:,4)';
H2_flow = TA_data(:,5)' .* (1/3600) .* (.3048^3) .* (.0899); %SCFH to kg/s
air_flow = TA_data(:,6)' .* (1/60) .* (.3048^3) .* (1.293); %SCFM to kg/s
T_air_in_stack = TA_data(:,12)' + 273.15; % K
T_air_out_stack = TA_data(:,13)' + 273.15; % K
T_water_reservoir = TA_data(:,14)' + 273.15; % K
T_water_in_stack = TA_data(:,15)' + 273.15; % K
T_water_before_HeatExchange = TA_data(:,16)' + 273.15; % K
T_stack = TA_data(:,17)' + 273.15; % K
P_air_in = TA_data(:,9)' + P_atm;  % Converted to absolute (Pa)
P_H2_in = TA_data(:,11)' + P_atm;  % Converted to absolute (Pa)

%Plots for Part 1
P_load = I_load .* V_load; %watts
P_stack = I_stack .* V_stack; %watts
P_accessory = P_stack - P_load; %watts

figure;
plot(P_load, I_load, P_load, I_stack);
xlabel('Power to Resistor Bank (W)');
ylabel('Current (A)');
title('Load and Stack Currents vs. Load Power');
legend('Load current', 'Stack current','Location','northwest');
set(gcf, 'color', 'w');
plotfixer;

figure;
```

```
plot(P_load, V_load, P_load, V_stack);
xlabel('Power to Resistor Bank (W)');
ylabel('Potential/Voltage (V)');
title('Load and Stack Potentials vs. Load Power');
legend('Load potential', 'Stack potential','Location','southwest');
set(gcf, 'color', 'w');
plotfixer;

% TODO: need to put where net power is zero
figure;
plot(P_load, P_stack, P_load, P_accessory);
xlabel('Power to Resistor Bank (W)');
ylabel('Power (W)');
title('Stack and Accessory Power vs. Load Power');
legend('Stack Power', 'Accessory Power');
set(gcf, 'color', 'w');
plotfixer;

figure;
[ax, h1, h2] = plotyy(P_load, H2_flow * 1000, P_load, air_flow * 1000);
xlabel('Power to Resistor Bank (W)');
ylabel(ax(1),'Mass Flow of Hydrogen (g/s)');
ylabel(ax(2), 'Mass flow of Air (g/s)');
title('Mass Flow Rate of Hydrogen and Air vs. Load Power');
set(gcf, 'color', 'w');
plotfixer;
```

Part A2

```
% lambda vs. P_load
MM.O2 = 32;
MM.N2 = 28.02;
MM.C = 12.01;
MM.H = 1.008;
MM.H2 = 2 * MM.H;
MM.H2O = 18.016;
MM.CH4 = MM.C + 4*MM.H;
MM.CO = MM.C + .5*MM.O2;
MM.CO2 = MM.C + MM.O2;
MM.air = 28.97;


H2_flow_mol_s = H2_flow ./ MM.H2 .* 1000;  %mol/sec
Air_flow_mol_s = air_flow ./MM.air .* 1000; %mol/sec
% divide by 4.76 to account for number of moles of air
lambda = (2*Air_flow_mol_s./H2_flow_mol_s) ./ 4.76; % TODO: check with TA

figure;
plot(P_load, lambda);
xlabel('Power to Resistor Bank (W)');
ylabel('\lambda');
title('\lambda vs. Load Power');
set(gcf, 'color', 'w');
```

```
plotfixer;
% n_1 vs. P_load
for i=1:length(T_stack)
    alpha(i) = john(T_stack(i), lambda(i), P_air_in(i));
    [deltaG_rxn(i)] = lucio(T_stack(i), P_air_in(i),...
                            P_H2_in(i),alpha(i),lambda(i));
end


deltaG = deltaG_rxn .* H2_flow_mol_s; % per mol H2 basis
LHV_H2 = 120 * 10^6;

eta_1_load = P_load ./ (LHV_H2.*H2_flow);
eta_2_load = P_load ./ (-deltaG);

eta_1_stack = P_stack ./ (LHV_H2.*H2_flow);
eta_2_stack = P_stack ./ (-deltaG);

P_lost_load = -deltaG - P_load;
P_lost_stack = -deltaG - P_stack;

figure;
plot(P_load, eta_1_load * 100, P_load, eta_1_stack * 100);
xlabel('Power (W)');
ylabel('\eta_1');
title('First Law Efficiency vs. Load');
legend('Load', 'Stack', 'Location', 'Southeast');
set(gcf, 'color', 'w');
plotfixer;

figure;
plot (P_load, eta_2_load * 100, P_load, eta_2_stack * 100);
xlabel('Power (W)');
ylabel('\eta_2');
title('Second Law Efficiency vs. Load');
legend('Load','Stack');
set(gcf, 'color', 'w');
plotfixer;

figure;
plot(P_load, P_lost_load, P_load, P_lost_stack);
xlabel('Power (W)');
ylabel('Lost Power (W)');
title('Lost Power vs. Load Power');
legend('Load', 'Stack');
set(gcf, 'color', 'w');
plotfixer;
```

Part B1

```
T_range = linspace(25, 1200, 20);
R = 8.3144621; %universal gas constant
```

```
% only supposed to plot 10^-3 < Kp < 10^3
% maybe split into two for loops and insert if statements?
% also could look into refining plot function
for i = 1:length(T_range)
    T = T_range(i) + 273;
    deltaG_smr(i) = lucio_smr(T);
    deltaG_wgs(i) = lucio_wgs(T);
    Kp_smr(i) = exp(-deltaG_smr(i) ./ (R*T));
    Kp_wgs(i) = exp(-deltaG_wgs(i) ./ (R*T));
end

figure;
semilogy(T_range, Kp_smr, T_range, Kp_wgs);
xlabel('Temperature [^{\circ}C]');
ylabel('K_p');
title('Equilibrium Constant vs. Temperature');
legend('K_p SMR','K_p WGS','Location','Northwest');
axis([25 1200 10^-3 10^3]);
set(gcf, 'color', 'w');
plotfixer;
```

Part B2

```
P_range = [1 10 100];
P_s = 1;


for i = 1:length(P_range)
    for j = 1:length(T_range)
        T = T_range(j) + 273.15;
        P = P_range(i);
        syms N_CO N_H2 N_CH4 N_H2O N_total
        assume(N_CO >= 0);
        assume(N_H2 >= 0);
        assume(N_CH4 >= 0);
        assume(N_H2O >= 0);
        assume(N_total >= 0);
        eqns(1) = N_total == N_CO + N_H2 + N_CH4 + N_H2O;
        eqns(2) = Kp_smr(j) == ((N_CO * N_H2^3) / (N_CH4 * N_H2O))...
                  * ((P / P_s) / N_total)^2;
        eqns(3) = 1 == N_CH4 + N_CO;
        eqns(4) = 10 == 4 * N_CH4 + 2 * N_H2O + 2 * N_H2;
        eqns(5) = 3 == N_H2O + N_CO;
        S = solve(eqns, 'Real', true);
        CO = double(S.N_CO);
        H2 = double(S.N_H2);
        CH4 = double(S.N_CH4);
        H2O = double(S.N_H2O);
        total = min(double(S.N_total));
        B2.CO(i, j) = min(CO) / total;
        B2.H2(i, j) = min(H2) / total;
        B2.CH4(i, j) = min(CH4) / total;
        B2.H2O(i, j) = min(H2O) / total;
    end

end

figure;
plot(T_range, B2.CO(1, :), '-g', T_range, B2.H2(1, :), '-r',...
     T_range, B2.CH4(1, :), '-c', T_range, B2.H2O(1, :), '-y',...
     T_range, B2.CO(2, :), '-go', T_range, B2.H2(2, :), '-ro',...
     T_range, B2.CH4(2, :), '-co', T_range, B2.H2O(2, :), '-yo', ...
     T_range, B2.CO(3, :), '-g+', T_range, B2.H2(3, :), '-r+',...
     T_range, B2.CH4(3, :), '-c+', T_range, B2.H2O(3, :), '-y+');
legend('CO 1 atm', 'H_2 1 atm', 'CH_4 1 atm', 'H_2O 1 atm',...
       'CO 10 atm', 'H_2 10 atm', 'CH_4 10 atm', 'H_2O 10 atm',...
       'CO 100 atm', 'H_2 100 atm', 'CH_4 100 atm', 'H_2O 100 atm',...
       'Location', 'bestoutside');
title('Equilibrium Composition of SMR vs. Temperature');
xlabel('Temperature [^{\circ}C]');
ylabel('Mole Fraction');
set(gcf, 'color', 'w');
plotfixer;
```

Part B3

```
for j = 1:length(T_range)
    T = T_range(j) + 273.15;
    syms N_CO N_H2O N_CO2 N_H2 N_total
    assume(N_CO >= 0);
    assume(N_H2O >= 0);
    assume(N_CO2 >= 0);
    assume(N_H2 >= 0);
    assume(N_total >= 0);
    eqns(1) = N_total == N_CO + N_H2O + N_CO2 + N_H2;
    eqns(2) = Kp_wgs(j) == ((N_CO2 * N_H2) / (N_CO * N_H2O));
    eqns(3) = 1 == N_CO + N_CO2;
    eqns(4) = 10 == 2 * N_H2O + 2 * N_H2;
    eqns(5) = 3 == N_CO + N_H2O + 2 * N_CO2;
    S = solve(eqns, 'Real', true);
    CO = double(S.N_CO);
    H2O = double(S.N_H2O);
    CO2 = double(S.N_CO2);
    H2 = double(S.N_H2);
    total = min(double(S.N_total));
    B3.CO(j) = min(CO) / total;
    B3.H2O(j) = min(H2O) / total;
    B3.CO2(j) = min(CO2) / total;
    B3.H2(j) = min(H2) / total;
end

figure;
plot(T_range, B3.CO(:), T_range, B3.H2O(:), T_range, B3.CO2(:),...
     T_range, B3.H2(:));
legend('CO 1 atm', 'H_2O 1 atm', 'CO_2 1 atm', 'H_2 1 atm',...
       'Location', 'bestoutside');
```

```
title('Equilibrium Composition of WGS vs. Temperature');
xlabel('Temperature [^{\circ}C]');
ylabel('Mole Fraction');
set(gcf, 'color', 'w');
plotfixer;
```

Part B4

```
R = 8.3144621; %universal gas constant
P_s = 101325; % Pa
P = 101325; % Pa


% Calculations for First Reformer (800 Celsius)

temp.r1 = 1073; % K, temperature for reformer 1 (r1)
gibbs.r1 = lucio_wgs(temp.r1);
Kp.r1 = exp(-gibbs.r1 ./ (R*temp.r1));

syms N_CO_1 N_H2O_1 N_CO2_1 N_H2_1 N_CH4_1 N_total
assume(N_CO_1 >= 0);
assume(N_H2O_1 >= 0);
assume(N_CO2_1 >= 0);
assume(N_H2_1 >= 0);
assume(N_CH4_1 >= 0);
assume(N_total >= 0);
eqns(1) = N_total == N_CO_1 + N_H2O_1 + N_CO2_1 + N_H2_1 + N_CH4_1;
eqns(2) = Kp.r1 == ((N_CO2_1 * N_H2_1) / (N_CO_1 * N_H2O_1));
eqns(3) = 1 == N_CO_1 + N_CO2_1;
eqns(4) = 10 == 2 * N_H2O_1 + 2 * N_H2_1;
eqns(5) = 3 == N_CO_1 + N_H2O_1 + 2 * N_CO2_1;
eqns(6) = 0 == N_CH4_1;
S_1 = solve(eqns, 'Real', true);
CO_1 = double(S_1.N_CO_1);
H2O_1 = double(S_1.N_H2O_1);
CO2_1 = double(S_1.N_CO2_1);
H2_1 = double(S_1.N_H2_1);
CH4_1 = double(S_1.N_CH4_1);
total = min(double(S_1.N_total));
r1.CO_1 = min(CO_1) / total;
r1.H2O_1 = min(H2O_1) / total;
r1.CO2_1 = min(CO2_1) / total;
r1.H2_1 = min(H2_1) / total;
r1.CH4_1 = min(CH4_1) / total;
r1.total = total;

% Calculations for First Shift Reactor (400 Celsius)
temp.r2 = 673; %K second reformer temperature
gibbs.r2 = lucio_wgs(temp.r2);
Kp.r2 = exp(-gibbs.r2 / (R*temp.r2));

syms N_CO_2 N_H2O_2 N_CO2_2 N_H2_2 N_CH4_2 N_total
assume(N_CO_2 >= 0);
assume(N_H2O_2 >= 0);
assume(N_CO2_2 >= 0);
assume(N_H2_2 >= 0);
assume(N_CH4_2 >= 0);
assume(N_total >= 0);
eqns(1) = N_total == N_CO_2 + N_H2O_2 + N_CO2_2 + N_H2_2 + N_CH4_2;
eqns(2) = Kp.r2 == ((N_CO2_2 * N_H2_2) / (N_CO_2 * N_H2O_2));
eqns(3) = 1 == N_CO_2 + N_CO2_2;
eqns(4) = 10 == 2 * N_H2O_2 + 2 * N_H2_2;
eqns(5) = 3 == N_CO_2 + N_H2O_2 + 2 * N_CO2_2;
eqns(6) = 0 == N_CH4_2;
S_2 = solve(eqns, 'Real', true);
CO_2 = double(S_2.N_CO_2);
H2O_2 = double(S_2.N_H2O_2);
CO2_2 = double(S_2.N_CO2_2);
H2_2 = double(S_2.N_H2_2);
CH4_2 = double(S_2.N_CH4_2);
total = min(double(S_2.N_total));
r2.CO_2 = min(CO_2) / total;
r2.H2O_2 = min(H2O_2) / total;
r2.CO2_2 = min(CO2_2) / total;
r2.H2_2 = min(H2_2) / total;
r2.CH4_2 = min(CH4_2) / total;
r2.total = total;

% Calculations for Second Shift Reactor (250 Celsius)

temp.r2 = 523; %K second reformer temperature
gibbs.r2 = lucio_wgs(temp.r2);
Kp.r3 = exp(-gibbs.r2 / (R*temp.r2));

syms N_CO_3 N_H2O_3 N_CO2_3 N_H2_3 N_CH4_3 N_total
assume(N_CO_3 >= 0);
assume(N_H2O_3 >= 0);
assume(N_CO2_3 >= 0);
assume(N_H2_3 >= 0);
assume(N_CH4_3 >= 0);
assume(N_total >= 0);
eqns(1) = N_total == N_CO_3 + N_H2O_3 + N_CO2_3 + N_H2_3 + N_CH4_3;
eqns(2) = Kp.r3 == ((N_CO2_3 * N_H2_3) / (N_CO_3 * N_H2O_3));
eqns(3) = 1 == N_CO_3 + N_CO2_3;
eqns(4) = 10 == 2 * N_H2O_3 + 2 * N_H2_3;
eqns(5) = 3 == N_CO_3 + N_H2O_3 + 2 * N_CO2_3;
eqns(6) = 0 == N_CH4_3;
S_3 = solve(eqns, 'Real', true);
CO_3 = double(S_3.N_CO_3);
H2O_3 = double(S_3.N_H2O_3);
CO2_3 = double(S_3.N_CO2_3);
H2_3 = double(S_3.N_H2_3);
CH4_3 = double(S_3.N_CH4_3);
total = min(double(S_3.N_total));
r3.CO_3 = min(CO_3) / total;
r3.H2O_3 = min(H2O_3) / total;
r3.CO2_3 = min(CO2_3) / total;
```

```
r3.H2_3 = min(H2_3) / total;
r3.CH4_3 = min(CH4_3) / total;
r3.total = total;

x = [0 1 2 3];
data.CO = [0 r1.CO_1 r2.CO_2 r3.CO_3];
data.H2O = [.75 r1.H2O_1 r2.H2O_2 r3.H2O_3];
data.CO2 = [0 r1.CO2_1 r2.CO2_2 r3.CO2_3];
data.H2 = [0 r1.H2_1 r2.H2_2 r3.H2_3];
data.CH4 = [.25 r1.CH4_1 r2.CH4_2 r3.CH4_3];

% finds T2 and T3 after shift reactors if adiabatic

R = 8.3144621; %universal gas constant

[N_CO, N_H2O, N_CO2, N_H2, total] = wgs_mols(P_atm, Kp.r1); %K,Pa
a1.CO = N_CO / total;
a1.H2O = N_H2O / total;
a1.CO2 = N_CO2 / total;
a1.H2 = N_H2 / total;
a1.total = total;

% First Shift Reactor
LHS = lucio_wgs_n(N_CO, N_H2O, N_CO2, N_H2, 400 + 273);

T = 400 + 273;
RHS = -Inf;
dT = .1; %K
while LHS > RHS
    T = T + dT;
    temp_Kp = exp(-lucio_wgs(T) / (R*T));
    [N_CO, N_H2O, N_CO2, N_H2, total] = wgs_mols(P_atm,temp_Kp); %K,P
    RHS = lucio_wgs_n(N_CO, N_H2O, N_CO2, N_H2, T);
end
a2.CO = N_CO / total;
a2.H2O = N_H2O / total;
a2.CO2 = N_CO2 / total;
a2.H2 = N_H2 / total;
a2.temp = T;
a2.total = total;
a2

% Second Shift Reactor
LHS = lucio_wgs_n(N_CO, N_H2O, N_CO2, N_H2, 250 + 273);
T = 250 + 273;
RHS = -Inf;
dT = .1; %K
while LHS > RHS
    T = T + dT;
    temp_Kp = exp(-lucio_wgs(T) / (R*T));
    [N_CO, N_H2O, N_CO2, N_H2, total] = wgs_mols(P_atm,temp_Kp); %K,P
    RHS = lucio_wgs_n(N_CO, N_H2O, N_CO2, N_H2, T);
end
a3.CO = N_CO / total;
```

```
a3.H2O = N_H2O / total;
a3.CO2 = N_CO2 / total;
a3.H2 = N_H2 / total;
a3.temp = T;
a3.total = total;
a3

x = [0 1 2 3];
data2.CO = [0 a1.CO a2.CO a3.CO];
data2.H2O = [.75 a1.H2O a2.H2O a3.H2O];
data2.CO2 = [0 a1.CO2 a2.CO2 a3.CO2];
data2.H2 = [0 a1.H2 a2.H2 a3.H2];
data2.CH4 = [.25 0 0 0];

figure;
plot(x, data.CH4, '-k');
hold all
plot(x, data.CO, '-r');
plot(x, data.H2O, '-g');
plot(x, data.CO2, '-c');
plot(x, data.H2, '-y');
plot(x, data2.CO, '--r');
plot(x, data2.H2O, '--g');
plot(x, data2.CO2, '--c');
plot(x, data2.H2, '--y');
hold off

title('Exit Compositions for Hydrogen Production Process')
xlabel('Stations [-]');
stations = {'Pre-Reformer','Reformer Exit','Shift Reactor 1','Shift Reactor 2'};
set(gca,'xtick',0:1:3);
set(gca,'XTickLabel',stations);
ylabel('Mole Fraction [-]')
legend('CH_4', 'CO Isothermal', 'H_2O Isothermal', 'CO_2 Isothermal', ...
    'H_2 Isothermal',...
       'CO Adiabatic', 'H_2O Adiabatic', 'CO_2 Adiabatic', ...
       'H_2 Adiabatic', 'Location', 'bestoutside');
set(gcf, 'color', 'white');
plotfixer;
```

Heat Addition Shit

```
total = r1.total;
raw_H_add_reform = (lucio_wgs_n(r1.CO_1 * total, r1.H2O_1 * total, ...
                    r1.CO2_1 * total, r1.H2_1 * total, 1073) ...
               - lucio_smr_n(1, 3, 0, 0, 1073));
H_add.reform = raw_H_add_reform / ((MM.CH4 + 3 * MM.H2O) / 1000) / 10^6;

total = r2.total;
H_add.shift1 = (lucio_wgs_n(r2.CO_2 * total, r2.H2O_2 * total, ...
    r2.CO2_2 * total, r2.H2_2 * total, 400 + 273) ...
               - lucio_wgs_n(r1.CO_1 * total, r1.H2O_1 * total, ...
               r1.CO2_1 * total, r1.H2_1 * total, 400 + 273)) ...
               / ((MM.CH4 + 3 * MM.H2O) / 1000) / 10^6;

total = r3.total;
H_add.shift2 = (lucio_wgs_n(r3.CO_3 * total, r3.H2O_3 * total,...
    r3.CO2_3 * total, r3.H2_3 * total, 250 + 273) ...
               - lucio_wgs_n(r2.CO_2 * total, r2.H2O_2 * total, ...
               r2.CO2_2 * total, r2.H2_2 * total, 250 + 273)) ...
               / ((MM.CH4 + 3 * MM.H2O) / 1000) / 10^6;

H_add
LHV_CH4 = 50050 * 10^3;
burn = (raw_H_add_reform / LHV_CH4);
methane_percent = burn / (burn + MM.CH4 / 1000)

num_iso = r3.H2_3 * r3.total * MM.H2 / 1000 * LHV_H2;
num_adi = a3.H2 * a3.total * MM.H2 / 1000 * LHV_H2;
den = (burn + MM.CH4 / 1000) * LHV_CH4;
eta_iso = num_iso / den
eta_adi = num_adi / den
```

Published with MATLAB® R2013a