

▼ Advanced Chemical Reaction Engineering (CBE 60546)

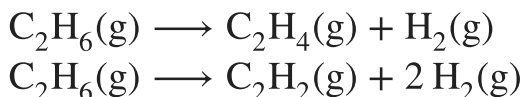
Homework 2

Due September 8, 2021

Carefully and neatly document your answers. You may use a mathematical solver like Jupyter/iPython. Use plotting software for all plots.

▼ 1 Shale, shale everywhere

1.1 Your nation has stumbled upon an enormous reserver of ethane (that's really true!), which you propose to use as a feedstock to produce ethylene, acetylene, and hydrogen via thermal dehydrogenation:



The thermodynamic data [here](#) is from the [CRC Handbook](#), and is based on 1 bar standard state. All the gases can be considered to be ideal over the range of interest here.

1. Plot the standard state enthalpy of ethane vs T from 600 to 1200K. What does the slope of the plot correspond to?

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from scipy.optimize import minimize
from scipy.optimize import root
from scipy.optimize import curve_fit
from google.colab import files # This is required only when you are using Colab.
```

```
# Upload HW2-thermo.csv with colab.files, read it and store it as a Pandas dataframe.
# If you use Jupyter on you own laptop, you don't need to upload the file.
```

```
files.upload() # interactive selection of file
df = pd.read_csv("HW2-thermo.csv")
df.describe()
```

Choose Files HW2-thermo.csv

- **HW2-thermo.csv**(text/csv) - 5621 bytes, last modified: 9/3/2021 - 100% done

Saving HW2-thermo.csv to HW2-thermo.csv

	Row	T (K)	Cp° (J/mol K)	S° (J/mol K)	- (G°-H° (T)) (J/mol K)	(H-H°(T)) (kJ/mo)	ΔHf° (kJ/mol)
count	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000	56.000000
mean	307.250000	857.010714	69.794214	250.533839	216.515464	36.406661	41.846536
std	228.769535	396.011967	34.374405	68.704863	48.629870	33.786517	118.696057
min	156.000000	298.150000	28.836000	130.680000	130.680000	0.000000	-107.292000
25%	169.750000	500.000000	40.236250	195.406750	189.358500	9.775500	-21.000000
50%	183.500000	850.000000	65.645000	256.331500	229.161500	27.364000	17.663500
75%	321.000000	1200.000000	94.969500	298.764750	248.308750	56.370250	94.743500

```
# Read ethane data from row 28 to 41
```

```
df_ethane = df[28:42]
```

```
df_ethane.tail() # show a few lines
```

	Row	Subheader	T (K)	Cp° (J/mol K)	S° (J/mol K)	- (G°- H°(T)) (J/mol K)	(H-H° (T)) (kJ/mo)	ΔHf° (kJ/mol)
		14. ETHANE						
37	193	C₂H₆ (g)	1100.0	128.553	343.597	273.590	77.007	-106.08
		14. ETHANE						
38	194	C₂H₆ (g)	1200.0	133.804	355.012	279.904	90.131	-106.74

```
# Customize the figure to plot; just to make things pretty. Not mandatory.
```

```
from matplotlib.ticker import FormatStrFormatter
```

```
from matplotlib.pyplot import *
```

```
font = {'family' : 'DejaVu Sans',
        'weight' : 'normal',
        'size' : 15}
```

```
rcParams['axes.linewidth'] = 2
```

```
rcParams['lines.linewidth'] = 2
```

```
rcParams['lines.markersize'] = 8
```

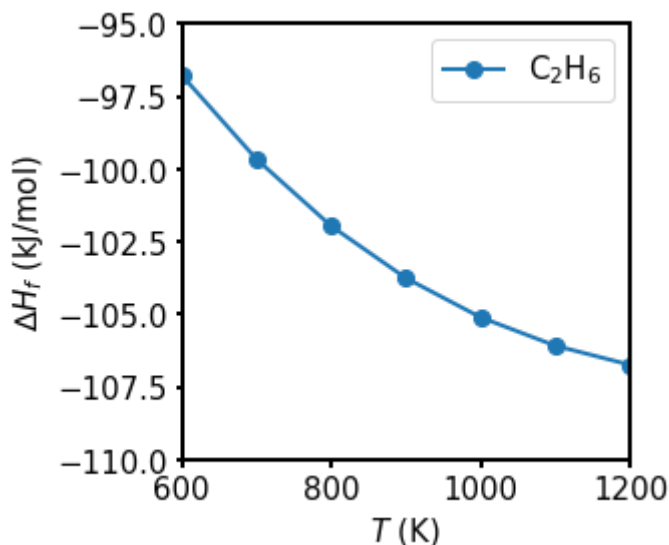
```
rcParams['xtick.major.width'] = 2
```

```
rcParams['ytick.major.width'] = 2
```

```
matplotlib.rc('font', **font)
```

```
plt.rcParams['figure.figsize'] = (4, 4)
```

```
# Plot ethane formation enthalpy from 600 to 1200 K. (I plotted all data and showed th
# Extract data from df; columns indexed by header
plt.plot(df_ethane['T (K)'], df_ethane['ΔHf° (kJ/mol)'], 'o-', label = 'C$$_2$H$_6$')
plt.xlim(600,1200)
plt.ylim(-110,-95)
plt.legend()
plt.xlabel('$T$ (K)')
plt.ylabel(r'$\Delta H_f$ (kJ/mol)')
plt.show()
```



```
# Just for fun, let's check the slope against tabulated heat capacities
print ('The slope is the heat capacity of reaction 2C(s,ref) + 3H2(g,ref) = C2H6(g).')
# For example, Cp(C2H6, 1000K) = 122.552 j/mol/k, Cp(C, 1000K) = 21.61 j/mol/k, Cp(H2,
Cp_reaction = 122.552-21.61*2-30.20*3
```

```
# Approximate the slope at 1000 K as (ΔHf°(1100K) - ΔHf°(900K))/200K
Cp_slope = (df_ethane[df_ethane['T (K)']==1100]['ΔHf° (kJ/mol)'].iloc[0]-df_ethane[df_
print ('Heat capacities at 1000 K read from thermodynamic table and calculated from th
```

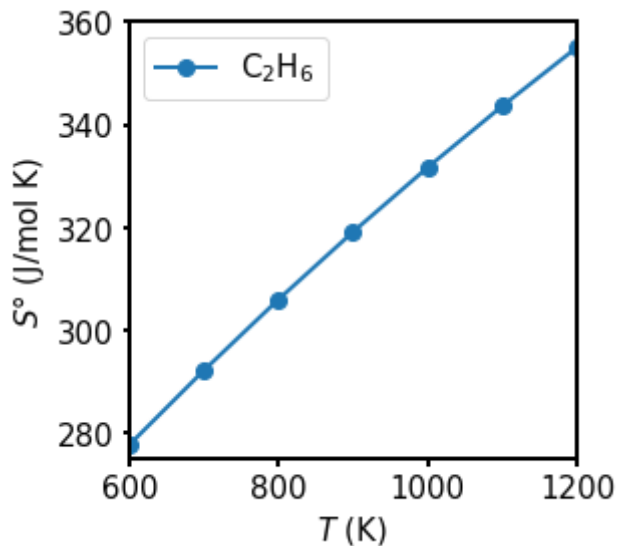
The slope is the heat capacity of reaction $2\text{C(s,ref)} + 3\text{H}_2(\text{g,ref}) = \text{C}_2\text{H}_6(\text{g})$.
Heat capacities at 1000 K read from thermodynamic table and calculated from the :

2. Plot the standard state entropy of ethane vs T from 600 to 1200K. What does the slope of the plot correspond to?

```
# Plot ethane entropy from 600 to 1200 K. (I plot all data and show the data from 600
plt.plot(df_ethane['T (K)'], df_ethane['S° (J/mol K)'], 'o-', label = 'C$$_2$H$_6$')
plt.xlim(600,1200)
plt.ylim(275,360)
plt.legend()
```

```
plt.xlabel('$T$ (K)')
plt.ylabel(r'$S^\circ$ (J/mol K)')
plt.show()

print ('The slope corresponds to Cv/T.')
```

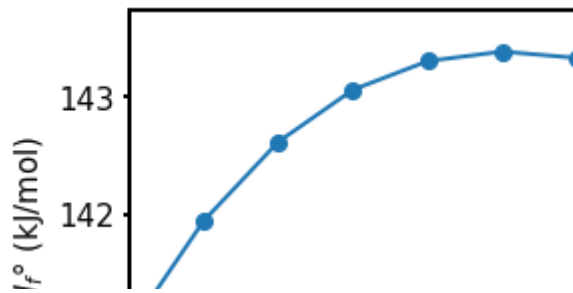


The slope corresponds to C_v/T .

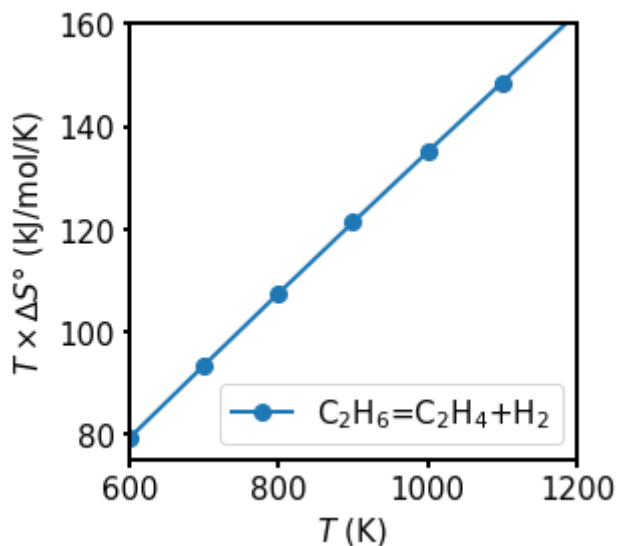
- Consider the ethane to ethylene reaction. Plot the standard state reaction enthalpy, T times the standard state reaction entropy, and the standard state reaction Gibbs energy vs T from 600 to 1200K.

```
# Read the C2H4 and H2 data
df_C2H4 = df[14:28]
df_H2 = df[42:56]

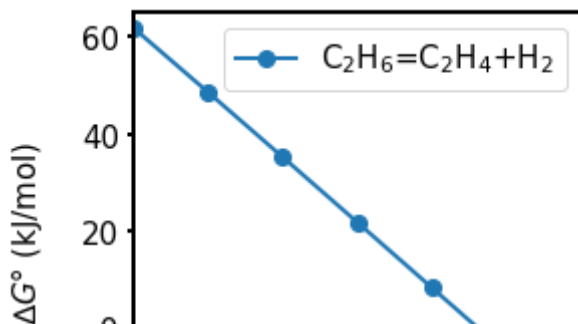
# Standard state reaction enthalpy vs T, C2H6 = C2H4 + H2
dH_C2H6_C2H4 = df_H2['ΔHf° (kJ/mol)'].values + df_C2H4['ΔHf° (kJ/mol)'].values - df_e
plt.plot(df_ethane['T (K)'], dH_C2H6_C2H4, 'o-', label = 'C$_2$H$_6$=C$_2$H$_4$+H$_2$')
plt.xlim(600,1200)
plt.ylim(140,)
plt.legend()
plt.xlabel('$T$ (K)')
plt.ylabel(r'$\Delta H_f^\circ$ (kJ/mol)')
plt.show()
```



```
# T time delta S vs T, C2H6 = C2H4 + H2
dS_C2H6_C2H4 = df_H2['S° (J/mol K)'].values + df_C2H4['S° (J/mol K)'].values - df_eth
plt.plot(df_ethane['T (K)'], dS_C2H6_C2H4*df_ethane['T (K)'].values/1000, 'o-', label
plt.xlim(600,1200)
plt.ylim(75,160)
plt.legend()
plt.xlabel('$T$ (K)')
plt.ylabel(r'$T \times \Delta S^\circ$ (kJ/mol/K)')
plt.show()
```



```
# delta G vs T, C2H6 = C2H4 + H2
dG_C2H6_C2H4 = dH_C2H6_C2H4 - df_ethane['T (K)'].values * dS_C2H6_C2H4 / 1000
plt.plot(df_ethane['T (K)'], dG_C2H6_C2H4, 'o-', label = 'C$$_2$H$$_6$=C$$_2$H$$_4$+H$$_2$')
plt.xlim(600,1200)
plt.ylim(-25,65)
plt.legend()
plt.xlabel('$T$ (K)')
plt.ylabel(r'$\Delta G^\circ$ (kJ/mol)')
plt.show()
```



4. What do the plots tell you about the conditions necessary to run these reaction thermally?
About what contributes to the free energy change with T ?

```
# Get T with delta G = 0 with linear interpolation
# np.interp only works for monotonically increasing sample points. dG_C2H6_C2H4 was de
T_G0 = np.interp(0,dG_C2H6_C2H4[:, :-1],df_ethane['T (K)'].values[:, :-1])
print ('Standard state delta G = 0 at {:.2f} K.'.format(T_G0))
print ('Plot tells us that increasing temperature favors products; ie, equilibrium exten
print ('A positive reaction entropy contributes to the free energy change with T.')
```

Standard state delta G = 0 at 1061.59 K.

Plot tells us that increasing temperature favors products; ie, equilibrium exten
A positive reaction entropy contributes to the free energy change with T.

5. Plot the free energy of the ethane to ethylene reaction vs ethane conversion at 800, 1000, and 1200K and 1 bar. Determine the equilibrium conversion of ethane at each temperature if this was the only reaction that occurs.

$$G(T, n_j) = \sum_j \mu_j(T) n_j$$

$$\mu_j(T) = \mu_j^o(T) + RT(\ln y_j + \ln P)$$

$$G(T, n_j) = \sum_j \mu_j^o(T) n_j + RT \sum_j n_j (\ln y_j + \ln P)$$

$$n_j = n_{j0} + \nu_j \xi; y_j = n_j / \sum n_j$$

$$G(T, \xi) = \sum_j \mu_j^o(T) n_{j0} + \xi \sum \nu_j \mu_j^o(T) + RT \sum_j N_j (\ln y_j + \ln P)$$

Normalize G by $n_{\text{C}_2\text{H}_6,0}$. Let $\xi' = \xi / n_{\text{C}_2\text{H}_6,0}$ be a dimensionless advancement.

$$G(T, \xi') / \text{molC}_2\text{H}_6 = \sum_j \mu_j^o(T) y_{j0} + \xi' \sum \nu_j \mu_j^o + RT \sum_j \frac{n_j}{n_{\text{C}_2\text{H}_6,0}} (\ln y_j + \ln P)$$

$$G(T, \xi') = \sum_j \mu_j^o(T) y_{j0} + \xi' \Delta G_r^o(T) + RT \sum_j \frac{n_j}{n_{\text{C}_2\text{H}_6,0}} (\ln y_j + \ln P)$$

$$G(T, \xi') = \sum_j \mu_j^o(T) y_{j0} + \xi' \Delta G_r^o(T) + RT \sum_j (y_{j0} + \nu_j \xi') (\ln y_j + \ln P)$$

Take $y_{\text{ethane},0} = 1$, $y_{\text{ethylene},0} = y_{\text{H}_2,0} = 0$
 $y_{\text{ethane},0} = 1, y_{\text{ethylene},0} = y_{\text{H}_2,0} = 0$

$$G(T, \xi') = \mu_{\text{C}_2\text{H}_6}^{\circ}(T) + \xi' \Delta G_r^{\circ}(T) + RT \sum_j (y_{j0} + \nu_j \xi') (\ln y_j + \ln P)$$

This G has a T dependent intercept that is the standard state free energy of ethane at the given T. That can be figured out from the Table above and is a decreasing function of T. For ease of visualization, one can subtract off this reference energy and normalize to RT to have all plots start at 0.

$$\bar{G} = (G - \mu_{\text{C}_2\text{H}_6}^{\circ})/RT = \xi' \Delta G_r^{\circ}/RT + \sum_j (y_{j0} + \nu_j \xi') (\ln y_j + \ln P)$$

This is what is plotted below.

```
# Reaction: C2H6 = C2H4 + H2
# First define the free energy (see the note) as a function of dimensionless advancement
# G_bar = (G(xi,T,P) - y_C2H6_0*mu_C2H6_0)/RT = ksi*deltaG/R/T + (y_H2_0 + nu_H2*ksi)
# G is the free energy of the system. ksi is the dimensionless advancement.
# deltaG is the standard free energy, y is the fraction, nu is the stoichiometric coefficient
def G_C2H6_C2H4_H2(ksi):
    global y_H2_0,y_C2H4_0,y_C2H6_0,nu_H2,nu_C2H4,nu_C2H6,P,deltaG,T,R
    y_H2 = (y_H2_0+ksi)/(1+ksi)
    y_C2H4 = (y_C2H4_0+ksi)/(1+ksi)
    y_C2H6 = (y_C2H6_0-ksi)/(1+ksi)
    G0xi = ksi*deltaG/R/T*1000
    Gxi_H2 = (y_H2_0 + nu_H2*ksi)*(np.log(y_H2) + np.log(P))
    Gxi_C2H4 = (y_C2H4_0 + nu_C2H4*ksi)*(np.log(y_C2H4) + np.log(P))
    Gxi_C2H6 = (y_C2H6_0 + nu_C2H6*ksi)*(np.log(y_C2H6) + np.log(P))
    G_bar = G0xi+Gxi_H2+Gxi_C2H4+Gxi_C2H6
    G = G_bar*R*T + df_ethane[df_ethane['T (K)']==T]['ΔGf° (kJ/mol)'].iloc[0]*1000
    return G_bar

# Define y, P and nu for this reaction
R = 8.314
y_H2_0 = 0
y_C2H4_0 = 0
y_C2H6_0 = 1
P = 1
nu_H2 = 1
nu_C2H4 = 1
nu_C2H6 = -1

# First define an array of dimensionless advancement ksi.
# N_C2H6 = N_C2H6_0 - N_C2H6_0*ksi. Conversion = (N_C2H6_0-N_C2H6)/N_C2H6_0 = ksi
ksi_array = np.arange(0.01,1,0.01)
conversion_array = ksi_array
m=000
```

```
T=800
```

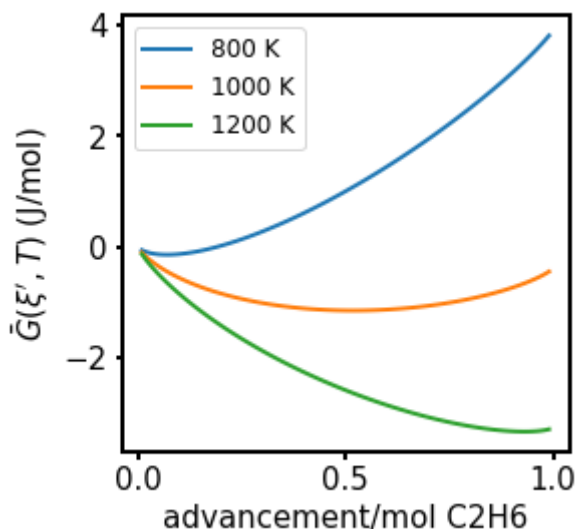
```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0] # Grab deltaG reaction from above
dG_reaction = G_C2H6_C2H4_H2(ksi_array) # compute advancement part
plt.plot(conversion_array, dG_reaction, '-', label = '800 K')
```

```
T=1000
```

```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
dG_reaction = G_C2H6_C2H4_H2(ksi_array)
plt.plot(conversion_array, dG_reaction, '-', label = '1000 K')
```

```
T=1200
```

```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
dG_reaction = G_C2H6_C2H4_H2(ksi_array)
plt.plot(conversion_array, dG_reaction, '-', label = '1200 K')
# plt.xlim(600,1200)
# plt.ylim(-25,65)
plt.legend(fontsize=12)
plt.xlabel('advancement/mol C2H6')
plt.ylabel(r'$\bar{G}(\xi, T)$ (J/mol)')
plt.show()
```



```
T=800
```

```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
res = minimize(G_C2H6_C2H4_H2,x0=[0.01],method='Nelder-Mead', tol=1e-6)
print ("Equilibrium conversion at 800 K is {:.2f}.".format(res.x[0]))
```

```
T=1000
```

```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
res = minimize(G_C2H6_C2H4_H2,x0=[0.01],method='Nelder-Mead', tol=1e-6)
print ("Equilibrium conversion at 1000 K is {:.2f}.".format(res.x[0]))
```

```
T=1200
```

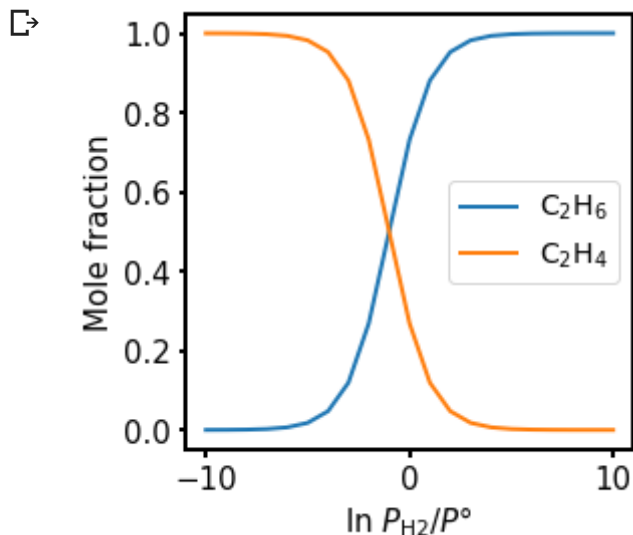
```
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
res = minimize(G_C2H6_C2H4_H2,x0=[0.9],method='Nelder-Mead', tol=1e-6)
print ("Equilibrium conversion at 1200 K is {:.2f}.".format(res.x[0]))
```


Equilibrium conversion at 800 K is 0.07.
 Equilibrium conversion at 1000 K is 0.52.
 Equilibrium conversion at 1200 K is 0.93.

6. Imagine a semi-permeable membrane through which you could control the partial pressure (chemical potential) of hydrogen in a reaction vessel. Plot the mole fractions of ethane and ethylene as a function of $\ln P_{H_2}$ at 1000K. (Assume the H_2 does not contribute to the total moles in the vessel.)

```
# In a reaction vessel, the total volume is constant. For this reaction, total pressure
#  $P_{C_2H_6} + P_{C_2H_4} = P_{C_2H_6,0}$ ,  $P_{C_2H_6}/P_{C_2H_6,0} + P_{C_2H_4}/P_{C_2H_6,0} = 1$ ,  $y_{C_2H_6} + y_{C_2H_4} =$ 
#  $K_{eq} = (P_{H_2}/P_0) \cdot (P_{C_2H_4}/P_0) / (P_{C_2H_6}/P_0) = (P_{H_2}/P_0) \cdot (P_{C_2H_4}/P_{C_2H_6,0}) / (P_{C_2H_6}/P_{C_2H_6,0})$ 
#  $(P_{H_2}/P_0) \cdot (1 - y_{C_2H_6}) / y_{C_2H_6} = K_{eq}$ ,  $y_{C_2H_6} = (P_{H_2}/P_0) / (P_{H_2}/P_0 + K_{eq})$ 
```

```
T=1000
deltaG = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]*1000 # kj/mol to j/mol
K_eq = np.exp(-deltaG/R/T)
ln_P_array = np.arange(-10,11,1)
P_H2_array = np.exp(ln_P_array)
y_C2H6_array = P_H2_array / (P_H2_array + K_eq)
plt.plot(ln_P_array, y_C2H6_array, '-', label = 'C$_2$H$_6$')
plt.plot(ln_P_array, 1-y_C2H6_array, '-', label = 'C$_2$H$_4$')
plt.legend(fontsize=14)
plt.xlabel(r'ln $P_{\rm H_2}$/$P^\circ$')
plt.ylabel(r'Mole fraction')
plt.show()
```



7. Now consider both reactions simultaneously. Determine the equilibrium ratio of ethylene to acetylene, starting from pure ethane, at 1000 K and 1, 10, and 100 bar total pressure (again, assume ideal behavior at all conditions). Can you explain the result?

$$G(J) = \sum_j \mu_j N_j$$

$$\mu_j = \mu_j^o + RT(\ln y_j + \ln P)$$

$$G(J) = \sum_j \mu_j^o N_j + RT \sum_j N_j (\ln y_j + \ln P)$$

$$N_j = N_{j0} + \nu_j \xi$$

$$G(J) = \sum_j \mu_j^o N_{j0} + \xi \sum \nu_j \mu_j^o + RT \sum_j N_j (\ln y_j + \ln P)$$

Normalize G with $N_{C_2H_6,0}$.

$$G(J/\text{molC}_2\text{H}_6) = \sum_j \mu_j^o y_{j0} + \xi' \sum \nu_j \mu_j^o + RT \sum_j y_j (\ln y_j + \ln P)$$

$$G(J/\text{molC}_2\text{H}_6) = \sum_j \mu_j^o y_{j0} + \xi' \Delta G_r^o + RT \sum_j y_j (\ln y_j + \ln P)$$

ξ' is the dimensionless advancement. $y_j = y_{j0} + \nu_j \xi'$.

$$G(J/\text{molC}_2\text{H}_6) = \sum_j \mu_j^o y_{j0} + \xi' \Delta G_r^o + RT \sum_j (y_{j0} + \nu_j \xi') (\ln(y_{j0} + \nu_j \xi') + \ln P)$$

For this problem:

$$G(J/\text{molC}_2\text{H}_6) = \mu_{C_2H_6}^o + \xi' \Delta G_r^o + RT \sum_j (y_{j0} + \nu_j \xi') (\ln(y_{j0} + \nu_j \xi') + \ln P)$$

```
# Reaction: C2H6 = C2H4 + H2 (ksi_1), C2H6 = C2H2 + 2H2 (ksi_2)
# First define the free energy as a function of dimensionless advancement.
# G_bar = (G(xi,T,P) - y_C2H6_0*mu_C2H6_0)/RT = ksi_1*deltaG_1/R/T + ksi_2*deltaG_2/R,

def G_C2H6_C2H4_C2H2_H2(ksi):
    global y_H2_0,y_C2H4_0,y_C2H2_0,y_C2H6_0,P,deltaG_1,deltaG_2,T,R
    y_H2 = (y_H2_0+ksi[0]+2*ksi[1])/(1+ksi[0]+2*ksi[1])
    y_C2H4 = (y_C2H4_0+ksi[0])/(1+ksi[0]+2*ksi[1])
    y_C2H2 = (y_C2H2_0+ksi[1])/(1+ksi[0]+2*ksi[1])
    y_C2H6 = (y_C2H6_0-ksi[0]-ksi[1])/(1+ksi[0]+2*ksi[1])
    term1 = ksi[0]*deltaG_1/R/T*1000 + ksi[1]*deltaG_2/R/T*1000
    term2 = (y_H2_0+ksi[0]+2*ksi[1])*(np.log(y_H2) + np.log(P))
    term3 = (y_C2H4_0 + ksi[0])*(np.log(y_C2H4) + np.log(P))
    term4 = (y_C2H2_0 + ksi[1])*(np.log(y_C2H2) + np.log(P))
    term5 = (y_C2H6_0 - ksi[0] - ksi[1])*(np.log(y_C2H6) + np.log(P))
    G_bar = term1+term2+term3+term4+term5
    G = G_bar*R*T + df_ethane[df_ethane['T (K)']==T]['ΔGf° (kJ/mol)'].iloc[0]*1000
    return G

df_C2H2 = df[0:14]
dH_C2H6_C2H2 = 2*df_H2['ΔHf° (kJ/mol)'].values + df_C2H2['ΔHf° (kJ/mol)'].values - df_
dS_C2H6_C2H2 = 2*df_H2['S° (J/mol K)'].values + df_C2H2['S° (J/mol K)'].values - df_e
dG_C2H6_C2H2 = dH_C2H6_C2H2 - df_ethane['T (K)'].values * dS_C2H6_C2H2 / 1000

# Define y, P and nu for this reaction
```

```

R = 8.314
y_H2_0 = 0
y_C2H4_0 = 0
y_C2H2_0 = 0
y_C2H6_0 = 1

T=1000
deltaG_1 = dG_C2H6_C2H4[df_ethane['T (K)'] == T][0]
deltaG_2 = dG_C2H6_C2H2[df_ethane['T (K)'] == T][0]
P = 1
res = minimize(G_C2H6_C2H4_C2H2_H2,x0=[0.1,0.001],method='Nelder-Mead', tol=1e-16)
print ("Equilibrium ratio of C2H4 to C2H2 at 1 bar is {:.2f}.".format(res.x[0]/res.x[1]))

P = 10
res = minimize(G_C2H6_C2H4_C2H2_H2,x0=[0.1,0.0001],method='Nelder-Mead', tol=1e-16)
print ("Equilibrium ratio of C2H4 to C2H2 at 10 bar is {:.2f}.".format(res.x[0]/res.x[1]))

P = 100
res = minimize(G_C2H6_C2H4_C2H2_H2,x0=[0.1,0.00001],method='Nelder-Mead', tol=1e-16)
print ("Equilibrium ratio of C2H4 to C2H2 at 100 bar is {:.2f}.".format(res.x[0]/res.x[1]))

print ("High pressure inhibits the formation of C2H2 more than C2H4 because of more H2 in C2H6")

Equilibrium ratio of C2H4 to C2H2 at 1 bar is 163.84.
Equilibrium ratio of C2H4 to C2H2 at 10 bar is 757.95.
Equilibrium ratio of C2H4 to C2H2 at 100 bar is 2727.35.
High pressure inhibits the formation of C2H2 more than C2H4 because of more H2 in C2H6

```

Alternative approach

	C_2H_6	C_2H_4	H_2	C_2H_2
Initial	1	0	0	0
Change	$-\epsilon_1$ $-\epsilon_2$	ϵ_1	ϵ_1 $+2\epsilon_2$	ϵ_2
End	1 $-\epsilon_1$ $-\epsilon_2$	ϵ_1	ϵ_1 $+2\epsilon_2$	ϵ_2

$$total = 1 + \epsilon_1 + 2\epsilon_2$$

Two equations:

$$K_{eq1} = \frac{y_{C_2H_4} y_{H_2}}{y_{C_2H_6}} \left(\frac{P}{P^o} \right) = \frac{\frac{\epsilon_1}{1+\epsilon_1+2\epsilon_2} \frac{\epsilon_1+2\epsilon_2}{1+\epsilon_1+2\epsilon_2}}{\frac{1-\epsilon_1-\epsilon_2}{1+\epsilon_1+2\epsilon_2}} \left(\frac{P}{P^o} \right) = \frac{\epsilon_1(\epsilon_1+2\epsilon_2)}{(1+\epsilon_1+2\epsilon_2)(1-\epsilon_1-\epsilon_2)} \left(\frac{P}{P^o} \right)$$

$$K_{eq2} = \frac{y_{C_2H_2} y_{H_2}^2}{y_{C_2H_6}} \left(\frac{P}{P^o} \right)^2 = \frac{\epsilon_2(\epsilon_1+2\epsilon_2)^2}{(1+\epsilon_1+2\epsilon_2)^2(1-\epsilon_1-\epsilon_2)} \left(\frac{P}{P^o} \right)^2$$

solve for ϵ_1 and ϵ_2

```

# Calculate  $\Delta G^\circ$  for the first reaction

ethane_G = df[' $\Delta G^\circ$  (kJ/mol)'][32:39]
hydrogen_G = df[' $\Delta G^\circ$  (kJ/mol)'][46:53]
ethylene_G = df[' $\Delta G^\circ$  (kJ/mol)'][18:25]

dG_r1 = hydrogen_G.values + ethylene_G.values - ethane_G.values
K_r1_1000K = np.exp(-dG_r1[4]/1000/0.008314)

# Calculate  $\Delta G^\circ$  for the second reaction
ethane_G = df[' $\Delta G^\circ$  (kJ/mol)'][32:39]
hydrogen_G = df[' $\Delta G^\circ$  (kJ/mol)'][46:53]
acetylene_G = df[' $\Delta G^\circ$  (kJ/mol)'][18:25]

dG_r2_1000K = 2*df[' $\Delta G^\circ$  (kJ/mol)'][50] + df[' $\Delta G^\circ$  (kJ/mol)'][8] - df[' $\Delta G^\circ$  (kJ/mol)']

#Calculate Keq for second reaction
K_r2_1000K = np.exp(-dG_r2_1000K/1000/0.008314)

#solve for extension of reaction
from scipy.optimize import fsolve
# print(c1,c2)
def equations(p):
    c1 = K_r1_1000K/P
    c2 = K_r2_1000K/P**2
    e1,e2 = p
    return(e1*(e1+2*e2)-c1*(1+e1+2*e2)*(1-e1-e2),e2*(e1+2*e2)**2-c2*(1+e1+2*e2)**2*(1-e1)
P = 1 #bar
e1,e2 = fsolve(equations, (0.5,0.5))
print('Equilibirum ratio of ethane to acethlyene at 1 bar is ',e1/e2)
print (e1,e2)
P = 10
e1,e2 = fsolve(equations, (0.5,0.5))
print('Equilibirum ratio of ethane to acethlyene at 10 bar is ',e1/e2)
print (e1,e2)

P = 100
e1,e2 = fsolve(equations, (0.5,0.5))
print('Equilibirum ratio of ethane to acethlyene at 100 bar is ',e1/e2)
print (e1,e2)

print('As pressure increase, the conversion of second reaction decrease. \nThis is bec

Equilibirum ratio of ethane to acethlyene at 1 bar is 163.83389678878956
0.5157785342538395 0.003148179615838399
Equilibirum ratio of ethane to acethlyene at 10 bar is 757.9113580419111
0.1881064466065581 0.00024819056293408413
Equilibirum ratio of ethane to acethlyene at 100 bar is 2727.188895799215
0.06050937334501712 2.2187452229004684e-05

```

As pressure increase, the conversion of second reaction decrease.

This is because the second reaction can produce more gas than the first reaction

▼ 2 Langmuir adsorption

2.1 The table below shows the mass of carbon monoxide taken up by a 10 wt% Ru on alumina sample (i.e., particles of Ru on an inert support) as a function of CO pressure at 100°C. CO is known to only adsorb on Ru at these conditions.

P_{CO} (torr)	100	150	200	250	300	400
CO adsorbed ($\mu\text{mol/g}$ sample)	1.28	1.63	1.77	1.94	2.06	2.21

1. Plot the data and use non-linear regression to estimate the value of the uptake equilibrium constant.

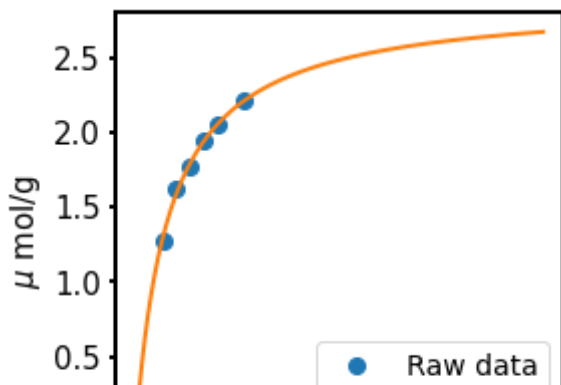
```
P_CO = np.array([100,150,200,250,300,400]) * 0.00133322 # From torr to bar
CO_ads = np.array([1.28,1.63,1.77,1.94,2.06,2.21])
```

```
# Function to fit: theta = K_ads*P/(1+K_ads*P), theta = CO_ads / CO_ads_max
def Langmuir(P,CO_ads_max, K_ads):
    return K_ads*P/(1+K_ads*P)*CO_ads_max
```

```
initial_guesses = 3,1 # For CO_ads_max and K_ads
popt, pcov = curve_fit(Langmuir, P_CO, CO_ads, initial_guesses)
print ('The uptake equilibrium constant is {:.2f}'.format(popt[1]))
```

The uptake equilibrium constant is 6.12.

```
plt.plot(P_CO,CO_ads, 'o', label = 'Raw data')
# Extrapolate the curve to low P to check if it passes (0,0) and high P to check the r
plt.plot(np.arange(0.001,2,0.001), Langmuir(np.arange(0.001,2,0.001), *popt), '-',label=
plt.legend(fontsize=14)
plt.xlabel(r'$P_{\rm CO}$ (bar)')
plt.ylabel(r'$\mu$ mol/g')
plt.show()
```

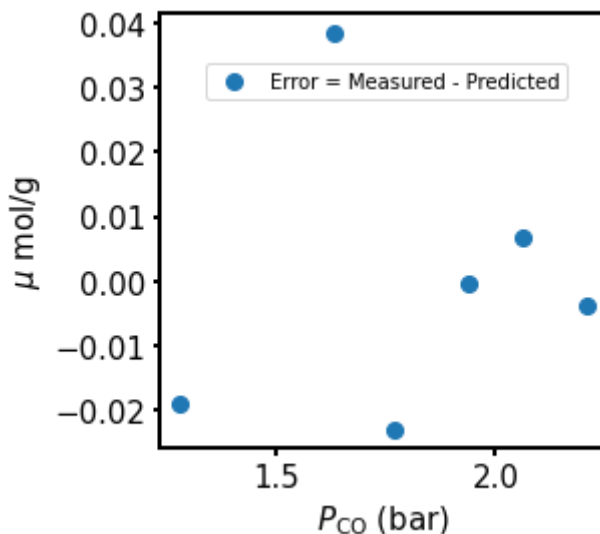


2. Plot the model error vs pressure. Do you have confidence in the model?

```

perr = np.sqrt(np.diag(pcov))
residuals = CO_ads - Langmuir(P_CO,*popt)
plt.plot(CO_ads, residuals,'o', label = 'Error = Measured - Predicted')
plt.legend(fontsize=10,loc=(0.1,0.8))
plt.xlabel(r'$P_{\rm CO}$ (bar)')
plt.ylabel(r'$\mu$ mol/g')
plt.show()
print ('The model is probably OK within the pressure range of the measurements.')

```



The model is probably OK within the pressure range of the measurements.

3. Use the data to estimate the fraction of exposed Ru atoms.

```

print ('The max uptake is {:.2f} micro mole/g.'.format(popt[0]))
Ru_weight_percentage = 0.1
Ru_atomic_weight = 101.07 # g/mol
Ru_total = 1*Ru_weight_percentage/Ru_atomic_weight*10**6 # Total Ru micro mole per gram
print ('The total Ru atom/(g catalyst) is {:.2f} micro mole.'.format(Ru_total))
fraction = popt[0]/Ru_total
print ('Exposed fraction is {:.3f}'.format(fraction))

```

The max uptake is 2.89 micro mole/g.
The total Ru atom/(g catalyst) is 989.41 micro mole.
Exposed fraction is 0.003.

✓ 0s completed at 7:08 PM

