Donovan Feist
CH EN 3553
HW 6

1.a) mole balance: design equation for a PFR

is $V = F_{A0}\int \frac{dX_A}{-r_A} \Rightarrow \frac{dX_A}{dV} = \frac{-r_A}{F_{A0}}$

$-r_A = kC_B^2 \Rightarrow \frac{dX_A}{dV} = \frac{kC_B^2}{F_{A0}}$

k follows Arrhenius equation: $k(T) = k(T_{ref})\exp\left(\frac{E_a}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right)$

$F_{A0} = F_0 y_{A0} = (5\ \frac{mol}{s})(0.2) = 1\ \frac{mol}{s}$; $\frac{1\ mol/s}{\nu_A} = \frac{1\ mol/s}{1} = 1\ \frac{mol}{s}$

$F_{B0} = F_0 y_{B0} = (5\ \frac{mol}{s})(0.6) = 3\ \frac{mol}{s}$; $\frac{3\ mol/s}{\nu_B} = \frac{3\ mol/s}{2} = 1.5\ \frac{mol}{s} \Rightarrow$

A is the limiting reactant

$C_B = \frac{C_{A0}(\Theta_B - \frac{\nu_B}{\nu_A}X_A)}{1+\epsilon X_A}\left(\frac{T_0}{T}\right)\cancel{\left(\frac{P}{P_0}\right)}^{1}$; $\delta = \frac{1}{1} - \frac{2}{1} - 1 = -2$; $\epsilon = \delta y_{A0} = (-2)(0.2) = -0.4$

$\Theta_B = \frac{F_{B0}}{F_{A0}} = \frac{3}{1} = 3 \Rightarrow C_B = \frac{C_{A0}(3-2X_A)}{1-0.4X_A}\left(\frac{T_0}{T}\right) \Rightarrow \frac{dX_A}{dV} = \frac{kC_B^2}{F_{A0}}$,

where equations for k + C_B + F_{A0} are now known.

energy balance: $\cancel{\frac{dE}{dt}}^{0} = \cancel{\dot{Q}}^{0} - \cancel{\dot{W}_s}^{0} - F_{A0}\sum\Theta_i C_{pi}(T-T_0) - F_{A0}X_A(\Delta H_R^\circ(T_R) + \cancel{\Delta C_p}^{0}(T-T_R)) \Rightarrow$

$0 = -F_{A0}(T-T_0)\sum\Theta_i C_{pi} - F_{A0}X_A\Delta H_R^\circ(T_R) \Rightarrow 0 = (T-T_0)\sum\Theta_i C_{pi} + X_A\Delta H_R^\circ(T_R)$

$\Rightarrow T = T_0 - \frac{X_A\Delta H_R^\circ(T_R)}{\sum\Theta_i C_{pi}}$; Using python (code included below) $\Rightarrow$

$\boxed{X_{out} = 0.0032 \text{ and } T_{out} = 500.03}$

b) rearranging the energy balance $\Rightarrow T_0 = T + \frac{X_A\Delta H_R^\circ(T_R)}{\sum\Theta_i C_{pi}}$.

Using $T = T_{max} = 600\ K \Rightarrow \boxed{T_0 = 591.13\ K}$

using the mole + energy balance derived in part a, along with the equations for k + C_B, the "odeint" function in python can be used to find the reactor volume. $\boxed{V = 69612\ L}$

2.a) feed is equimolar in A + B, $\nu_A = \nu_B = 1 \Rightarrow$ A is limiting reactant

C_p is independent of T

energy balance: $\cancel{\frac{dE}{dt}}^{0} = \dot{Q} - \cancel{\dot{W}_s}^{0} - F_{A0}\sum\Theta_i C_{pi}(T - T_0) - F_{A0}X_A(\Delta H_R^\circ(T_R) + \cancel{\Delta \hat{C}_p}^{0}(T - T_R))$

$\Rightarrow 0 = \dot{Q} - F_{A0}\sum\Theta_i C_{pi}\cancel{(T - T_0)}^{0\ \text{isothermal}} - F_{A0}X_A\Delta H_R^\circ(T_R) \Rightarrow \dot{Q} = F_{A0}X_A\Delta H_R^\circ$

$\Delta H_R^\circ(T_R)$ is found by using the adiabatic data.

For the adiabatic case, the energy balance becomes

$0 = -F_{A0}\sum\Theta_i C_{pi}(T - T_0) - F_{A0}X_A\Delta H_R^\circ(T_R) \Rightarrow \Delta H_R^\circ(T_R) = \frac{-\sum\Theta_i C_{pi}(T - T_0)}{X_A}$

Θ_i is calculated with $\Theta_i = \frac{F_{i0}}{F_{A0}} \Rightarrow \Delta H_R^\circ(T_R) = -7500.0\ \frac{kJ}{mol}$

$\Delta H_R^\circ(T_R)$ is then plugged into the isothermal energy balance, the solved for $\dot{Q} \Rightarrow \boxed{\dot{Q} = -750{,}000\ \frac{kJ}{min}}$

b) $F_{A1} = F_{A0}(1 - X_{A iso})$; mole balance: design equation for CSTR: $V = \frac{F_{A1}X_{A2}}{-r_A}$; $-r_A = kC_AC_B$; $C_A = C_{A1}(1 - X_{A2})$;

$C_B = C_{B1}(1 - X_{A2})$; $k(T) = k(T_r)\exp\left(\frac{E_a}{R}\left(\frac{1}{T_r} - \frac{1}{T}\right)\right)$

$K(T_r)$ is found by using the conditions from the first reactor: $V = \frac{F_{A0} X_{iso}}{K_{300} C_A C_B} \Rightarrow K_{300} = 0.15625 \frac{L}{mol \cdot min}$

energy balance for adiabatic conditions:

$$\cancel{\frac{dE}{dt}}^{0} = \dot{Q} - \cancel{\dot{W}_s}^{0} - F_{A0} \Sigma \Theta_i C_{pi} (T - T_0) - F_{A0} X_A (\Delta H_R^\circ(T_R) + \cancel{\Delta C_p}^{0} (T - T_R)),$$

$$\dot{Q} = UA(T_a - T) \Rightarrow X_{A_{energy}} = \frac{UA(T_a - T) - F_{A0} \Sigma \Theta_i C_{pi} (T - T_0)}{F_{A0} \Delta H_R^\circ(T_R)}$$

solving mole balance for: $X_{A_{mole}} = \frac{-V r_A}{F_{A1}}$

K_{350} is found by using the adiabatic conditions from the first reactor: $K_{350} = \frac{F_{A0} X_{adi}}{V C_A C_B} \Rightarrow K_{350} = 0.556 \frac{L}{mol \cdot min}$

K_{300} + K_{350} are used to find E_a with $E_a = \ln\left(\frac{K(T)}{K(T_r)}\right) \cdot R \cdot \left(\frac{1}{T_r} - \frac{1}{T}\right)^{-1} \Rightarrow$

$E_a = 12102 \frac{J}{mol}$; the conversion of reactor 2 is found by changing T until $X_{A_{energy}} - X_{A_{mole}} = 0$ using Python (code is included) $\Rightarrow \boxed{X_{overall} = 0.453}$

c) $UA = 10 \frac{kJ}{m^3 \cdot min \cdot K}$ $\quad V = 1 m^3$ $\quad T_a = 300K$

mole balance: $\frac{dX}{dV} = \frac{-r_A}{F_{A1}} = \frac{K C_{A1}^2 (1-X)^2}{F_{A1}}$

$K = K_r \exp\left(\frac{E_a}{R}\left(\frac{1}{T_r} - \frac{1}{T}\right)\right)$

energy balance: $\frac{dT}{dV} = \frac{r_A \Delta H_R - UA(T - T_a)}{F_{A1} \cdot \Sigma \Theta_i C_{pi}}$

"odeint" in Python was used to integrate over the volume (code included below) $\Rightarrow$

$$\boxed{X_{overall} = 0.416}$$

d) $r_A = k_{-1} C_C - k_1 C_A C_B = \frac{k_1}{K_C} C_C - k_1 C_A C_B = k_1\left(\frac{C_C}{K_C} - C_A C_B\right)$

mole balance : $\frac{dX}{dV} = \frac{-r_A}{F_{A1}}$

energy balance : $\frac{dT}{dV} = \frac{r_A \Delta H_R - UA(T - T_a)}{F_{A1} \cdot \sum \Theta_i C_{pi}}$

$K_c = K_{c,T_r} \exp\left(\frac{\Delta H_R}{R}\left(\frac{1}{T_r} - \frac{1}{T}\right)\right)$

$k = k_{ref} \exp\left(\frac{E_a}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right)$

using "odeint" in Python (code included below) $\Rightarrow$

$$\boxed{X_{overall} = 0.283}$$

3. $\Delta H_R^\circ(T_R) = \sum \nu_i H_i^\circ \Rightarrow \Delta H_{R,273}^\circ = -5 \frac{Kcal}{mol}$

mole balance: CSTR design equation: $V = \frac{F_{A0}X_A}{-r_A}$

$-r_A = K C_A C_B$; $C_A = C_{A0}(1-X_A)$; $C_B = C_{B0}(1-X_A)$

$K = K_r \exp\left(\frac{E_a}{R}\left(\frac{1}{T_r} - \frac{1}{T}\right)\right)$

energy balance: $\cancel{\frac{dE}{dt}}^{0} = \cancel{\dot{Q}}^{0} - \cancel{\dot{W}_s}^{0} - F_{A0}\sum \Theta_i C_{pi}(T-T_0) - F_{A0}X_A(\Delta H_R + \cancel{\Delta \hat{C}_p}^{0}(T-T_R)) \Rightarrow$

$\sum \Theta_i C_{pi}(T-T_0) + X_A \Delta H_R^\circ = 0 \Rightarrow T = T_0 - \frac{X_A \Delta H_R^\circ}{\sum \Theta_i C_{pi}}$

$\Delta H_R(T) = \Delta H_R^\circ(T_r) + \Delta C_{pi}(T-T_r)$, $\Delta C_{pi} = \sum \nu_i C_{pi}$

the equations above were used in a function in Python, with inputs of X + T. The "fsolve" functionality was then used to find the roots of the defined function (code included below) $\Rightarrow$

$$\boxed{\begin{array}{l} X = 0.8836 \\ T = 500.81 \text{ K} \end{array}}$$

4. mole balance: CSTR design equation: $V = \frac{F_{A0} X_A}{-r_A}$

$-r_A = k$; $\tau = \frac{V}{\dot{V}} = \frac{F_{A0} X_A / -r_A}{\dot{V}} = \frac{C_{A0} \dot{V} X_A}{\dot{V} k} = \frac{C_{A0} X_A}{k}$

energy balance: $\cancel{\frac{dE}{dt}}^{0} = \cancel{\dot{Q}}^{0} - \cancel{\dot{W}_s}^{0} - F_{A0} \sum \Theta_i C_{pi} (T - T_0) - F_{A0} X_A (\Delta H_R^\circ + \cancel{\Delta C_p}^{0} (T - T_R))$

$\Rightarrow 0 = -F_{A0} \sum \Theta_i C_{pi} (T - T_0) - F_{A0} X_A \Delta H_R^\circ \Rightarrow 0 = \sum \Theta_i C_{pi} (T - T_0) + X_A \Delta H_R^\circ$

$\Rightarrow X_A = \frac{-(T - T_0) \sum \Theta_i C_{pi}}{\Delta H_R^\circ}$; $\sum \Theta_i C_{pi} = C_{pA}$; $k = k_{ref} \cdot \exp\left(\frac{E}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right)$

$$\Rightarrow \boxed{\tau = \frac{-C_{A0} (T - T_0) C_{p,A}}{\Delta H_R^\circ k_{ref} \cdot \exp\left(\frac{E}{R}\left(\frac{1}{T_{ref}} - \frac{1}{T}\right)\right)}}$$

Donovan Feist

CH EN 3553

Homework 6

Problem 1

Part a

In [1]:

```
#imports
import numpy as np
from scipy.integrate import odeint
from scipy.optimize import fsolve

#given values
V=150 #L
k_ref=0.0055 #L/(mol*s)
T_ref=300 #K
Ea=500 #J/mol
T_0=500 #K
R=8.314 #J/(mol*K)
c_pA=150 #J/(mol*K)
c_pB=150 #J/(mol*K)
c_pI=150 #J/(mol*K)
ΔH_R_circle=-7000 #J/mol
F_0=5 #mol/s
V̇_0=50 #L/s
y_A0=0.2 #unitless
y_B0=0.6 #unitless
y_I0=0.2 #unitless

#calculated values
F_A0=F_0*y_A0 #mol/s
c_A0=F_A0/V̇_0 #mol/L
Θ_A=1 #unitless
Θ_B=3 #unitless
Θ_I=1 #unitless

#creating differential function to integreate over
def func(X,V):
    T=T_0-X*ΔH_R_circle/(Θ_A*c_pA+Θ_B*c_pB+Θ_I*c_pI) #K
    k=k_ref*np.e**(Ea/R*(1/T_ref-1/T)) #L/(mol*s)
    c_B=c_A0*(3-2*X)/(1-0.4*X)*(T_0/T) #mol/L
    dXdV=k*c_B**2/F_A0 #1/L
    return dXdV

#initial condition for X
initial=[0] #unitless

#Volume array
V=np.linspace(0,V,1000) #L

#solution array for conversion
X=odeint(func,initial,V) #unitless
print('Outlet conversion =',X[-1],'.')

#plugging X back into equation for T to find final T
T=T_0-X[-1]*ΔH_R_circle/(Θ_A*c_pA+Θ_B*c_pB+Θ_I*c_pI) #K
print('Outlet temperature =',T,'K.')
```

```
Outlet conversion = [0.00321502] .
Outlet temperature = [500.03000684] K.
```

Part b

In [2]:
```
T=600 #K
Xfinal=0.95 #conversion

T_0=T+Xfinal*ΔH_R_circle/(Θ_A*c_pA+Θ_B*c_pB+Θ_I*c_pI) #K
print('T_0=',T_0,'K.')
```

```
T_0= 591.1333333333333 K.
```

In [3]:
```
#creating differential function to integreate over
def func(V,X):
    c_B=c_A0*(3-2*X)/(1-0.4*X)*(T_0/T) #mol/L
    k=k_ref*np.exp(Ea/R*(1/T_ref-1/T)) #L/(mol*s)
    dVdX=[F_A0/(k*c_B**2)] #L
    return dVdX

#initial condition for V
V0=[0] #L

#conversion array
X=np.linspace(0,Xfinal) #unitless

#solution array for volume
sol=odeint(func,V0,X) #L
print('Reactor volume =',sol[-1],'L.')
```

```
Reactor volume = [69611.5909745] L.
```

Problem 2

Part a

In [4]:
```
#isothermal data
V1=1 #m^3
V̇_0=0.5 #m^3/min
Tiso=300 #K
Xiso=0.20
c_A0=1 #mol/L
c_B0=1 #mol/L

#adiabatic data
Tadi_out=350 #K
Xadi=0.40

#other data
c_pA=25 #kJ/(mol*K)
c_pB=35 #kJ/(mol*K)
c_pC=60 #kJ/(mol*K)
UA=4.0 #kJ/(min*K)
Ta=350 #K
R=8.314 #J/(mol*K)

#calculated values
Θ_A=1
Θ_B=1
Θ_C=0
F_A0=V̇_0*c_A0*1000 #mol/min (1000 converts m^3 into L)

#calculating ΔH_R(T_R)
ΔH_R=-(Tadi_out-Tiso)*(Θ_A*c_pA+Θ_B*c_pB)/Xadi #kJ/mol
print('ΔH_R=',ΔH_R,'kJ/mol')

#calculating Q̇
Q̇=F_A0*Xiso*ΔH_R
print('Q̇=',Q̇,'kJ/min')
```

```
ΔH_R= -7500.0 kJ/mol
Q̇= -750000.0 kJ/min
```

Part b

In [5]:

```
#calculated values
F_A1=F_A0*(1-Xiso) #mol/min
F_B0=V̇_0*c_B0*1000 #mol/min (1000 converts m^3 into L)
F_B1=F_B0*(1-Xiso)
F_C1=F_A0*Xiso
Θ_A=1
Θ_B=F_B1/F_A1
Θ_C=F_C1/F_A1
F_A0=V̇_0*c_A0*1000 #mol/min (1000 converts m^3 into L)
c_A1=F_A1/(V̇_0*1000) #mol/L (1000 converts m^3 into L)
c_B1=F_B1/(V̇_0*1000) #mol/L (1000 converts m^3 into L)
sigΘcp=Θ_A*c_pA+Θ_B*c_pB+Θ_C*c_pC #kJ/mol
T0=300 #K

#calculating k_300
c_A=c_A0*(1-Xiso)
c_B=c_B0*(1-Xiso)
k_300=F_A0*Xiso/(V1*c_A*c_B*1000) #L/(mol*min) (1000 converts m^3 into
 L)
print('k_300=',k_300,'L/(mol*min).')

#calculating k_350
c_A=c_A0*(1-Xadi)
c_B=c_B0*(1-Xadi)
k_350=F_A0*Xadi/(V1*c_A*c_B*1000) #L/(mol*min) (1000 converts m^3 into
 L)
print('k_350=',k_350,'L/(mol*min).')

#calculating Ea
Ea=np.log(k_350/k_300)*R/(1/300-1/350) #J/mol
print('Ea=',Ea,'J/mol')

X0=0.2
def fun(A):
    T,X=A
    k=k_300*np.e**(Ea/R*(1/300-1/T)) #L/(mol*min)
    Ebal=UA*(Ta-T)-F_A1*sigΘcp-F_A1*X*ΔH_R*(T-T0)
    c_A=c_A1*(1-X)
    c_B=c_B1*(1-X)
    Mbal=V1*1000-F_A1*(X-X0)/(k*c_A*c_B)
    return (Ebal,Mbal)

sol=fsolve(fun,(300,0.2))
print('Conversion out of 2nd CSTR =',sol[1],'.')

#calculating total conversion
Xoverall=(F_A0-F_A1*(1-sol[1]))/F_A0
print('Overall Conversion =',Xoverall,'.')
```

```
k_300= 0.15624999999999997 L/(mol*min).
k_350= 0.5555555555555556 L/(mol*min).
Ea= 22147.44663579755 J/mol
Conversion out of 2nd CSTR = 0.3167993855037119 .
Overall Conversion = 0.45343950840296954 .
```

Part c

In [6]:
```
Ta=300 #K
UAgiven=10 #kJ/(m^3*min*K)
UA=UAgiven/1000 #kJ/*L*min*K (1000 converts m^3 to L)

def func(x,V):
    k=k_300*np.exp((Ea/R)*(1/300-1/x[1])) #L/(mol*min)
    dXdV=k*c_B1**2*((1-x[0])**2)/F_A1 #1/L
    dTdV=(-k*c_A1*c_B1*(1-x[0])**2*ΔH_R-UA*(x[1]-Ta))/(F_A1*sigΘcp) #K/L
    return [dXdV,dTdV]

#initial conditions for x
x=[0,300] #[conversion,K]

#Volume array
V=np.linspace(0,V1*1000,1000) #L

#solution array for conversion
sol=odeint(func,x,V)
print('Outlet conversion of PFR =',sol[-1,0],'.')

#calculating overall conversion
Xoverall=(F_A0-F_A1*(1-sol[-1,0]))/F_A0
print('Overall conversion=',Xoverall,'.')
```

```
Outlet conversion of PFR = 0.2703733410290806 .
Overall conversion= 0.4162986728232645 .
```

Part d

In [7]:

```
Tr=310 #K
Kc_Tr=2 #dm^3/mol = L/mol
c_C1=F_C1/(V̇_0*1000)

def func(x,V):
    X=x[0]
    T=x[1]
    Kc=Kc_Tr*np.exp((ΔH_R/(R/1000))*(1/Tr-1/T)) #L/mol
    k=k_300*np.exp((Ea/R)*(1/300-1/T)) #L/(mol*min)
    c_B=c_B1*(1-X)
    c_C=c_C1*X
    r_A=-k*(c_B**2-c_C/Kc)
    dXdV=-r_A/F_A1
    dTdV=(r_A*ΔH_R-UA*(T-Ta))/(F_A1*sigΘcp) #K/L
    return [dXdV,dTdV]

#initial conditions for x
x=[0,300] #[conversion,K]

#Volume array
V=np.linspace(0,V1*1000,1000) #L

#solution array for conversion
sol=odeint(func,x,V)
print('Outlet conversion of PFR =',sol[-1,0],'.')

#calculating overall conversion
Xoverall=(F_A0-F_A1*(1-sol[-1,0]))/F_A0
print('Overall conversion=',Xoverall,'.')
```

```
Outlet conversion of PFR = 0.10418549561169185 .
Overall conversion= 0.2833483964893535 .
```

Problem 3

```
In [8]: #known values
V=10 #L
T0=300 #K
V̇0=2 #L/s
c=6 #mol/L
H_A273_cir=-10 #kcal/mol
H_B273_cir=-5 #kcal/mol
H_P273_cir=-20 #kcal/mol
c_PA=10 #cal/(mol*K)
c_PB=12 #cal/(mol*K)
c_PP=22 #cal/(mol*K)
k_300=0.02 #L/(mol*s)
T_k_300=300 #K
Ea=8000 #cal/mol
T_r=273 #K
R=1.987 #cal/(mol*K)

#calculated values
ΔH_R273_cir=H_P273_cir-H_A273_cir-H_B273_cir #kcal/mol
print('ΔH_R273_cir=',ΔH_R273_cir,'kcal/mol')
Θ_A=1
Θ_B=1
Θ_P=0
x_A0=0.5
x_B0=0.5
Δc_Pi=c_PP-c_PA-c_PB #cal/(mol*K)
F_A0=x_A0*c*V̇0 #mol/s

def func(x):
    ΔH_R=ΔH_R273_cir*1000+Δc_Pi*(x[1]-T_r) #cal/mol (1000 converts kcal
 to cal)
    k=k_300*np.e**(Ea/R*(1/T_k_300-1/x[1])) #L/(mol*s)
    c_A=x_A0*c*(1-x[0]) #mol/L
    c_B=x_B0*c*(1-x[0]) #mol/L
    r_A=-k*c_A*c_B #mol/(L*s)
    r1=T0-x[0]*ΔH_R/(Θ_A*c_PA+Θ_B*c_PB)-x[1] #K
    r2=V-F_A0*x[0]/(-r_A)
    return [r1,r2]

sol=fsolve(func,[.9,500])
print('X=',sol[0],'.')
print('T=',sol[1],' K.')
```

```
ΔH_R273_cir= -5 kcal/mol
X= 0.8835805208367906 .
T= 500.81375473563423  K.
```