Homework 7

Problem 3.1

Calculate dew point temperature at 2 atm for an equimolar mixture of ethylene, n-butane, iso-pentane, and n-hexane and the composition of the first droplet using the DePriester correlation.

Component	$z_i = y_i$	a_1	a_2	a_3	b_1	b_2	b_3
Ethylene	0.25	-600076.875	0	7.90595	-0.84677	42.94594	0
n-Butane	0.25	-1280557	0	7.94986	-0.96455	0	0
iso-Pentane	0.25	-1481583	0	7.58071	-0.93159	0	0
n-Hexane	0.25	-1778901	0	6.96783	-0.84634	0	0

For the dew point temperature,

$$\sum_{i=1}^{C} \frac{z_i}{K_i} = 1 \tag{1}$$

Using the dePriester correlation,

$$ln(K) = \frac{a_1}{T^2} + \frac{a_2}{T} + a_3 + b_1 ln(P) + \frac{b_2}{P^2} + \frac{b_3}{P} \tag{2}$$

where a_1, a_2, a_3, b_1, b_2, and b_3 are constants listed in the table above. T and p are in \degree$R$ and $psia$ respectively.

The following code iterates T in (2) until (1) holds true.

In [ ]:

```
import numpy as np
import scipy.optimize as opt
from redlich_kwong_kvalue import redlich_kwong_kvalue
p = 2                                                           #pressure atm
zi = .25                                                        #equimol
```

In [ ]:

```
a1 = np.array([-600076.875,-1280557,-1481583,-1778901])         #depriester coeff
a2 = np.zeros(4)
a3 = np.array([7.90595,7.94986,7.58071,6.96783])
b1 = np.array([-.84677,-.96455,-.93159,-.84634])
b2 = np.array([42.94594,0,0,0])
b3 = np.zeros(4)

def depriester(a1,a2,a3,b1,b2,b3,T,p):                          #input t and p as
    p = p*14.6959
    T = T*1.8
    K = np.exp(a1/T**2+a2/T+a3+b1*np.log(p)+b2/p**2+b3/p)
    return K
```

```
def resFun_dep(T,p):
    Ki = depriester(a1,a2,a3,b1,b2,b3,T,p)
    kz = np.sum(zi/Ki)-1
    return kz
T_dep = opt.fsolve(resFun_dep,100,p)[0]
print(T_dep,resFun_dep(T_dep,p))
comp = zi/depriester(a1,a2,a3,b1,b2,b3,T_dep,p)
print(comp,sum(comp))
```

```
327.48783201025356 2.922107000813412e-13
[0.00863138 0.09162384 0.21143786 0.68830692] 1.0000000000002922
```

Problem 3.2

Calculate dew point temperature at 2 atm for an equimolar mixture of ethylene, n-butane, iso-pentane, and n-hexane and the composition of the first droplet using Raoult's law.

Component	$z_i = y_i$	a	b	c
Ethylene	0.25	3.87261	584.146	-18.307
n-Butane	0.25	4.35576	1175.581	-2.071
iso-Pentane	0.25	3.97183	1021.864	-43.231
n-Hexane	0.25	4.00266	1171.53	-48.784

For Raoult's law,

$$K_i = \frac{P_i^s}{P} \tag{3}$$

where P_i^s can be found by the Antoine equation,

$$log(P_i^s) = a - \frac{b}{c+T} \tag{4}$$

where a, b, and c are constants listed in the table above. The following code combines (3) and (4) then iterates T until (1) holds.

In [ ]:

```
def antoine(a,b,c,T):                                  #T in k
    return (10**(a-b/(T+c)))*.986923                #return pSat in atm
a = np.array([3.87261,4.35576,3.97183,4.00266])
b = np.array([584.146,1175.581,1021.864,1171.53])
c = np.array([-18.307,-2.071,-43.231,-48.784])

def resFun_rlt(T,p):
    Ki = antoine(a,b,c,T)/p
    kz = np.sum(zi/Ki)-1
    return kz

T_rlt = opt.fsolve(resFun_rlt,100,p)[0]
print(T_rlt,resFun_rlt(T_rlt,p))
comp = zi/(antoine(a,b,c,T_rlt)/p)
print(comp,sum(comp))
```

```
330.8175973983606 6.661338147750939e-16
[0.00502667 0.0841154  0.19323956 0.71761837] 1.0000000000000007
```

Problem 3.3

Calculate dew point temperature at 2 atm for an equimolar mixture of ethylene, n-butane, iso-pentane, and n-hexane and the composition of the first droplet using The Redlich-Kwong equation of state.

Component	Tc (K)	Pc (MPa)	ω
Ethylene	282.4	5.032	0.085
n-Butane	425.2	3.797	0.193
iso-Pentane	460.4	3.381	0.228
n-Hexane	507.4	3.012	0.305

The RK EOS is shown in the "Thermo Review" lecture notes on slide 9 and it takes critical temperatures and pressures as well as the mixture composition, temperature and pressure to solve for the fugacity coefficients. The fugacity coefficients are related to the K value by

$$K_i = \frac{\phi^L}{\phi_i^V} \tag{5}$$

At the dew point, $z_i = y_i$ so for the RK EOS, P, Tc, Pc, and y_i are known. The initial guess for x_i and T are the values obtained by solving this problem using Raoult's law. </br></br> The code below uses the initial guesses to iterate T and x_i until the error is less than 0.01

In [ ]:

```
def rk_residual(T,P,Tc,Pc,x,y):
    K = redlich_kwong_kvalue(Tc,Pc,T,P,x,y)
    return np.sum(y/K)-1

Tc = np.array([282.4,425.2,460.4,507.4])  # ensure proper units
Pc = np.array([5.032,3.797,3.381,3.012]) * 1e6  # ensure proper units
K_rlt = antoine(a,b,c,T_rlt)/p
K_rk = K_rlt  # start with a guess for the K-value from Raoult's law.
x_rk = zi/K_rlt  # guess for the liquid (from the K-value guess)
T_rk = T_rlt
err  = 1

# print(rk_residual(330,p*101325,Tc,Pc,x_rk,zi))
while err>0.01:
    T = opt.fsolve(rk_residual,T_rk, args=(p*101325,Tc,Pc,x_rk,zi))
    err = abs(T[0] - T_rk)
    # update temperature and composition
    T_rk = T[0]
    K_rk = redlich_kwong_kvalue(Tc,Pc,T_rk,p*101325,x_rk,zi)
    x_rk = zi/K_rk

print('Dew point temperature (K): {:.1f}\n'.format(T_rk) )
print(x_rk,sum(x_rk))
```

```
Dew point temperature (K): 310.4

[0.01085676 0.11629604 0.23856354 0.63428366] 1.000000000000001
```

Problem 3.4

Calculate dew point temperature at 2 atm for an equimolar mixture of ethylene, n-butane, iso-pentane, and n-hexane and the composition of the first droplet using The Soave-Redlich-Kwong equation of state.

The SRK EOS is similar to the RK EOS except it inclues an acentric factor, ω. The code below follows the same strategy as with the RK EOS

In [ ]:

```
def srk_residual(T,P,Tc,Pc,x,y,w):
    K = redlich_kwong_kvalue(Tc,Pc,T,P,x,y,w)
    return np.sum(zi/K)-1

# define the acentric factor for each species
w = np.array([.085,.193,.228,.305])
K_srk = K_rlt  # start with a guess for the K-value from Raoult's law.
x_srk = zi/K_rlt  # guess for the liquid (from the K-value guess)
T_srk = T_rlt
err2  = 1

# print(rk_residual(330,p*101325,Tc,Pc,x_rk,zi))
while err2>0.01:
    T = opt.fsolve(srk_residual,T_srk, args=(p*101325,Tc,Pc,x_srk,zi,w))
    err2 = abs(T[0] - T_srk)
    # update temperature and composition
    T_srk = T[0]
    K_srk = redlich_kwong_kvalue(Tc,Pc,T_srk,p*101325,x_srk,zi,w)
    x_srk = zi/K_srk

print('Dew point temperature (K): {:.1f}\n'.format(T_srk) )

print(x_srk,sum(x_srk))
```

```
Dew point temperature (K): 330.5

[0.00783842 0.09096958 0.20238401 0.69880799] 1.0000000000000027
```

Method	T (K)	Ethylene	n-Butane	iso-Pentane	n-Hexane
DePriester	327.5	0.009	0.092	0.211	0.688
Raoult	330.8	0.005	0.084	0.193	0.718
RK	310.4	0.011	0.116	0.239	0.634
SRK	330.5	0.008	0.090	0.202	0.699