cheg325 homework 5 SIS 11.2-4

AUTHOR
k. wodehouse

PUBLISHED
March 28, 2025

(a)

starting off with our most useful equation for liquid liquid equlibrium

$$x_1^I \gamma_1^I = x_1^{II} \gamma_1^{II}$$

now if we apply the data given in the problem statement

$$0.48\gamma_1(x_1 = 0.48) = 0.94\gamma_1(x_1 = 0.94)$$

we know the one constant margules is as follows

$$\frac{\underline{G}^{\text{ex}}}{RT} = Ax_1x_2$$

$$\ln \gamma_1 = Ax_2^2$$

so now the equation we ultimately need to solve is

$$x_1^I \exp[A(1-x_1^I)^2] = x_1^{II} \exp[A(1-x_1^{II})^2]$$

$$\frac{\exp[A(1-x_1^{II})^2]}{\exp[A(1-x_1^I)^2]} = \frac{x_1^I}{x_1^{II}}$$

$$\exp[A(1-x_1^{II})^2 - A(1-x_1^I)^2] = \frac{x_1^I}{x_1^{II}}$$

$$A\left((1-x_1^{II})^2 - (1-x_1^I)^2\right) = \ln \frac{x_1^I}{x_1^{II}}$$

which gives

$$A = \frac{\ln \frac{x_1^I}{x_1^{II}}}{(1-x_1^{II})^2 - (1-x_1^I)^2} = 2.52$$

```
import numpy as np
A = np.log(0.48/0.94) / ((1-0.94)**2 - (1-0.48)**2)
print(f'A: {A:.2f}')
```

```
A: 2.52
```

which is fine and all, but for our 1-constant margules we know that the excess gibbs is modeled as being perfectly symmetric which means our activity coefficients are symmetric too. we need to check this to confirm or reject the 1-constant margules as modeling the system well.

```
A = np.log((1-0.48)/(1-0.94)) /  ((0.94)**2 - (0.48)**2)
print(f'A: {A:.2f}')
```

```
A: 3.31
```

which is very clearly not the same number! this means that the activity coefficients will need different values at infinite dilution and cannot be accurately modeled with the one-constant margules model.

(b)

this part is a little annoying since i already arbitrarily defined benzene as component 1. looking at our regular solution theory

$$RT \ln \gamma_i = \underline{V}_i(\delta_i - \bar{\delta})^2 \qquad (9.6\text{-}11)$$

which, rearranging for the activity coeff

$$\gamma_i = \exp\left(\frac{\underline{V}_i(\delta_i - \bar{\delta})^2}{RT}\right)$$

and like... the $\bar{\delta}$ is (for each phase)

$$\bar{\delta} = (1-\Phi_2)\delta_1 + (\Phi_2)\delta_2 = \Phi_2(\delta_2 - \delta_1) + \delta_1$$

which makes

$$(\delta_1 - \bar{\delta})^I = (\delta_1 - \Phi_2(\delta_2 - \delta_1) - \delta_1) = -\Phi_2(\delta_2 - \delta_1)$$

and, somewhat obviously

$$((\delta_1 - \bar{\delta})^2)^I = \Phi_2^2(\delta_1 - \delta_2)^2$$

so our activity coefficient is

$$\gamma_i = \exp\left(\frac{\underline{V}_i \Phi_j^2(\delta_1 - \delta_2)^2}{RT}\right)$$

now we just have the same problem as before but this time i'll be even more lazy and let mathematica do the algebra for me because its 11:19pm on a tuesday during spring break.

$$x_1^I \gamma_1^I = x_1^{II} \gamma_1^{II}$$

$$x_1^I \exp\left(\frac{\underline{V}_1(\Phi_2^I)^2(\delta_1 - \delta_2)^2}{RT}\right) = x_1^{II} \exp\left(\frac{\underline{V}_1(\Phi_2^{II})^2(\delta_1 - \delta_2)^2}{RT}\right)$$

now we're finally left with n equations and n unkonwns the other equations are

$$\Phi_2^I = \frac{(1-x_1^I)\underline{V}_2}{x_1^I\underline{V}_1 + (1-x_1^I)\underline{V}_2}$$

$$\Phi_2^{II} = \frac{(1-x_1^{II})\underline{V}_2}{x_1^{II}\underline{V}_1 + (1-x_1^{II})\underline{V}_2}$$

from table 9.6-1 we get $\underline{V}$ for benzene and the other is given in the problem.

```
V1 = 89.0 * 1e-3# cm^3 / mol --> m3/mol
V2 = 0.226

x11 = 0.48
x12 = 0.94

phi21 = (1-x11) * V2 / (x11*V1 + (1-x11)*V2)
phi22 = (1-x12) * V2 / (x12*V1 + (1-x12)*V2)

print(f'Phi_2 I: {phi21:.4f}')
print(f'Phi_2 II: {phi22:.4f}')

phi11 = 1 - phi21
phi12 = 1 - phi22

print(f'Phi_1 I: {phi11:.4f}')
print(f'Phi_1 II: {phi12:.4f}')
```

```
Phi_2 I: 0.7334
Phi_2 II: 0.1395
Phi_1 I: 0.2666
Phi_1 II: 0.8605
```

now we can calculate the solubility parameter for perfluoro-n-heptane by isolating $(\delta_1 - \delta_2)$ and then using that to find our δ_2. a very similar algebra process as the first part so i'll just cut to the chase

$$(\delta_1 - \delta_2)^2 = \frac{RT \ln \frac{x_i^{II}}{x_i^I}}{\underline{V}_i((\Phi_j^I)^2 - (\Phi_j^{II})^2)}$$

now we can just plug in the experimental data point given by the problem and find the values for δ_2

```
from scipy.constants import R
T = 273.15 + 100.0

leftside_1 = R*T * np.log(x12 / x11) / (V1 * (phi21**2 - phi22**2)) / 4184

d1_minus_d2_1 = np.sqrt(leftside_1)

delta_1 = 9.2
delta_2_1 = [delta_1 + d1_minus_d2_1, delta_1 - d1_minus_d2_1]
print(f"set: {np.round(delta_2_1,3)}")
```

```
set: [12.487  5.913]
```

the 5.91 value is pretty close to the value of 6 in the table so in some way it lines up with experiment. not sure how (without already knowing it's 6) you'd figure out to use the lower value.

```
# filler text
```