cheg325 homework4 extra1

AUTHOR
k.wodehouse

PUBLISHED
March 1, 2025

using this as a notebook to run deepseek's code

(a)

```python
import numpy as np
from scipy.optimize import fsolve

# Given data
z = np.array([0.55, 0.25, 0.20])  # Mole fractions of C5, C6, C7 in feed
P_vap = np.array([2.755, 1.021, 0.390])  # Vapor pressures at 69°C in bar
L = 0.5
V = 1 - L

# Define the function to find the root of
def f(P):
    K = P_vap / P
    x = z / (L + K * V)
    return np.sum(x) - 1

# Initial guess for pressure
P_initial_guess = 1.0  # Start with an initial guess of 1 bar

# Solve for the pressure
P_solution = fsolve(f, P_initial_guess)[0]

# Calculate the equilibrium compositions
K_solution = P_vap / P_solution
x_solution = z / (L + K_solution * V)
y_solution = K_solution * x_solution

print(f"Part a: Pressure = {P_solution:.4f} bar")
print(f"Liquid composition (x_C5, x_C6, x_C7) = {x_solution}")
print(f"Vapor composition (y_C5, y_C6, y_C7) = {y_solution}")
```

```
Part a: Pressure = 1.4906 bar
Liquid composition (x_C5, x_C6, x_C7) = [0.38620617 0.29674503 0.3170488 ]
Vapor composition (y_C5, y_C6, y_C7) = [0.71379383 0.20325497 0.0829512 ]
```

the output isn't pretty, but it matches up with my solution (which is hopefully right).

(b)

```python
import numpy as np
from scipy.optimize import fsolve

# Given data
z = np.array([0.55, 0.25, 0.20])  # Mole fractions of C5, C6, C7 in feed
P_vap = np.array([2.755, 1.021, 0.390])  # Vapor pressures at 69°C in bar
x_C5_target = 0.30

# Define the function to find the root of
def f(variables):
    P, L = variables
    V = 1 - L
    K = P_vap / P

    # Calculate x_C5 and ensure it matches the target
    x_C5 = z[0] / (L + K[0] * V)

    # Calculate x_C6 and x_C7
    x_C6 = z[1] / (L + K[1] * V)
    x_C7 = z[2] / (L + K[2] * V)

    # Sum of liquid mole fractions should be 1
    sum_x = x_C5 + x_C6 + x_C7

    # Return the difference from the target conditions
    return [x_C5 - x_C5_target, sum_x - 1]

# Initial guesses for pressure and L
initial_guesses = [1.0, 0.5]  # Start with an initial guess of 1 bar and L = 0.5

# Solve for the pressure and L
solution = fsolve(f, initial_guesses)
P_solution, L_solution = solution

# Calculate the equilibrium compositions
V_solution = 1 - L_solution
K_solution = P_vap / P_solution
x_solution = z / (L_solution + K_solution * V_solution)
y_solution = K_solution * x_solution

print(f"Part b: Pressure = {P_solution:.4f} bar")
print(f"Fraction of feed that is liquid (L) = {L_solution:.4f}")
print(f"Liquid composition (x_C5, x_C6, x_C7) = {x_solution}")
print(f"Vapor composition (y_C5, y_C6, y_C7) = {y_solution}")
```

```
Part b: Pressure = 1.2850 bar
Fraction of feed that is liquid (L) = 0.2715
Liquid composition (x_C5, x_C6, x_C7) = [0.3        0.29400379 0.40599621]
Vapor composition (y_C5, y_C6, y_C7) = [0.64318246 0.23359847 0.12321907]
```

this output is similarly ugly with way too many sigfigs and literally just printing out lists, but the numbers match my solution.

what worked?

- asking it to write code
- breaking down the question into part a and part b (since it failed to do both at the start)
- literally copy pasting screenshots of the question into it

what didn't work?

- asking it to solve it all at once
- commenting on an inconsistency in the provided code (the 'fixed' code was not fixed)
- asking it to solve the question without saying to use code (it didn't give numerical answers with the first prompt)

how could these results be verified?

- checks for physical viability of the answers (does $\sum x_i = 1$? is the pressure reasonably close to the component's vapor pressures?)
- putting their code into IDE and testing if it runs correctly
- reading through their derivations to make sure they didn't make an algebra mistake
- impossible to truly verify them without knowing the correct answer already from my previous work with this question

```
# filler
```