

You:
solve 10.1-10

DeepSeek:

To solve problem 10.1-10, we need to perform a flash calculation for the given mixture at 69°C. The mixture consists of 55 mol% n-pentane (C5), 25 mol% n-hexane (C6), and 20 mol% n-heptane (C7). The vapor pressures at 69°C are given as:

- $P_{C5vap} = 2.755 \text{ bar}$
- $P_{C6vap} = 1.021 \text{ bar}$
- $P_{C7vap} = 0.390 \text{ bar}$

Part a: Pressure for Half the Feed as Liquid

We need to find the pressure where the liquid stream contains exactly half the number of moles of the feed. This implies that the fraction of the feed that is liquid (L) is 0.5.

The flash calculation involves solving the following equations:

1. Material Balance:

$$z_i = x_i L + y_i V$$

where z_i is the mole fraction of component i in the feed, x_i is the mole fraction in the liquid, y_i is the mole fraction in the vapor, L is the fraction of the feed that is liquid, and V is the fraction that is vapor ($V = 1 - L$).

2. Equilibrium Relationship:

$$y_i = K_i x_i$$

where K_i is the equilibrium constant for component i , given by:

$$K_i = P_{i,vap} / P$$

where P is the total pressure.

Given $L = 0.5$, we can set up the equations and solve for P .

Part b: Pressure for n-Pentane Mole Fraction of 0.30 in Liquid

We need to find the pressure where the liquid stream has an n-pentane mole fraction of 0.30. This involves solving for P and L such that $x_{C5} = 0.30$.

Solution Approach

1. For Part a:

- Assume a pressure P .
- Calculate K_i for each component.

- Use the material balance and equilibrium relationships to solve for x_i and y_i .
- Iterate to find the pressure P that satisfies $L = 0.5$.

2. For Part b:

- Assume a pressure P .
- Calculate K_i for each component.
- Use the material balance and equilibrium relationships to solve for x_i and y_i .
- Iterate to find the pressure P and L that satisfy $x_{C5} = 0.30$.

Numerical Solution

This problem typically requires iterative numerical methods or the use of process simulation software to solve accurately. However, the steps outlined above provide the framework for solving the problem.

If you have access to software like Aspen Plus or similar, you can set up the flash calculation with the given conditions and solve for the required pressures and compositions. Alternatively, you can use iterative methods like the Newton-Raphson method to solve the equations manually or with a programming tool like Python or MATLAB.

Would you like a more detailed step-by-step solution or assistance with setting up the equations for a specific method?

You:

implement the solution math in python using numpy and scipy's fsolve function

DeepSeek:

To solve the problem using Python, we can use the `numpy` library for numerical operations and `scipy.optimize.fsolve` for solving the system of nonlinear equations. Below is a Python implementation for both parts of the problem.

Python Implementation

python

Copy

```
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

# Part a: Find pressure P where L = 0.5
def equations_a(variables):
    P, x_C5, x_C6, x_C7, y_C5, y_C6, y_C7 = variables
    L = 0.5
    V = 1 - L

    # Equilibrium constants
    K = P_vap / P

    # Material balance and equilibrium relationships
    eq1 = z[0] - (x_C5 * L + y_C5 * V)
    eq2 = z[1] - (x_C6 * L + y_C6 * V)
    eq3 = z[2] - (x_C7 * L + y_C7 * V)
    eq4 = y_C5 - K[0] * x_C5
```

```

eq4 = y_C5 - K[0] * x_C5
eq5 = y_C6 - K[1] * x_C6
eq6 = y_C7 - K[2] * x_C7

return [eq1, eq2, eq3, eq4, eq5, eq6]

# Initial guesses for variables
initial_guesses_a = [1.0, 0.5, 0.25, 0.25, 0.5, 0.25, 0.25]

# Solve the system of equations
solution_a = fsolve(equations_a, initial_guesses_a)

P_a = solution_a[0]
x_a = solution_a[1:4]
y_a = solution_a[4:7]

print(f"Part a: Pressure = {P_a:.4f} bar")
print(f"Liquid composition (x_C5, x_C6, x_C7) = {x_a}")
print(f"Vapor composition (y_C5, y_C6, y_C7) = {y_a}")

# Part b: Find pressure P and L where x_C5 = 0.30
def equations_b(variables):
    P, L, x_C6, x_C7, y_C5, y_C6, y_C7 = variables
    x_C5 = 0.30
    V = 1 - L

    # Equilibrium constants
    K = P_vap / P

    # Material balance and equilibrium relationships
    eq1 = z[0] - (x_C5 * L + y_C5 * V)
    eq2 = z[1] - (x_C6 * L + y_C6 * V)
    eq3 = z[2] - (x_C7 * L + y_C7 * V)
    eq4 = y_C5 - K[0] * x_C5
    eq5 = y_C6 - K[1] * x_C6
    eq6 = y_C7 - K[2] * x_C7

    return [eq1, eq2, eq3, eq4, eq5, eq6]

# Initial guesses for variables
initial_guesses_b = [1.0, 0.5, 0.25, 0.25, 0.5, 0.25, 0.25]

# Solve the system of equations
solution_b = fsolve(equations_b, initial_guesses_b)

P_b = solution_b[0]
L_b = solution_b[1]
x_b = [0.30, solution_b[2], solution_b[3]]
y_b = solution_b[4:7]

print(f"\nPart b: Pressure = {P_b:.4f} bar")
print(f"Fraction of feed that is liquid (L) = {L_b:.4f}")
print(f"Liquid composition (x_C5, x_C6, x_C7) = {x_b}")
print(f"Vapor composition (y_C5, y_C6, y_C7) = {y_b}")

```

Explanation

1. Part a:

- We define a system of equations based on material balance and equilibrium relationships.
- We use `fsolve` to find the pressure P and the compositions x_i and y_i that satisfy the equations when $L = 0.5$.

2. Part b:

- We define a similar system of equations but with the additional constraint that $x_{C5} = 0.30$.
- We use `fsolve` to find the pressure P , the fraction of the feed that is liquid L , and the compositions x_i and y_i .

Running the Code

When you run the code, it will output the pressure and compositions for both parts of the problem. Note that the initial guesses are important for the convergence of the `fsolve` function, and you may need to adjust them based on the specific problem.

This implementation provides a numerical solution to the problem using Python, leveraging the power of `numpy` and `scipy` for efficient computation.

You:

let's go one part at a time. for part a try setting up a function that will have a root when the right pressure is input instead of trying to `fsolve` with 3 variables. you may continue to assume this is an ideal gas mixture vapor phase and ideal mixture liquid phase.

DeepSeek:

To solve part (a) of the problem, we can simplify the approach by focusing on finding the pressure P that satisfies the condition where the liquid stream contains exactly half the number of moles of the feed ($L = 0.5$). We can set up a function that will have a root when the correct pressure P is input. This function will use the material balance and equilibrium relationships.

Simplified Approach for Part (a)

1. Material Balance:

$$z_i = x_i L + y_i V$$

$$\text{where } L = 0.5 \text{ and } V = 0.5.$$

2. Equilibrium Relationship:

$$y_i = K_i x_i$$

$$\text{where } K_i = P_i^{\text{vap}} / P$$

3. Combining the Equations:

$$z_i = x_i L + K_i x_i V$$

$$z_i = x_i (L + K_i V)$$

$$x_i = z_i / (L + K_i V)$$

4. Sum of Liquid Mole Fractions:

$$\sum x_i = 1$$

Substituting x_i :

$$\sum \frac{z_i}{L + K_i V} = 1$$

$$\sum L + K_i V Z_i = 1$$

We can define a function $f(P)$ that represents the difference between the sum of the liquid mole fractions and 1. The root of this function will give us the correct pressure P .

Python Implementation for Part (a)

python

Copy

```
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}")
```

Explanation

1. Function Definition:

- The function $f(P)$ calculates the sum of the liquid mole fractions x_i and subtracts 1. The root of this function corresponds to the pressure P where the sum of the liquid mole fractions equals 1.

2. Solving for Pressure:

- We use `fsolve` to find the root of $f(P)$, which gives us the pressure P .

3. Calculating Compositions:

- Once the pressure P is known, we calculate the equilibrium constants K_i , liquid mole fractions x_i , and vapor mole fractions y_i .

Running the Code

When you run the code, it will output the pressure and the equilibrium compositions for the liquid and vapor phases. This approach simplifies the problem by focusing on finding the pressure that satisfies the material balance and equilibrium conditions.

This method leverages the ideal gas and ideal mixture assumptions to streamline the calculation process.

You:

good work--that's the correct solution for part a. now try part b with the same assumptions

DeepSeek:

For part (b), we need to find the pressure P and the fraction of the feed that is liquid L such that the liquid stream has an n-pentane mole fraction of 0.30 ($x_{C5} = 0.30$). We will use the same assumptions of an ideal gas mixture for the vapor phase and an ideal mixture for the liquid phase.

Simplified Approach for Part (b)

1. Material Balance:

$$z_i = x_i L + y_i V$$

where $V = 1 - L$.

2. Equilibrium Relationship:

$$y_i = K_i x_i$$

where $K_i = P_i^{\text{vap}} / P$.

3. Given Condition:

$$x_{C5} = 0.30$$

4. Combining the Equations:

For n-pentane (C_5):

$$z_{C5} = x_{C5} L + K_{C5} x_{C5} V$$

$$z_{C5} = x_{C5} (L + K_{C5} V)$$

$$L + K_{C5} V = z_{C5} / x_{C5}$$

For the other components (C_6 and C_7):

$$x_i = (z_i - K_i V) / L$$

5. Sum of Liquid Mole Fractions:

$$\sum x_i = 1$$

Substituting x_i :

$$x_{C5} + \sum_{i \neq C5} (z_i - K_i V) / L = 1$$

We can define a function $f(P, L)$ that represents the difference between the sum of the liquid mole fractions and 1. The root of this function will give us the correct pressure P and fraction L .

Python Implementation for Part (b)

python
Copy

```
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}")
```

Explanation

1. Function Definition:

- The function $f(P, L)$ calculates the difference between the target n-pentane mole fraction and the sum of the liquid mole fractions minus 1. The root of this function corresponds to the pressure P and fraction L where the conditions are satisfied.

2. Solving for Pressure and Fraction:

- We use `fsolve` to find the root of $f(P, L)$, which gives us the pressure P and fraction L .

3. Calculating Compositions:

- Once the pressure P and fraction L are known, we calculate the equilibrium constants K_i , liquid

mole fractions x_i , and vapor mole fractions y_i .

Running the Code

When you run the code, it will output the pressure, the fraction of the feed that is liquid, and the equilibrium compositions for the liquid and vapor phases. This approach leverages the ideal gas and ideal mixture assumptions to streamline the calculation process for part (b).