Homework Assignment 2

Problem 1
An equimolar mixture of propane and butane is fed to a flash separator operating at 40 degC. Assume Raoult's law holds.

1.1 Plot the bubble and dew curves (p – x curves) as a function of the mole fraction of propane. What additional information do you require to complete your calculation? Find it online and cite your source.

We begin from Raoult's Law, which states that:

$$x_i.p_i^* = y_i.p$$
$$\sum_i x_i.p_i^* = p$$

where:
x_i: liquid mole fraction of component i
y_i: vapor mole fraction of component i
p_i: saturation pressure of component i
p: pressure of the entire mixture

Bubble point is the first point when the liquid starts to vaporize and hence, the mixture is transitioning from an all liquid phase to multi-phase. Using Raoult's law and the fact that sum of mole fractions of all components must result 1, we get:

$$\sum_i x_i = \sum_i y_i = 1$$

$$p = x_p.p_p^* + x_b.p_b^*$$

$$p = x_p.p_p^* + (1 - x_p).p_b^* \; (Bubble\, Curve)$$

$$p = \frac{p_p^*.p_b^*}{y_p.p_b^* + (1 - y_p).p_p^*} \; (Dew\, Curve)$$

The subscripts p and b denote properties of propane and butane respectively. To plot these pressures with respect to liquid and vapor mole fractions, we have to calculate the saturation pressures of both propane and butane. This can be done using the Antoine equation, with the parameters A, B, C obtained from the *NIST Chemistry Webbook*.

Antoine Equation:

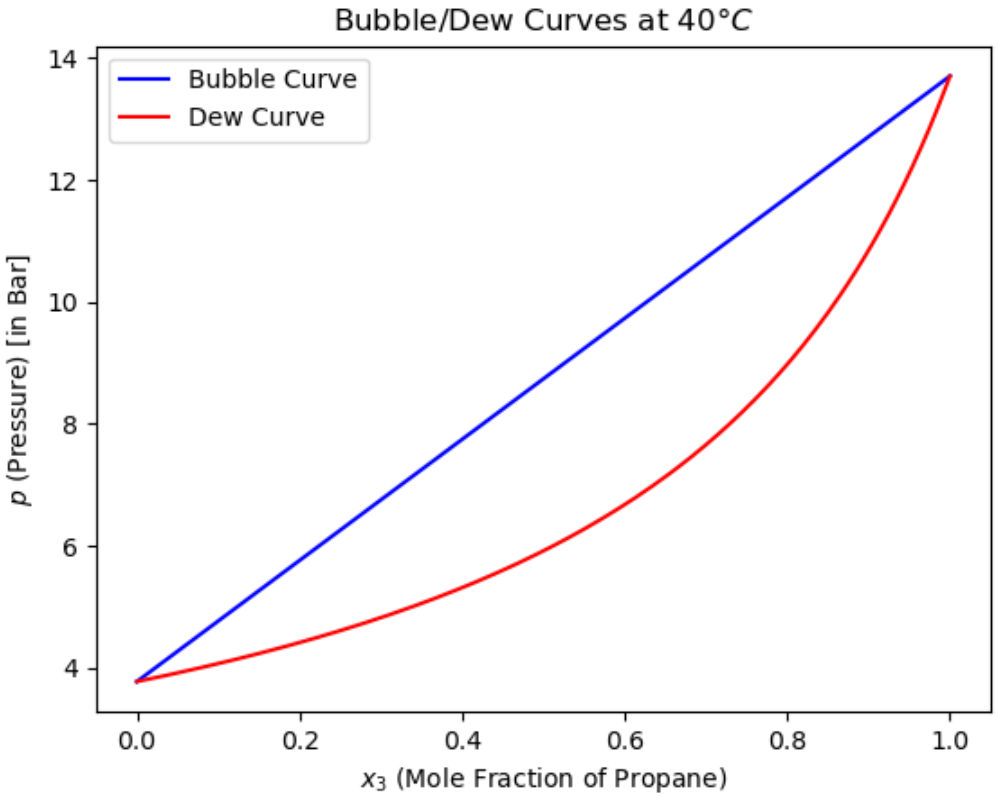
$$log(p^*) = A - \frac{B}{T + C}$$

Element	A	B	C
Propane	4.53678	1149.36	24.906
Butane	4.35576	1175.581	-2.071

Computing the value using $T = 313.15\, K$ with a python function, we get:

$$p_p^* = 13.705\, bar$$
$$p_b^* = 3.773\, bar$$

Using the pressure values, we obtain the following plot:



1.2 Determine the pressure at which the separator should be operated so that an equal number of moles of liquid and vapor are produced.

We begin with the information that the mixture is equimolar. So the number of moles of propane and butane are equal. Let L and V be the number of moles in liquid and vapor respectively. Using the mole fractions, we can represent the total number of moles as a sum of liquid contribution and vapor contribution.

$$n_p = n_b$$

$$x_p.L + y_p.V = x_b.L + y_b.V$$

$$(x_p - x_b).L = (y_b - y_p).V$$

$$\frac{L}{V} = \frac{y_b - y_p}{x_p - x_b}$$

$$\frac{L}{V} = \frac{1 - 2.y_p}{2.x_p - 1}$$

We need to determine the pressure for which $L = V$. We can substitute the values of mole fractions in the above equation

$$x_p = \frac{p - p_b^*}{p_p^* - p_b^*}$$

$$y_p = \frac{x_p . p_p^*}{p}$$

$$y_p = \frac{p - p_b^*}{p_p^* - p_b^*} . \frac{p_p^*}{p}$$

Solving the equation in python, we obtain the value of pressure to be **7.191 bar**

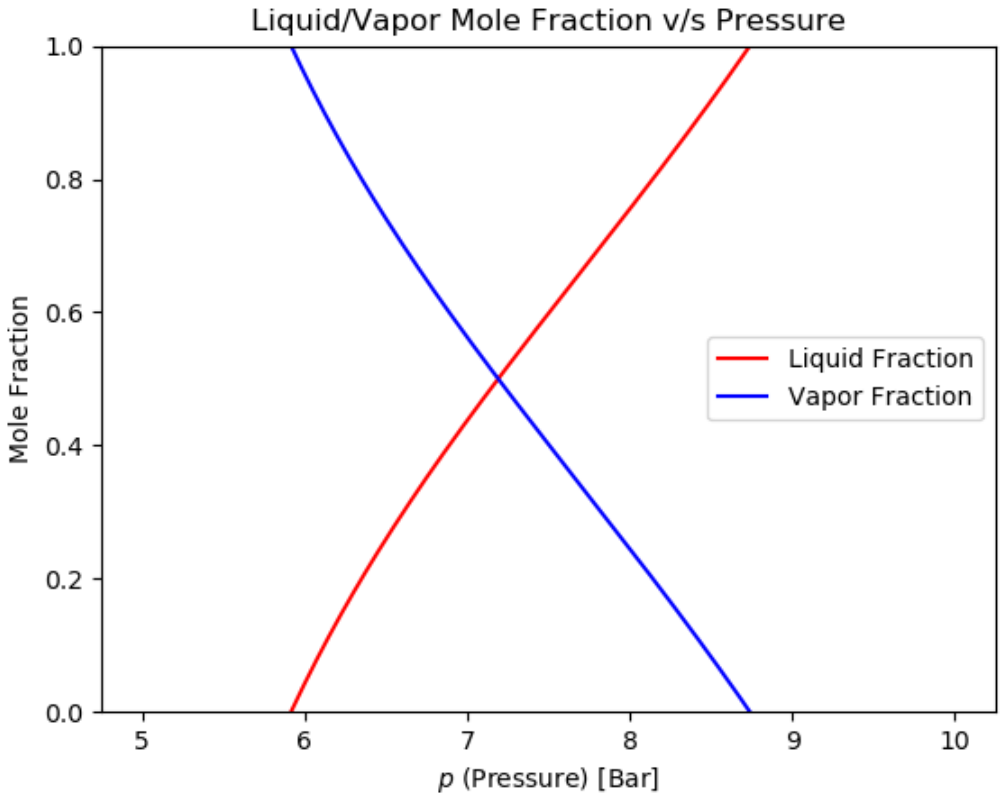
1.3. Plot the liquid fraction and vapor fraction as a function of pressure.

The liquid and vapor fractions are defined as:

$$\frac{L}{L+V} = \frac{1}{1+\frac{V}{L}}$$

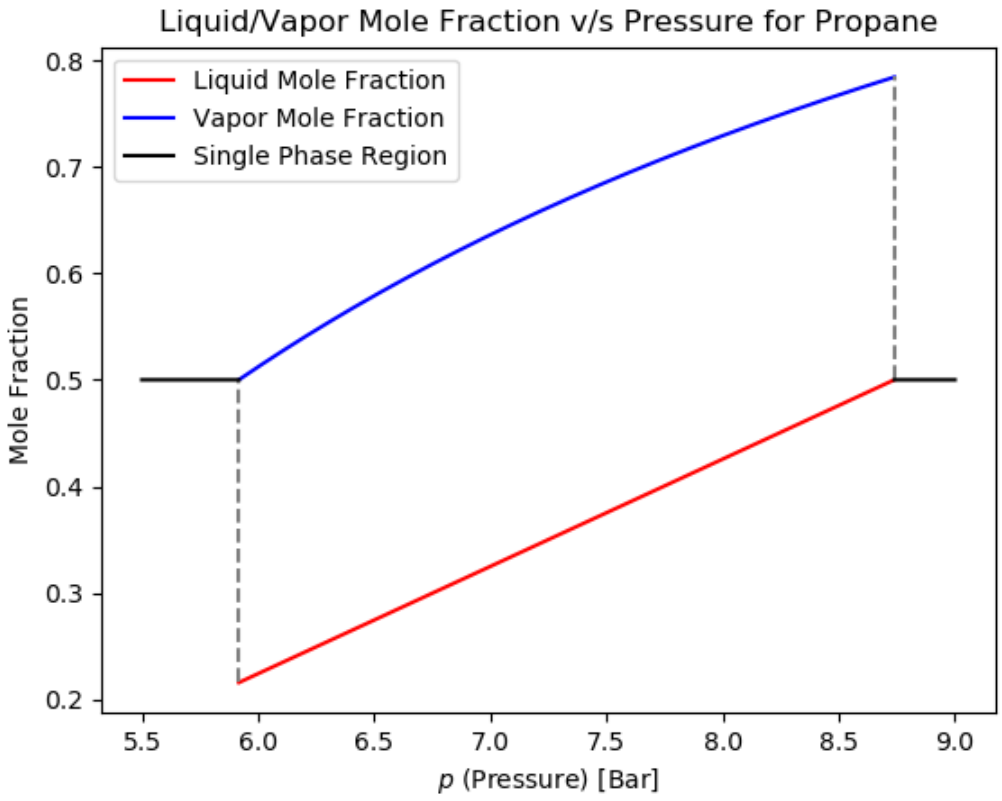
$$\frac{V}{L+V} = 1 - \frac{L}{L+V}$$

We can use the L/V expression from 1.2 and plot the values of the liquid and vapor fractions on y-axis with the pressure on the x-axis. We take a range of pressure values, say 5 - 10 bar and plot the values obtained. We get the below plot. The intersection point is at 0.5 mole fraction which is the point we calculated in 1.2 with a pressure value of 7.191 bar.



1.4. Plot the mole fraction of propane in the liquid and vapor as a function of pressure.

From the above obtained graphs, we find that the coexistence exists only between 5.917 bar and 8.739 bar. Lower pressure region is all liquid and high pressure region is all vapor, with a mole fraction of 0.5, as the mixture is equimolar. Using the above information, we obtain the following plot:



Problem 2
Construct a temperature - mole fraction phase diagram (bubble and dew curves) for a mixture of ethanol (1) with hexane (2) at a total pressure of 101.3 kPa.

Use the Wilson model:

$$ln\gamma_1 = -ln[x_1 + x_2\Lambda_{12}] + x_2\left[\frac{\Lambda_{12}}{x_1 + x_2\Lambda_{12}} - \frac{\Lambda_{21}}{x_2 + x_1\Lambda_{21}}\right]$$

$$ln\gamma_2 = -ln[x_2 + x_1\Lambda_{21}] + x_1\left[\frac{\Lambda_{21}}{x_1 + x_2\Lambda_{21}} - \frac{\Lambda_{12}}{x_1 + x_2\Lambda_{12}}\right]$$

with the parameters $\Lambda_{12} = 0.0952$ and $\Lambda_{21} = 0.2713$. Vapor pressures (with pressure and temperature unites of kPa and K, respectively) are given by

$$ln(p_1^*) = 16.1952 - \frac{3423.53}{T - 55.7172}$$

$$ln(p_2^*) = 14.0568 - \frac{2825.42.53}{T - 42.7089}$$

Hand in your code plus its output in the form of a clearly labeled plot. Also include a brief but clear explanation of what your code does. State whether there are positive or negative deviations from ideal behavior, and explain your reasoning. If there is an azeotrope, give its coordinates.

We begin with the Raoult's Law for non-ideal mixture:

$$x_i\gamma_i p_i^* = y_i p$$

As suggested, we can use the Wilson's model to compute the activity coefficients (γ_i), and substitute the expression for p_i^* to get a relation purely between temperature T and mole fraction x_1 and y_1.

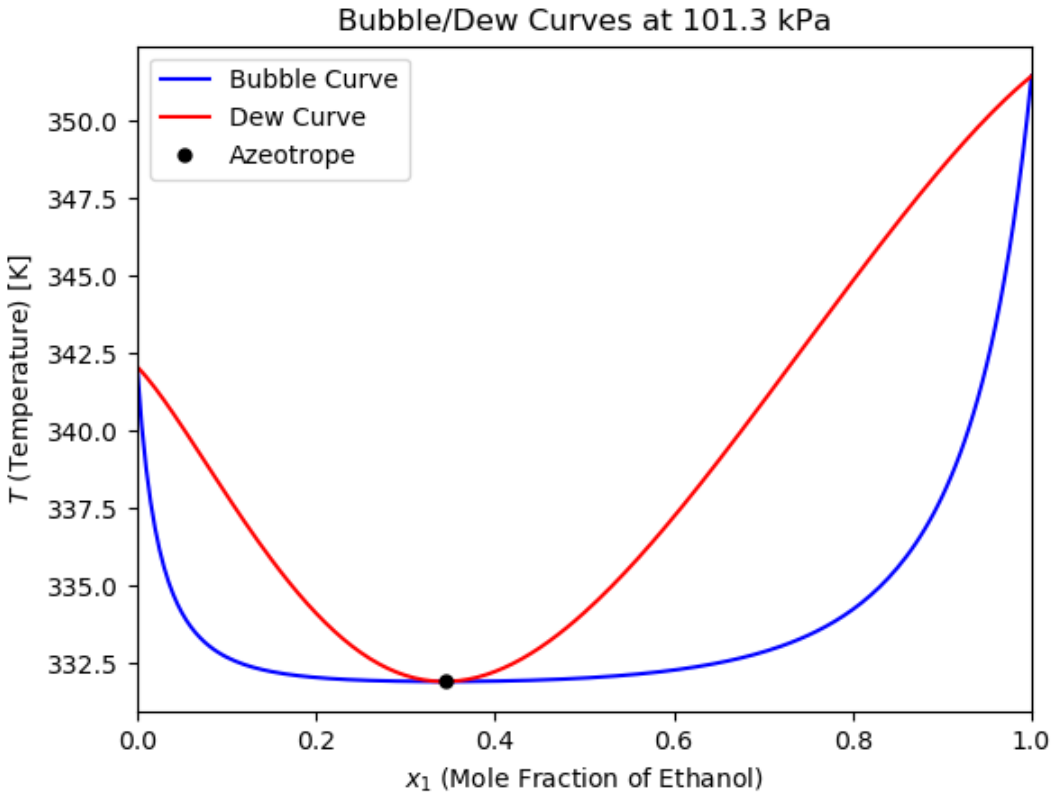
$$\sum_i x_i = 1 = \sum_i y_i$$

Computing for both conditions, we get the following equations:

$$\frac{x_1\gamma_1 p_1^*}{p} + \frac{x_2\gamma_2 p_2^*}{p} = 1$$

$$\frac{y_1 p}{\gamma_1 p_1^*} + \frac{y_2 p}{\gamma_2 p_2^*} = 1$$

Plugging in the expressions and using python solver to compute the temperature values, we obtain the following plot:



The azeotrope was found at **331.901 K** and at a mole fraction of **0.345**. The precision of these values can be improved if necessary, by taking a smaller step size in mole fraction line space.

In this graph, we observe that the boiling point of the azeotrope is less than the boiling points of the pure constituents. Hence, it is a positive azeotrope. This indicates that the deviation from ideal condition is positive. This is supported by the fact that both the activity coefficients are equal to or greater than 1 for all mole fraction values.

Appendix: Python Script

```
import matplotlib.pyplot as plt
import numpy as np
from sympy import *
from shapely.geometry import LineString

# Properties for Propane and Butane [From NIST]
A_propane, B_propane, C_propane = [4.53678, 1149.36, 24.906] # bar, bar*K, K

A_butane, B_butane, C_butane = [4.35576, 1175.581, -2.071] # bar, bar*K, K

T = 273.15 + 40 # Given temp, Units: K

# Antoine equation computes the pure pressure at certain temperature
def pure_pressure(A,B,C,T):
    p_star = 10**(A-B/(T+C)) # Antoine equation
    return p_star

# Pure pressure values
p_star_propane = pure_pressure(A_propane,B_propane,C_propane,T)
p_star_butane = pure_pressure(A_butane,B_butane,C_butane,T)

print("Pure Pressure (Propane): "+str(p_star_propane)+" bar")
print("Pure Pressure (Butane): "+str(p_star_butane)+" bar")

# Problem 1.1 Plotting the bubble and dew curves

# Defining line space for propane mole fraction
x_propane = np.linspace(0,1,201)

# Using Raoult's Law and the fact that x_propane + x_butane = 1, we get bubble
    ↪ curve pressure equation
p_bubble = p_star_butane + (p_star_propane - p_star_butane) * x_propane

# Using Raoult's Law and the fact that y_propane + y_butane = 1, we get dew curve
    ↪ pressure equation (here both x and y are represented on the same x-axis),
    ↪ hence y_propane = x_propane
p_dew = p_star_propane * p_star_butane / (p_star_propane+(p_star_butane-
    ↪ p_star_propane) * x_propane)

# Plotting the two curves
plt.figure(1)
plt.clf()
plt.plot(x_propane,p_bubble,color='blue',label='Bubble Curve')
plt.plot(x_propane,p_dew,color='red',label='Dew Curve')
plt.title("Bubble/Dew Curves at $40 \degree C$")
plt.xlabel(r"$x_3$"+" (Mole Fraction of Propane)")
plt.ylabel(r"$p$"+" (Pressure) [in Bar]")
plt.legend()
plt.savefig("BubbleDewCurve.png")

# Problem 1.2 Determine pressure where x = y
```

```
# Equating number of moles of liquid and vapor, and solving for p
p = var('p')
x_p = (p - p_star_butane)/(p_star_propane - p_star_butane)
y_p = x_p * (p_star_propane/p)
p_equal_moles = solve(1 - (1 - 2*y_p)/(2*x_p - 1))

print("Pressure for Equal Liquid and Vapor Mole Fractions:")
print(p_equal_moles) # Positive root is the pressure we seek

# Problem 1.3 Plotting liquid and vapor mole fractions

p_set = np.linspace(5,10,201)

x_p = (p_set - p_star_butane)/(p_star_propane - p_star_butane)
y_p = x_p * (p_star_propane/p_set)
L_by_V = (1 - 2*y_p)/(2*x_p - 1)

liq_mol_frac = 1 / (1 + 1/L_by_V)
vap_mol_frac = 1 - liq_mol_frac

# Plotting the mole fraction data sets
plt.figure(2)
plt.clf()
plt.plot(p_set,liq_mol_frac,color='red',label='Liquid Fraction')
plt.plot(p_set,vap_mol_frac,color='blue',label='Vapor Fraction')
plt.title("Liquid/Vapor Mole Fraction v/s Pressure")
plt.ylabel("Mole Fraction")
plt.xlabel(r"$p$"+" (Pressure) [Bar]")
plt.legend()
ax = plt.gca()
ax.set_ylim([0, 1])
plt.savefig("LiqGasMoleFractionvsPressure_Problem1_3.png")

# Problem 1.4: Plotting mole fraction of propane in liq and vapor as a function of
    ↪ pressure

# Computing pressure limits where vapor and liquid phases coexist
liq_curve = LineString(np.column_stack((p_set,liq_mol_frac)))
vap_curve = LineString(np.column_stack((p_set,vap_mol_frac)))
x_axis = LineString(np.column_stack((p_set,np.zeros(p_set.size))))

pressure_value_low = liq_curve.intersection(x_axis).xy[0][0]
pressure_value_high = vap_curve.intersection(x_axis).xy[0][0]

print('Pressure range for V-L coexistence is '+str(pressure_value_low)+' bar to '+
    ↪ str(pressure_value_high)+' bar')

# Pressure set with values a bit beyond the coexistence regions
p_set_coexistence = np.linspace(pressure_value_low,pressure_value_high,201)
x_p = (p_set_coexistence - p_star_butane)/(p_star_propane - p_star_butane)
y_p = x_p * (p_star_propane/p_set_coexistence)

plt.figure(3)
plt.clf()
```

```
plt.plot(p_set_coexistence,x_p,color='red',label='Liquid Mole Fraction')
plt.plot(p_set_coexistence,y_p,color='blue',label='Vapor Mole Fraction')
plt.plot([p_set_coexistence[0],p_set_coexistence[0]],[x_p[0],y_p[0]],'g--',color='
    ↪ grey')
plt.plot([p_set_coexistence[p_set_coexistence.size-1],p_set_coexistence[
    ↪ p_set_coexistence.size-1]],[x_p[p_set_coexistence.size-1],y_p[
    ↪ p_set_coexistence.size-1]],'g--',color='grey')
plt.plot([5.5,pressure_value_low],[0.5,y_p[0]], color='black',label='Single Phase
    ↪ Region')
plt.plot([9,pressure_value_high],[0.5,x_p[p_set_coexistence.size-1]],color='black'
    ↪ )
plt.title("Liquid/Vapor Mole Fraction v/s Pressure for Propane")
plt.ylabel("Mole Fraction")
plt.xlabel(r"$p$"+" (Pressure) [Bar]")
plt.legend()
ax = plt.gca()
plt.savefig("LiqGasMoleFractionvsPressurePropane_Problem1_4.png")

# Problem 2: Construct T-x phase diagram

# Defining a function to solve for the T values for a given x value
def bubble_and_dew_temps(x1):
    T = var('T')

    # Defining intermediate variables
    p = 101.3 # kPa
    p1_star = exp(16.1952 - (3423.53/(T - 55.7172)))
    p2_star = exp(14.0568 - (2825.42/(T - 42.7089)))

    x2 = 1 - x1

    Gamma12 = 0.0952
    Gamma21 = 0.2713

    gamma_1 = exp(-log(x1 + x2*Gamma12) + x2*(Gamma12/(x1 + x2*Gamma12) - Gamma21
        ↪ /(x2 + x1*Gamma21)))
    gamma_2 = exp(-log(x2 + x1*Gamma21) + x1*(Gamma21/(x2 + x1*Gamma21) - Gamma12
        ↪ /(x1 + x2*Gamma12)))

    bubble_temp_value = nsolve(p - x1*gamma_1*p1_star - x2*gamma_2*p2_star
        ↪ ,(300,500), solver = 'bisect')


    # Redefining to avoid confusion, plot is to be on the same x-axis linespace
    y1 = x1
    y2 = 1 - y1
    dew_temp_value = nsolve(1/p - y1/(gamma_1*p1_star) - y2/(gamma_2*p2_star)
        ↪ ,(300,500), solver = 'bisect')

    return bubble_temp_value, dew_temp_value

# Computing the T value set for the mole fraction set
x1 = np.linspace(0,1,201)
T_bubble = np.zeros(x1.size)
```

```
T_dew = np.zeros(x1.size)

for i in range(x1.size):
    T_bubble[i], T_dew[i] = bubble_and_dew_temps(x1[i])

# Finding the Azeotrope
azeotrope_min = []
azeotrope_avg = []
azeotrope_mole_frac = []

for i in range(x1.size):
    if x1[i] >= 0.3 and x1[i] <= 0.4: # visually, the range we can expect an
        ↪ azeotrope
        azeotrope_min += [abs(T_bubble[i] - T_dew[i])]
        azeotrope_avg += [(T_bubble[i] + T_dew[i])/2]
        azeotrope_mole_frac += [x1[i]]

azeotrope_min = np.asarray(azeotrope_min) # Converting to numpy array
index_min = np.argmin(azeotrope_min) # Finding the index of the minimum value,
    ↪ which we will use for the plot

print('Azeotrope␣coordinates:␣['+str(azeotrope_mole_frac[index_min])+',␣'+str(
    ↪ azeotrope_avg[index_min])+']')

# Plotting the two curves
plt.figure(4)
plt.clf()
plt.plot(x1,T_bubble,color='blue',label='Bubble␣Curve')
plt.plot(x1,T_dew,color='red',label='Dew␣Curve')
plt.plot(azeotrope_mole_frac[index_min],azeotrope_avg[index_min],'ro',color='black
    ↪ ', markersize=5, label="Azeotrope")
plt.title("Bubble/Dew␣Curves␣at␣101.3␣kPa")
plt.xlabel(r"$x_1$"+"␣(Mole␣Fraction␣of␣Ethanol)")
plt.ylabel(r"$T$"+"␣(Temperature)␣[K]")
plt.legend()
ax = plt.gca()
ax.set_xlim([0, 1])
plt.savefig("TempvsMoleFrac_Problem2.png")
```