

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования «Российский химико-технологический университет  
имени Д.И. Менделеева»

Факультет цифровых технологий и химического инжиниринга  
Кафедра информационных компьютерных технологий

### **ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 3**

#### **ПО КУРСУ**

#### **«ЦИФРОВОЕ МОДЕЛИРОВАНИЕ ФИЗИКО-ХИМИЧЕСКИХ СИСТЕМ»:**

#### **«Моделирование бинарных парожидкостных систем с использованием модели Вильсона»**

Ведущий преподаватель

Ст. преподаватель

Скичко Е.А.

**СТУДЕНТ** группы КС-26

Золотухин А.А.

**Москва**

**2024**

## Задание

1. Используя приведенные данные (табл. 1), подобрать константы в модели Вильсона и построить у-х и Р-х диаграмму.

а. Исследуемая система (2 вариант): Ацетон + Хлороформ.

Таблица 1

Вариант 2		
Данные VLE, T	298.15K	
X <sub>1</sub>	Y <sub>1</sub>	P, bar
0.100	0.055	0.2420
0.300	0.275	0.2216
0.500	0.589	0.2287
0.700	0.780	0.2546
0.900	0.951	0.2887
Параметры		
	1	2
A	16.65	15.97
B	2940.46	2696.79
C	-35.93	-46.16
T <sub>c</sub> , K	508.10	536.40
P <sub>c</sub> , bar	47.01	54.72
w	0.31	0.22

Теоретическое обоснование решения.

**Модель Вильсона**<sup>1</sup> хорошо описывает парожидкостное равновесие VLE (Vapour-Liquid Equilibrium) сильно неидеальных систем со специфическим взаимодействием между молекулами компонентов.

Концентрация *i*-ого компонента ( $y_i$ ) в паровой фазе определяется уравнением:

$$y_i = \gamma_i x_i \frac{P_i^0}{P}, \quad (1)$$

где  $\gamma_i$  - коэффициент активности *i*-ого компонента,

$x_i$  - концентрация *i*-ого компонента в жидкости,

$P_i^0$  - давление паров *i*-ого компонента, Па,

$P$  - общее давление в системе, Па.

При этом для компонентов 1 и 2:

$$\ln \gamma_1 = -\ln(x_1 + \Lambda_{12}x_2) + x_2 \left( \frac{\Lambda_{12}}{x_1 + \Lambda_{12}x_2} - \frac{\Lambda_{21}}{x_2 + \Lambda_{21}x_1} \right) \quad (2)$$

$$\ln \gamma_2 = -\ln(x_2 + \Lambda_{21}x_1) - x_1 \left( \frac{\Lambda_{12}}{x_1 + \Lambda_{12}x_2} - \frac{\Lambda_{21}}{x_2 + \Lambda_{21}x_1} \right), \quad (3)$$

где  $\Lambda_{12} = \frac{V_2^L}{V_1^L} e^{-\frac{\lambda_{12} - \lambda_{11}}{RT}},$

$$\Lambda_{21} = \frac{V_1^L}{V_2^L} e^{-\frac{\lambda_{21} - \lambda_{22}}{RT}}.$$

Здесь  $V_1^L$  и  $V_2^L$  - молярные объемы чистых жидких компонентов 1 и 2,

---

<sup>1</sup> первая модель локального состава, применяемая для описания уравнения состояния жидкостей.

$\lambda_{12} - \lambda_{11}$  и  $\lambda_{21} - \lambda_{22}$  - параметр, определяющий разность энергии взаимодействия молекул 1-го и 2-го компонентов друг с другом и между собой, рассчитывается по экспериментальным данным бинарной смеси.

Находим давления паров чистых жидкостей  $P_{vp_1}$  и  $P_{vp_2}$  при  $T$  по уравнению Антуана:

$$\ln P_{vp_1} = A_1 - \frac{B_1}{T - C_1} \quad (4)$$

$$\ln P_{vp_2} = A_2 - \frac{B_2}{T - C_2}, \quad (5)$$

где  $T$  - термодинамическая температура,  $K$ ,

$A_1, B_1, C_1, A_2, B_2, C_2$  - константы, специфичные для вещества, которые зависят от конкретного рассматриваемого вещества.

Предполагается, что имеется несколько экспериментальных точек для смеси при температуре  $T$ . Таких точек пять, т.е. пяти значениям  $x$  соответствуют пять экспериментальных равновесных значений  $y$  и  $P$ . Для каждой точки следует рассчитать  $\gamma_1$  и  $\gamma_2$  из (1), (4) и (5) по уравнениям :

$$\gamma_1 = \frac{y_1 P}{x_1 P_{vp_1}} \quad (6)$$

$$\gamma_2 = \frac{y_2 P}{x_2 P_{vp_2}} \quad (7)$$

Для каждой точки рассчитывают избыточную мольную энергию Гиббса ( $g^E$ ):

$$g^E = RT(x_1 \ln \gamma_1 + x_2 \ln \gamma_2), \quad (8)$$

где  $R$  - универсальная газовая постоянная,  $Дж/(моль * K)$ .

Избыточная энергия Гиббса по Вильсону:

$$g^E = RT(-x_1 \ln(x_1 + \Lambda_{12}x_2) - x_2 \ln(x_2 + \Lambda_{21}x_1)) \quad (9)$$

Далее нужно подобрать константы уравнения (9) таким образом, чтобы минимизировать расхождение между значениями  $g^E$ , рассчитанным по уравнению и определенным по экспериментальным данным.

Затем, используя (6) и (7), нахожу  $\gamma_1$  и  $\gamma_2$  при произвольно выбранных значениях  $x_i$  в диапазоне от  $x_i = 0$  до  $x_i = 1$ .

В конце, для каждого выбранного значения  $x_i$  нахожу соответствующие значения  $y_i$  и  $P$  путем решения уравнений (6) и (7), используя соотношения массового баланса  $x_2 = 1 - x_1$  и  $y_2 = 1 - y_1$ . По полученным результатам нужно построить диаграммы  $y - x$  и  $P - x$ .

Код.

```
import numpy as np
import matplotlib.pyplot as plt

T = 298.15
R_temp = 8.31
X_1 = [0.100, 0.300, 0.500, 0.700, 0.900]
Y_1 = [0.055, 0.275, 0.589, 0.780, 0.951]
P_bar = [0.2420, 0.2216, 0.2287, 0.2546, 0.2887]
P = [i * 1e5 for i in P_bar]
R = 1.987
A_1, A_2 = 16.65, 15.97
B_1, B_2 = 2940.46, 2696.79
```

```
C_1, C_2 = -35.93, -46.16
```

```
Tc_1, Tc_2 = 508.10, 536.40
```

```
Pc_1, Pc_2 = 47.01, 54.72
```

```
w_1, w_2 = 0.31, 0.22
```

```
def gamma_calc(X_1, X_2, Y_1, Y_2, P_vp_1, P_vp_2):
```

```
    gamma_1 = []
```

```
    gamma_2 = []
```

```
    for i in range(len(X_1)):
```

```
        gamma_1.append((Y_1[i] * P[i]) / (X_1[i] * P_vp_1))
```

```
        gamma_2.append((Y_2[i] * P[i]) / (X_2[i] * P_vp_2))
```

```
    return (gamma_1, gamma_2)
```

```
def gibbs_energy_calc(X_1, X_2, gamma_1, gamma_2):
```

```
    g_e = []
```

```
    for i in range(0, len(X_1)):
```

```
        g_e.append(R * T * (X_1[i] * np.log(gamma_1[i]) +  
X_2[i] * np.log(gamma_2[i])))
```

```
    return g_e
```

```
def gamma_Wilson_calc(X_1, X_2, lambda_12, lambda_21):
```

```
    gamma_1 = []
```

```
    gamma_2 = []
```

```
    for i in range(len(X_1)):
```

```

        gamma_1.append(np.exp(-1 * np.log(X_1[i] + lambda_12 *
X_2[i]) + X_2[i] * ((lambda_12 / (X_1[i] + lambda_12 * X_2[i])) -
(lambda_21 / (X_2[i] + lambda_21 * X_1[i])))))

        gamma_2.append(np.exp(-1 * np.log(X_2[i] + lambda_21 *
X_1[i]) - X_1[i] * ((lambda_12 / (X_1[i] + lambda_12 * X_2[i])) -
(lambda_21 / (X_2[i] + lambda_21 * X_1[i])))))

    return (gamma_1, gamma_2)

def gibbs_energy_Wilson_calc(X_1, X_2, lambda_12, lambda_21):

    g_e = []

    for i in range(0, len(X_1)):

        g_e.append(R * T * (-X_1[i] * np.log(X_1[i] +
lambda_12 * X_2[i]) - X_2[i] * np.log(X_2[i] + lambda_21 *
X_1[i])))

    return g_e

def main():

    X_2 = []

    for i in range(0, len(X_1)):

        X_2.append(1 - X_1[i])

    Y_2 = []

    for i in range(0, len(Y_1)):

        Y_2.append(1 - Y_1[i])

    P_vp_1 = np.exp(A_1 - B_1 / (T + C_1))

```

```

P_vp_2 = np.exp(A_2 - B_2 / (T + C_2))

gamma_1, gamma_2 = gamma_calc(X_1, X_2, Y_1, Y_2, P_vp_1,
P_vp_2)

g_e = gibbs_energy_calc(X_1, X_2, gamma_1, gamma_2)

X_1_W = np.arange(0, 1.01, 0.01)
X_2_W = []
for i in range(0, len(X_1_W)):
    X_2_W.append(1 - X_1_W[i])

X_1_W = np.arange(0, 1.01, 0.01)
X_2_W = []
for i in range(0, len(X_1_W)):
    X_2_W.append(1 - X_1_W[i])

alpha_12, alpha_21 = 116, -506
Z_RA_1 = 0.29056 - 0.08775 * w_1
Z_RA_2 = 0.29056 - 0.08775 * w_2
Tr_1 = T / Tc_1
Tr_2 = T / Tc_2
V_L_1 = (R_temp * Tc_1 / Pc_1) * Z_RA_1 ** (1 + (1 - Tr_1)
** (2 / 7))

```



```

V_L_2 = (R_temp * Tc_2 / Pc_2) * Z_RA_2 ** (1 + (1 - Tr_2)
** (2 / 7))

lambda_12 = V_L_2 / V_L_1 * np.exp(-1 * alpha_12 / (R *
T))

lambda_21 = V_L_1 / V_L_2 * np.exp(-1 * alpha_21 / (R *
T))

gamma_1_W, gamma_2_W = gamma_Wilson_calc(X_1_W, X_2_W,
lambda_12, lambda_21)

Y_1_W = [0]

for i in range(1, len(X_1_W)):
    Y_1_W.append(1 / ((gamma_2_W[i] * X_2_W[i] * P_vp_2) /
(gamma_1_W[i] * X_1_W[i] * P_vp_1) + 1))

Y_2_W = []

for i in range(0, len(Y_1_W)):
    Y_2_W.append(1 - Y_1_W[i])

plt.figure

plt.grid(True)

plt.title("Ацетон (y-x)")

plt.plot(X_1_W, Y_1_W, color='blue')

plt.scatter(X_1, Y_1, color='orange')

plt.xlabel("X")

plt.ylabel("Y")

plt.show()

```

```

plt.figure

plt.grid(True)

plt.title("Хлороформ (y-x)")

plt.plot(X_2_W, Y_2_W, color='blue')

plt.scatter(X_2, Y_2, color='orange')

plt.xlabel("X")

plt.ylabel("Y")

plt.show()

P_1_W = []

for i in range(0, len(X_1_W)):

    P_1_W.append(gamma_1_W[i] * X_1_W[i] * P_vp_1)

P_2_W = []

for i in range(0, len(X_2_W)):

    P_2_W.append(gamma_2_W[i] * X_2_W[i] * P_vp_2)

P_W = []

for i in range(0, len(P_1_W)):

    P_W.append((P_1_W[i] + P_2_W[i]) / 750.1)

plt.figure

plt.grid(True)

```

```

plt.title("Ацетон-Хлороформ (P-y-x)")

plt.plot(X_1_W, P_W, color="blue")

plt.plot(Y_1_W, P_W, color="orange")

plt.scatter(X_1, P_bar, color='orange')

plt.scatter(Y_1, P_bar, color='orange')

plt.xlabel("X, Y")

plt.ylabel("Давление (P), бар")

plt.show()

g_e_W = gibbs_energy_Wilson_calc(X_1_W, X_2_W, lambda_12,
lambda_21)

main()

```

Результаты расчетов.



