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 № 14**

**ПО КУРСУ**

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

**«Построение и анализ диаграмм фазового равновесия»**

Ведущий преподаватель

Ст. преподаватель Скичко Е.А.

**СТУДЕНТ группы КС-20** Мелехин А.А.

**Москва**

**2024**

# **Задание**

**Используя:** модель Пенга-Робинсона для газовой фазы и модель *modified UNIFAC (Dortmund)* для жидкой фазы *binary\_PR\_UNIFAC\_mod.py*, модель Пенга-Робинсона для газовой и 1-2 жидких фаз *binary\_PR.py* найти с помощью регрессии на листах 1 или 2 книги BIPJ xlsx по экспериментальным данным *Dortmund Data Base* *http://www.ddbst.com/en/EED/VLE/VLEindex.php*),

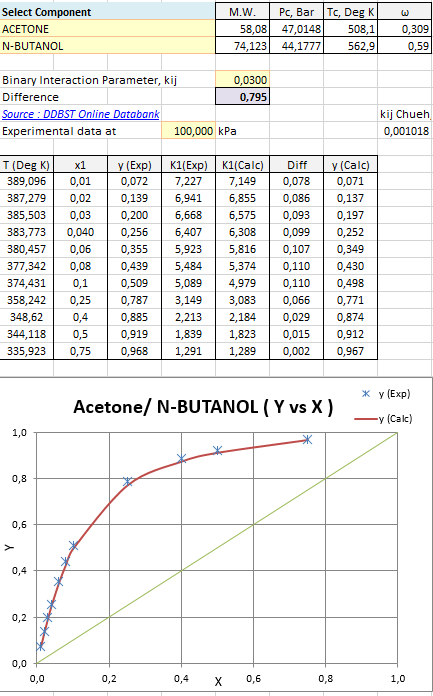
**1)** построить **y-x** диаграмму для первого компонента при T = 333.15 K.

**2)** построить **P-x** диаграмму при T = 333.15 K, отметить на ней все фазы и чистые компоненты

**3)** построить **T-x** диаграмму при P = 1 бар, отметить на ней все фазы и чистые компоненты. Определите тип диаграммы из изученных и отклонение от идеальности (например, «диаграмма кипения с азеотропом с положительным отклонением от закона Рауля»),

**4)** Поставьте фигуративную точку *D* на *T-x* диаграмме в двухфазной области. Определите равновесные составы фаз, определяемые фигуративной точкой *D*, определите долю каждой фазы по правилу рычага. Проанализируйте фазовое состояние системы: *ацетон – 1-бутанол (н-бутанол)* на основании диаграммы кипения при P = 100000 Па, T= 333,15 K. Проведите анализ процесса нагревания системы с молярной долей 0.5.

**5)** Рассчитайте среднюю относительную ошибку определения мольной доли а) в газовой фазе и б) в жидкой фазе для каждой из моделей по пяти парам (P,T) из экспериментальных данных базы Dortmund Data Base *http://www.ddbst.com/en/EED/VLE/VLEindex.php* (необходимо провести flash-расчеты с обеими моделями при этих пяти парах (P,T), затем посчитать относительную ошибку для каждой модели)



**Код**

**программа binary\_PR.py**

from thermo import ChemicalConstantsPackage, CEOSGas, CEOSLiquid, PRMIX, FlashVL, FlashVLN, FlashPureVLS

from thermo.interaction\_parameters import IPDB

constants, properties = ChemicalConstantsPackage.from\_IDs(['acetone', '1-butanol']) # данные вещества

T = 333.15 # температура

P = 1e5 # давление

zs = [.5, .5] # мольные доли

k12 = 0.03 # коэффициент из excel

kijs = [[0, k12], [k12, 0]]

eos\_kwargs = dict(Tcs=constants.Tcs, Pcs=constants.Pcs, omegas=constants.omegas, kijs=kijs)

gas = CEOSGas(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

liquid = CEOSLiquid(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

flasher = FlashVL(constants, properties, liquid=liquid, gas=gas)

\_ = flasher.plot\_Txy(P=P, pts=100) # создание Tx-y диаграммы

\_ = flasher.plot\_Pxy(T=T, pts=100) # создание Px-y диаграммы

\_ = flasher.plot\_xy(T=T, pts=100) # создание x-y диаграммы

liquid2 = CEOSLiquid(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

flasher2 = FlashVLN(constants, properties, liquids=[liquid, liquid2], gas=gas)

res = flasher2.flash(T=T, P=P, zs=zs)

print('There are %s phases present at %f K and %f bar' %(res.phase\_count,T,P/1e5))

if res.VF > 0:

    print(res.gas.zs)

if res.VF == 1:

    print("Only vapour")

else:

    print("Liquid0: ")

    print(res.liquid0.zs)

    if res.liquid\_count>1:

        print("LIQUID PHASE SEPARATION")

        print("Liquid1: ")

        print(res.liquid1.zs)

**Программа PR\_UNIFAC.py**

#############

# UNIFAC + PR

#############

from thermo import \*

from thermo.unifac import DOUFSG, DOUFIP2016

constants, properties = ChemicalConstantsPackage.from\_IDs(['acetone', '1-butanol']) # данные вещества

T = 333.15 # температура

P = 1e5 # давление

zs = [.5, .5] # мольные доли

k12 = 0.03 # коэффициент из excel

kijs = [[0, k12], [k12, 0]]

eos\_kwargs = dict(Tcs=constants.Tcs, Pcs=constants.Pcs, omegas=constants.omegas) #, kijs=kijs

gas = CEOSGas(PRMIX, HeatCapacityGases=properties.HeatCapacityGases, eos\_kwargs=eos\_kwargs)

GE = UNIFAC.from\_subgroups(chemgroups=constants.UNIFAC\_Dortmund\_groups, version=1, T=T, xs=zs, interaction\_data=DOUFIP2016, subgroups=DOUFSG)

# Configure the liquid model with activity coefficients

liquid = GibbsExcessLiquid(

    VaporPressures=properties.VaporPressures,

    HeatCapacityGases=properties.HeatCapacityGases,

    VolumeLiquids=properties.VolumeLiquids,

    GibbsExcessModel=GE,

    equilibrium\_basis='Psat', caloric\_basis='Psat',

    T=T, P=P, zs=zs)

# Create a flasher instance, assuming only vapor-liquid behavior

flasher = FlashVL(constants, properties, liquid=liquid, gas=gas)

\_ = flasher.plot\_Txy(P=P, pts=100) # создание Tx-y диаграммы

\_ = flasher.plot\_Pxy(T=T, pts=100) # создание Px-y диаграммы

\_ = flasher.plot\_xy(T=T, pts=100) # создание x-y диаграммы

# VLLE flash

liquid2 = GibbsExcessLiquid(

    VaporPressures=properties.VaporPressures,

    HeatCapacityGases=properties.HeatCapacityGases,

    VolumeLiquids=properties.VolumeLiquids,

    GibbsExcessModel=GE,

    equilibrium\_basis='Psat', caloric\_basis='Psat',

    T=T, P=P, zs=zs)

flasher2 = FlashVLN(constants, properties, liquids=[liquid, liquid2], gas=gas)

x1\_exp=[0.01, 0.03, 0.1, 0.5, 0.85]

y1\_exp=[0.250369, 0.494505, 0.725424, 0.839958, 0.908808]

myT=[365.699, 355.741, 342.005, 332.748, 330.14]

zs=[0.1, 0.2, 0.7, 0.8, 0.87]

for i in range(5):

    res = flasher2.flash(T=myT[i], P=P, zs=[zs[i], 1-zs[i]])

    print('There are %s phases present at %f K and %f bar' %(res.phase\_count,myT[i],P/1e5))

    if res.VF > 0:

        print("x: ")

        print(res.gas.zs)

    if res.VF == 1:

        print("Only vapour")

    else:

        print("Liquid0: ")

        print(res.liquid0.zs)

        if res.liquid\_count>1:

            print("LIQUID PHASE SEPARATION")

            print("Liquid1: ")

            print(res.liquid1.zs)

**Программа main.py**

# http://vle-calc.com/phase\_diagram.html

from thermo import \*

from thermo.unifac import DOUFSG, DOUFIP2016

from thermo import ChemicalConstantsPackage, CEOSGas, CEOSLiquid, PRMIX, FlashVL, FlashVLN, FlashPureVLS

from thermo.interaction\_parameters import IPDB

def calc2(P, T, z):

    constants, properties = ChemicalConstantsPackage.from\_IDs(['acetone', '1-butanol']) # данные вещества

    T = 333.15

    P = 1e5

    zs = [z, 1-z]

    k12 = 0.03

    kijs = [[0, k12], [k12, 0]]

    eos\_kwargs = dict(Tcs=constants.Tcs, Pcs=constants.Pcs, omegas=constants.omegas) #, kijs=kijs

    gas = CEOSGas(PRMIX, HeatCapacityGases=properties.HeatCapacityGases, eos\_kwargs=eos\_kwargs)

    GE = UNIFAC.from\_subgroups(chemgroups=constants.UNIFAC\_Dortmund\_groups, version=1, T=T, xs=zs, interaction\_data=DOUFIP2016, subgroups=DOUFSG)

    liquid = GibbsExcessLiquid(

    VaporPressures=properties.VaporPressures,

    HeatCapacityGases=properties.HeatCapacityGases,

    VolumeLiquids=properties.VolumeLiquids,

    GibbsExcessModel=GE,

    equilibrium\_basis='Psat', caloric\_basis='Psat',

    T=T, P=P, zs=zs)

    liquid2 = GibbsExcessLiquid(

        VaporPressures=properties.VaporPressures,

        HeatCapacityGases=properties.HeatCapacityGases,

        VolumeLiquids=properties.VolumeLiquids,

        GibbsExcessModel=GE,

        equilibrium\_basis='Psat', caloric\_basis='Psat',

        T=T, P=P, zs=zs)

    flasher = FlashVLN(constants, properties, liquids=[liquid, liquid2], gas=gas)

    res = flasher.flash(T=T, P=P, zs=zs)

    ans = [[0.0, 0.0], [0.0, 0.0]]

    if res.VF > 0:

        ans[1] = res.gas.zs

    if res.VF == 1:

        pass

    else:

        ans[0] = res.liquid0.zs

    return ans

def calc1(P, T, z):

    constants, properties = ChemicalConstantsPackage.from\_IDs(['acetone', '1-butanol']) # данные вещества

    zs = [z, 1-z]

    k12 = 0.03

    kijs = [[0, k12],

            [k12, 0]]

    eos\_kwargs = dict(Tcs=constants.Tcs, Pcs=constants.Pcs, omegas=constants.omegas, kijs=kijs)

    gas = CEOSGas(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

    liquid = CEOSLiquid(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

    liquid2 = CEOSLiquid(PRMIX, eos\_kwargs, HeatCapacityGases=properties.HeatCapacityGases, T=T, P=P, zs=zs)

    flasher = FlashVLN(constants, properties, liquids=[liquid, liquid2], gas=gas)

    res = flasher.flash(T=T, P=P, zs=zs)

    ans = [[0.0, 0.0], [0.0, 0.0]]

    if res.VF > 0:

        ans[1] = res.gas.zs

    if res.VF == 1:

        pass

    else:

        ans[0] = res.liquid0.zs

    return ans

T\_e = [116.027, 106.992, 100.954, 89.1905, 67.0128]

P\_e = [1.0, 1.0, 1.0, 1.0, 1.0]

T\_e = [round(num+273.15, 2) for num in T\_e] # перевод цельсий - кельвин

P\_e = [round(num\*1e5, 2) for num in P\_e] # перевод bar - Pa

x1\_e = [0.008, 0.06, 0.1, 0.2, 0.6]

y1\_e = [0.0585364, 0.356559, 0.510315, 0.728513, 0.943063]

zs = [(y1\_e[i]+x1\_e[i])/2 for i in range(len(x1\_e))]

x2\_e = [1-num for num in x1\_e]

y2\_e = [1-num for num in y1\_e]

# ошибка для модели binary\_PR.py

x1\_r = []

x2\_r = []

y1\_r = []

y2\_r = []

error\_x1 = 0

error\_y1 = 0

error\_x2 = 0

error\_y2 = 0

for i in range(len(P\_e)):

    r\_info = calc1(P\_e[i], T\_e[i], zs[i])

    x1\_r.append(r\_info[0][0])

    y1\_r.append(r\_info[1][0])

    x2\_r.append(r\_info[0][1])

    y2\_r.append(r\_info[1][1])

    error\_x1 += (abs(x1\_e[i]-x1\_r[i])/x1\_e[i])\*100

    error\_y1 += (abs(y1\_e[i]-y1\_r[i])/y1\_e[i])\*100

    error\_x2 += (abs(x2\_e[i]-x2\_r[i])/x2\_e[i])\*100

    error\_y2 += (abs(y2\_e[i]-y2\_r[i])/y2\_e[i])\*100

# вывод средней отсночительной ошибки в процентах

print("error for model binary\_PR.py.py")

print("average error for x1:", round(error\_x1/5.0, 2))

print("average error for y1:", round(error\_y1/5.0, 2))

print("average error for x2:", round(error\_x2/5.0, 2))

print("average error for y2:", round(error\_y2/5.0, 2))

# ошибка для модели PR\_UNIFAC.py

x1\_r = []

x2\_r = []

y1\_r = []

y2\_r = []

error\_x1 = 0

error\_y1 = 0

error\_x2 = 0

error\_y2 = 0

for i in range(len(P\_e)):

    r\_info = calc2(P\_e[i], T\_e[i], zs[i])

    x1\_r.append(r\_info[0][0])

    y1\_r.append(r\_info[1][0])

    x2\_r.append(r\_info[0][1])

    y2\_r.append(r\_info[1][1])

    error\_x1 += (abs(x1\_e[i]-x1\_r[i])/x1\_e[i])\*100

    error\_y1 += (abs(y1\_e[i]-y1\_r[i])/y1\_e[i])\*100

    error\_x2 += (abs(x2\_e[i]-x2\_r[i])/x2\_e[i])\*100

    error\_y2 += (abs(y2\_e[i]-y2\_r[i])/y2\_e[i])\*100

# вывод средней отсночительной ошибки в процентах

print("\nerror for model PR\_UNIFAC.py")

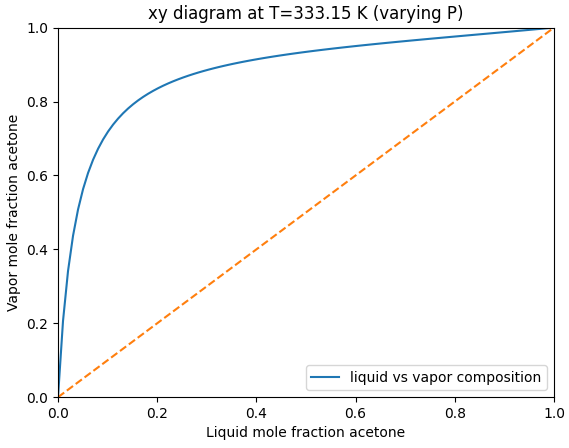
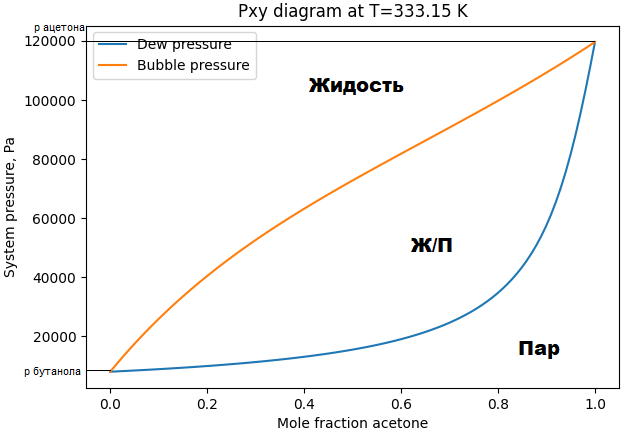
print("average error for x1:", round(error\_x1/5.0, 2))

print("average error for y1:", round(error\_y1/5.0, 2))

print("average error for x2:", round(error\_x2/5.0, 2))

print("average error for y2:", round(error\_y2/5.0, 2))

**Результаты расчетов**

**Задания 1 – 3 для модели binary\_PR.py******

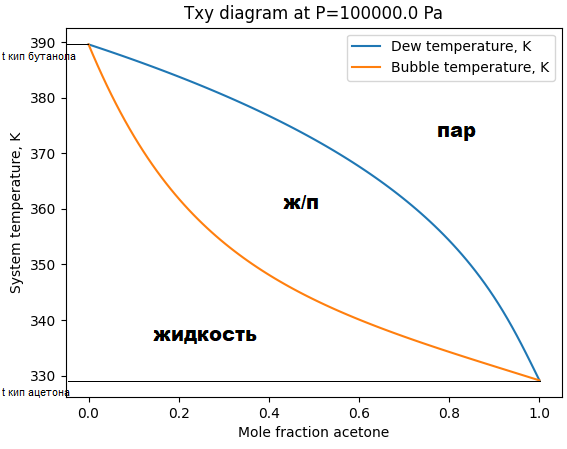
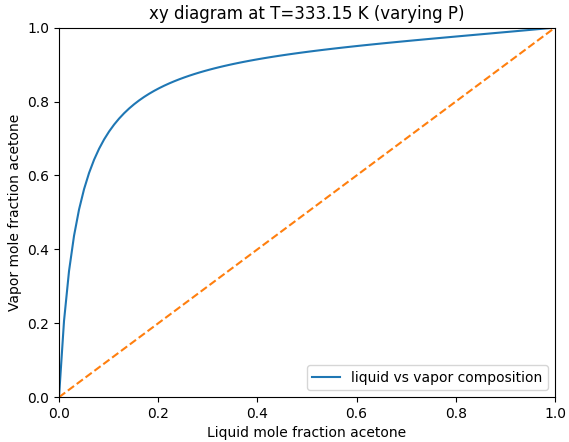
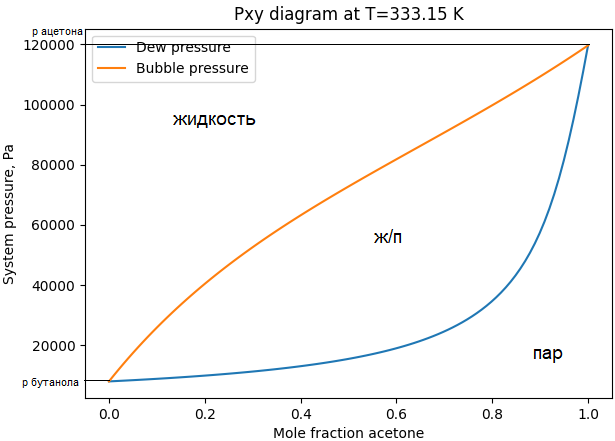
****

Диаграмма кипения бинарной системы *ацетон – 1-бутанол (н-бутанол)* без азеотропа с положительным отклонением от закона Рауля.

**Задания 1 - 3 для модели PR\_UNIFAC.py**

****

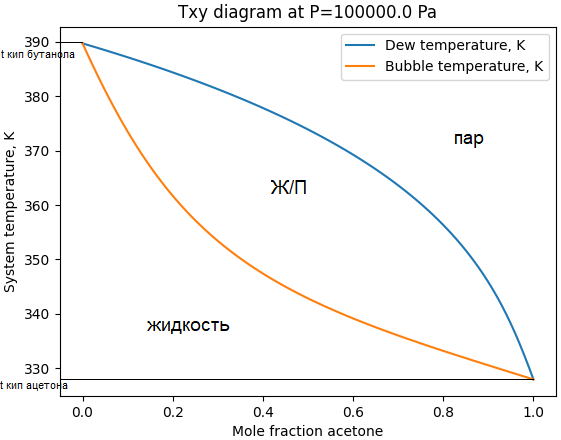
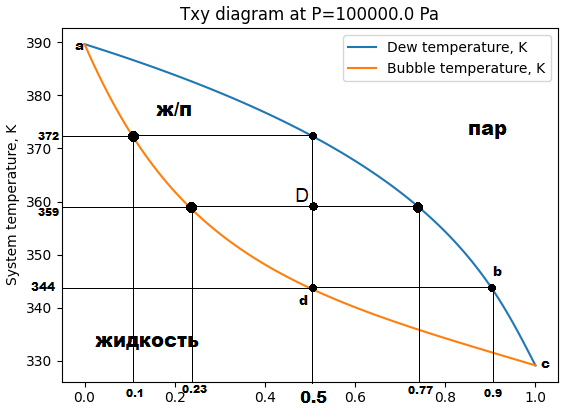
****

Диаграмма кипения бинарной системы *ацетон – 1-бутанол (н-бутанол)* без азеотропа с положительным отклонением от закона Рауля.

**Задание 4**



**Анализ:** выше кривой abc состава насыщенного пара все системы находятся в состоянии пара. Системы гомогенные, однофазные. Ниже кривой аdс состава кипящей жидкости все системы находятся в жидком состоянии. Системы гомогенные, однофазные. Между кривыми *аbс* и *аdс* система гетерогенная, две фазы, жидкость и пар. Для определения состава равновесных фаз через точку заданного состояния необходимо провести изотерму, так как фазы, находящиеся в равновесии, должны иметь одинаковую температуру. Пересечение изотермы с кривой adc состава кипящей жидкости дает состав жидкой фазы, который определяется по оси абсцисс. Пересечение изотермы с кривой abc насыщенного пара дает состав пара. Так, система с молярным содержанием 0.5 при 359 К — гетерогенная точка. Она содержит две фазы, находящиеся в равновесии. Одна фаза — кипящая жидкость с молярным содержанием 0.23, другая — насыщенный пар с молярным содержанием 0.77. Если жидкую систему с молярным содержанием 0.5 (точка *d*) нагреть до 344 К, то начнется кипение системы. Молярный состав пара, находящегося в равновесии с кипящей жидкостью, будет 0.9 Состав пара беднее *бутанолом*, чем жидкость. Из жидкой фазы в пар преимущественно уходит *ацетон*. Отсюда жидкая фаза обедняется *ацетоном*, и ее состав меняется, как это показано стрелками на графике. Вместе с изменением состава кипящей жидкости меняется и состав пара, находящегося в равновесии с ней. Изменение состава пара показано стрелками на кривой abc. При 359 К в равновесии будут находиться нар и жидкость. Молярные составы жидкой фазы 0.23 и пара 0.77. Изменение состава жидкости и пара и температуры фазового равновесия будет происходить до тех пор, пока состав пара не станет равным состав: у, исходной жидкости. При 372 К молярный состав пара будет 0.5, вся жидкая фаза превратится в пар. Система станет гомогенной. Молярный состав последней капли жидкости 0.1. При дальнейшем нагревании состав пара меняться не будет.

**Определение долей по правилу рычага:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Температура, К** | **Доля вещества 1 в жидкости** | **Доля вещества 2 в жидкости** | **Доля вещества 1 в паре** | **Доля вещества 2 в паре** |
| 344 | 0.5 | 0.5 | 0.9 | 0.1 |
| 359 | 0.23 | 0.77 | 0.77 | 0.23 |
| 372 | 0.1 | 0.9 | 0.5 | 0.5 |

***Задание 5***

error for model binary\_PR.py.py

average error for x1: 24.1

average error for y1: 14.32

average error for x2: 20.54

average error for y2: 11.36

error for model PR\_UNIFAC.py

average error for x1: 185.77

average error for y1: 100.0

average error for x2: 23.41

average error for y2: 100.0