

**TOMSK
POLYTECHNIC
UNIVERSITY**



**ТОМСКИЙ
ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ**

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное
образовательное учреждение высшего образования
«Национальный исследовательский Томский политехнический университет» (ТПУ)

Инженерная школа природных ресурсов
Направление подготовки Химическая технология
Отделение химической инженерии

PYTHON ДЛЯ ЗАДАЧ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

Отчет по лабораторной работе № 1

Выполнил студент гр. 9дм21

(Подпись)

Шуриков М К

_____ 2023 г.

Отчет принят:

Преподаватель
доцент ОХИ ИШПР, к.т.н.

(Подпись)

В.А. Чузлов

_____ 2023 г.

Томск 2023 г.

Задание 1

Используя исходные данные из примера, рассчитайте, реализовав соответствующие функции:

Состав потока в мольных долях:

$$\chi_i = \frac{\frac{\omega_i}{M_i}}{\sum_{i=1}^n \frac{\omega_i}{M_i}}$$

где χ_i - мольная доля i -го компонента; ω_i - массовая доля i -го компонента; M_i - молярная масса i -го компонента; n - число компонентов в системе; i - индекс компонента в системе.

Плотность потока:

$$\rho = \frac{1}{\sum_{i=1}^n \frac{\omega_i}{\rho_i}}$$

где ρ - плотность потока; ω_i — массовая доля i -го компонента; ρ_i — плотность i -го компонента; n — число компонентов в системе; i — индекс компонента в системе.

Среднюю молекулярную массу потока:

$$m = \frac{1}{\sum_{i=1}^n \frac{\omega_i}{M_i}}$$

где m — средняя молекулярная масса потока; ω_i — массовая доля i -го компонента; M_i — молярная масса i -го компонента; n — число компонентов в системе; i — индекс компонента в системе.

Исходные данные:

Параметр	C ₁	C ₂	C ₃	iC ₄	nC ₄	iC ₅
ω_i	0.1	0.1	0.1	0.4	0.2	0.05
ρ_i , г/см ³	0.416	0.546	0.585	0.5510	0.6	0.616
M_i , г/моль	16	30	44	58	58	72

Программная реализация:

```

def mass_to_mol_fractions(
    mass_fractions: list[float],
    mol_weight: list[float],
) -> list[float]:

    mf_by_mw = [mf / mw for mf, mw in zip(mass_fractions, mol_weight)]
    s = sum(mf_by_mw)

    return [x / s for x in mf_by_mw]

def density(
    mass_fractions: list[float],
    density_list: list[float],
) -> list[float]:

    dens = [mf / rho for mf, rho in zip(mass_fractions, density_list)]
    s = sum(dens)

    return [1 / s]

def average_mass(
    mass_fractions: list[float],
    mol_weight: list[float],
) -> list[float]:

    mf_by_mw = [mf / mw for mf, mw in zip(mass_fractions, mol_weight)]
    s = sum(mf_by_mw)

    return [1 / s]

mf = [.1, .1, .1, .4, .2, .05, .03, .02]
mw = [16, 30, 44, 58, 58, 72, 72, 86]
density_list = [0.416, 0.546, 0.585, 0.5510, 0.6, 0.616]
#nm = ['C1', 'C2', 'C3', 'iC4', 'nC4', 'iC5', 'nC5', 'C6']

vf = mass_to_mol_fractions(
    mass_fractions=mf, mol_weight=mw)

print('Состав потока в мольных долях')

for v in vf:
    print(f'{v:6.4f}')

print(f'Sum: {(sum(vf)):0.1f}')
print()

rho = density(
    mass_fractions=mf, density_list=density_list)

```

```

print('Плотность потока')
for r in rho:
    print(f'{r:6.4f}')
print()

a_mass = average_mass(
    mass_fractions=mf, mol_weight=mw)

print('Средняя масса потока')
for a in a_mass:
    print(f'{a:6.4f}')

```

Ответ:

Состав потока в мольных долях

```

0.2655
0.1416
0.0965
0.2929
0.1465
0.0295
0.0177
0.0099
Sum: 1.0

```

Плотность потока

```
0.5764
```

Средняя масса потока

```
42.4727
```

Задание 2

Пусть на смешение поступают материальные потоки следующего состава (массовые доли):

Поток	C ₁	C ₂	C ₃	iC ₄	nC ₄	iC ₅
1	0.1	0.1	0.1	0.4	0.2	0.05
2	0.1	0.2	0.1	0.3	0.1	0.15
3	0.1	0.1	0.15	0.35	0.1	0.05

Расходы потоков 200, 250 и 120 кг/ч, соответственно. Необходимо рассчитать состав итогового потока в массовых долях, реализовав соответствующую функцию.

Состав смесового потока можно найти следующим образом:

$$\omega_i = \frac{\sum_{j=1}^n G_j \omega_{i,j}}{\sum_{j=1}^n G_j}$$

где ω_i — массовая доля i -го компонента в смешанном потоке; $\omega_{i,j}$ — массовая доля i -го компонента в j -ом потоке; G_j — массовый расход j -го потока; j — индекс потока; i — индекс компонента в системе; n — число потоков, подаваемых на смешение.

Программная реализация:

```
def mass_fractions_from_G(
    mass_fraction: list[float],
    flow_rate: list[float],
) -> list[float]:

    mf_by_fr = [mf * fr for mf, fr in zip(mass_fraction, flow_rate)]
    s = sum(mf_by_fr)

    return [x / s for x in mf_by_fr]

mf = [.1, .1, .1, .4, .2, .05, .03, .02]
G = [16, 30, 44, 58, 58, 72, 72, 86]
nm = ['C1', 'C2', 'C3', 'iC4', 'nC4', 'iC5', 'nC5', 'C6']

vf = mass_fractions_from_G(
    mass_fraction=mf, flow_rate=G)

for v, name in zip(vf, nm):
    print(f'ω({name:^3s}): {v:6.4f}')
```

Ответ:

```
ω(C1 ): 0.0312
ω(C2 ): 0.0585
ω(C3 ): 0.0858
ω(iC4): 0.4524
ω(nC4): 0.2262
ω(iC5): 0.0702
ω(nC5): 0.0421
ω(C6 ): 0.0335
```

Задание 3

Абсолютная плотность газов и паров [кг/м³] вычисляется по формуле:

$$\rho = \frac{M}{22.4} \cdot \frac{T_0 \cdot P}{T \cdot P_0}$$

где M — молярная масса газа или пара, [кг/кмоль];

$T_0 = 273.15$ — нормальная температура, [K];

T — температура, при которой определяется плотность, [K];

$P_0 = 101325$ — нормальное давление, [Па];

P — давление, при котором определяется плотность, [Па].

Необходимо реализовать функцию для определения плотности метана (CH_4) при $P = 200$ кПа и температуре $T \in [200; 500]$ с шагом $h = 50$ [K].

Программная реализация:

```
T = [i for i in range(200,501) if i%50 == 0]

def density(
    temp: list[float],
    mw: float,
    p: float,
) -> list[float]:

    rho = [mw / 22.4 * (273.15 * p) / (t * 101325) for t in temp]

    return rho

t = [i for i in range(200,501) if i%50 == 0]
p = 200e3
mw = 16

rho = density(mw = mw, p = p, temp = t)
for t, r in zip(t, rho):
    print(f'T = {t} K, ρ = {r:6.4f} kg/m3')
```

Ответ:

$T = 200$ K, $\rho = 1.9256$ kg/m³
 $T = 250$ K, $\rho = 1.5404$ kg/m³
 $T = 300$ K, $\rho = 1.2837$ kg/m³
 $T = 350$ K, $\rho = 1.1003$ kg/m³
 $T = 400$ K, $\rho = 0.9628$ kg/m³
 $T = 450$ K, $\rho = 0.8558$ kg/m³
 $T = 500$ K, $\rho = 0.7702$ kg/m³

Задание 4

Коэффициент сжимаемости учитывает отклонение реального газа от уравнения состояния идеального газа. При точных расчетах коэффициент сжимаемости определяют по формуле:

$$z = 1 + \frac{P_r}{T_r} \left(0.144 + 0.073\omega - \frac{0.33 - 0.46\omega}{T_r} - \frac{0.138 + 0.5\omega}{T_r^2} - \frac{0.012 + 0.097\omega}{T_r^3} - \frac{0.0073\omega}{T_r^8} \right)$$

где ω — ацентрический фактор, вычисляемый по уравнению:

$$\omega = \frac{3}{7} \left(\frac{\lg P_r - 5}{\frac{T}{T_r - T_b} - 1} \right) - 1$$

T_r — приведенная температура: $T_r = \frac{T}{T_c}$

$P_r = 0.2634$ — приведенное давление; $T_b = 272.65$ — температура кипения, [K]; $T_c = 425.15$, [K].

Необходимо реализовать функции для расчёта фактора сжимаемости z при $T \in [200; 400]$ с шагом $h = 25$ [K].

Программная реализация:

```
def reduced_temperature(
    temp_list: list[float],
    temp_critical: float,
) -> list[float]:

    Tr = [T / temp_critical for T in temp_list]

    return Tr

T = [float(t) for t in range(200, 401) if t % 25 == 0]
Tc = 425.15
```

```
Tr = reduced_temperature(temp_list = T, temp_critical = Tc)

for Tr_, T_ in zip(Tr, T):
    print(f'For T = {T_} K, Tr = {Tr_:6.4f}')
```

```
For T = 200.0 K, Tr = 0.4704
For T = 225.0 K, Tr = 0.5292
For T = 250.0 K, Tr = 0.5880
For T = 275.0 K, Tr = 0.6468
For T = 300.0 K, Tr = 0.7056
For T = 325.0 K, Tr = 0.7644
For T = 350.0 K, Tr = 0.8232
For T = 375.0 K, Tr = 0.8820
For T = 400.0 K, Tr = 0.9408
```

```
import math

def acentric_factor(
    Tr: list[float],
    temp_list: list[float],
    boiling_point: float,
    press_reduced: float,
) -> list[float]:

    w = [3 / 7 * ( (math.log(press_reduced, 10) - 5) / ( T / ( Tr * boiling_point )
    - 1 ) ) - 1 for T, Tr in zip(temp_list, Tr)]

    return w

Tb = 272.65
Pr = 0.2634
w = acentric_factor(Tr = Tr, temp_list = T, boiling_point = Tb, press_reduced =
Pr)
for w_, t_ in zip(w, T):
    print (f'For T = {t_} K, w = {w_:6.4f}')
```

```
For T = 200.0 K, w = -5.2751
For T = 225.0 K, w = -5.2751
For T = 250.0 K, w = -5.2751
For T = 275.0 K, w = -5.2751
For T = 300.0 K, w = -5.2751
For T = 325.0 K, w = -5.2751
For T = 350.0 K, w = -5.2751
For T = 375.0 K, w = -5.2751
For T = 400.0 K, w = -5.2751
```



```
def compressibility_coeff(
    press_reduced: float,
    Tr: list[float],
    w: list[float],
) -> list[float]:

    z = [1 + ( press_reduced / Tr ) * ( 0.144 + 0.073 * w - (0.33 - 0.46 * w) /
Tr - (0.138 + 0.5 * w) / Tr ** 2 - (0.012 + 0.097 * w) / Tr ** 3 - (0.0073 * w) /
Tr ** 8 ) for w, Tr in zip(w, Tr)]

    return z

z = compressibility_coeff(Tr = Tr, w = w, press_reduced = Pr)
for z_ , t_ in zip(z, T):
    print (f'For T = {t_} K, z = {z_:6.4f}')
```

Ответ:

```
For T = 200.0 K, z = 15.5863
For T = 225.0 K, z = 7.5219
For T = 250.0 K, z = 4.3377
For T = 275.0 K, z = 2.8628
For T = 300.0 K, z = 2.0904
For T = 325.0 K, z = 1.6475
For T = 350.0 K, z = 1.3765
For T = 375.0 K, z = 1.2030
For T = 400.0 K, z = 1.0883
```

Задание 5

Реализуйте функцию, возвращающую словарь, в котором ключами будут имена C1-C5, а значениями другой словарь, содержащий молекулярную массу, температуру и плотность соответствующих алканов. Общая формула для алканов: C_nH_{2n+2} .

Температуру кипения можно определить по следующей формуле:

$$T_b = 1090 - \exp(6.9955 - 0.11193 \cdot N_C^{\frac{2}{3}})$$

где N_C — число атомов углерода в молекуле алкана.

Формула для вычисления плотности:

$$\rho = 1.07 - \exp(3.56073 - 2.93886 \cdot MW^{0.1})$$

где MW — молекулярная масса алкана.

Программная реализация:

```
#boiling point
import math
def boiling_point(
    carbon: list[float],
) -> list[float]:

    bp = [1090 - math.e ** (6.9955 - 0.11193 * n ** (2/3)) for n in carbon]

    return bp
#boiling point output
carbon = [i for i in range(1,6)]
bp = boiling_point(carbon = carbon)
for bp_ in bp:
    print (f'bp = {bp_:6.4f} K')
```

```
bp = 113.8952 K
bp = 176.0077 K
bp = 225.0461 K
bp = 266.5901 K
bp = 303.0107 K
```

```
#density
import math
def density(
    molecular_weight: list[float],
) -> list[float]:

    d = [1.07 - math.e ** (3.56073 - 2.93886 * mw ** 0.1) for mw in molecular_weight]

    return d
#density output
molecular_weight = [12.011 * i + 1.00784 * (2 * i + 2) for i in range(1,6)]
d = density(molecular_weight = molecular_weight)
for d_ in d:
    print (f'density = {d_:6.4f} g/cm3')
```

```
density = 0.3425 g/cm3
density = 0.5043 g/cm3
density = 0.5885 g/cm3
density = 0.6430 g/cm3
density = 0.6823 g/cm3
```

```
compound = ['C1', 'C2', 'C3', 'C4', 'C5']
keys_2 = ['mw', 'tb', 'rho']
tb = [-180, 50, 200, 300, 350]
rho = [220, 300, 320, 400, 450]
mw = [16, 20, 30, 40, 50]

properties = list(zip(mw, tb, rho))
results = {}
for i, name in enumerate(compound):
    d = dict(zip(keys_2, properties[i]))
    results[name] = d

results
```

Ответ:

```
{'C1': {'mw': 16, 'tb': -180, 'rho': 220},
'C2': {'mw': 20, 'tb': 50, 'rho': 300},
'C3': {'mw': 30, 'tb': 200, 'rho': 320},
'C4': {'mw': 40, 'tb': 300, 'rho': 400},
'C5': {'mw': 50, 'tb': 350, 'rho': 450}}
```