

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ**

Федеральное государственное автономное образовательное  
учреждение высшего образования  
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ  
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**




Инженерная школа природных ресурсов  
Направление подготовки 18.04.01 «Химическая технология»  
Образовательная программа «Химическая технология подготовки нефти и газа»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2**

По дисциплине
<b>РУТНОН ДЛЯ ЗАДАЧ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ</b>

Студент

Группа	ФИО	Подпись	Дата
<b>2ДМ22</b>	<b>Лукьянов Д.М.</b>		<b>02.12.2023</b>

Руководитель

Должность	ФИО	Ученая степень, звание	Подпись	Дата
<b>доцент ОХИ ИШПР</b>	<b>Чузлов В.А.</b>	<b>к.т.н.</b>		<b>04.12.2023</b>

## ЗАДАНИЕ 1

Формула нормализованной гауссовой функции со средним значением  $\mu$  и стандартным отклонением  $\sigma$ :

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Необходимо написать функцию, основанную на использовании массивов NumPy для вычисления гауссовых функций при  $\mu = 0$  и  $\sigma^2 = 0.5, 1.0, 1.5$ . Использовать сетку из 1000 точек в интервале  $-10 \leq x \leq 10$ . Постройте графики данных функций.

## РЕШЕНИЕ 1

### Программная реализация:

---

Cell 1

---

```
import numpy as np
Polynomial = np.polynomial.Polynomial
from scipy.optimize import least_squares
import matplotlib.pyplot as plt
```

---

Cell 2

---

```
def gauss(mu: float,
          sig_square: list,
          x_start: float,
          x_end: float) -> np.array:

    x = np.linspace(x_start, x_end, 1000)
    line_number = len(sig_square)

    res = np.ones((line_number, x.size), dtype=float)

    for i in range(len(sig_square)):
        res[i] = 1 / (sig_square[i] * (2 * np.pi)**0.5) * np.exp(-(x - mu)**2 / (2
        * sig_square[i]**2))

    return x, res
```

```

ss_list = [0.5, 1.0, 1.5]

x, y = gauss(0, ss_list, -10, 10)

fig = plt.figure(figsize = (8,6), dpi=450)
ax = fig.add_subplot()
ax.plot(x, y[0], 'k', label=f'$\sigma$ = {ss_list[0]}')
ax.plot(x, y[1], '-.k', label=f'$\sigma$ = {ss_list[1]}')
ax.plot(x, y[2], '--k', label=f'$\sigma$ = {ss_list[2]}')
ax.legend()
ax.set_ylabel('Значение Y')
ax.set_xlabel('Значение X');

```

**Ответ:**

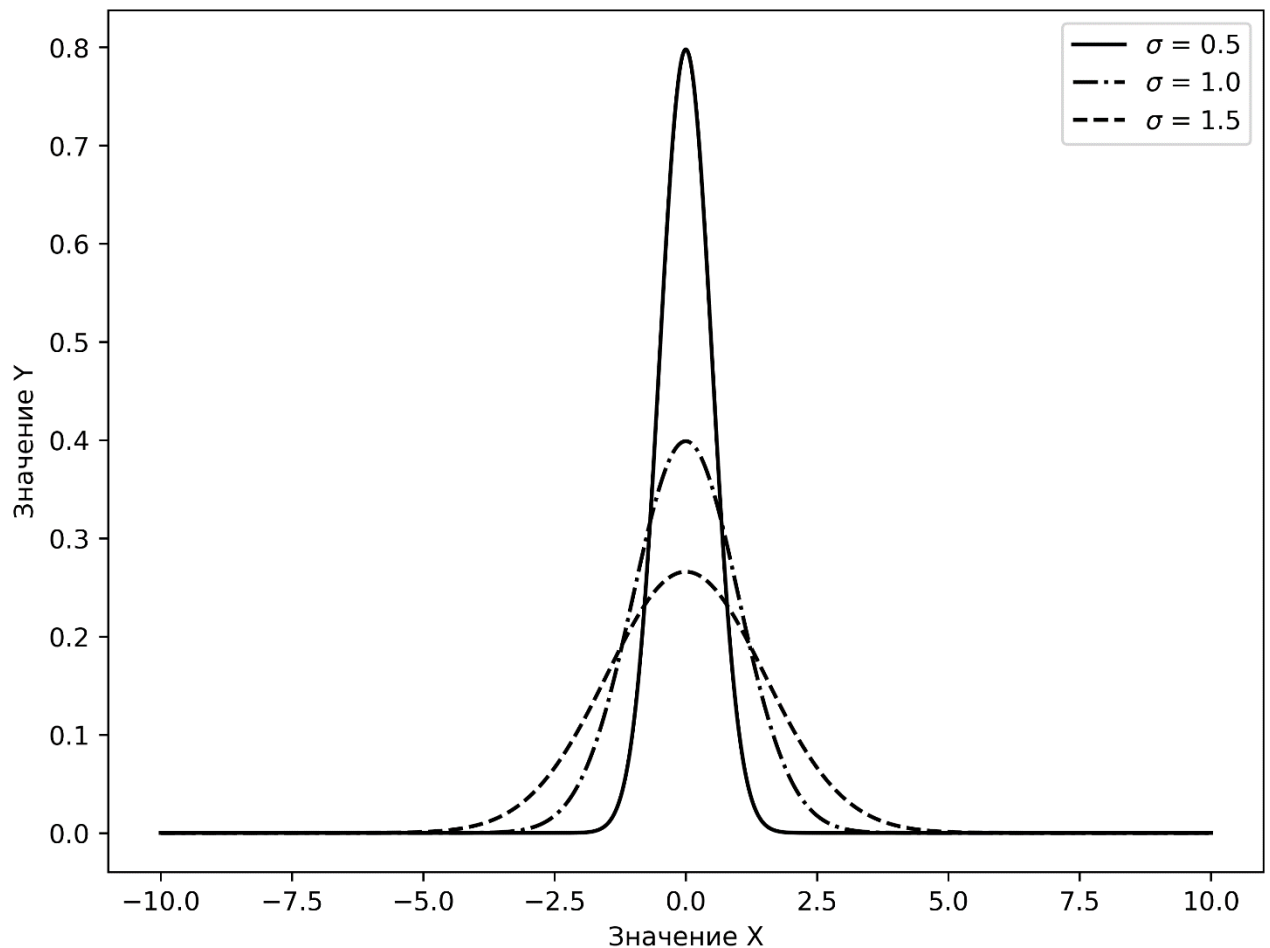


Рисунок 1 – Гауссовы функции при  $\mu = 0$  и  $\sigma^2 = 0.5, 1.0, 1.5$

## ЗАДАНИЕ 2

Уравнение Ван-дер-Ваальса, описывающее состояние газа, можно записать в виде следующей формулы как зависимость давления  $p$  газа от его молярного объема  $V$  и температуры  $T$ :

$$p = \frac{RT}{V - b} - \frac{a}{V^2}$$

где  $a$  и  $b$  – специальные молекулярные константы;

$R = 8.314$  Дж/(моль·К) – универсальная газовая константа.

Формулу легко преобразовать для вычисления температуры по заданному давлению и объему, но ее форма, представляющая молярный объем в отношении к давлению и температуре, является кубическим уравнением:

$$pV^3 - (pb + RT)V^2 + aV - ab = 0$$

Все три корня этого уравнения ниже критической точки  $(T_c, p_c)$  являются действительными: наибольший и наименьший соответствует молярному объему газообразной фазы и жидкой фазы соответственно. Выше критической точки, где не существует жидкая фаза, только один корень является действительным и соответствует жидкая фаза, только один корень является действительным и соответствует молярному объему газа (в этой области его также называют сверхкритической жидкостью, или сверхкритической средой).

Критическая точка определяется по условию  $\left(\frac{\partial p}{\partial V}\right)_T = \left(\frac{\partial^2 p}{\partial V^2}\right)_T = 0$  и для идеального газа Ван-дер-Ваальса выводятся формулы:

$$T_c = \frac{8a}{27Rb} \quad p_c = \frac{a}{27b^2}$$

Для  $NH_3$  константы Ван-дер-Ваальса  $a = 0.4225$  Па·м<sup>6</sup>·моль<sup>-2</sup> и  $b = 37.07 \cdot 10^{-6}$  м<sup>3</sup>·моль<sup>-1</sup>.

- Найти критическую точку для аммиака, затем определить молярный объем при комнатной температуре и давлении (298 К, 1 атм) и при следующих условиях (500 К, 12 МПа);
- Изотерма – это множество точек  $(p, V)$  при постоянной температуре, соответствующее уравнению состояния газа. Построить изотерму ( $p$  в зависимости от  $V$ ) для аммиака при температуре 350 К, используя уравнения Ван-дер-Ваальса, и сравнить ее с изотермой при температуре 350 К для идеального газа, уравнения состояния которого имеет вид  $p = RT/V$  (принять значение  $p$  принадлежащими интервалу  $[101325; 1000000]$  Па, 1000 элементов).

## РЕШЕНИЕ 2

### Программная реализация:

---

Cell 3

---

```
R = 8.314
a = 0.4225
b = 37.07E-6
tc = 8 * a / (27 * R * b)
pc = a / (27 * b**2)
print(f'Критическая температура аммиака {tc:.2f} K\
\nКритическое давление аммиака {pc/1000:.0f} kPa\n')

cond1 = [298, 101325]
cond2 = [500, 12E6]
poly1 = Polynomial([-a * b, a, -(cond1[1] * b + R * cond1[0]), cond1[1]])
poly2 = Polynomial([-a * b, a, -(cond2[1] * b + R * cond2[0]), cond2[1]])
roots1 = poly1.roots()
roots2 = poly2.roots()
def choose_roots(cond: list,
                  roots: list) -> list:

    critical = False

    num_roots = roots.size
    mask = np.full((num_roots,), fill_value = True, dtype=bool)
```

```

    for i in range(num_roots):
        if np.imag(roots[i]) != 0:
            mask[i] = False
            critical = True

    if critical == False:
        vml = min(roots)
        vmv = max(roots)
    else:
        res = np.real(roots[mask][0])
        vmv, vml = res, res

    return vmv, vml

vmv1, vml1 = choose_roots(cond1, roots1)
vmv2, vml2 = choose_roots(cond2, roots2)

print(f'При T = {cond1[0]:.2f} K, p = {cond1[1]/1000:.0f} kPa:\
\nМольный объем жидкости = {vmv1:.2e} м3/моль\
\nМольный объем газа = {vml1:.2e} м3/моль\
\n\nПри T = {cond2[0]:.2f} K, p = {cond2[1]/1000:.0f} kPa:\
\nМольный объем жидкости = {vmv2:.2e} м3/моль\
\nМольный объем газа = {vml2:.2e} м3/моль')

p_ig = np.linspace(101325, 1000000, 1000)
T = 350
v_ig = R * T / p_ig
p_vander = R * T / (v_ig - b) - a / (v_ig**2)

fig = plt.figure(figsize = (8,6), dpi=450)
ax = fig.add_subplot()
ax.plot(v_ig, p_ig/1000, 'k', label='Уравнения Менделеева-Клапейрона')
ax.plot(v_ig, p_vander/1000, 'r', label='Уравнение Ван-дер-Ваальса')
ax.legend()
ax.set_ylabel('Давление, кПа')
ax.set_xlabel('Мольный объем, м$^3$/моль');

```

**Ответ:**

Критическая температура аммиака 406.18 К

Критическое давление аммиака 11387 кПа

При  $T = 298.00$  К,  $p = 101$  кПа:

Мольный объем жидкости =  $2.43 \cdot 10^{-2}$  м<sup>3</sup>/моль

Мольный объем газа =  $5.44 \cdot 10^{-5}$  м<sup>3</sup>/моль

При  $T = 500.00$  К,  $p = 12000$  кПа:

Мольный объем жидкости =  $2.72 \cdot 10^{-4}$  м<sup>3</sup>/моль

Мольный объем газа =  $2.72 \cdot 10^{-4}$  м<sup>3</sup>/моль

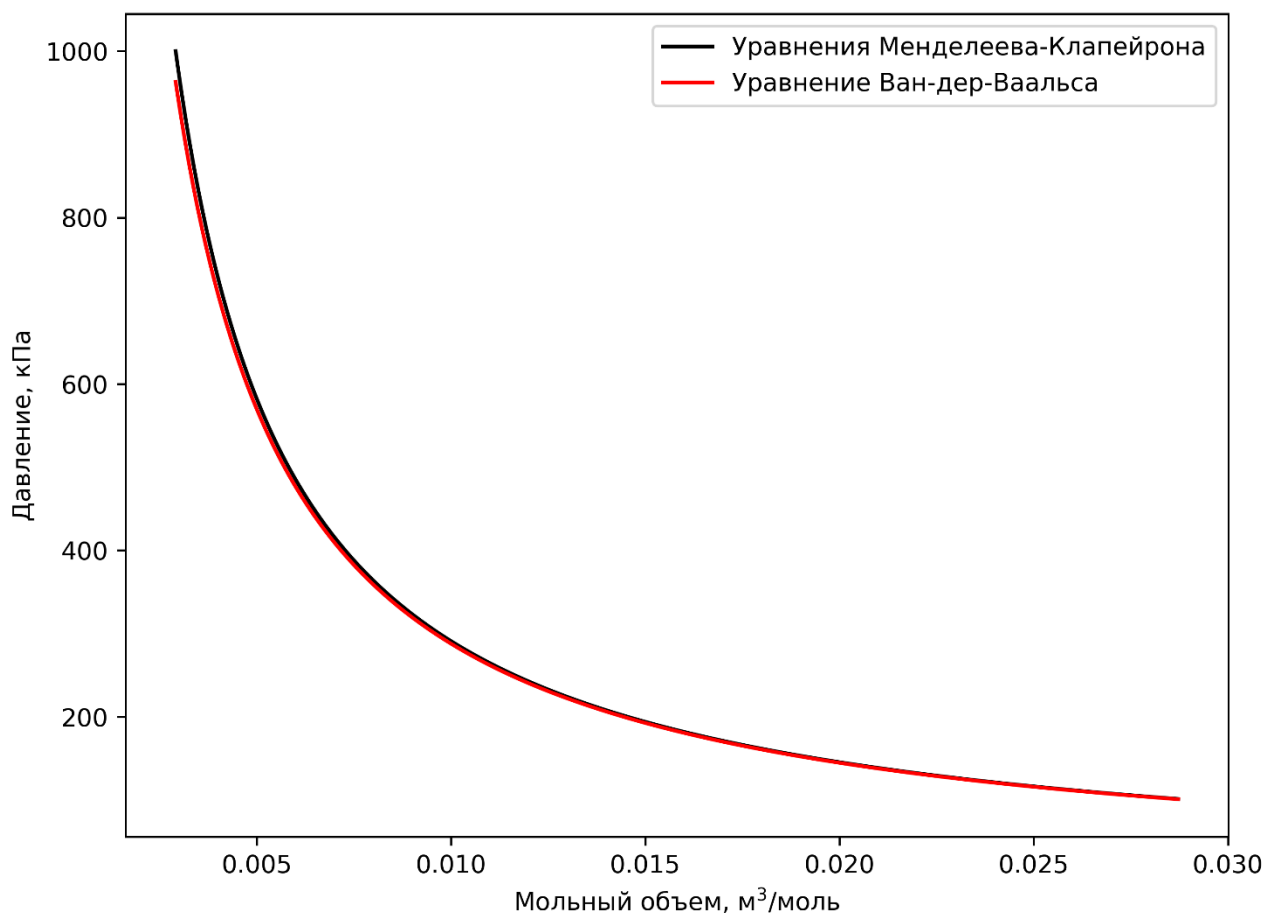


Рисунок 2 – Сравнение изотерм по уравнениям идеального газа и Ван-дер-Ваальса

### ЗАДАНИЕ 3

Закон Бугера-Ламберта-Бера связывает концентрацию вещества  $c$  в образце раствора с интенсивностью света, проходящего через этот образец  $I_t$  с заданной толщиной слоя вещества  $l$  при известной длине волны  $\lambda$ :

$$I_t = I_0 e^{-\alpha c l}$$

где  $I_0$  – интенсивность света на входе в веществе;  
 $\alpha$  – коэффициент поглощения при длине волны  $\lambda$ .

После проведения ряда измерений, позволяющих определить часть света, которая прошла сквозь раствор,  $I_t, I_0$ , коэффициент поглощения  $\alpha$  можно при помощи линейной аппроксимации:

$$y = \ln\left(\frac{I_t}{I_0}\right) = -\alpha c l$$

Несмотря на то что эта прямая проходит через начало координат ( $y = 0$  при  $c = 0$ ), мы будем выполнять подгонку для более общего линейного отношения:

$$y = mc + k$$

где  $m = -\alpha l$  с проверкой  $k$  на приближение к нулю.

При рассмотрении образца раствора с толщиной слоя 0.8 см при измерениях были получены данные, приведенные в таблице: отношение  $I_t/I_0$  при пяти различных концентрациях:

$C$ , моль/л	$I_t/I_0$
0.4	0.891
0.6	0.841
0.8	0.783
1.0	0.744
1.2	0.692

Используя линейную аппроксимацию, определите коэффициент  $\alpha$ .



## РЕШЕНИЕ 3

### Программная реализация:

---

Cell 4

---

```
conc = np.array([0.4, 0.6, 0.8, 1.0, 1.2])
ratio = np.array([0.891, 0.841, 0.783, 0.744, 0.692])
ln_ratio = np.log(ratio)
l = 0.8
def linear(x, params):
    k, b = params
    return k * x + b
def residuals(params, x, y, func):
    return y - func(x, params)
res = least_squares(residuals, x0=[1, 1], args=(conc, ln_ratio, linear))
k, b = res.x
alpha = -k / l
print(f'alpha = {alpha:.2f} л/(см·моль)')
fig = plt.figure(figsize = (8,6), dpi=450)
xlim = [0, 1.5]
ax = fig.add_subplot(xlim=xlim, ylim=[-0.4, 0])
ax.plot(xlim, [k * xlim[0] + b, k * xlim[1] + b], 'k', label='Аппроксимация')
ax.scatter(conc, np.log(ratio), c='r', label='Эксперимент')
ax.legend()
ax.set_ylabel('$\ln\{(I_t/I_0)\}$')
ax.set_xlabel('Концентрация, моль/л');
```

### Ответ:

$$\alpha = 0.39 \frac{\text{л}}{\text{МОЛЬ} \cdot \text{СМ}}$$

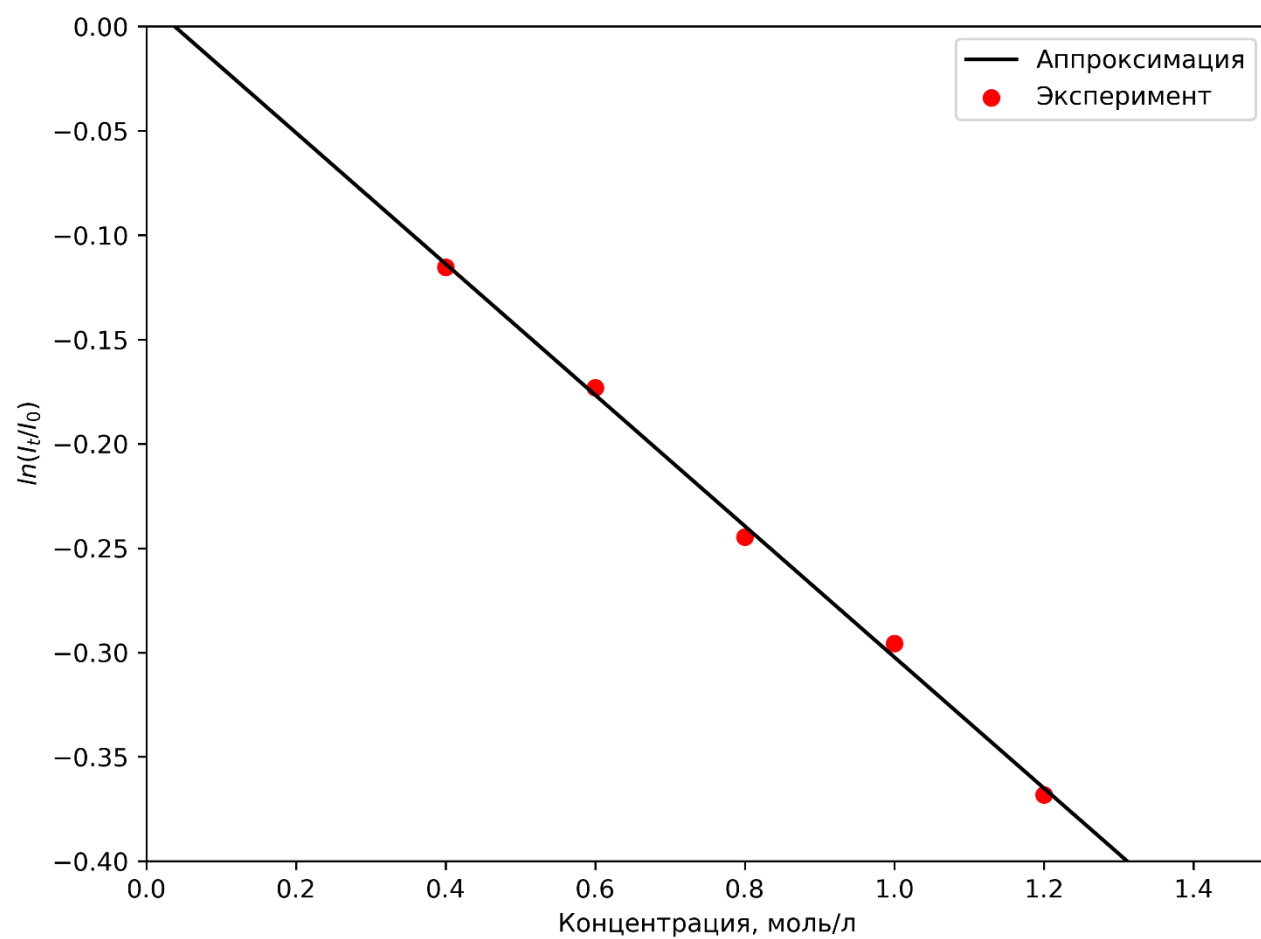


Рисунок 3 – Линейная аппроксимация экспериментальных данных