**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 |
| **PYTHON ДЛЯ ЗАДАЧ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ** |

Студент

|  |  |  |  |
| --- | --- | --- | --- |
| **Группа** | **ФИО** | **Подпись** | **Дата** |
| **2ДМ22** | **Лукьянов Д.М.** |  | **02.12.2023** |

Руководитель

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Должность** | **ФИО** | **Ученая степень, звание** | **Подпись** | **Дата** |
| **доцент ОХИ ИШПР** | **Чузлов В.А.** | **к.т.н.** |  | **04.12.2023** |

# Задание 1

Формула нормализованной гауссовой функции со средним значением и стандартным отклонением :

Необходимо написать функцию, основанную на использовании массивов NumPy для вычисления гауссовых функций при . Использовать сетку из 1000 точек в интервале . 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:**

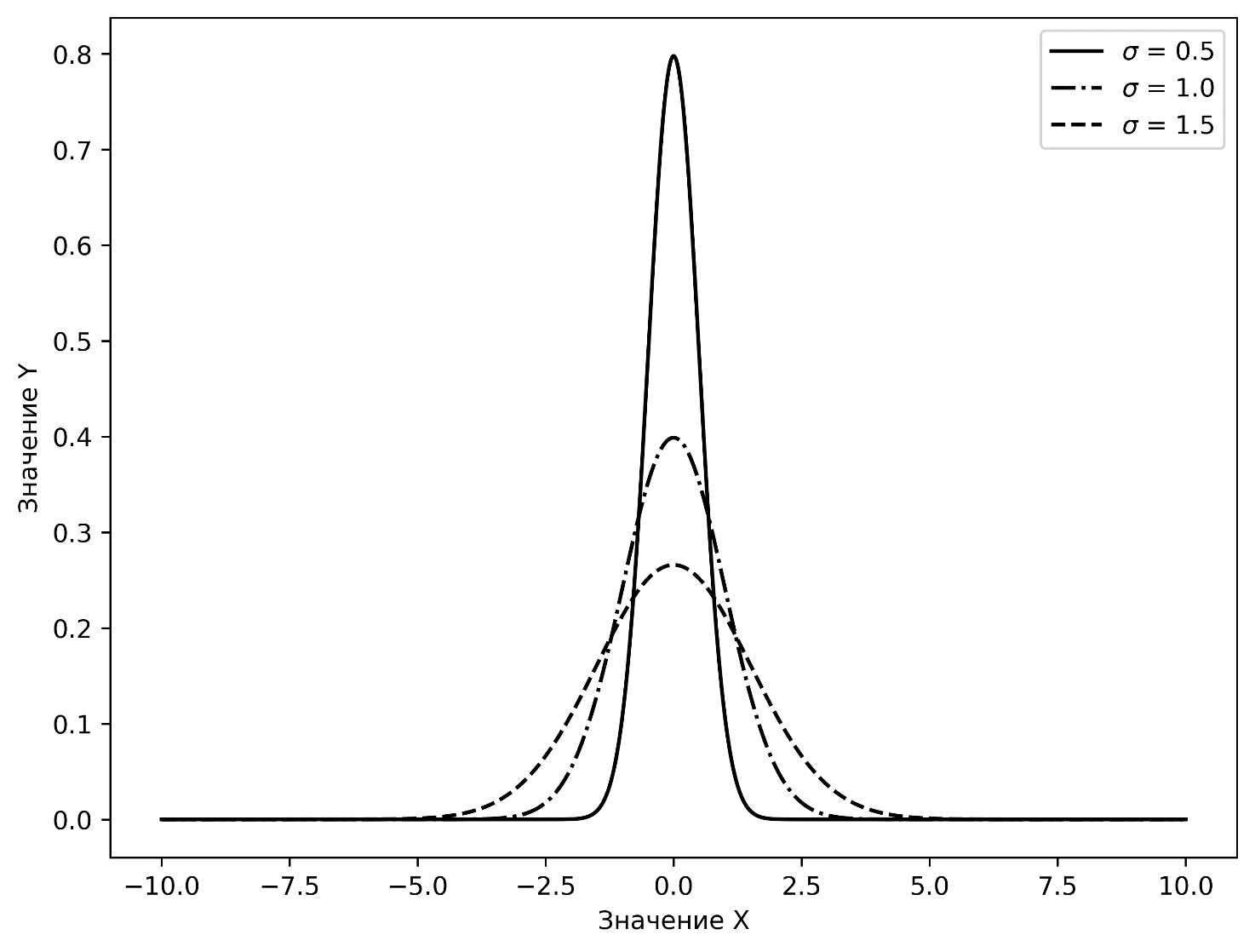


Рисунок 1 – Гауссовы функции при

# Задание 2

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

где  – специальные молекулярные константы;

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 уравенением:

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

Критическая точка определяется по условию и для идеального газа Ван-дер-Ваальса выводятся формулы:

Для константы Ван-дер-Ваальса и  
.

* Найти критичесую 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 ( при постоянной температуре, соответсвующее уравнению состояния газа. Построить изотерму ( в зависимости от ) для аммиака при температуре 350 К, используя уравнения Ван-дер-Ваальса, и сравнить ее с изотермой при температуре 350 К для идеального газа, уравления состояния которого имеет вид (принять значение принадлежащими интервалу , 1000 элементов).

# Решение 2

**Програмная 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 Т = {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 Т = {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 K

Критическое давление аммиака 11387 kPa

При Т = 298.00 K, p = 101 kPa:

Мольный объем жидкости = 2.43e-02 м3/моль

Мольный объем газа = 5.44e-05 м3/моль

При Т = 500.00 K, p = 12000 kPa:

Мольный объем жидкости = 2.72e-04 м3/моль

Мольный объем газа = 2.72e-04 м3/моль

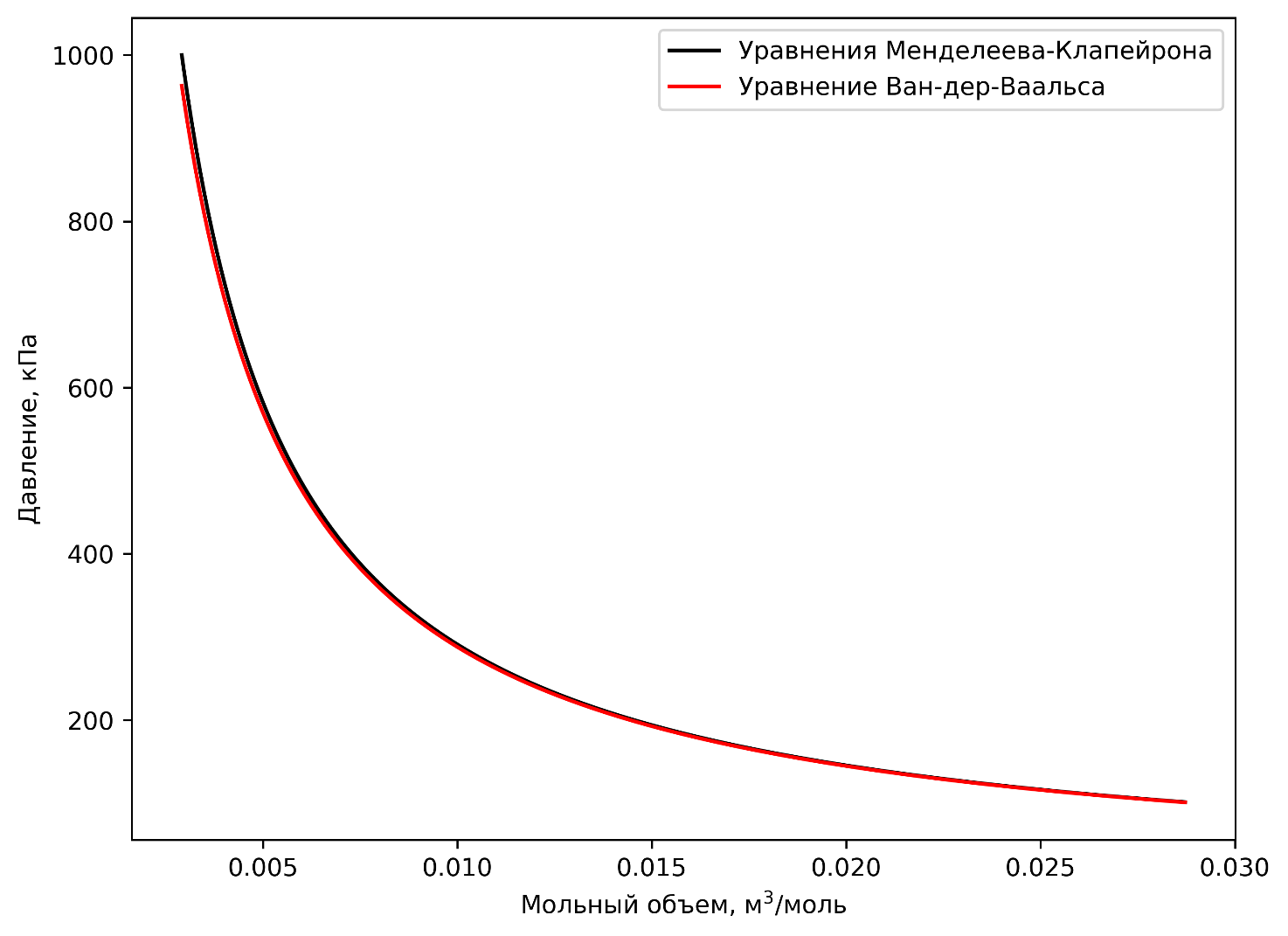


Рисунок 2 – Сравнение изотерм по уравнениям идеального газа и Ван‑дер-Ваальса

# Задание 3

Закон Бугера‑Ламберта‑Бера связывает концентрацию вещества в образце раствора с интенсивностью света, проходящего через этот образец с заданной толщиной слоя вещества при известной длине волны :

где  – интенсивность света на входе в веществе;

 – коэффициент поглощения при длине волны .

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

Несмотря на то что эта прямамя проходит через начало координат ( при ), мы будем выполнять подгонку для более общего линейного отношения:

где с проверкой на приближение к нулю.

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

|  |  |
| --- | --- |
|  |  |
| 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 .

# 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:**

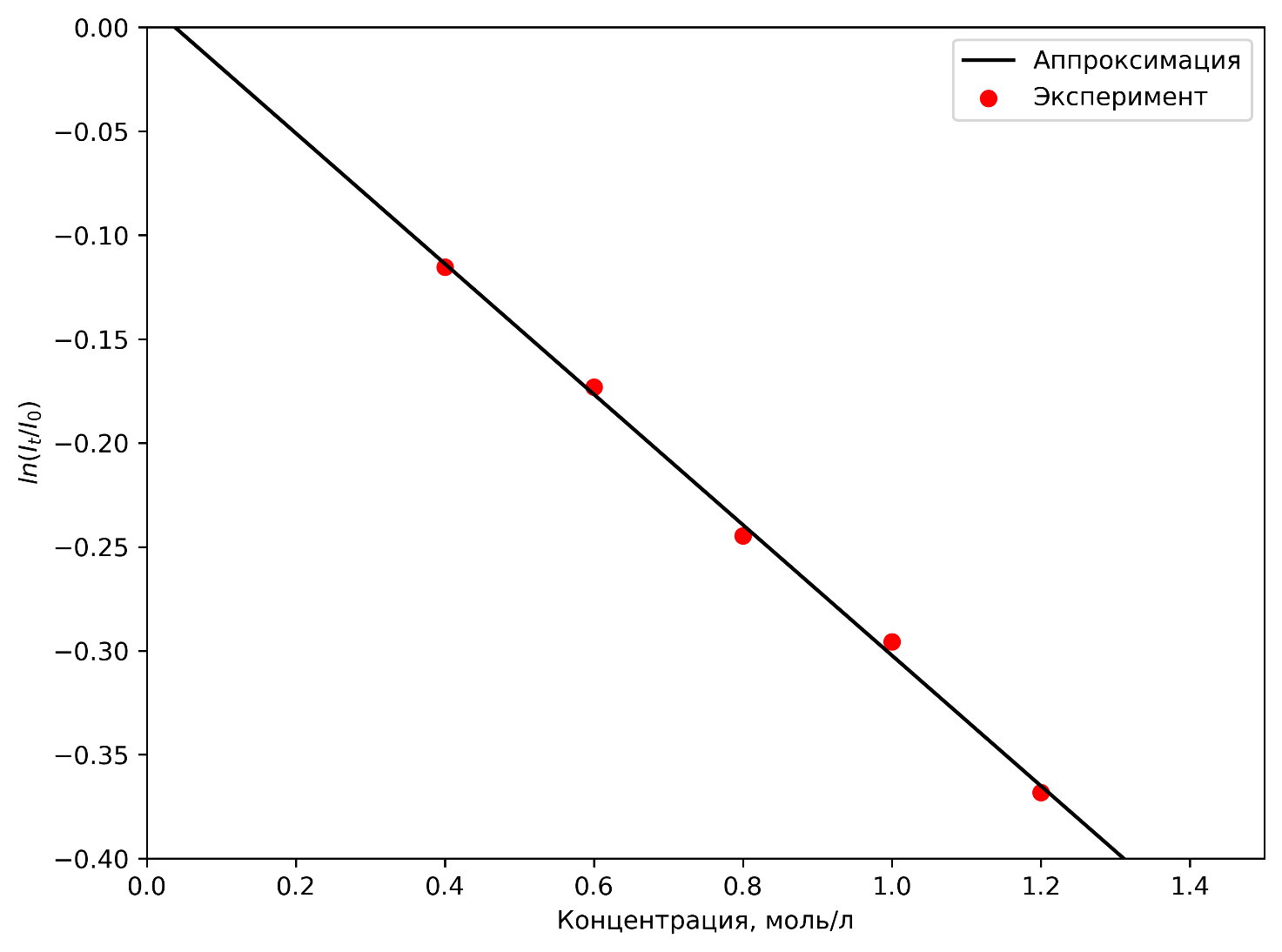


Рисунок 3 – Линейная аппроксимация экспериментальных данных