

**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 № 3

По дисциплине
РУТНОН ДЛЯ ЗАДАЧ ХИМИЧЕСКОЙ ТЕХНОЛОГИИ

Студент

Группа	ФИО	Подпись	Дата
2ДМ22	Лукьянов Д.М.		03.12.2023

Руководитель

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

ЗАДАНИЕ 1

Дана зависимость давления паров вещества от температуры:

$T, ^\circ\text{C}$	$p, \text{атм}$
40	0,2453
50	0,5459
60	1,2151
70	2,7042
80	6,0184
90	13,3943
100	29,8096

Определить значения давления паров при $T \in [40; 100]$ с шагом 5°C , используя

- Кубический сплайн;
- Одну из аппроксимирующих функций: проверить линейную, степенную и экспоненциальную аппроксимирующие функции, выбрать наиболее подходящую. (по значению суммарной ошибки) и провести расчеты с использованием данной функции.

РЕШЕНИЕ 1

Программная реализация:

Cell 1

```
import numpy as np
import scipy as sp
from scipy.optimize import least_squares
from scipy.interpolate import interp1d
from scipy.integrate import solve_ivp
from scipy.integrate import quad
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.patches import Polygon
```

Cell 2

```
t = np.array([40, 50, 60, 70, 80, 90, 100])
p = np.array([0.2453, 0.5459, 1.2151, 2.7042, 6.0184, 13.3943, 29.8096])
x_eval = np.arange(40, 101, 5)
```

```

cubic = interp1d(t, p, kind='cubic')
def linear_model(x, params):
    k, b = params
    return k * x + b
def residuals(params, x, y, func):
    return y - func(x, params)
x0 = 0.1, 0.1

def power_model(x, params):
    a, b = params
    return a * x**b

def exp_model(x, params):
    a, b = params
    return a * np.exp(b * x)

line = least_squares(residuals, x0=x0, args=(t, p, linear_model))
linear_params, linear_cost = line.x, line.cost
print(f'Линейная аппроксимация:\n'
      f'\nk = {linear_params[0]:.3f}, b = {linear_params[1]:.3f}, \n'
      f'суммарная ошибка = {linear_cost:.4f}')

power = least_squares(residuals, x0=x0, args=(t, p, power_model))
power_params, power_cost = power.x, power.cost
print(f'Степенная аппроксимация:\n'
      f'\na = {power_params[0]:.3e}, b = {power_params[1]:.3f}, \n'
      f'суммарная ошибка = {power_cost:.4f}')

exp = least_squares(residuals, x0=x0, args=(t, p, exp_model))
exp_params, exp_cost = exp.x, exp.cost
print(f'Экспоненциальная аппроксимация:\n'
      f'\na = {exp_params[0]:.3e}, b = {exp_params[1]:.3f}, \n'
      f'суммарная ошибка = {exp_cost:.3e}')
cubic_spline_eval = cubic(x_eval)
line_eval = linear_params[0] * x_eval + linear_params[1]
power_eval = power_params[0] * x_eval**power_params[1]
exp_eval = exp_params[0] * np.exp(exp_params[1] * x_eval)

```

Cell 3

```

df_res = pd.DataFrame({'T, °C': x_eval,
                       'Cubic_spline_eval': cubic_spline_eval,

```

```
'line_eval': line_eval,  
'power_eval': power_eval,  
'exp_eval': exp_eval})
```

df_res

Cell 4

```
t_space = np.linspace(40, 100, 100)  
xlim = [t_space[0], t_space[-1]]  
ylim = [0, 30]  
fig = plt.figure(figsize=(8,6), dpi=450)  
ax = fig.add_subplot(xlim=xlim, ylim=ylim)  
  
ax.plot(t_space, cubic(t_space), '.-k', label='Кубический сплайн')  
ax.plot(t_space, linear_params[0] * t_space + linear_params[1], c='orange',  
        label='Линейная аппроксимация')  
ax.plot(t_space, power_params[0] * t_space**power_params[1], '-g',  
        label='Степенная аппроксимация')  
ax.plot(t_space, exp_params[0] * np.exp(t_space*exp_params[1]), '--b',  
        label='Экспоненциальная аппроксимация')  
ax.scatter(t, p, c='r', label='Эксперимент')  
  
ax.legend()  
ax.set_ylabel('Давление, атм')  
ax.set_xlabel('Температура, °C');
```

Ответ:

Линейная аппроксимация:

$k = 0.426$, $b = -22.094$, суммарная ошибка = 95.2305

Степенная аппроксимация:

$a = 1.031e-12$, $b = 6.724$, суммарная ошибка = 0.9686

Экспоненциальная аппроксимация:

$a = 1.000e-02$, $b = 0.080$, суммарная ошибка = 4.009e-09

Можно видеть, что экспоненциальная аппроксимация характеризуется наименьшей суммарной ошибкой, т.е. является наиболее удобной из рассмотренных для данного набора точек.

Значения в интервале $T \in [40; 100]$ °C представлены в таблице:

T, °C	Cubic_spline_eval	line_eval	power_eval	exp_eval
40	0,2453	-5,0660	0,0609	0,2453
45	0,3709	-2,9375	0,1345	0,3660
50	0,5459	-0,8091	0,2731	0,5460
55	0,8131	1,3193	0,5184	0,8145
60	1,2151	3,4478	0,9305	1,2151
65	1,8085	5,5762	1,5939	1,8127
70	2,7042	7,7047	2,6233	2,7042
75	4,0417	9,8331	4,1717	4,0342
80	6,0184	11,9616	6,4383	6,0184
85	8,9285	14,0900	9,6783	8,9784
90	13,3943	16,2185	14,2136	13,3943
95	20,1199	18,3469	20,4447	19,9819
100	29,8096	20,4754	28,8642	29,8096

Можно видеть, что кубический сплайн и экспоненциальная аппроксимация позволяют получить похожие результаты.

График:

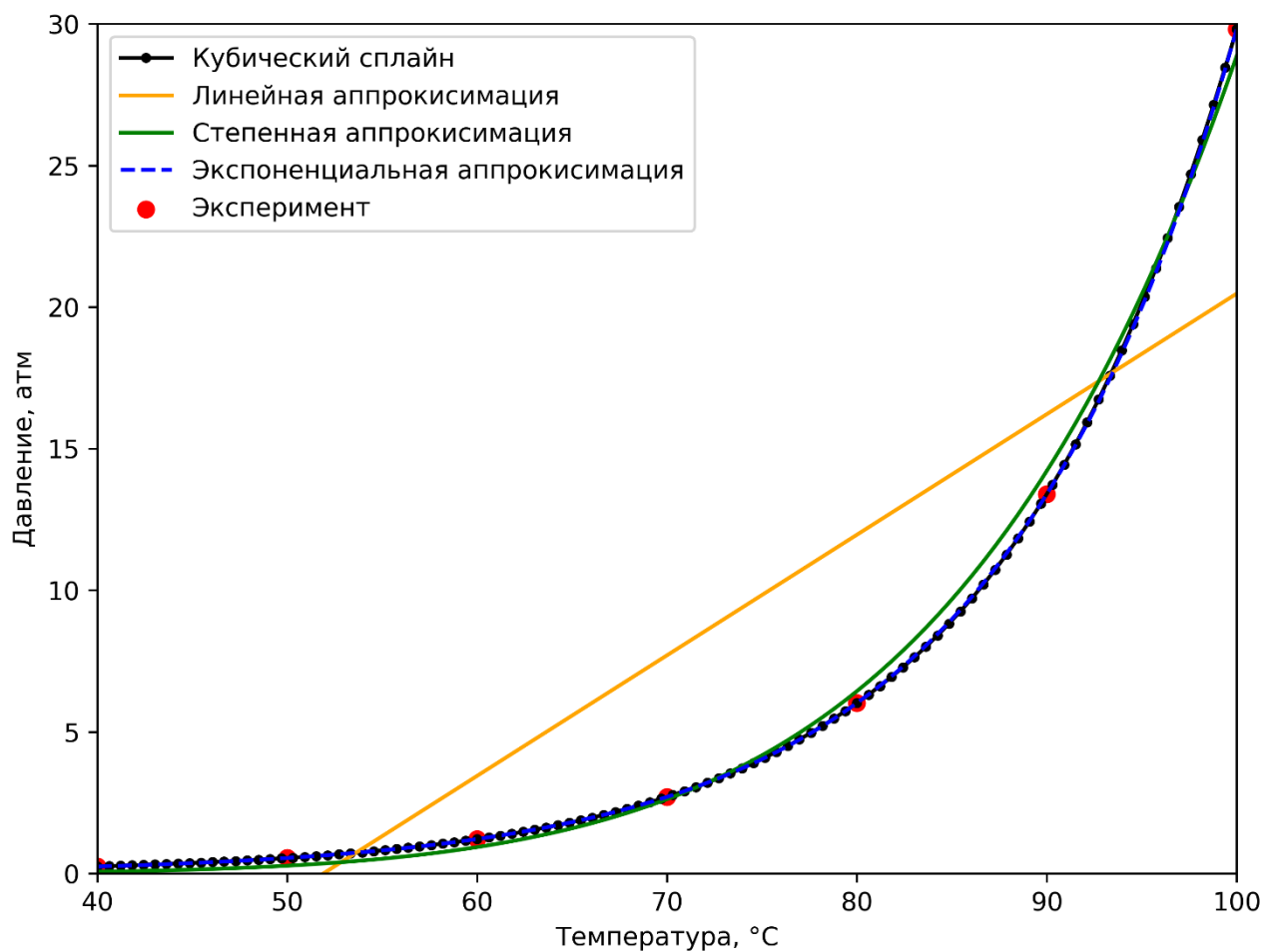
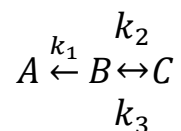


Рисунок 1 – Графическое сравнение интерполяции кубическим сплайном и аппроксимаций линейной, степенной и экспоненциальной функциями

ЗАДАНИЕ 2

Дана схема химических превращений:



$$C_{A_0} = 0,8 \left(\frac{\text{МОЛЬ}}{\text{Л}} \right) \quad k_1 = 0,8 \text{ (с}^{-1}\text{)}$$

$$C_{B_0} = 0,2 \left(\frac{\text{МОЛЬ}}{\text{Л}} \right) \quad k_2 = 0,96 \text{ (с}^{-1}\text{)}$$

$$C_{C_0} = 0,0 \left(\frac{\text{МОЛЬ}}{\text{Л}} \right) \quad k_3 = 0,1 \text{ (с}^{-1}\text{)}$$

Решите систему дифференциальных уравнений изменения концентраций веществ во времени при помощи функции `scipy.integrate.solve_ivp()` на отрезке $[0; 5]$ с шагом $h = 0,1$. По результатам расчетов постройте зависимость $C(t)$ для каждого компонента при помощи библиотеки `matplotlib`.

РЕШЕНИЕ 2

Программная реализация:

Cell 5

```
t_start, t_end, t_step = 0, 5, 0.1
t_eval = np.arange(t_start, t_end+t_step, t_step)
start_conc = [0.8, 0.2, 0]
rate_constants = (0.8, 0.96, 0.1)

def derivatives(t, y, *rate_constants):
    c_a, c_b, c_c = y
    k1, k2, k3 = rate_constants
    dca_dt = k1 * c_b
    dcb_dt = -k1 * c_b - k2 * c_b + k3 * c_c
    dcc_dt = k2 * c_b - k3 * c_c

    return dca_dt, dcb_dt, dcc_dt

solution = solve_ivp(derivatives, (t_start, t_end), start_conc,
                     t_eval=t_eval, args=rate_constants)
c_a, c_b, c_c = solution.y[0], solution.y[1], solution.y[2]
```

Cell 6

```
xlim = [t_eval[0], t_eval[-1]]
fig = plt.figure(figsize=(8,6), dpi=450)
ax = fig.add_subplot(xlim=xlim)

ax.plot(t_eval, c_a, 's-b', label='$C_a$')
ax.plot(t_eval, c_b, '*-r', label='$C_b$')
ax.plot(t_eval, c_c, 'o-g', label='$C_c$')

ax.legend()
ax.set_ylabel('Концентрация, моль/л')
ax.set_xlabel('Время, с');
```

Ответ:

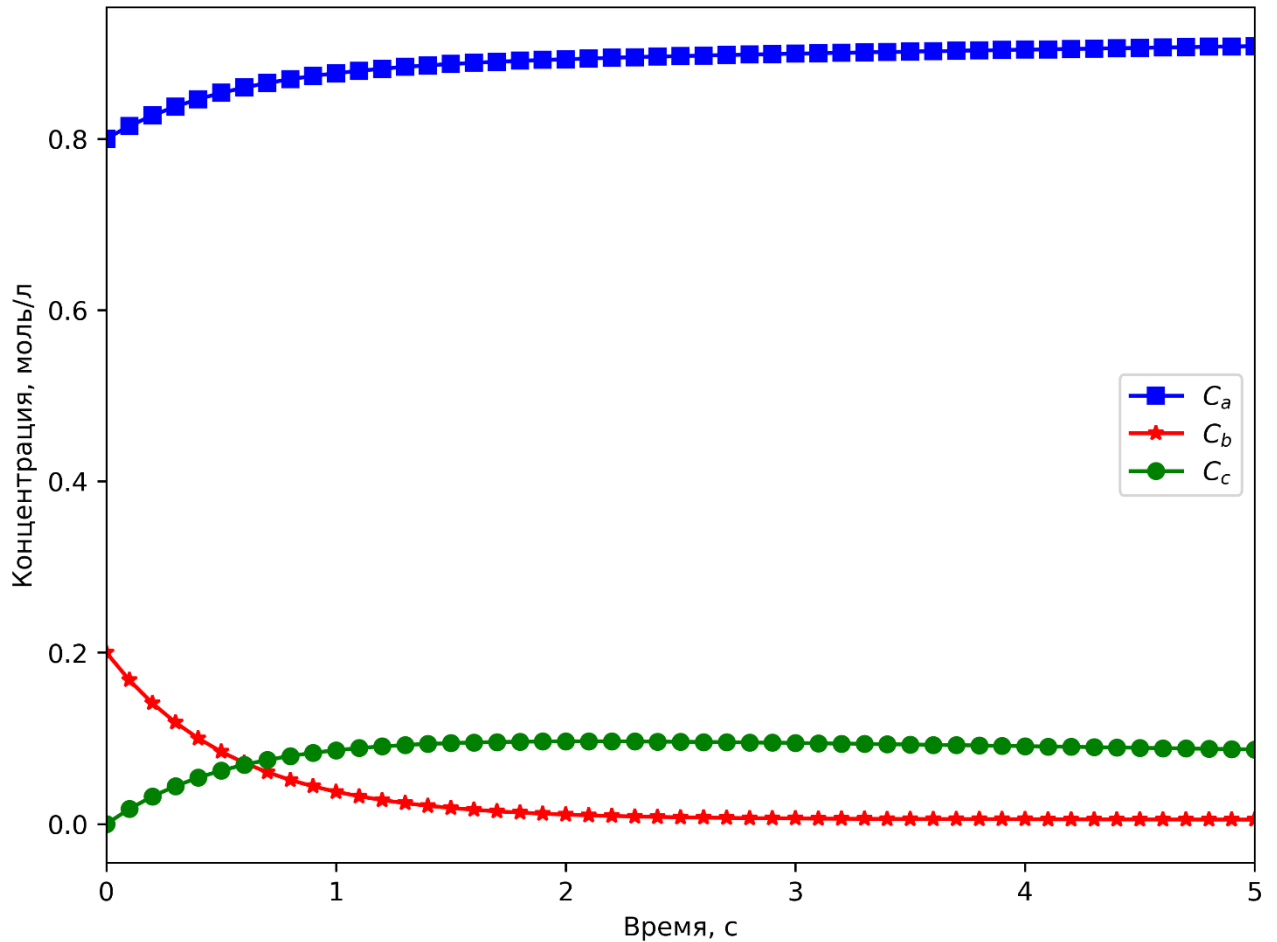


Рисунок 2 – Зависимость концентраций веществ от времени

ЗАДАНИЕ 3

Используйте функцию `scipy.integrate.quad()` для вычисления значения энтропии воды при ее нагревании от 400 до 500 К по формуле:

$$\Delta S = \eta \int_{400}^{500} \frac{C_v(T) dT}{T}$$

$$C_v(T) = R \sum_{j=1}^{12} A_j \tau^{j-1}$$

$$\tau = 1 - \frac{T}{T_c}$$

где T – температура, К;

$\eta = 3$ – количество молей;

C_v – теплоемкость, Дж/(моль·К);

R – универсальная газовая постоянная;

$T_c = 647,126$ – критическая температура, К.

Коэффициенты полинома $A(1) - A(12)$:

Коэффициент	Значение
A_1	7,7305055
A_2	-24,93618016
A_3	195,5654567
A_4	1986,485797
A_5	-53305,43411
A_6	505697,1723
A_7	-2724774,677
A_8	9167737,673
A_9	-19622033,78
A_{10}	25984725,33
A_{11}	-19419431,35
A_{12}	6263206,554

РЕШЕНИЕ 2

Программная реализация:

Cell 7

tc = 647.126

eta = 3

```
a_arr = [7.4305055,  
         -24.93618016,  
         195.5654567,  
         1986.485797,  
         -53305.43411,  
         505697.1723,  
         -2724774.677,  
         9167737.673,  
         -19622033.78,  
         25984725.33,  
         -19419431.35,  
         6263206.554]
```

```
def func(t, tc, eta, a_arr):
```

```
    R = 8.314
```

```

tau = 1 - t / tc
a_arr_len = len(a_arr)
cv = 0
for i in range(a_arr_len):
    cv += R * a_arr[i] * tau**(i)
return eta * cv / t

t_space = np.arange(350, 551, 1)
der = func(t_space, tc, eta, a_arr)
t_space_window = t_space[50:151]
der_window = der[50:151]
area = quad(func, t_space_window[0], t_space_window[-1], args=(tc, eta, a_arr))

```

Cell 8

```

xlim = [350, 550]
ylim = [0, 0.7]
fig = plt.figure(figsize=(8,6), dpi=450)
ax = fig.add_subplot(xlim=xlim, ylim=ylim)

ax.plot(t_space, der, 'r')

verts = [(400, 0), *zip(t_space_window, der_window), (500, 0)]
poly = Polygon(verts, facecolor='0.8', edgecolor='0')
ax.add_patch(poly)
ax.text(410, 0.2, '$\Delta S = \eta \int_{400}^{500} \frac{C_v(T)}{T} dT$ = ' + f'{area[0]:.1f} Дж', fontsize=14)
ax.set_ylabel('$\eta \frac{C_v(T)}{T}, \frac{Дж}{К}$', fontsize=14)
ax.set_xlabel('T, К');

```

Ответ:

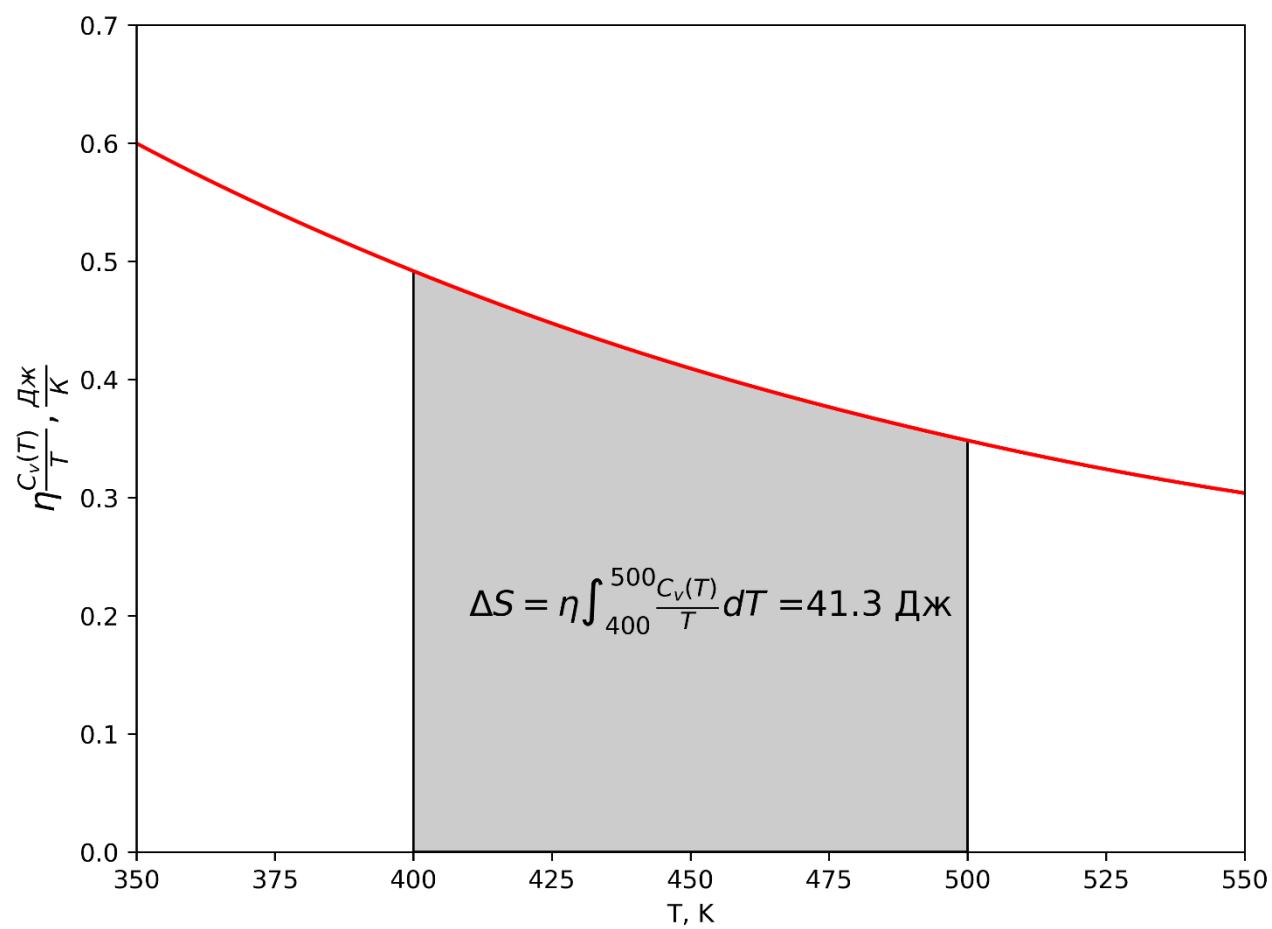


Рисунок 3 – Изменение энтропии 3 молей воды при их нагреве от 400 К до 500 К