**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 |
| **PYTHON 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:

|  |  |
| --- | --- |
|  |  |
| 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 с шагом 5 °С, используя

* Кубический сплайн;
* Одну из аппроксимируюющих функций: проверить линейную, степенную и экспоненциальную аппроксимирующие функции, выбрать наиболее подходящу. (по значению суммарной ошибки) и провести расчеты с использованиеи 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:\

\nk = {linear\_params[0]:.3f}, b = {linear\_params[1]:.3f}, \

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:\

\na = {power\_params[0]:.3e}, b = {power\_params[1]:.3f}, \

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:\

\na = {exp\_params[0]:.3e}, b = {exp\_params[1]:.3f}, \

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 аппрокисимация')

ax.plot(t\_space, power\_params[0] \* t\_space\*\*power\_params[1], '-g',

label='Степенная аппрокисимация')

ax.plot(t\_space, exp\_params[0] \* np.exp(t\_space\*exp\_params[1]), '--b',

label='Экспоненциальная аппрокисимация')

ax.scatter(t, p, c='r', label='Эксперимент')

ax.legend()

ax.set\_ylabel('Давление, атм')

ax.set\_xlabel('Температура, °С');

**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. явяляется наиболее удочной из рассмотренных для данного набора точек.***

***Значениия в интервале 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 аппроксимция позволяют получить похожие результаты.***

**График:**

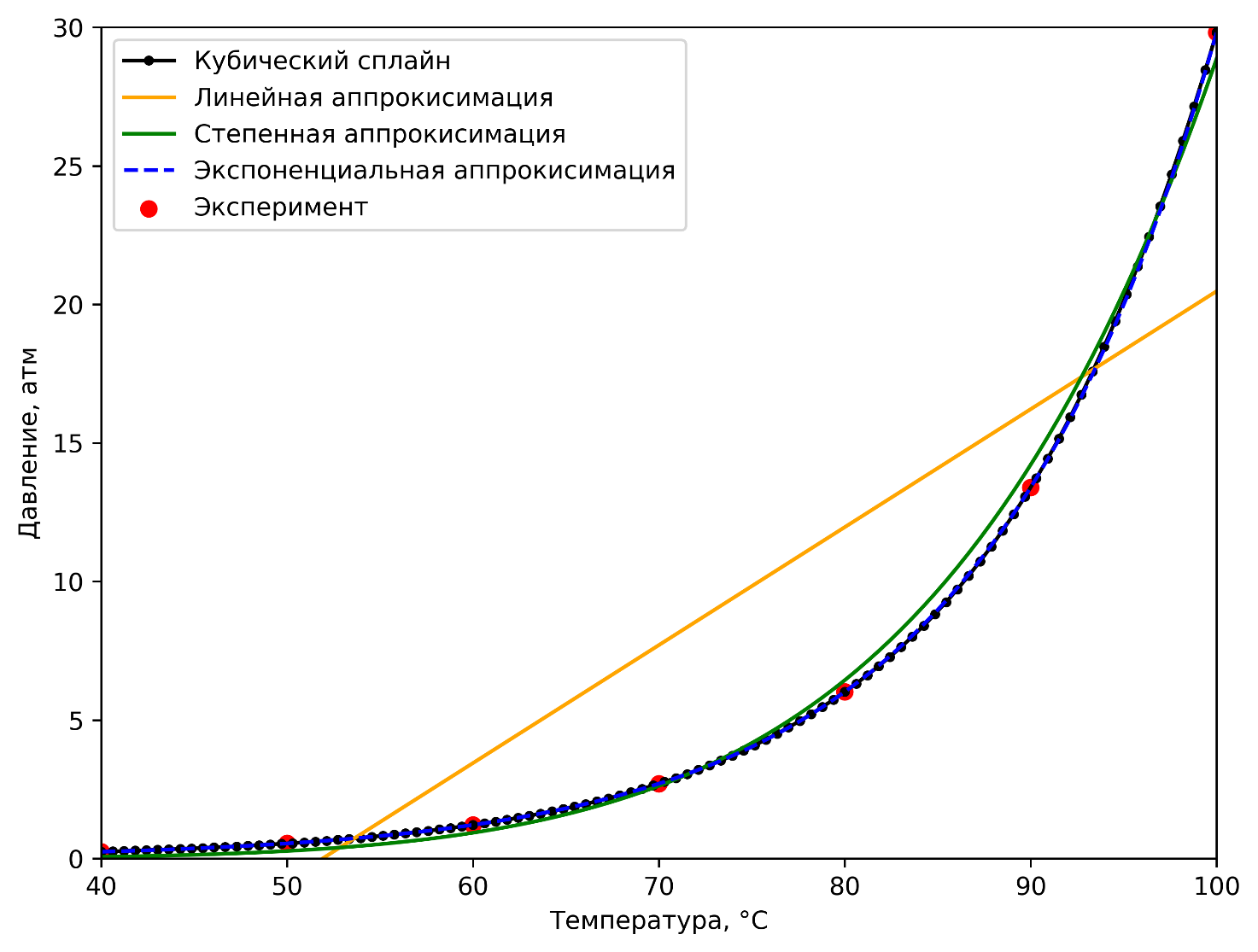


Рисунок  – Графической сравненний интерполяции кубическим сплайном и аппроксимаций линейной, степенной и экспоненциальной функциями

# Задание 2

Дана схема химических превращений:

Решите систему дифферинциалных уравнений изменения концентраций веществ во времени при помощи функции scipy.integrate.solve\_ivp() на отрезке [0; 5] с шагом . По результатам расчетов постройте зависимость 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:**

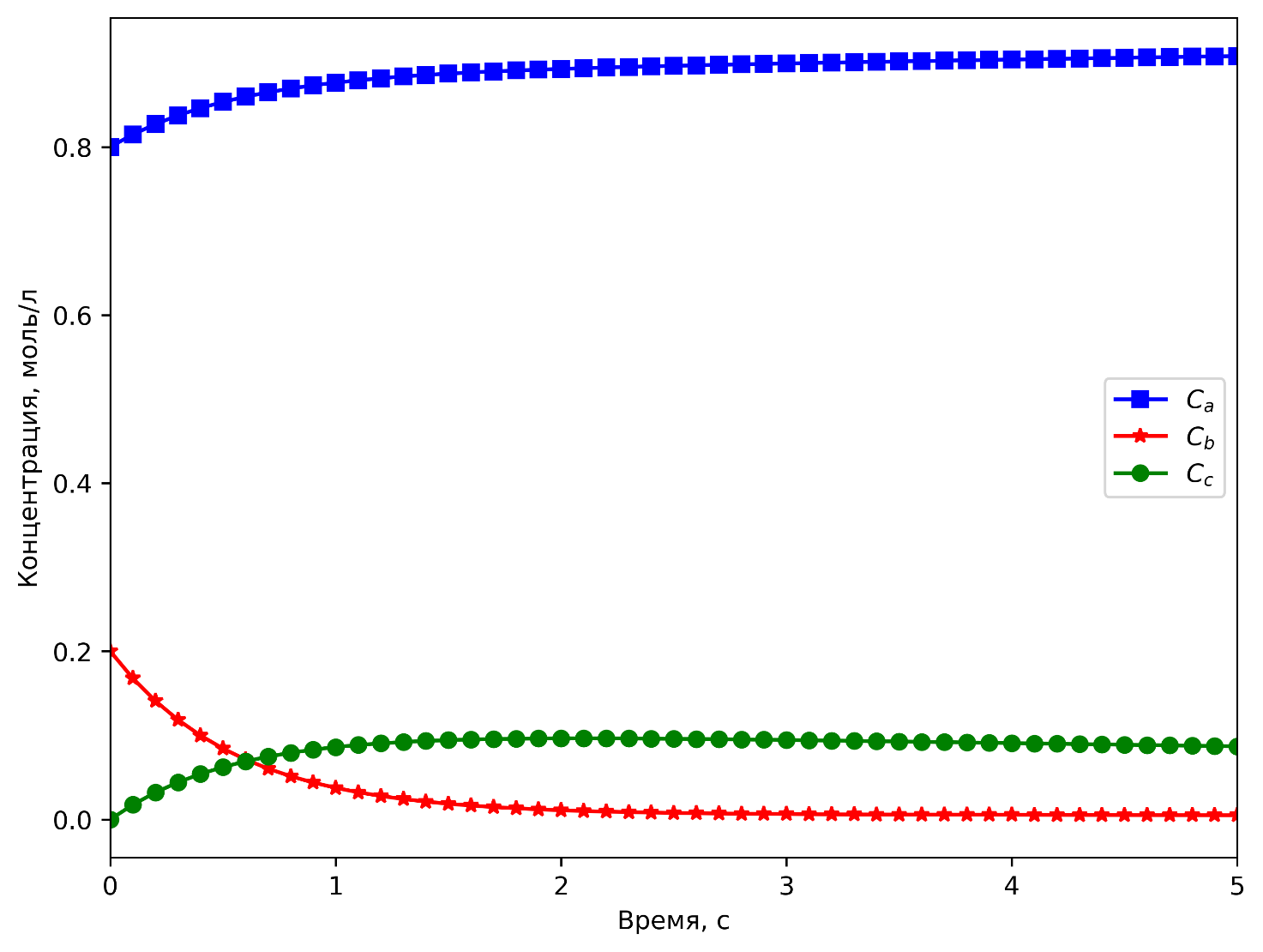


Рисунок  – Зависимость концентраций веществ от времени

# Задание 3

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

где  – температура, К;

 – количество молей;

 – темпоемкость, Дж/(моль·К);

 – универсальная газовая постоянная;

 – критическая температура, К.

Коэффициенты полинома

|  |  |
| --- | --- |
| Коэффициент | Значение |
|  | 7,7305055 |
|  | -24,93618016 |
|  | 195,5654567 |
|  | 1986,485797 |
|  | -53305,43411 |
|  | 505697,1723 |
|  | -2724774,677 |
|  | 9167737,673 |
|  | -19622033,78 |
|  | 25984725,33 |
|  | -19419431,35 |
|  | 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, K');

**Ответ:**

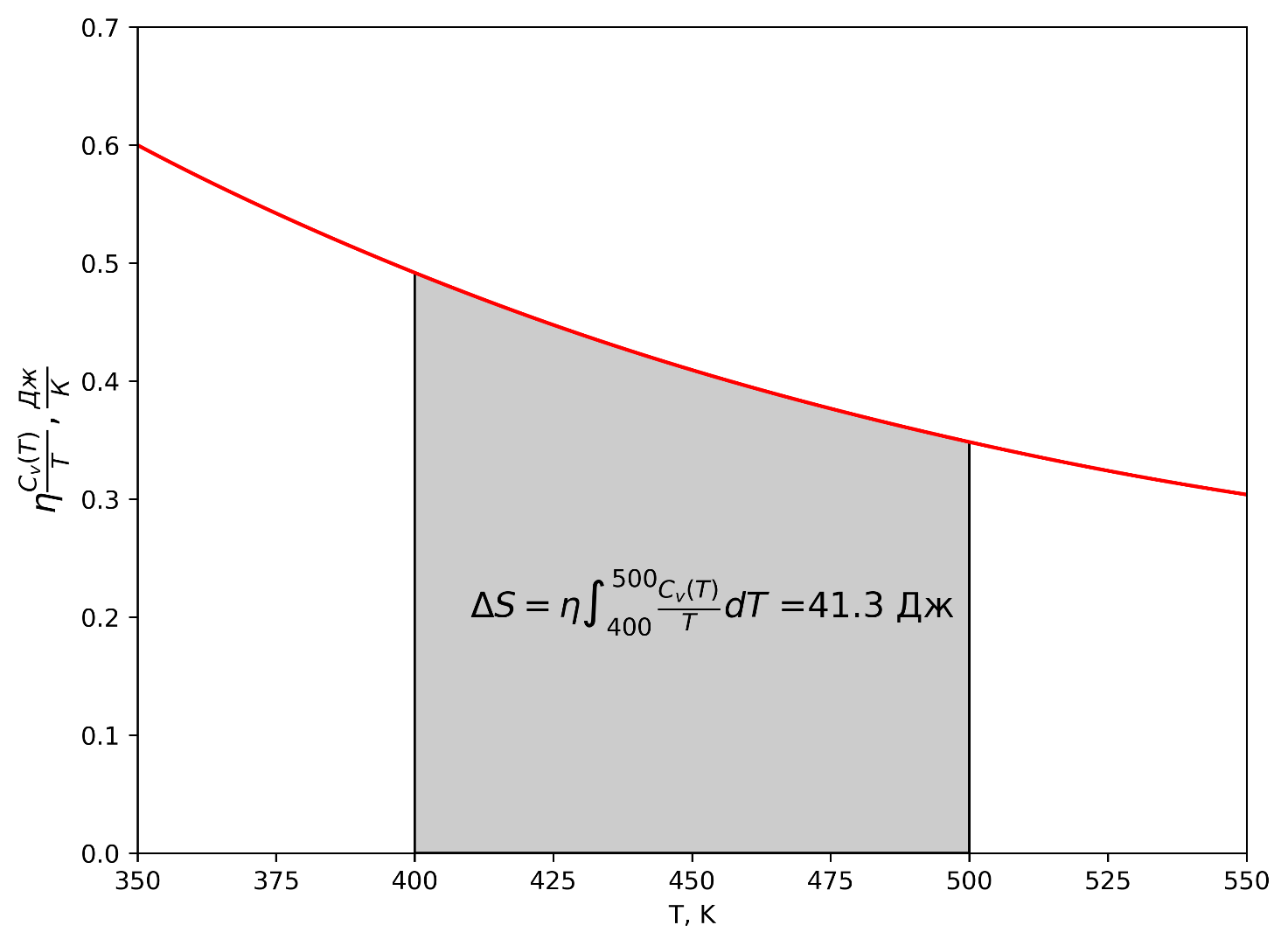


Рисунок  – Изменение энтропиии 3 молей воды при их нагреве от 400 К до 500 К