

# Отчёт по лабораторной работе №5

Группа Б9120-01.03.02миопд

Агличиев Александр

20 декабря 2023 г.

# 1 Метод замены ядра на вырожденное

## 1.1 Постановка задачи

Необходимо решить интегральное уравнение методом замены ядра на вырожденное:

$$u(x) = \int_1^0 \frac{\sin(0,6xs)}{s} ds = x$$

## 1.2 Решение

Аппроксимируем ядро уравнение  $K(x, s) = \frac{\sin(0,6xs)}{s}$  суммой трех членов разложения  $K(x, s)$  в ряд Тейлора, то есть положим

$$\frac{\sin(0,6xs)}{s} \approx 0.6x - 0.036x^3s^2 + 0.000648x^5s^4$$

Тогда решение будем искать в виде:

$$u(x) = x + C_1x + C_2x^3 + C_3x^5$$

Обозначив:

$$\alpha_1 = x$$

$$\beta_1 = 0.6$$

$$\alpha_2 = x^3$$

$$\beta_2 = -0.036s^2$$

$$\alpha_3 = x^5$$

$$\beta_3 = 0.000648s^4$$

Найдём по следующим формулам коэффициенты:

$$f_i = \int_b^a \beta_i(s) f(s) ds$$

$$A_{ij} = \int_b^a \alpha_j(s) \beta_i(s)$$

Система принимает вид:

$$\begin{cases} 0.7C_1 - 0.15C_2 - 0.1C_3 = 0.3, \\ 0.009C_1 + 1.006C_2 - 0.0045C_3 = -0.009, \\ -0.0001296C_1 - 0.0000925C_2 - C_3 = -0.000183 \end{cases}$$

Решим эту систему и найдем коэффициенты  $C_i$

$$C_1 = 0,426, C_2 = -0.0127, C_3 = 0.000184$$

Построим график решение на интервале  $x \in [0, 1]$ .

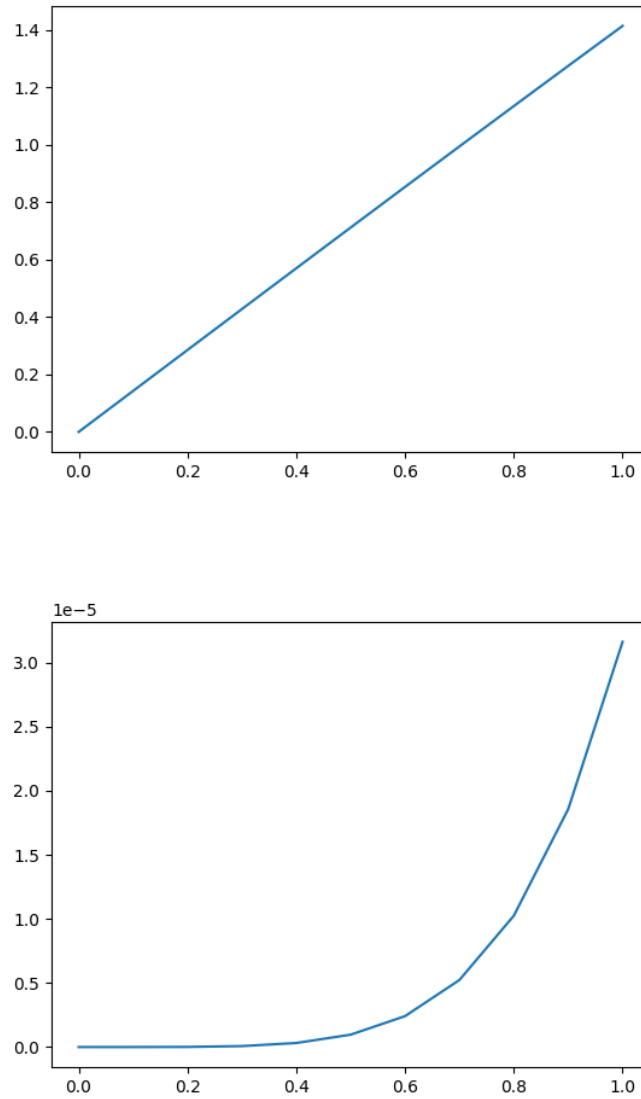


Рис. 1: График погрешности

### 1.3 Код программы

```
import numpy as np
import scipy.integrate as integrate
import matplotlib.pyplot as plt

alphas = np.array([lambda x: x, lambda x: x**3, lambda x: x**5])
betas = np.array([lambda s: 0.6, lambda s: -0.036*s**2, lambda s:
    0.000648*s**3])

def f(x):
    return x

def u(x, cs):
```

```

alpha_values = np.array([alpha(x) for alpha in alphas])
return f(x) + alpha_values.dot(cs)

if __name__ == '__main__':
    fs = np.array([integrate.quad(lambda x: betta(x) * f(x), 0,
        1)[0] for betta in betas])
    A = []
    for beta in betas:
        row = [integrate.quad(lambda x: beta(x) * alpha(x), 0, 1)
            [0] for alpha in alphas]
        A.append(row)
    A = np.array(A)
    A = -1*A
    for i in range(3):
        A[i][i] = 1+A[i][i]

    cs = np.linalg.solve(A, fs)
    interval = np.linspace(0, 1, 11)
    plt.plot(interval, [u(x, cs) for x in interval])
    diff = [u(x, cs) - integrate.quad(lambda s: np.sin(0.6*x*s)/s
        * u(s, cs), 0, 1)[0] - x for x in interval]
    plt.show()

    plt.plot(interval, diff)
    plt.show()

```