



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
**«Дальневосточный федеральный университет»**  
**(ДВФУ)**

---

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ  
ТЕХНОЛОГИЙ**

**Департамент математического и компьютерного моделирования**

**Курсовой проект**  
**«Метод вырожденных ядер для решения уравнения Фредгольма**  
**2-го рода»**

по дисциплине «Математическое и компьютерное моделирование»

Направление подготовки  
01.03.02 «Прикладная математика и информатика»

Выполнил студент  
гр. Б9120-01.03.02миопд  
Агличиев А.О. \_\_\_\_\_  
(ФИО) (подпись)

Проверил доцент, к.ф.-м.н.  
Колобов А.Г. \_\_\_\_\_  
(ФИО) (подпись)

« 23 » января 2023 г.

г. Владивосток  
2024

# Содержание

<b>1</b>	<b>Введение</b>	<b>3</b>
<b>2</b>	<b>Основная часть</b>	<b>4</b>
2.1	Постановка задачи . . . . .	4
2.2	Описание алгоритма решения задачи . . . . .	5
2.3	Описание тестов, использованных для отладки . . . . .	6
2.4	Вычислительные эксперименты . . . . .	8
<b>3</b>	<b>Заключение</b>	<b>11</b>
<b>4</b>	<b>Список использованных источников</b>	<b>12</b>
<b>5</b>	<b>Приложения (тексты программ)</b>	<b>12</b>

# 1 Введение

Объектом исследования являются численные методы решения задач математической физики, а также программное обеспечение, реализующее эти методы.

Цель работы – ознакомиться с численными методами решения задач математической физики решить предложенные типовые задачи, сформулировать выводы по полученным решениям, отметить достоинства и недостатки методов, сравнить удобство использования и эффективность работы каждой использованной программы, приобрести практические навыки и компетенции, а также опыт самостоятельной профессиональной деятельности, а именно:

- создать алгоритм решения поставленной задачи и реализовать его, протестировать программы;
- освоить теорию вычислительного эксперимента; современных компьютерных технологий;
- приобрести навыки представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

Работа над курсовым проектом предполагает выполнение следующих задач:

- дальнейшее углубление теоретических знаний обучающихся и их систематизацию;
- получение и развитие прикладных умений и практических навыков по направлению подготовки;
- овладение методикой решения конкретных задач;
- развитие навыков самостоятельной работы;
- развитие навыков обработки полученных результатов, анализа и осмысления их с учетом имеющихся литературных данных;

- приобретение навыков оформления описаний программного продукта;
- повышение общей и профессиональной эрудиции.

Изученный студентом в ходе работы материал должен способствовать повышению его качества знаний, закреплению полученных навыков и уверенности в выборе путей будущего развития своих профессиональных способностей.

## 2 Основная часть

### 2.1 Постановка задачи

Интегральные уравнения – функциональное уравнение, которое содержит интегральное преобразование над искомой функцией.

В достаточно общем случае линейные интегральные уравнения могут быть представлены в виде

$$g(x)y(x) - \lambda \int_{\Omega} K(x, s)y(s)ds = f(x), x \in Q$$

, где  $K(x, s)$  - ядро,  $f(x)$  - правая часть уравнения с область определения  $Q$ ,  $\lambda$  - параметр уравнения,  $y(s)$  - искомая функция с область определения  $\Omega$ .

Если  $\Omega$  постоянная и  $g(x) \neq 0$ , то интегральное уравнение в этом случае есть уравнение Фредгольма 2-го рода

$$y(x) - \lambda \int_a^b K(x, s)y(s)ds = f(x) \quad (1)$$

В данной курсовой работе решение уравнения Фредгольма 2-го рода будем искать с помощью метода вырожденных ядер.

## 2.2 Описание алгоритма решения задачи

Ядро  $K(x, s)$  называется вырожденным, если оно представимо в виде

$$K(x, s) \approx \sum_{i=1}^n \alpha_i(x) \beta_i(s)$$

, где функции  $\alpha_i(x)$  и  $\beta_i(s)$  линейно независимы на отрезке  $[a, b]$ .

Данный метод основан на том, что для интегрального уравнения с вырожденным ядром может быть получено точное решение. Заменим в 1 ядро  $K(x, s)$  вырожденным

$$y(x) - \lambda \int_a^b \left[ \sum_{i=1}^n \alpha_i(x) \beta_i(s) \right] y(s) ds = f(x) \quad (2)$$

$$y(x) - \lambda \sum_{i=1}^n \alpha_i(x) \int_a^b \beta_i(s) y(s) ds = f(x)$$

Если обозначить

$$\int_a^b \beta_i(s) y(s) ds = C_i$$

, то можно получить следующую форму решения:

$$y(x) = f(x) + \lambda \sum_{i=1}^n C_i \alpha_i(x) \quad (3)$$

Подставляя 3 в 2, получим:

$$C_i - \lambda \sum_{j=1}^n C_j \int_a^b \alpha_j(s) \beta_i(s) ds = \int_a^b \beta_i(s) f(s) ds$$

Если ввести обозначения,

$$A_{ij} = \int_a^b \alpha_j(s) \beta_i(s) ds$$

$$f_{ij} = \int_a^b \beta_i(s) f(s) ds$$

, получим

$$C_i - \lambda \sum_{j=1}^n A_{ij} C_j = f_i$$

Получаем СЛАУ относительно  $C_i$ . Решив эту систему получим решение уравнения в виде 3. Если ядро не является вырожденным, то его можно аппроксимировать таковым и, применив данный метод, получить приближенное решение. В данной работе для аппроксимации ядра будет использовать первые 3 члена разложения в ряд Тейлора и аппроксимация интерполяционным многочленом Лагранжа с помощью 3 точек.

## 2.3 Описание тестов, использованных для отладки

Для тестов было выбрано следующее интегральное уравнения с известным точным решением:

$$y(x) - \int_0^1 x^2 \cos(xs) y(s) ds = x - x \sin x - \cos x + 1, \quad y = x$$

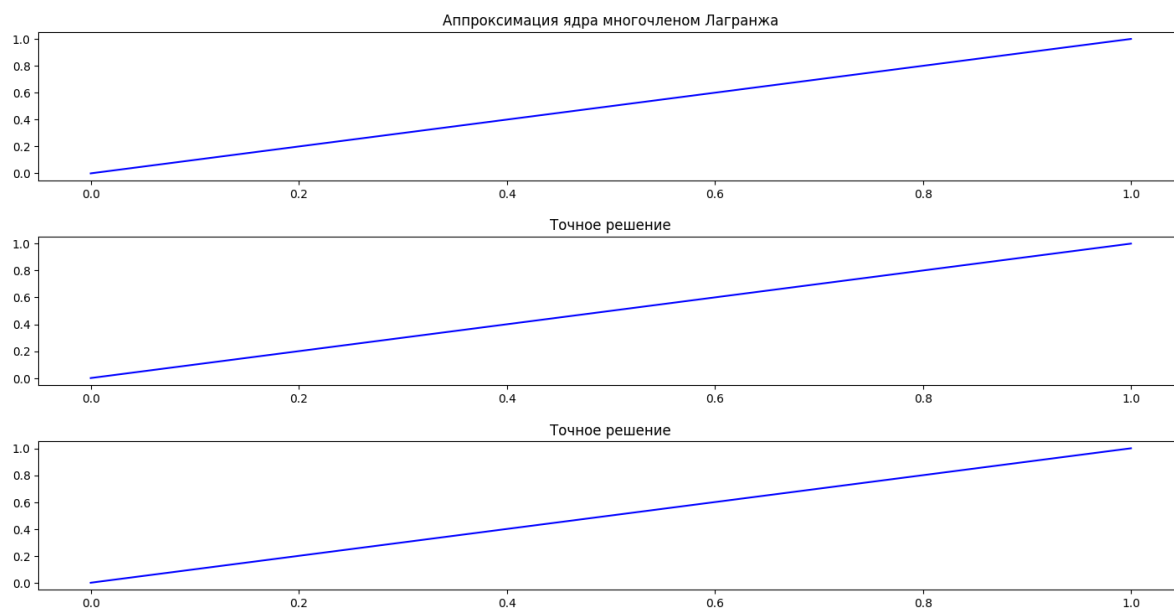


Рис. 1: Графики решения ИУ

$x$	Точное решение	Лагранж	Тейлор
0	0	0	0
0,1	0,1	0,1	0,1
0,2	0,2	0,2	0,2
0,3	0,3	0,29999	0,3
0,4	0,4	0,4	0,4
0,5	0,5	0,5	0,5
0,6	0,6	0,6	0,60001
0,7	0,7	0,69999	0,70002
0,8	0,8	0,8	0,80004
0,9	0,9	0,9	0,90008
1	1	1	1,0001

## 2.4 Вычислительные эксперименты

$$y(x) - \int_0^1 \frac{\sin(0.6xs)}{s} y(s) ds = x$$

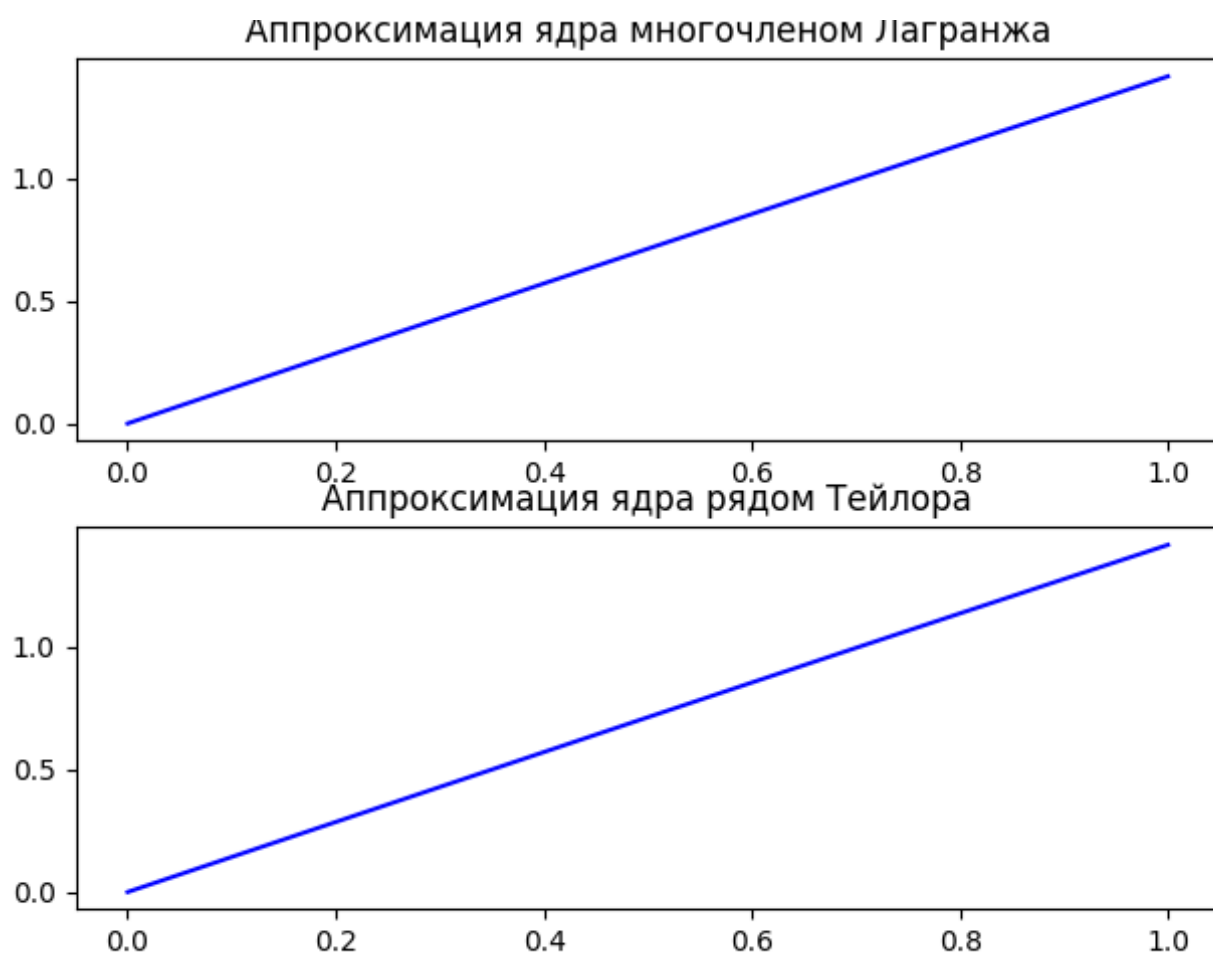


Рис. 2: Графики решения ИУ



$x$	Лагранж	Тейлор
0	0	0
0,1	0,14302	0,14257
0,2	0,28566	0,28506
0,3	0,42793	0,42741
0,4	0,56981	0,56952
0,5	0,56982	0,56952
0,6	0,85247	0,85277
0,7	0,99324	0,93375
0,8	1,13362	1,13420
0,9	1,27362	1,27406
1	1,41326	1,41325

$$y(x) - \int_0^1 \frac{xs}{1 + 0.1xs} y(s) ds = e^{-x}$$

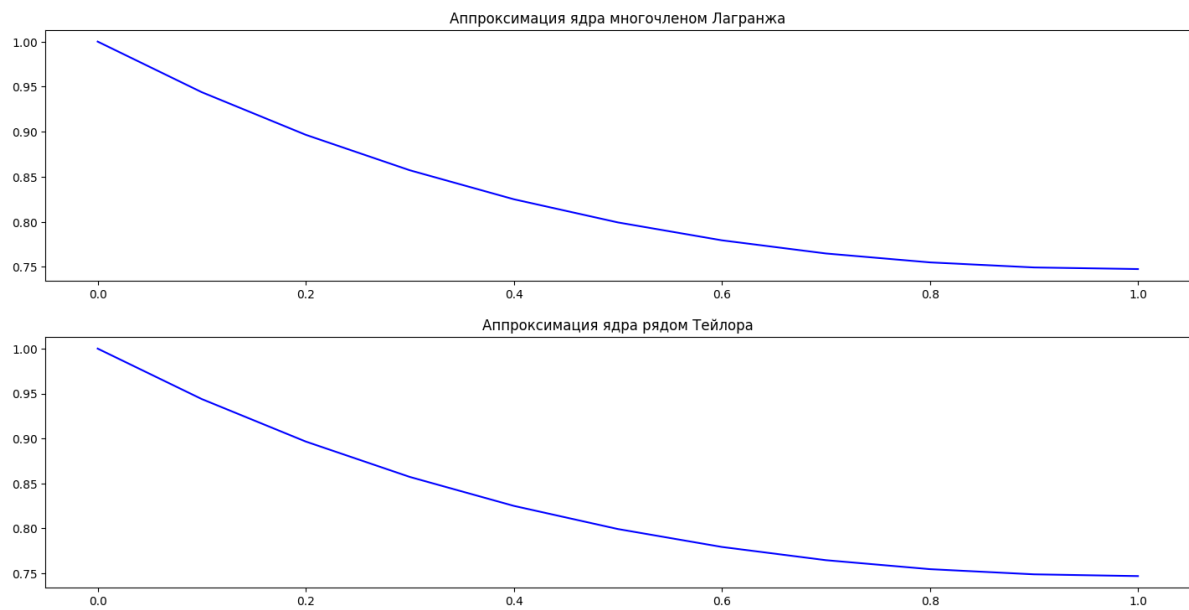


Рис. 3: Графики решения ИУ

$x$	Лагранж	Тейлор
0	1	1
0,1	0,94386	0,94386
0,2	0,89654	0,89653
0,3	0,85718	0,85714
0,4	0,82500	0,82491
0,5	0,79930	0,79913
0,6	0,77942	0,77917
0,7	0,76480	0,76444
0,8	0,75492	0,75444
0,9	0,74930	0,74868
1	0,74751	0,74673

$$y(x) - \int_0^1 (1+s)(e^{0.2xs} - 1)y(s)ds = \frac{1}{x}$$

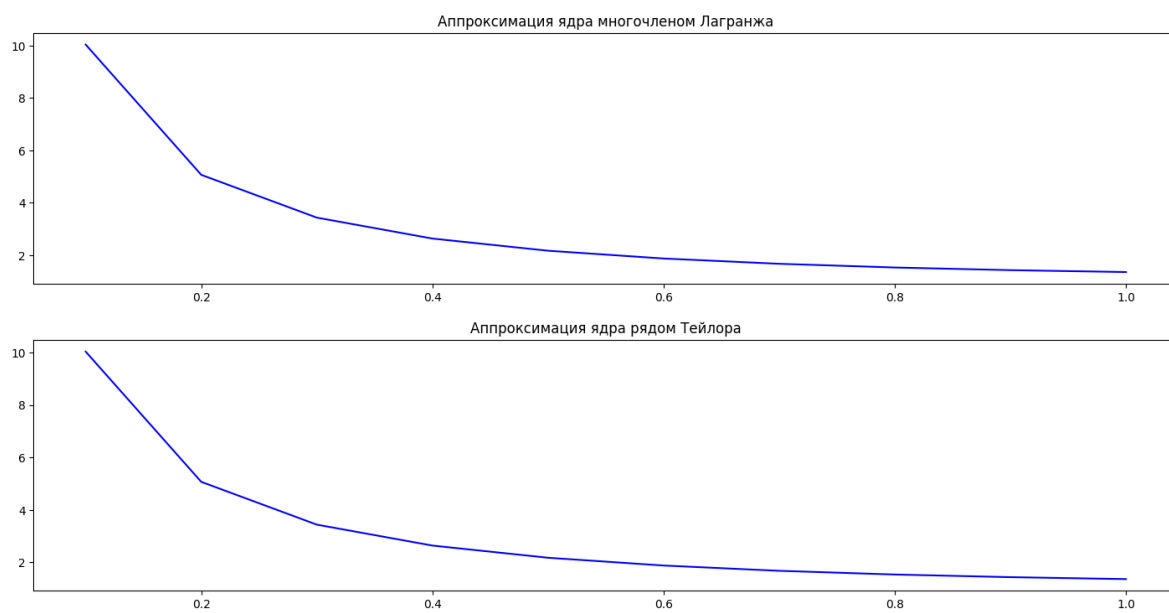


Рис. 4: Графики решения ИУ

$x$	Лагранж	Тейлор
0	$\infty$	$\infty$
0,1	10,03433	10,03437
0,2	5,06910	5,06914
0,3	3,42762	3,43766
0,4	2,63991	2,63993
0,5	2,17596	2,17595
0,6	1,87910	1,87906
0,7	1,67791	1,67784
0,8	1,53667	1,53658
0,9	1,43554	1,435456
1	1,362622	1,36255

### 3 Заключение

В результате работы над курсовым проектом были реализованы метод вырожденных ядер для решения уравнения Фредгольма 2-го рода. Метод были протестирован и отлажены на тестовом примере, а затем применен для вычислительных экспериментов.

Приобрел практические навыки владения:

- современными численными методами решения задач линейной алгебры;
- основами алгоритмизации для численного решения задач линейной алгебры на языке программирования Python 3;
- инструментальными средствами, поддерживающими разработку программного обеспечения для численного решения задач линейной алгебры;

а также навыками представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

## 4 Список использованных источников

1. Верлань А.Ф. Интегральные уравнения: методы, алгоритмы, программы/ А.Ф. Верлань, В.С. Сизиков – Киев: Наукова думка, 1986 г. – 544 с.
2. Колобов А.Г. Лабораторные работы по Численным методам / А.Г. Колобов, Л.А. Молчанова – Владивосток: Изд-во Дальневост. ун-та, 2007 г. - 36 с.

## 5 Приложения (тексты программ)

```
import math

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

def K(x, s):
    return (1+s)*(np.e**(0.2*x*s)-1)

def f(x):
    return 1/x

def lagrange():
    interval = np.linspace(0, 1, 3)
    betas = []
    alphas = []
    numerators = []
    for i in range(len(interval)):
        denominator = 1
        numerators.append([lambda x: 1])
        for j in range(len(interval)):
            if i == j:
                continue
            numerators[i].append(lambda z, bound_j=j: z-interval[bound_j])
            denominator *= (interval[i] - interval[j])

        alphas.append(lambda p, bound_i=i, bound_den=denominator: (math.prod
(num(p) for num in numerators[bound_i]) /
```

```

bound_den))
    betas.append(lambda s, bound_i=i: K(interval[bound_i], s))
return alphas, betas

def solve_system(alphas, betas):
    fs = np.array([integrate.quad(lambda x: beta(x) * f(x), 0, 1)[0] for
beta 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(len(betas)):
        A[i][i] = 1+A[i][i]

    return np.linalg.solve(A, fs)

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

```