Федеральное государственное автономное образовательное учреждение высшего образования **«Национальный исследовательский университет ИТМО»**

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа **№3**

**«Численное интегрирование»**

по дисциплине «Вычислительная математика**»**

Вариант: **1**

**Преподаватель:**   
Рыбаков С.Д.

**Выполнил:**

Алферов Глеб Александрович

**Группа:** Р3207

Санкт-Петербург, 2025 г.

Цель работы: найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

# 1. Вычислительная реализация задачи

1. Вычислить интеграл точно:
2. Вычислить интеграл по формуле Ньютона–Котеса при :

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при :

* **Метод средних прямоугольников**:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

* **Метод трапеций**:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

* **Метод Симпсона**:

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

1. **Сравнить результаты с точным значением интеграла:**

Точное значение интеграла на интервале

1. Метод Ньютона–Котесапри : , значения совпадают.
2. Метод средних прямоугольников при :
3. Метода трапеций при :
4. Метода Симпсонапри : , значения совпадают.
5. **Относительная погрешность вычислений для каждого метода.**
6. Ньютона–Котеса: погрешности нет.
7. Средних прямоугольников:
8. Трапеции:
9. Симпсон: погрешности нет.

# 2. Программная реализация задачи

Метод прямоугольников:

import tasks\_enum  
  
  
def calc\_mid(f, a, b, n):  
 h = (b - a) / n  
 result = 0  
 func = tasks\_enum.Functions.get\_func\_lambda(f)  
 for i in range(n):  
 result += func((a + (i + 0.5) \* h))  
  
 return result \* h  
  
  
def calc\_left(f, a, b, n):  
 h = (b - a) / n  
 result = 0  
 func = tasks\_enum.Functions.get\_func\_lambda(f)  
 for i in range(n):  
 result += func((a + i \* h))  
  
 return result \* h  
  
  
def calc\_right(f, a, b, n):  
 h = (b - a) / n  
 result = 0  
 func = tasks\_enum.Functions.get\_func\_lambda(f)  
 for i in range(1, n+1):  
 result += func((a + i \* h))  
  
 return result \* h

Выберите функцию из списка:

1. -x^3 - x^2 - 2x + 1

2. 5x^2 + 3x

3. cos(x) + x

4. sin(x)

Введите one, two, three или four для выбора: one

Выберите метод решения:

1. Левые прям.

2. Средние прям.

3. Правые прям.

4. Трапеции

5. Симпсона

2

Начальное значение разбиения интервала: 4

Введите точность системы >0 и <1: 0.01

Введите интервал через пробел: 0 2

Значение: -8.6640625

Число разбиения интервала: 32

Метод трапеций:

import tasks\_enum  
  
  
def calc\_trapez(f, a, b, n):  
 h = (b - a) / n  
 func = tasks\_enum.Functions.get\_func\_lambda(f)  
 result = 0.5 \* (func(a) + func(b))  
 for i in range(1, n):  
 result += func(a + i \* h)  
  
 return result \* h

Выберите функцию из списка:

1. -x^3 - x^2 - 2x + 1

2. 5x^2 + 3x

3. cos(x) + x

4. sin(x)

Введите one, two, three или four для выбора: two

Выберите метод решения:

1. Левые прям.

2. Средние прям.

3. Правые прям.

4. Трапеции

5. Симпсона

4

Начальное значение разбиения интервала: 4

Введите точность системы >0 и <1: 0.01

Введите интервал через пробел: 1 2

Значение: 16.169921875

Число разбиения интервала: 16

Метод Симпсона:

import tasks\_enum  
  
  
def calc\_simpson(f, a, b, n):  
 h = (b - a) / n  
 func = tasks\_enum.Functions.get\_func\_lambda(f)  
 result = func(a) + func(b)  
 for i in range(1, n):  
 if i % 2 == 1:  
 result += 4 \* func((a + i \* h))  
 else:  
 result += 2 \* func((a + i \* h))  
  
 return result \* h / 3

Выберите функцию из списка:

1. -x^3 - x^2 - 2x + 1

2. 5x^2 + 3x

3. cos(x) + x

4. sin(x)

Введите one, two, three или four для выбора: three

Выберите метод решения:

1. Левые прям.

2. Средние прям.

3. Правые прям.

4. Трапеции

5. Симпсона

5

Начальное значение разбиения интервала: 4

Введите точность системы >0 и <1: 0.01

Введите интервал через пробел: 1 3

Значение: 3.299633710831437

Число разбиения интервала: 8

# Вывод

В ходе выполнения лабораторной работы мной были реализованы численные методы интегрирования с использованием Python. В ходе вычислительной реализации задачи были рассчитаны интегралы различными методами и проведено сравнение результатов с точными значениями интегралов.