Министерство образования и науки РФ

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

факультет программной инженерии и компьютерной техники

**ЛАБОРАТОРНАЯ РАБОТА №3**

‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №6

*Студент:*

Карандашева Анастасия Денисовна

Группа Р3268

*Преподаватель:*

Машина Екатерина Александровна

Санкт-Петербург, 2024

1. **Цель работы**

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

1. **Порядок выполнения работы**

Задание 1: вычислительная часть

1.  Найти точное решение определённого интеграла

2. Найти решение того же самого интеграла методом трапеций, методом Симпсона и методом средних прямоугольников, посчитать погрешность для каждого из этих методов.

Задание 2: Программная реализация поиска определённого интеграла с заданной точностью для выбранной функции одним из 5 методов на выбор пользователя

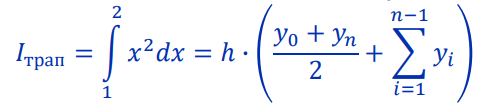
- Метод трапеций

- Метод Симпсона

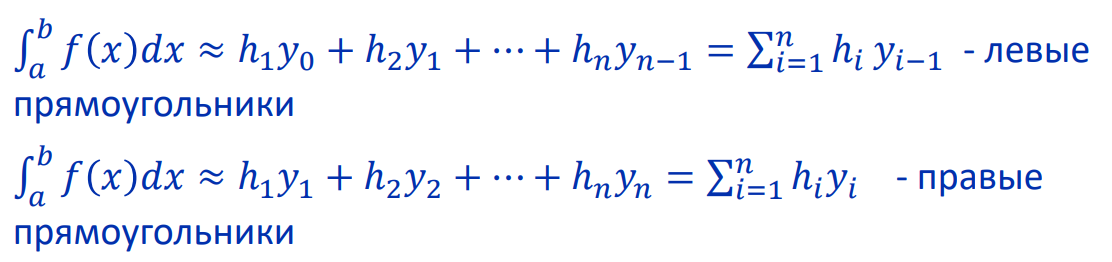
- Метод правых прямоугольников  
- Метод средних прямоугольников

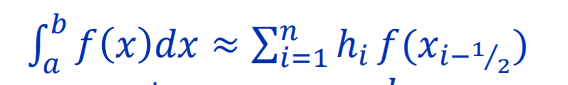
  - Метод левых прямоугольников

1. **Рабочие формулы**

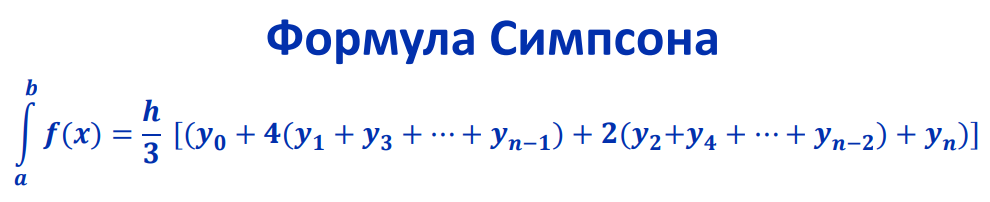
Метод трапеций: ****

Метод прямоугольников:

****

****

Формула Симпсона:

****

1. **Вычислительная часть**
2. Точное вычисление интеграла

= = (1,2) =

1. Метод Ньютона-Котеса, n=6

f(x0) = f(a) = f(1) = 5

Разбиваем (1,2) на 6 равных отрезков

f(x1) = f(1.166666667) = 9.06944

f(x2) = 14

f(x3) = 19.875

f(x4) = 26.77778

f(x5) = 34.79167

f(x6) = f(2) =44

Погрешность : R = |21.41667 – 21.4166667| = 0.0000033

1. Метод средних прямоугольников

n=10; h =

|  |  |  |
| --- | --- | --- |
| i | x | f(x) |
| 1 | 1,05 | 6,135375 |
| 2 | 1,15 | 8,625125 |
| 3 | 1,25 | 11,421875 |
| 4 | 1,35 | 14,543625 |
| 5 | 1,45 | 18,008375 |
| 6 | 1,55 | 21,834125 |
| 7 | 1,65 | 26,038875 |
| 8 | 1,75 | 30,640625 |
| 9 | 1,85 | 35,657375 |
| 10 | 1,95 | 41,107125 |

Погрешность: R=|21.4166667 – 21.40125|=0,0154167

1. Метод трапеций n=10, h=0,1

|  |  |  |
| --- | --- | --- |
| i | x | f(x) |
| 0 | 1 | 5 |
| 1 | 1,1 | 7,343 |
| 2 | 1,2 | 9,984 |
| 3 | 1,3 | 12,941 |
| 4 | 1,4 | 16,232 |
| 5 | 1,5 | 19,875 |
| 6 | 1,6 | 23,888 |
| 7 | 1,7 | 28,289 |
| 8 | 1,8 | 33,096 |
| 9 | 1,9 | 38,327 |
| 10 | 2 | 44 |

Погрешность: |21,4475-21.4166667 |= 0,0308333

1. Метод Симпсона n=10 h=0,1

Погрешность : R=|21.4166667 -21.4166667 |= 0

1. **Листинг программы**

import numpy as np

# Функции для интегрирования

def f1(x):

    return 3 \* x \*\* 3 + 5 \* x \*\* 2 + 3 \* x - 6

def f2(x):

    return np.sin(x)

def f3(x):

    return np.exp(x)

def f4(x):

    return np.cos(x)

def f5(x):

    return np.sqrt(x)

# Методы интегрирования

def left\_rectangle\_method(f, a, b, n):

    h = (b - a) / n

    integral = sum([f(a + i \* h) for i in range(n)])

    return h \* integral

def right\_rectangle\_method(f, a, b, n):

    h = (b - a) / n

    integral = sum([f(a + (i + 1) \* h) for i in range(n)])

    return h \* integral

def middle\_rectangle\_method(f, a, b, n):

    h = (b - a) / n

    integral = sum([f(a + (i + 0.5) \* h) for i in range(n)])

    return h \* integral

def trapezoidal\_method(f, a, b, n):

    h = (b - a) / n

    integral = sum([(f(a + i \* h) + f(a + (i + 1) \* h)) for i in range(n)]) / 2

    return h \* integral

def simpson\_method(f, a, b, n):

    h = (b - a) / n

    x = np.linspace(a, b, n + 1)

    integral = (h / 6) \* (f(a) + f(b) + 2 \* sum([4 \* f(x[i]) for i in range(1, n, 2)]) + 4 \* sum([2 \* f(x[i]) for i in range(2, n - 1, 2)]))

    return integral

# Правило рунге для проверки точности

def runge\_rule(method, f, a, b, tol, max\_intervals):

    n = 1

    prev\_integral = method(f, a, b, n)

    n \*= 2

    integral = method(f, a, b, n)

    while n <= max\_intervals and abs(integral - prev\_integral) >= tol:

        prev\_integral = integral

        n \*= 2

        integral = method(f, a, b, n)

    return integral, n

# Функции и методы

functions = {1: f1, 2: f2, 3: f3, 4: f4, 5: f5}

methods = {

    1: left\_rectangle\_method,

    2: right\_rectangle\_method,

    3: middle\_rectangle\_method,

    4: trapezoidal\_method,

    5: simpson\_method

}

def main():

    print("Выбери функцию для интегрирования:")

    print("1. 3x^3 + 5x^2 + 3x - 6")

    print("2. sin(x)")

    print("3. e^x")

    print("4. cos(x)")

    print("5. sqrt(x)")

    choice = int(input())

    if choice not in functions:

        raise ValueError("Введён неверный номер функции")

    f = functions[choice]

    a = float(input("Введи левый предел интегрирования: "))

    b = float(input("Введи правый предел интегрирования: "))

    tolerance = float(input("Задай точность вычисления: "))

    initial\_intervals = 4

    print("Выбери метод интегрирования:")

    print("1. Метод левых прямоугольников")

    print("2. Метод правых прямоугольников")

    print("3. Метод средних прямоугольников")

    print("4. Метод трапеций")

    print("5. Метод Симпсона")

    method\_choice = int(input())

    integral, num\_intervals = runge\_rule(methods[method\_choice], f, a, b, tolerance, max\_intervals)

    print("Значение интеграла:", integral)

    print("Число разбиения интервала:", num\_intervals)

try:

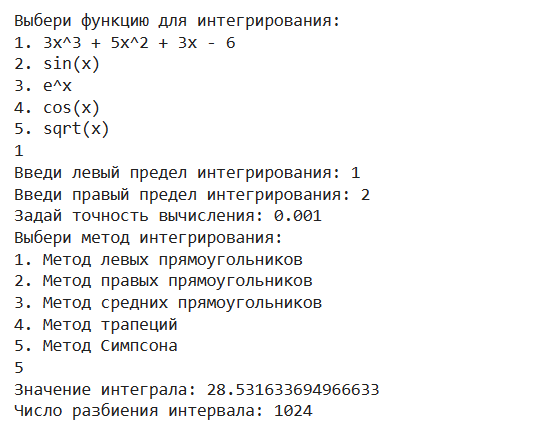
    main()

except ValueError as e:

    print("Ошибка: ", e)

except KeyboardInterrupt as e:

1. print(e)
2. **Результаты выполнения программы**



1. **Выводы**

В ходе выполнения лабораторной работы были изучены численные методы нахождения определённого интеграла. Реализована программа на python, позволяющая вычислять определённый интеграл заданной функции методами левых, правых, средних прямоугольников, методом трапеций или методом Симпсона с заданной точностью, которая достигается с помощью применения правила Рунге, и пределами интегрирования. В качестве результата получены значения интегралов и количество разбиений интервала для достижения заданной пользователем точности.