МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

**«Московский технический университет связи и информатики»**

Кафедра «Информатика»

**Отчет по лабораторной работе №4**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Тогузов А. А.

Вариант 24.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

Задания для численного интегрирования:

* – подынтегральная функция;
* a=-1, b=2–пределы интегрирования;
* методы интегрирования – трапеций, средних прямоугольников;
* начальный шаг интегрирования h0=0.75.

Вычисление интегралов с шагом  и  ( и ) и оценка его погрешности по правилу Рунге

Правило Рунге применяют для вычисления погрешности путём двойного просчёта интеграла с шагами h/2 и h, при этом погрешность вычисляется по формуле .

Полагают, что интеграл вычислен с точностью Е, если  тогда , где  – уточненное значение интеграла, p – порядок метода.

Вычислим интеграл по формуле

трапеций и оценим погрешность интегрирования методом двойного просчёта:



*I = - 0.049 = 6.689*

средних прямоугольников и оценим погрешность интегрирования методом двойного просчёта:



*I = 6.663 + 0.024 = 6.687*

##### Реализация данных методов в Python

Метод трапеций.

import numpy as np

from scipy.integrate import quad

import math

def f(x):

    return math.sqrt(1 + x\*\*2) + math.exp(-x)

a = -1

b = 2

def trap(n):

    summ = 0

    for i in range(1, n):

        summ += f(a + i\*(b-a)/n)

    return (b-a)/(2\*n)\*(f(a) + f(b) + 2\*summ)

error = 10\*\*9

I\_n = -1

I\_2n = -1

exps = [-2, -3, -4]

for exp in exps:

    required\_error = 10\*\*exp

    n = 1

    error = 10\*\*9

    while (abs(error) > required\_error):

        n \*= 2

        I\_n = trap(n)

        I\_2n = trap(2\*n)

        error = (I\_2n - I\_n)/3

    print("Необходимая погрешность: ", required\_error)

    print("Значение интеграла: ", I\_2n + error)

    print("Значение погрешности: ", error)

    print()

|  |  |
| --- | --- |
| 10^-2 | -0.0030637622970433043 |
| 10^-3 | -0.0007661360578922564 |
| 10^-4 | -4.788732439623781e-05 |

Рисунок 1 – Результат работы кода для первого метода

Метод средних прямоугольников.

import math

def f(x):

    return math.sqrt(1 + x\*\*2) + math.exp(-x)

a = -1

b = 2

def midpoint(n):

    h = (b - a) / n

    summ = 0

    for i in range(n):

        x\_midpoint = a + (i + 0.5) \* h

        summ += f(x\_midpoint)

    return h \* summ

error = 10\*\*9

I\_n = -1

I\_2n = -1

exps = [-2, -3, -4]

for exp in exps:

    required\_error = 10\*\*exp

    n = 1

    error = 10\*\*9

    while abs(error) > required\_error:

        n \*= 2

        I\_n = midpoint(n)

        I\_2n = midpoint(2 \* n)

        error = (I\_2n - I\_n) / 3

    print("Необходимая погрешность: ", required\_error)

    print("Значение интеграла: ", I\_2n + error)

    print("Значение погрешности: ", error)

    print()

|  |  |
| --- | --- |
| 10^-2 | 0.006115046649274388 |
| 10^-3 | 0.00038304357680522355 |
| 10^-4 | 9.577159174837628e-05 |

Рисунок 2 – Результат работы кода для второго метода