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 заданию №4**

**по дисциплине**

**«Численные методы»**

Выполнил: студент гр. БЭИ2202

Кулешов А. С.

Вариант 16.

Проверил: доц. каф. «Информатика»

Мацкевич А. Г.

Москва, 2023 г.

1. **Задания для численного интегрирования:**

* – подынтегральная функция;
* **a=1, b=2**–пределы интегрирования;
* методы интегрирования – трапеций, Симпсона;
* начальный шаг интегрирования **h0=0.25.**

1. **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:



*2.365+2.222 + 2\*(2.677+2.765+2.613))=2,587*

*2.365+2.222 + 2\*(2.677+2.765+2.613+2.548+2.750+2.719+2.447))=2,601*

*I = 2.601 + 0.0046 = 2.6056*

* **Симпсона** и оценим погрешность интегрирования методом **двойного просчета**:

 где 

##### **3. Вычисление определенных интегралов в Python**

from scipy.integrate import quad

from math import sin, sqrt

def integrand(x):

    return 4\*sin(x) - sqrt(x)

result, error = quad(integrand, 1, 2)

print("Значение интеграла:", result)

print("Погрешность:", error)



Рисунок 1 – вычисление интеграла при помощи ЯП Python

Теперь реализую код на Python, который вычисляет интеграл методом трапеций до определённой точности. В соответствии с условием погрешность

import numpy as np

def f(x):

    return 4\*sin(x) - sqrt(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 = np.arange(-2,-6,-0.1)

dots = []

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("Число точек для достижения точности: ", n)

  dots.append(n)

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

  print("Значение погрешности: ", error)

  print()

import matplotlib.pyplot as plt

plt.plot(exps, dots)

plt.xlabel('Значение экспоненты погрешности')

plt.ylabel('Число точек для достижения нужной точности')

plt.title('График')

plt.show()

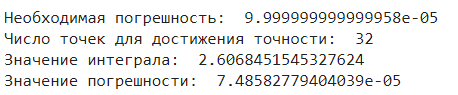


Рисунок 2 – результат работы кода для расчёта интеграла

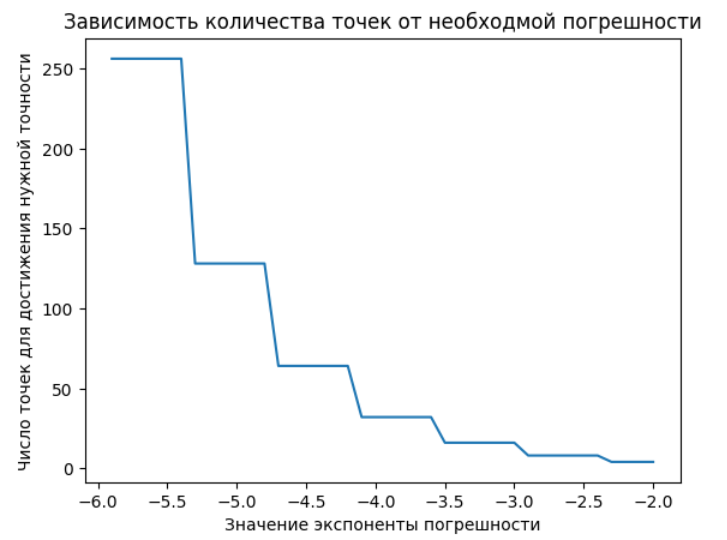


Рисунок 3 – зависимость количества точек от необходмой погрешности