**МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ**

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ**

**ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ**

**Лабораторная работа №3**

**«Численные методы»**

**Вариант 8**

Ёды Никиты Дмитриевича

студента 3 курса, 6 группы

специальность «прикладная математика»

Преподаватель:

Будник А.М.

**Постановка задачи**

Для вычисления интеграла

с точностью необходимо:

1. Применить правило Рунге, используя составную квадратурную формулу правых прямоугольников. Определить величину шага разбиения исходного отрезка интегрирования, достаточного для достижениея точности .
2. Пользуясь выражением для погрешности интегрирования. Определить шаги h в двух нижеуказанных квадратурных формулах, которые обеспечат требуемую точность результата:

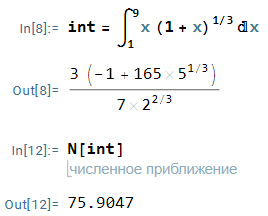
а) составная квадратурная формула средних прямоугольников.

б) составная квадратурная формула Симпсона.

1. Применить квадратурную формулу НАСТ Гаусса при . Оценить погрешность интегрирования через формулу остаточного члена
2. Провести сравнительный анализ полученных в п.п.1-3 результатов.

**Решение интеграла**

Листинг программы Wolfram Mathematica:



Результат: *75.9047*

**Правило Рунге**

Составная квадратурная формула правых прямоугольников для разбиения отрезка [a, b] в n частей имеет вид:

Правило Рунге позволяет оценить погрешность численного интегрирования и улучшить точность. Согласно этому правилу, если — значе-ние интеграла, полученное при n разбиениях, а — значение интеграла, полученное при 2n разбиениях, то:

*=> ;*

**Реализация**

|  |
| --- |
| import numpy as np  def f(x):      return x\*(1+x)\*\*(1/3)  def composite\_right\_rectangle(f, a, b, n):      h = (b - a) / n      x = np.linspace(a + h, b, n)      return h \* np.sum(f(x))  def runge\_rule(f, a, b, epsilon):      n = 10      I\_n = composite\_right\_rectangle(f, a, b, n)      while True:          n \*= 2          I\_2n = composite\_right\_rectangle(f, a, b, n)          if np.abs(I\_2n - I\_n) <= epsilon:              break          I\_n = I\_2n      h = (b - a) / n      return I\_2n, n, h  a = 1  b = 9  epsilon = 1e-5  integral, n, h = runge\_rule(f, a, b, epsilon)  print(f"Значение интеграла: {integral:.6f}")  print(f"Количество разбиений: {n}")  print(f"Шаг для разбиения: {h:.6f}") |

**Вывод по полученным результатам**

Значение интеграла:

*75.904679*

Количество разбиений:

*10485760*

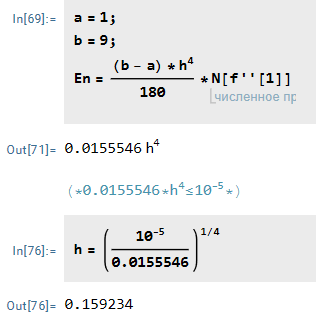
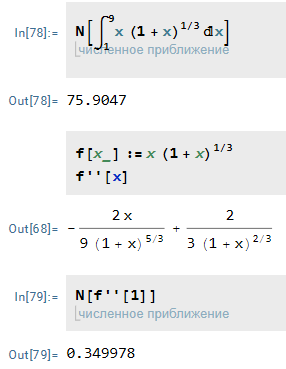
Шаг для разбиения:

*0.000001*

**Определение шагов**

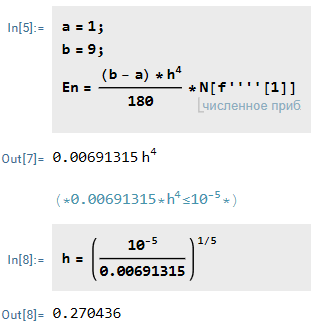
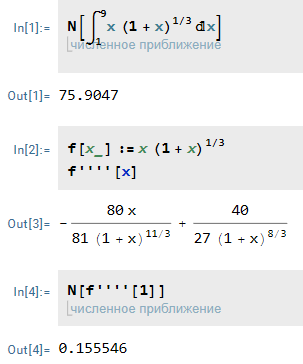
а) составная квадратурная формула средних прямоугольников

Погрешность составной квадратурной формулы средних прямоугольников оценивается как:



Отсюда следует, что *.159234*

б) составная квадратурная формула Симпсона

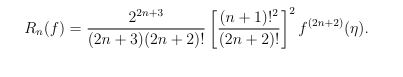


Отсюда следует, что

**Формула НАСТ Гаусса**

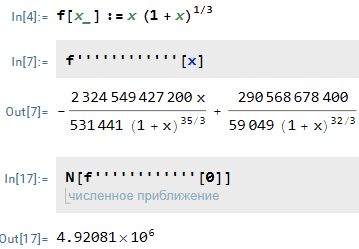






**Реализация**

Вычисляю значение 12-ой производной



Реализация метода на Python:

|  |
| --- |
| import numpy as np  import math  def f(x):      return x\*(1 + x)\*\*1/3  # Узлы и веса для квадратурной формулы Гаусса при n=4  nodes = np.array([-0.8611363115940526, -0.3399810435848563, 0.3399810435848563, 0.8611363115940526])  weights = np.array([0.34785484513745385, 0.6521451548625461, 0.6521451548625461, 0.34785484513745385])  def transform(t):  return 8\*t + 10  def residual\_error(f\_2n\_2\_eta):      return 2 / ((2\*4 + 3) \* math.factorial(2\*4 + 2)) \* (math.factorial(4 + 1) / (2 \* math.factorial(2\*4 + 2))\*\*2) \* f\_2n\_2\_eta  integral = 0.9 \* sum(weights[i] \* f(transform(nodes[i])) for i in range(4))  print(f"Значение интеграла: {integral:.6f}")  f\_2n\_2\_eta = 4.92081 \* 10\*\*6  error = residual\_error(f\_2n\_2\_eta)  print(f"Оценка погрешности: {error:.6e}") |

**Вывод по полученным результатам**

Значение интеграла:

*78.800000*

Оценка погрешности:

*5.617021e-13*

**Анализ методов**

|  |  |  |  |
| --- | --- | --- | --- |
|  | h | Значение интеграла | Погрешность |
| Правые пр-ки | 0.000001 | 75.904679 |  |
| Средние пр-ки | .159234 | - |  |
| Симпсон |  | - |  |
| НАСТ Гаусс | - | 78.800000 | 5.61\* |

Более точный метод - метод НАСТ Гаусса, т.к. во всех остальных случаях погрешность .

|  |  |
| --- | --- |
| Формула правых прямоугольников | Метод вычисляет площадь под кривой, используя значения функции на правых концах интервалов. Погрешность оценивается как O(h^2), где h - шаг разбиения. |
| Составная квадратурная формула средних прямоугольников | Метод вычисляет площадь под кривой, используя значения функции в серединах интервалов. Погрешность оценивается как O(h^2). |
| Составная квадратурная формула Симпсона: | Составная квадратурная формула Симпсона: Метод использует аппроксимацию кривой параболами, проходящими через три точки. Погрешность оценивается как O(h^4). |
| Формула НАСТ Гаусса | Метод использует определенные веса и узлы для аппроксимации интеграла. Он обеспечивает точные результаты для полиномов до 7-й степени. Погрешность оценивается как O(h^10). |