## Национальный исследовательский университет «МЭИ» (Московский Энергетический Институт)

**Кафедра математического и компьютерного моделирования**

## Численные методы

## Лабораторная работа №5:

## «ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ»

*Выполнил:* Солонин Е. В. А-14-19

*Преподаватель:* Амосова О. А.

Вариант 41

2021 Москва

**Постановка задачи:**

**Задача 5.1.** Вычислить значение интеграла , где , с помощью квадратурных формул левых прямоугольников, Гаусса и по формуле индивидуального варианта.

**ПОРЯДОК РЕШЕНИЯ ЗАДАЧИ**

1. Вычислить аналитически значение интеграла .

2. Используя выражение для остаточного члена интегрирования (см. **ПРИЛОЖЕНИЕ 5.B**), оценить шаг интегрирования h , при котором величина погрешности квадратурной формулы будет меньше . Вычислить число отрезков разбиения n.

3. Вычислить значение интеграла  по составной квадратурной формуле левых прямоугольников с найденным шагом . Найти величину погрешности .

4. Проделать те же действия (п. 2 - 3) для вычисления интеграла  по квадратурной формуле из индивидуального варианта.

5. Основываясь на заданной степени многочлена m, выбрать число узлов для квадратуры Гаусса, обеспечивающее вычисление интеграла без погрешности. Вычислить интеграл .

6. Результаты внести в таблицу 5.1

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 5.1.41 | 4.8 | 1.5 | 6.3 | –2.7 | 3.7 | 4.4 |

|  |  |
| --- | --- |
| 5,11,17,23,29,35,**41**,47,53,59 | (7) формула Милна |

**Теоретический материал:**



Формула **левых** прямоугольников:  остаточный член 

Формула **Милна**: 

остаточный член 

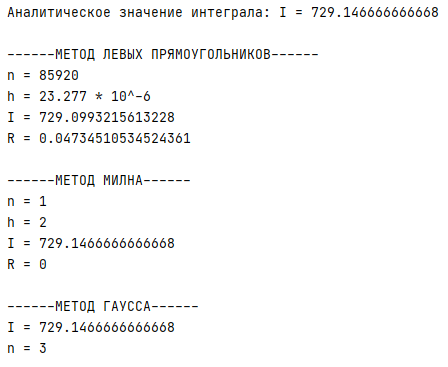
Формула **Гаусса**: 

|  |  |
| --- | --- |
|  | Число узлов 3 |
|  |  |
|  |  |
|  | 0 |
|  |  |
|  |  |
|  |  |

**Код программы** *(пояснения к коду красным цветом)***:**

import numpy  
import scipy  
from scipy import optimize  
  
  
def f(x):  
 result = 0  
 for i in range(len(coefs)):  
 result += coefs[i] \* x\*\*i  
 return result  
  
  
*# производная функции в точке x*def df\_dx(x):  
 result = 0  
 for i in range(1, len(coefs)):  
 result += coefs[i] \* i \* x \*\* (i - 1)  
 return result  
  
  
*# нахождение интеграла по формуле левых прямоугольников с заданным шагом h и числом разбиений n*def integral\_left\_rect(h, n):  
 f\_i = []  
 for x in numpy.arange(a, b, h):  
 f\_i.append(f(x))  
 result = 0  
 for i in range(n - 1):  
 result += f\_i[i]  
 return result \* h  
  
  
*# нахождение интеграла по формуле Милна с заданным шагом h и числом разбиений n*def integral\_miln(h, n):  
 *# берем только граничные точки* x = [a, b]  
 result = 0  
 for i in range(1, n + 1):  
 result += 7 \* f(x[i - 1])\  
 + 32 \* f(x[i - 1] + h / 4)\  
 + 12 \* f(x[i - 1] + h / 2)\  
 + 32 \* f(x[i] - h / 4)\  
 + 7 \* f(x[i])  
 return result \* h / 90  
  
  
*# функция, выводящая результаты, относящиеся к формуле левых прямоугольников*def left\_rect():  
 *# максимум производной на [a, b]* M1 = -scipy.optimize.minimize\_scalar(lambda x: -numpy.abs(df\_dx(x)), bounds=[a, b], method=**'bounded'**).fun  
 *# оценка шага h* h\_estimate = 2 \* epsilon / (M1 \* (b - a))  
 *# число разбиений n - округляем в большую сторону* n = int(numpy.ceil((b - a) / h\_estimate))  
 h = (b - a) / n  
 I = integral\_left\_rect(h, n)  
 print(**'------МЕТОД ЛЕВЫХ ПРЯМОУГОЛЬНИКОВ------'**)  
 print(**'n ='**, n)  
 print(**'h ='**, **"{:.3f}"**.format(h \* 10 \*\* 6), **'\* 10^-6'**)  
 print(**'I ='**, I)  
 print(**'R ='**, numpy.abs(integral - I))  
 print()  
  
  
*# функция, выводящая результаты, относящиеся к формуле Милна*def miln():  
 *# максимум шестой производной равен нуля для данного многочлена  
 # поэтому отрезок не разбиваем* n = 1  
 h = b - a  
 I = integral\_miln(h, n)  
 print(**'------МЕТОД МИЛНА------'**)  
 print(**'n ='**, n)  
 print(**'h ='**, h)  
 print(**'I ='**, I)  
 print(**'R ='**, 0)  
 print()  
  
  
*# функция,вычисляющая интеграл и выводящая результаты, относящиеся к формуле Гаусса*def gauss():  
 n = 3  
 A = [5/9, 8/9, 5/9]  
 t = [-numpy.sqrt(3/5), 0, numpy.sqrt(3/5)]  
 result = 0  
 for i in range(n):  
 result += A[i] \* f((a + b) / 2 + (b - a) / 2 \* t[i])  
 result \*= (b - a) / 2  
 print(**'------МЕТОД ГАУССА------'**)  
 print(**'I ='**, result)  
 print(**'n ='**, n)  
 print()  
  
  
coefs = [4.8, 1.5, 6.3, -2.7, 3.7, 4.4]  
integral = 729.146666666668  
a = 1  
b = 3  
epsilon = 0.05  
  
print(**'**\n**Аналитическое значение интеграла: I ='**, integral, **'**\n**'**)  
  
left\_rect()  
miln()  
gauss()

**Результаты выполнения:**

****Вычисляется интеграл c точностью 0.05

|  |  |  |
| --- | --- | --- |
| **Найденное точное значение интеграла** | Число разбиений отрезка n  Шаг интегрирования h | Значение интеграла, вычисленное по составной формуле  Величина погрешности интеграла, вычисленного по составной формуле |
| **Метод Левых прямоугольников** |  |  |
| Метод Милна |  |  |
| Метод Гаусса | Число узлов квадратуры |  |

**Анализ результатов:**

Для метода Милна , поэтому сразу получаем n, h и R.

Формула Гаусса точна для многочленов степени 2N+1 при N+1 узлах. Нехитрым способом выясняем что N=2, а число узлов N=3. Коэффициенты A, t берем из таблицы (см. теор.материал)

­­­­­­\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Постановка задачи:**

**Задача 5.2.** Вычислить интеграл  с точностью 

**ПОРЯДОК РЕШЕНИЯ ЗАДАЧИ**

1. Вычислить интеграл  с помощью средствпакета PYTHON.

2. Составить программу, содержащую следующие разделы:

а) процедуру-функцию, вычисляющую интеграл по составной квадратурной формуле из индивидуального варианта  с заданным шагом h.

б) подпрограмму, вычисляющую значение интеграла с заданной точностью ; оценку погрешности производить на основе правила Рунге.

с) вычисление уточненного значение интеграла и величину погрешности

Результатом работы программы должны быть следующие величины:

n- число разбиений отрезка интегрирования, при котором заданная точность

достигнута,

 и - - полученные значения интеграла при шагах и  соответственно,

 и  - величины абсолютных погрешностей,

 и  - величины для уточненного значения интеграла.

3.Вычислить интеграл по программе и полученные данные свести в таблицу 5.2 .

|  |  |  |
| --- | --- | --- |
| 5.2.41 |  | [0,10] |

|  |  |
| --- | --- |
| 5,11,17,23,29,35,**41**,47,53,59 | (7) формула Милна |

**Теоретический материал:**



Формула **Милна**: 

остаточный член 

**Правило Рунге**:  где  — порядок точности формулы.

**Уточнение по Рунге**:  где  — порядок точности формулы.

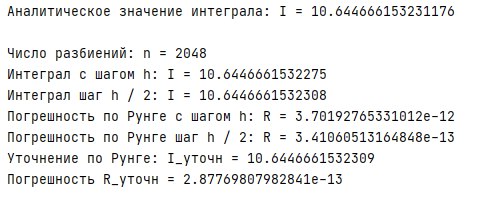
**Код программы** *(пояснения к коду красным цветом)***:**

import numpy  
import scipy  
from scipy import integrate  
from sympy import \*  
  
  
def f(x):  
 return sqrt(x) \* (sin(3 \* x)) \*\* 2  
  
  
*# оценка погрешности по правилу Рунге*def Runge(h):  
 return (integral\_miln(h / 2) - integral\_miln(h)) / (2 \*\* 6 - 1)  
  
  
*# вычисление интеграла по методу Милна с шагом h*def integral\_miln(h):  
 n = int(numpy.ceil((b - a) / h))  
 h = (b - a) / n  
 x = []  
 for x\_i in numpy.arange(a, b + h \* 0.01, h):  
 x.append(x\_i)  
  
 result = 0  
 for i in range(1, n + 1):  
 result += 7 \* f(x[i - 1])\  
 + 32 \* f(x[i - 1] + h / 4)\  
 + 12 \* f(x[i - 1] + h / 2)\  
 + 32 \* f(x[i] - h / 4)\  
 + 7 \* f(x[i])  
 return result \* h / 90  
  
  
*# вычисление интеграла по методу Милна с точностью epsilon  
# оценка погрешности по Рунге  
# уточнение по Рунге  
# погрешность с учетом уточнения*def miln():  
 *# подбираем шаг h, при котором будет достигнута точность epsilon* h = (b - a) / 2  
 while abs(Runge(h)) > epsilon:  
 h /= 2  
 h = h / 2  
  
 *# число разбиений n - округляем в большую сторону* n = int(numpy.ceil((b - a) / h))  
 I\_h = integral\_miln(h)  
 I\_h\_2 = integral\_miln(h / 2)  
 I\_clarify = I\_h\_2 + Runge(h)  
  
 print(**'Число разбиений: n ='**, n)  
 print(**'Интеграл с шагом h: I ='**, I\_h)  
 print(**'Интеграл шаг h / 2: I ='**, I\_h\_2)  
 print(**'Погрешность по Рунге с шагом h: R ='**, abs(I\_h - integral))  
 print(**'Погрешность по Рунге шаг h / 2: R ='**, abs(I\_h\_2 - integral))  
 print(**'Уточнение по Рунге: I\_уточн ='**, I\_clarify)  
 print(**'Погрешность R\_уточн ='**, abs(I\_clarify - integral))  
  
  
a = 0  
b = 10  
epsilon = 10 \*\* (-12)  
*# вычисление интеграла с помощью пакетных средств*integral = scipy.integrate.quad(f, a, b, epsabs=epsilon, epsrel=epsilon)[0]  
  
print(**'**\n**Аналитическое значение интеграла: I ='**, integral, **'**\n**'**)  
miln()

**Результаты выполнения:**

Вычисляется интеграл c точностью 

|  |  |
| --- | --- |
|  | **Метод Милна** |
| Число разбиений отрезка |  |
| Интеграл с шагом h |  |
| Интеграл с шагом h/2 |  |
| Погрешность по Рунге с шагом h |  |
| Погрешность по Рунге с шагом h/2 |  |
| Уточнение по Рунге |  |
| Погрешность для уточненного значения |  |

****