A picture containing logo

Description automatically generated

Лабораторная работа №3

по Вычислительной Математики

Вариант 13

Выполнил:

Пурэвсурэн Билгуун

Группа Р3213

Преподователь:

Малышева Татьяна Алексеевна

г. Санкт-Петербург 2022г

1. **Цель работы:**

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

Изучить численные методы интегрирования и реализовать некоторые из них средствами программирования.

1. **Задание лабораторной работы:**

1. Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.

2. Пределы интегрирования задаются пользователем.

3. Точность вычисления задается пользователем.

4. Начальное значение числа разбиения интервала интегрирования: n=4.

5. Ввод исходных данных осуществляется с клавиатуры.

**Программная реализация задачи:**

1. Реализовать в программе методы по выбору пользователя, исходя из варианта:

• Метод прямоугольников (3 модификации: левые, правые, средние)

• Метод трапеций

• Метод Симпсона

2. Методы должны быть оформлены в виде отдельной(ого) функции/класса.

3. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.

4. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.

5. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

**Вычислительная реализация задачи:**

1. Вычислить интеграл, приведенный в таблице 1 (столбец 3), точно.

2. Вычислить интеграл по формуле Ньютона – Котеса при n = 6.

3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при n = 6.

4. Сравнить результаты с точным значением интеграла.

5. Определить относительную погрешность вычислений.

6. В отчете ***отразить последовательные вычисления***.

1. **Вычислительная часть:**

1. Вычислить интеграл, приведенный в таблице 1 (столбец 3), точно.

Table

Description automatically generated 2. Вычислить интеграл по формуле Ньютона – Котеса при n = 6.

Интеграл = -76.980247

3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при n = 6.

|  |  |  |
| --- | --- | --- |
| Ср. Прямоугольн | | -81.019 |
| Трапеций |  | -81.96 |
| Симпсона |  | -81.3333 |

|  |  |  |
| --- | --- | --- |
| Ньютон-Котеса | | 4.3530531 |
| Ср-прямоуг |  | -0.3143 |
| Трапеций |  | 0.6267 |
| Симпсона |  | 0 |

4. Сравнить результаты с точным значением интеграла.

5. Определить относительную погрешность вычислений.

|  |  |  |  |
| --- | --- | --- | --- |
| Ньютон-Котеса | | 5.35211664 | % |
| Ср-прямоуг |  | 0.38643458 | % |
| Трапеций |  | 0.7705331 | % |
| Симпсона |  | 0 | % |

1. **Программная реализация задачи:**
2. **Метод прямоугольников (все)** - метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на многочлен нулевой степени, то есть константу, на каждом элементарном отрезке.

Формулы:

1. Левых прямоугольников: Text

   Description automatically generated with medium confidence
2. Правых прямоугольников:

A picture containing text, clock

Description automatically generated

1. Средних прямоугольников:

A picture containing clock

Description automatically generated

Программная реализация:

def rectangle\_method(f, a, b, e, min\_n = 4, max\_itr = 10):

"""Метод прямоугольников"""

data = {}

type = input('\nТип метода: \n1)Левые \n2)Правые \n3)Средние \nВаш выбор: ')

while (type != '1') and (type != '2') and (type != '3'):

print("Нужно ввести либо 1, либо 2, либо 3!")

type = input('\nТип метода: \n1)Левые \n2)Правые \n3)Средние \nВаш выбор: ')

n = min\_n

result = float('inf')

maxN = 1 << 20

while n <= n \* (2 \*\* max\_itr) and n < maxN:

try:

last\_result = result

result = 0

x = a

h = (b - a) / n

for i in range(n):

if type == '1':

result += f(x)

elif type == '2':

result += f(x+h)

elif type == '3':

result += f(x + h / 2)

x += h

result \*= h

if abs(result - last\_result) <= e:

break

else:

n \*= 2

except ZeroDivisionError:

print('\Интеграл либо не существует, либо является бесконечностью')

break

if (n >= maxN):

print('Введенный погрешность не достигнут!')

data['result'] =result

data['n'] = n

return data

1. **Метод трапеции -** метод численного интегрирования функции одной переменной, заключающийся в замене на каждом элементарном отрезке подынтегральной функции на многочлен первой степени, то есть линейную функцию.

Формула метода трапеции:

Table

Description automatically generated

Программная реализация:

def trapezoid\_method(f,a,b,e,min\_n = 4, max\_itr = 10):

"""Метод трапеции"""

data = {}

n = min\_n

result = float('inf')

while n <= n \* (2 \*\* max\_itr):

last\_result = result

result = (f(a) + f(b)) / 2

h = (b-a) / n

x = a + h

for i in range(n - 1):

result += f(x)

x += h

result += h

if abs(result - last\_result) <= e:

break

else:

n \*= 2

data['result'] = result

data['n'] = n

return data

1. **Результаты выполнения программы.**

Text

Description automatically generated

Text

Description automatically generated

1. **Вывод.**

В результате выполнения данной лабораторной работой я познакомился с численными методами интегрирования и реализовал метод прямоугольников и метод трапеций языке программирования Python, закрепив знания.

Text

Description automatically generated