**Национальный исследовательский университет ИТМО**

**Факультет ПиИКТ**

**Изображение выглядит как логотип

Автоматически созданное описание**

**Лабораторная работа № 3**

по «Вычислительной математике»

Численное интегрирование

Работу выполнил: Велюс Арина

Группа: Р32151

Преподаватель: Машина Екатерина Александровка

**Санкт-Петербург**

**2023 г.**

**Цель работы:**

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

**Вариант №2**

**Изображение выглядит как текст

Автоматически созданное описание**

**Задание лабораторной работы:**

**Обязательное  задание (до 80 баллов)**

**Исходные данные:**

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

1. Методы должны быть оформлены в виде отдельной(ого) функции/класса.
2. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.
3. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.
4. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

**Вычислительная реализация задачи:**

1. Вычислить интеграл, приведенный в таблице 1, точно.
2. Вычислить интеграл по формуле Ньютона – Котеса при n=5.
3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при n=10 .
4. Сравнить результаты с точным значением интеграла.
5. Определить относительную погрешность вычислений.
6. В отчете ***отразить последовательные вычисления***.

**Интеграл для вычислительной реализации:**

Изображение выглядит как Шрифт, рукописный текст, каллиграфия, линия

Автоматически созданное описание

Вычисление точно:

-3/4x^4 - 5/3x\*3 + 2x^2 - 2x

-3/4 \* 1 - 5/3 \* (-1) -2x \* 1 - 2 \* (-1) - -3/4 \* 3 - 5/3 \* (-3) -2x \* 3 - 2 \* (-3) = -10/3 = -3.333

Метод Ньютона—Котеса:

(5 \* 0.4 / 288) \* (19 \* f(-3) + 75 \* f((-3) + 0.4) + 50 \* f((-3) + 0.8) + 50 \* f((-3) + 1.2) + 75 \* f((-3) + 1.6) + 19 \* f((-3) + 2))

(5 \* 0.4 / 288) \* (19 \* 22 + 75 \* 6.53 + 50 \* (-3.06) + 50 \* (-7.91) + 75 \* (-9.17) + 19 \* (-8.0)) = -3.334

Метод прямоугольников:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| x\_i | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 | -1.00 | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 |
| y\_i | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 | -8.00 | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 |
| x\_i/2 | -2.10 | -1.90 | -1.70 | -1.50 | -1.30 | -1.10 | -2.10 | -1.90 | -1.70 | -1.50 | -1.30 |
| y\_i/2 | -4.67 | -7.07 | -8.51 | -9.12 | -9.06 | -8.46 | -4.67 | -7.07 | -8.51 | -9.12 | -9.06 |

I\_лев = -0.160

I\_прав = -6.160

I\_сред = -3.420

Метод трапеций:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| x\_i | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 | -1.00 | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 |
| y\_i | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 | -8.00 | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 |

I = -3.160

Метод Симпсона:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| x\_i | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 | -1.00 | -2.00 | -1.80 | -1.60 | -1.40 | -1.20 |
| y\_i | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 | -8.00 | -6.00 | -7.90 | -8.91 | -9.17 | -8.82 |

I = -3.333

**Листинг программы:**

class IntegralMethod:  
 *"""  
 Класс, содержащий статические методы для численного интегрирования различными методами.  
 """* @staticmethod  
 def left\_rectangle(integral, left, right, n):  
 *"""  
 Метод прямоугольников (левая сумма).* ***:param*** *integral: Объект Integral, представляющий интеграл.* ***:param*** *left: Левый предел интегрирования.* ***:param*** *right: Правый предел интегрирования.* ***:param*** *n: Количество шагов разбиения.* ***:return****: Приближенное значение интеграла методом прямоугольников (левая сумма).  
 """* result = 0  
 step = abs(left - right) / n  
 i = left  
 while i < right:  
 result += integral.calculate(i)  
 i += step  
 return result \* step  
  
 @staticmethod  
 def center\_rectangle(integral, left, right, n):  
 *"""  
 Метод прямоугольников (средняя сумма).* ***:param*** *integral: Объект Integral, представляющий интеграл.* ***:param*** *left: Левый предел интегрирования.* ***:param*** *right: Правый предел интегрирования.* ***:param*** *n: Количество шагов разбиения.* ***:return****: Приближенное значение интеграла методом прямоугольников (средняя сумма).  
 """* result = 0  
 step = abs(left - right) / n  
 i = left + step / 2  
 while i < right:  
 result += integral.calculate(i)  
 i += step  
 return result \* step  
  
 @staticmethod  
 def right\_rectangle(integral, left, right, n):  
 *"""  
 Метод прямоугольников (правая сумма).* ***:param*** *integral: Объект Integral, представляющий интеграл.* ***:param*** *left: Левый предел интегрирования.* ***:param*** *right: Правый предел интегрирования.* ***:param*** *n: Количество шагов разбиения.* ***:return****: Приближенное значение интеграла методом прямоугольников (правая сумма).  
 """* result = 0  
 step = abs(left - right) / n  
 i = left + step  
 while i <= right:  
 result += integral.calculate(i)  
 i += step  
 return result \* step  
  
 @staticmethod  
 def trapezoid\_method(integral, left, right, n):  
 *"""  
 Метод трапеций.* ***:param*** *integral: Объект Integral, представляющий интеграл.* ***:param*** *left: Левый предел интегрирования.* ***:param*** *right: Правый предел интегрирования.* ***:param*** *n: Количество шагов разбиения.* ***:return****: Приближенное значение интеграла методом трапеций.  
 """* result = 0  
 step = abs(left - right) / n  
 i = left + step  
 while i < right:  
 result += integral.calculate(i)  
 i += step  
 result += (integral.calculate(left) + integral.calculate(right)) / 2  
 return result \* step  
  
 @staticmethod  
 def simpson\_method(integral, left, right, n):  
 *"""  
 Метод Симпсона.* ***:param*** *integral: Объект Integral, представляющий интеграл.* ***:param*** *left: Левый предел интегрирования.* ***:param*** *right: Правый предел интегрирования.* ***:param*** *n: Количество шагов разбиения (должно быть четным).* ***:return****: Приближенное значение интеграла методом Симпсона.  
 """* result = 0  
 step = abs(left - right) / n  
 for i in range(1, n):  
 if i % 2:  
 result += 2 \* integral.calculate(left + step \* i)  
 else:  
 result += 4 \* integral.calculate(left + step \* i)  
 result += integral.calculate(left) + integral.calculate(right)  
 return result \* step / 3

**Примеры и результаты работы программы:**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описаниеИзображение выглядит как линия, диаграмма, График

Автоматически созданное описание**

**Вывод:**

В ходе данной лабораторной работы были изучены несколько методов для численного интегрирования. Все методы просты в программной реализации и быстро вычисляют интегралы с хорошей точностью.