НАЦИОНАЛЬНЫЙ

ИССЛЕДОВАТЕЛЬСКИЙ

УНИВЕРСИТЕТ ИТМО

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 - Программная инженерия

Дисциплина – Вычислительная математика

Лабораторная работа №3

Вариант №11

Выполнил: Мухсинов С.П

Группа: P3217

Преподаватель: Малышева Т.А

Санкт-Петербург

2024

**Цель работы**:

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами

**Задание**:

**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:

• 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 = 6.
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 для каждого метода.
6. В отчете отразить последовательные вычисления.

**Используемые формулы и методы**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, рукописный текст

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, линия

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, линия, диаграмма

Автоматически созданное описание

Изображение выглядит как текст, линия, диаграмма, Шрифт

Автоматически созданное описание

Изображение выглядит как текст, Шрифт, снимок экрана, документ

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

**Вычислительная реализация**

**1. Вычислить интеграл, приведенный в таблице 1, точно:**

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

**Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание**

Ошибка:

Относительная погрешность: 0%

**3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при 𝑛 = 10 .**

**3.1 Формула средних прямоугольников:**

Ошибка:

Относительная погрешность: 0.045%

**3.2 Формула трапеций**

Ошибка:

Относительная погрешность: 0.09%

**3.3 Формула Симпсона**

Ошибка:

Относительная погрешность: 0%

**Программная реализация**

**Метод левых прямоугольников:**

@Override  
public double calculateIntegral(int n) {  
 double h = (B - A)/n;  
 double s = 0;  
 for(int i = 0; i < n; i++){  
 s += f.at(A + i \* h);  
 }  
 return h\*s;  
}

**Метод правых прямоугольников:**

@Override  
public double calculateIntegral(int n) {  
 double h = (B - A)/n;  
 double s = 0;  
 for(int i = 1; i <= n; i++){  
 s += f.at(A + i \* h);  
 }  
 return h\*s;  
}

**Метод средних прямоугольников:**

@Override  
public double calculateIntegral(int n) {  
 double s = 0;  
 double h = (B - A)/n;  
 double m;  
 for(int i = 1; i <= n; i++){  
 m = ((A + (i - 1) \* h) + (A + i \* h))/2;  
 s += f.at(m);  
 }  
 return h\*s;  
}

**Метод трапеций:**

@Override  
public double calculateIntegral(int n) {  
 double h = (B - A)/n;  
 double s = 0;  
 for(int i = 1; i <= n; i++){  
 s += (f.at(A + (i -1) \* h) + f.at(A + i \* h))/2;  
 }  
 return s\*h;  
}

**Метод Симпсона:**

@Override  
public double calculateIntegral(int n) {  
 n = 2 \* n;  
 double h = (B - A) / n;  
 double s = f.at(A) + f.at(B);  
 for(int i = 1; i < n; i+=2){  
 s += 4 \* (f.at(A + i\*h));  
 }  
 for(int i = 2; i < n; i+=2){  
 s+= 2 \* (f.at(A + i \* h));  
 }  
 return (s\*h)/3;  
}

**Пример работы программы**

**Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание**

**Вывод**

В ходе выполнения лабораторной работы мы изучили некоторые методы численного интегрирования и реализовали программу, вычисляющую приближенное значение определенного интеграла с требуемой точностью.