Университет ИТМО

Мегафакультет компьютерных технологий и управления

Факультет программной инженерии и компьютерной техники

Изображение выглядит как Шрифт, текст, белый, Графика

Автоматически созданное описание

ЛАБОРАТОРНАЯ РАБОТА №2

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ

Вариант №10

Группа: P3211

Студент: Орчиков Даниил Валерьевич

Преподаватель: Малышева Татьяна Алексеевна

г. Санкт-Петербург

2022

Оглавление

[Цель работы 2](#_Toc161070016)

[Вычислительная реализация задачи: 2](#_Toc161070017)

[1 часть. Решение нелинейного уравнения 2](#_Toc161070018)

[Рабочие формулы 2](#_Toc161070019)

[Решение 3](#_Toc161070020)

[2 часть. Решение системы нелинейных уравнений 4](#_Toc161070021)

[Рабочие формулы 4](#_Toc161070022)

[Решение 4](#_Toc161070023)

[Программная реализация задачи 5](#_Toc161070024)

[Рабочие формулы 5](#_Toc161070025)

[Листинг программы 6](#_Toc161070026)

[Примеры и результаты работы программы 10](#_Toc161070027)

[Пример 1 10](#_Toc161070028)

[Пример 2 10](#_Toc161070029)

[Пример 3 11](#_Toc161070030)

[Вывод 11](#_Toc161070031)

Цель работы

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

# Вычислительная реализация задачи:

## Рабочие формулы

Метод прямоугольников

* Левых
* Правых
* Средних

Метод трапеций

Метод Симпсона

Формула Ньютона-Котеса

## Интеграл

Точное значение:

## Формула Ньютона-Котеса

Значение, вычисленное по формуле, полностью соответствует точному значению интеграла.

## Формула средних прямоугольников

## Формула Симпсона

Значение, вычисленное по формуле, полностью соответствует точному значению интеграла.

# Программная реализация задачи

## Листинг программы

Представлен только код, непосредственно выполняющий вычисления

Весь код можно посмотреть [тут (GitHub)](https://github.com/DaniilOrchikov/Computational_mathematics/tree/master/laba3)

function calc(fun, a, b, accuracy, method) {  
 function f() {  
 let I = 0  
 let h = (b - a) / k  
 switch (method) {  
 case "left":  
 for (let i = a; i <= b - h; i += h)  
 I += fun(i)  
 break  
 case "right":  
 for (let i = a + h; i <= b; i += h)  
 I += fun(i)  
 break  
 case "middle":  
 for (let i = a + h / 2; i < b; i += h)  
 I += fun(i)  
 break  
 case "trapezoid":  
 for (let i = 1; i < *Math*.abs(b - a) / h; i += 1)  
 I += fun(a + i \* h)  
 I += (fun(a) + fun(b)) / 2  
 break  
 case "Simpson":  
 for (let i = 1; i <= *Math*.abs(b - a) / h - 1; i += 1)  
 I += fun(a + i \* h) \* (i % 2 ? 4 : 2)  
 I += fun(a) + fun(b)  
 I /= 3  
 break  
 }  
 return I \* h  
 }  
  
 let k = 4  
 let Ik, Ik1  
 Ik = f()  
 Ik1 = Ik  
 do {  
 Ik = Ik1  
 k \*= 2  
 Ik1 = f()  
 } while (*Math*.abs(Ik - Ik1) >= accuracy)  
 return {res: Ik1, intervals: k}  
}

# Примеры и результаты работы программы

Отмечу, что формулы для отображения графика и отрезков на которые он разбивается генерируются. Их не нужно вводить в калькулятор вручную.

## Пример 1

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

## Пример 2

Изображение выглядит как текст, диаграмма, График, линия

Автоматически созданное описание

# Вывод

Во время выполнения данной лабораторной работы я познакомился с различными численными методами вычисления интегралов и написал реализацию некоторых из них на языке JavaScript.