МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ

Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 4

по дисциплине

“Вычислительная математика”

Вариант “Метод трапеций”

Выполнила:

Студентка группы P3232

Копалина Майя
Алексеевна

Преподаватель:

Перл Ольга

Вячеславовна

УНИВЕРСИТЕТ ИТМО

Санкт-Петербург, 2024

Задание:
Придумать алгоритм и написать код, решающий данные функции методом трапеций.

Описание метода:
Метод трапеций — это численный метод для приближенного вычисления определенного интеграла. Он основан на аппроксимации подынтегральной функции кусочно-линейной функцией, используя трапеции для приближенного вычисления площади под графиком функции.

Для вычисления определенного интеграла от функции f(x) на интервале [a, b] с использованием n равномерно распределенных узлов, следуют следующие шаги:

1. Определяется шаг h = (b - a) / n.
2. Вычисляются значения функции f(x) в узлах x0 = a, x1 = a + h, ..., xn = b.
3. Суммируются значения функции f(x) в узлах, взвешенные соответствующими коэффициентами:
   (f(x0) + 2*f(x1) + 2*f(x2) + ... + 2*f(x(n-1)) + f(xn)).
4. Полученная сумма умножается на h/2.

<u>Пример:</u> вычислим интеграл $\int_0^1 x^2dx$ с помощью метода трапеций.

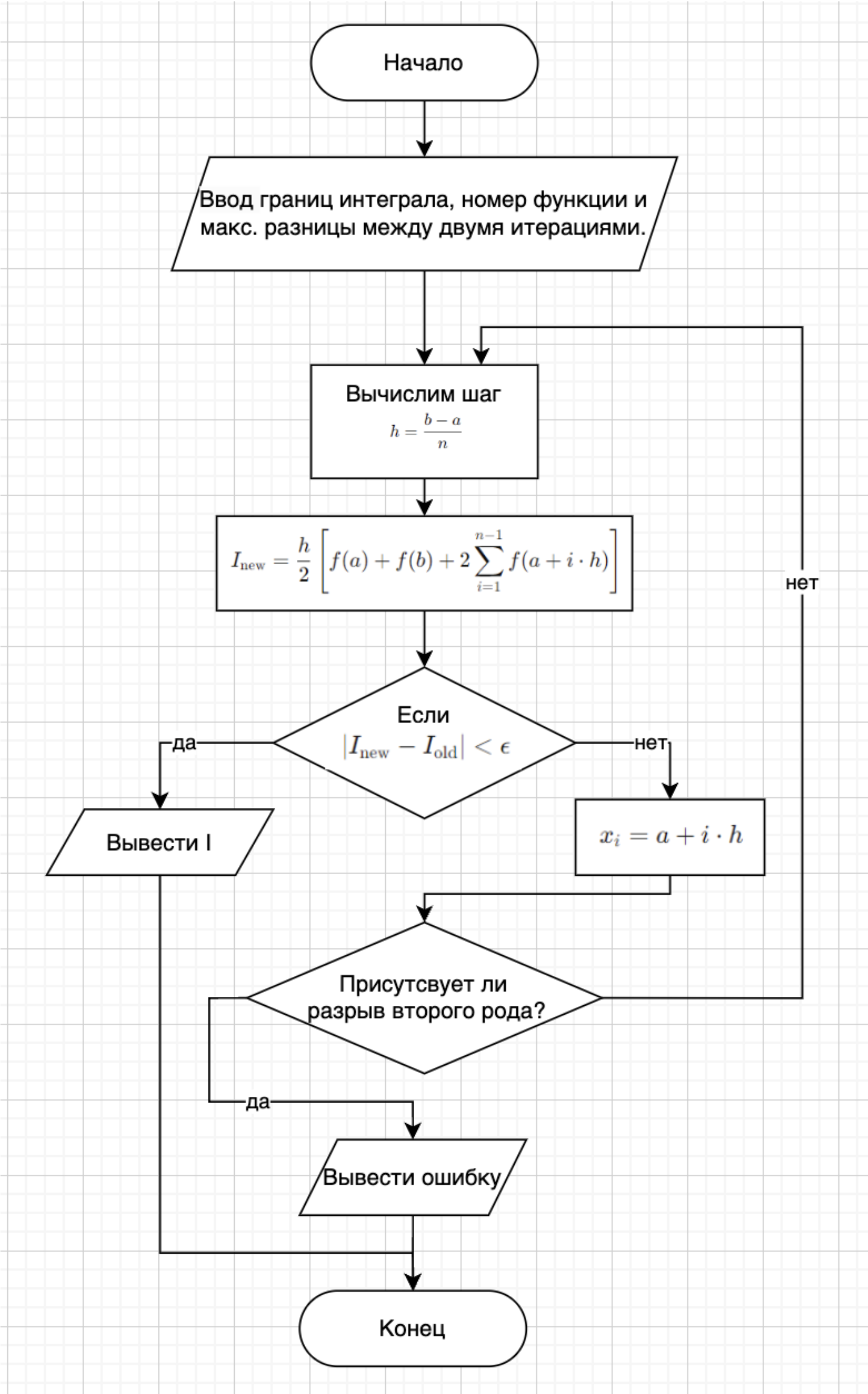
Разделим интервал [0,1] на 4 равных отрезка:

[0, 1/4], [1/4, 1/2], [1/2, 3/4], [3/4, 1]

Вычислим значения функции $f(x)=x2$ в концах отрезков:

f (0) = 0, f (1/4) = 1/16, f (1/2) = 1/4, f (3/4) = 9/16, f (1) = 1

Блок-схема:



Начало
Ввод границ интеграла, номер функции и макс. разницы между двумя итерациями.
Вычислим шаг
$h = \frac{b-a}{n}$
$I_{\text{new}} = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a + i \cdot h)\right]$
Если
$|I_{\text{new}} - I_{\text{old}}| < \epsilon$
да
нет
Вывести I
$x_i = a + i \cdot h$
Присутсвует ли разрыв второго рода?
нет
да
Вывести ошибку
Конец

Программа, написанная на python:

```
class Result:
    error_message = ""
    has_discontinuity = False
    eps = 1e-10

    @staticmethod
    def first_function(x: float):
        if x == 0:
            raise ValueError(Result.error_message)
        if abs(x) < Result.eps:
            return (math.sin(Result.eps)/Result.eps + math.sin(-Result.eps)/-
Result.eps)/2
        return 1/x

    @staticmethod
    def second_function(x: float):
        if x == 0:
            return (math.sin(Result.eps) / Result.eps + math.sin(-Result.eps) / -
Result.eps) / 2
        return math.sin(x) / x

    @staticmethod
    def third_function(x: float):
        return x*x+2

    @staticmethod
    def fourth_function(x: float):
        return 2*x+2

    @staticmethod
    def five_function(x: float):
        if x <= 0:
            Result.has_discontinuity = True
            Result.error_message = "Integrated function has discontinuity or does not
defined in current interval"
            return math.nan
        return math.log(x)

    def get_function(n: int):
        if n == 1:
            return Result.first_function
        elif n == 2:
            return Result.second_function
```

```
        elif n == 3:
            return Result.third_function
        elif n == 4:
            return Result.fourth_function
        elif n == 5:
            return Result.five_function
        else:
            raise NotImplementedError(f"Function {n} not defined.")

    # Complete the 'calculate_integral' function below.
    #
    # The function is expected to return a DOUBLE.
    # The function accepts following parameters:
    #  1. DOUBLE a
    #  2. DOUBLE b
    #  3. INTEGER f
    #  4. DOUBLE epsilon

    @staticmethod
    def calculate_approximation(a: float, b: float, h, func):
        sum_elements = 0

        values = [a]
        loop_value = a

        while loop_value < b:
            loop_value += h
            values.append(loop_value)
        for i in range(len(values)):
            try:
                func_value = func(values[i])
                if func_value is None:
                    raise ValueError("Integrated function has discontinuity or does not
defined in current interval")
                sum_elements += func_value
            except (ZeroDivisionError, ValueError):
                if abs(a) == abs(b):
                    return 0
                else:
                    return "error"
        return (h / 2) * (func(a) + 2 * sum_elements - func(b))

    @staticmethod
    def calculate_integral(a: float, b: float, f: int, epsilon: float):
        func = Result.get_function(f)
```

```
    n = 1
    integral_old = 0
    while True:
        h = (b - a) / n
        integral_new = 0.5 * (func(a) + func(b))
        for i in range(1, n):
            integral_new += func(a + i * h)
        integral_new *= h
        if abs(integral_new - integral_old) < epsilon:
            return integral_new
        integral_old = integral_new
        n *= 2

        x_values = [a + i * h for i in range(n + 1)]
        for x in x_values:
            if math.isnan(func(x)) or math.isinf(func(x)):
                Result.has_discontinuity = True
                Result.error_message = "Integrated function has discontinuity or does
not defined in current interval"
                return 0
```

Примеры работы программы:

1. Входные данные:
   a = 1.0
   b = 2.0
   f = 1
   epsilon = 0.0001
   Результат:
   0.6931471805599454

2. Входные данные:
   a = 0.1
   b = 1.0
   f = 2
   epsilon = 0.00001
   Результат:
   0.45969769413186023

3. Входные данные:
   a = -2.0
   b = 2.0
   f = 3
   epsilon = 0.0001
   Результат:
   12.666666666666664

4. Входные данные:
   a = -1.0
   b = 1.0
   f = 4
   epsilon = 0.00001
   Результат:
   3.9999999999999996

5. Входные данные:
   a = 0.01
   b = 1.0
   f = 5
   epsilon = 0.0001
   Результат:
   Integrated function has discontinuity or does not defined in current interval

Вывод:

Метод трапеций является численным методом интегрирования, который использует аппроксимацию подынтегральной функции линейной функцией между соседними точками. Для каждого участка между двумя соседними точками подынтегральной функции строится трапеция, а затем суммируются все трапеции для получения приближенного значения интеграла.

Преимущества:

- Простая реализация и понятная геометрическая интерпретация.
- Не требует вычисления производных.
- Применим к широкому классу функций.
- Относительно высокая точность.

Недостатки:

- Чувствителен к выбору шага интегрирования.
- Может иметь медленную сходимость для некоторых функций.
- Алгоритмическая сложность может быть высокой.

Анализ применимости метода:

Подходит:

- Метод трапеций подходит для вычисления интегралов от гладких функций.
- Метод трапеций может быть применен к функциям с разрывами, если шаг интегрирования достаточно мал.

Не подходит:

- Метод трапеций не подходит для вычисления интегралов от функций с резкими изменениями графика.

Алгоритмическая сложность:

O(n), где n - количество узлов (или интервалов), на которые разбивается область интегрирования.

Сравнение с другими методами:

Метод Симпсона – это численный метод вычисления определенного интеграла, основанный на аппроксимации площади под кривой параболами.
Метод трапеций является простым и эффективным методом численного интегрирования. Он особенно подходит для случаев, когда требуется высокая точность и несложно вычислить значения функции.
Метод Симпсона обеспечивает более высокую точность, чем метод трапеций, но требует больше вычислений.

Метод трапеций:

- Простая реализация.
- Не требуется вычисления производных.

- Подходит для функций с гладким графиком.

<u>Метод Симпсона:</u>

- Более высокая точность.
- Менее чувствителен к выбору шага интегрирования.
- Подходит для функций с более сложным графиком.

<u>Анализ численной ошибки:</u>

Численная ошибка метода трапеций может возникать из-за неточности вычисления значений функции, ошибок округления при выполнении арифметических операций и условий сходимости.

Для уменьшения численной ошибки:

- Можно использовать более точные методы вычисления значений функции.
- Можно уменьшить шаг интегрирования.

Результат выполнения программы на платформе.

Тесты пройдены (кроме одного...).

Assignment results

Score	Plagiarism
85 / 100	**42.553192%**

Number	Status	Score
1	Success	14 / 14
2	Success	14 / 14
3	Failed	0 / 15
4	Success	15 / 15
5	Success	14 / 14
6	Success	14 / 14
7	Success	14 / 14