Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

Факультет Программной инженерии и компьютерной техники

Лабораторная работа по

Вычислительной математике №4

Вариант «Метод Трапеций»

Работу выполнила:

Касьяненко В.М.

Группа:

P3220

Санкт-Петербург,

2024

Описание численного метода.

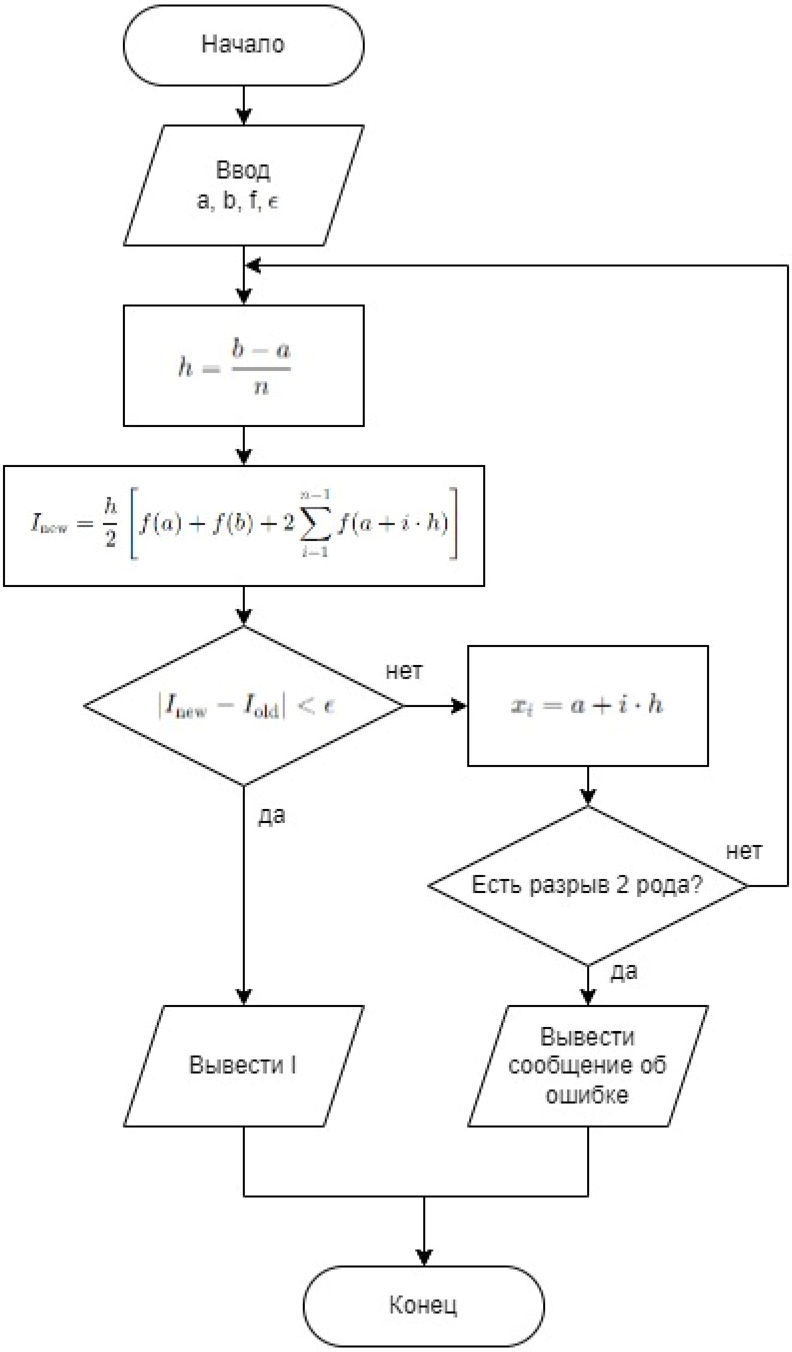
Метод трапеций – это численный метод для приближенного вычисления определенного интеграла. Он основан на аппроксимации подынтегральной функции с использованием трапеций.

Процесс метода трапеций следующий:

1. Разбиение интервала интегрирования $[a, b]$ на n равных частей с помощью точек $x_0 = a$, $x_1, x_2, \dots, x_{n-1}, x_n = b$. Расстояние между соседними точками $\Delta x = \frac{b-a}{n}$.
2. По каждому из этих интервалов $[x_{i-1}, x_i]$ строится трапеция. Площадь каждой трапеции можно вычислить по формуле площади трапеции: $S_i = \frac{h}{2}(f(x_{i-1}) + f(x_i))$, где h - высота трапеции (ширина интервала x_{i-1} до x_i), $f(x_{i-1})$ и $f(x_i)$ - значения функции в точках x_{i-1} и x_i соответственно.
3. Площади всех трапеций суммируются для получения приближенного значения определенного интеграла: $I \approx \sum_{i=1}^{n} S_i$.

Чем больше количество трапеций (или, что равносильно, меньше шаг Δx), тем более точное приближение интеграла обеспечивается методом трапеций.

Блок-схема

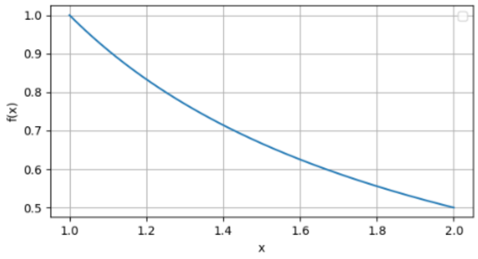

Начало
Ввод
a, b, f, ϵ
$h = \frac{b-a}{n}$
$I_{\text{new}} = \frac{h}{2}\left[f(a) + f(b) + 2\sum_{i=1}^{n-1} f(a + i \cdot h)\right]$
$|I_{\text{new}} - I_{\text{old}}| < \epsilon$
нет
$x_i = a + i \cdot h$
да
Есть разрыв 2 рода?
нет
да
Вывести I
Вывести сообщение об ошибке
Конец

Код численного метода

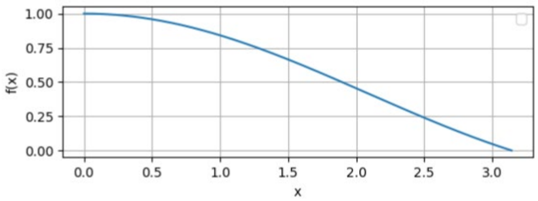
```
def calculate_integral(a, b, f, epsilon):
        func = Result.get_function(f)
        n = 1
        integral_old = 0
        while True:
            h = (b - a) / n
            integral_new = 0.5 * (func(a) + func(b))
            for i in range(1, n):
                integral_new += func(a + i * 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

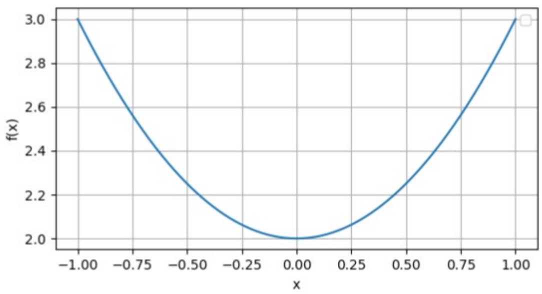
1)
Входные данные:
1
2
1
0.0001
Выходные данные:
0.6931624388834033



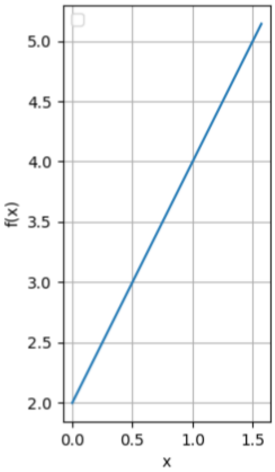
2)
Входные данные:
1
3.14159
2
0.0001
Выходные данные:
1.8519210729517013



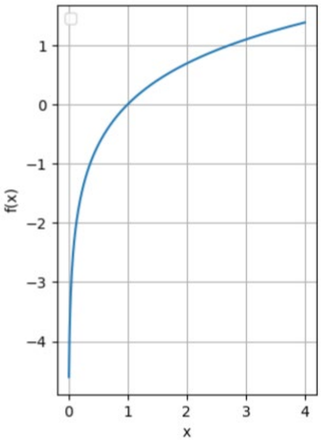
3)
Входные данные:
1
-1
3
0.0001
Выходные данные:
-4.66668701171875



4)
Входные данные:
0
1.5708
4
0.01
Выходные данные:
5.60901264



5)
Входные данные:
0
4
5
0.001
Выходные данные:
Integrated function has discontinuity or does not defined in current interval



Выводы

В результате выполнения лабораторной работы был реализован метод трапеций для вычисления интеграла от выбранной функции на заданном интервале. Метод обеспечивает вычисление интеграла с заданной точностью, используя итеративный процесс с увеличением числа разбиений до достижения требуемой разницы между итерациями.

Результаты запуска реализованного метода на различных данных: если выбранная функция непрерывна на заданном интервале и не имеет разрывов второго рода, то метод трапеций должен сойтись к корректному значению интеграла при достаточно малом значении ε. Результаты будут корректными и близкими к ожидаемому значению интеграла, если функция имеет разрыв второго рода или не определена на каком-то участке интервала интегрирования, то метод может выдать сообщение об ошибке "Integrated function has discontinuity or does not defined in current interval". В этом случае интеграл может быть невозможно вычислить.

Сравнение метода трапеций с методом прямоугольников и методом Симпсона. Метод прямоугольников обычно требует большего числа подинтервалов для достижения той же точности, что и метод трапеций, также он менее точен, чем метод трапеций для большинства функций. Метод Симпсона требует больше вычислительных ресурсов из-за сложности вычисления квадратичных интерполяционных полиномов, сложнее в реализации, чем метод трапеций. Метод прямоугольников является наиболее простым в реализации, но обычно менее точным по сравнению с методом трапеций и методом Симпсона. В то время как метод Симпсона обеспечивает более высокую точность, чем метод трапеций, но требует больше вычислительных ресурсов. Метод трапеций обеспечивает баланс между простотой реализации и точностью. Он обычно более точен, чем метод прямоугольников, но менее точен, чем метод Симпсона.

Анализ применимости метода: метод трапеций хорошо работает для непрерывных функций. Если функция имеет разрывы, особенно разрывы второго рода, метод может потребовать дополнительной обработки или давать неточные результаты, при наличии устранимых разрывов первого рода метод трапеций может быть применен с некоторыми оговорками. Он всё равно будет сходиться к правильному значению интеграла, но может потребоваться больше итераций для достижения заданной точности.

Общая алгоритмическая сложность метода трапеций может быть оценена как $O(nm)$, где n – количество разбиений интервала интегрирования, а m – сложность вычисления функции в одной точке.

Метод трапеций аппроксимирует подынтегральную функцию с помощью кусочно-линейной функции, состоящей из трапеций. Ошибка возникает из-за того, что подынтегральная функция не полностью аппроксимируется этими трапециями. Это приводит к тому, что фактическое значение интеграла отличается от вычисленного методом трапеций. Также во время выполнения численных вычислений могут возникать ошибки округления, особенно при использовании чисел с плавающей точкой. Эти ошибки могут накапливаться и влиять на итоговый результат.

В целом, метод трапеций остается важным инструментом для численного интегрирования благодаря своей универсальности, относительной простоте реализации и достаточной точности для большинства практических задач. Однако при его применении необходимо учитывать его ограничения и применять соответствующие методы управления ошибками для обеспечения точности результатов.