Университет ИТМО

Факультет программной инженерии и компьютерной техники

Лабораторная работа №4
по дисциплине “Вычислительная математика”
Вариант: Метод трапеций

Выполнила:
Шевченко Д. П.,
группа P3230

Преподаватель:
Перл О.В.

Санкт-Петербург
2024

Оглавление

Описание численного метода

Метод трапеций заключается в разбиении отрезка интегрирования на промежуточные отрезки. Основан на аппроксимации подынтегральной функции линейной функции, заданной трапециями.

Допустим, есть интеграл, который нужно вычислить:

$$\int_a^b f(x)dx$$

Сначала нужно разделить отрезок интегрирования [a, b] точками на n равных частей с определенным шагом разбиения:

$$h = \frac{b-a}{n}$$

Найдем узлы по формуле:

$$x_i = a + i * h$$
$$i = 1, \dots, n$$

Соединяем две точки, стоящие рядом, и ищем площадь получившейся трапеции:

$$S_i = h * \frac{y_i + y_{i-1}}{2}$$

Для вычисления интеграла нужно сложить все найденные площади трапеций:

$$S_1 + \cdots + S_n = h * (y_1 + .. + y_{n-1} + \frac{y_0 + y_n}{2})$$

Эта сумма будет приближенной площадью фигуры под графиком функции f(x) на отрезке [a, b].

Графически изобразила на рисунке 1.

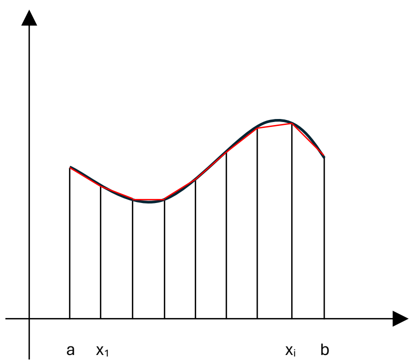
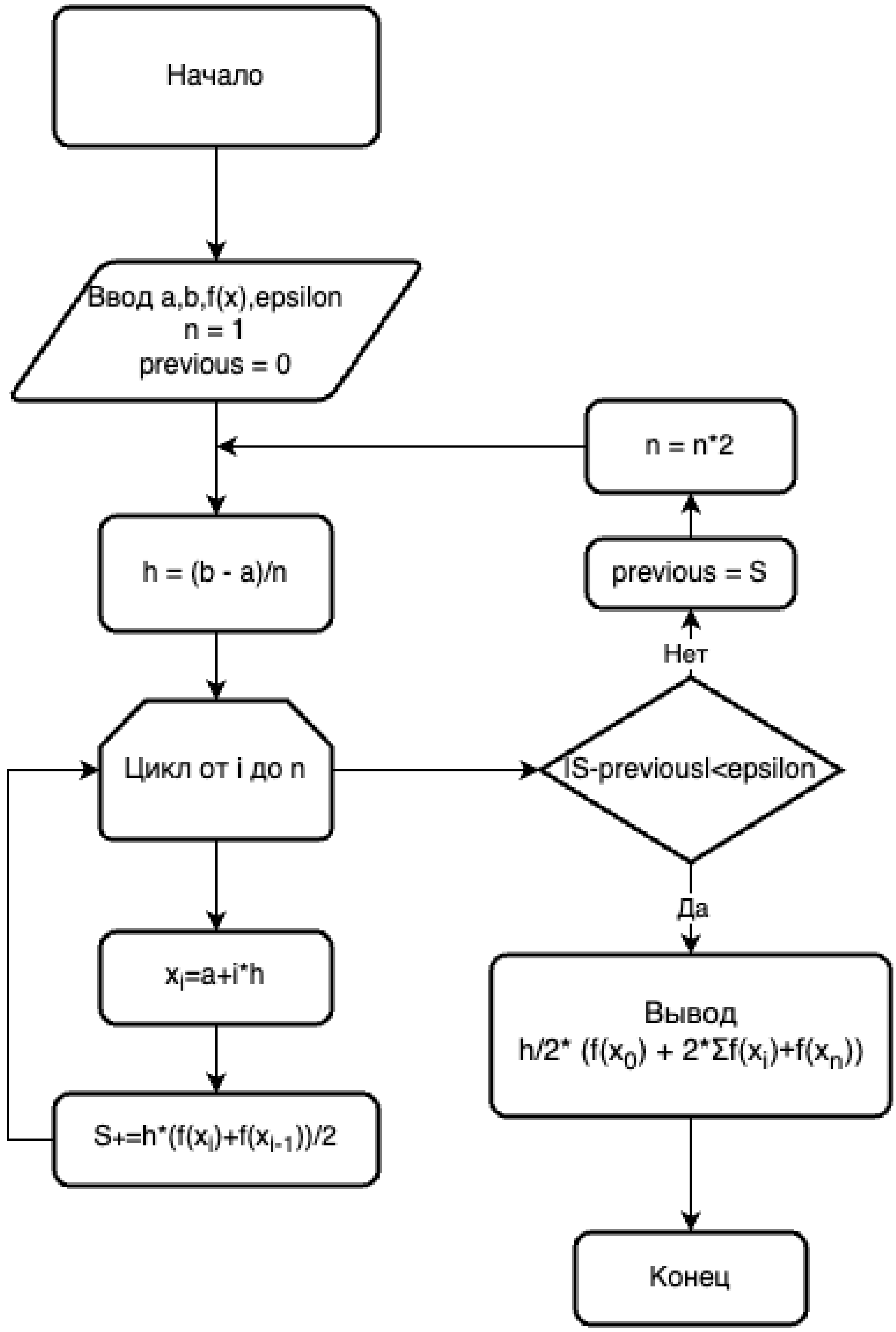


Рисунок 1

Блок схема



Начало
Ввод a,b,f(x),epsilon
n = 1
previous = 0
n = n*2
h = (b - a)/n
previous = S
Нет
Цикл от i до n
|S-previous|<epsilon
Да
x_i=a+i*h
Вывод
h/2* (f(x_0) + 2*Σf(x_i)+f(x_n))
S+=h*(f(x_i)+f(x_{i-1}))/2
Конец

Код

```
def check_discontinuity(f, x):
    if f == 5 and x <= 0:
        return True
    return False

def calculate_integral(a, b, f, epsilon):
    function = Result.get_function(f)
    if Result.check_discontinuity(f, a) or Result.check_discontinuity(f, b)
or (f == 1 and (a == 0 or b == 0)):
        Result.has_discontinuity = True
        Result.error_message = "Integrated function has discontinuity or does
not defined in current interval"
        return 0
    summ = (function(a) + function(b))
    n = 1
    previous = 0
    delta = b - a
    result = 0
    check = epsilon + 0.001
    while abs(check - previous) > epsilon:
        previous = result
        h = delta / n
        for i in range(n):
            if Result.check_discontinuity(f, a + i * h):
                Result.has_discontinuity = True
                Result.error_message = "Integrated function has discontinuity
or does not defined in current interval"
                return 0
            result += function(a + i * h)
        result += summ / 2
        result *= h
        check = result
        n = 2 * n
    return result
```

Примеры работы программы

1) Случай, когда a > b, значение должно получиться отрицательным

```
5
2
1
0.0001
-0.9162251656490367
```

2) a = 0, функция 1/x, соответственно 0 это точка разрыва второго рода

```
0
3
1
0.001
Integrated function has discontinuity or does not defined in current interval
```

3) Корректный вывод

```
1
2
3
0.001
4.334228736766923
```

4) В логарифм не может быть подставлено значение <= 0

```
-3
1
5
0.001
Integrated function has discontinuity or does not defined in current interval
```

5) Корректный вывод

```
0
2
2
0.001
1.6060493726838136
```

Выводы

1) Примеры работы программы на странице 6
2) По сравнению с методом прямоугольников точность вычислений будет выше и потребуется меньше интервалов. Но метод прямоугольников имеет меньшую погрешность и его проще реализовать. А также в методе трапеций шаг может быть произвольным, в то время как метод прямоугольников не применим с функциями с конечным числом точек, не аналитическими.
   Метод Симпсона является более точным,но сложным и ресурсозатратным из-за интерполяционного многочлена второй степени.
3) Анализ применимости метода:

   Данный метод применим к непрерывным функциям и функциям с конечным числом точек. Может давать неточные результаты для функций с разрывами. Если есть устранимый разрыв первого рода, то метод может быть применим с бóльшим количеством шагов. Если функция имеет разрыв второго рода или не определена в какой-нибудь точке интервала, то метод не может быть применим.

4) Анализ алгоритмической сложности:
   Сложность равна O(n^2), так как мы продолжаем вычисления n раз, где n – количество отрезков, на которые разбивается интервал, и в цикле while продолжаем вычислять значение функции так же n раз.

5) Анализ численной ошибки:
   Абсолютная погрешность метода может быть вычислена по формуле:
   $$\max|f''(x)| * \frac{(b-a)^3}{12n^2}$$
   Погрешность у метода трапеций больше чем у метода треугольников. Для уменьшения численной ошибки нужно увеличить количество отрезков, на которые разбивается интервал. Чем меньше шаг разбиения и более гладкая функция, тем погрешность меньше.
   Возникающая ошибка равна сумме площадей между кривой функции и хордами, которые соединяют точки разбиения.

   Метод трапеций является простым в реализации. Для большей точности может понадобиться большое число трапеций.