Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

Вычислительная математика

Лабораторная работа №4.

Метод средних прямоугольников

Выполнил:

Маликов Глеб Игоревич

Группа № P3224

Преподаватели:

Перл Ольга Вячеславовна

Хохлов Александр Алексеевич

г. Санкт-Петербург

2024

Оглавление

Задание

Реализуйте метод средних прямоугольников для вычисления интеграла от выбранной функции на интервале от a до b.

Если функция имеет разрыв второго рода или "скачок", или если функция не определена какой-либо частью в интервале от a до b, то вам следует указать переменные error_message и hasDiscontinuity.

Сообщение об ошибке, которое вы должны указать: "Integrated function has discontinuity or does not defined in current interval".

Если функция имеет устранимый разрыв первого рода, то вы должны уметь вычислить интеграл.

Если a > b, то интеграл должен иметь отрицательное значение.

Формат ввода:

```
a
b
f
epsilon
```

, где a и b - границы интеграла, f - номер функции, epsilon - максимальная разница между двумя вашими итерациями (итерация — это некоторое разбиение на отрезки).

Формат вывода:

```
I
```

, где I - ваш вычисленный интеграл для текущего количества разбиений.

Описание численного метода

Метод средних прямоугольников — это численный метод для приближенного вычисления определенных интегралов.

Для вычисления определённого интеграла интервал интегрирования $[a; b]$ разделяется на n равных подинтервалов с точками $x_0, x_1, \dots, x_n$ и шириной $h = \frac{b-a}{n}$. При этом точки x_0 и x_n обозначают a и b соответственно.

Для каждого подинтервала $[x_{i-1}, x_i]$ выбирается средняя точка $\xi_i = \frac{x_{i-1}+x_i}{2}$. Так приближенное значение интеграла вычисляется как сумма площадей прямоугольников, основания которых — это под интервалы, а высоты - значения функции в средних точках:

$$\int_a^b f(x)dx \approx h\sum_{i=1}^{n} f(\xi_i)$$

Блок-схемы

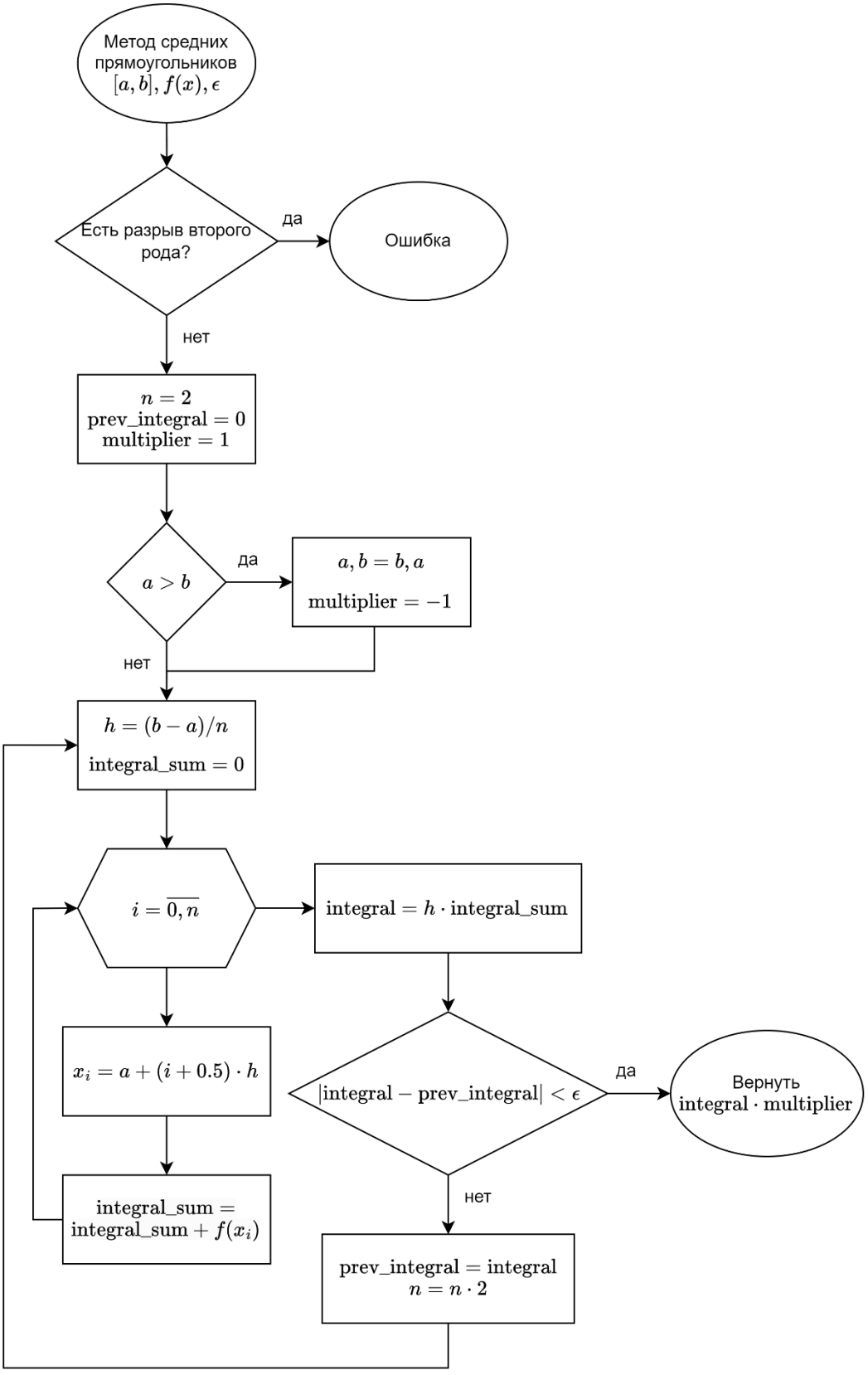

Схема 1 - Метод средних квадратов

Код

has_function_discontinuity

```
def has_function_discontinuity(f, a: float, b: float) -> bool:
    """
    Проверяет, имеет ли выбранная функция разрыв в интервале [a, b].
    :param f: Выбранная функция.
    :param a: Нижняя граница интервала.
    :param b: Верхняя граница интервала.
    :return: True, если функция имеет разрыв в интервале [a, b], иначе
False.
    """
    try:
        if a <= 0 <= b:
            f(0)
        if a < 0 or b < 0:
            f(-1)
        f(a)
        f(b)
    except Exception:
        Result.has_discontinuity = True
        Result.error_message = "Integrated function has discontinuity or
does not defined in current interval"
        return True
```

calculate_integral

```
def calculate_integral(a: float, b: float, f: int, epsilon: float) ->
float:
    """
    Вычисляет интеграл выбранной функции на интервале [a, b] с
использованием метода средних прямоугольников.
    :param a: Нижняя граница интервала.
    :param b: Верхняя граница интервала.
    :param f: Номер выбранной функции.
    :param epsilon: Максимальная разница между двумя итерациями.
    :return: Вычисленный интеграл для текущего количества разбиений.
    """
    Result.eps = epsilon
    func = Result.get_function(f)
    if Result.has_function_discontinuity(func, a, b):
        return 0.0

    n = 2  # Initial number of partitions
    prev_integral = 0.0
    integral = 0.0
    multiplier = 1
```

```
if a > b:
    a, b = b, a
    multiplier = -1

while True:
    h = (b - a) / n
    integral_sum = 0.0

    for i in range(n):
        xi = a + (i + 0.5) * h
        integral_sum += func(xi)

    integral = h * integral_sum

    if abs(integral - prev_integral) < epsilon:
        break

    prev_integral = integral
    n *= 2

return integral * multiplier
```

Примеры работы программы

Пример 1

Функция 1 с интервалом без разрывов

Ввод:

```
0.1
3
1
1e-6
```

Вывод:

```
3.401197251266905
```

Информация:

```
Interval width: 0.000177001953125
Number of partitions: 16384
```

Пример 2

Функция 1 с разрывом второго рода в 0

Ввод:

```
0
4
1
1e-6
```

Вывод:

```
Integrated function has discontinuity or does not defined in current interval
```

Пример 3

Функция 1 с разрывом второго рода 0

Ввод:

```
-1
3
1
1e-10
```

Вывод:

```
Integrated function has discontinuity or does not defined in current interval
```

Пример 4

Функция 2 со значениями a = b

Ввод:

```
0
0
2
1e-6
```

Вывод:

```
0.0
```

Информация:

```
Interval width: 0.0
Number of partitions: 2
```

Пример 5

Функция 2 со значениями a <b

Ввод:

```
-6.7
-2.3
2
1e-6
```

Вывод:

```
-0.29100220387273734
```

Информация:

```
Interval width: 0.0021484375
Number of partitions: 2048
```

Пример 6

Функция 2 со значениями a <b

Ввод:

```
6
-6
2
1e-6
```

Вывод:

```
-2.849374982548296
```

Информация:

```
Interval width: 0.0029296875
Number of partitions: 4096
```

Пример 7

Функция 3

Ввод:

```
123
456
3
1e-6
```

Вывод:

```
30986648.999998454
```

Информация:

```
Interval width: 0.00015878677368164062
Number of partitions: 2097152
```

Пример 8

Функция 4

Ввод:

```
-30
-50
4
1e-6
```

Вывод:

```
1560.0
```

Информация:

```
Interval width: 5.0
Number of partitions: 4
```

Пример 9

Функция 5

Ввод:

```
0.1
3
5
```

```
1e-6
```

Вывод:

```
0.6260955772055777
```

Информация:

```
Interval width: 0.0007080078125
Number of partitions: 4096
```

Пример 10

Функция 5 не определена для отрицательных значений

Ввод:

```
0
-4
5
1e-6
```

Вывод:

```
Integrated function has discontinuity or does not defined in current interval
```

Вывод

Метод средних прямоугольников значительно упрощает вычисление определённых интегралов, особенно когда аналитическое вычисление сложно или невозможно. Однако, по сравнению с более сложными методами, такими как метод трапеций или метод Симпсона, он может быть менее точным или требовать больше итераций для достижения заданной точности.

Одним из недостатков метода средних прямоугольников является его потенциальная неточность для функций, которые быстро изменяются или имеют высокие производные. Как можно увидеть в примере 7, где количество интервалов более 2 миллионов, вычисление данного интервала занимает значительно больше времени чем функции с меньшими значениями. Если потребуется вычисление интегралов с ещё более высокими значениями, то количество нужных интервалов будет чрезмерно высоким.

Алгоритмическая сложность метода средних прямоугольников составляет $O(n)$ где n - количество разбиений. Количество итерации программы зависит от нескольких факторов, включая сложность функции, размер интервала интегрирования и заданную точность. Функции, которые быстро изменяются или имеют высокие производные, могут потребовать большего количества итераций.

Код успешно решает задачу вычисления определённых интегралов с необходимой точностью и корректно определяет если есть разрывы второго рода для предоставленных функции. Однако стоит отметить, что определение данных разрывов часто требует аналитическое определение разрывных точек.

Численная ошибка метода простых прямоугольников зависит от размера шага (или количества разбиений) и второй производной функции. В общем случае ошибка пропорциональна квадрату размера шага, что означает, что уменьшение размера шага в два раза приведет к уменьшению ошибки в четыре раза. Однако, для функций с высокими производными или разрывами, ошибка может быть значительно больше.