Задание

Наименование задачи: произвести интегрирование функций на заданном интервале методом прямоугольников.

Вид решения: программа и отчёт.

Реализация решения: язык C или C++.

Разработать алгоритм и написать программу реализующую: интегрирование функций на заданном интервале методом прямоугольников, расчёт остаточного члена. При равном шаге сетки сравнить полученный результат с результатами полученными иными методами численного интегрирования. Оценить вычислительную сложность.

Функции для интегрирования:
1. $f(x) = x^2, x \in [-5; 5]$;
2. $f(x) = \sin^2 x, x \in [-\pi; \pi]$;
3. $f(x) = \sin 2x + \cos 7x + 8, x \in [-\pi; \pi]$;
4. $f(x) = 2x^4 + x^3 + 2x^2 + 3x + 24, x \in [-1; 3]$;
5. $f(x) = \ln(x^2 + 1) + \sin\frac{x}{3} + 17, x \in [-100; 100]$;
6. $f(x) = 5^x + \sin x + x + 11,\ x \in [-\pi; \pi]$;
7. $f(x) = x^5 + 2x^4 + 3x^3 + 4x^2 + 5x + 6, x \in [-7; 7]$;

Task: perform integration of functions on a given interval by the method of rectangles.

Элементы теории

Метод прямоугольников — метод численного интегрирования функции одной переменной, заключающийся в замене подынтегральной функции на многочлен нулевой степени, то есть константу, на каждом элементарном отрезке. Если рассмотреть график подынтегральной функции, то метод будет заключаться в приближённом вычислении площади под графиком суммированием площадей конечного числа прямоугольников, ширина которых будет определяться расстоянием между соответствующими соседними узлами интегрирования, а высота — значением подынтегральной функции в этих узлах.

Если отрезок $[a, b]$ является элементарным и не подвергается дальнейшему разбиению, значение интеграла можно найти по

1. Формуле левых прямоугольников: $\int_a^b f(x)dx \approx f(a)(b-a)$.
2. Формуле правых прямоугольников: $\int_a^b f(x)dx \approx f(b)(b-a)$.
3. Формуле средних прямоугольников: $\int_a^b f(x)dx \approx f(\frac{a+b}{2})(b-a)$.

В случае разбиения отрезка интегрирования на n элементарных отрезков приведённые выше формулы применяются на каждом из этих элементарных отрезков между двумя соседними узлами. В результате получаются следующие формулы:

1. Для левых прямоугольников: $\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f(x_i)(x_{i+1} - x_i)$
2. Для правых прямоугольников: $\int_a^b f(x)dx \approx \sum_{i=1}^{n} f(x_i)(x_i - x_{i-1})$
3. Для средних прямоугольников: $\int_a^b f(x)dx \approx \sum_{i=0}^{n-1} f\left(\frac{x_i + x_{i+1}}{2}\right)(x_{i+1} - x_i) = \sum_{i=1}^{n} f\left(\frac{x_{i-1} + x_i}{2}\right)(x_i - x_{i-1})$.

$x_i = a + ih, \qquad h = \frac{b-a}{n},$

где h - шаг сетки.

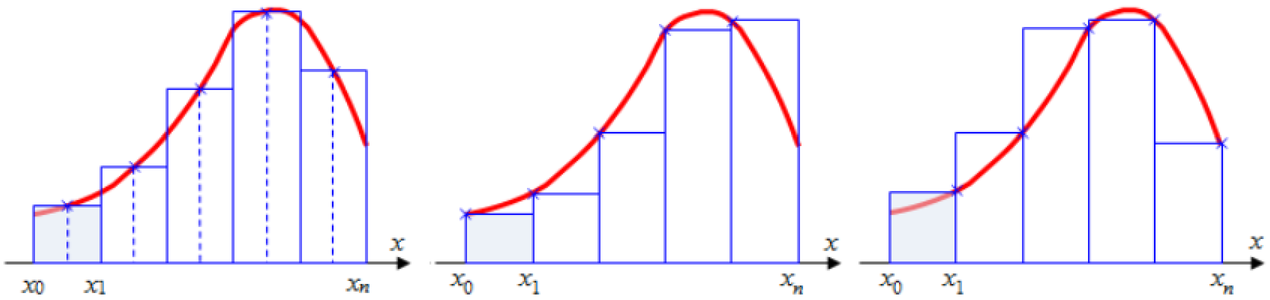

Рисунок 1. Метод средних, левых и правых прямоугольников соответственно

Для формул правых и левых прямоугольников остаточный член для каждого элементарного отрезка равен

$$R(f) = \frac{f'(\xi)}{2} h^2$$

Для формулы средних прямоугольников:

$$R(f) = \frac{f''(\xi)}{24} h^3$$

где $\xi \in [a, b]$

Метод правых прямоугольников в приведенной ниже программе не реализуется ввиду его схожести с методом левых прямоугольников.

Листинг программы

```
#include <iostream>

#include <cmath>

using namespace std;


double func(double, int);
```

```
double derivative(double, int);
double derivative_2(double, int);

void left_rectangles(double, double, int, int);
void middle_rectangles(double, double, int, int);

int main() {
    setlocale(LC_ALL, "Russian");
    const double pi = 3.14;

    int i;
    cout << "Введите номер интегрируемой функции" << endl;
    cin >> i;
    if (!(i >= 1 && i <= 7)) {
        cout << "Нет функции с таким номером" << endl;
        return 0;
    }

    int n;
    cout << "Введите количество отрезков, на которые разбивается исходный отрезок" << 
endl;
    cin >> n;

    double a, b;
    if (i == 1) {
        a = -5; b = 5;
    }
    if (i == 2) {
        a = -pi; b = pi;
```

```
    }
    if (i == 3) {
        a = -pi; b = pi;
    }
    if (i == 4) {
        a = -1; b = 3;
    }
    if (i == 5) {
        a = -100; b = 100;
    }
    if (i == 6) {
        a = -pi; b = pi;
    }
    if (i == 7) {
        a = -7; b = 7;
    }

    left_rectangles(a, b, n, i);
    middle_rectangles(a, b, n, i);
    return 0;
}
void middle_rectangles(double a, double b, int n, int i) {
    double integral_sum = 0;
    double h = (b - a) / n;
    double left = a;
    double R = 0; //Остаточный член
    for (int j = 0; j < n; j++) {
        R += derivative_2(left + h / 2, i);
        integral_sum += func(left + h / 2, i) * h;
```

```
        left += h;
    }
    R *= pow(h , 3) / 24;
    cout << endl << "Результаты метода средних прямоугольников:" << endl;
    cout << "Приближенное значение интеграла: " << integral_sum << endl;
    cout << "Остаточный член: " << R << endl;
}
void left_rectangles(double a, double b, int n, int i) {
    double integral_sum = 0;
    double h = (b - a) / n;
    double left = a;
    double R = 0; //Остаточный член
    for (int j = 0; j < n; j++) {
        R += derivative(left + h / 2, i);
        integral_sum += func(left, i) * h;
        left += h;
    }
    R *= pow(h, 2) / 2;
    cout << endl << "Результаты метода левых прямоугольников:" << endl;
    cout << "Приближенное значение интеграла: " << integral_sum << endl;
    cout << "Остаточный член: " << R << endl;
}
double func(double x, int i) {
    if (i == 1) {
        return pow(x, 2);
    }
    if (i == 2) {
        return sin(x) * sin(x);
    }
```

```
    if (i == 3) {
        return sin(2 * x) + cos(7 * x) + 8;
    }
    if (i == 4) {
        return 2 * pow(x, 4) + pow(x, 3) + 2 * pow(x, 2) + 3 * x + 24;
    }
    if (i == 5) {
        return log(pow(x, 2) + 1) + sin(x / 3) + 17;
    }
    if (i == 6) {
        return pow(5, x) + sin(x) + x + 11;
    }
    if (i == 7) {
        return pow(x, 5) + 2 * pow(x, 4) + 3 * pow(x, 3) + 4 * pow(x, 2) + 5 * x + 6;
    }
}
double derivative(double x, int i) {
    if (i == 1) {
        return 2 * x;
    }
    if (i == 2) {
        return 2 * sin(x) * cos(x);
    }
    if (i == 3) {
        return 2 * cos(2 * x) - 7 * sin(7 * x);
    }
    if (i == 4) {
        return 8 * pow(x, 3) + 3 * pow(x, 2) + 4 * x + 3;
    }
```

```
    if (i == 5) {
        return 2 * x / (pow(x, 2) + 1) + cos(x / 3) / 3;
    }
    if (i == 6) {
        return pow(5, x) * log(5) + cos(x) + 1;
    }
    if (i == 7) {
        return 5 * pow(x, 4) + 8 * pow(x, 3) + 9 * pow(x, 2) + 8 * x + 5;
    }
}
double derivative_2(double x, int i) {
    if (i == 1) {
        return 2;
    }
    if (i == 2) {
        return 2 * (pow(cos(x), 2) - pow(sin(x), 2));
    }
    if (i == 3) {
        return -(4 * sin(2 * x) + 49 * cos(7 * x));
    }
    if (i == 4) {
        return 24 * pow(x, 2) + 6 * x + 4;
    }
    if (i == 5) {
        return 2 / (pow(x, 2) + 1) - 4 * pow(x, 2) / pow((pow(x, 2) + 1), 2) - sin(x / 3) / 9;
    }
    if (i == 6) {
        return pow(5, x) * pow(log(5), 2) - sin(x);
    }
```

```
    if (i == 7) {
        return 20 * pow(x, 3) + 24 * pow(x, 2) + 18 * x + 8;
    }
}
```

Вывод программы

Введите номер интегрируемой функции

1

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 83.3335

Остаточный член: -6.27765e-15

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 83.3332

Остаточный член: 8.33333e-05

Введите номер интегрируемой функции

2

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 3.14159

Остаточный член: 1.42504e-18

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 3.14159

Остаточный член: -1.04687e-08

Введите номер интегрируемой функции

3

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 50.2432

Остаточный член: -2.00038e-05

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 50.2432

Остаточный член: -2.56496e-07

Введите номер интегрируемой функции

4

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 243.835

Остаточный член: 0.432

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 244.266

Остаточный член: 0.000176

Введите номер интегрируемой функции

5

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 4848.14

Остаточный член: 0.188141

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 4848.33

Остаточный член: 6.666e-05

Введите номер интегрируемой функции

6

Введите количество отрезков, на которые разбивается исходный отрезок

1000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 165.861

Остаточный член: 0.511402

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 166.371

Остаточный член: 0.000414122

Введите номер интегрируемой функции

7

Введите количество отрезков, на которые разбивается исходный отрезок

10000

Результаты метода левых прямоугольников:

Приближенное значение интеграла: 14419.2

Остаточный член: 25.0194

Результаты метода средних прямоугольников:

Приближенное значение интеграла: 14444.3

Остаточный член: 0.000457333

Выводы

Результаты работы показали, что количество итераций цикла равно количеству элементарных отрезков на которые разбивается область интегрирования. Следовательно, вычислительную сложность алгоритма можно оценить как $O(n)$. Также заметим, что метод средних прямоугольников дает наиболее приближенное значение интеграла.

Заключение

В ходе выполнения домашнего задания были изучены теоретические основы метода прямоугольников для приближенного интегрирования. Также была разработана программа на языке C++ для метода левых и средних прямоугольников, сравнены результаты алгоритмов при равном шаге сетки и оценены вычислительные сложности.

Список литературы

1. Т.В. Иванова. Численные методы в оптике. Учебное пособие. – СПб: Университет ИТМО, 2017 - 84 с.
2. https://ru.wikipedia.org/wiki/Метод_прямоугольников.