МИНОБРНАУКИ РОССИИ
УНИВЕРСИТЕТ ИТМО
Факультет программной инженерии и компьютерных технологий

ОТЧЕТ
по лабораторной работе №3
по дисциплине «Вычислительная математика»
Вариант 15

Преподаватель: Малышева Т.А.
Выполнил: Сабуров В. А.
Группа: P3210

Санкт-Петербург
2021

Цель лабораторной работы

Реализовать метод прямоугольников, метод трапеций и метод Симпсона для нахождения приближенного значения интеграла.

Описание использованного метода

Метод прямоугольников

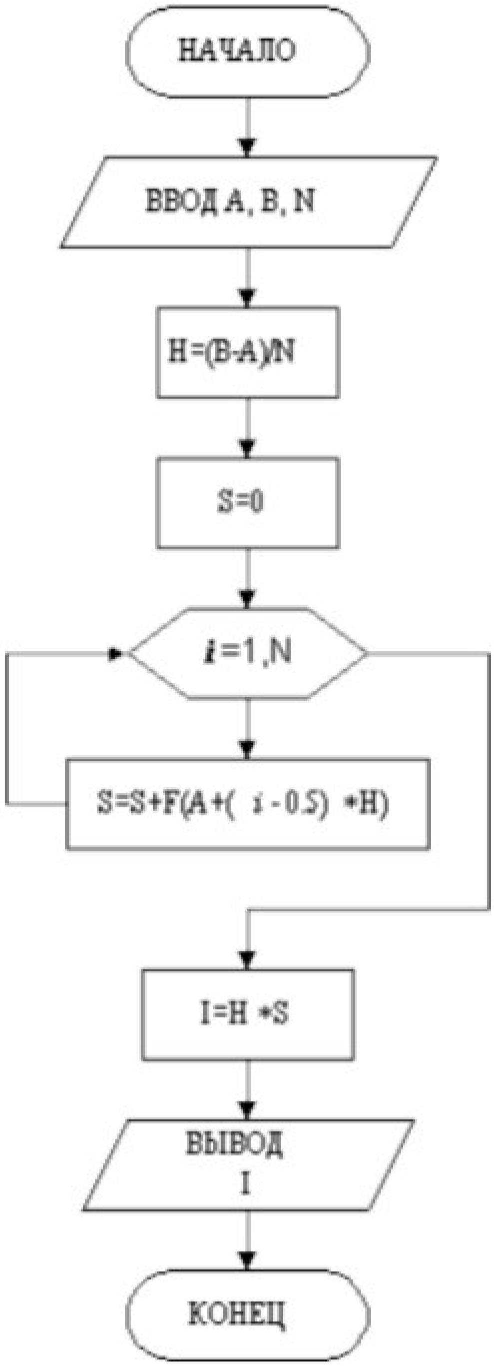
Использует непосредственную замену определенного интеграла интегральной суммой.

На каждом шаге интегрирования функция аппроксимируется полиномом нулевой степени – отрезком, параллельным оси абсцисс. Различают метод левых, правых и средних прямоугольников.

Площадь криволинейной трапеции приближенно заменяется площадью многоугольника, составленного из *n*- прямоугольников. Таким образом, вычисление определенного интеграла сводится к нахождению суммы *n*- элементарных прямоугольников.

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

В качестве точек ξ_i могут выбираться левые ($\xi_i = x_{i-1}$) или правые ($\xi_i = x_i$) границы отрезков, получим формулы левых и правых прямоугольников.


НАЧАЛО
ВВОД A, B, N
H=(B-A)/N
S=0
i=1,N
S=S+F(A+(i-0.5)*H)
I=H*S
ВЫВОД
I
КОНЕЦ

Методы левых и правых прямоугольников

$\int_a^b f(x)dx \approx h_1 y_0 + h_2 y_1 + \cdots + h_n y_{n-1} = \sum_{i=1}^{n} h_i\, y_{i-1}$ - левые прямоугольники

$\int_a^b f(x)dx \approx h_1 y_1 + h_2 y_2 + \cdots + h_n y_n = \sum_{i=1}^{n} h_i y_i$ - правые прямоугольники

При $h_i = h = \frac{b-a}{n} = const$:

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

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

Метод трапеций

Подынтегральную функцию на каждом отрезке $[x_i; x_{i+1}]$ заменяют интерполяционным многочленом первой степени:

$$f(x) \approx \varphi_i(x) = a_i x + b$$

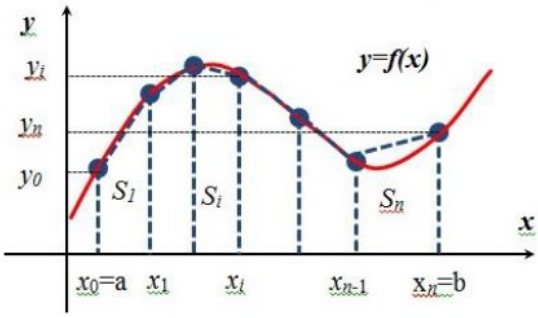
Используют линейную интерполяцию, т.е. график функции $y = f(x)$ представляется в виде ломаной, соединяющий точки (x_i, y_i) . Площадь всей фигуры (криволинейной трапеции):

$$S_{общ} = S_1 + S_2 + \cdots + S_n = \frac{y_0 + y_1}{2}h_1 + \frac{y_1 + y_2}{2}h_2 + \cdots + \frac{y_{n-1} + y_n}{2}h_n$$

$$y_0 = f(a), \qquad y_n = f(b), \qquad y_i = f(x_i), \qquad h_i = x_i - x_{i-1}$$

Складывая все эти равенства, получаем формулу трапеций для численного интегрирования:

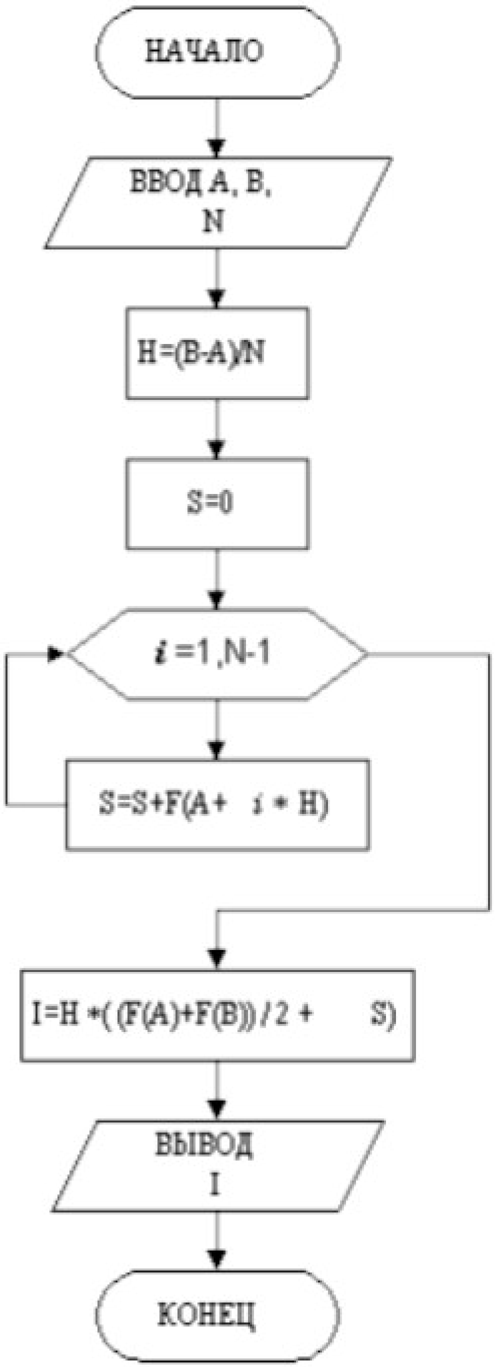
$$\int_a^b f(x)dx = \frac{1}{2}\sum_{i=1}^{n} h_i(y_{i-1} + y_i)$$



При $h_i = h = \frac{b-a}{n} = const$ формула трапеций:

$$\int_a^b f(x)dx = h\cdot\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right)$$

или $$\int_a^b f(x)dx = \frac{h}{2}\cdot\left(y_0 + y_n + 2\sum_{i=1}^{n-1} y_i\right)$$


НАЧАЛО
ВВОД А, В, N
Н=(В-А)/N
S=0
i =1,N-1
S=S+F(A+ i * H)
I=H *((F(A)+F(B)) / 2 + S)
ВЫВОД I
КОНЕЦ

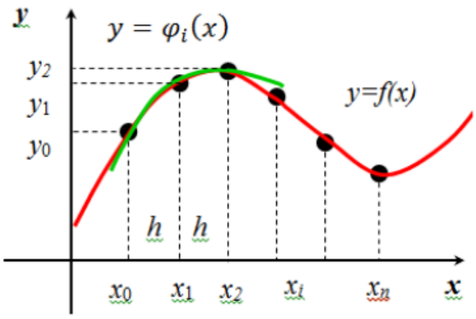
Метод Симпсона

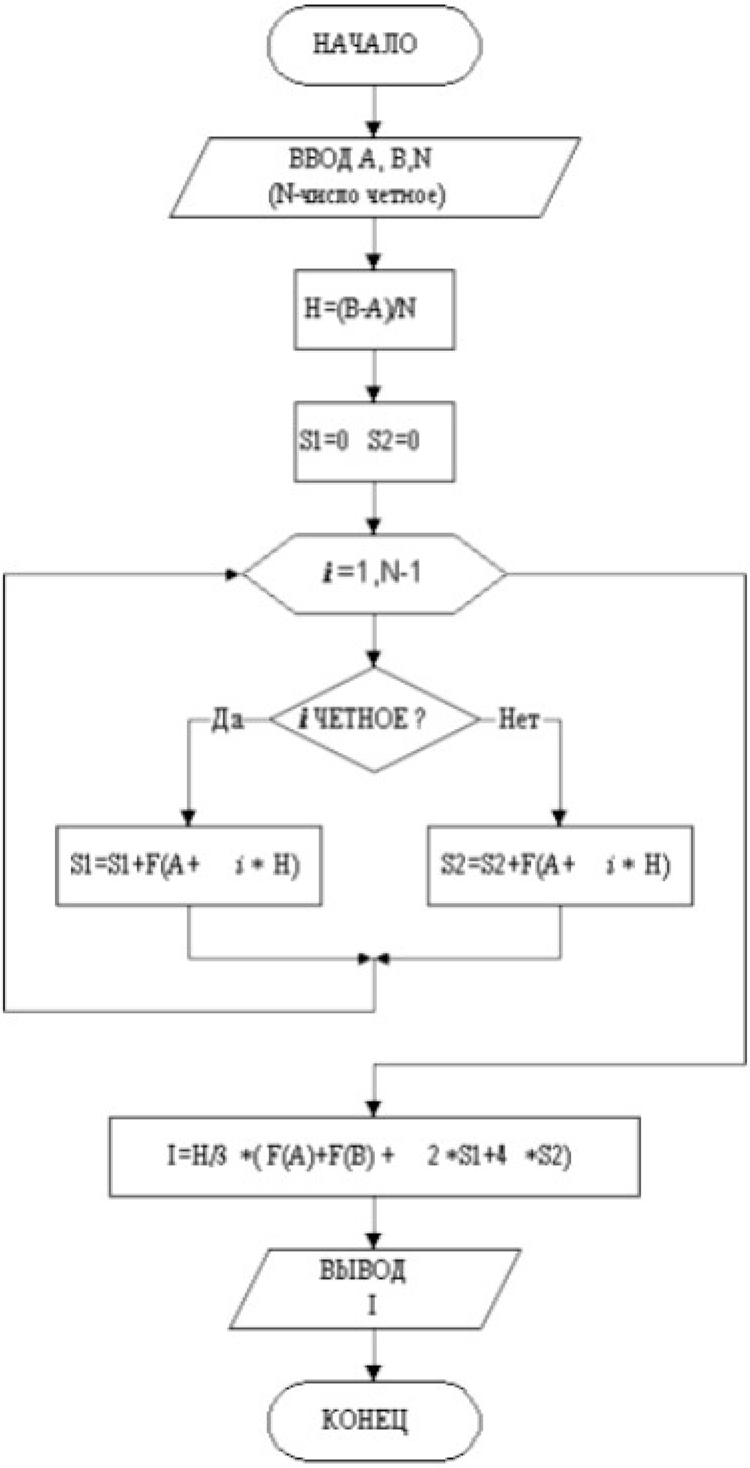
Разобьем отрезок интегрирования [*a*, *b*] на четное число *n* равных частей с шагом *h*. На каждом отрезке $[x_0, x_2]$, $[x_2, x_4]$, …, $[x_{i-1}, x_{i+1}]$, … , $[x_{n-2}, x_n]$ подынтегральную функцию заменим интерполяционным многочленом второй степени:

$$f(x) \approx \varphi_i(x) = a_i x^2 + b_i x + c_i, \quad x_{i-1} \le x \le x_{i+1}$$

Коэффициенты этих квадратных трехчленов могут быть найдены из условий равенства многочлена и подынтегральной функции в узловых точках.

В качестве $\varphi_i(x)$ можно принять интерполяционный многочлен Лагранжа второй степени, проходящий через точки (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}).




НАЧАЛО
ВВОД A, B,N
(N-число четное)
H=(B-A)/N
S1=0 S2=0
i =1,N-1
i ЧЕТНОЕ ?
Да
Нет
S1=S1+F(A+ i * H)
S2=S2+F(A+ i * H)
I=H/3 *(F(A)+F(B) + 2 *S1+4 *S2)
ВЫВОД
I
КОНЕЦ

Листинг программы

https://github.com/SuperJaremy/Integral

Метод левых прямоугольников

https://github.com/SuperJaremy/Integral/blob/8414f52083594bbb56aadaff8f08ce2f735c9006/src/main/java/edu/methods/rectangulars/LeftRectangular.java

```
package edu.methods.rectangulars;

import edu.Equation;
import edu.methods.Rectangular;

public class LeftRectangular extends Rectangular {
    public LeftRectangular(double a, double b, Equation equation, double epsilon) {
        super(a, b, equation, epsilon);
    }

    @Override
    protected double sum(Equation equation, double a, double b, double n) {
        double h = (b - a) / n;
        double x = a;
        double sum = equation.getValue(x);
        for (int i = 1; i < n; i++) {
            x+=h;
            sum+=equation.getValue(x);
        }
        return sum;
    }
}
```

Метод средних прямоугольников

https://github.com/SuperJaremy/Integral/blob/8414f52083594bbb56aadaff8f08ce2f735c9006/src/main/java/edu/methods/rectangulars/MiddleRectangular.java

```
package edu.methods.rectangulars;

import edu.Equation;
import edu.methods.Rectangular;

public class MiddleRectangular extends Rectangular {
    public MiddleRectangular(double a, double b, Equation equation, double epsilon) {
        super(a, b, equation, epsilon);
    }

    @Override
    protected double sum(Equation equation, double a, double b, double n) {
        double h = (b-a)/n;
```

```
        double sum = 0;
        double x = a;
        double previousX;
        for(int i = 1; i<=n;i++){
            previousX = x;
            x+=h;
            sum+=equation.getValue((x+previousX)/2);
        }
        return sum;
    }
}
```

Метод правых прямоугольников

https://github.com/SuperJaremy/Integral/blob/8414f52083594bbb56aadaff8f08ce2f735c9006/src/main/java/edu/methods/rectangulars/RightRectangular.java

```
package edu.methods.rectangulars;

import edu.Equation;
import edu.methods.Rectangular;

public class RightRectangular extends Rectangular {
    public RightRectangular(double a, double b, Equation equation, double epsilon) {
        super(a, b, equation, epsilon);
    }

    @Override
    protected double sum(Equation equation, double a, double b, double n) {
        double h = (b - a) / n;
        double sum = 0;
        double x = a;
        for (int i = 1; i <= n; i++) {
            x += h;
            sum += equation.getValue(x);
        }
        return sum;
    }
}
```

Метод трапеций

https://github.com/SuperJaremy/Integral/blob/8414f52083594bbb56aadaff8f08ce2f735c9006/src/main/java/edu/methods/Trapezoidal.java

```
package edu.methods;

import edu.Equation;

public class Trapezoidal extends Method {
```

```
    public Trapezoidal(double a, double b, Equation equation, double epsilon) {
        super(a, b, equation, epsilon);
    }

    @Override
    protected double useMethod(double a, double b, Equation equation, int n) {
        double h = (b - a) / n;
        double x = a;
        double sum = equation.getValue(x) / 2;
        for (int i = 1; i < n; i++) {
            x += h;
            sum += equation.getValue(x);
        }
        sum += equation.getValue(b) / 2;
        return h * sum;
    }
}
```

Метод Симпсона

https://github.com/SuperJaremy/Integral/blob/8414f52083594bbb56aadaff8f08ce2f735c9006/src/main/java/edu/methods/Simpson.java

```
package edu.methods;

import edu.Equation;

public class Simpson extends Method{
    public Simpson(double a, double b, Equation equation, double epsilon) {
        super(a, b, equation, epsilon);
    }

    @Override
    public double useMethod(double a, double b, Equation equation, int n) {
        double h = (b-a)/n;
        double x = a;
        double sum = equation.getValue(a);
        for(int i=1; i<n; i++){
            x+=h;
            if(i%2==0)
                sum+=2*equation.getValue(x);
            else
                sum+=4*equation.getValue(x);
        }
        return h/3*(sum+equation.getValue(b));
    }

}
```

Результаты выполнения программы

```
Выберите уравнение
1. 5x^3-2x^2+3x-15
2. xsin(3x)-23x+14e^(-x)
3. cos(-15x)*sin(11x)*e^(-3x)
4. x^(1/3)*e^(sin(x)*cos(-2*x))
4
Введите нижнюю границу интегрирования
-4
Введите нижнюю границу инетгрирования
4
Введите точность
0.001
Выберите метод вычисления
1. Метод левых прямоугольников
2. Метод средних прямоугольников
3. Метод правых прямоугольников
4. Метод трапеций
5. Метод Симпсона
5
Полученное значение: -2.248
Количество разбиений: 64
```

Вычисление заданного интеграла

$$\int_1^2 (5x^3 - 2x^2 + 3x - 15)dx$$

Точное значение интеграла: 3.5833

Вычисление методом трапеций (n=10):

i	x_i	y_i	Значение интеграла
0	1	-9	-0.45
1	1.1	-7.4649	-1.1964
2	1.2	-5.6399	-1.7604
3	1.3	-3.4949	-2.1099
4	1.4	-0.9999	-2.2099
5	1.5	1.875	-2.0224
6	1.6	5.16	-1.5064
7	1.7	8.885	-0.6179
8	1.8	13.08	0.69
9	1.9	17.775	2.4675
10	2	23	3.6175

Абсолютная погрешность вычислений: 0.0342

Вывод

Я реализовал методы левых, средних и правых прямоугольников, метод трапеций и метод Симпсона для вычисления определённых интегралов. Самым точным методом оказался метод Симпсона.