Федеральное государственное автономное образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет информационных технологий, механики и оптики»

Лабораторная работа №3
Вариант: Метод прямоугольников

Выполнил:
Воробьев Кирилл
P3231

Преподаватель:
Перл Ольга Вячеславовна

Санкт-Петербург, 2022г

Описание методов

Метод средних прямоугольников:

Суть метода центральных прямоугольников заключается в том, что подынтегральную функцию $y = f(x)$ заменяют отрезке $[x_i, x_{i+1}]$ прямой $y = f(x_i + \frac{h}{2})$, т.е значением функции в середине i-го отрезка.

Метод левых прямоугольников:

Суть метода левых прямоугольников заключается в том, что подынтегральную функцию $y = f(x)$ заменяют на каждом отрезке $[x_i, x_{i+1}]$ прямой $y = f(x_i)$. Площадь i-ой элементарной трапеции S_i вычисляется как площадь прямоугольника со сторонами $h = x_{i+1} - x_i$ и $f(x_i)$.

Метод правых прямоугольников:

Суть метода правых прямоугольников заключается в том, что подынтегральную функцию $y = f(x)$ заменяют на каждом отрезке $[x_i, x_{i+1}]$ прямой $y = f(x_{i+1})$. Площадь i-ой элементарной трапеции S_i вычисляется как площадь прямоугольника со сторонами $h = x_{i+1} - x_i$ и $f(x_{i+1})$.

Расчетные формулы методов

Метод средних прямоугольников

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

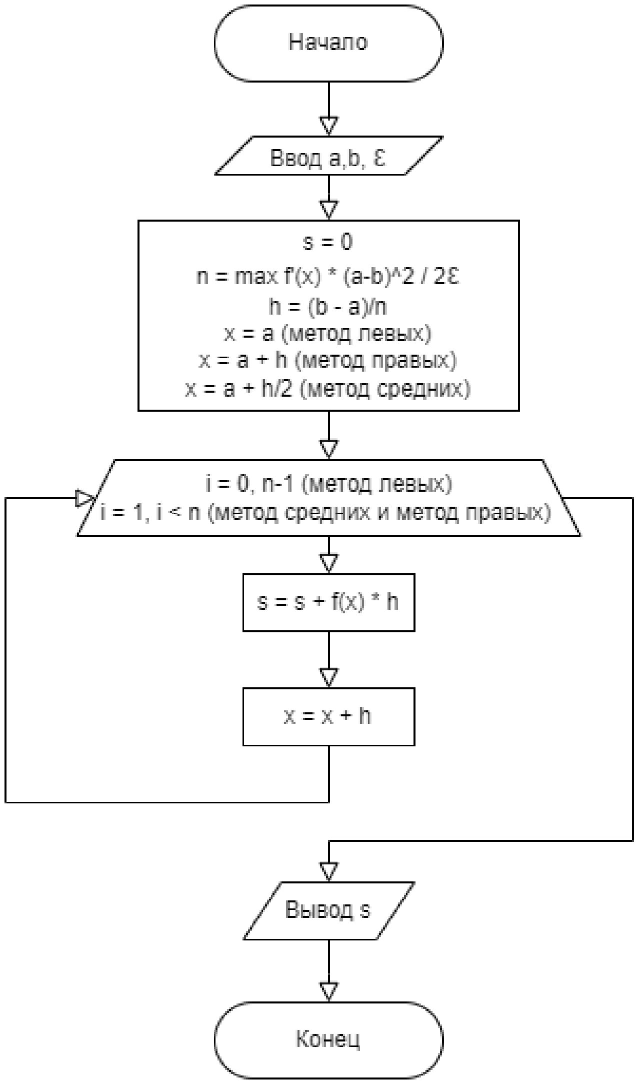
Метод левых прямоугольников

$$\int_a^b f(x)\,dx \approx h \cdot \sum_{i=0}^{n-1} f(x_i)$$

Метод правых прямоугольников

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

Блок-схема методов



Методы в коде

Метод средних прямоугольников

```
public static double doMethod(double leftBorder, int numberOfSegments, double step, Integral integral) throws Imp
    double sum = 0.0;
    for (int i = 1; i <= numberOfSegments; i++) {
        double x = (leftBorder + (i - 1) * step) + (step / 2);
        double functionValue = RectangleMethod.getFunctionValue(integral.getFunction(x), x, integral);
        sum += step * functionValue;
    }
    return sum;
}
```

Метод левых прямоугольников

```
public static double doMethod(double leftBorder, int numberOfSegments, double step, Integral integral) throws Impos
    double sum = 0.0;
    for (int i = 0; i < numberOfSegments; i++) {
        double x = leftBorder + i * step;
        double functionValue = RectangleMethod.getFunctionValue(integral.getFunction(x), x, integral);
        sum += step * functionValue;
    }
    return sum;
}
```

Метод правых прямоугольников

```
public static double doMethod(double leftBorder, int numberOfSegments, double step, Integral integral) throws Imposs
    double sum = 0.0;
    for (int i = 1; i <= numberOfSegments; i++) {
        double x = leftBorder + i * step;
        double functionValue = RectangleMethod.getFunctionValue(integral.getFunction(x), x, integral);
        sum += step * functionValue;
    }
    return sum;
}
```

Метод нахождения кол-ва отрезков

```
static int getNumberOfSegments(double leftBorder, double rightBorder, double maxDerivative,
                               double accuracy) {
    return (int) ((maxDerivative * Math.pow((rightBorder - leftBorder), 2)) / (2 * accuracy));
}
```

Метод нахождения шага интегрирования

```
static double getStep(double leftBorder, double rightBorder, int numberOfSegments) {
    return (rightBorder - leftBorder) / numberOfSegments;
}
```

Метод устранения разрыва в точке при его наличии

```
static double getFunctionValue(Double functionValue, double x, Integral integral) throws ImpossibleToBridgeTheGapExc
    if (functionValue.isNaN() || functionValue.isInfinite()) {
        if (Validator.checkFunctionForDefine(integral, x)) {
            if (Validator.checkFunctionForPossibilityToBridgeTheGap(integral, x)) {
                System.out.println("Обнаружен устраняемый разрыв 1-го рода");
                LinePainter.printLine();
                functionValue = (integral.getFunction( x: x - 0.00001) + integral.getFunction( x: x + 0.00001)) / 2.0;
            } else throw new ImpossibleToBridgeTheGapException();
        } else throw new FunctionDoesNotDefineException();
    }
    return functionValue;
}
```

Вывод

В ходе лабораторной работы я изучил основные методы численного интегрирования. Метод прямоугольников на мой взгляд оказался самым простым в реализации, но при этом для получения желаемой точности приходится сделать значительно больше итераций, чем при использовании остальных методов. Также, можно сделать вывод, что метод средних прямоугольников в случае линейных функций имеет меньшую погрешность, чем методы левых и правых прямоугольников.