Санкт-Петербургский Национальный Исследовательский Университет
Информационных Технологий, Механики и Оптики,
ПИиКт

(Метод Симпсона)

Лабораторная работа №2

по дисциплине

«Вычислительная математика»

Выполнил: Студент группы Р3210
Кадыров Амирджон
Преподаватель:
Перл Ольга Вячеславовна

Санкт-Петербург
2020 г.

Задание (Метод Симпсона)

<u>Лабораторная работа 2 (Интегрирование)</u>

Варианты:

- Метод прямоугольников
  (должен быть реализован расчет 3мя модификациями: левые, правые, средние)
- Метод трапеций
- Метод Симпсона

Пользователь выбирает функцию, интеграл которой он хочет вычислить (3-5 функций), из тех, которые предлагает программа.

В численный метод должен быть передан параметр-агрегат на подпрограмму вычисления значения функции в точке x.

Пользователь задает пределы интегрирования и точность.

NOTE! Если нижний предел интегрирования >= верхнего предела - интеграл должен считаться корректно!

В результате должны получить:

- значение интеграла
- количество разбиений, на которое пришлось разбить
- полученную погрешность

Для оценки погрешности использовать оценку Рунге.

Метод Симпсона (парабол)

Формулой Симпсона называется интеграл от интерполяционного многочлена второй степени на отрезке $[a, b]$:

$$\int_a^b f(x)dx \approx \int_a^b p_2(x)dx = \frac{b-a}{6}\left(f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right),$$

где $f(a)$, $f((a+b)/2)$ и $f(b)$ — значения функции в соответствующих точках (на концах отрезка и в его середине).

Правило Рунге

Интеграл вычисляется по выбранной формуле (прямоугольников, трапеций, парабол Симпсона) при числе шагов, равном n, а затем при числе шагов, равном 2n. Погрешность вычисления значения интеграла при числе шагов, равном 2n, определяется по формуле Рунге:

$\Delta_{2n} \approx \Theta|I_{2n} - I_n|$, для формул прямоугольников и трапеций $\Theta = \frac{1}{3}$, а для формулы Симпсона $\Theta = \frac{1}{15}$.[2]

Таким образом, интеграл вычисляется для последовательных значений числа шагов $N = n_0, 2n_0, 4n_0, \ldots$, где n_0 — начальное число шагов. Процесс вычислений заканчивается, когда для очередного значения N будет выполнено условие $\Delta_{2n} < \epsilon$, где ε — заданная точность.

Листинг

SimpsonMethod

```
class SimpsonMethod {
    private double low;
    private double high;
    private double accuracy;
    private Function function;
    private int number_of_steps;
```

```java
    private double err;
    private double result;
    private int b = 1;

    SimpsonMethod(Function function, double low, double high, double accuracy) {
        this.function = function;
        if (high > low) {
            this.low = low;
            this.high = high;
        } else {
            b = -1;
            this.low = high;
            this.high = low;
        }
        this.accuracy = accuracy;
    }

    private double calculate_integral(int number_of_steps) {
        double sum = 0;
        double h = (high - low) / number_of_steps;
        for (int i = 1; i < number_of_steps; i++) {
            sum += 4 * function.y(low + i * h);
            ++i;
            sum += 2 * function.y(low + i * h);
        }
        return (sum + function.y(low) - function.y(high)) * h / 3;
    }

    void Simpson_Method() {
        double In;
        double I2n;
        for (int n = 4; n <= 10000; n += 2) {
            In = calculate_integral(n);
            int n2 = n*2;
            I2n = calculate_integral(n2);
            if ((Math.abs(I2n - In) / 15) < accuracy) {
                result = I2n;
                number_of_steps = n;
                err = Math.abs(I2n - In) / 15;
                break;
            }
        }
        result *= b;
    }

    double getNumber_of_steps() {
        return number_of_steps;
    }

    double getErr() {
        return err;
    }

    double getResult() {
        return result;
    }
}
```

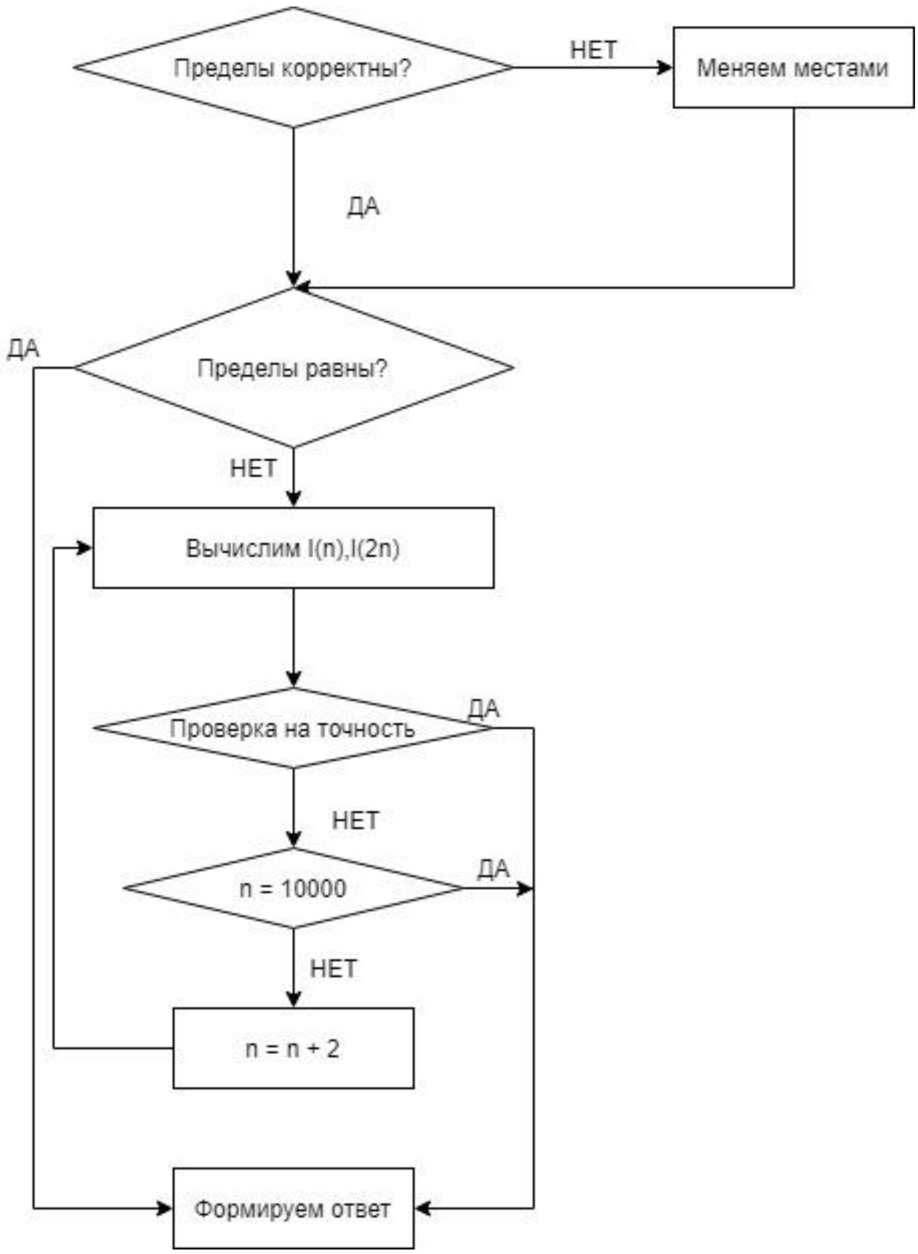
Function

```
class Function {
    private int num;

    Function(int num){
        this.num = num;
    }

    double y(double x){
        if(num == 1){
            return x*x*x+x*x+x+1;
        }else if(num == 2){
            return Math.log(x) + x;
        }else if(num == 3){
            return Math.sin(x) + x;
        }else{
            return 1/(x*x);
        }
    }

}
```

Тестовые данные

F: x^3 + x^2 + x + 1 Low: -1 High: 12 Accuracy: 0,00000001 ОТВЕТ I: 5844.58333 Количество шагов: 4.0 Погрешность: 0.0	F: log(x) + x Low: 5 High: 1 Accuracy: 0,00001 ОТВЕТ I: -16.04707 Количество шагов: 6.0 Погрешность: 9.06E-5
F: sin(x) + x Low: 4 High: 12 Accuracy: 0,001 ОТВЕТ I: 62.50197 Количество шагов: 8.0 Погрешность: 5.927E-4	F: 1/x^2 Low: -1 High: 1 Accuracy: 0,00000001 ОТВЕТ Интеграл расходится

Блок-схема



ВЫВОД:

В ходе выполнения работы было изучено интегрирование функций методом Симпсона, а также методы прямоугольников и трапеций. Результат с исп методом Симпсона получится более точным(т. к. мы отрезки разделяем параболами)