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

Факультет программной инженерии и компьютерной техники

Отчет по лабораторной работе №3

«Численное интегрирование»

По дисциплине «Вычислительная математика»

Вариант 8

Выполнила: Иванова Мария Максимовна

Группа: P3208

Преподаватель: Машина Екатерина Алексеевна

Санкт-Петербург

~ 2024 ~

Цель работы: найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Рабочие формулы используемых методов:

- Метод средних прямоугольников:
  $x_{i-\frac{1}{2}} = \frac{x_{i-1}+x_i}{2}$
  $\int_a^b f(x)dx = h\sum_{i=1}^{n} f(\,x_{i-\frac{1}{2}})$

- Метод трапеций

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

- Метод Симпсона*:*

  $\int_a^b f(x)dx = \frac{h}{3}(y_0 + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2}) + y_n)$

1. **Вычислительная реализация задачи:**

Вычислительная реализация. Вариант 8

① Точное вычисление интеграла:

$$\int_2^3 (3x^3-2x^2-7x-8)\,dx = \left(\frac{3x^4}{4} - \frac{2x^3}{3} - \frac{7x^2}{2} - 8x\right)\Big|_2^3 =$$

$$= \frac{3\cdot3^4}{4} - \frac{2\cdot3^3}{3} - \frac{7\cdot3^2}{2} - 8\cdot3 - \frac{3\cdot2^4}{4} + \frac{2\cdot2^3}{3} + \frac{7\cdot2^2}{2} + 8\cdot2 =$$

$$= \frac{127}{12} = 10,583$$

② Вычисление интеграла по формуле Ньютона-Котеса при $n=6$.

$$\int_2^3 (3x^3-2x^2-7x-8)\,dx = \sum_{i=0}^{n} f(x_i)\,c_n^i = \left(f(x_1)+f(x_5)\right)\cdot\frac{216}{840} +$$

$$+\left(f(x_2)+f(x_4)\right)\cdot\frac{27}{840} + f(x_3)\cdot\frac{272}{840} + \left(f(x_0)+f(x_6)\right)\cdot\frac{41}{840} =$$

$$\approx \cancel{10,583}\ 10,624 \qquad \Delta I = |10,583-10,624| = 0,041\ (0,38\%)$$

Вычисление интеграла по формуле средних прямоугольников:

$$n=10 \qquad h = \frac{b-a}{2} = \frac{3-2}{10} = \frac{1}{10} = 0,1$$

i	1	2	3	4	5	6	7	8	9	10
$x_{i-\frac{1}{2}}$	2,05	2,15	2,25	2,35	2,45	2,55	2,65	2,75	2,85	2,95
$y_{i-\frac{1}{2}}$	-4,91	-2,48	0,2968	3,4386	6,9633	10,889	15,2338	20,015	25,252	30,9621

$$I = 0,1(-4,91-2,48+0,2968+3,4386+6,9633+10,889+15,2338+20,015+$$

$$+25,252+30,9621) = 10,567$$

$$\Delta I = |10,567 - 10,5831| = 0,016\ (0,15\%)$$

Вычисление интеграла по формуле трапеций:

$n=10, \quad h=\frac{3-2}{10}=0,1$

i	0	1	2	3	4	5	6	7	8	9	10
x_i	2	2,1	2,2	2,3	2,4	2,5	2,6	2,7	2,8	2,9	3
y_i	-6	-3,738	-1,136	1,821	5,152	8,875	13,008	17,569	22,576	28,047	34

$$\cancel{I=\frac{-6-3,738-1,136+1,821+5,152+8,875+13,008+17,569+22,576+28,047+34}{10}=}$$

$$I=0,1\left(\frac{-6+34}{2}+(-3,738-1,136+1,821+5,152+8,875+13,008+17,569+22,576+28,047)\right)=0,1(14+97,039)=11,1039$$

$$\Delta I=|10,583-11,1039|=0,5209 \ (4,92\ \%)$$

Вычисление интеграла по формуле Симпсона:

$n=10$

$$I=\frac{0,1}{3}\left[\left(-6+4(-3,738+1,821+8,875+17,569+28,047)+2(-1,136+5,152+13,008+22,576)+34\right)\right]=\frac{0,1}{3}(-6+4\cdot 52,934+2\cdot 39,6+34)=10,6312$$

$$\Delta I=|10,583-10,6312|=0,0482 \ (0,45\%)$$

2. Программная реализация задачи:

Класс, реализующий метод левых прямоугольников

```
package org.example;

import static java.lang.Math.abs;

//метод левых прямоугольников
public class Method1 {
    Functions functions = new Functions();

    private int maxIteration = 4;
    public double solve(double a, double b, int functionChoice, double
epsilon){
        double h = (b - a)/4;
        double x = a;
        double integral1 = 0;
```

```
        double integral2 = 1;
        maxIteration = 4;
        while (Math.abs(integral2 - integral1) > epsilon) {
            x = a;
            integral2 = integral1;
            integral1 = 0;
            for (int i = 0; i <= maxIteration; i++) {
                integral1 += h * functions.getFunction(functionChoice, x);
                x += h;
            }
            maxIteration *= 2;
            h = (b-a)/maxIteration;
        }
        return integral2;
    }

    public int getMaxIteration(){
        return maxIteration/2;
    }

}
```

Класс, реализующий метод правых прямоугольников

```
package org.example;

//метод правых прямоугольников
public class Method2 {
    Functions functions = new Functions();
    private int maxIteration = 4;
    public double solve(double a, double b, int functionChoice, double
epsilon){
        double h = (b - a)/4;
        double x = h;
        double integral1 = 0;
        double integral2 = 1;
        int maxIteration = 4;
        while (Math.abs(integral2 - integral1) > epsilon) {
            x = h;
            integral2 = integral1;
            integral1 = 0;
            for (int i = 0; i <= maxIteration; i++) {
                integral1 += h * functions.getFunction(functionChoice, x);
                x += h;
            }
            maxIteration *= 2;
            h = (b-a)/maxIteration;
        }
        return integral2;
    }
    public int getMaxIteration(){
        return maxIteration/2;
    }

}
```

Класс, реализующий метод средних прямоугольников

```
package org.example;

//метод средних прямоугольников
public class Method3 {
```

```
    private int maxIteration = 4;
    Functions functions = new Functions();
    public double solve(double a, double b, int functionChoice, double
epsilon){
        double h = (b - a)/4;
        double x = h/2;
        double integral1 = 0;
        double integral2 = 1;
        while (Math.abs(integral2 - integral1) > epsilon) {
            x = h/2;
            integral2 = integral1;
            integral1 = 0;
            for (int i = 0; i <= maxIteration; i++) {
                integral1 += h * functions.getFunction(functionChoice, x);
                x += h;
            }
            maxIteration *= 2;
            h = (b-a)/maxIteration;
        }
        return integral2;
    }

    public int getMaxIteration(){
        return maxIteration/2;
    }

}
```

Класс, реализующий метод трапеций

```
package org.example;

//метод трапеций
public class Method4 {


    private int maxIteration = 4;
    Functions functions = new Functions();
    public double solve(double a, double b, int functionChoice, double
epsilon){
        double h = (b - a)/4;
        double x = a;
        double integral1 = 0;
        double integral2 = 1;
        while (Math.abs(integral2 - integral1) > epsilon) {
            x = a;
            integral2 = integral1;
            integral1 = 0;
            for (int i = 0; i <= maxIteration; i++) {
                integral1 += h*(functions.getFunction(functionChoice,
x)+functions.getFunction(functionChoice, x+h))/2;
                x += h;
            }
            maxIteration *= 2;
            h = (b-a)/maxIteration;
        }
        return integral2;
    }

    public int getMaxIteration(){
        return maxIteration/2;
    }
```

```
}
```

Класс, реализующий метод Симпсона

```java
package org.example;

//метод Симпсона
public class Method5 {


    private int maxIteration = 4;
    Functions functions = new Functions();
    public double solve(double a, double b, int functionChoice, double
epsilon){
        double h = (b - a)/4;
        double x = a;
        double integral1 = 0;
        double integral2 = 1;
        while (Math.abs(integral2 - integral1) > epsilon) {
            x = a;
            double [] y = new double[maxIteration+1];
            integral2 = integral1;
            integral1 = 0;
            for (int i = 0; i <= maxIteration; i++) {
                y[i] = functions.getFunction(functionChoice, x);
                x+=h;
            }
            double sum1 = 0;
            double sum2 = 0;
            for (int i = 1; i < maxIteration; i=i+2){
                sum1+=y[i];
            }
            for (int i = 1; i <= maxIteration/2 -1 ; i++ ){
                sum2 += y[i*2];
            }
            integral1 = h/3*(y[0]+4*sum1+2*sum2+y[maxIteration]);
            maxIteration *= 2;
            h = (b-a)/maxIteration;

        }
        return integral2;
    }

    public int getMaxIteration(){
        return maxIteration/2;
    }

}
```

Результаты работы программы:

Выберете функцию:
1 - x^2
2 - 3x^3-2x^2-7x-8
3 - 2x^3-3x^2+5x-9
4 - 1/x
5 - tan(x)

1
Введите границы интегрирования
1 2
Выберете метод интегрирования:
1 - Метод левых прямоугольников
2 - Метод правых прямоугольников
3 - Метод средних прямоугольников
4 - Метод трапеций
5 - Метод Симпсона

5
Выберете точность вычисления
0,01
Значение интеграла: 2.333333333333333
Число разбиения интервала интегрирования: 8.0

Process finished with exit code 0

Выберете функцию:
1 - x^2
2 - 3x^3-2x^2-7x-8
3 - 2x^3-3x^2+5x-9
4 - 1/x
5 - tan(x)

5
Введите границы интегрирования
1 6
Интеграл не существует

Process finished with exit code 0

Выберете функцию:
1 - x^2
2 - 3x^3-2x^2-7x-8
3 - 2x^3-3x^2+5x-9
4 - 1/x
5 - tan(x)

3
Введите границы интегрирования
1 2
Выберете метод интегрирования:
1 - Метод левых прямоугольников
2 - Метод правых прямоугольников
3 - Метод средних прямоугольников
4 - Метод трапеций
5 - Метод Симпсона

4
Выберете точность вычисления
0,01
Значение интеграла: -1.0
Число разбиения интервала интегрирования: 8.0

Вывод:
Во время выполнения работы мне удалось изучить методы численного интегрирования, такие как метод прямоугольников, метод трапеций, метод Ньютона-Котеса и метод Симпсона. Самыми точными оказались методы Симпсона и Ньютона-Котеса, однако они довольно затратные по памяти, так как требуют много шагов и пред подсчётов. Методы прямоугольников и трапеции показались мне более простыми для понимания, но они дают большую погрешность (при не большом количестве интервалов).
Таким образом, метод Симпсона показался мне наиболее удобным и надежным, так как имеет довольно простую формулу в сравнении с методом Ньютона-Котесса.