Федеральное государственное автономное
образовательное учреждение высшего образования
«Санкт-Петербургский национальный исследовательский университет
ИТМО»

ФАКУЛЬТЕТ ПРОГРАММНОЙ ИНЖЕНЕРИИ И КОМПЬЮТЕРНОЙ
ТЕХНИКИ

ЛАБОРАТОРНАЯ РАБОТА № 3

по дисциплине

‘ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА’

Вариант №14

Выполнила
Студентка группы P32151
Ярусова Анна Александровна
Преподаватель:
Машина Екатерина Алексеевна

Санкт-Петербург,
2023

Цель

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами

Задание

1. № варианта определяется как номер в списке группы согласно ИСУ.
2. **Исходные данные:**
   - Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.
   - Пределы интегрирования задаются пользователем.
   - Точность вычисления задается пользователем.
   - Начальное значение числа разбиения интервала интегрирования: n=4.
   - Ввод исходных данных осуществляется с клавиатуры.
3. **Программная реализация задачи:**
   - Реализовать в программе методы по выбору пользователя: Метод прямоугольников (3 модификации: левые, правые, средние), Метод трапеций, Метод Симпсона
   - Методы должны быть оформлены в виде отдельной(ого) функции/класса.
   - Вычисление значений функции оформить в виде отдельной(ого) функции/класса.
   - Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.
   - Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.
4. **Вычислительная реализация задачи:**
   - Вычислить интеграл $\int_2^4 (2x^3 - 2x^2 + 7x - 14)dx$,точно.
   - Вычислить интеграл по формуле Ньютона – Котеса при n = 5.

- Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при n = 10.
- Сравнить результаты с точным значением интеграла.
- Определить относительную погрешность вычислений для каждого метода.
- В отчете отразить последовательные вычисления.

5. **Необязательное задание:**
    - Установить сходимость рассматриваемых несобственных интегралов 2 рода (2-3 функции). Если интеграл - расходящийся, выводить сообщение: «Интеграл не существует».
    - Если интеграл сходящийся, реализовать в программе вычисление несобственных интегралов 2 рода (заданными численными методами).
    - Рассмотреть случаи, когда подынтегральная функция терпит бесконечный разрыв: 1) в точке a, 2) в точке b, 3) на отрезке интегрирования

Описание метода, расчетные формулы

Метод прямоугольников:

Идея этого метода в том, чтобы считать интеграл через вычисление площади под интегральной кривой. Её разбивают на прямоугольники ширины h высоты равной *f*(*x*i) где xi принадлежит отрезку (основанию) нашего прямоугольника.

Различают метод левых, правых и средних прямоугольников.

Площади всех прямоугольников суммируются и получается приближенное значение интеграла, чем больше будет отрезков (а соответственно и прямоугольников), тем точнее будет результат.

Рабочая формула метода: средних, левых и правых соответственно.

$$\int_a^b f(x)dx = h\sum_{i=1}^{n} f(x_{i-1/2})$$

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

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

Метод трапеций:

Метод трапеции работает аналогичным образом -- мы приближаем значение интеграла через площадь под интегралом, но на этот раз трапециями с основаниями параллельными оси y. Получаются прямоугольные трапеции с боковой стороной соединяющей точки *f*(*x*i) и *f*(*x*(i+1)) - значения функции в начале и конце отрезка.

Рабочая формула метода:

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

Метод Симпсона:

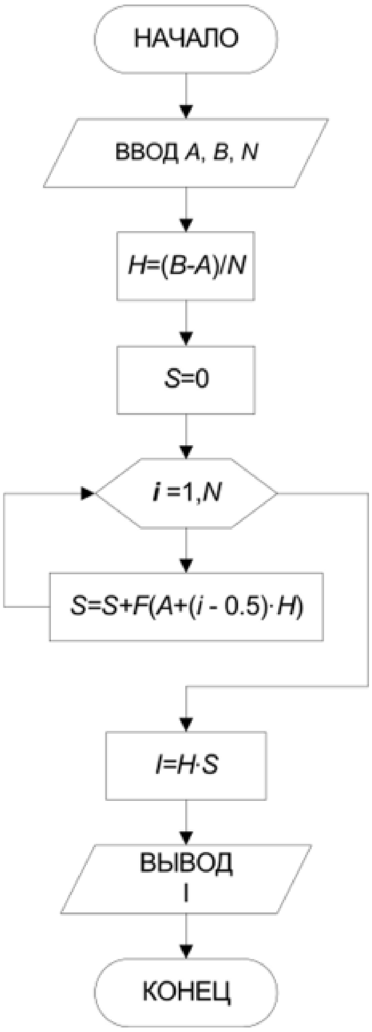
Метод Симпсона приближает подынтегральную площадь параболами, проведенными через три соседние точки -- на которые мы разбиваем наш интервал от a до b. Такую параболу можно построить пользуясь интерполяционным многочленом Лагранжа второй степени, проходящий через точки (x(i-1), y(i-1)), (xi, yi), (x(i+1), y(i+1)).

Рабочая формула метода:

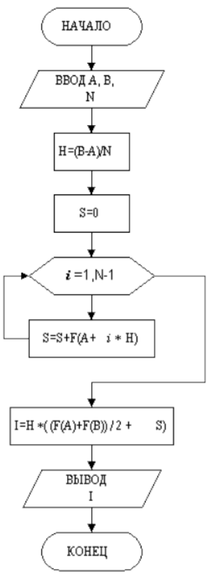
$$\int_a^b f(x) = \frac{h}{3}\left[(y_0 + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2}) + y_n)\right]$$

Блок-схема

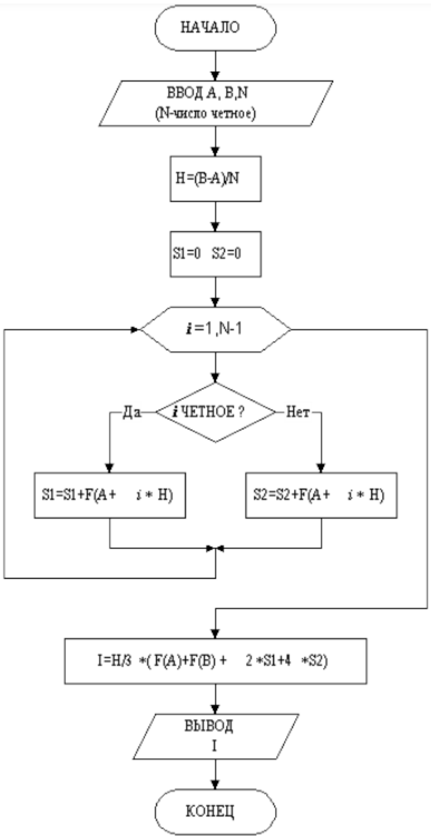
Метод прямоугольников(средних):



<u>Метод трапеций:</u>



<u>Метод Симпсона:</u>



Вычислительная часть

$$\int_2^4 (2x^3 - 2x^2 + 7x - 14)dx$$

1. По формуле Ньютона-Лейбница:

$$\int_2^4 (2x^3 - 2x^2 + 7x - 14)\,dx = (2x^4/4 - 2x^3/3 + 7x^2/2 - 14x) =$$

$$= (x^4/2 - 2x^3/3 + 7x^2/2 - 14x) = (4^4/2 - 2*4^3/3 + 7*4^2/2 - 14*4) -$$

$$- (2^4/2 - 2*2^3/3 + 7*2^2/2 - 14*2) = (256/3) - (-34/3) =$$

$$= 290/3 \approx 96,667$$

2. По формуле Ньютона – Котеса при n =5:

5	$c_5^0 = c_5^5 = \dfrac{19(b-a)}{288}$	$c_5^1 = c_5^4 = \dfrac{75(b-a)}{288}$	$c_5^2 = c_5^3 = \dfrac{50(b-a)}{288}$

$$c_5^0 = c_5^5 = 19 * (4 - 2)/288 = 38/288 = 19/144$$

$c^1_5 = c^4_5 = 75 * (4 - 2)/288 = 150/288 = 25/48$

$c^2_5 = c^3_5 = 50 * (4 - 2)/288 = 100/288 = 25/72$

$\int_2^4 (2x^3 - 2x^2 + 7x - 14)\,dx = c^0_5 * f(2) + c^1_5 * f(2,4) + c^2_5 * f(2,8) +$
$+ c^3_5 * f(3,2) + c^4_5 * f(3,6) + c^5_5 * f(4) = 19/144 * 8 + 25/48 * 2366/125 +$
$25/72 * 4228/125 + 25/72 * 6682/125 + 25/48 * 9824/125 + 19/144 * 110 =$
$= 290/3 \approx 96,667$

Погрешность Δ = 0

3. По формуле средних прямоугольников при n = 10:

   h = (4-2)/10 = 0,2

$\int_2^4 (2x^3 - 2x^2 + 7x - 14)\,dx = 0,2 * \sum_1^{10} f(x_{i-1/2}) = 0,2$ * (482,8) = 96,56

Погрешность Δ =8/75 ≈ 0,107

i	0	1	2	3	4	5	6	7	8	9	10
x_i	2	2,2	2,4	2,6	2,8	3	3,2	3,4	3,6	3,8	4
$x_{i-1/2}$		2,1	2,3	2,5	2,7	2,9	3,1	3,3	3,5	3,7	3,9
$f(x_{i-1/2})$		5201 /500	7927 /500	89 /4	14843 /500	19129 /500	24031 /500	29597 /500	287 /4	42913 /500	50759 /500

4. По формуле трапеций при n = 10:

   h = (4-2)/10 = 0,2

$\int_2^4 (2x^3 - 2x^2 + 7x - 14)\,dx = 0,2 * ((8 + 110)/2 + \sum_1^{n-1} f(x_i)) =$
$= 0,2 * (59 + 43 + 382,4) = 0,2 * 484,4 = 96,88$

Погрешность Δ =16/75 ≈ 0,213

i	0	1	2	3	4	5	6	7	8	9	10
x_i	2	2,2	2,4	2,6	2,8	3	3,2	3,4	3,6	3,8	4
$f(x_i)$	8	1627 /125	2366 /125	3229 /125	4228 /125	43	6682 /125	8161 /125	9824 /125	11683 /125	110

5. По формуле Симпсона при n = 10:
   h = (4-2)/10 = 0,2

$$\int_{2}^{4}(2x^3 - 2x^2 + 7x - 14)\,dx = 0,2\,/\,3 * (8 + 4 * (240,6) + 2 * (184,8) + 110) =$$

$$= 0,2\,/\,3 * 1450 = 290/3 \approx 96,667$$

Погрешность Δ = 0

i	0	1	2	3	4	5	6	7	8	9	10
x_i	2	2,2	2,4	2,6	2,8	3	3,2	3,4	3,6	3,8	4
$f(x_i)$	8	1627 /125	2366 /125	3229 /125	4228 /125	43	6682 /125	8161 /125	9824 /125	11683 /125	110

6. Погрешность при вычислении методом Симпсона и по формуле Ньютона – Котеса получилась нулевая, а самая большая погрешность получилась при вычислении методом трапеций.

Листинг численного метода

Метод левых прямоугольников:

```
public double calculateIntegral(MethodData methodData) {
        double result = 0;
        double h = (methodData.getB() - methodData.getA()) / 
methodData.getN();
        for (int i = 0; i < methodData.getN(); i++) {
            double x = 
methodData.getEquation().apply(methodData.getA() + i * h);
            if (Double.isNaN(x) || Double.isInfinite(x)) {
                throw new IllegalArgumentException("Integral 
doesn't exist, function is not continuous in " + 
(methodData.getA() + i * h));
            } else {
                result += x;
            }
        }
        return result * h;
    }
```

Метод средних прямоугольников:

```
public double calculateIntegral(MethodData methodData) {
```

```
        double result = 0;
        double h = (methodData.getB() - methodData.getA()) /
methodData.getN();
        for (int i = 0; i < methodData.getN(); i++) {
            double x =
methodData.getEquation().apply(methodData.getA() + (i + 0.5) *
h);
            if (Double.isNaN(x) || Double.isInfinite(x)) {
                throw new IllegalArgumentException("Integral
doesn't exist, function is not continuous in " +
(methodData.getA() + (i + 0.5) * h));
            } else {
                result += x;
            }
        }
        return result * h;
    }
```

<u>Метод правых прямоугольников:</u>

```
public double calculateIntegral(MethodData methodData) {
        double result = 0;
        double h = (methodData.getB() - methodData.getA()) /
methodData.getN();
        for (int i = 1; i <= methodData.getN(); i++) {
            double x =
methodData.getEquation().apply(methodData.getA() + i * h);
            if (Double.isNaN(x) || Double.isInfinite(x)) {
                throw new IllegalArgumentException("Integral
doesn't exist, function is not continuous in " +
(methodData.getA() + i * h));
            } else {
                result += x;
            }
        }
        return result * h;
    }
```

<u>Метод трапеций:</u>

```
public double calculateIntegral(MethodData methodData) {
        double result =
methodData.getEquation().apply(methodData.getA()) / 2 +
methodData.getEquation().apply(methodData.getB()) / 2;
        boolean flag = false;
```

```
        double h = (methodData.getB() - methodData.getA()) /
methodData.getN();
        for (int i = 1; i < methodData.getN(); i++) {
            double x =
methodData.getEquation().apply(methodData.getA() + i * h);
            if (Double.isNaN(x) || Double.isInfinite(x)) {
                throw new IllegalArgumentException("Integral
doesn't exist, function is not continuous in " +
(methodData.getA() + i * h));
            } else {
                result += x;
            }
        }
        return result * h;
    }
```

<u>Метод Симпсона:</u>

```
public double calculateIntegral(MethodData methodData) {
        double result =
methodData.getEquation().apply(methodData.getA()) +
methodData.getEquation().apply(methodData.getB());
        double h = (methodData.getB() - methodData.getA()) /
methodData.getN();
        for (int i = 1; i < methodData.getN(); i++) {
            double x =
methodData.getEquation().apply(methodData.getA() + i * h);
            if (Double.isNaN(x) || Double.isInfinite(x)) {
                throw new IllegalArgumentException("Integral
doesn't exist, function is not continuous in " +
(methodData.getA() + i * h));
            } else {
                if (i % 2 == 0) {
                    result += 2 * x;
                } else {
                    result += 4 * x;
                }
            }
        }
        return result * h / 3;
    }
```

Пример работы

Пример 1:

```
Choose the equation:
1. 2x^3 - 2x^2 + 7x - 14 (default)
2. x^3 - 2x^2 + 3x - 4
3. sin x
4. cos x
5. x^2 - 2x + 1
1
Choose the method:
1. Method of left rectangles (default)
2. Method of right rectangles
3. Method of middle rectangles
4. Method of trapezium
5. Method of Simpson
2
Choose epsilon
0,01
Choose left bound
2
Choose right bound
4
Integral value: 96.69157028198242
Number of intervals: 4096
```

Пример 2:

```
Choose the equation:
1. 1 / sqrt(x) (default)
2. 1 / x
3. 1 / (1 - x^2)
3
Choose the method:
1. Method of left rectangles (default)
2. Method of right rectangles
3. Method of middle rectangles
4. Method of trapezium
5. Method of Simpson
4
Choose epsilon
0,01
Choose left bound
0
Choose right bound
2
Integral doesn't exist, function is not continuous in 1.0
```

Вывод

В ходе выполнения лабораторной работы я узнала о различных численных методах вычисления интеграла, написала для этого код на языке Java.