Федеральное государственное автономное

образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет программной инженерии и компьютерной техники

Лабораторная работа № 3

«Численное интегрирование»

По дисциплине «Вычислительная математика»

Вариант 12

Выполнила:

Студентка группы P3217

Русакова Е.Д.

Преподаватель:

Малышева Т.А.

Санкт-Петербург

2024

Оглавление

Цель работы:

Найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

Задание:

Обязательное задание (до 80 баллов)

Исходные данные:

1. Пользователь выбирает функцию, интеграл которой требуется вычислить (3-5 функций), из тех, которые предлагает программа.
2. Пределы интегрирования задаются пользователем.
3. Точность вычисления задается пользователем.
4. Начальное значение числа разбиения интервала интегрирования: n=4.
5. Ввод исходных данных осуществляется с клавиатуры.

Программная реализация задачи:

1. Реализовать в программе методы по выбору пользователя:
   - Метод прямоугольников (3 модификации: левые, правые, средние)
   - Метод трапеций
   - Метод Симпсона
2. Методы должны быть оформлены в виде отдельной(ого) функции/класса.
3. Вычисление значений функции оформить в виде отдельной(ого) функции/класса.
4. Для оценки погрешности и завершения вычислительного процесса использовать правило Рунге.
5. Предусмотреть вывод результатов: значение интеграла, число разбиения интервала интегрирования для достижения требуемой точности.

Вычислительная реализация задачи:

1. Вычислить интеграл, приведенный в таблице 1, точно.
2. Вычислить интеграл по формуле Ньютона – Котеса при $n = 6$.
3. Вычислить интеграл по формулам средних прямоугольников, трапеций и Симпсона при $n = 10$.
4. Сравнить результаты с точным значением интеграла.
5. Определить относительную погрешность вычислений для каждого метода.
6. В отчете ***отразить последовательные вычисления***.

Необязательное задание (до 20 баллов)

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

Задание для варианта 12:

Интеграл для вычислительной части:

$$\int_{1}^{2} (x^3 + 2x^2 - 3x - 12)dx$$

Описание методов для решения интегралов:

Метод прямоугольников

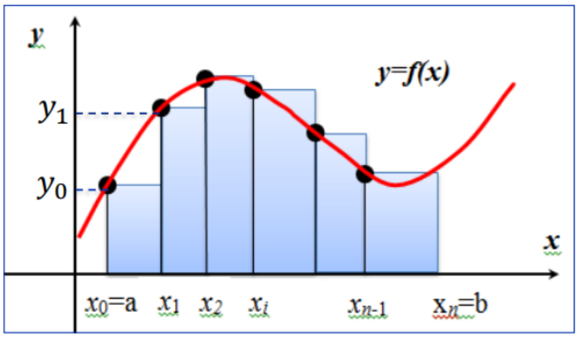
Идея метода: На каждом шаге интегрирования функция аппроксимируется полиномом нулевой степени – отрезком, параллельным оси абсцисс. Площадь криволинейной трапеции приближенно заменяется площадью многоугольника, составленного из n прямоугольников. Различают методы левых, правых и средних прямоугольников.

Метод левых прямоугольников:

$$\int_{a}^{b} f(x)dx = \sum_{i=1}^{n} h_i y_{i-1}$$

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

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

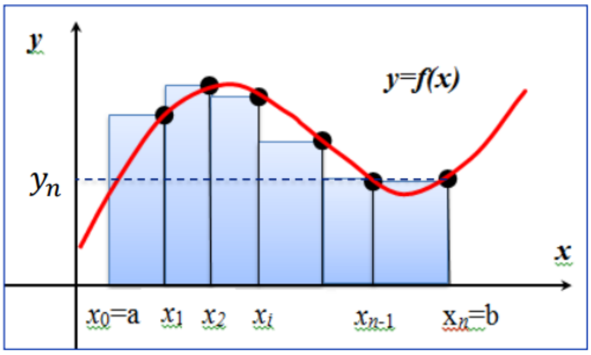


Метод правых прямоугольников:

$$\int_{a}^{b} f(x)dx = \sum_{i=1}^{n} h_i y_i$$

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

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

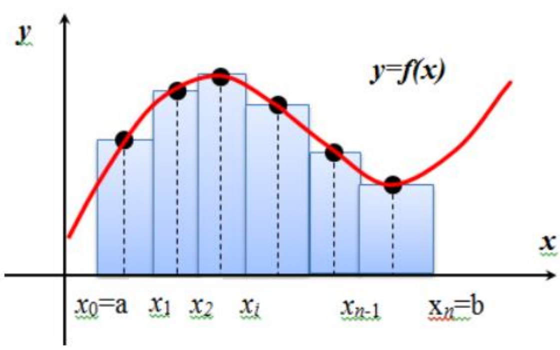


Метод средних прямоугольников:

$$\int_{a}^{b} f(x)dx = \sum_{i=1}^{n} h_i f\left(x_{i-\frac{1}{2}}\right)$$

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

$$\int_{a}^{b} f(x)dx = h \sum_{i=1}^{n} f\left(x_{i-\frac{1}{2}}\right)$$



Метод трапеций

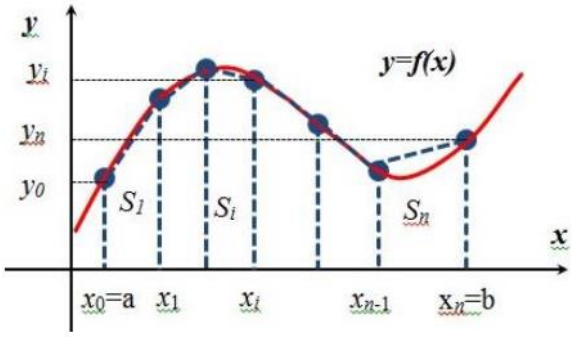
Идея метода: Подынтегральная функция на каждом отрезке $[x_i; x_{i+1}]$ заменяется интерполяционным многочленом первой степени $f(x) \approx \varphi_i(x) = a_i x + b$. График функции $y = f(x)$ представляется в виде ломаной, соединяющий точки (x_i, y_i) . Площадь всей фигуры (криволинейной трапеции) равно сумме площадей трапеций, полученных на каждом отрезке.

Рабочие формулы:

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

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

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



Метод Симпсона

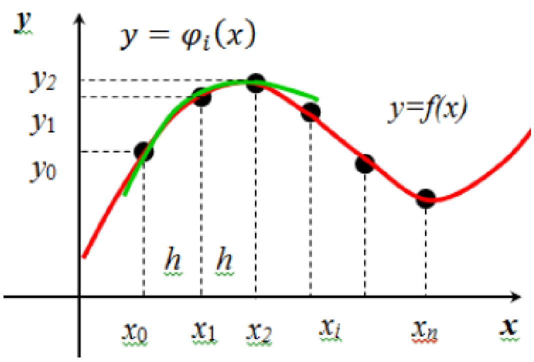
Идея метода: Отрезок интегрирование разбивается на четное число n равных частей с шагом h. На каждом отрезке $[x_i; x_{i+2}]$, $i = 0,2, \dots, n-2$ подынтегральную функцию заменим интерполяционным многочленом второй степени:

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

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 $(x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1})$

Рабочие формулы:

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



Метод Ньютона-Котеса

подынтегральная функция $f(x)$ заменяется на отрезке $[a,b]$ интерполяционным многочленом Лагранжа $L_n(x)$, совпадающий с $f(x)$ в узлах интерполяции $x_0, x_1, \dots, x_n \in [a,b]$. Полином Лагранжа имеет вид:

$$L_n(x) = \sum_{i=0}^{n} f(x_i) L_n^i(x)$$

где $L_n^i(x)$ - коэффициенты Лагранжа (полиномы степени n):

$$L_n^i(x) = \frac{(x-x_0)(x-x_1)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)(x_i-x_1)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}$$

Коэффициенты Котеса:

$$c_n^i = \int_a^b L_n^i(x)dx$$

$$\int_a^b f(x)dx \approx \int_a^b L_n(x)dx = \sum_{i=0}^{n} f(x_i) c_n^i$$

Вычислительная часть:

Интеграл:

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx$$

Точное вычисление:

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = \left(\frac{x^4}{4} + \frac{2}{3}x^3 - \frac{3}{2}x^2 - 12x\right)\Bigg|_1^2$$

$$= \left(\frac{2^4}{4} + \frac{2^4}{3} - \frac{3}{2} * 2^2 - 12 * 2\right) - \left(\frac{1}{4} + \frac{2}{3} - \frac{3}{2} - 12\right)$$

$$= 4 - 6 - 24 + 12 + \frac{16-2}{3} - \frac{1}{4} + \frac{3}{2} = -14 + \frac{14}{3} - \frac{1}{4} + \frac{3}{2}$$

$$= -14 + \frac{14 * 4 - 3 + 3 * 6}{12} = -14 + \frac{71}{12} = -\frac{97}{12} \approx -8{,}0833333$$

Формула Ньютона-Котеса:

$$n = 6, a = 1, b = 2$$

i	0	1	2	3	4	5	6
x_i	1	1,166667	1,333333	1,5	1,666667	1,833333	2
$f(x_i)$	-12	-11,1898	-10,0741	-8,625	-6,81481	-4,61574	-2
c_n^i	0,04881	0,257143	0,032143	0,32381	0,032143	0,257143	0,04881

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = \sum_{i=0}^{n} f(x_i)c_n^i$$
$$= -12 * 0{,}04881 - 11{,}1898 * 0{,}257143 - 10{,}0741 * 0{,}032143 - 8{,}625 * 0{,}32381 - 6{,}81481 * 0{,}032143 - 4{,}61574 * 0{,}257143 - 2 * 0{,}04881 = -8{,}0833333$$

Погрешность вычислений:

$$|R| = \left|I - I_{simp}\right| = |-8{,}0833333 + 8{,}0833333| = 0$$

Метод средних прямоугольников:

$$n = 10, a = 1, b = 2$$

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

i	0	1	2	3	4	5	6	7	8	9
x_i	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9
$x_{i-\frac{1}{2}}$		1,05	1,15	1,25	1,35	1,45	1,55	1,65	1,75	1,85
$f\left(x_{i-\frac{1}{2}}\right)$		-11,7874	-11,2841	-10,6719	-9,94463	-9,09637	-8,12113	-7,01287	-5,76562	-4,37337

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = h\sum_{i=1}^{n} f\left(x_{i-\frac{1}{2}}\right)$$
$$= 0{,}1(-11{,}7874 - 11{,}2841 - 10{,}6719 - 9{,}94463 - 9{,}09637 - 8{,}12113 - 7{,}01287 - 5{,}76562 - 4{,}37337) = 0{,}1 * (-80{,}8875) = -8{,}08875$$

Погрешность вычислений:

$$|R| = \left|I - I_{simp}\right| = |-8{,}0833333 + 8{,}08875| \approx 0{,}005417$$

Метод трапеций:

$$n = 10, a = 1, b = 2$$

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

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2
$y_i = f(x_i)$	-12	-11,549	-10,992	-10,323	-9,536	-8,625	-7,584	-6,407	-5,088	-3,621	-2

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = h\left(\frac{y_0 + y_n}{2} + \sum_{i=1}^{n-1} y_i\right)$$
$$= 0{,}1\left(\left(\frac{-12-2}{2}\right) + (-11{,}549 - 10{,}992 - 10{,}323 - 9{,}536 - 8{,}625 - 7{,}584 - 6{,}407 - 5{,}088 - 3{,}621)\right) = 0{,}1 * (-80{,}725) = -8{,}0725$$

Погрешность вычислений:

$$|R| = |I - I_{simp}| = |-8{,}0833333 + 8{,}0725| = 0{,}0108333$$

Метод Симпсона:

$$n = 10, a = 1, b = 2$$

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

i	0	1	2	3	4	5	6	7	8	9	10
x_i	1	1,1	1,2	1,3	1,4	1,5	1,6	1,7	1,8	1,9	2
$y_i = f(x_i)$	-12	-11,549	-10,992	-10,323	-9,536	-8,625	-7,584	-6,407	-5,088	-3,621	-2

$$\int_1^2 (x^3 + 2x^2 - 3x - 12)dx = \frac{h}{3}[y_0 + 4(y_1 + y_3 + \cdots + y_{n-1}) + 2(y_2 + y_4 + \cdots + y_{n-2}) + y_n]$$
$$= \frac{0{,}1}{3}(-12 + 4(-11{,}549 - 10{,}323 - 8{,}625 - 6{,}407 - 3{,}621)$$
$$+ 2(-10{,}992 - 9{,}536 - 7{,}584 - 5{,}088) - 2) = \frac{0{,}1}{3} * (-242{,}5) = -8{,}0833333$$

Погрешность вычислений:

$$|R| = |I - I_{simp}| = |-8{,}0833333 + 8{,}0833333| = 0$$

Программная реализация:

Описание разработанной программы:

Разработанная программа позволяет найти численное решение интеграла, который пользователь выбирает из предложенных программой, одним из предложенных программой методов (прямоугольников, трапеций или Симпсона). Границы интегрирования и точность вычисления вводятся пользователем с клавиатуры. Также пользователь может выбрать нужно ли выводить все итерации решения или только ответ. Также программы выводит точное значение интеграла, посчитанное по формуле Ньютона-Лейбница из заданных первообразных для каждой функции. Программа может определить расходящийся интеграл, но только если разрыв на границе интервала.

Исходный код программы:

Полный код программы выложен на Github и доступен по ссылке [lenapochemy/comp-math-lab3: вычмат лаба 3 (github.com)]

Далее приведен код классов, которые отвечают за решение

RectangleMethod.java - Метод прямоугольников

```
package methods;

import data.IntegratedFunction;

public class RectangleMethod extends AbstractMethod{
```

```java
    private TypeOfRectangleMethod type;
    public RectangleMethod(IntegratedFunction function, double eps, double a,
double b, TypeOfRectangleMethod type){
        super(function, eps, a, b);
        this.type = type;
        this.k = 2;
    }


    public void oneTypeSolve(){
        n = firstN;
        this.result_last = 0;
        this.delta = Double.MAX_VALUE;
        while (true) {
            h = (b - a) / n;
//            System.out.println(a + " " + b + " " + h);
            sum = 0;
            for (double i = a; i <= b; i = rounding(i + h)) {
                switch (type) {
                    case MEDIUM -> {
                        x = i + h / 2;
                        if (x > b) continue;
                    }
                    case LEFT -> {
                        if (i == b) continue;
                        x = i;
                    }
                    default -> {
                        if (i == a) continue;
                        x = i;
                    }
                }
                f_x = function.function.apply(x);
                if(f_x == Double.POSITIVE_INFINITY || f_x ==
Double.NEGATIVE_INFINITY) {
                    x = x + 0.0000001;
                    f_x = function.function.apply(x);
                }
                sum += f_x;
            }
            result = h * sum;
            writeIteration("Новое значение интеграла: " + result + " при
числе разбиений: " + n);
            writeIteration("------------------------------");
            if(checkEndCondition()) break;
            result_last = result;
            n = n * 2;
        }
        writeResult("Результат:\nЗначение интеграла: " + result + " при числе
разбиений: " + n);

    }
    @Override
    public void solve() {
        if(type == TypeOfRectangleMethod.ALL){
            writeResult("------------------------------");
            writeResult("Метод левых прямоугольников:\n");
            type = TypeOfRectangleMethod.LEFT;
            oneTypeSolve();
```

```
            writeResult("------------------------------");
            writeResult("Метод средних прямоугольников:\n");
            type = TypeOfRectangleMethod.MEDIUM;
            oneTypeSolve();
            writeResult("------------------------------");
            writeResult("Метод правых прямоугольников:\n");
            type = TypeOfRectangleMethod.RIGHT;
            oneTypeSolve();
        } else oneTypeSolve();
    }
}
```

TrapezoidMethod.java – Метод трапеций

```
package methods;

import data.IntegratedFunction;

public class TrapezoidMethod extends AbstractMethod{

    public TrapezoidMethod(IntegratedFunction function, double eps, double a,
double b){
        super(function, eps, a, b);
        this.k = 2;
    }

    @Override
    public void solve(){
        n = firstN;
        delta = Double.MAX_VALUE;

        while (true) {
            h = (b - a) / n;
            sum = 0;
//            sum = (function.apply(a) + function.apply(b)) / 2;
            for (double i = a ; i <= b ; i = rounding(i + h)) {
                f_x = function.function.apply(i);
                if(f_x == Double.POSITIVE_INFINITY || f_x ==
Double.NEGATIVE_INFINITY) {
                    x = x + 0.0000001;
                    f_x = function.function.apply(x);
                }
                if(i == a || i == b) sum += (f_x / 2);
                else sum += f_x;
//                writeIteration("x = " + i + " f(x) = " + f_x + " sum = " +
sum);
            }
            result = h * sum;
            writeIteration("Новое значение интеграла: " + result + " при
числе разбиений: " + n);
            writeIteration("------------------------------");
            if(checkEndCondition()) break;
            result_last = result;
            n = n * 2;
        }
        writeResult("Результат:\nЗначение интеграла: " + result + " при числе
разбиений: " + n);

    }
}
```

SimpsonMethod.java – Метод Симпсона

```java
package methods;

import data.IntegratedFunction;

public class SimpsonMethod extends AbstractMethod{


    public SimpsonMethod(IntegratedFunction function, double eps, double a,
double b){
        super(function, eps, a, b);
        this.k = 4;
    }
    @Override
    public void solve() {
        n = firstN;

        while (true) {
            h = (b - a) / n;
//              sum = function.apply(a) + function.apply(b);
            sum = 0;

            int j = 0;
            for (double i = a; i <= b; i = rounding(i + h), j++) {
                f_x = function.function.apply(i);
                if(f_x == Double.POSITIVE_INFINITY || f_x ==
Double.NEGATIVE_INFINITY) {
                    x = x + 0.0000001;
                    f_x = function.function.apply(x);
                }
//                  writeIteration("x = " + i + " f(x) = " + f_x);
                if(i == a || i == b){
                    sum += f_x;
                } else {
                    if (j % 2 == 0) {
                        sum += 2 * f_x;
                    } else sum += 4 * f_x;
                }
            }

            result = (h / 3) * sum;
            writeIteration("Новое значение интеграла: " + result + " при
числе разбиений: " + n);
            writeIteration("------------------------------");
            if(checkEndCondition()) break;
            result_last = result;
            n = n * 2;
        }
        writeResult("Результат:\nЗначение интеграла: " + result + " при числе
разбиений: " + n);

    }
}
```

Примеры работы программы:

Пример 1

```
Выберите функцию для решения:
    1. 1 / sqrt(x)
    2. x³-4.5x²-9.21x-0.383
    3. x³+2x²-3x-12
    4. ln(x) / sqrt(x)
    5. 1 / (1-x)
3
Введите значение точности [0.000001; 1]: 0.001
Введите левой границы интервала: 2
Введите правой границы интервала: 3
Выберите метод решения уравнения:
    1. Метод прямоугольников
    2. Метод трапеций
    3. Метод Симпсона
3
Нужно выводить результат каждой итерации решения? (y/n) y
Новое значение интеграла: 9.416666666666666 при числе разбиений: 4
------------------------------
Новое значение интеграла: 9.416666666666666 при числе разбиений: 8
------------------------------
Результат:
Значение интеграла: 9.416666666666666 при числе разбиений: 8
------------------------------
Точное значение функции = 9.416673000000007
```

Пример 2

```
Выберите функцию для решения:
    1. 1 / sqrt(x)
    2. x³-4.5x²-9.21x-0.383
    3. x³+2x²-3x-12
    4. ln(x) / sqrt(x)
    5. 1 / (1-x)
3
Введите значение точности [0.000001; 1]: 0.001
Введите левой границы интервала: 2
Введите правой границы интервала: 3
Выберите метод решения уравнения:
    1. Метод прямоугольников
    2. Метод трапеций
    3. Метод Симпсона
2
Нужно выводить результат каждой итерации решения? (y/n) y
```

```
Новое значение интеграла: 9.515625 при числе разбиений: 4
------------------------------
Новое значение интеграла: 9.44140625 при числе разбиений: 8
------------------------------
Новое значение интеграла: 9.4228515625 при числе разбиений: 16
------------------------------
Новое значение интеграла: 9.418212890625 при числе разбиений: 32
------------------------------
Новое значение интеграла: 9.41705322265625 при числе разбиений: 64
------------------------------
Результат:
Значение интеграла: 9.41705322265625 при числе разбиений: 64
------------------------------
Точное значение функции = 9.416673000000007
```

Пример 3

```
Выберите функцию для решения:
    1. 1 / sqrt(x)
    2. x³-4.5x²-9.21x-0.383
    3. x³+2x²-3x-12
    4. ln(x) / sqrt(x)
    5. 1 / (1-x)
5
Введите значение точности [0.000001; 1]: 0.0001
Введите левой границы интервала: 2
Введите правой границы интервала: 4
Выберите метод решения уравнения:
    1. Метод прямоугольников
    2. Метод трапеций
    3. Метод Симпсона
1
Выберите тип метода прямоугольников:
    1. Левые
    2. Правые
    3. Средние
    4. Все типы сразу
4
Нужно выводить результат каждой итерации решения? (y/n) n
------------------------------
Метод левых прямоугольников:

Результат:
Значение интеграла: -1.098775066745417 при числе разбиений: 4096
```

```
------------------------------
Метод средних прямоугольников:

Результат:
Значение интеграла: -1.0985761265510816 при числе разбиений: 64
------------------------------
Метод правых прямоугольников:

Результат:
Значение интеграла: -1.098449545912084 при числе разбиений: 4096
------------------------------
Точное значение функции = -1.0986122886681098
```

Пример 4

```
Выберите функцию для решения:
    1. 1 / sqrt(x)
    2. x³-4.5x²-9.21x-0.383
    3. x³+2x²-3x-12
    4. ln(x) / sqrt(x)
    5. 1 / (1-x)
1
Введите значение точности [0.000001; 1]: 0.001
Введите левой границы интервала: -1
Функция неопределена на отрицательных числах, выберите неотрицательное значение левой границы интервала
Введите левой границы интервала: 0
Введите правой границы интервала: 1
Выберите метод решения уравнения:
    1. Метод прямоугольников
    2. Метод трапеций
    3. Метод Симпсона
3
Нужно выводить результат каждой итерации решения? (y/n) y
Новое значение интеграла: 264.89374078722017 при числе разбиений: 4
------------------------------
Новое значение интеграла: 94.72447129060706 при числе разбиений: 8
------------------------------
Новое значение интеграла: 39.72160815092106 при числе разбиений: 16
------------------------------
Новое значение интеграла: 18.247683487156582 при числе разбиений: 32
------------------------------
```

```
------------------------------
Новое значение интеграла: 9.208355433280644 при числе разбиений: 64
------------------------------
Новое значение интеграла: 5.250708385391823 при числе разбиений: 128
------------------------------
Новое значение интеграла: 3.4776171500057522 при числе разбиений: 256
------------------------------
Новое значение интеграла: 2.6722585490161253 при числе разбиений: 512
------------------------------
Новое значение интеграла: 2.303794035525132 при числе разбиений: 1024
------------------------------
Новое значение интеграла: 2.134946335461949 при числе разбиений: 2048
------------------------------
Новое значение интеграла: 2.057925458718522 при числе разбиений: 4096
------------------------------
Новое значение интеграла: 2.0232377725965063 при числе разбиений: 8192
------------------------------
Новое значение интеграла: 2.0080100283667517 при числе разбиений: 16384
------------------------------
Новое значение интеграла: 2.001643833364257 при числе разбиений: 32768
------------------------------
Результат:
Значение интеграла: 2.001643833364257 при числе разбиений: 32768
------------------------------
Точное значение функции = 2.0
```

Пример 5

```
Выберите функцию для решения:
    1. 1 / sqrt(x)
    2. x³-4.5x²-9.21x-0.383
    3. x³+2x²-3x-12
    4. ln(x) / sqrt(x)
    5. 1 / (1-x)
5
Введите значение точности [0.000001; 1]: 0.0001
Введите левой границы интервала: 0
Введите правой границы интервала: 1
Интеграл расходится
```

Вывод:

При выполнении лабораторной работы я познакомилась с различными методами численного интегрирования и выполнила программную реализацию некоторых из них.