Университет ИТМО, факультет ПИиКТ

Лабораторная работа №2

Дисциплина: Вычислительная математика

Вариант 19

Выполнил: Щелыкалов Виктор

Группа: Р3214

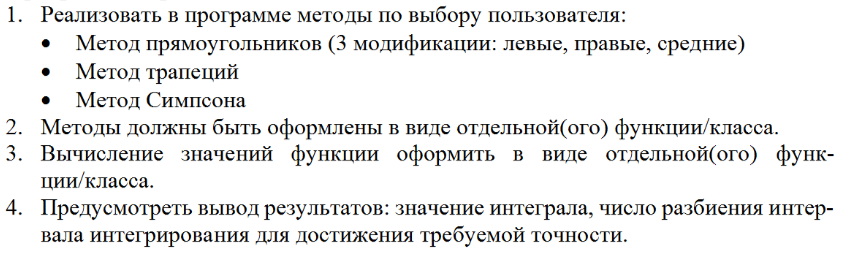
Преподаватель:

Малышева Татьяна Алексеевна

Санкт-Петербург, 2020 год

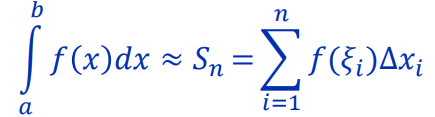
**Цель работы:** найти приближенное значение определенного интеграла с требуемой точностью различными численными методами.

**Текст задания:**

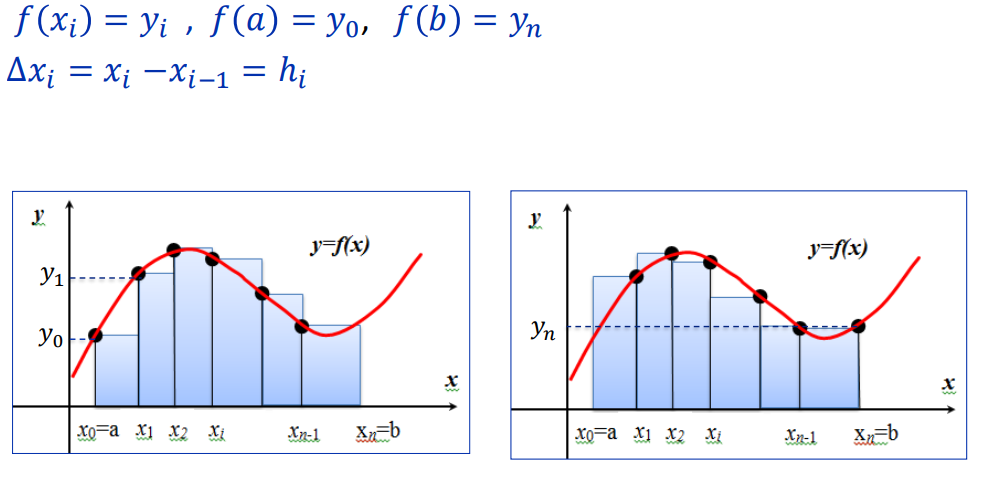


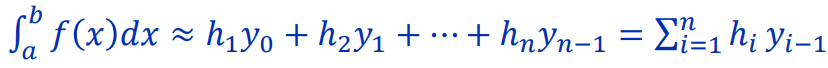
**Ход работы:**

Метод прямоугольников использует непосредственную замену определенного интеграла интегральной суммой. На каждом шаге интегрирования функция аппроксимируется полиномом нулевой степени – отрезком, параллельным оси абсцисс. Различают метод левых, правых и средних прямоугольников. Площадь криволинейной трапеции приближенно заменяется площадью многоугольника, составленного из n- прямоугольников. Таким образом, вычисление определенного интеграла сводится к нахождению суммы n- элементарных прямоугольников.

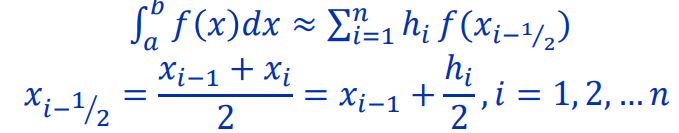


Обозначим



 - левые прямоугольники

 - правые прямоугольники

 - средние прямоугольники

**Листинг:**

import java.util.ArrayList;

import java.util.List;

public class Solver {

private static double summ;

private static double answer;

private static double currE;

private static double previousAns;

public static List<Double> rightRectangle(int u, double a, double b, int n, double e) {

summ = 0;

answer = 0;

currE = Double.MAX\_VALUE;

double h;

double currA;

List<Double> list = new ArrayList<>();

while (currE > e) {

currA = a;

h = (b - a) / n; // считаем h

summ = 0;

switch (u) {

case 1:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (Math.pow(currA, 2) + 1) / 2;

}

break;

case 2:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (Math.pow(currA, 3) + 2 \* Math.pow(currA, 2) - currA + 1);

}

break;

case 3:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (2 \* currA - 1);

}

break;

case 4:

for (int i = 1; i <= n; i++) {

currA += h;

summ += 2;

}

break;

default:

System.out.println("Некорректный номер уравнения!");

System.exit(0);

}

previousAns = answer;

answer = summ \* h;

n = n \* 2;

currE = Math.abs(previousAns - answer);

}

list.add(0, (double) n);

list.add(1, answer);

return list;

}

public static List<Double> leftRectangle(int u, double a, double b, int n, double e) {

answer = 0;

currE = Double.MAX\_VALUE;

double h;

double currA;

List<Double> list = new ArrayList<>();

while (currE > e) {

currA = a;

summ = 0;

h = (b - a) / n; // считаем h

switch (u) {

case 1:

for (int i = 1; i <= n; i++) {

summ += (Math.pow(currA - h / 2, 2) + 1) / 2;

currA += h;

}

break;

case 2:

for (int i = 1; i <= n; i++) {

summ += (Math.pow(currA - h / 2, 3) + 2 \* Math.pow(currA - h / 2, 2) - (currA - h / 2) + 1);

currA += h;

}

break;

case 3:

for (int i = 1; i <= n; i++) {

summ += (2 \* (currA - h / 2) - 1);

currA += h;

}

break;

case 4:

for (int i = 1; i <= n; i++) {

summ += 2;

currA += h;

}

break;

default:

System.out.println("Некорректный номер уравнения!");

System.exit(0);

}

previousAns = answer;

answer = summ \* h;

n = n \* 2;

currE = Math.abs(previousAns - answer);

}

list.add(0, (double) n);

list.add(1, answer);

return list;

}

public static List<Double> middleRectangle(int u, double a, double b, int n, double e) {

answer = 0;

currE = Double.MAX\_VALUE;

double h;

double currA;

List<Double> list = new ArrayList<>();

while (currE > e) {

currA = a;

summ = 0;

h = (b - a) / n; // считаем h

switch (u) {

case 1:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (Math.pow(currA - h / 2, 2) + 1) / 2;

}

break;

case 2:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (Math.pow(currA - h / 2, 3) + 2 \* Math.pow(currA - h / 2, 2) - (currA - h / 2) + 1);

}

break;

case 3:

for (int i = 1; i <= n; i++) {

currA += h;

summ += (2 \* (currA - h / 2) - 1);

}

break;

case 4:

for (int i = 1; i <= n; i++) {

currA += h;

summ += 2;

}

break;

default:

System.out.println("Некорректный номер уравнения!");

System.exit(0);

}

previousAns = answer;

answer = summ \* h;

n = n \* 2;

currE = Math.abs(previousAns - answer);

}

list.add(0, (double) n);

list.add(1, answer);

return list;

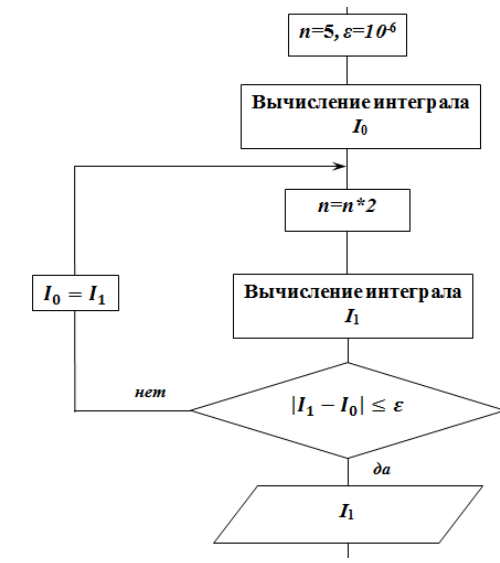
}

}

**Примеры работы программы:**

|  |  |
| --- | --- |
| Выберите номер функции:  1: (x^2+1)/2  2: (x^3 + 2x^2 - x + 1)  3: (2x-1)  4: (2 + 0\*x)  3  Введите верхний и нижний пределы интегрирования соответственно:  9 1  Введите точность:  8  По формуле левых прямоугольников: 64.0, число разбиений - 32.0  По формуле средних прямоугольников: 72.0, число разбиений - 16.0  По формуле правых прямоугольников: 80.0, число разбиений - 16.0 | Выберите номер функции:  1: (x^2+1)/2  2: (x^3 + 2x^2 - x + 1)  3: (2x-1)  4: (2 + 0\*x)  1  Введите верхний и нижний пределы интегрирования соответственно:  -12 1  Введите точность:  0,001  Меняю местами  По формуле левых прямоугольников: -294.66755310786357, число разбиений - 2097152.0  По формуле средних прямоугольников: -294.6665793657303, число разбиений - 2048.0  По формуле правых прямоугольников: -294.6657802272936, число разбиений - 1048576.0 |
| Выберите номер функции:  1: (x^2+1)/2  2: (x^3 + 2x^2 - x + 1)  3: (2x-1)  4: (2 + 0\*x)  4  Введите верхний и нижний пределы интегрирования соответственно:  1 -3  Введите точность:  0,01  По формуле левых прямоугольников: 8.0, число разбиений - 16.0  По формуле средних прямоугольников: 8.0, число разбиений - 16.0  По формуле правых прямоугольников: 8.0, число разбиений - 16.0 | Выберите номер функции:  1: (x^2+1)/2  2: (x^3 + 2x^2 - x + 1)  3: (2x-1)  4: (2 + 0\*x)  2  Введите верхний и нижний пределы интегрирования соответственно:  -0,45 -2,3  Введите точность:  0,002  По формуле левых прямоугольников: 5.458003526894453, число разбиений - 512.0  По формуле средних прямоугольников: 5.458833632469182, число разбиений - 128.0  По формуле правых прямоугольников: 5.458747725296023, число разбиений - 128.0 |

**Блок-схема:**



**Вычисления:**

*Точно:*

|  |  |
| --- | --- |
|  |  |

*Метод трапеций:*

*Погрешность вычислений составляет:*

**Вывод:** в результате выполнения работы я познакомился с методами решения интегралов путем реализации Java-программы. Освоил метод треугольников и на практике понял, почему средняя модификация дает наиболее точный результат(т.к. берется средняя точка на графике). Научился пользоваться методом трапеций на листике, и теперь я могу считать интегралы быстро и необычно.