САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО

Дисциплина: Математический анализ

Отчет

по лабораторной работе №1

«Интеграл Римана (вариант 78)»

Выполнил(а): Миленин Иван Александрович

студ. гр. М3135

Санкт-Петербург

2021

Цель работы: составить интегральную сумму для интеграла Римана функции $f(x) = x^3$ по промежутку [0;2]. Вычислить ее и найти предел. Доказать, что соответствующий интеграл существует. Проверить с помощью формулы Ньютона-Лейбница. написать программу, вычисляющую интегральные суммы для данной функции на данном отрезке. Входные данные для программы: число точек разбиения, способ выбора оснащения (левые, правые, средние, случайные точки). Разбиение равномерное.

Теоретическая часть

Рассмотрим достаточное условие интегрируемости:

«Если функция $f(x)$ кусочно-непрерывна на промежутке [a, b], то на этом промежутке она интегрируема, т.е. существует $\int_a^b f(x)dx$»

Докажем, что функция $f(x) = x^3$ является непрерывной на промежутке [0;2].

Функция $y = f(x)$ называется непрерывной на отрезке [a; b], если она является непрерывной в интервале (a; b), непрерывной справа в точке α, то есть $f(a+0) = f(a)$ и непрерывной слева в точке b, то есть $f(b-0) = f(b)$.

Функция $f(x)$ называется непрерывной справа в точке α, если $f(a+0) = \lim\limits_{x\to a+0} f(x) = f(a)$.

Функция $f(x)$ называется непрерывной слева в точке b, если $f(b-0) = \lim\limits_{x\to b-0} f(x) = f(b)$.

Нам дано $f(x) = x^3; a = 0; b = 2$.

$f(a+0) = f(0+0) = \lim\limits_{x\to 0+0} x^3 = 0 = f(0) = f(a)$.

$f(b-0) = f(2-0) = \lim\limits_{x\to 2-0} x^3 = 8 = f(2) = f(a)$.

Таким образом функция является непрерывной. Также дополнительно приведён график функции, демонстрирующий её непрерывность (см. рисунок №1).

Далее рассмотрим необходимое условие интегрируемости.

«Пусть $f \in R[a,b]$, тогда f ограничена на $[a,b]$»

Зная, что функция, ограниченная на этом отрезке, является непрерывной на этом отрезке и то, что данная нам функция является непрерывной, то необходимое условие интегрируемости является доказанным.

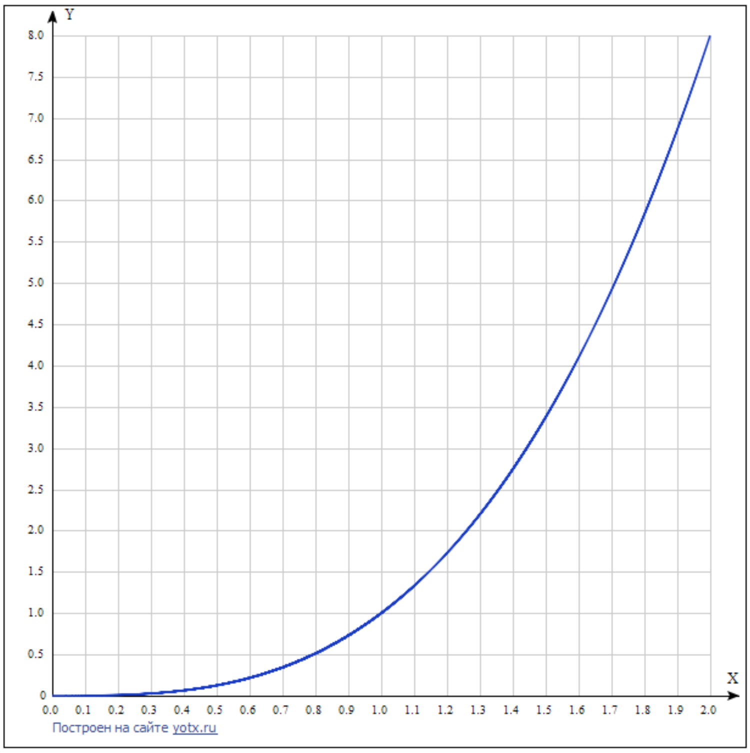

Рисунок №1 – графи функции $f(x) = x^3$ на отрезке [0, 2]

Далее получим интегральную сумму и найдем её предел, сравнив это значение со значением, найденным по формуле Ньютона-Лейбница.

Разобьем отрезок [0, 2] на n равных частей точками:

$$x_0 = 0, x_1 = \frac{2}{n}, x_2 = \frac{4}{n}, \ldots, x_n = \frac{2n}{n} = 2.$$

Получим n частичных сегментов:

$$\left[0, \frac{2}{n}\right], \left[\frac{2}{n}, \frac{4}{n}\right], \ldots, \left[\frac{2n-2}{n}, \frac{2n}{n}\right].$$

Длина каждого частичного сегмента равна $\frac{2}{n}$. В качестве точек ξ_i возьмём самые правые точки частичных сегментов. Далее вычислим интегральную сумму нашей функции в этих точках:

$$\sigma_\tau(f, \xi) = \sum_{i=1}^{n} f(\xi_i)\Delta x_i = \left(\frac{2}{n}\right)^3 * \frac{2}{n} + \left(\frac{4}{n}\right)^3 * \frac{2}{n} + \left(\frac{6}{n}\right)^3 * \frac{2}{n} + \cdots + \left(\frac{2\text{n}}{n}\right)^3 * \frac{2}{n}$$
$$= \frac{2}{n^4} * (2^3 + 4^3 + 6^3 + \cdots + (2n)^3) = \frac{4n^2(n+1)^2}{n^4} = \frac{4(n+1)^2}{n^2};$$

Далее найдём предел нашей интегральной суммы:

$$\lim_{n \to +inf} \sum_{i=1}^{n} f(\xi_i)\Delta x_i = \lim_{n \to +inf} \frac{4(n+1)^2}{n^2} = \lim_{n \to +inf} \frac{4n^2 + 8n + 4}{n^2} = \lim_{n \to +inf} \frac{4 + \frac{8}{n} + \frac{4}{n^2}}{1} = 4;$$

Далее распишем данный нам интеграл через формулу Ньютона-Лейбница:

$$\int_0^2 x^3 dx = \frac{x^4}{4}\Big|_0^2 = \frac{2^4}{4} - \frac{0^4}{4} = 4;$$

Таким образом был получен одинаковый результат при нахождении интеграла через предел интегральной суммы и через формулу Ньютона-Лейбница.

Практическая часть

Ниже в листинге приведена программа. Разберём её основные методы.

На вход нам поступает число разбиений и тип оснащений. Далее мы переходим в цикл, где левой и правой границе цикла соответствуют значения, заданные нам в техническом задании в отрезке данной нам функции (в моём случае это 0 и 2 соответственно). Один шаг в цикле соответствует значению, равному длине сегмента, который мы получили при делении нашего отрезок на n разбиений (длины сегментов равны $\frac{b-a}{n}$).

После этого мы переходим в метод `getEps(int type, double a, double b, double nowI, double n, int counter)`, в котором мы получаем значение нашего оснащения в зависимости от того, какой мы запросили способ выбора оснащений в начале работы программы.

После получения нашего оснащения мы отправляемся в метод `f(double x)`, который возвращает значение функции при данном нам оснащении и складываем с предыдущими значениями оснащения.

Стоит отметить, что при увеличении числа разбиений интегральная сумма стремится к 4 (см. график №1). Также приведены графики с количеством разбиений равным 4 и 8.

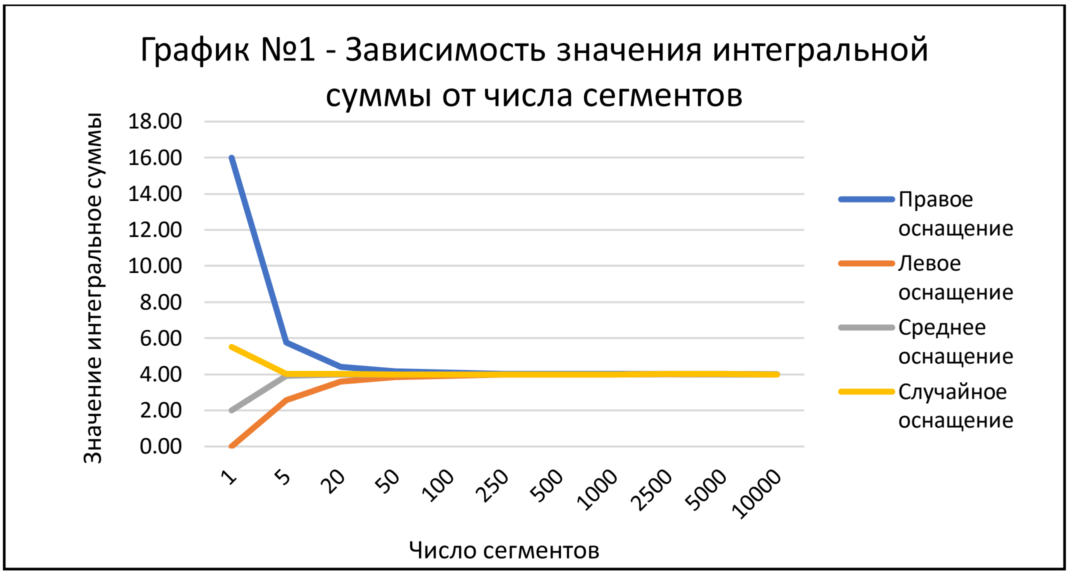

График №1 - Зависимость значения интегральной суммы от числа оснащений

При разбиении на 4 равных сегмента (см. Рисунок №2) интегральная сумма при левом оснащении составляет 2.25, при правом оснащении 6.25, при среднем оснащении 5.36 и при случайном оснащении 3.7.

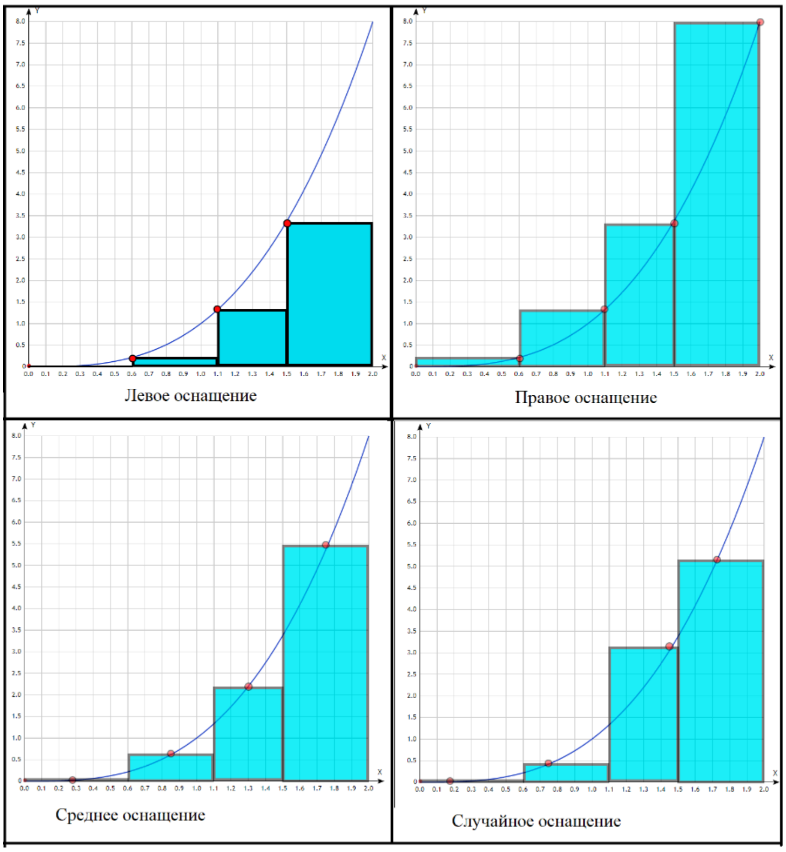

Рисунок №2 – графики оснащений при n = 4.

При разбиении на 8 равных сегмента (см. Рисунок №3) интегральная сумма при левом оснащении составляет 3.0625, при правом оснащении 5.063, при среднем оснащении 3.97 и при случайном оснащении 4.11.

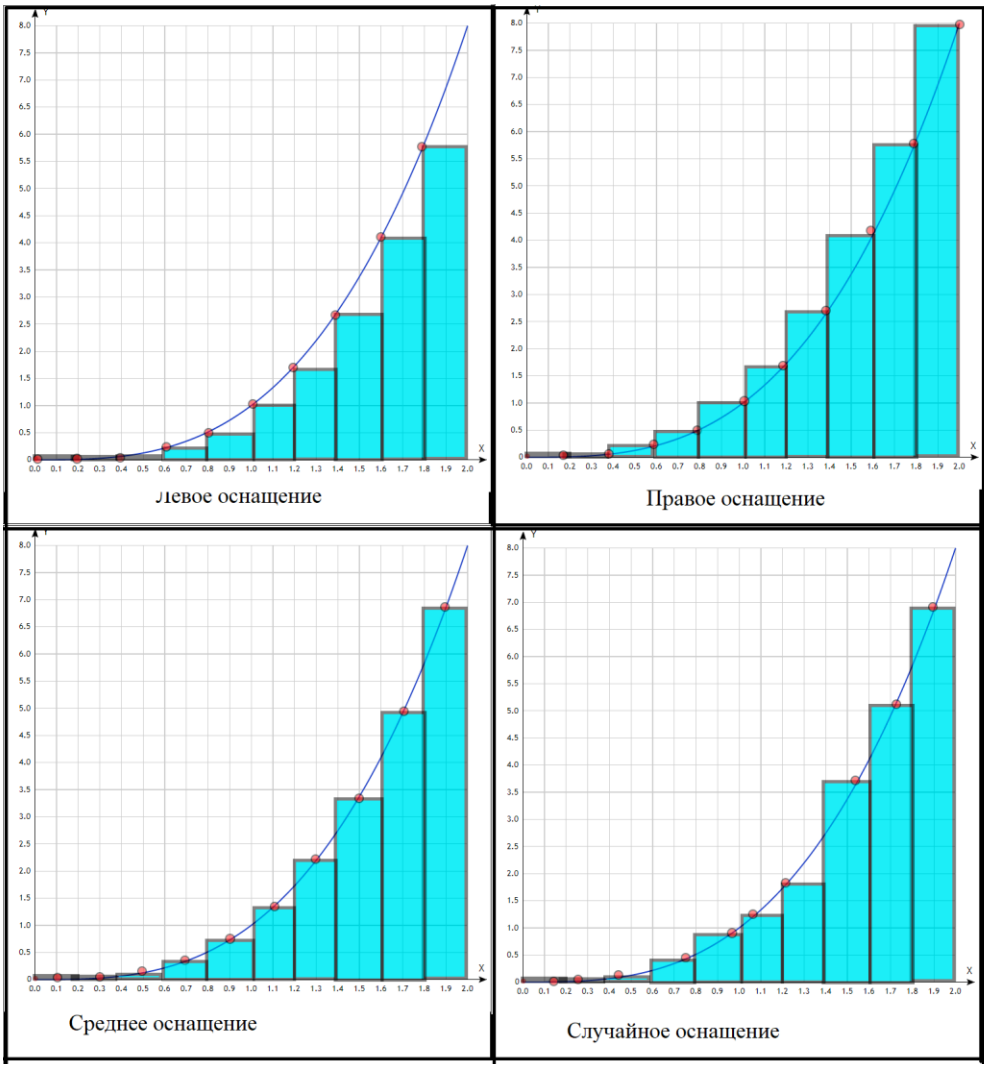

Рисунок №3 – графики оснащений при n = 8.

Однако при числе разбиений, равном 100, интеграл при левом оснащении уже равен 3.92, при правом оснащении 4.08, при среднем оснащении 3.99 и при случайном оснащении 4.01. Таким образом при увеличении числа оснащений мы видим значительное улучшение результата (приближение его к 4).

Листинг

Main.java; Java11;

```java
package com.company;

import java.util.Scanner;

public class Main {

    public static double f(double x) {
        return Math.pow(x, 3);
    }

    public static void printing() {
        System.out.println("Введите способ выбора оснащений (Ввод без скобок):");
        System.out.println("[0] - Левые оснащения");
        System.out.println("[1] - Правые оснащения");
        System.out.println("[2] - Средние оснащения");
        System.out.println("[3] - Случайные оснащения");
    }

    public static double getEps(int type, double a, double b, double nowI, double n, int counter) {
        switch (type) {
            case 0:
                return nowI;
            case 1:
                return (nowI + (b - a) / n);
            case 2:
                return (2 * nowI + (b - a) / n) / 2;
            case 3:
                return counter * ((b - a) / n) + Math.random() * ((b - a) / n);
        }
        return 0;
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.println("Введите число разбиений");
        double n = sc.nextDouble(), a = 0, b = 2;

        int type, counter = 0;
        double endSum = 0;

        OUT:
        while (true) {
            printing();
            type = sc.nextInt();
            switch (type) {
                case 0:
                case 1:
                case 2:
                case 3:
                    break OUT;
                default:
                    System.out.println("Введено неверное число");
            }
        }

        for (double i = a; i < b; i += (b - a) / n) {

            double eps = getEps(type, a, b, i, n, counter++);

            endSum += f(eps) * ((b - a) / n);
        }

        System.out.println(endSum);
    }
}
```