Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное

Образовательное учреждение высшего образования

ПСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Институт инженерных наук

Кафедра информационно-коммуникационных технологий

ЛАБОРАТОРНАЯ РАБОТА №3

ВЫЧИСЛЕНИЕ ОПРЕДЕЛЕННЫХ ИНТЕГРАЛОВ С ЗАДАННЫМ ШАГОМ ИНТЕГРИРОВАНИЯ

Вариант 7

Выполнили: студенты
Антонова П.С., Разгонова Е.В.

Группа: 0432-04

Проверил: Трофимов В.М.

Псков

2021

Постановка задачи

Целью лабораторной работы является написание программы для вычисления интеграла вида $\int_0^2 x * shh(x)dx$ методами центральных прямоугольников, трапеций и Симпсона при одинаковом заданном количестве отрезков разбиения – 48.

Теоретическая справка

Численные методы нахождения значений определенного интеграла используются в тех случаях, когда неизвестна первообразная для функции $f(x)$ или ее вычисление трудновыполнимо. Большая часть методов основана на замене основной подынтегральной функции на функцию упрощённого вида и дальнейшее интегрирование отрезков разбиения этой функции.

— **Метод прямоугольников**

Этот метод основан на приближении кусочно-постоянными функциями. Рабочая формула метода

$$I = h\sum_{i=0}^{n-1} f(x_i + \Delta),$$

$$h = \frac{b-a}{n}, \qquad x_i = a + i * h,$$

где n – число отрезков разбиения. В зависимости от значения погрешности метода, будет меняться подход к расчётам, так:

— если $\Delta = 0$ – получаем рабочую формулу левых прямоугольников;

— если $\Delta = h/2$ – формула центральных прямоугольников;

— если $\Delta = h$ – формула правых прямоугольников.

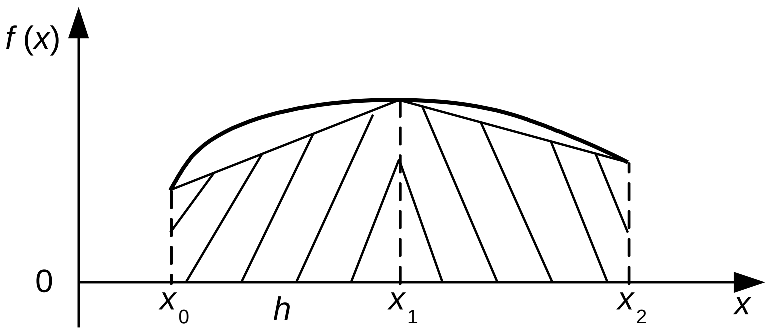
— **Метод трапеций**

В этом случае приближение происходит по кусочно-линейной функции. Рабочая формула метода:

$$I = \frac{h}{2}\left[f(x_0) + f(x_n) + 2\sum_{i=i}^{n-1} f(x_i)\right]$$

$$h = \frac{b-a}{n}, \qquad x_i = a + i * h,$$

где n – число отрезков разбиения.



— **Метод Симпсона**

При использовании данного метода подынтегральная функция приближается с помощью отрезков парабол. Рабочая формула метода:

$$I = \frac{h}{3}\left[f(x_0) + f(x_n) + 2\sum_{i=2(2)}^{n-2} f(x_i) + 4\sum_{i=1(2)}^{n-1} f(x_i)\right]$$

$$h = \frac{b-a}{n}, \qquad x_i = a + i * h$$

Здесь n – чётное число отрезков разбиения (обязательное условие).

Формулу также можно привести к следующему виду:

$$I = \frac{h}{3}\left\{f(x_0) + f(x_n) + \sum_{i=1}^{n-1}\left[3 - (-1)^i\right]f(x_i)\right\}$$

Разработка программного решения

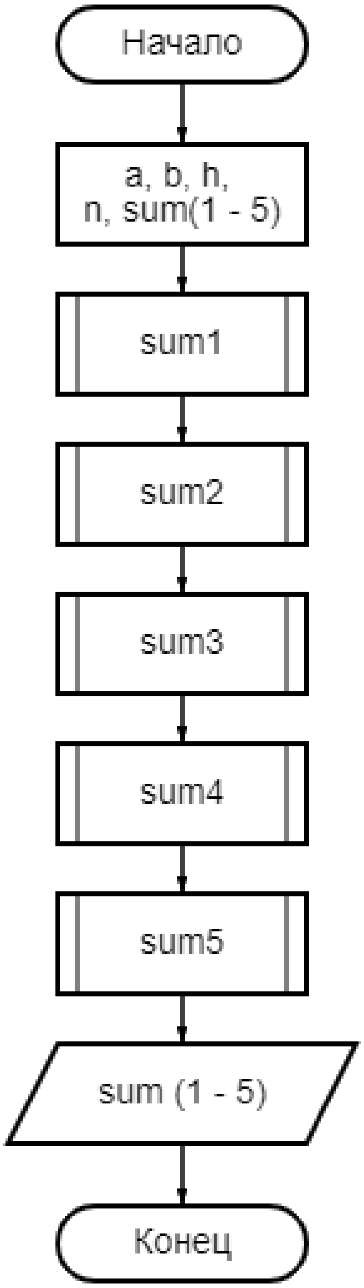


Рис. 1. Общая блок-схема программы

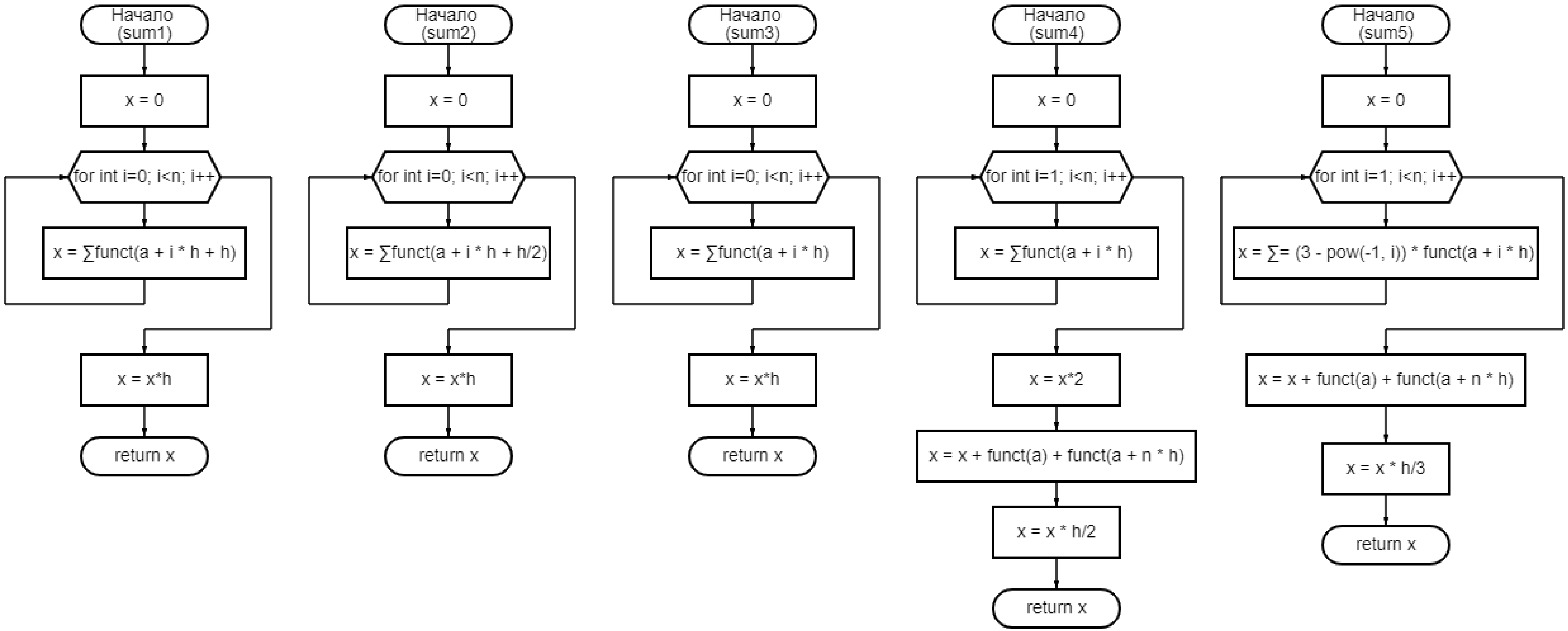


Рис. 2. Блок-схемы методов

```
// Вычисление определенных интегралов с заданным шагом интегрирования

#include<iostream>
#include<cmath>

using namespace std;

double funct(double x)
{
	return x * sinh(x);
}

double pr(double n, double a, double h)
{
    double x = 0;

	for (int i = 0; i < n; i++)
	{
		x += funct(a + i * h + h);
	}
	x *= h;

	return x;
}

double cn(double n, double a, double h)
{
	double x = 0;
	for (int i = 0; i < n; i++)
	{
		x += funct(a + i * h + h/2);
	}
	x *= h;

	return x;
}

double lp(double n, double a, double h)
{
	double x = 0;
	for (int i = 0; i < n; i++)
	{
		x += funct(a + i * h);
	}
	x *= h;

	return x;
}

double trp(double n, double a, double h)
{
	double x = 0;
	for (int i = 1; i < n; i++)
	{
		x += funct(a + i * h);
	}
```

```
        x *= 2;
        x = x + funct(a) + funct(a + n * h);
        x *= (h / 2);

        return x;
}

double smps(double n, double a, double h)
{
        double x = 0;
        for (int i = 1; i < n; i++)
        {
                x += (3 - pow(-1, i)) * funct(a + n * h);
        }
        x = x + funct(a + i * h);
        x *= (h / 3);

        return x;
}

int main()
{
        setlocale(0, "");
        cout.precision(6);

        int n = 48;
        double a, b, h, sum1, sum2, sum3, sum4, sum5;

        a = 0;
        b = 2;
        h = (b - a) / n;

        sum1 = pr(n, a, h);
        sum2 = cn(n, a, h);
        sum3 = lp(n, a, h);
        sum4 = trp(n, a, h);
        sum5 = smps(n, a, h);

        cout << "Формула правых прямоугольников - [ " << sum1 << " ]" << endl;
        cout << "Формула центральных прямоугольников - [ " << sum2 << " ]" <<
endl;
        cout << "Формула левых прямоугольников - [ " << sum3 << " ]" << endl;
        cout << "Метод трапеций - [ " << sum4 << " ]" << endl;
        cout << "Метод Симпсона - [ " << sum5 << " ]" << endl;

        return 0;
}
```

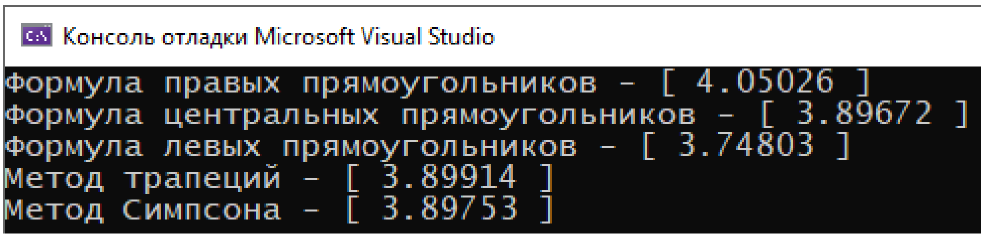


Рис. 2. Результат работы программы

Метод нахождения		Значение интеграла
Метод прямоугольников	Формула левых прямоугольников	4.05026
	Формула центральных прямоугольников	3.89672
	Формула правых треугольников	3.74803
Метод трапеций		3.89914
Метод Симпсона		3.89753

$$\int_0^2 x \cdot \sinh(x)\,dx = 3.898$$

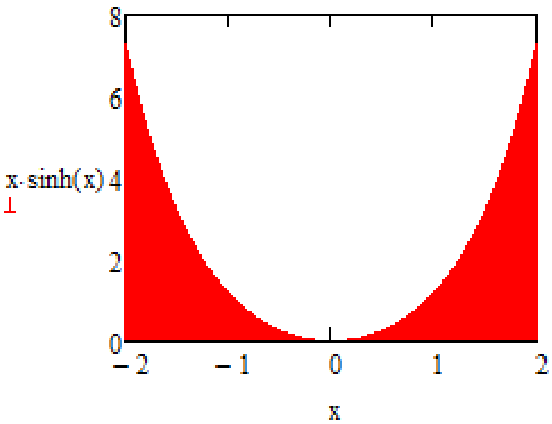


Рис. 3. Проверка решения в Mathcad

Вывод: каждый из методов максимально близко вычисляет значение заданной интегральной функции, но в ходе лабораторной работы было замечено, что наиболее точным к действительному является значение, вычисленное по методу Симпсона, наименее близким по величине – метод правых прямоугольников.