Отчет по практической работе №6

Вайкус Яков Михайлович

ИC-32-1

**Задание.** Вычислить интеграл от заданной функции *f(x)* на отрезке *[а; b]* при делении отрезка на 10 равных частей четырьмя способами:

1) по формуле прямоугольников;

1. по формуле трапеций;
2. по формуле Симпсона;
3. по формуле Гаусса

В расчетах оставлять не менее 6 значащих цифр после запятой.

Произвести оценку погрешности методов интегрирования и сравнить точность полученных результатов.

Код программы:

using static System.Math;

internal class Program

{

    static void Main(string[] args)

    {

        double a = 0; // Левая граница

        double b = 1; // Правая граница

        double n = 10; // Количество разбиений

        double h = (b - a) / n;

        // Эталонное значение интеграла (вычисляется методом Симпсона с высокой точностью)

        double exactValue = 0.603792;

        // Результаты методов

        double rectResult = Rectangle(a, b, h, n);

        double trapezoidResult = Trapezoid(a, b, h, n);

        double simpsonResult = Simpson(a, b, h, n);

        double gaussResult = Gauss(a, b, h, n);

        // Вычисление погрешностей

        double rectError = Abs(rectResult - exactValue);

        double trapezoidError = Abs(trapezoidResult - exactValue);

        double simpsonError = Abs(simpsonResult - exactValue);

        double gaussError = Abs(gaussResult - exactValue);

        string rectErrorFormatted = rectError.ToString("F10");

        string trapezoidErrorFormatted = trapezoidError.ToString("F10");

        string simpsonErrorFormatted = simpsonError.ToString("F10");

        string gaussErrorFormatted = gaussError.ToString("F10");

        // Вывод результатов

        Console.WriteLine("Метод прямоугольников:");

        Console.WriteLine($"Результат: {rectResult}, Погрешность: {rectErrorFormatted}");

        Console.WriteLine("Метод трапеций:");

        Console.WriteLine($"Результат: {trapezoidResult}, Погрешность: {trapezoidErrorFormatted}");

        Console.WriteLine("Метод Симпсона:");

        Console.WriteLine($"Результат: {simpsonResult}, Погрешность: {simpsonErrorFormatted}");

        Console.WriteLine("Метод Гаусса:");

        Console.WriteLine($"Результат: {gaussResult}, Погрешность: {gaussErrorFormatted}");

        // Сравнение методов

        Console.WriteLine("\nНаиболее точный метод:");

        double minError = Min(Min(rectError, trapezoidError), Min(simpsonError, gaussError));

        if (minError == rectError) Console.WriteLine("Метод прямоугольников");

        if (minError == trapezoidError) Console.WriteLine("Метод трапеций");

        if (minError == simpsonError) Console.WriteLine("Метод Симпсона");

        if (minError == gaussError) Console.WriteLine("Метод Гаусса");

    }

    static double f(double x)

    {

        return 0.37 \* Math.Pow(Math.E, Math.Sin(x));

    }

    static double Rectangle(double a, double b, double h, double n)

    {

        double I = 0;

        for (int i = 0; i <= n - 1; i++)

        {

            I += f(a + h \* i + h / 2);

        }

        return I \* h;

    }

    static double Trapezoid(double a, double b, double h, double n)

    {

        double sum = 0;

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

        {

            sum += f(a + i \* h);

        }

        return h \* (((f(a) + f(b)) / 2) + sum);

    }

    static double Simpson(double a, double b, double h, double n)

    {

        double sum1 = 0, sum2 = 0;

        for (int i = 1; i <= n - 1; i += 2)

        {

            sum1 += f(a + i \* h);

        }

        for (int i = 2; i <= n - 2; i += 2)

        {

            sum2 += f(a + h \* i);

        }

        return (h / 3) \* (f(a) + 4 \* sum1 + 2 \* sum2 + f(b));

    }

    static double Gauss(double a, double b, double h, double n)

    {

        double sum = 0;

        for (int i = 0; i <= n - 1; i++)

        {

            sum += f(a + h \* i + h / 2 - h / (2 \* Pow(3.0, 0.5))) +

                   f(a + h \* i + h / 2 + h / (2 \* Pow(3.0, 0.5)));

        }

        return h / 2 \* sum;

    }

}

Вывод:  
