

Министерство образования и науки РФ  
Федеральное государственное автономное образовательное учреждение высшего  
образования  
«Омский государственный технический университет»

Факультет (институт) Информационных технологий и компьютерных систем

Кафедра Прикладная математика и фундаментальная информатика

### Расчетно-графическая работа

по дисциплине Алгоритмизация и программирование

на тему Вычисление определенного интеграла методом адаптивных квадратур

Пояснительная записка

Шифр проекта 020-РГР-02.03.02-№ 20-ПЗ

Студента Остапченко Евгения Антоновича

фамилия, имя, отчество полностью

Курс 1

Группа ФИТ-221

Направление (специальность) 02.03.02

Фундаментальная информатика и информационные технологии

код, наименование

Руководитель ст. преподаватель

ученая степень, звание

Федотова И.В.

фамилия, инициалы

Выполнил \_\_\_\_\_

дата, подпись студента

**Работа защищена с количеством баллов**

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

\_\_\_\_\_  
дата, подпись руководителя

Омск 2023

## Содержание

|  |   |
|--|---|
| Задание.....                             | 3 |
| Математическая формулировка задачи ..... | 4 |
| Текст программы на языке C#.....         | 5 |
| Список литературы .....                  | 8 |

## Задание

Реализовать алгоритм вычисления определенного интеграла методом адаптивных квадратур.

Для вычисления выберем произвольный интеграл. Например:

$$\int_a^b (1 + x + \sin(2x))dx$$

## **Математическая формулировка задачи**

Адаптивная квадратура – разновидность численной интеграции, которая обнаруживает области, в которых аппроксимация имеет наибольшую погрешность, и совершенствует используемый для работы метод. Программа работает на основе формулы трапеций с адаптивной квадратурой. На графике интегрирования при использовании метода трапеций там, где кривая приближается к прямой линии, трапеции подходят к ней очень близко, но там, где она резко изгибается, трапеции отстоят от нее. Программа, использующая адаптивную квадратуру, ищет области, где фигуры не примыкают вплотную к кривой, и увеличивает их количество на этих участках.

Суть программы заключается в следующем: часть аппроксимирующей области вначале заполняется одной трапецией, а затем разделяется на две меньшие фигуры. Если разница между областью, занимаемой большой трапецией, и суммой областей меньших трапеций превышает определенный процент, программа выбирает разбиение и переходит к аналогичному расчету каждой отдельной области.

## Текст программы на языке C#

```
static float AdaptiveMetod( float xmin, float xmax, Int32 num_intervals,
float max_slice_error)
{
    float dx=(xmax-xmin)/num_intervals;
    //Console.WriteLine(dx);
    //double total =0;
    float total_area=0;
    float x=xmin;
    for (int i=1;i<num_intervals;i++)
    {
        total_area = total_area + SliceArea(x,x+dx,max_slice_error);
        x=x+dx;
        //Console.WriteLine("chek", total_area);
    }
    return total_area;
}

static float SliceArea( float x1, float x2, float max_slice_error)
{
    float y1= Function(x1);
    float y2= Function(x2);
    float xm= (x1+x2)/2;
    //Console.WriteLine(xm);
    float ym= Function(xm);

    float area12 = (x2 - x1) * (y1 + y2) / 2;
    float area1m = (xm - x1) * (y1 + ym) / 2;
    float aream2 = (x2 - xm) * (ym + y2) / 2;
    float area1m2 = area1m + aream2;

    float error = (area1m2-area12)/area12;

    if (Math.Abs(error)<max_slice_error)
    {
        //Console.WriteLine(area1m2);
        return area1m2;
    }
    return SliceArea( x1, xm, max_slice_error) + SliceArea( xm, x2,
max_slice_error);
}

static float Function(float x)
{
    float sin_x=(float)(Math.Sin(2*x));
    float y=1+x+sin_x;
    return y;
}
```

```

static void Avtor()
{
    Console.WriteLine("Выполнил: Остапченко Евгений");
    Console.WriteLine("Курс: 1");
    Console.WriteLine("Группа: ФИТ-221");
}

static void Menu()
{
    Console.WriteLine("\nМеню РГР:");
    Console.WriteLine("1. Вычисление");
    Console.WriteLine("2. Об авторе");
    Console.WriteLine("3. Выход\n");
    Console.Write("Введите действие: ");
    int nom=Convert.ToInt32(Console.ReadLine());
    switch (nom)
    {
        case 1:
            Console.WriteLine("Подинтегральное выражение:
1+x+sin(2x)\nВведите интервал [a,b]:");
            Console.Write("Введите a = ");
            float a=(float)Convert.ToDouble(Console.ReadLine());
            Console.Write("Введите b = ");
            float b=(float)Convert.ToDouble(Console.ReadLine());
            int interval=10000; //число интервалов
            Console.WriteLine("Интеграл равен:
"+AdaptiveMetod(a,b,interval,0.01f));
            Menu();
            break;
        case 2:
            Avtor();
            Menu();
            break;
        case 3:
            Console.WriteLine("Досвидание!");
            break;
        default:
            Console.WriteLine("Ошибка!");
            Menu();
            break;
    }
}

Menu();

```

## Разработка интерфейса пользователя

На рисунке 2 представлено основное меню программы. Управление осуществляется при помощи ввода номера пункта меню с клавиатуры и нажатия клавиши Enter.

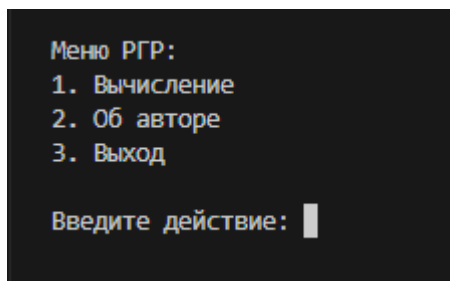


Рисунок 2 – Основное меню программы

На рисунке 3 представлена функция вычисления интеграла с использованием адаптивных квадратур.

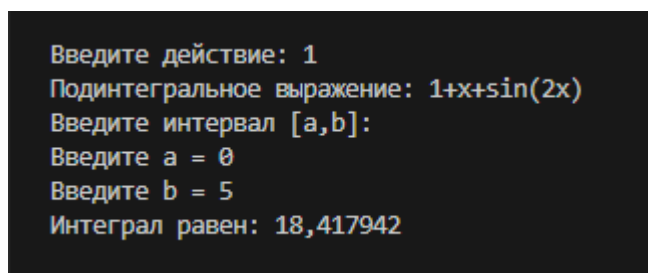


Рисунок 3 – Функция вычисления интеграла

На рисунке 4 представлена подпрограмма «Об авторе».

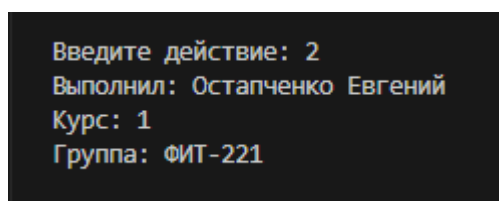


Рисунок 4 – Подпрограмма «Об авторе».

## Список литературы

- 1) Программирование на С# для начинающих. Основные сведения  
- Алексей Васильев 2018 г.
- 2) Алгоритмы. Теория и практическое применение / Род Стивенс –  
Москва: Издательство «Э», 2016. – 544 с.
- 3) [https://en.wikipedia.org/wiki/Adaptive\\_Simpson%27s\\_method](https://en.wikipedia.org/wiki/Adaptive_Simpson%27s_method) (Дата  
обращения 15.05.2023)