

Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Інститут прикладної математики та фундаментальних наук

Кафедра прикладної математики

Звіт
до лабораторної роботи №3
Варіант - 7

Виконала:
студентка групи
ПМ-31
Яворська Олена

Прийняв:
Строчик М. М.

Львів 2020

Загальні положення

Використовуючи об'єкти типу *DataGridView* та *Chart*, побудувати:

- А) графік явно заданої функції;
- В) графік функції, заданої параметрично.

На формі необхідно розташувати компоненти *DataGridView* та *Chart*, поля для введення меж зміни аргументу, поле для введення кількості точок табулювання, поля для відображення найбільшого та найменшого значень функції на заданому інтервалі. У таблицю *DataGridView* необхідно вивести значення аргументу (аргументів) і відповідні табульовані значення функції. На графіках необхідно відобразити осі координат з позначками значень та відповідними підписами. Елементи керування, розташовані на формі, повинні коректно відображатися при зміні розмірів форми. Передбачити можливість як зчитування даних з файлу для побудови графіків, так і збереження табульованих значень та графіків у відповідні файли. Додатковими опціями можуть бути сітка з можливістю її відображення/приховування; побудова графіка для діапазону значень, виділених користувачем у таблиці *DataGridView*; масштабування графіка.

Завдання:

- А) Графік явно заданої функції:

$$y = \exp(-2x) \sin^2 x$$

- В) Для параметрично заданої функції побудувати графіки функцій $x(t)$, $y(t)$, $y(x)$, $x(y)$ та $x(y)$:

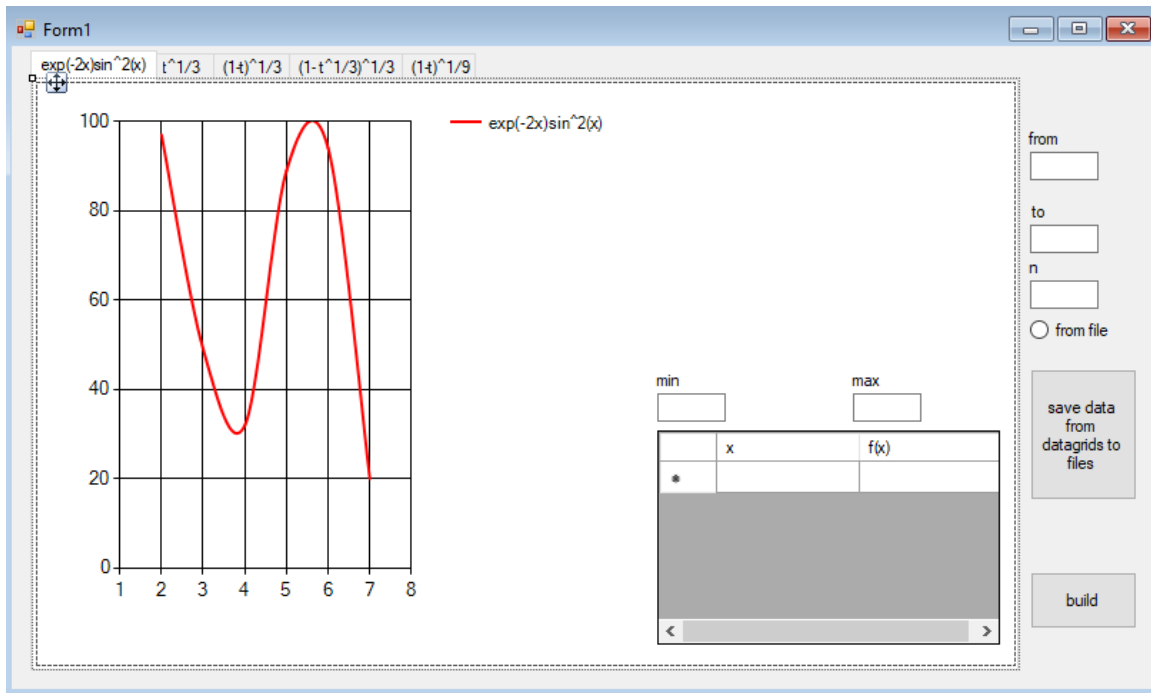
$$x(t) = t^{1/3},$$

$$y(t) = (1-t)^{1/3}$$

Хід роботи:

- 1) Створила макет програми, познайомилася з елементом форми *TabControl*, оскільки мені здалося логічним створити елементи *DataGridView* та *Chart* для кожного мого графіка в окремій вкладці (*TabPage*). Через це мені довелося прописувати, наприклад,

збереження кожного DataGridView з даними та графік одної з моїх функцій(точками x та y) у текстовий файл окремо для кожного елемента DataGridView. Кнопки, які спільні для всіх графіків- початок/кінець відрізка та кількість точок розбиття, кнопки «побудувати графіки» та «зберегти дані з DataGridView у текстові файли» я залишила на спільній формі, натомість на вкладках розташувала для кожного графіка елементи DataGridView та Chart, також текстові поля для виводу максимуму і мінімуму функцій(для конкретного графіка, ці дані відрізн. для різних графіків).



2) Передбачила як зчитування даних з файлу для побудови графіків, так і збереження табульованих значень та графіків у відповідні файли.

Збереження графіків:

```
//збереження графіків у файли
private void button2_Click(object sender, EventArgs e)
{
    this.chart1.SaveImage(@"F:\chart1.png", ChartImageFormat.Png);
    this.chart2.SaveImage(@"F:\chart2.png", ChartImageFormat.Png);
    this.chart3.SaveImage(@"F:\chart3.png", ChartImageFormat.Png);
    this.chart4.SaveImage(@"F:\chart4.png", ChartImageFormat.Png);
    this.chart5.SaveImage(@"F:\chart5.png", ChartImageFormat.Png);
}
```

Збереження таблиць значень:

```
private void button1_Click_1(object sender, EventArgs e)

    //записуємо дані з кожного "datagridview в окремий файл
    {
        string docPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
        StreamWriter sw1 = new StreamWriter(Path.Combine(docPath,
"datagridview1.txt"));
        StreamWriter sw2 = new StreamWriter(Path.Combine(docPath,
"datagridview2.txt"));
        StreamWriter sw3 = new StreamWriter(Path.Combine(docPath,
"datagridview3.txt"));
        StreamWriter sw4 = new StreamWriter(Path.Combine(docPath,
"datagridview4.txt"));
        StreamWriter sw5 = new StreamWriter(Path.Combine(docPath,
"datagridview5.txt"));
        string point = ";"; //розділювач
        for (int i = 1; i < dataGridView1.RowCount; i++)
        {
            for (int j = 0; j < dataGridView1.ColumnCount; j++)
            {
                sw1.Write(Convert.ToString(dataGridView1[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView2.RowCount; i++)
        {
            for (int j = 0; j < dataGridView2.ColumnCount; j++)
            {
                sw2.Write(Convert.ToString(dataGridView2[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView3.RowCount; i++)
        {
            for (int j = 0; j < dataGridView3.ColumnCount; j++)
            {
                sw3.Write(Convert.ToString(dataGridView3[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView4.RowCount; i++)
        {
            for (int j = 0; j < dataGridView4.ColumnCount; j++)
            {
                sw4.Write(Convert.ToString(dataGridView4[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView5.RowCount; i++)
        {
            for (int j = 0; j < dataGridView5.ColumnCount; j++)
```

```

        {
            sw5.Write(Convert.ToString(dataGridView5[j, i].Value +
point));
        }

    }
    //закриваємо файли, куди записували
    sw1.Close();
    sw2.Close();
    sw3.Close();
    sw4.Close();
    sw5.Close();
}

```

Зчитування даних про відрізок та к-ість його розбиттів з файла:

```

//можливість отримати дані про відрізок та к-ість проміжків розбиття з файлу
private void radioButton1_CheckedChanged(object sender, EventArgs e)
{
    FileInfo f = new
FileInfo(@"C:\Users\Olena\source\repos\lab3_yav0\lab3_yav0\data.txt");//розташування моєї лр
    using (FileStream fs = f.Open(FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
    {
        using (StreamReader sr = new StreamReader(fs,
Encoding.UTF8))//зчитує файл з даними
        {
            from.Text = sr.ReadLine();//перший рядок з файлу
            to.Text = sr.ReadLine();//другий, де я записала кінець відрізка
            number.Text = sr.ReadLine();//відповідно, третій рядок мого
файлу
        }
    }
}

```

Всі графіки/текстові файли зі значеннями іксів і функцій в них в архіві з програмою.

3) На графіках необхідно відобразила осі координат з позначками значень та відповідними підписами.

```

//підписуємо осі
//в кожного графіка на кожній вкладці підписуємо осі x і f(x)
string ax = "x";
string ay = "f(x)";
chart1.ChartAreas[0].AxisX.Title = ax;
chart1.ChartAreas[0].AxisY.Title = ay;
chart2.ChartAreas[0].AxisX.Title = ax;

```

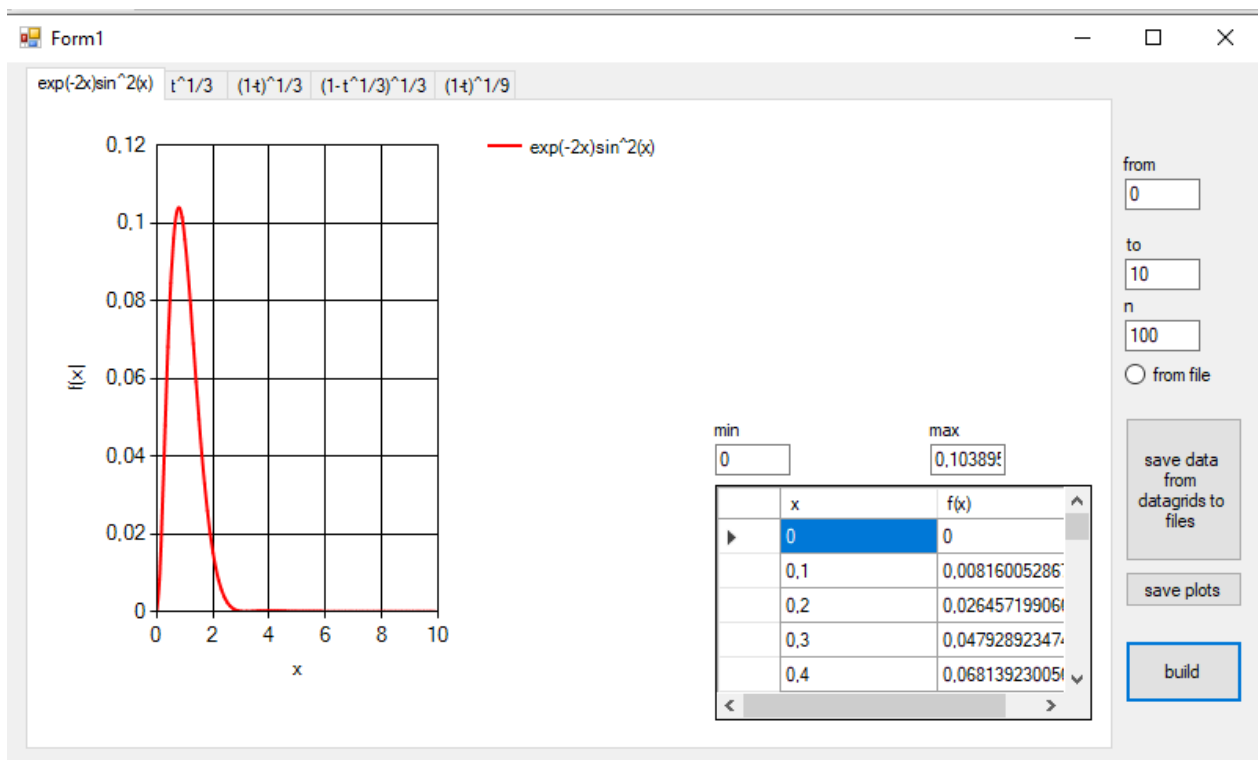
```

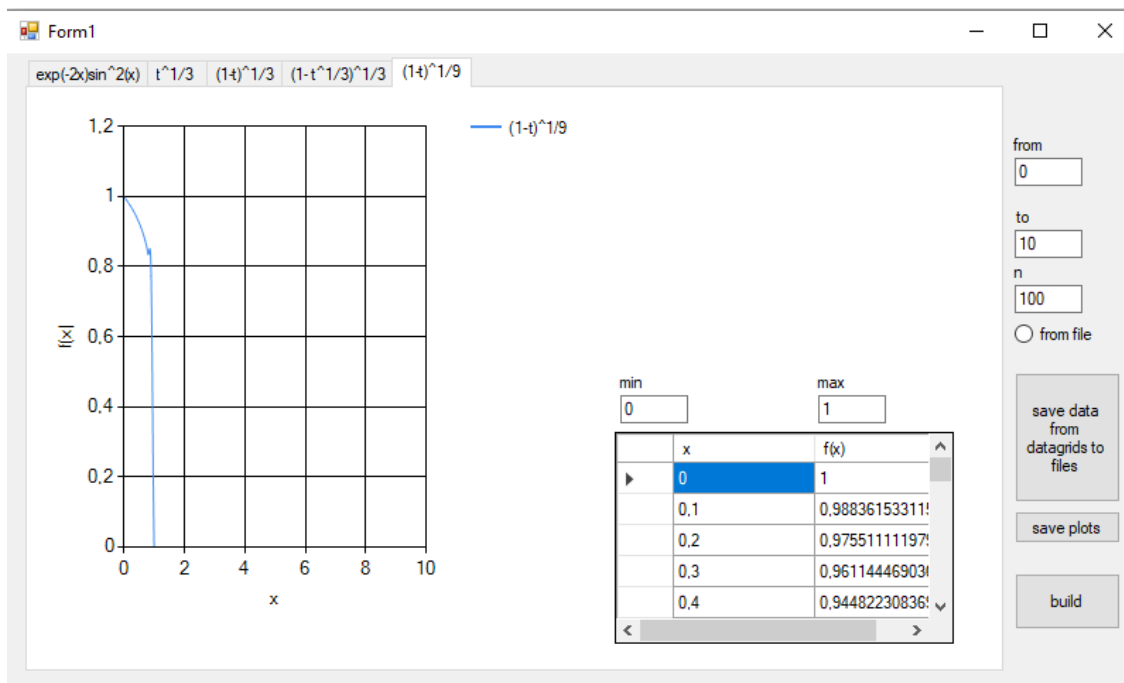
chart2.ChartAreas[0].AxisY.Title = ay;
chart3.ChartAreas[0].AxisX.Title = ax;
chart3.ChartAreas[0].AxisY.Title = ay;
chart4.ChartAreas[0].AxisX.Title = ax;
chart4.ChartAreas[0].AxisY.Title = ay;
chart5.ChartAreas[0].AxisX.Title = ax;
chart5.ChartAreas[0].AxisY.Title = ay;

```

- 4) Передбачила оновлення даних при побудові графіка на новому відрізку DataGridView. Дозволила ввід в текстбокси лише цифр. При обраному RadioButton ці цифри беруться з мого файлу. Встановила дефолтні значення для текстбоксів з даними про відрізок та к-ість точок табулювання.

Результат виконання програми:



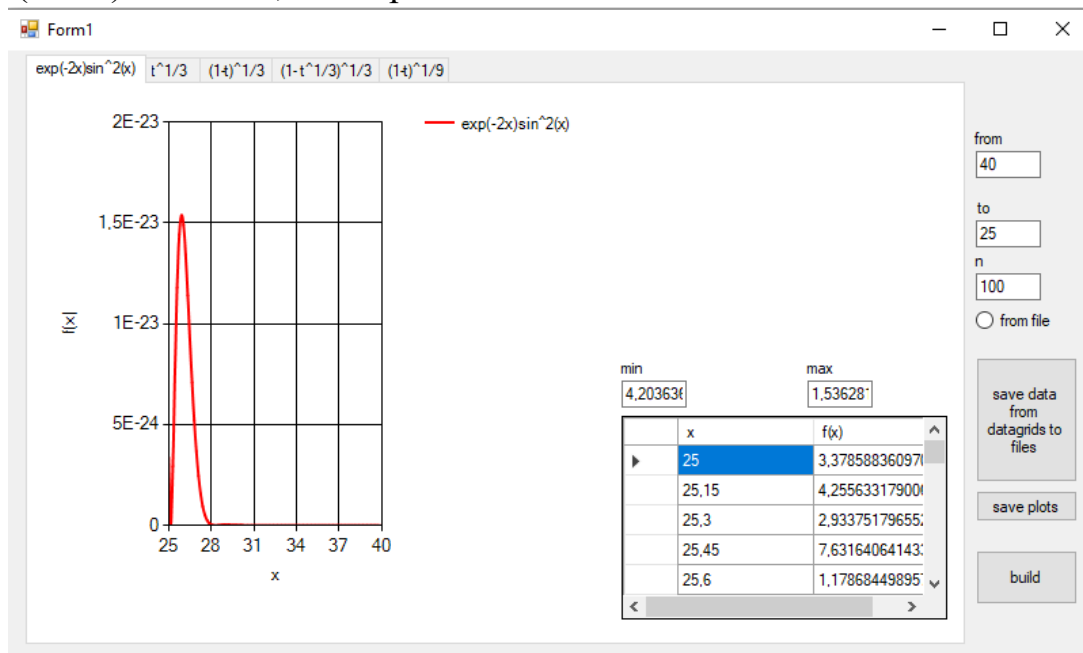


datagridview1: Блокнот

Файл Редагування Формат Вигляд Довідка

```
0,1;0,00816005286773079;0,2;0,0264571990669968;0,3;0,0479289234743883;0,4;0,0681392300563708;0,5;0,0845566654125147;0,6;0,0960270768111626;0,7;0,102341791266289;0,8;0,10136495809646724;4,5;0,000117926105601656;4,6;9,97685091142379E-05;4,7;8,27113691512309E-05;4,8;6,7210200400839E-05;4,9;5,35226121870517E-05;5,4,17468591241283E-05;5,1;-08;8,6;1,82808681041796E-08;8,7;1,21972734241148E-08;8,8;7,77330829833109E-09;8,9;4,66949411253077E-09;9,2,58668482652999E-09;9,1,1,26966626356187E-09;9,2;5,0718028881
```

Тут(вище) показано, як збереглися дані з DataGridView.



Скрін вище :на випадок, якщо початок відрізка число більше за його кінець. Цифри в таблиці можуть здатися недостовірними, але вони типу double, далі є множник $E^{(-23 \text{ або } -26 \dots)}$, тобто це дуже маленькі числа, як на графіку.

Висновок: на цій лабораторній роботі я працювала з новими для мене елементами форми DataGridView та Chart, познайомилася з елементом форми TabControl, який здався мені доцільним для перемикання між графіками 5 рівнянь та їх таблиць значень табульованих функцій в точках з вказаного мною відрізка. Вперше працювала з графіками та зчитуванням з/записом в файл(особливо збереженням зображень) в c#/winforms.

Додаток:

Program.cs:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace lab3_yav0
{
    static class Program
    {
        /// <summary>
        /// The main entry point for the application.
        /// </summary>
        [STAThread]
        static void Main()
        {
            Application.EnableVisualStyles();
            Application.SetCompatibleTextRenderingDefault(false);
            Application.Run(new Form1());
        }
    }
}
```

Form1.cs:

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Windows.Forms.DataVisualization.Charting;
using System.IO; //бібліотека для читання-запису в файл

namespace lab3_yav0
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            //дефолтні значення при загрузці, їх можна змінити ,але лише на цифри, далі буде
            обмеження вводу
        }
    }
}
```

```

        //у відповідні текстбокси
        number.Text = "100";
        from.Text = "0";
        to.Text = "10";
    }

    private void dataGridView1_CellContentClick(object sender,
DataGridViewCellEventArgs e)
    {
    }

    private void chart1_Click(object sender, EventArgs e)
    {
    }

    private void to_TextChanged(object sender, EventArgs e)
    {
    }

    private void button_build_Click(object sender, EventArgs e)
    {
        //dataGridView1.Refresh();
        //очищуємо dataGridView, щоб при повторному натисканні на button_build
будувалася заново таблиця значень
        //якщо робити лише dataGridView1.Refresh(), то нові значення додаватимуться
після старих

        //якщо нічого не робити dataGridView не зміниться при зміні меж і графіків
        dataGridView1.Rows.Clear();
        dataGridView2.Rows.Clear();
        dataGridView3.Rows.Clear();
        dataGridView4.Rows.Clear();
        dataGridView5.Rows.Clear();
        double Xmin = double.Parse(from.Text);

        double Xmax = double.Parse(to.Text);
        //якщо введуть початок відрізка, більший за кінець
        if (Xmin > Xmax)
        {
            double temp;
            temp = Xmin;
            Xmin = Xmax;
            Xmax = temp;
        }

        int count = int.Parse(number.Text); //к-ість відрізків розбиття
        //int count = 0;
        //if (int.TryParse(number.Text, out count)) ;
        double Step; //крок, з яким рахуємо значення ф-ії в точках
        Step = (Math.Abs(Xmax - Xmin)) / count; //модуль, щоб не вибивало
помилки, коли введуть поч відр, більший за кінець

// Массив
значень X - спільний для обох графіків

        double[] x = new double[count];

```

```

        //підписуємо осі
        //в кожного графіка на кожній вкладці підписуємо осі x і f(x)
        string ax = "x";
        string ay = "f(x)";
        chart1.ChartAreas[0].AxisX.Title = ax;
        chart1.ChartAreas[0].AxisY.Title = ay;
        chart2.ChartAreas[0].AxisX.Title = ax;
        chart2.ChartAreas[0].AxisY.Title = ay;
        chart3.ChartAreas[0].AxisX.Title = ax;
        chart3.ChartAreas[0].AxisY.Title = ay;
        chart4.ChartAreas[0].AxisX.Title = ax;
        chart4.ChartAreas[0].AxisY.Title = ay;
        chart5.ChartAreas[0].AxisX.Title = ax;
        chart5.ChartAreas[0].AxisY.Title = ay;
    // Масиви Y – по одному для кожного графіка

    double[] y1 = new double[count];
    double[] y2 = new double[count];
    double[] y3 = new double[count];
    double[] y4 = new double[count];
    double[] y5 = new double[count];
    // рахуєм точки для графіків функцій в циклі
    for (int i = 0; i < count; i++)

        { // вичисляємо X на кожному кроці

            x[i] = Xmin + Step * i;
            // вичисляємо значення функцій в точках X
            y1[i] = Math.Exp(-2*x[i])*Math.Pow(Math.Sin(x[i]), 2); //графік явно
заданої функції
            y2[i] = Math.Pow(x[i], 1.0/3.0); //x(t) для параметрично заданої функції
            y3[i] = Math.Pow((1-x[i]), 1.0 / 3.0); //y(t) для параметрично заданої функції
            y4[i] = Math.Pow((1 - Math.Pow(x[i], 1.0 / 3.0)), 1.0 / 3.0); //x(y)
для параметрично заданої функції
            y5[i] = Math.Pow((1 - x[i]), 1.0 / 9.0); //y(x) для параметрично
заданої функції

            //додаємо виведення іксів і значень функцій в них з кожного графіка
у відповідну dataGridView
            dataGridView1.Rows.Add(x[i], y1[i]);
            dataGridView2.Rows.Add(x[i], y2[i]);
            dataGridView3.Rows.Add(x[i], y3[i]);
            dataGridView4.Rows.Add(x[i], y4[i]);
            dataGridView5.Rows.Add(x[i], y5[i]);
            //dataGridView1.Rows[0].Cells[i].Value = x[i];
        } // Настроюєм осі графіка, для кожного графіка на кожній вкладці додаємо підписи
осей :x і f(x)

        chart1.ChartAreas[0].AxisX.Minimum = Xmin;
        chart1.ChartAreas[0].AxisX.Maximum = Xmax;
        chart2.ChartAreas[0].AxisX.Minimum = Xmin;
        chart2.ChartAreas[0].AxisX.Maximum = Xmax;
        chart3.ChartAreas[0].AxisX.Minimum = Xmin;
        chart3.ChartAreas[0].AxisX.Maximum = Xmax;
        chart4.ChartAreas[0].AxisX.Minimum = Xmin;
        chart4.ChartAreas[0].AxisX.Maximum = Xmax;
        chart5.ChartAreas[0].AxisX.Minimum = Xmin;
        chart5.ChartAreas[0].AxisX.Maximum = Xmax;

```

```

        //визначаємо крок сітки ,закоментовано-необов'язковщ
        //при малих відрізках і великому числі розбиттів сітка незручна
        //chart1.ChartAreas[0].AxisX.MajorGrid.Interval = Step;
        // Додаємо вичислені значення в графіки
        chart1.Series[0].Points.DataBindXY(x, y1);
        chart2.Series[0].Points.DataBindXY(x, y2);
        chart3.Series[0].Points.DataBindXY(x, y3);
        chart4.Series[0].Points.DataBindXY(x, y4);
        chart5.Series[0].Points.DataBindXY(x, y5);
    }
    //заповняємо min i max для графіка 1
    double y1_min = func_min(y1, count);
    min.Text = y1_min.ToString();
    double y1_max = func_max(y1, count);
    max.Text = y1_max.ToString();
    //заповняємо min i max для графіка 2
    double y2_min = func_min(y2, count);
    min_2.Text = y2_min.ToString();
    double y2_max = func_max(y2, count);
    max_2.Text = y2_max.ToString();
    //заповняємо min i max для графіка 3
    double y3_min = func_min(y3, count);
    min_3.Text = y3_min.ToString();
    double y3_max = func_max(y3, count);
    max_3.Text = y3_max.ToString();
    //заповняємо min i max для графіка 4
    double y4_min = func_min(y4, count);
    min_4.Text = y4_min.ToString();
    double y4_max = func_max(y4, count);
    max_4.Text = y4_max.ToString();
    //заповняємо min i max для графіка 5
    double y5_min = func_min(y5, count);
    min_5.Text = y5_min.ToString();
    double y5_max = func_max(y5, count);
    max_5.Text = y5_max.ToString();
}

//функція для визнач min f(x)
private double func_min(double[] y, double count)
{
    double y_min = y[0];

    for (int i = 0; i < count-1; i++)
    {
        if (y[i] < y_min)
        {
            y_min = y[i];
        }
    }
    return y_min;
}

//функція для визнач максимуму f(x)
private double func_max(double[] y, double count)
{
    double y_max = y[0];

```

```

        for (int i = 0; i < count - 1; i++)
        {
            if (y[i] > y_max)
            {
                y_max = y[i];
            }
        }
        return y_max;
    }

    private void from_TextChanged(object sender, EventArgs e)
    { }

    private void tabPage1_Click(object sender, EventArgs e)
    {
    }

    private void chart3_Click(object sender, EventArgs e)
    {
    }

    private void tabPage1_Click_1(object sender, EventArgs e)
    {
    }

    // всі KeyPress призн, щоб не дати ввести в текстбокси щось, крім цифр
    private void from_KeyPress(object sender, KeyPressEventArgs e)
    {
        char number = e.KeyChar;
        if (!Char.IsDigit(number) && number != 8 && number != 44) // цифри, клавіша
BackSpace и кома
        {
            e.Handled = true;
        }
    }

    private void to_KeyPress(object sender, KeyPressEventArgs e)
    {
        char number = e.KeyChar;
        if (!Char.IsDigit(number) && number != 8 && number != 44) // цифри, клавіша
BackSpace и кома
        {
            e.Handled = true;
        }
    }

    private void number_KeyPress(object sender, KeyPressEventArgs e)
    {
        char number = e.KeyChar;
        if (!Char.IsDigit(number) && number != 8 && number != 44) // цифри, клавіша
BackSpace и кома
    }

```

```

        {
            e.Handled = true;
        }
    }

    private void textBox2_TextChanged(object sender, EventArgs e)
    {

    }

    //кнопка save data to file, збереження данних з dataGridView1 в файл
    private void button1_Click(object sender, EventArgs e)
    {

    }

    private void button1_Click_1(object sender, EventArgs e)
    //записуємо дані з кожного "datagridview в окремий файл
    {
        string docPath =
Environment.GetFolderPath(Environment.SpecialFolder.MyDocuments);
        StreamWriter sw1 = new StreamWriter(Path.Combine(docPath,
"datagridview1.txt"));
        StreamWriter sw2 = new StreamWriter(Path.Combine(docPath,
"datagridview2.txt"));
        StreamWriter sw3 = new StreamWriter(Path.Combine(docPath,
"datagridview3.txt"));
        StreamWriter sw4 = new StreamWriter(Path.Combine(docPath,
"datagridview4.txt"));
        StreamWriter sw5 = new StreamWriter(Path.Combine(docPath,
"datagridview5.txt"));
        string point = ";"; //розділювач
        for (int i = 1; i < dataGridView1.RowCount; i++)
        {
            for (int j = 0; j < dataGridView1.ColumnCount; j++)
            {
                sw1.Write(Convert.ToString(dataGridView1[j, i].Value +
point));
            }

        }
        for (int i = 1; i < dataGridView2.RowCount; i++)
        {
            for (int j = 0; j < dataGridView2.ColumnCount; j++)
            {
                sw2.Write(Convert.ToString(dataGridView2[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView3.RowCount; i++)
        {
            for (int j = 0; j < dataGridView3.ColumnCount; j++)
            {
                sw3.Write(Convert.ToString(dataGridView3[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView4.RowCount; i++)
    }

```

```

        {
            for (int j = 0; j < dataGridView4.ColumnCount; j++)
            {
                sw4.Write(Convert.ToString(dataGridView4[j, i].Value +
point));
            }
        }
        for (int i = 1; i < dataGridView5.RowCount; i++)
        {
            for (int j = 0; j < dataGridView5.ColumnCount; j++)
            {
                sw5.Write(Convert.ToString(dataGridView5[j, i].Value +
point));
            }
        }
        //закриваємо файли, куди записували
        sw1.Close();
        sw2.Close();
        sw3.Close();
        sw4.Close();
        sw5.Close();
    }
    //можливість отримати дані про відрізок та к-ість проміжків розбиття з файлу
    private void radioButton1_CheckedChanged(object sender, EventArgs e)
    {
        FileInfo f = new
FileInfo(@"C:\Users\Olena\source\repos\lab3_yav0\lab3_yav0\data.txt");//розташування моєї лр
        using (FileStream fs = f.Open(FileMode.Open, FileAccess.Read,
FileShare.ReadWrite))
        {
            using (StreamReader sr = new StreamReader(fs,
Encoding.UTF8))//зчитує файл з даними
            {
                from.Text = sr.ReadLine();//перший рядок з файлу
                to.Text = sr.ReadLine();//другий, де я записала кінець відрізка
                number.Text = sr.ReadLine();//відповідно, третій рядок мого
файлу
            }
        }
    }
    //збереження графіків у файли
    private void button2_Click(object sender, EventArgs e)
    {
        this.chart1.SaveImage(@"F:\chart1.png", ChartImageFormat.Png);
        this.chart2.SaveImage(@"F:\chart2.png", ChartImageFormat.Png);
        this.chart3.SaveImage(@"F:\chart3.png", ChartImageFormat.Png);
        this.chart4.SaveImage(@"F:\chart4.png", ChartImageFormat.Png);
        this.chart5.SaveImage(@"F:\chart5.png", ChartImageFormat.Png);
    }
}
}
}

```

