**Лабораторная работа № 3.2  
Построение двумерных графиков функций**

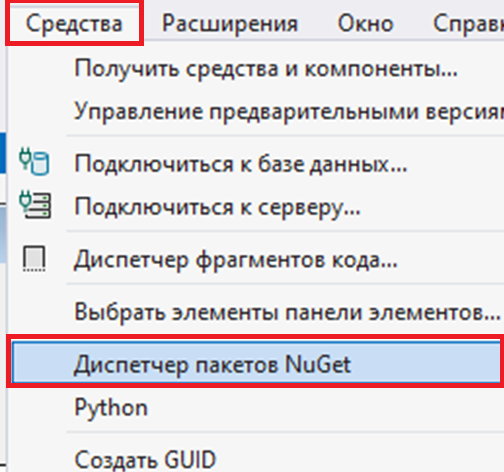
**Цель лабораторной работы**

Закрепление теоретических знаний по созданию Windows Forms приложений с отображением графиков.

**Постановка задачи**

**Построение графиков с помощью ScottPlot.**

Рассмотрим пример добавления к проекту библиотеки для построения графиков ScottPlot. Необходимо открыть диспетчер пакетов **NuGet** (рисунок 1).



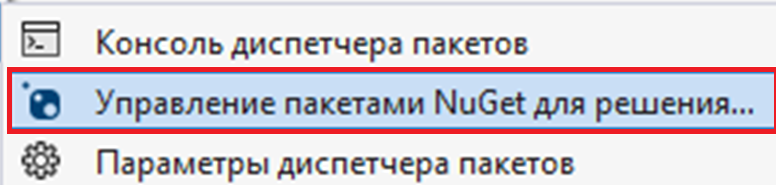


Рисунок 1 – Установка пакетов с помощью NuGet

Необходимо перейти на вкладку «Обзор» найти библиотеки ScottPlot и ScottPlot.WinForms, и нажать «Установить» (рисунок 2).

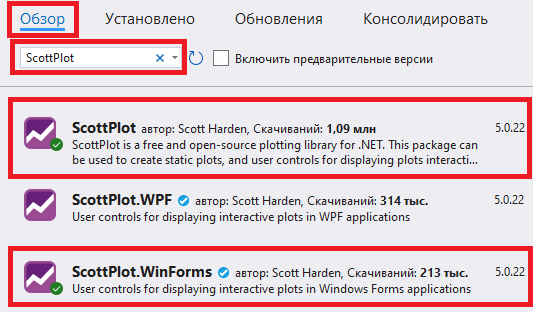


Рисунок 2 – Установка пакета ScottPlot

Элемент для построения графиков (FormsPlot) появится в панели элементов (рисунок 3). После установки пакета **возможно придется перезапустить приложение** (если появляется ошибка при добавлении элемента FormsPlot).

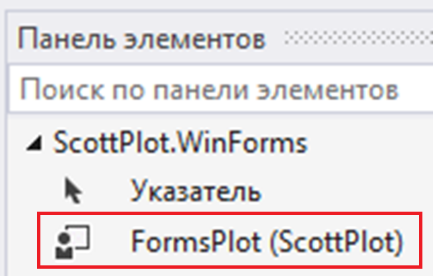


Рисунок 3 – Отображение ScottPlot в панели элементов

Добавим подписи осей, название графика и отобразим что получилось (рисунок 4).

formsPlot1.Plot.XLabel("X, м");

formsPlot1.Plot.YLabel("Y, м");

formsPlot1.Plot.Title("График");

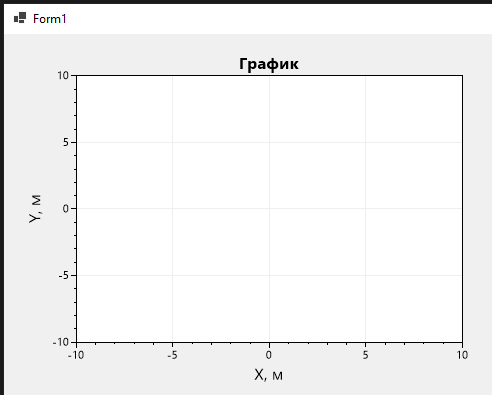


Рисунок 4 – Пустой график ScottPlot на форме

Построим график функции *f(x) = x*2 на участке от 0 до 100 (рисунок 5).

var xs = new List<double>();

var ys = new List<double>();

for (int i = 0; i <= 100; i++)

{

xs.Add(i);

ys.Add(i \* i);

}

var scatter = formsPlot1.Plot.Add.Scatter(xs, ys);

// Убираем маркеры (точки) и оставляем линию

scatter.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter.Label = "x^2"; // Подписии линии графика

formsPlot1.Plot.ShowLegend();

plot.Axes.AutoScale(); // Автоматически приближать к области графика

formsPlot.Refresh(); // Вызов метода перерисовки

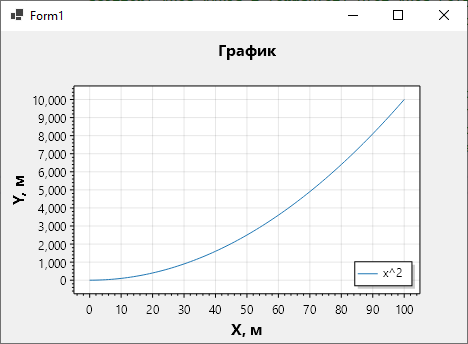


Рисунок 5 – Отображение графика функции на форме

Основное преимущество ScottPlot это **интерактивность**, график можно масштабировать и перемещаться по графику, также он сам определяет интервалы подписей у осей.

**Чтобы переместиться по графику** нужно зажать ЛКМ и двигать мышью, чтобы приблизиться (рисунок 6) или отдалиться – вращать колесико мыши, также чтобы приблизить некоторую область можно зажать колесико и выделить прямоугольник. Чтобы вернуться к исходному виду нужно нажать на колесико.

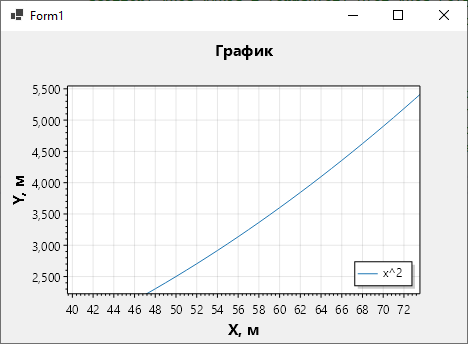


Рисунок 6 – Приближение участка графика

ScottPlot поддерживает рисование по второстепенным осям, это полезно, когда 2 набора данных сильно отличаются по значениям, но их требуется представить на одном графике (Например, давление и температура).

var xs = new List<double>();

var ys = new List<double>();

var ys2 = new List<double>();

for (int i = 0; i <= 100; i++)

{

xs.Add(i);

ys.Add(i \* i);

ys2.Add(Math.Sin(i));

}

Отобразим *x*2 на основной оси, а sin(*x*) на второстепенной оси *Oy.* Также сделаем график синуса пунктирным (рисунок 7).

var scatter1 = formsPlot1.Plot.Add.Scatter(xs, ys);

scatter1.Axes.XAxis = formsPlot1.Plot.Axes.Bottom;

scatter1.Axes.YAxis = formsPlot1.Plot.Axes.Left;

scatter1.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter1.Label = "x^2";

var scatter2 = formsPlot1.Plot.Add.Scatter(xs, ys2);

scatter2.Axes.XAxis = formsPlot1.Plot.Axes.Bottom;

scatter2.Axes.YAxis = formsPlot1.Plot.Axes.Right;

scatter2.MarkerStyle = ScottPlot.MarkerStyle.None;

scatter2.LinePattern = ScottPlot.LinePattern.Dashed;

scatter2.Label = "sin(x)";

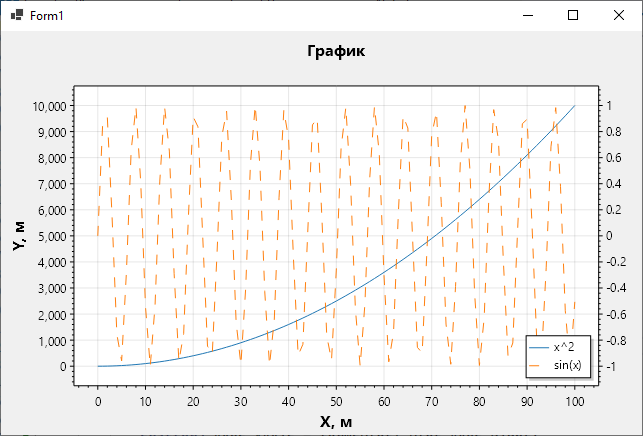


Рисунок 7 – Отображение графиков  
на главной и второстепенной осях

Рассмотрим пример построения гистограмм (рисунок 8):

var plot = formsPlot1.Plot;

double[] xs1 = { 1, 2, 3, 4 };

double[] ys1 = { 5, 10, 7, 13 };

var bars1 = plot.Add.Bars(xs1, ys1);

bars1.Label = "Alpha";

double[] xs2 = { 6, 7, 8, 9 };

double[] ys2 = { 7, 12, 9, 15 };

var bars2 = plot.Add.Bars(xs2, ys2);

bars2.Label = "Beta";

plot.ShowLegend(Alignment.UpperLeft);

// Убираем отсут от нижней границы

plot.Axes.Margins(bottom: 0);

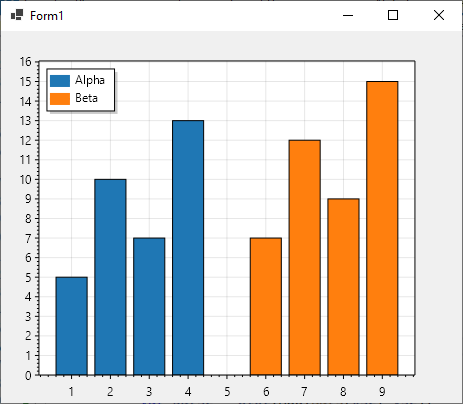


Рисунок 8 – Отображение гистограммы

Рассмотрим пример построения круговой диаграммы (рисунок 9):

var plot = formsPlot1.Plot;

List<PieSlice> slices = new()

{

new PieSlice(){ Value = 5, FillColor = Colors.Red, Label = "Red" },

new PieSlice(){ Value = 2, FillColor = Colors.Orange, Label ="Orange"},

new PieSlice(){ Value = 8, FillColor = Colors.Gold, Label ="Yellow"},

new PieSlice(){ Value = 4, FillColor = Colors.Green, Label ="Green"},

new PieSlice(){ Value = 8, FillColor = Colors.Blue, Label ="Blue" }};

var pie = plot.Add.Pie(slices);

pie.ExplodeFraction = 0.1;

plot.ShowLegend();

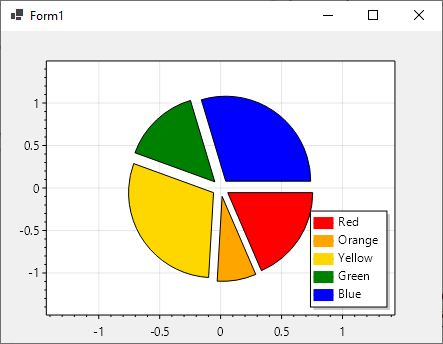


Рисунок 9 – Отображение гистограммы

**Анимация.**

Здесь необходим компонент Timer. Требуется добавить 2 кнопки – для запуска и остановки движения.

public partial class Form1 : Form

{

private double x = 0, speed = 0.1;

private List<double> xs = new();

private List<double> ys = new();

public Form1()

{

InitializeComponent();

}

private void buttonStart\_Click(object sender, EventArgs e)

{

timer.Start();

}

private void buttonStop\_Click(object sender, EventArgs e)

{

timer.Stop();

}

private double getF(double x) => Math.Sin(x);

private void timer\_Tick(object sender, EventArgs e)

{

x += speed;

xs.Add(x);

ys.Add(getF(x));

var plot = formsPlot.Plot;

plot.Clear();

var scatter = plot.Add.Scatter(xs, ys);

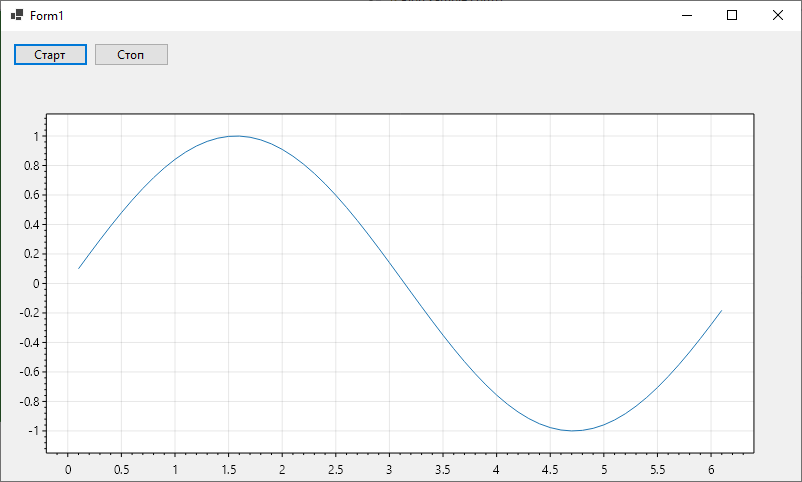
scatter.MarkerStyle = MarkerStyle.None;

plot.Axes.AutoScale();

formsPlot.Refresh();

}

}

****

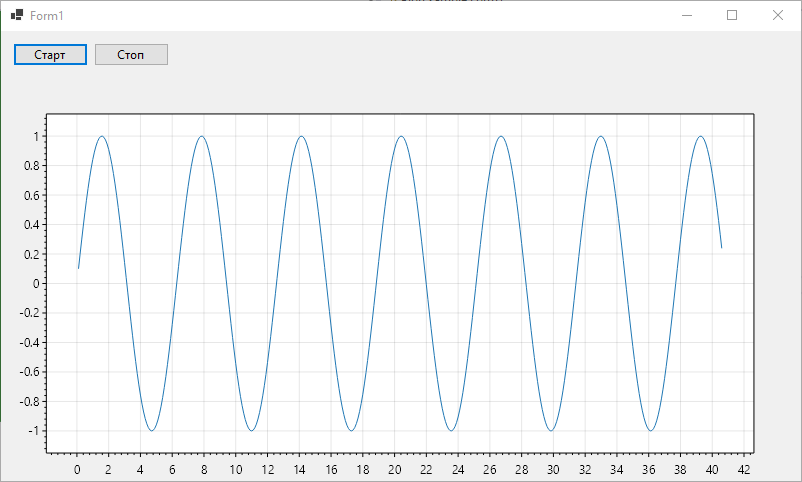
****

Рисунок 10 – Отображение анимации синусоиды

Примеры построения графиков и работы в ScottPlot приводятся в документации на сайте: <https://scottplot.net/cookbook/5.0/>

**Задание 1**

Необходимо построить 2D графики в среде MS Visual Studio на языке C# (компонент ScottPlot).

1. Построить график функции *y=f(x)*. Должны быть подписаны оси координат с единицами измерения, отображены линии сеток. Предусмотреть возможность задания цвета и толщины графика.
2. Реализовать возможность установки (и удаления) маркера при нажатии ПКМ по графику, предусмотреть ввод имени маркера. Маркер должен появляться в легенде.
3. Реализовать возможность отображения значения функции рядом с мышью при наведении мыши на график.

Переменные являются осями координат для графиков, параметры вводятся с формы.

|  |  |
| --- | --- |
| **№ Варианта** | **График** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 |  |
| 9 |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |
| 15 |  |
| 16 |  |
| 17 |  |
| 18 |  |

**Задание 2**

Используя метод CubicSpline.InterpolateNatural интерполировать данные и построить график функции.

В качестве данных используем таблицу стандартной атмосферы: <https://ru.wikipedia.org/wiki/%D0%A1%D1%82%D0%B0%D0%BD%D0%B4%D0%B0%D1%80%D1%82%D0%BD%D0%B0%D1%8F_%D0%B0%D1%82%D0%BC%D0%BE%D1%81%D1%84%D0%B5%D1%80%D0%B0>

Независимая переменная (*x*) – высота, м. Зависимая переменная (*y*) – по варианту.

Варианты:

1. температура,
2. скорость звука,
3. давление,
4. плотность,
5. кинематическая вязкость.