**Scientific graphics in C# - part 2**

**Posted on October 9, 2011 by Paul**

*The code for this tutorial is on GitHub:* <https://github.com/sol-prog/GraphDemo2>.

In the [first part](https://solarianprogrammer.com/2011/10/06/2d-scientific-graphics-in-c/) of this tutorial I've shown you how to plot some data on a Windows Form using C#. It is time now to improve the graphical aspect of our DemoGraph, we will start by properly formatting the numbers from the x axis, we could do this by modifying the Plot.cs class:

1 ...

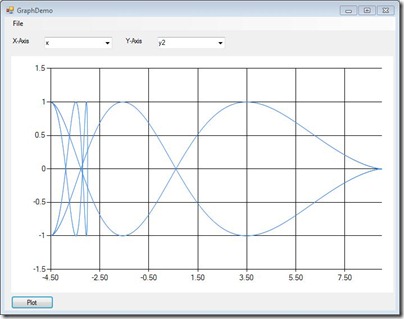
2 chart.Series.Add("Series0");

3 chart.Series[0].ChartType = SeriesChartType.Line;

4 chart.ChartAreas[0].AxisX.LabelStyle.Format = "{F2}";

5 ...

The above line of code will show the numbers from the x axis as two digits floating numbers:



Definitely better! A further improvement will be to add names on the axes, this way when you will save a picture you will know which variable was plotted on which axis. We will use the header from the input file to get these names:

1 ...

2 **public** Plot(Read rr, ComboBox xBox, ComboBox yBox, Chart chart)

3 {

4 **int** indX = xBox.SelectedIndex;

5 **int** indY = yBox.SelectedIndex;

6 **float**[,] data = rr.get\_Data();

7 **int** nLines = rr.get\_nLines();

8 **string** []header = rr.get\_Header();

9

10 chart.Series.Clear(); *//ensure that the chart is empty*

11 chart.Series.Add("Series0");

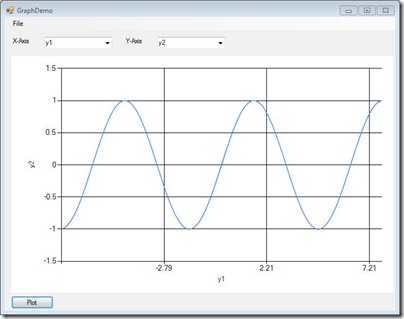
12 chart.Series[0].ChartType = SeriesChartType.Line;

13 chart.ChartAreas[0].AxisX.LabelStyle.Format = &quot;{F2}&quot;;

14 chart.ChartAreas[0].AxisX.Title = header[indX];

15 chart.ChartAreas[0].AxisY.Title = header[indY];

16 ...



Obviously in the above picture we need to do something about the number of grid lines corresponding to the x axis, suppose we want to see six grid lines instead of three, we can achieve this by specifying the length of the grid interval and (equally important for the overall aspect) the length of the label interval:

1 ...

2 **public** Plot(Read rr, ComboBox xBox, ComboBox yBox, Chart chart)

3 {

4 **int** indX = xBox.SelectedIndex;

5 **int** indY = yBox.SelectedIndex;

6 **float**[,] data = rr.get\_Data();

7 **int** nLines = rr.get\_nLines();

8 **int** nColumns = rr.get\_nColumns();

9 **string** []header = rr.get\_Header();

10

11 chart.Series.Clear(); *//ensure that the chart is empty*

12 chart.Series.Add("Series0");

13 chart.Series[0].ChartType = SeriesChartType.Line;

14 chart.ChartAreas[0].AxisX.LabelStyle.Format = "{F2}";

15 chart.ChartAreas[0].AxisX.Title = header[indX];

16 chart.ChartAreas[0].AxisY.Title = header[indY];

17

18 **float** x\_grids = 6;

19 **float**[] ext = get\_Extrema(data, nLines, nColumns, indX);

20 chart.ChartAreas[0].AxisX.MajorGrid.Interval = (ext[1] - ext[0]) / x\_grids;

21 chart.ChartAreas[0].AxisX.LabelStyle.Interval = (ext[1] - ext[0]) / x\_grids;

22 chart.ChartAreas[0].AxisX.MajorTickMark.Interval = (ext[1] - ext[0]) / x\_grids;

23 ...

We will also need an auxiliary function that will return the minima and maxima of data column corresponding to the x axis:

1 **class** Plot

2 {

3 **public** Plot(Read rr, ComboBox xBox, ComboBox yBox, Chart chart)

4 {

5 ...

6 }

7

8 **private** **float** []get\_Extrema(**float**[,] data,**int** nL,**int** nC,**int** idx)

9 {

10 **float** min = data[0, idx],max = data[0, idx];

11 **float**[] res = **new** **float**[2];

12 **for** (**int** i = 1; i < nL; i++)

13 {

14 **if** (min > data[i, idx]) min = data[i, idx];

15 **if** (max < data[i, idx]) max = data[i, idx];

16 }

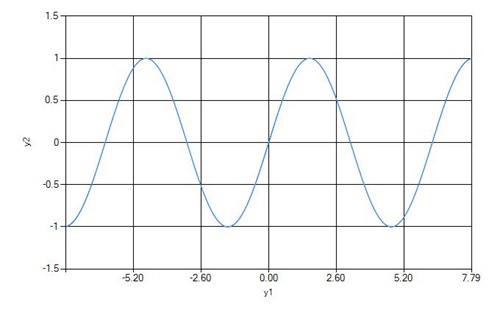
17 res[0] = min; res[1] = max;

18 **return** res;

19 }

20 }

Of course a similar procedure can be applied if you need to customize the properties of the y axis. With the above code this is how our application looks:



A further enhancement will be to change the format in which a picture is saved from jpg to a higher quality tiff file. Unfortunately the Chart control doesn't include the possibility to save an image as a PostScript file. However, tiff files are accepted even for journal publications, so if you need a high quality output just change jpg to tiff in the Save function.

In a real life application you won't use hard coded values for the number of grid lines of an axis, or for the color of a graph - you will add a Settings panel to your application from which the user will be able to change these parameters. This tutorial was made only for illustrative purposes so I wasn't always careful with all the little details of a robust application.

Next time I will show you how to add more than a graphics on a Chart control and how to use legends and colors. Also I will present you, for the sake of argument how you can achieve a similar quality in about twenty lines of code with Python and Matlab!

<https://solarianprogrammer.com/2011/10/09/scientific-graphics-in-c-part-2/>