

Creating Plots with Gnuplot

May 18, 2017

What is Gnuplot

Gnuplot is a freely available command-line based interactive plotting program.

```
$ gnuplot
```

```
  G N U P L O T
  Version 4.6 patchlevel 0      last modified 2012-03-04
  Build System: Linux x86_64

  Copyright (C) 1986-1993, 1998, 2004, 2007-2012
  Thomas Williams, Colin Kelley and many others

  gnuplot home:      http://www.gnuplot.info
  faq, bugs, etc:   type "help_FAQ"
  immediate help:   type "help"  (plot window: hit 'h')
```

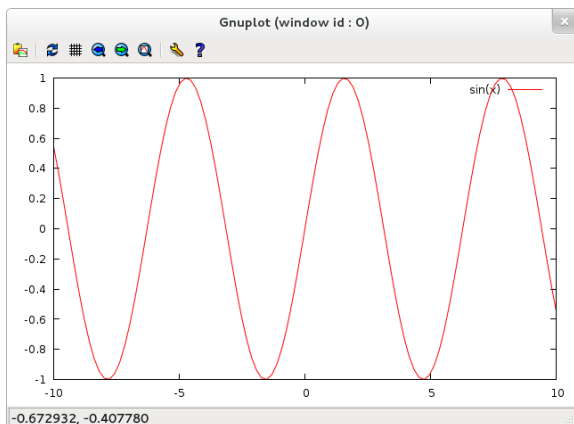
```
Terminal type set to 'wxt'
gnuplot>
```

Gnuplot basics

Let us follow the same examples we used in Python and try to recreate the same plots using Gnuplot.

Plotting a simple sine function:

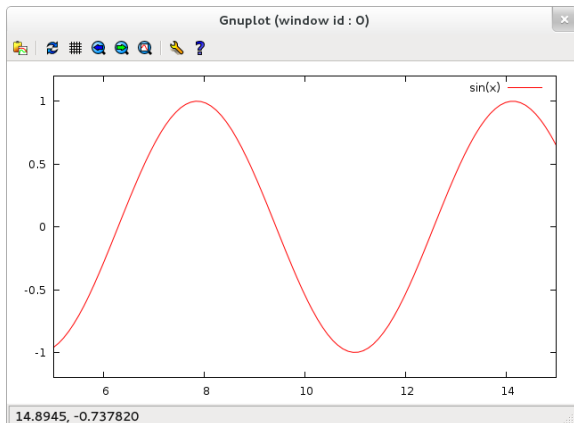
```
gnuplot> plot sin(x)
```



Gnuplot basics

Now let us adjust the axes limits, just as before. In Gnuplot we have to use the `xrange` and `yrange` variables.

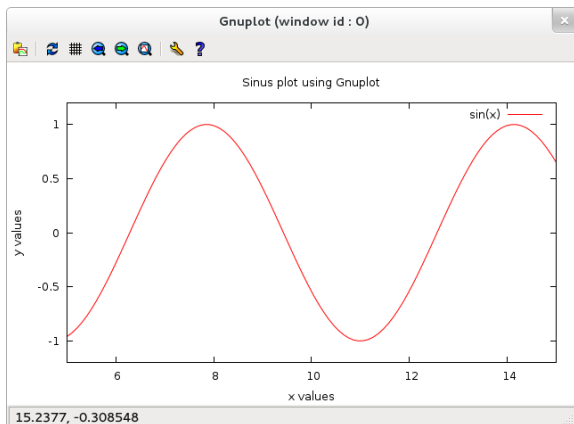
```
gnuplot> set xrange [5: 15]
gnuplot> set yrange [-1.2:1.2]
gnuplot> plot sin(x)
```



Gnuplot basics

Next, let us add axes labels and a title.

```
gnuplot> set xlabel "x values"  
gnuplot> set ylabel "y values"  
gnuplot> set title "Sinus plot using Gnuplot"  
gnuplot> plot sin(x)
```



Gnuplot basics

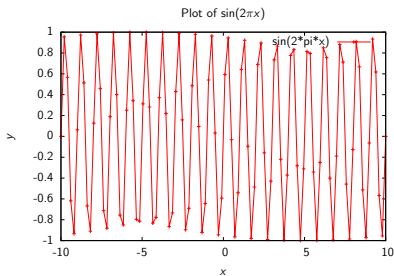
Depending on the terminal we can also use \LaTeX labels.

```
gnuplot> set term epslatex color standalone
gnuplot> set output 'sin_latex.tex'
gnuplot> set xlabel '$x$'; set ylabel '$y$'
gnuplot> set title 'Plot of  $\sin(2 \pi x)$ '
gnuplot> plot sin(2*pi*x)
$ latex sin_latex.tex # compile LaTeX and create a DVI file
$ dvi2pdf sin_latex.dvi # converts DVI to PDF
```

Gnuplot basics

Depending on the terminal we can also use \LaTeX labels.

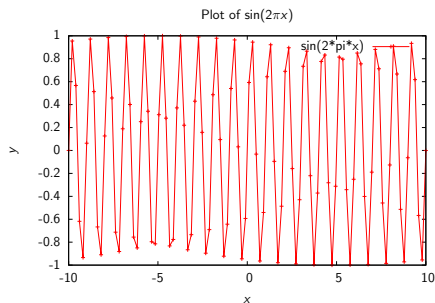
```
gnuplot> set term epslatex color standalone
gnuplot> set output 'sin_latex.tex'
gnuplot> set xlabel '$x$'; set ylabel '$y$',
gnuplot> set title 'Plot of  $\sin(2\pi x)$ '
gnuplot> plot sin(2*pi*x)
$ latex sin_latex.tex # compile LaTeX and create a DVI file
$ dvi2pdf sin_latex.dvi # converts DVI to PDF
```



What happened to the plot? Since we do not set an array of function arguments, Gnuplot chooses its own number of samples.

Gnuplot basics

The default sample size is 100. So let's replot the same plot using lines with points.

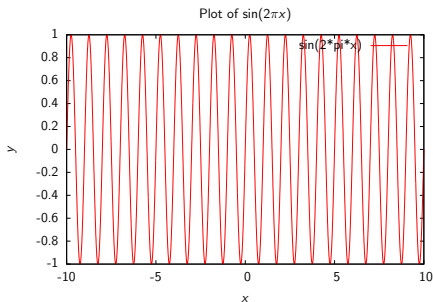


Gnuplot basics

We can increase the number of points by manually setting the sample size

```
gnuplot> set samples 5000
```

and replotting the same function again.



Gnuplot basics

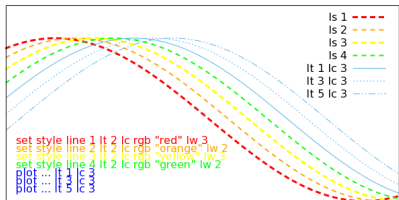
Depending on the terminal you can set different line styles using the `linetype` / `lt` command.

The width of the lines can be altered using the `linewidth` / `lw` command.

The color of the lines is set via the `linecolor` / `lc` switch.

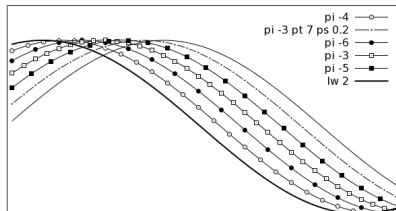
For scatter plots the size and type of the points can be altered using `pointtype` / `pt` and `pointsize` / `ps`.

Independent colors and dot/dash styles



You will only see dashed lines if your current terminal setting permits it

The pointinterval property is another way to create interrupted lines

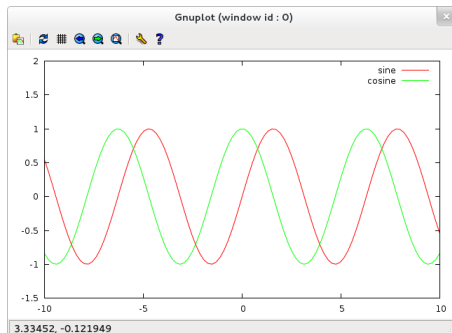


This technique works best for equally spaced data points

Gnuplot basics

To add a legend you need to set a title for each function you are plotting.

```
gnuplot> set yrange [-1.5:2.0]
gnuplot> set key top right
gnuplot> plot sin(x) title 'sine', cos(x) title 'cosine'
```

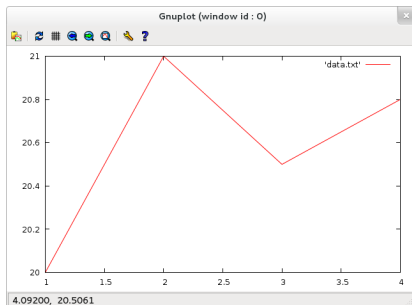


Gnuplot basics

Now, to plot discrete values instead of the built-in functions, we need to load data from a file.

```
gnuplot> plot 'data.txt' w l
```

This reads the file `data.txt` and automatically uses the first two columns as `x` and `y` axes, respectively.



You can specify which columns you want to plot with the using `x:y` command.

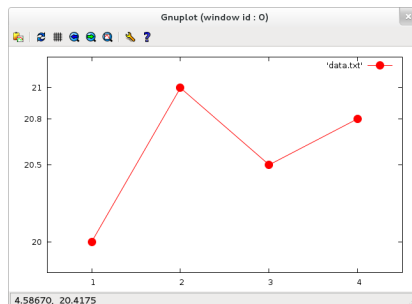
```
gnuplot> plot 'data.txt' using 4:5 with lines
```

```
gnuplot> p 'data.txt' u 6:7 w lp
```

Gnuplot basics

Manipulating the axes ticks is just as easy as it was with Python.

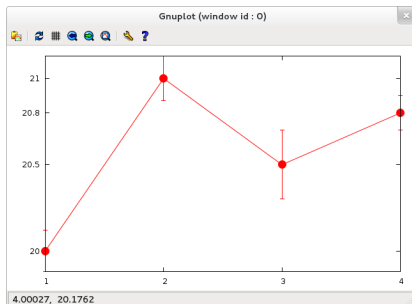
```
set xrange [0.5:4.5]
set yrange [19.8:21.2]
set xtics (1, 2, 3, 4)
set ytics (20, 21, 20.5, 20.8)
plot 'data.txt' w lp ps 2 pt 7
```



Gnuplot basics

To plot errorbars you need to add `with yerrorbars` to your plot command. Gnuplot then assumes the third column contains the error data. Or you can specify a column as before.

```
gnuplot> plot 'data.txt' u 1:2:3 w yerrorbars
```



Gnuplot basics

Now let us put everything together to create the complete plot with title, labels and create a PDF.

```
# contents of plot.gp
set output 'plot.tex'
set xrange[0.5:4.5]
set yrange[19.8:21.2]
set xtics (1, 2, 3, 4)
set ytics (20, 21, 20.5, 20.8)
set xlabel 'This is $x$'
set ylabel 'This is $y$'
set title 'My beautiful plot'
unset key
plot 'data.txt' with lines,\
      'data.txt' w yerrorbars pt 7 ps 2 lc 1
```

```
$ gnuplot plot.gp
$ latex plot.tex
$ dvi2pdf plot.dvi
```

Gnuplot basics

The final result then looks like this.

