

(Very) brief introduction to gnuplot

Start gnuplot:

```
bash$ gnuplot
```

Plot a function:

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

Plot two functions:

```
gnuplot> plot sin(x), 3.2*x+5.9
```

Define variables and functions:

```
gnuplot> sigma=0.77
```

```
gnuplot> gauss(q,a)=exp(-(q/a)**2)/(a*sqrt(pi))
```

```
gnuplot> plot gauss(x,sigma)
```

Use gnuplot as calculator:

```
gnuplot> print 1.+2.*3./4., 1+2*3/4
```

```
gnuplot> print sin(pi/3.)**2
```

```
gnuplot> print gauss(0.5,sigma)
```

Browse the builtin help system:

```
gnuplot> help
```

```
gnuplot> help plotting
```

```
gnuplot> help style
```

Plotting data from files

Suppose you have a file "mydata.txt" containing data:

#	Nr.	foo	bar
	1	3.1	4.1
	2	5.9	2.6
	3	5.3	5.8

Plot "bar" (column 3) against "foo" (column 2):

```
gnuplot> plot "mydata.txt" using 2:3
```

Plot the square of "bar" against "foo":

```
gnuplot> plot "mydata.txt" using 2:($3**2)
```

Fitting functions to data

Define an expression to be fitted and give suitable initial guesses for the parameters:

```
gnuplot> f(x)=a*x+b  
gnuplot> a=3  
gnuplot> b=5
```

Perform the fit and plot the resulting function:

```
gnuplot> fit f(x) "mydata.txt" u 2:($3**2) via a,b  
gnuplot> plot "mydata.txt" u 2:($3**2), f(x)
```

Refine the appearance of the plot

Set axis ranges:

```
gnuplot> set autoscale x  
gnuplot> set yrange [0:40]
```

Label the axes and give the plot a title:

```
gnuplot> set title "observation run #1"  
gnuplot> set xlabel "foo"  
gnuplot> set ylabel "bar^2"  
gnuplot> set key bottom right
```

Plot with different styles:

```
gnuplot> plot "mydata.txt" u 2:3 with lines linetype 3  
gnuplot> plot sin(x) with linespoints pointtype 5  
gnuplot> plot sin(x) w lp lt 3 pt 5, cos(x) w l lt 4 lw 3
```

Show available styles for the current "terminal":

```
gnuplot> test
```

Save the finished plot in a file

Need to print a plot? Need a plot for inclusion in another document? Generate a Postscript file:

```
gnuplot> set terminal postscript enhanced color  
gnuplot> set output "myplot.ps"  
gnuplot> replot  
gnuplot> unset output  
gnuplot> set terminal x11
```

Gnuplot script files

For complex plots it's often more convenient to create a text file, for example "myplot.plt", containing all the plotting commands, and then run this file with

```
bash$ gnuplot myplot.plt
```

from the shell, or

```
gnuplot> load "myplot.plt"
```

from within gnuplot.

Example gnuplot script "myplot.plt":

```
# generate a postscript plot from my data
set term post enh eps color solid 24
set output "myplot.ps"
set encoding iso_8859_1 # for real minus signs

a=3; b=5 # initial guess for parameter fit
fit a*x+b "mydata.txt" u 2:($3**2) via a,b
set auto x
set yrange [0:40]
set title "observation run #1"
set xlabel "foo"
set ylabel "bar^2"
set key top left reverse Left
plot \
  "mydata.txt" u 2:($3**2) w p pt 7 title "observed", \
  a*x+b w l lt 2 title "linear fit to data", \
  3.2*x+5.9 w l lt 3 title "theoretical prediction"
```

View generated plot:

```
bash$ gv myplot.ps
```

