The basics
of
gnuplot

Course material for PHYS2200 class

Storrs, October 25, 2016

Plotting functions

To start the *interactive* use of gnuplot, type **gnuplot** at the command prompt.

```
% gnuplot

        G N U P L O T
        G N U P L O T
        Version 5.0 patchlevel 3    last modified 2016-02-21

        Copyright (C) 1986-1993, 1998, 2004, 2007-2016
        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>
```

In our first graph we want to plot a sinusoid and a cosinus. Therefore we specify our functions and plot them:

```
gnuplot> a = 0.9
gnuplot> f(x) = a * sin(x)
gnuplot> g(x) = a * cos(x)
gnuplot> plot [0:10] f(x) title 'sin(x)' with lines linestyle 1, \
>                    g(x) notitle w l ls 2
```

The definitions of functions in gnuplot are straight forward. We want to plot more than one function thats why we have to divide the two commands with a comma. The backslash tells gnuplot that we have a line break at this position. We can also abbreviate commands ('w' for with, etc.).

The result of the command is presented in Fig. 1.

We can use gnuplot in non-interactive mode by saving the plotting commands to a file, say **functionplot.gp**,

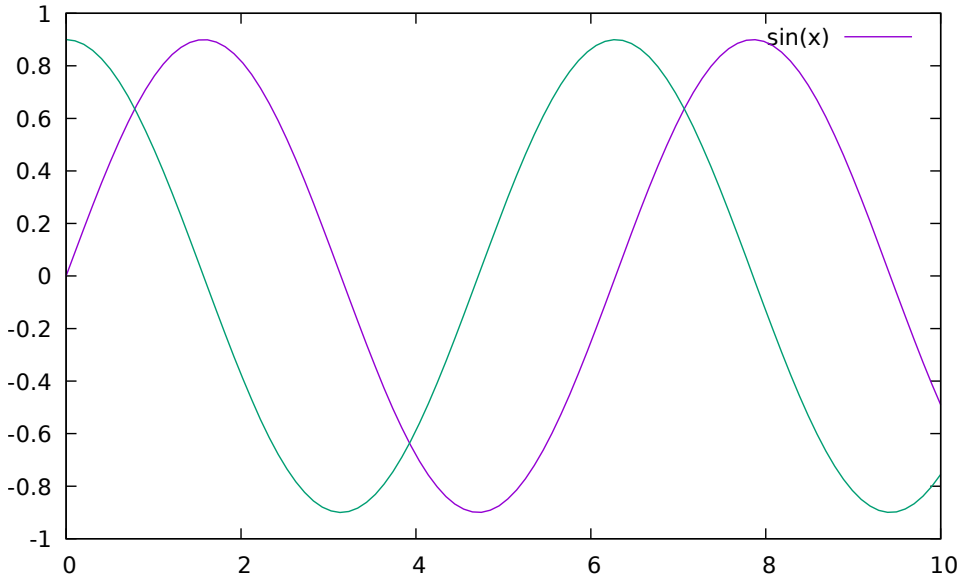

Figure 1: The graph produced by the first plot command (colors online).

Listing 1: Gnuplot script stored in the file 'functionplot.gp'

```
a = 0.9
f(x) = a * sin(x)
g(x) = a * cos(x)
plot [0:10] f(x) title 'sin(x)' with lines linestyle 1, \
     g(x) notitle w l ls 2
```

and executing them:

```
% gnuplot -p functionplot.gp
```

The command line option '-p' instructs gnuplot to keep the figure on the screen after finishing plotting.

Plotting data

Plotting data, e.g. results of calculations or measurement, works basically like the plotting of functions. We need a data file and some commands to manipulate the data. Let's start

with the basic plotting of simple data. A *data file* can be a text file containing the datapoints as columns.

Listing 2: Data file 'dataplot.res'

```
# dataplot.res
# X     Y     Z
  1.    1.    1.
  2.    4.    8.
  3.    9.   27.
  4.   16.   64.
  5.   25.  125.
```

A line starting with a '#' is a comment and is ignored by gnuplot.

A command to plot the data in the file is

```
gnuplot> plot "dataplot.res" using 1:2 with linespoints, \
>             "" u 1:3 w lp
```

The result of the command is presented in Fig. 2.

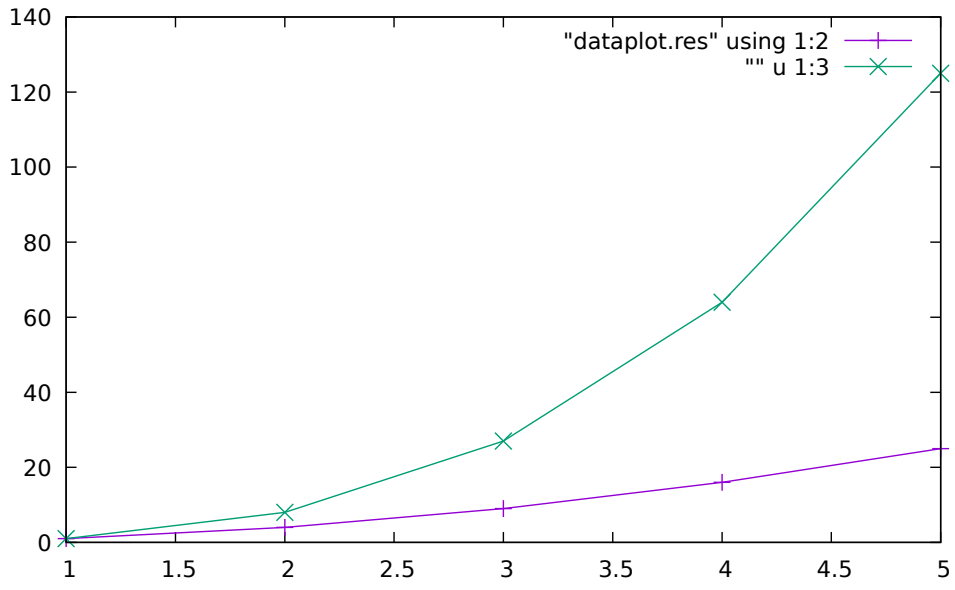

Figure 2: The graph produced by the second plot command (colors online).

Output terminals

Gnuplot can produce plots in a lot of different formats. It uses different **output terminals** that can produce output files. We already used the wxt terminal there which displays the result on the screen.

For example, to save the output as a pdf file, we use the following commands:

```
gnuplot> set terminal pdfcairo
gnuplot> set output "functionplot.pdf"
```

followed by the *plot* command. Alternatively we can include the above two lines at the beginning of a gnuplot script.

To save the output as a png file, we can use the following commands:

```
gnuplot> set t pngcairo
gnuplot> set o "functionplot.png"
```