

Brief introduction to gnuplot

- gnuplot is available for almost every platform (operating system): Linux, MacOS X, Windows, ...
- download, e.g, from <http://gnuplot.info/>
- under Linux: start interactive session in terminal via `gnuplot`
- quit gnuplot by command `exit`

- gnuplot can plot basic functions (independent variable / dummy variable is x) and combinations of them, default plot symbol for functions: solid line
- examples
 - `plot sin(x)`
 - `plot x**3 + 0.5*sqrt(2)`
- plotting more than one function by using comma separated list:
`plot sin(x), cos(1/x), tanh(x+2)`

gnuplot plots data from files in ascii table format, i.e.

```
# this is a comment
4.5 91 -0.5
5.6 70 0.8
19 200 1.1
```

- Columns are separated by blanks. Can be changed before plotting, e.g.,
set datafile separator "," (comma separated)
set datafile separator "\t" (separated by tabs)
- plot "file.txt" → default: plots 2nd column over 1st column
- plot 'filexyz.txt' using (\$2):(\$3) → plots 3rd column over 2nd column
- plot 'data.txt' u (log10(\$1)):(log10(\$2)) → plots the decadic logarithm of the data in columns 1 and 2 (double-logarithmic plot)

with help of the Levenberg-Marquardt algorithm `gnuplot` can fit any function with free parameters to data:

① define function: $f(x) = a * x + b$

② fitting examples:

```
fit f(x) "data.txt" via a, b
```

```
fit f(x) "data.txt" u (log10($1)):(log10($2)) via a, b
```

③ plotting data and function:

```
plot "data.txt", f(x)
```

- x- and y-axis labels: `set xlabel "d in pc",`
`set ylabel "t in Ga"`
- key (legend): is automatically generated, can be written by option `title`: `plot`
`"data.txt" title "observation (1998)" \`
`, f(x) t "model 17-04"`

→ requires execution of previous plot command (or just replot)

gnuplot supports many different output formats (see `→help terminal`)

- ➊ `set terminal pdf enhanced color` → sets terminal (output *format*) to colored pdf with special characters
- ➋ `set output "myplot.pdf"` → name of the file for output (don't forget it!)
- ➌ `plot "data.txt", f(x) or replot`
- ➍ either: `set term qt` (resetting terminal to previous output format) or `quit` → this assures that the plot is *written* to the file (otherwise: empty file)

if output is written to PDF or PS file, via option enhanced:

Input	Output in PDF/PS
<code>T_0</code>	T_0 (subscript)
<code>e^{-x}</code>	e^{-x} (superscript)
<code>{/Symbol Qp}</code>	$\Theta\pi$

besides interactive mode, gnuplot supports also non-interactive script mode

- write all instructions into an ASCII text file (e.g., “myplot.gplt”) comments begin with a # (like in makefile and shell) line continuation via backslash \
- execute gnuplot script from shell:
gnuplot myplot.gplt

→ useful for automated PDF creation

→ easy re-use of formatting and plot instructions (labels, sizes, ...)

Example for fitting and pdf output I

```
set terminal pdf enhanced color
set xlabel "1/T [100/K]"
set ylabel "ln(p/p_0)"
ln_p(x) = b + a*x
set fit errorvariables
R=8.314
p_0=1.019
fit [*:] ln_p(x) "enthalpie.dat" \
using (1./(($2)+273.15)):(log((1.019+($1))/1.019)) via a,b
set output "enthalpy.pdf"
plot 'enthalpie.dat' \
using (1e2/(($2)+273.15)):(log((p_0+($1))/p_0)) \
with points ps 1 linewidth 3 title "data" \
, ln_p(1e-2*x) with lines linecolor "black" \
t sprintf("enthalpy [kJ/mol]=%5.3f +/- %5.3f",a*R*1e-3,R*a_err*1e-3)
```

Example for fitting and pdf output II

