

GNU PLOT

Gnuplot je pokročilý program určený pro tvorbu grafů. Je zdarma, dostupný pod Windows, OSX i Linux.

Stáhnout si jej můžete z oficiálních stránek:

<http://www.gnuplot.info/download.html>

Poté, co program rozbalíte/nainstalujete, stačí jej spustit příkazem *gnuplot* (Linux), ve Windows pak souborem *gnuplot.exe* (či *wgnuplot.exe*) z podadresáře *bin*.

Po spuštění programu se otevře příkazová řádka, kam můžete zadávat jednotlivé kontrolní příkazy.

Nahrání souboru s daty:

Data v souboru by měla být rozdělena do sloupečků oddělených tabulátorem či mezerou. V jednom sloupečku x-ové hodnoty, v dalším pak y-ové. Pokud mají naměřené hodnoty známou chybu (např. standardní odchylku arit. průměru σ), uvádí se v dalších dvou sloupcích a to minimální hodnota (= hodnota $- 1\sigma$) a maximální hodnota (= hodnota $+ 1\sigma$) dané veličiny.

Čísla musejí obsahovat desetinné tečky, nikoliv čárky!!!

Zde je ukázka formátu vstupního souboru:

#x	y	y_min	y_max
1	20.6	15.6	25.6
2	41.6	36.6	46.6
3	62.8	57.8	67.8
4	84.4	79.4	89.4
5	104.2	99.2	109.2

Data ze souboru „data.txt“ vykreslíme příkazem:

```
plot 'data.txt' using 1:2
```

nebo (s errorbary):

```
plot 'data.txt' using 1:2:3:4 with yerrorbars
```

Za přepínačem *using* následují čísla sloupců obsahujících jednotlivá data podle následujícího klíče: *x:y:y_min:y_max*

Takže pokud jsou například v prvním sloupci čísla měření, ve druhém sloupci y-nové hodnoty, v dalších sloupcích jejich chyby a v posledním sloupečku x-ové hodnoty, použijeme příkaz:

```
... using 5:2:3:4 ...
```

Je důležité, aby se vstupní soubor nacházel ve stejném adresáři, ze kterého jsme gnuplot spouštěli, v opačném případě musíme použít příkaz *cd* pro změnu adresáře nebo přidat cestu k souboru jako součást názvu, např.:

```
cd 'C:/Dokumenty/praktika zima/uloha 1/'  
plot 'data.txt' ....
```

nebo

```
plot 'C:/Dokumenty/praktika zima/uloha 1/data.txt' ...
```

Fitování:

Abychom mohli naměřená data nafitovat určitou funkcí, musí program tuto funkci znát. Řadu základních funkcí má již gnuplot nadefinovanou implicitně:

Function	Returns
abs(x)	absolute value of x, x
acos(x)	arc-cosine of x
asin(x)	arc-sine of x
atan(x)	arc-tangent of x
cos(x)	cosine of x, x is in radians.
cosh(x)	hyperbolic cosine of x, x is in radians
erf(x)	error function of x
exp(x)	exponential function of x, base e
inverf(x)	inverse error function of x
invnorm(x)	inverse normal distribution of x
log(x)	log of x, base e
log10(x)	log of x, base 10
norm(x)	normal Gaussian distribution function
rand(x)	pseudo-random number generator
sgn(x)	1 if x > 0, -1 if x < 0, 0 if x=0
sin(x)	sine of x, x is in radians
sinh(x)	hyperbolic sine of x, x is in radians
sqrt(x)	the square root of x
tan(x)	tangent of x, x is in radians
tanh(x)	hyperbolic tangent of x, x is in radians

Bessel, gamma, ibeta, igamma, and lgamma functions are also supported. Many functions can take complex arguments. Binary and unary operators are also supported.

Pokud potřebujeme něco složitějšího, je třeba to nadefinovat, např.:

$$f(x) = A \cdot \exp\left(-\frac{(x-B)^2}{2C^2}\right)$$

což je Gaussova funkce v proměnné x a s parametry A, B, a C: $f(x) = A e^{-\frac{(x-B)^2}{2C^2}}$.

Aby fitování zkonvergovalo ke konečnému výsledku, je dobré nastavit všem parametrům nějakou rozumnou počáteční hodnotu, která řádově odpovídá očekávanému výsledku:

```
A=1
B=0
C=1
```

A pak už můžeme zahájit proces fitování:

```
fit f(x) 'data.txt' using 1:2:3:4 via A,B,C
```

Přepínač *via* programu říká, jaké parametry se mají během fitu optimalizovat.

Pokud fitování dokonverguje k nějakému výsledku, vypíše program nalezené hodnoty zadaných parametrů i s jejich chybou. To je to, co nás zajímá.

Na závěr si data i výslednou funkci vykreslíme:

```
plot 'data.txt' using 1:2:3:4 with yerrorbars, f(x)
```

Finální úpravy:

Do grafu je vhodné přidat popisky os:

```
set xlabel  "{/Symbol D}x [s]"  #řecké písmeno Delta
set ylabel  "v [m/s]"
```

Případně změnit jejich rozsah:

```
set xrange [ 0 : 20 ]
set yrange [ 100 : 300 ]
```

Někdy je vhodné zobrazit některou z os v logaritmické škále:

```
set logscale y
```

Pomocná mřížka:

```
set grid x y
```

Legenda se vytváří automaticky, názvy jednotlivých položek nastavíme přepínačem *title* v příkazu *plot*:

```
titulek1="naměřená data"
titulek2="fit"
plot 'data.txt' using 1:2 title titulek1, f(x) title titulek2
```

Výsledný obrázek je možné vložit do schránky a pak jej uložit pomocí jakéhokoliv editoru obrázků (včetně „malování“) nebo nastavit ukládání do souboru (postscript):

```
set terminal postscript
set output 'obrazek.ps'
plot ...
```

nebo třeba png:

```
set terminal png
set output 'obrazek.png'
plot ...
```

Na závěr ukázková sekvence příkazů:

```
f(x)=A*x+B
A=1
B=0
fit 'data.txt' using 1:2 via A,B
titulek1="naměřená data"
titulek2="lineární funkce"
set xlabel  "t [s]"
set ylabel  "h [cm]"
set grid x y
```

```
set terminal png
set output 'obrazek.png'
plot 'data.txt' using 1:2 title titulek1, \
      f(x) title titulek2
```

Další zdroje:

Na následujících stránkách naleznete další návody a tutoriály k programu GNUPLOT:

<http://www.root.cz/serialy/gnuplot/>

<http://www.duke.edu/~hpgavin/gnuplot.html>

<http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>

<http://www.ibm.com/developerworks/library/l-gnuplot/>