

Gnuplot – program za risanje grafov

B. Golli, PeF & IJS

17. november 2009

Kazalo

1	Uvod	2	4.1.1	Resolucija slike	7
2	Namestitev programa na različnih platformah	2	4.1.2	Odsekoma definirane krivulje	7
2.1	MS Windows	2	4.1.3	Oznake na oseh	7
2.2	Namestitev programa na drugih operacijskih sistemih	2	4.2	Parametrično podane krivulje	8
3	Gnuplot na hitro	3	4.3	Grafi v 3D	8
3.1	Risanje funkcij	3	4.4	Preglednica manj pogostih funkcij	8
3.1.1	Sprememba intervala na abscisi in ordinati	3	5	Predstavitev in analiza meritev	9
3.1.2	Velikost slike in razmerje višine in širine	3	5.1	Grafični prikaz meritev z odstopanji	9
3.1.3	Barva in debelina črte	4	5.2	Gladka krivulja skozi izmerke	9
3.1.4	Oznake	4	5.3	Računske operacije na in med stolpci v tabeli	9
3.1.5	Zgled	4	5.4	Prilagajanje krivulje skozi izmerjene točke	10
3.1.6	Preglednica nekaterih predefiniranih funkcij	4	5.5	Prilagajanje z upoštevanjem merskih napak	10
3.2	Meritve	5	5.6	Več slike v istem okviru	11
3.2.1	Prikaz izmerjenih količin	5	6	Terminalske izbire	11
3.2.2	Izbira znaka za točke	5	6.1	Terminal windows	11
3.2.3	Prilagajanje premice	5	6.2	Terminali wxt in x11	11
3.3	Izpis slike na datoteko	6	6.3	Izpis v formatu PostScript	12
3.4	Zaključek	6	6.4	Izbira enhanced	12
4	Pregled nekaterih ukazov za risanje funkcij	7	6.4.1	Indeksi in eksponenti	12
4.1	Krivulje v 2D	7	6.4.2	Grške črke in posebni znaki	13
			7	Literatura	13

1 Uvod

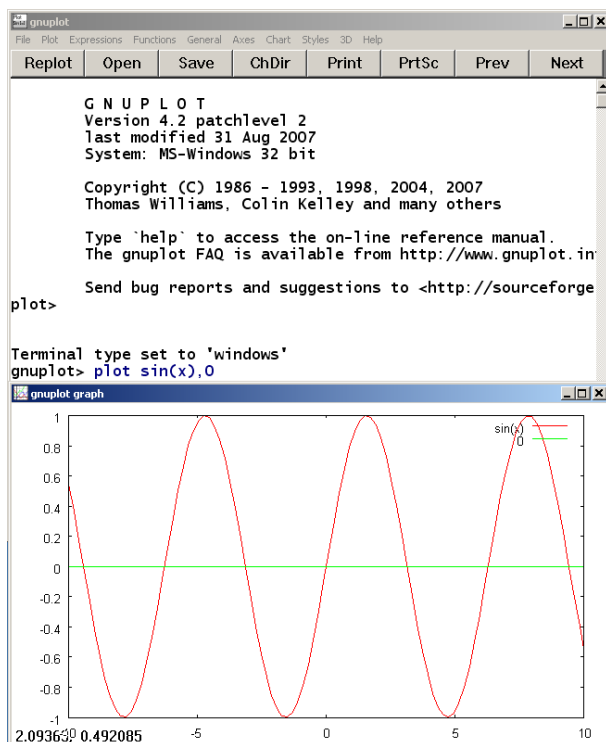
Gnuplot je program za risanje grafov in prikaz meritev, namenjen za uporabo tako v raziskovalnem okolju kot v šoli. Program je prosto dosegljiv in teče na vseh mogočih platformah. Na razpolago je izvorna koda, ki jo nače-

loma prevedemo za izbrani operacijski sistem, za najbolj razširjene operacijske sisteme pa dobimo program v že prevedeni obliki. Podrobnejše informacije o programu najdemo na domači strani <http://www.gnuplot.info/>

2 Namestitev programa na različnih platformah

2.1 MS Windows

Na <http://www.gnuplot.info/>, kjer najdemo številne koristne informacije o programu, izberemo **Download** in tu **gnuplot**. Trenutna verzija programa je 4.2.6. Za delo pod MS Windows je zanimiv predvsem paket `gp426win32.zip` (številka 426 pomeni trenutno verzijo; pri naslednjih verzijah bo seveda drugačna). Lahko tudi prenesemo precej obsežna navodila na 205 straneh `gnuplot-4.2.6.pdf`. Drugi paketi so za običajnega uporabnika manj zanimivi.



Paket odpremo in ga namestimo, recimo na `c:\`. Program ustvari mapo `gnuplot` s pod-

mapami `bin, ...` V podmapii `bin` je več programov, za nas je uporaben osnovni program `wgnuplot.exe`. Program požene in na zaslonu se pojavi okno z komandnim promptom `gnuplot>` in z meniji na vrhu. Z desnim klikom na miško lahko izberemo nabor črk, ki je najbolj primeren za naš zaslon.

Preskusimo, če deluje. Natipkajmo

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

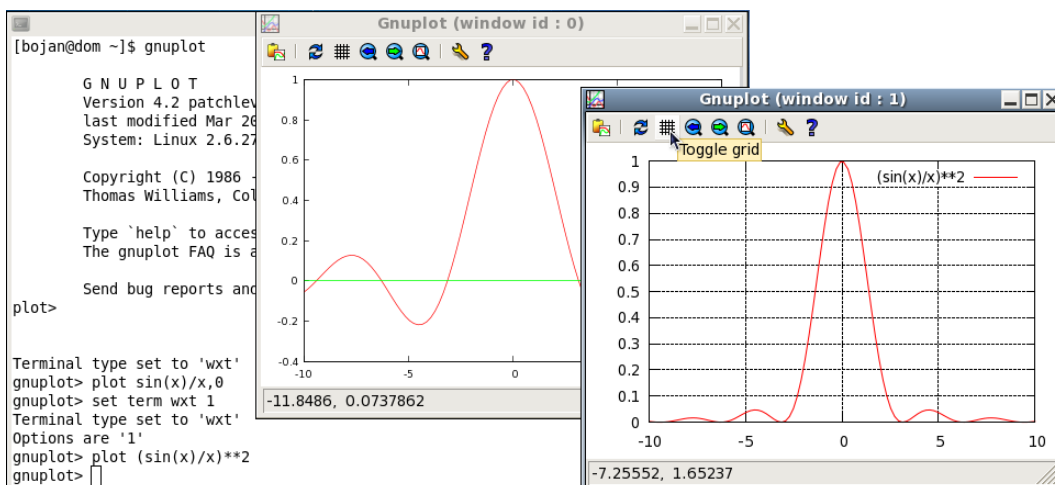
Če je program pravilno nameščen, se odpre okno s grafom sinusne krivulje (glej sliko). Vidimo, da je program sam izbral primeren interval na vodoravni in navpični osi. Pomoč dobimo v meniju ali z ukazom

```
gnuplot>help
```

2.2 Namestitev programa na drugih operacijskih sistemih

Podrobnejša navodila o namestitvah dobimo na <http://www.gnuplot.info/>. Gnuplot je vključen v standardne distribucije Linuxa. V distribucijah s paketi `rpm` lahko preverimo, če je program sploh nameščen, z `rpm -q gnuplot`. Pokličemo ga v terminalskem oknu z ukazom `gnuplot`. V tem okolju je mogoče delati hkrati z več grafičnimi okni, ki omogočajo tudi interaktivno poseganje v sliko (glej sliko).

Seveda obstaja tudi na Macu; več koristnih informacij dobimo na <http://lee-phillips.org/info/Macintosh/gnuplot.html>.



3 Gnuplot na hitro

3.1 Risanje funkcij

Gnuplot je nadvse primeren za risanje funkcij, saj lahko na enostaven način vnesemo različne funkcije, spreminjamo parametre funkcij in parametre slike. Kot smo videli pri preprostem zgledu v prejšnjem poglavju, program sam izbere smiseln interval, v katerem nariše funkcijo. Interval lahko po želji poljubno spreminjamo, prav tako lahko spreminjamo velikost slike v vodoravni in navpični smeri, spreminjamo barvo in debelino črt, graf opremimo z oznakami, ... Sliko funkcije lahko enostavno izvozimo v različne grafične formate in vključimo v dokument.

Ukaze, ki smo jih že vtipkali, lahko prikličemo s tipko \uparrow , sliko, ki smo jo nazadnje narisali z ukazom `plot` pa z `gnuplot>replot`

V vrstici lahko napišemo več ukazov. Ločimo jih s podpičjem.

3.1.1 Sprememba intervala na abscisi in ordinati

Interval na abscisi spremenimo z

```
set xrange[0:4]
```

Funkcijo $y(x)$ bo program prikazal v intervalu $0 \leq x \leq 4$. Na ordinati program izbere tolikšen interval, da zajame vso funkcijo, razen če funkcija v kateri od točk intervala ne divergira. Avtomatsko izbiro spremenimo na primer z

```
set yrange[-1:1]
```

Avtomatsko izbiro intervala ponovno dobimo z ukazom

```
set autoscale
```

samo za os x pa z

```
set autoscale x
```

3.1.2 Velikost slike in razmerje višine in širine

Velikost slike spreminjamo z ukazi oblike

```
set size r
```

ali

```
set size r_x, r_y
```

pri čemer je r faktor, s katerim povečamo ($r > 1$) ali zmanjšamo ($r < 1$) velikost slike. V drugem primeru navedemo ločena faktorja za smer x in smer y . Oznake pri tem ohranijo velikost. Če nas torej motijo premajhne oznake na oseh, zmanjšamo razmerje. Z ukazom

```
set size square
```

dosežemo, da ima okvir slike obliko kvadrata.

Z izbiro

```
set size 0.6,0.9
```

pa dosežemo, da je navpična stranica daljša od vodoravne.

3.1.3 Barva in debelina črte

Če rišemo več krivulj na istem grafu, bo prva rdeča, naslednja zelena, tretja modra, ... Izbiro barve lahko spremenimo z določilom `lt n`, $-1 \leq n \leq 16$ (-1 je črna). Prav tako lahko spremenimo debelino črte z določilom `lw d`, $d = 1$ je privzeta velikost, $d < 1$ pomeni tanjšo, $d > 0$ pa debelejšo krivuljo. Zgled:

```
plot sin(x) lt 3 lw 1.7
```

krivulja bo modra in 1,7 krat debelejša od privzete.

Če pri izpisu slike na datoteko (glej nadaljevanje) zahtevamo, da naj bo slika črno-bela, program barve spremeni v različno črtkane krivulje.

3.1.4 Oznake

Osi označimo z ukazoma `set xlabel` in `set ylabel`, na primer

```
set xlabel "t [s]"
```

```
set ylabel "s [m]"
```

Gnuplot napiše legendo, ki pojasnjuje pomen krivulj ali točk na sliki, v desni zgornji kot slike. Oznaka je kar enaka imenu funkcije oziroma imenu datoteke s podatki. Oznako spremenimo z določilom `t`, na primer `t'meritve'`.

Če legende ne želimo, zahtevamo

```
set key off
```

Legendo prestavimo iz zgornjega desnega kot v točko s koordinatama x, y z ukazom

```
set key x,y
```

3.1.5 Zgled

Kako podamo funkcijo, si oglejmo kar na primerih funkcij, ki jih najbolj pogosto srečamo v fiziki.

Parabolo narišemo z ukazom

```
plot x**2 - 2*x + 1
```

Znak za množenje je torej `*`, za potenciranje pa `**`. Funkcijo lahko podamo v parametrični obliki:

$$y(x) = a*x**2 - b*x + c$$

in parametre podamo posebej:

```
a=1; b=-2; c=1
```

```
plot y(x)
```

Vrednosti parametrov a , b , c lahko spremenimo.

Eksponentno funkcijo dobimo z `exp(x)`

Primeri:

$$y(x) = a*x**2 + b*x + c$$

$$z(x) = d*\exp(-k*x)$$

```
set xrange[0:2]
```

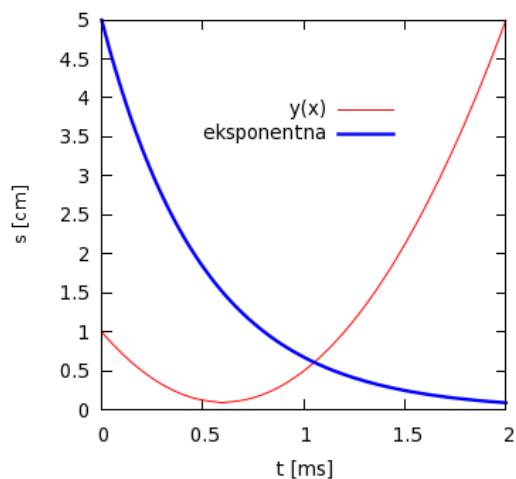
```
set size 0.6,0.8
```

```
set xlabel "t [ms]"; set ylabel "s [cm]"
```

```
a=2.5; b=-3; c=1; d=5; k=2
```

```
set key 1.5,4
```

```
plot y(x), z(x) lt 3 lw 2.4 t"eksponentna"
```



3.1.6 Preglednica nekaterih predefiniranih funkcij

Najbolj pogoste so:

abs(x)	absolutna vrednost
sqrt(x)	kvadratni koren
exp(x)	eksponentna funkcija
log(x)	logaritem z osnovo e
sin(x)	sinus, x je v radianih
cos(x)	kosinus, x je v radianih
tan(x)	tangens, x je v radianih
pi	3,1415926...

3.2 Meritve

3.2.1 Prikaz izmerjenih količin

Rezultate meritev, ki smo jih zbrali na datoteki, recimo:

```
t[s]    x1[m]
2.050   0.478
2.100   0.493
2.150   0.507
2.200   0.521
2.250   0.534
2.300   0.546
2.350   0.557
2.400   0.565
2.450   0.572
2.500   0.578
...
```

na direktoriju (mapi) `c:/meritve`, prikažemo v grafu, tako da ukažemo:

```
cd "c:/meritve"
plot "data.dat" us 1:2
```

S prvim ukazom smo se postavili v izbran direktorij (mapo), z drugim pa zahtevamo, da so na grafu na abscisi vrednosti v prvem stolpcu, na ordinati pa na drugem. Velikost oznake izberemo z določilom `ps`, $p = 1$ je privzeta izbira.

Če želimo točke povezati s črto, dodamo

```
plot "data.dat" us 1:2 w li
```

3.2.2 Izbira znaka za točke

Obliko znaka izberemo z določilom

```
with points n
```

($n = 0$ točka, 1 križec, 4 kvadrateg, 6 krogec, 8 in 10 trikotnik, 12 romb, ...), velikost pa z

```
set pointsize p
```

p določa relativno velikost znaka ($p = 1$ je privzeta izbira). Zgornje ukaze lahko skrajšamo na primer z

```
plot "data.dat" us 1:2 pt 6 lt -1 ps 1.5
```

pri tem z določilom `pt` izberem znak (krožec), z določilom `lt` barvo (črno) in z določilom `ps` velikost znaka (krožec je za faktor 1,5 večji od privzete vrednosti).

3.2.3 Prilagajanje premice

Skozi izmerjene točke pogosto želimo potegniti krivuljo in parametre v krivulji določiti tako, da bo ujemanje čim boljše.

Za zgled najprej določimo najbolj optimalno premico. Bolj zahtevne zglede obravnavamo v poglavju 5. Definiramo

$$y(x) = k \cdot x + n$$

Parametra k in n optimiziramo z ukazom

```
fit [2:3] y(x) 'data.dat' us 1:2 via k,n
```

V oglatem oklepaju smo navedli interval, znotraj katerega naj prilagaja premico, `data.dat` je ime datoteke s podatki, `us 1:2` pomeni, da je neodvisna spremenljivka x v prvem stolpcu, v drugem pa odvisna spremenljivka. Določilo `via k,n` pa pove, katere parametre naj optimizira. Če pri tem navedemo manj parametrov, kot jih je v funkciji, moramo vrednosti za preostale predpisati.

Če program uspe najti optimalni nabor parametrov, dobimo sporočilo, ki se konča nekako takole:

```
Final set of paramtrs  Asympt Standard Err
=====
k          = 0.143008  +/- 0.008222 (5.75%)
n          = 0.207706  +/- 0.0209  (10.06%)
```

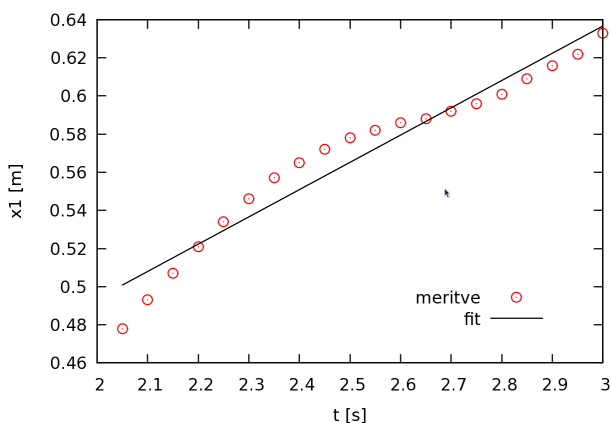
correlation matrix of the fit parameters:

```
          k          n
k          1.000
n         -0.994    1.000
```

Dobimo vrednosti iskanih parametrov in njihove napake ter korelacijsko matriko. Če sedaj ukažemo

```
set key 2.9, .5
set xlabel "t [s]"; set ylabel "x1 [m]"
plot "data.dat" us 1:2 t'meritve' pt 6
      ps 1.3, y(x) lt-1 t'fit'
```

program nariše izmerjene točke in optimalno premico.



3.3 Izpis slike na datoteko

Če želimo sliko shraniti na datoteko in kasneje vključiti v besedilo, imamo na voljo številne formatne zapise, recimo PNG, GIF, JPEG, EPS, ... Vse možnosti dobimo z ukazom `help terminal`, a najbolj pogosto bomo uporabljali format PNG ali GIF, format EPS pa je namenjen za slike v znanstvenih in strokovnih revijah.

Če želimo sliko shraniti v formatu PNG v nekoliko večji resoluciji, na začetku zahtevamo:

```
set terminal png large
```

Sedaj slike na ekranu ne bomo videli neposredno, zato je dobro, da določimo vse parametre slike, ko sliko še vidimo na ekranu. Nato povemo ime datoteke, na kateri naj bo shranjena slika:

```
set output "slika.png"
```

ter ponovimo ukaz `plot`:

```
plot "data.dat" us 1:2 ...
```

Oglejmo si sliko, ki smo jo dobili. Prav lahko se zgodi, da debelina črt ni takšna, kot smo jo videli na terminalu, tudi barva je lahko drugačna. V tem primeru ponovimo ukaza `set output` in `plot ...` in v ukazu `plot` izberemo drugačno debelino ali barvo črte ali znaka.

V zaslonski način se v operacijskem sistemu MS Windows vrnemo z ukazom:

```
set terminal windows
```

v UNIXu pa z:

```
set terminal x11
```

ali

```
set terminal wxt
```

Določilo `terminal` lahko okrajšamo v `term`. V poglavju 6 so navedene dodatne izbire, ki jih navedemo za ukazom `set terminal ime_terminala`, in s katerimi lahko spremenimo nabor znakov ali dobimo dodatne matematične znake.

3.4 Zaključek

Program končamo z ukazom `exit` ali `quit`. Pred tem lahko ukaze shranimo in jih pri naslednjem zagonu programa ponovno naložimo. Ukaz ima obliko

```
save {<izbire>} '<ime datoteke>'
```

{<izbire>} so lahko `functions` (tj. definicije funkcij), `variables` (vrednosti spremenljivk), `terminal` ali `set` (v tem primeru shrani vse ukaz `set`, tudi tiste, ki jih je sam postavil). Če izbir ne navedemo, program shrani funkcije, spremenljivke, izbire in zadnji ukaz `plot`.

Pri ponovnem zagonu lahko stare podatke preberemo z

```
load '<ime datoteke>'
```

4 Pregled nekaterih ukazov za risanje funkcij

4.1 Krivulje v 2D

4.1.1 Resolucija slike

Pri risanju funkcij, katerih vrednost se hitro spreminja, je ugodno povečati število točk na sliki. To storimo z ukazom

```
set sample n
```

pri čemer je n dovolj veliko število, recimo $n = 1000$.

4.1.2 Odsekoma definirane krivulje

Navpične črte in krivulje, katerih definicija se spreminja v različnih intervalih, rišemo z ukazom, ki ima sintakso (A ? B : C). Ukaz B se izvrši, če je pogoj A izpolnjen, sicer se izvrši ukaz C. Primer

```
H(x)=(x<4 ? 0 : 1)
```

je funkcija, ki ima za $x < 4$ vrednost 0 in za $x > 4$ vrednost 1.

Ukaz lahko gnezdimo in stopničasto funkcijo narišemo takole

```
h(x)=(x<2 ? 0 : (x<4 ? 1 : (x<6 ? 2 : 3)))
```

Tretji primer je odvisnost faze od frekvence pri vsiljenem nihanju:

$$\delta(\omega) = \arctan \frac{2\beta\omega}{\omega_0^2 - \omega^2}$$

Argument funkcije arkus tangens (atan) je negativen za $\omega > \omega_0$, zato funkcija v $\omega = \omega_0$ preskoči od vrednosti $\pi/2$ na vrednost $-\pi/2$. Zvezen potek dobimo tako, da na intervalu $\omega > \omega_0$ funkciji prištejemo π :

```
del(x)=atan(2*b*x/(om0**2-x**2))
```

```
f(x)=(x<om0 ? del(x) : del(x)+pi)
```

pri čemer za neodvisno spremenljivko ω pišemo x , $om0$ je ω_0 in b .

Odvisnost amplitude od frekvence pri vsiljenem nihanju:

$$s(\omega) = s_0 \frac{\omega^2}{\sqrt{(\omega_0^2 - \omega^2)^2 + 4\beta^2\omega^2}}$$

zapišemo kot

```
s(x)=s0*om0**2/sqrt((om0**2-x**2)**2+
4*b**2*x**2)
```

z s_0 smo označili odmik pri frekvenci 0.

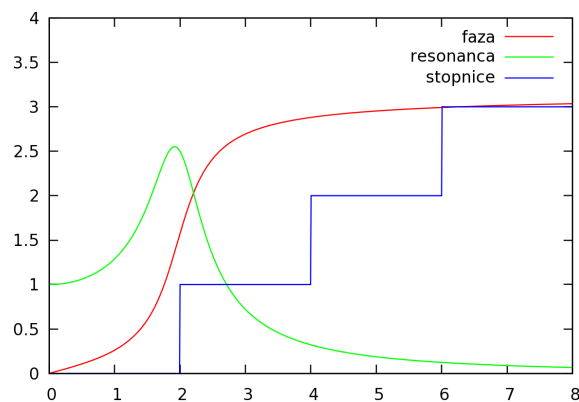
Funkcije $h(x)$, $f(x)$ in $s(x)$ narišemo z ukazi

```
set sample 1000
```

```
set xrange[0:8]; set yrange[0:4]
```

```
s0=1; om0=2; b=0.4
```

```
plot f(x) t"faza", s(x) t"resonanca",
h(x) t"stopnice"
```



4.1.3 Oznake na oseh

Gnuplot sam izbere številčenje na oseh in razmike, v katerih postavlja črtice. Če želimo to izbiro spremeniti, uporabimo ukaz `set xticks Δx` in `set yticks Δy` . Z

```
set xtics 0.1
```

tako zahtevamo, da so črtice na osi x v razmiku 0.1 enote. Če osi y ne želimo oštevilčiti, zahtevamo

```
set format y""
```

Če želimo imeti os y tudi brez črtic pa

```
unset ytics
```

Logaritemsko skalo na izbrani osi, na primer y , dobimo z `set logscale y`, **mrežo** pa z ukazom `set grid`. (Glej zgled pri razdelku 6.4.1.)

Kot smo že povedali, osi označimo z ukazoma `xlabel` in `ylabel`. Poljuben napis v sliko

vstavimo na primer z ukazom
`set label "Legenda" at 3,1 center font`
`"Helvetica,16"`

ki postavi (središče) napis Legenda na koordinato (3,1), nabor znakov je Helvetica v velikosti 16 pt. Če namesto določila uporabimo `left` (`right`) bo na tej kordinati začetek (konec) napis.

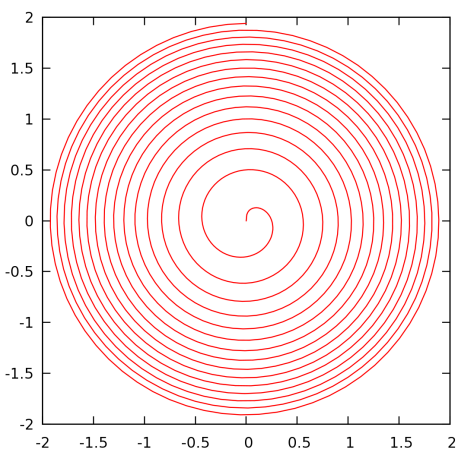
4.2 Parametrično podane krivulje

V parametrični obliki podamo krivuljo tako, da povemo, kako se koordinati x in y spreminjata v odvisnosti od parametra t . Primer:

$$x(t) = a \sin t, \quad y(t) = a \sin t,$$

je krožnica z radijem a , ko t preteče vrednosti od 0 do 2π . Če dopustimo, da radij narašča s parametrom t , dobimo spiralo:

```
set parametric
set size square; set sample 1000
set trange[0:30*pi]; set key off; a=0.2
plot a*sqrt(t)*sin(t), a*sqrt(t)*cos(t)
```



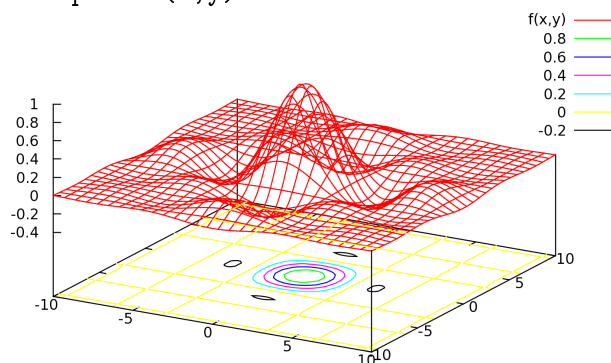
4.3 Grafi v 3D

Funkcija ima dve neodvisni spremenljivki x in y , recimo

$$f(x,y) = \frac{\sin x}{x} \frac{\sin y}{y}$$

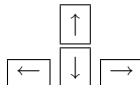
Narišemo jo z ukazom

```
f(x,y) = sin(x)*sin(y)/x/y
set isosamples 30,30
splot f(x,y)
```



Z ukazom `set isosamples 30,30` smo povedali, s koliko črtami v smeri x in smeri y naj opazori funkcijo. Konturo z izohipsami na spodnji ravnini smo dobili z zahtevo

```
set contour
```

Sliko na zaslonu vrtimo s tipkami 

4.4 Preglednica manj pogostih funkcij

<code>log10(x)</code>	logaritem z osnovo 10
<code>asin(x)</code>	arkus sinus
<code>acos(x)</code>	arkus kosinus
<code>atan(x)</code>	arkus tangens
<code>sinh(x)</code>	hiperbolični sinus
<code>cosh(x)</code>	hiperbolični kosinus
<code>tanh(x)</code>	hiperbolični tangens
<code>norm(x)</code>	Gaussova porazdelitev
<code>invnorm(x)</code>	inverzna Gaussova porazdelitev
<code>erf(x)</code>	funkcija napak
<code>inverf(x)</code>	inverzna funkcija napak
<code>rand(x)</code>	generator naključnih števil

in še:

`acosh`, `arg`, `asinh`, `atan2`, `atanh`, `besj0`, `besj1`, `besy0`, `besy1`, `ceil`, `cosh`, `erfc`, `floor`, `gamma`, `ibeta`, `inverf`, `igamma`, `imag`, `int`, `lambertw`, `lgamma`, `real`, `sgn`, `sinh`, `tanh`, `column`, `defined`, `tm_hour`, `tm_mday`, `tm_min`, `tm_mon`, `tm_sec`, `tm_wday`, `tm_yday`, `tm_year`, `valid`.

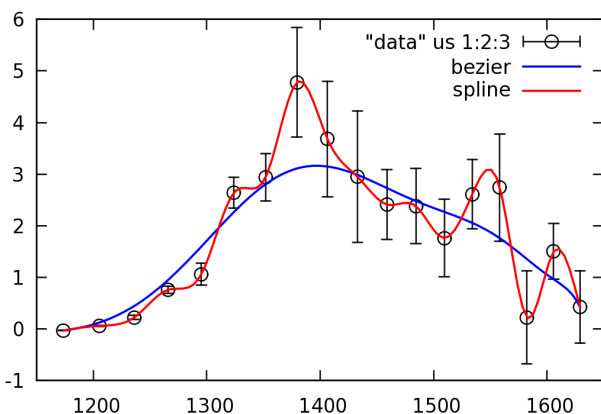
5 Predstavitev in analiza meritev

5.1 Grafični prikaz meritev z odstopanji

Pri resnejših meritvah poleg samega izmerka navedemo tudi odstopanje (napako). V datoteki zapišemo napake v posebnem stolpcu. Program jih nariše skupaj z izmerki z ukazom

```
plot "data" us 1:2:3 w err pt 6 lt -1
    ps 1.5
```

če je v prvem stolpcu datoteke data.dat neodvisna spremenljivka, v drugem odvisna in v tretjem napaka odvisne spremenljivke.



5.2 Gladka krivulja skozi izmerke

Če želimo skozi izmerjene točke potegniti gladko krivuljo uporabimo določilo smooth. Izbiramo lahko med enostavno povezavo med točkami (unique), gladko krivuljo iz zlepkov skozi točke (cspline) ali Bezierovo krivuljo (bezier). Krivulje na sliki v prejšnjem razdelku smo dobili z

```
plot "data" us 1:2:3 smooth bezier,
    "data" us 1:2:3 smooth cspline
```

Program pri tem predpostavlja, da so podatki opremljeni z napako v tretjem stolpcu. Če imamo podatke brez ocene napake, napišemo

```
plot "data.dat" us 1:2:(1) smooth ...
```

Z (1) povemo, da napaka 1 velja za vse izmerke.

5.3 Računske operacije na in med stolpci v tabeli

Pogosto želimo prikazati nekoliko spremenjene izmerke, poiskati zvezo med količinami v tabeli, ... To lahko storimo znotraj programa, vendar le v okviru podatkov v isti vrstici.

Če imamo podatke x in y v prvem in drugem stolpcu, prikažemo odvisnost $y(x^2)$ (namesto $y(x)$) z ukazom

```
plot "data" us ($1**2):2 ...
```

(... so druga določila). Stolpce, nad katerimi izvajamo operacijo, moramo torej dati v oklepaj, \$1, \$2, \$3, ... pa so oznake stolpcev v računskih operacijah.

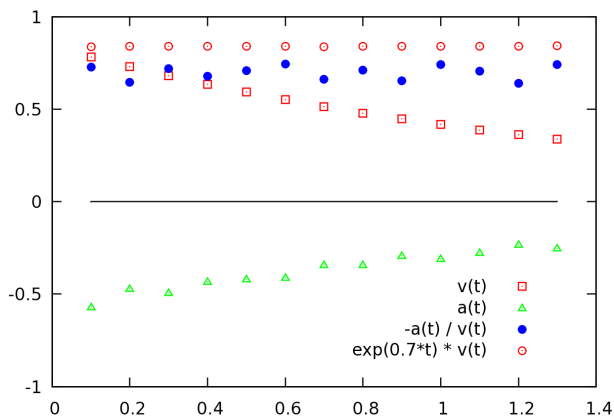
Pri zgledu imamo v datoteki data izmerke hitrosti in pospeškov pri pojemajočem gibanju, za katerega domnevamo, da je eksponentno pojemajoče. V tem primeru bi veljalo

$$v(t) = v_0 e^{-\beta t}, \quad a(t) = -\beta v_0 e^{-\beta t}, \quad -\frac{a(t)}{v(t)} = \beta.$$

Če je v prvem stolpcu čas, v tretjem hitrost in v četrtem pospešek, lahko preverimo, če zveza med pospeškom in hitrostjo res velja, tako da narišemo odvisnost razmerja od časa (polni modri krogi):

```
plot "data" us 1:(-$4/$3) pt 7 lt 3
```

Vidimo, da je razmerje (v okviru napak) konstantno in enako 0,7 (tj. $\beta = 0,7$).



Nato se stolpec hitrosti pomnožimo z $e^{\beta t}$ (rdeči krožci):

plot "data" us 1:(exp(0.7*\$1)*\$3) pt 6
in vidimo, da je rezultat res konstanta in je začetna hitrost v_0 enaka 0,84.

5.4 Prilagajanje krivulje skozi izmerjene točke

V poglavju 3.2.3 s prilagajanjem premice, nismo dobili zadovoljivega ujemanja z izmerjenimi podatki. S polinomo 3. stopnje dobimo boljše prilagajanje:

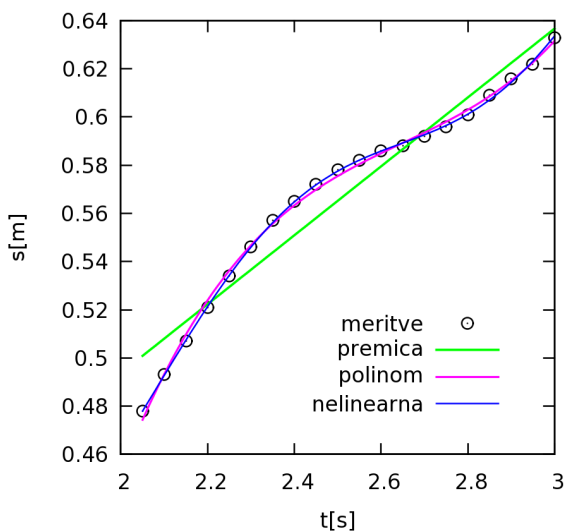
```
p(x)=a*x**3 + b*x**2 +c*x + d
fit p(x) 'data.dat' us 1:2 via a,b,c,d
```

Še boljše prilagajanje lahko dobimo s funkcijami, v katerih se parametri pojavijo *nelinearno*, na primer:

$$f(t) = x_0 + vt + u \cos(\omega t + \delta)$$

V takšnih primerih je potrebno, da navedemo približno začetno vrednost vsaj nekaterih parametrov; v našem primeru lahko za v in x_0 vzamemo kar vrednosti k in n , ki smo ju določili pri prilagajanju premice:

```
f(x) = x0 + v*x + u*cos(om*x+del)
v=k; x0=n
fit f(x) 'data.dat' us 1:2 via
x0,v,u,om,del
```



Končni rezultat našega zgled izpišemo z ukazi:

```
set xlabel "t[s]"; set ylabel "s[m]"
set size square; set size 0.8
set xtics 0.2; set key 2.9,.52
plot "data.dat" us 1:2 pt 6 lt-1 ps 1.2
t'meritve',y(x) lw 1.5 t'premica',p(x)
lt 4 lw 1.3 t'polinom',f(x) lt 3
t'nelinearna'
```

5.5 Prilagajanje z upoštevanjem mer-skih napak

Če poznamo napake izmerkov, je smiselna zahteva, da je odstopanje krivulje od izmerkov manjše v primeru, ko je napaka izmerkov manjša, in večje pri večjih napakah. Značilna situacija je v primeru, prikazanem v razdelku 5.1. V programu zahtevamo, da so točke utežene z inverzno vrednostjo napake, tako da v ukazu fit za določilom us navedemo še tretji stolpec, v katerem so napake (paziti moramo, da ni katera od napak enaka 0).

V primeru pri 5.1 iščemo teoretično odvisnost s splošno obliko resonančne krivulje

$$s(W) = A \frac{\Gamma + D(W - M)}{\sqrt{(W - M)^2 + (\Gamma + D(W - M))^2}}$$

pri čemer je W neodvisna spremenljivka, določiti pa moramo parametre A , Γ , D in M . Ker gre za nelinearen primer, je ugodno, da podamo približek za nekatere parametre. Vrh resonance je približno pri M , višina pri tej vrednosti pa ustreza parametru A .

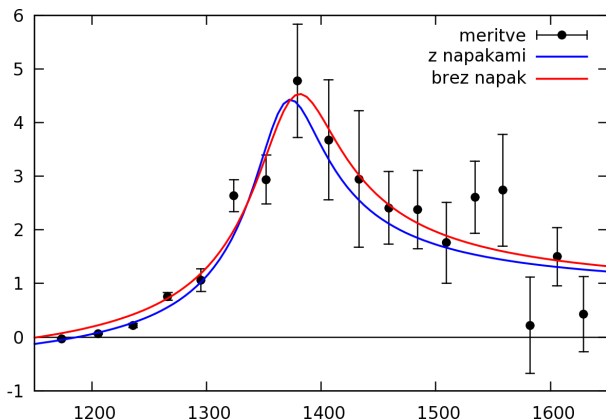
$$s(x) = A * (G + D * (x - M)) / \sqrt{(x - M)^2 + (G + D * (x - M))^2}$$

$M=1400$; $A=5$

```
fit [1170:1650] s(x) "data" us 1:2:3
via M,G,D,A
```

Vidimo, da se pri manjših vrednostih neodvisne spremenljivke, kjer je napaka manjša, teoretična krivulja v tem primeru (modra) veliko bolj prilega izmerjenim vrednostim kot krivu-

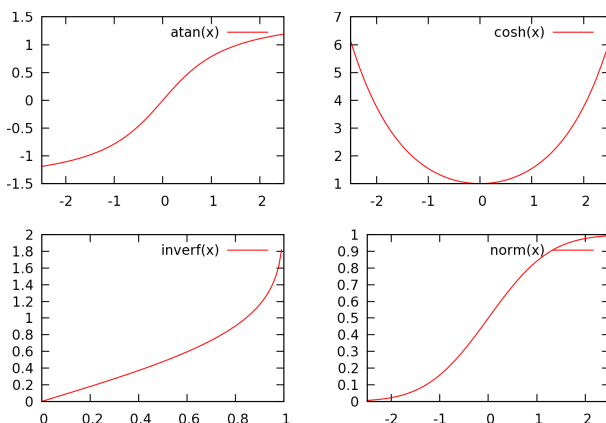
lja, pri kateri so vse točke enakovredne (rdeča):



5.6 Več slike v istem okviru

Več slik lahko razporedimo znotraj istega okvira z ukazom `set multiplot`, izbiro velikosti posamezne slike `set size` in izbiro koordinatnega izhodišča `set origin`. Koordinatno izhodišče (0,0) okvira je levo spodaj, stranici sta dolgi eno enoto. Navedimo kar zgled

```
set multiplot
set size 0.5,0.5
set xrange[-2.5:2.5]
set origin 0.0,0.5; plot atan(x)
set origin 0.5,0.5; plot cosh(x)
set origin 0.5,0.0; plot norm(x)
set xrange[0:1]
set origin 0.0,0.0; plot inverf(x)
unset multiplot
```



6 Terminalske izbire

Številni terminali omogočajo dodatne izbire, predvsem vključitev dodatnih naborov znakov in simbolov. Navedimo le nekatere, ki jih najpogosteje uporabljamo.

6.1 Terminal windows

V operacijskem sistemu MSWindows dobimo sliko v oknu na zaslonu z ukazom

```
set term windows izbire
```

Med *izbirami* omenimo predvsem možnost `enhanced`, ki jo podrobneje predstavimo v nadaljevanju. Izbiramo lahko med barvno (`color`, privzeta izbira) in črno-belo (`monochrome`) sliko. Z določilom `font` izberemo drugačen nabor znakov. Katere možnosti imamo na voljo, je odvisno od naborov znakov, ki so implementirani v sistemu. Zgled

```
set term windows enhanced font "Arial,14"
set format x "%g^o"
set xlabel "{/Symbol J}"
set ylabel "I [{/Symbol m}A]"
...
```

Zahtevali smo razširjeno (`enhanced`) izbiro in nabor znakov `Arial` na velikosti 14 pt. Za ukazom `set format x` smo na osi x ustvarili skalo v kotnih stopinjah, za oznako kota pa uporabili ϑ . Oznaka na osi y bo I [μA].

Sliko izpišemo z ukazom

```
set output "prn"
```

6.2 Terminali wxt in x11

V operacijskih sistemih UNIX sta na voljo dva zaslonska terminala, s tem da novejši `wxt` nudi več dodatnih možnosti. Poleg izbir, ki smo jih navedli pri terminalu `windows`, lahko delamo hkrati z več terminali. V tem primeru takoj za ukazom `set term wxt` (ali `X11`) navedemo številko terminala. (Glej zgled na začetku.)

6.3 Izpis v formatu PostScript

Fizikalne znanstvene revije običajno zahtevajo, da so slike pripravljene v formatu EPS (Encapsulated PostScript). Za razliko od običajnega formata PostScript, pri katerem slika zavzame celo stran, je mogoče sliko v formatu EPS enostavno vključiti v dokument.

Osnovni ukaz ima obliko

```
set term postscript izbire
```

Kot *izbire* lahko navedemo

```
{landscape | portrait | eps}
```

kar pomeni, da lahko izbiramo med ležečim (landscape, privzeto), pokončnim (portrait) in formatom EPS. Slika je črnobela, namesto črt različnih barv dobimo različno črtkane črte. Izbire, ki jih dodajamo: za barvno sliko

```
color
```

za polne (neprekinjene) črte:

```
solid
```

(ta izbira je smiselna pri barvni sliki), razmik med črticami pri črtkanih črtah lahko spremenimo z

```
dl s
```

pri čemer je $s = 1$ privzeta dolžina črtice, debelino črt okvirja z

```
lw d
```

$d = 1$ je privzeta izbira. Posebne znake (glej naslednji razdelek) dobimo z

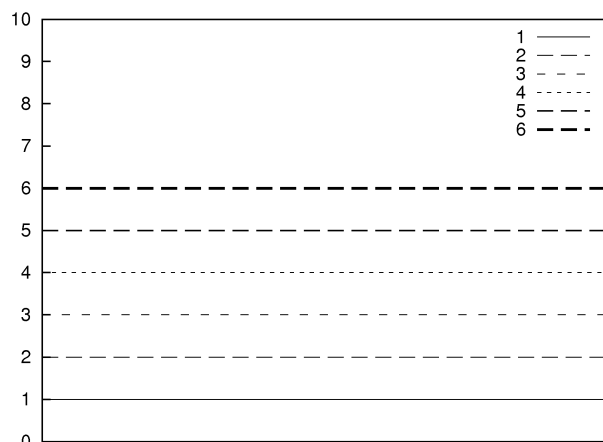
```
enhanced
```

Črte grafov so običajno nekoliko tanjše kot v terminalskem oknu; njihovo dbelino lahko spremenimo z izbiro $lw\ d$ ($d = 1$ je privzeta debelina) v ukazu plot. Zgled:

```
set size 0.7; set yrange[0:10]
unset xtics; set ytics 1
set term postscript eps dl 3
set output "crte.eps"
plot 1,2,3,4,5 lt 2 lw 2,6 lt 2 lw 4
```

Z ukazi v prvih dve vrstici smo le definirali okvir slike, v ukazu set term smo zahtevali format EPS in dolžino črtic, ki je 3 krat večja od privzete. Črte pri 1, 2, 3 in 4 ustrezajo prvemu

($lt\ 1$), drugemu, tretjemu in četrtemu tipu črt; črta pri 5 je dvakrat debelejša črta tipa 2, črta pri 6 pa štirikrat debelejša.



6.4 Izbira enhanced

Z opcijo enhanced lahko izbiramo med številnimi nabori znakov (fonti), grškimi črkami in raznimi simboli, pišemo matematične formule ... Vse možnosti najdemo v gradivu, ki ga dobimo ob instalaciji in sicer v poddirektoriju (mapi) postscript-terminal v direktoriju docs. Tu omenimo le najbolj pogoste

6.4.1 Indeksi in eksponenti

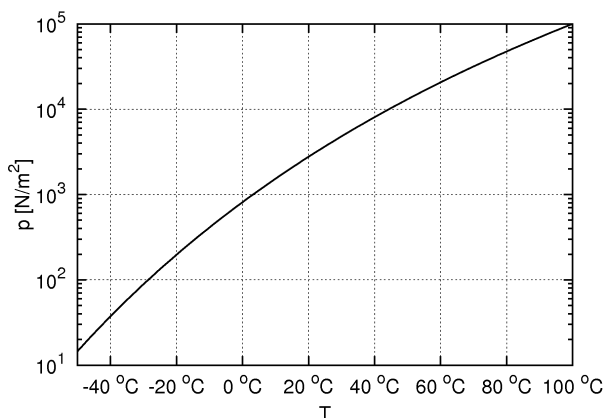
Indekse in ekponente v primeru $a_{i,j}^2$; pišemo kot $a_{\{i,j\}^2}$. Če želimo številke na osi y pisati v eksponentnem zapisu, recimo $2 \cdot 10^3$, zahtevamo

```
set format y "%2.0t{/Symbol\327}10^{-%L}"
```

Pri tem "%2.0t" predstavlja mantiso števila, zapisano brez decimalk (v našem primeru 2), %L pa desetiški eksponent (3). Mantiso lahko tudi izpustimo (kot na zgledu). Če želimo mantiso zapisati z dvema decimalkama, namesto "%2.0t" uporabimo "%4.2t". Zgled:

```
set term postscript eps enhanced
set key off; set size 0.6; set grid
t0=273.; t1=373.; p1=100000.; k=4900.
p(x)=p1*exp(k*(1/t1 - 1/(t0+x)))
```

```
set xlabel "T"; set ylabel "p [N/m^2]"
set format x "%g ^oC"
set format y "10^{%L}"
set logscale y; set xrange[-50:100]
set output "ClausClap.eps"
plot p(x) lt -1
```



V zgladu `%g` pomeni običajni zapis števila na skali. Če želimo recimo na osi x zapisati kotne stopinje, uporabimo ukaz `set format x "%g^o"`

Če bi želeli na našem zgladu izpisati oznake na osi y tudi pri vmesnih vrednostih (kar je smiselno, če razpon prednosti ni tako velik kot pri našem zgladu), recimo pri $2 \cdot 10^n$ in $5 \cdot 10^n$, bi morali vrednosti eksplicitno navesti z ukazom `set ytics`

```
set yrange [10:2000]
set ytics (10,20,50,100,200,500,1000,2000)
set format y "%2.0t{/Symbol \327}10^{%L}"
```

6.4.2 Grške črke in posebni znaki

Grške črke dobimo kot `{/Symbol a}` (za α), druge grške črke pa tako, da namesti a pišemo:

a	α	b	β	c	χ	d	δ
D	Δ	e	ϵ	f	ϕ	F	Φ
g	γ	G	Γ	h	η	k	κ
l	λ	m	μ	n	ν	p	π
q	θ	r	ρ	s	σ	S	Σ
t	τ	w	ω	x	ξ	y	ψ
Y	Ψ	z	ζ	j	φ	J	ϑ

Nekatere posebne znake dobimo tako, da navedemo zaporedno številko znaka, recimo `{/Symbol \240}` je znak za Euro. Kompletne table je v originalni dokumentaciji v datoteki `ps_guide.ps`, najbolj pogosti znaki pa so

245	∞	260	\circ	261	\pm	264	\times
265	α	266	∂	273	\approx	301	\Im
302	\Re	321	∇	327	\cdot	345	Σ

7 Literatura

- Uradna stran: <http://www.gnuplot.info/>
- Priročnik s številnimi zgledi: <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>
- Zgledi: <http://gnuplot.sourceforge.net/demo/>
- Vprašanja: <http://www.gnuplot.info/faq/faq.html>
- Kratek priročnik: <http://www.duke.edu/~hpgavin/gnuplot.html>
- Pregled ukazov in trikov (predvsem za Unix): <http://sparky.rice.edu/gnuplot.html>
- Gnuplot na Macintosh-u: <http://lee-phillips.org/info/Macintosh/gnuplot.html>
- 3D zgledi: <http://ayapin.film.s.dendai.ac.jp/~matuda/Gnuplot/pm3d.html>
- <http://mathewpeet.org/computing/gnuplot/>
- <http://dynamicnetservices.com/~will/gnuplot/>