

Eine Einführung in gnuplot

Julius Hannink

Für den FSR Physik Göttingen

25.02.2010

Ablauf

- 1 Motivation
- 2 Plotten
- 3 "Speichern"
- 4 Fitten
- 5 Was kann gnuplot noch?

Was ist ein Plot bzw. ein Fit?

Plotten

Ein Plot bzw. Diagramm dient der Visualisierung von Messdaten und evtl. dem Vergleich mit einem mathematischen Modell.

Fitten

Fitten heißt, an einen Datensatz ein Modell anpassen. Man variiert dabei die Parameter des analytischen Modells und vergleicht es mit dem Datensatz. Im Grunde wollen wir also eine Kurve an Datenpunkte anpassen.

Warum gnuplot?!

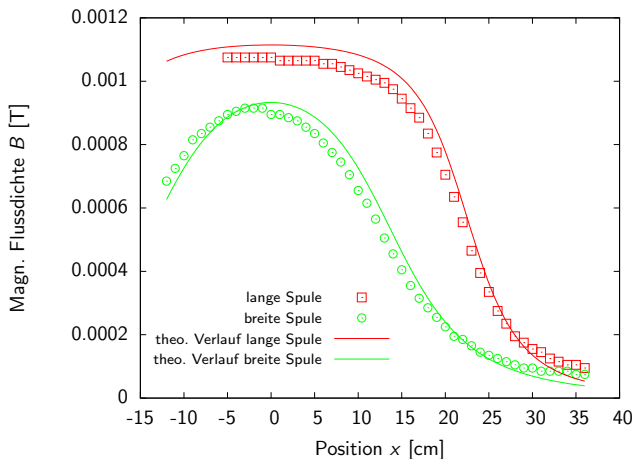
Vorteile:

- Es ist kostenlos für alle Systeme (Linux, Windows,..) verfügbar
- Die Plots sind publikationstauglich
- Viele verschiedene Ausgabeformate
- Lineare und Nichtlineare Fits sind möglich
- Schnittstelle zu C, C++

Nachteile:

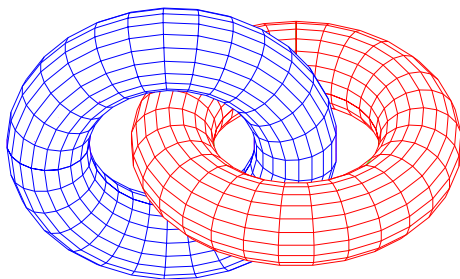
- gnuplot ist kommandogesteuert \Rightarrow Es gibt eine Lernkurve

Beispiel 1: Magnetfelder



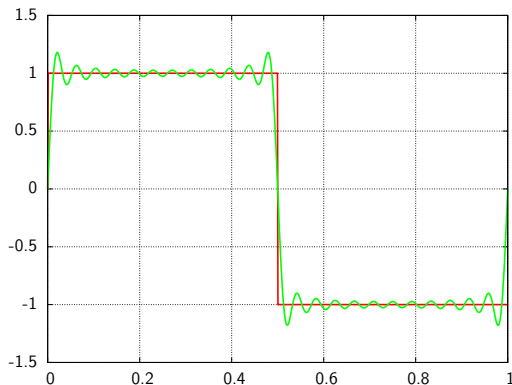
Das Grundpraktikum lebt von Plots..

Beispiel 2: Mathematische Sachverhalte



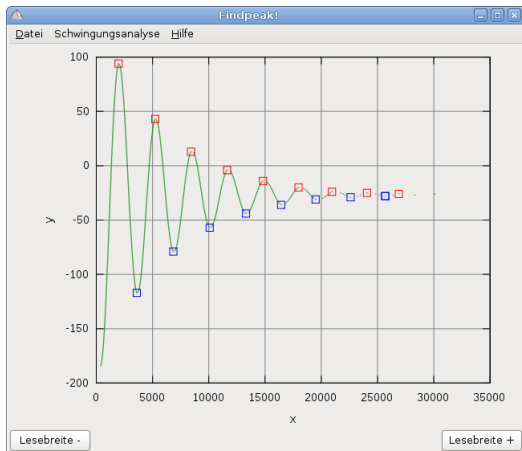
*Torus gefällig?
gnuplot hilft der Anschauung auf die Sprünge..*

Beispiel 3: Animationen - Fourierreihe



Dir fehlt die Anschauung zu Konvergenzfragen? gnuplot hilft!

Beispiel 4: gnuplot in eigenen Programmen



Beim Programmieren hilft gnuplot bei der Visualisierung..

Wie plote ich eine Funktion?

- Plotten mit dem Befehl `plot` (kann mit `p` abgekürzt werden)
- Potenzen in gnuplot:
`f(x)=x**2`
- Funktionen werden durch Kommata getrennt
- `help functions` zeigt alle eingebauten Funktionen wie `sin`, `cos`, ...

Beispiel 1

```
set grid
f(x)=x**2
plot f(x)
```

Beispiel 2

```
g(x)=4*x+2
p f(x),g(x), x+1
```

Achtung!

Den natürlichen Logarithmus $\ln()$ kennt gnuplot als `log()`.

Wie plote ich Daten?

Das Format

- Daten befinden sich in Textdateien
- Zahlenformat: X.XeX (z.B 3.3e4)
- Kommentare mit #
- Spalten werden mit Leertaste oder Tab getrennt
- Zwei Zeilenumbrüche trennen Datensätze (Linie wird nicht druchgezogen)

```
daten.txt
```

```
#Messdaten in Spalten:
```

```
#x y
```

```
1 4.2
```

```
2 4.7
```

```
3 5.2
```

Export-Import

Nazu jedes Tabellenkalkulationsprogramm (z.B. OpenOffice oder gnumeric) unterstützt den Import und Export von Texttabellen!

Wie plote ich Daten?

Der Plot

- Plotten von Daten mit `plot 'dateiname'`
- Geplottet wird die 1. gegen die 2. Spalte
- Messfehler aus Spalte 3 werden mit `with errorbars` geplottet (kurz: `w e`)

daten.txt

```
#x y Messfehler
0 0 0.5
1 1 0.5
2 4 5e-1
3 9 0.5
```

Beispiel

```
#Plotten aus daten.txt:
p 'daten.txt' with errorbars
```

Wie plote ich Daten?

Operationen auf Datensätzen

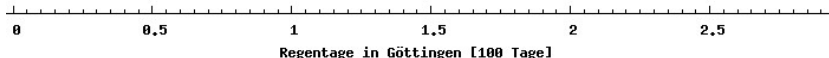
- Sind die Spalten der Textdatei nicht richtig sortiert, hilft `using`
- `using` (kurz: `u`) weist (x, y, y_{err}) den Spalten zu
- Die 3. angegebene Spalte wird stets dem Fehlerbalken zugewiesen
- Mit `using` lassen sich Daten auch umrechnen:
u 1:(log(\$2)) trägt Spalte 1 gegen $\log(\text{Spalte 2})$ auf

Beispiel

```
# Plotten aus daten.txt
set grid
p 'daten.txt' using 1:2:3 w e
#Oder:
p 'daten.txt' using 2:1
```

Beschriftung und Formatierung

Beschriftungen



- Eine Funktion bekommt mit `title '...'` einen Namen
- Eine Achse beschriftet man mit `set xlabel '...'`
- Der ganze Plot wird mit `set title '...'` benannt

Beispiel

```
#Umlaute in Windows:
set encoding iso_8859_1
set title 'Übung'
set xlabel 'x[cm]'
set ylabel 'y[cm]'
p f(x) title 'Testfunktion'
```

Achtung!

Achsenbeschriftungen immer mit Einheiten versehen!!

Beschriftung und Formatierung

Linientyp und Linienbreite

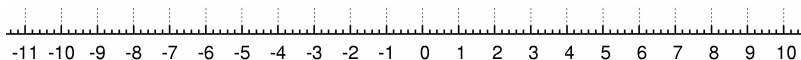
- `linetype x` ändert die Farbe der Linie bzw. des Datenpunktes (kurz: `lt x`)
- Bei Datenplots wird mit `linetype` auch die Form der Punkte geändert
- `linewidth x` ändert die Linienbreite (kurz: `lw x`)
- `test` zeigt eine Übersicht an zur Verfügung stehenden Typen

Beispiel

```
set grid
p f(x) linetype 4 linewidth
1.5
#Oder kurz:
p f(x) lt 4 lw 1.5
```

Beschriftung und Formatierung

Achsenmarkierungen



- Die Abstände der Markierungen und Gitternetzlinien können variiert werden
- Mit `set xtics x.x` setzt man eine neue Intervallbreite
- Die Anzahl(!) an Zwischenmarkierungen wird mit `set mxtics x` gesetzt

Beispiel

```
set grid
set xtics 2.0
set ytics 1.0
set mxtics 2
set mytics 1
plot f(x)
```

Beschriftung und Formatierung

Der Wertebereich

- Mit `set xrange [von:bis]` wird nur dieser Bereich geplottet
- Analog mit dem Wertebereich von `y`
- Alternativ: `p [x_von:x_bis] [y_von:y_bis] f(x)`

Beispiel

```
set xrange[0:10]
set yrange[-5:5]
# Oder:
p [4:5][0:2] f(x)
```

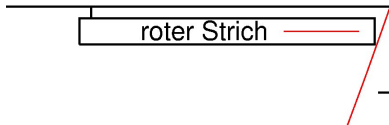

Beschriftung und Formatierung

Die Legende

- Mit `set key` wird die Legende eingeschaltet und positioniert
- Mögliche Positionen: `below`, `top`, `outside`, `right`, ...
(`help key` hilft)
- Weiter Beschriftungen können mit `set label '...'` at `x.x,y.y` gesetzt und positioniert werden

Beispiel

```
set key outside
set label 'Test' at 3,0
p f(x) title 'Testfunktion'
```



Nützliches zwischendurch

- Alle Einstellungen können mit `reset` zurückgesetzt werden
- `unset ...` setzt spezielle Einstellungen zurück bzw. schaltet sie aus
- `replot` aktualisiert den Plot nachdem Einstellungen geändert wurden
- Mit `Strg+Z` schiebt man `gnuplot` in den "Hintergrund" der Konsole, mit `fg` holt man es zurück (unter Linux)
- Direkt aus `gnuplot` kann man mittels `!` Befehle auf der Konsole ausführen z.B. `!cd ../Test`

Die Rahmendatei

- Bei zu vielen, sich wiederholenden Eingaben in die gnuplot-Konsole lohnt sich eine Rahmendatei – eine Textdatei, in der einfach alle Befehle untereinander stehen.
- Rahmendateien werden mit `load 'rahmendatei.plot'` in gnuplot geladen oder beim Start als Parameter übergeben:
`gnuplot rahmendatei.plot`
- Kommentare funktionieren wie bei Datensätzen mit `#`
- Pausen zwischen Befehlen sind mit `pause t` möglich, wobei `t` die Zeit in Sek. angibt. Bei `t=-1` wird auf das Drücken der Enter-Taste gewartet.

Die Rahmendatei

Beispiel

rahmendatei.plot

```
#Alle Einstellungen zurücksetzen:  
reset  
#Titel setzen:  
set title 'Rahmendatei - Beispiel'  
set xlabel 'x[cm]'  
set ylabel 'y[cm]'  
# Datei 'daten.txt' halblog. auftragen:  
p 'daten.txt' u 1:(log($2)) title 'Daten'  
#Auf die Enter-Taste warten:  
pause -1
```

"Speichern"

Was ist ein Terminal?

- Das sog. Terminal entscheidet, was gnuplot mit deinem Plot macht
- Standardmäßig ist `wxt` oder `x11` eingestellt. Beide Terminals werfen dir den Plot direkt auf den Bildschirm
- Andere Terminals schreiben deinen Plot in eine Datei.
Beispiele: `postscript`, `png`, `epslatex`, `svg`
- `set terminal` listet alle vorhandenen Terminals
- `set terminal x` wählt das Terminal (kurz: `set term`)
- `set output 'dateiname'` wählt die Ausgabedatei

"Speichern"

Das Terminal png

- Ein einfaches Terminal ist z.B. png
- Dein Plot wird ein Bild
- Den Dateinamen setzt du mit
`set output 'bild.png'`

Rahmendatei_png.plot

```
reset
#Terminal wählen:
set terminal png
#Datei wählen:
set output 'bild.png'
#Datei 'schreiben':
p 'daten.txt', f(x)
```

"Speichern"

Das Terminal `epslatex` – Unsere Verbindung zu \LaTeX

- `epslatex` plottet Bild und vor allem Beschriftungen in \LaTeX -lesbarer Form
- `help epslatex` gibt Aufschluss über mögliche Optionen, z.B.
`set term epslatex color`
- Die Ausgabedatei muss auf `.tex` enden!

Beispiel

```
set term epslatex color solid
set output 'test.tex'
set xlabel '$\alpha$ [rad]'
set ylabel '$F(\alpha)$ [N]'
...
```

Achtung!

`epslatex` erzeugt 2 Dateien: Die `.tex`-Datei enthält alle Beschriftungen und die Position des Bildes, die `.eps`-Datei enthält das Bild. Erst durch Einfügen der `.tex`-Datei in ein \LaTeX -Dokument werden sie vereint!

"Speichern"

Das Terminal epslatex – Unsere Verbindung zu L^AT_EX

- Für pdflatex muss das Bild allerdings als .pdf vorliegen:
epstopdf dateiname.eps
- In L^AT_EX wird der Plot dann mit `\input{dateiname}` eingebunden

Achtung!

Aufgrund eines gnuplot-Problems muss vor der Konvertierung zu .pdf die Ausgabedatei geschlossen werden. Dies geschieht durch ein einfaches `set output` oder Beenden von gnuplot.

"Speichern"

Das Terminal epslatex – Unsere Verbindung zu L^AT_EX

```
rahmendatei_latex.plot
```

```
reset
```

```
set term epslatex color solid
```

```
set output 'test.tex'
```

```
set xlabel '$\alpha$ [rad]'
```

```
set ylabel '$F(\alpha)$ [N]'
```

```
p 'daten.txt' u 1:2 title '$F(\alpha)$'
```

```
#eps zu pdf konvertieren (für pdflatex):
```

```
set output
```

```
!epstopdf test.eps
```

Fitten - Kurvenlegen durch Datenpunkte

- Zu einem Satz gemessener Daten möchte man ein Modell haben, welches das physikalische System möglichst gut beschreibt:

Beispiel

Pauline und Peter sind eifrige Naturforscher. Jede Woche "messen" sie die Anzahl Staubmäuse unter ihrem Bett. Nach einem Semester tragen sie die Daten auf und versuchen die zugrundeliegende Gesetzmäßigkeit zu finden. Damit wollen sie nach dem Saugen die Mausanzahl der x -ten Woche voraussagen können.

Linearer Fit \equiv Ausgleichsgrade

- Das einfachste Modell ist linear, das probieren wir zuerst!
- In gnuplot wird mit `fit` gefittet
- Anzugeben sind: vermutetes Modell $f(x) = mx + b$, die zugrundeliegenden Daten und anzupassende Parameter des Modells (via `m,b`)

Beispiel

```
f(x)=m*x+b
```

```
fit f(x) 'daten.txt' via m,b
```

```
p 'daten.txt', f(x)
```

Linearer Fit \equiv Ausgleichsgrade

- Hat man Messfehler zu Daten und möchte diese in den Fit mit einbeziehen, benutzt man `using`:

Beispiel

```
f(x)=m*x+b
```

```
fit f(x) 'daten.txt' using 1:2:3 via m,b
```

```
p 'daten.txt', f(x)
```

Linearer Fit \equiv Ausgleichsgrade

Wie geht das? – Die χ^2 -Methode

Um ein Modell an einen vorhandenen Datensatz anzupassen, bestimmt man die Summe der quadrierten Differenzen, die bei einem perfekten Fit minimal wird:

$$\chi^2(\text{parameter}) = \sum_{i=1}^N \left(\frac{x_i^{\text{exp}} - x_i^{\text{theo}}(\text{parameter})}{\sigma_i} \right)^2$$

- x_i^{exp} ist der i-te Messwert mit Fehler σ_i
- $x_i^{\text{theo}}(\text{parameter})$ der vorhergesagte Wert an dieser Stelle

Linearer Fit \equiv Ausgleichsgrade

Ist das gut? – reduziertes χ^2

Das reduzierte χ^2 gibt uns ein Maß dafür, wie gut unser Fit ist:

$$\chi_{red}^2 = \frac{\chi^2}{N - p_{Anzahl}}$$

- N Anzahl Datenpunkte und p_{Anzahl} Anzahl Parameter
- $\chi_{red}^2 \gg 1$ Fehler unteschätzt, Modell oder Statistik falsch
- $\chi_{red}^2 \ll 1$ Fehler überschätzt, zu allgemeines Modell oder Stat. falsch

Mehr dazu in der nächsten Woche!

Linearer Fit \equiv Ausgleichsgrade

Beispiel: Logbuch

fit.log

After 5 iterations the fit converged.

final sum of squares of residuals : 9.0698

rel. change during last iteration : -3.44266e-11

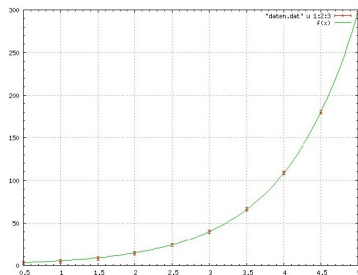
degrees of freedom (FIT_NDF): 9

rms of residuals (FIT_STDFIT) = $\sqrt{\text{WSSR}/\text{ndf}}$: 1.00387

variance of residuals (**reduced chisquare**) = WSSR/ndf : **1.00776**

Nichtlineares Fitten – Eine Kunst

- Beliebige Funktionen können angefittet werden
- Startwerte der Parameter setzen (default: alle 1.0 aber wir wollen das globale Minimum, nicht irgendein lokales)!!
- Modell möglichst als Linearkombination von Funktionen darstellen



Beispiel

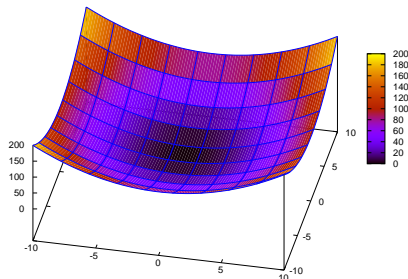
$$f(x) = a \cdot \exp(b \cdot x)$$

$$a = 1.0; \quad b = 0.9$$

```
fit f(x) 'daten.txt' u 1:2:3
via a,b
p 'daten.txt', f(x)
```


3D-Plots

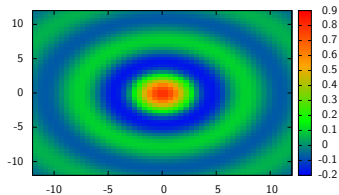
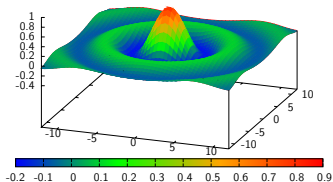
- `splot z(x,y)` plottet Gitternetzlinien im \mathbb{R}^3
- `using` bekommt hier die Spaltennummern von $(x, y, z, Zerr)$
- `set hidden3d` erkennt eine echte Oberfläche
- `set pm3d at s` projiziert eine Farbkodierung auf die Oberfläche



Beispiel

```
set hidden3d
set pm3d at s
splot x**2+y**2
```

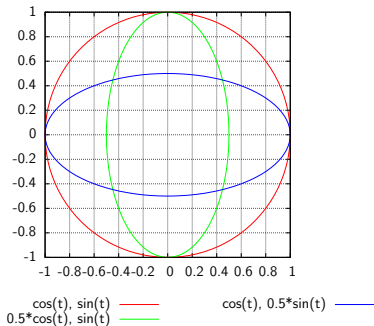
Die Farbpalette



- `set palette` kümmert sich die benutzen Farben
- `set colorbox` verschiebt die Farblegende an die richtige Stelle
- `set pm3d map` liefert 2D-Farbkarten einer Oberfläche

Parameterdarstellungen

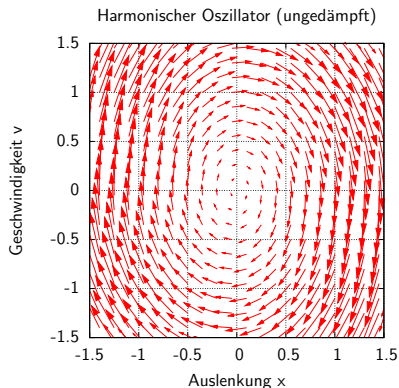
- Für Kurven in Parameterdarstellung: `set parametric`
- Standardvariablen: `t` (`plot`) und `u,v` (`splot`)
- Wertebereich des Parameters: `set trange [von:bis]`



Beispiel

```
set parametric
p [0:2*pi] cos(t),sin(t) lw 2
```

Vektorfelder



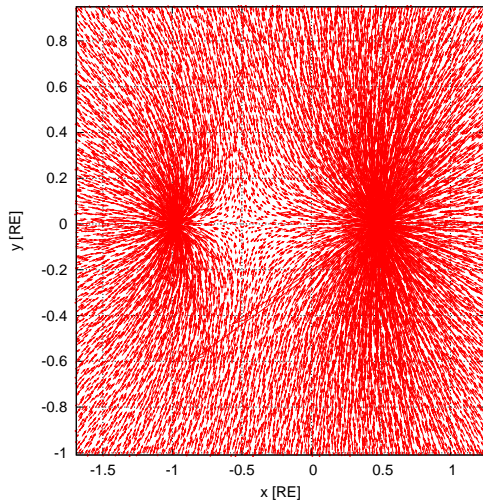

- Vektorfelder können nur aus Daten geplottet werden
- Benötigtes Dateiformat: $(x, y, \Delta x, \Delta y)$
- gnuplot malt Pfeile von $(x, y) \rightarrow (x + \Delta x, y + \Delta y)$

Beispiel

`p 'daten.txt' using 1:2:3:4 with vectors head filled`

Vektorfelder

Beispielplot

Deflection field of a binary lens ($q=0.5$, $s=1.5$ [RE])vectors scaled with 0.05 

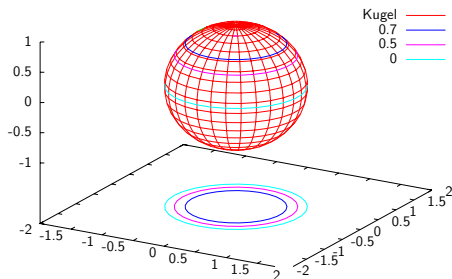
Multiplots

- Mehrere Plots über- oder nebeneinander sind mit Multiplot möglich
- Beginn mit `set multiplot`
Ende mit `unset Multiplot`
- Größe der Plots mit `set size`
`x.x,y.y`
- Positionierung mit `set origin`
`x.x,y.y` (Untere linke Ecke)

Beispiel

```
set mulitplot
set size 0.5,1.0
set origin 0.0,0.0
p x**2
set origin 0.5,0.0
p x**3
unset multiplot
```

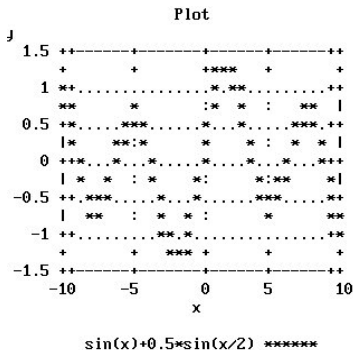
Konturplots



- `set contour` liefert Höhenlinien zu `splots`
(Optionen: `base`, `surface`, `both`)
- Anzahl Höhenlinien ändern oder nur ganz spezielle zeichnen:
`set cntrparam levels discrete 1,2,3`

gnuplot und ssh

- Zugriff per ssh auf den CIP-Pool
- ssh -X nicht verfügbar oder zu langsam
- gnuplot auf Konsolenbasis:
set term dumb



Nützliche Links

- <http://www.gnuplot.info> (gnuplot Homepage)
- <http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>
(not so FAQs)
- <http://gnuplot.sourceforge.net/demo/> (gnuplot Demos)
- <http://dynamicnetservices.com/~will/gnuplot/>
(Fun with gnuplot)

Fragen?

Vielen Dank an Markus Hundertmark, an dessen Vorträgen aus den Vorjahren ich mich orientiert habe.