

Einführung in *gnuplot*

Bei *gnuplot* handelt es sich um ein freies Programm zur grafischen Darstellung von Messdaten und zum plotten von Funktionen. Dies soll eine kurze Einführung in dieses Programm und Auflistung einiger weniger nützlicher Befehle sein. Weitere Informationen befinden sich natürlich auch im Internet (z.B. <http://www.gnuplot.info/>, dort gibt es auch einen Punkt download und auch die documentation zu *gnuplot*, die zwar deutlich ausführlicher aber nicht sehr einfach zu lesen ist). *gnuplot* bietet noch weitaus mehr Funktionen und Möglichkeiten als hier dargestellt sind, diese Kurzeinführung soll nur eine kleine Kostprobe bieten. *gnuplot* wird über das Kommandozeilen-Terminal gesteuert. Aufgerufen wird es durch den Befehl

```
gnuplot
```

Beendet wird es durch Eingabe von z.B.

```
q //auch exit und quite funktionieren
```

Innerhalb von *gnuplot* existiert eine *history*-Funktion, dies es erlaubt, vorangegangene Eingaben mittels Pfeiltasten (up, down) wiederzuholen und zu verändern. Über den Befehl

```
help
```

kann die Hilfefunktion innerhalb des Programms aufgerufen werden. Eine Funktion kann in *gnuplot* einfach durch den Befehl

```
plot sin(x) //plottet die Funktion sin(x)
```

geplottet werden. Alternativ erreicht man dies auch durch

```
f(x)=sin(x) //definiert Funktion f(x)
```

```
plot f(x) //plottet diese
```

Dabei versucht *gnuplot* automatisch eine möglichst gute Wahl der Einstellungen vorzunehmen. Möchte man ein bestimmtes Intervall per Hand einstellen, so geschieht dies über

```
plot [-1:5] sin(x) //plottet sin(x) im Intervall [-1,5]
```

```
plot [-1:5] [-2:1] sin(x) //plottet sin(x) im Rechteck [-1,5]x[-2,1]
```

gnuplot gibt dem geplotteten Graphen den Namen der Funktion (z.B. $\sin(x)$ oder $f(x)$). Man kann dem Graphen auch eigene Namen geben

```
plot sin(x) title 'Sinusschwingung' //Namen geändert
plot sin(x) notitle //ohne Namen
```

Befinden sich die Daten in einem file, so kann auch dieses von *gnuplot* verwendet werden

```
plot 'a.dat' //plottet Daten aus a.dat
```

Dabei ist zu beachten, dass *gnuplot* in dem Verzeichnis sucht, in dem es aufgerufen wurde. Standardmäßig werden die Daten der 2. Spalte (y-Spalte) gegen die der 1. Spalte (x-Spalte) aufgetragen. Dies kann umgeändert werden

```
plot 'a.dat' using 2:3 //plottet 2. Spalte gegen 3. Spalte aus a.dat
```

Die Daten aus a.dat werden standardmäßig als Punkte ausgegeben

```
plot 'a.dat' with lines //Werte durch Linie verbunden
plot 'a.dat' with linespoints //Werte als Punkte dargestellt
                                und durch Linie verbunden
plot 'a.dat' with boxes //erstellt ein Histogramm
```

Möchte man die Datenpunkte mit Fehlerbalken darstellen, so benutzt man

```
plot 'a.dat' using 1:2:3 with xerrorbars //Spalte 3: Fehler von x
plot 'a.dat' using 1:2:3 with yerrorbars //Spalte 4: Fehler von y
plot 'a.dat' using 1:2:3:4 with xyerrorbars //Spalte 3/4: Fehler von x/y
```

Hierbei ist die 1. Spalte der x-Wert, die 2. Spalte der y-Wert. Möchte man den Fehler nach unten und oben unterschiedlich haben, so benutzt man

```
plot 'a.dat' using 1:2:3:4 with xerrorbars //Spalte 3/4:
                                Fehler von x nach unten/oben
```

Um die Farbe des Graphen abzuändern, muss man zusätzlich eine Nummer eingeben, z.B.

```
plot sin(x) 3
```

Nun ist die Farbe des Graphen blau (default: 1 rot).

gnuplot kann auch mehrere Graphen in einem Schaubild plotten, dafür werden die einzelnen Funktionen durch Kommata voneinander abgetrennt

```
plot [-1:1] [-0.5:2] sin(x),cos(x),2*x+0.5 notitle //plottet Funktionen
                                     in diesem Rechteck
```

Man kann die *gnuplot*-Standardeinstellungen per Hand ändern

```
set xrange[-2:3] //x-Ausschnitt des plots fest auf [-2,3] setzten
set yrange[0:5] //y-Ausschnitt des plots fest auf [0,5] setzten
set xlabel 'Zeit t' //Beschriftung der x-Achse
set ylabel 'y-Achse' //Beschriftung der y-Achse
set grid //fügt Gitternetz ein
set logscale //skaliert Achsen logarithmisch
set nokey //entfernt Bezeichnungen in oberer Ecke
set style data linespoints //setzt Standardanzeige auf Punkte mit Linie
```

Hat man sich die Mühe gemacht, die Standardeinstellungen nach seinen Wünschen anzupassen, so kann man diese Einstellungen abspeichern und zu einem späterem Zeitpunkt (z.B. nach einem Neustart) laden

```
save 'savefile.plt' //speichert Einstellungen in savefile.plt
load 'savefile.plt' //lädt entsprechende Einstellungen
```

Um all diese Einstellungen rückgängig zu machen, um also *gnuplot* auf die Standardeinstellungen zurückzusetzen, verwendet man

```
reset
```

Um einen plot in Polarkoordinaten zu erzeugen, benutzt man folgende Befehle

```
set polar //Polarkoordinaten
set angles degrees //Winkelmaß
set grid polar //Polargitternetz einfügen
plot cos(t),sin(0.5*t)*0.75 //Funktionen plotten
```

gnuplot kann auch dreidimensionale plots erstellen, beispielsweise

```
splot sin(x)+2*y //plottet dies Funktion
splot 'a.dat' //macht dreidimensionalen plot von a.dat
```

Die mit *gnuplot* erzeugten plots können auch abgespeichert werden, beispielsweise

```
set terminal jpeg //erzeugt als Ausgabe ein jpeg-Bild
set output 'bild.jpeg' //erzeugt die Ausgabedatei
splot sin(x)*y**2 //plottet den 3d-plot sin(x)*y^2 in Ausgabedatei
```

Andere Ausgabeformate werden angezeigt über

```
set terminal //listet mögliche Ausgabeformate auf
```