

# Anhang D

## Kurze Einführung in gnuplot

`gnuplot` ist ein flexibles und recht einfach zu bedienendes Programm zur graphischen Darstellung von Daten. Es hat den zusätzlichen Vorteil, für alle gängigen PC-Betriebssysteme verfügbar zu sein. Unter LINUX wird es an der Kommandozeile einfach mit

```
gnuplot
```

aufgerufen. Die Shell übergibt die Eingabeaufforderung an das Programm. Die graphische (2D-)Darstellung von Funktionen und Daten erfolgt mit dem `plot`-Befehl. Für WINDOWS gibt es `gnuplot` wie gewohnt mit einem entsprechenden Desktopsymbol.

### D.1 Mathematische Funktionen

Man kann mehrere mathematische Funktionen gleichzeitig plotten lassen, die in verschiedenen Farben dargestellt werden. Die Funktionen werden dabei durch Kommata voneinander getrennt.

```
plot Ausdruck1 , Ausdruck2; , ...
```

**Beispiel:**

1. Plot von  $x^2$ :

```
plot x ** 2
```

2. Plot von  $\sin(3x) + \cos(4x)$ :

```
plot sin(3 * x) + cos(4 * x)
```

3. Plot von  $\exp(-x^2)$  und  $\log(-1/x)$ :

```
plot exp(-x ** 2) , log(-1/x)
```

## D.2 Plotten von Daten

Man kann auch Listen bzw. Wertetabellen, die man in einer Datei gespeichert hat, mit `gnuplot` graphisch darstellen.

- Enthält die Datei nur eine Spalte, so werden die Werte dort als Funktionswerte  $f(i)$  mit  $i = 1, 2, \dots$  interpretiert.
- Bei zwei Spalten wird die erste Spalte als  $x$ -Koordinate und die zweite als Funktionswert aufgefasst. Dies ist auch die Standardeinstellung bei mehr als zwei Spalten.
- Bei mehr als zwei Spalten wählt man die entsprechenden Spaltenpaare mit der Option `using`. Die beiden Spaltennummern werden durch `:` getrennt, und die erste als Liste der  $x$ - die zweite als Liste der  $y$ -Werte interpretiert. Verwendet man `using` mit nur einer Spaltennummer, so wird wie bei einspaltigen Daten verfahren.

**Beispiel:**

```
plot "mehrere.dat" using 1:3
```

plottet die Daten der dritten Spalte in der Datei `mehrere.dat` gegen die in Spalte 1 eingetragenen Werte.

- Der Dateiname muss in Anführungsstriche oder in Hochkommata gesetzt werden.

Standardmäßig werden die Daten als Punkte in der  $(x, y)$ -Ebene dargestellt. Man kann jedoch mit Hilfe von `w(ith)` eine andere Darstellungsform wählen:

Kurzform	ausführl. Form	Darstellung
<code>w p</code>	<code>with points</code>	Punkte (Standard)
<code>w l</code>	<code>with lines</code>	Linie
<code>w lp</code>	<code>with linespoints</code>	Punkte mit Linien verbunden
<code>w i</code>	<code>with impulses</code>	vertikale Linien

**Beispiel:**

Die Datei `wertetab.dat` enthalte die Wertetabelle einer Funktion. Ein approximativer Funktionsgraph wird durch den Aufruf

```
plot "wertetab.dat" w l
```

gezeichnet.

Die Datei `daten` enthalte mehrere Spalten. Um die Daten in der dritten Spalte mit vertikalen Linien zu zeichnen, verwendet man:

```
plot "daten" using 3 w i
```

Natürlich kann man auch Daten und mathematische Funktionen gleichzeitig plotten lassen, indem die entsprechenden Ausdrücke durch Kommata trennt.

## D.3 Darstellungsbereiche

Die  $x$ - und  $y$ -Darstellungsbereiche werden von `gnuplot` anhand der Daten automatisch gewählt. Möchte man sich nur bestimmte Intervalle anzeigen lassen, so kann man die Kommandos

```
set xrange [untere Grenze : obere Grenze]
```

und

```
set yrange [untere Grenze : obere Grenze]
```

verwenden. Durch anschließenden Aufruf von `replot` wird die Darstellung aktualisiert.

`gnuplot` hat noch eine Vielzahl von weiteren Funktionen. Mit `help commands` bzw. `unter`

```
http://www.gnuplot.info/
```

findet man weitere Informationen.