

Eine Anleitung für gnuplot

Lukas Semrau

23. Januar 2022

SKRIPTE UND PLOTS. Alle *gnuplot*-Skripte und *png*-Plots (und andere Dateien) gibt es unter lukas-semrau.de/gp.

gnuplot ist ein Programm, dass durch Skripte Messdaten oder mathem. Funktionen darstellen kann. (vgl. Abb. 1)

1. Einfache Plots

Nach der Installation von *gnuplot* kann man es im Windows / Linux / Mac Terminal mit dem Befehl:

```
1 path> gnuplot
```

ausführen. Allerdings geht das auch mit dem Hauseigenen Terminal von *gnuplot*, bei welchem man sich den ersten Befehl erspart. Wir versuchen nun die Funktion $f(x) = \sin(x)$ in einem (sehr einfachen) Plot darzustellen.

```
1 gnuplot> p sin(x)
```

Code 1: Skript für Abb.2.

ABER WIE FUNKTIONIERT DAS? Mit dem Befehl *plot* oder *p* zeichnet man die nachfolgende Funktion. Dabei muss man, ähnlich wie in anderen Sprachen, mit der Notation einiger Funktionen aufpassen. (vgl. Tab. 1)

SOLLEN 2 FUNKTIONEN IN EINEM GEMEINSAMEN KOORDINATENSYSTEM erscheinen, so werden die Zuordnungsvorschriften mit einem Komma von einander getrennt. Wir möchten Beispielsweise die Funktionen $f(x) = \sin x$ und $g(x) = \cos x$ zusammen plotten, so müssen wir folgenden Code schreiben:

```
1 gnuplot> plot sin(x), cos(x)
```

oder man definiert sich zu erst die verschiedenen Funktionen:¹

```
1 gnuplot> f(x)=sin(x)
2 gnuplot> g(x)=cos(x)
3 gnuplot> plot f(x), g(x)
```

Code 2: (oben) 2 Funktionen im Koordinatensystem ohne vorherige Definition der Funktionen. (unten Funktionen werden zunächst definiert.)

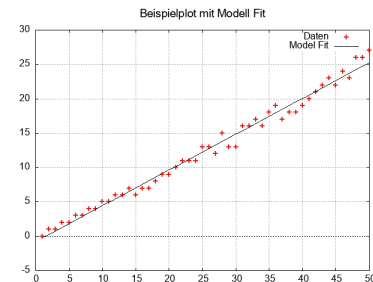


Abbildung 1: Beispielplot (s. Plot 1.)

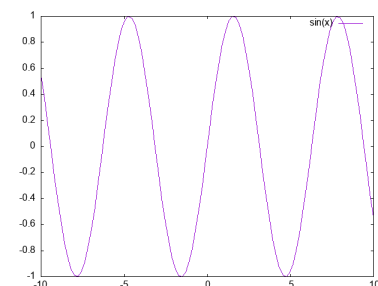


Abbildung 2: Plot zu Code 1. (s. plot2)

Mathe	gnuplot
1, 5	1.5
$\frac{1}{2}$	1./2
x^n	x**n
e^x	exp(x)
\sqrt{x}	sqrt(x)
$ x $	abs(x)

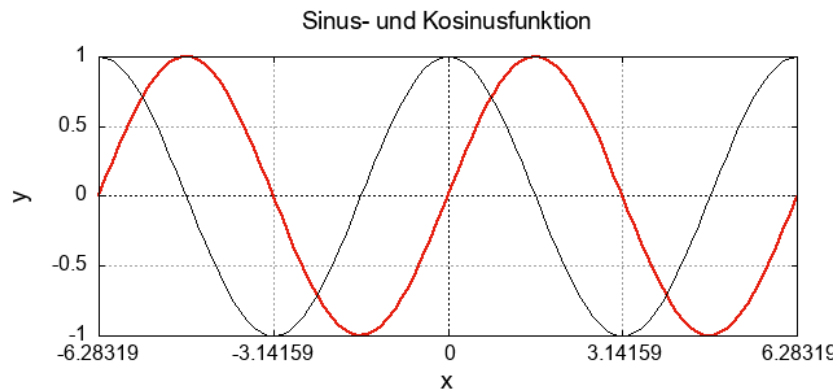
Tabelle 1: Notation einiger Funktionen

¹ Es gibt einen kleinen Unterschied zwischen den beiden Codes: Definiert man die Funktionen vorher und gibt ihnen einen Namen, so ändert sich die Legende (*key*) oben rechts in den Namen der Funktion.

2. Einfaches Layout

IN DIESEM ABSCHNITT möchte ich die Basics des graphischen zeigen, also wie man die Achsen beschriftet oder eine Gitter hinzufügt.

Abbildung 3: Plot, den wir am Ende erzeugen wollen.



Zunächst müssen wir uns anschauen, was diesen Plot ausmacht:

- **Achsenbeschriftungen:** Diese macht man mit den Befehlen `set xlabel` und `set ylabel`

```
1 gnuplot> set xlabel "x"
2 gnuplot> set ylabel "y"
```

Um Beschriftungen auszublenden nutzen wir den Befehl `unset`.

- **Begrenzungen in x - und y -Richtung:** Diese Abgrenzung macht man mit den Befehlen `xrange` und `yrange`^{2,3}

```
1 gnuplot> set xlabel [-2*pi:2*pi]
2 gnuplot> set yrange [-1:1]
```

- **Andeutung der Achsen:** Dies Andeutung macht man mit dem Befehl `zeroaxis`⁴

```
1 gnuplot> set zeroaxis
```

² Hier verwenden wir die Konstante π , das machen wir mit dem Befehl `pi`

³ Bitte auf die eckigen Klammern achten; Werte werden mit einem ":" getrennt.

⁴ Meistens lässt man etwas erscheinen / man stellt etwas ein mit dem Befehl `set`.

Übungen

[Übung 1] Plotten sie die Funktion $f(x)$.

$$f(x) = e^{\sqrt[3]{x^2}}$$

[Übung 2] Plote die Exponentialfunktion e^x im Intervall von -2 bis 2.

[Übung 3] Zu [Übung 2]: Blenden sie einen Titel und Achsenbeschriftungen ein.

3. Daten plotten

Daten erstellen

Zum Erstellen von Beispieldaten nutze ich eine einfache Exceltabelle, die nach einem physikalischen Gesetz Daten generiert oder nach Zufall handelt.

Möchten wir beispielsweise einen Zeit-Ort-Diagramm plotten, müssen wir uns für die Generierung der Daten Gedanken über die genutzte Formel machen und was Variablen sind und was nicht. Für ein Zeit-Ort-Diagramm nutzen wir die Formel

$$x(t) = \frac{1}{2} \cdot a \cdot t^2.$$

Um jetzt auch noch ein Paar Messfehler einzubauen, sagen wir, dass der Wert im folgenden Intervall liegt:

$$x_{\text{echt}} \in [0.95x; 1.05x]$$

Eine Beispielsformel in einem Feld wäre dann:^{5,6}

```
1 =0.5*0.5*t*t*Zufallsbereich(95; 105) / 100
```

Code 3: Bsp.-Formel für $x_a(t)$ mit a

Daten Plotten

```
1 gnuplot> set grid
2 gnuplot> set zeroaxis
3 gnuplot> set xlabel "t in s" font ", 14"
4 gnuplot> set ylabel "x in m" font ", 14"
5 gnuplot> set title "Zeit-Ort-Diagramm" font ", 16"
6 gnuplot> set size ratio 1
7 gnuplot> plot "daten.txt" lc 7 lw 5 title "Daten"
```

Code 4: Code zu Abb. 4

In Code 4 ist die wichtige Zeile die Letzte. In dieser Zeile steht:

```
1 gnuplot> plot "daten.txt"
```

Code 5: Es wird nur das Wichtigste gezeigt.

Gibt man also - in Anführungszeichen - den Namen der Datei⁷ (mit Dateityp an), so wird diese ausgelesen und dargestellt.

⁵ Exceltabelle im Ordner dataplot

⁶ Die Exceltabelle muss in eine .txt-Datei exportiert werden.

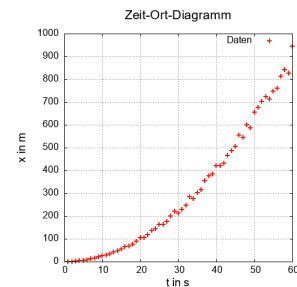


Abbildung 4: Der Plot des Zeit-Ort-Diagramms. (vgl. dataplot1)

1	0,2525
2	0,98
3	2,2725

Tabelle 2: erste Zeilen aus der Datei data.txt

⁷ Persönlich empfehle ich Dateien des Typs .dat oder .txt

Daten fitten.

Um für die Daten ein Modell berechnen zu lassen, müssen wir zunächst eine Beispielfunktion⁸ mit den gesuchten Parametern finden und diese zuerst definieren. Ist dies geschehen nutzen wir Zeile 2 in Code 6

```
1 gnuplot> f(x)=0.5*a*x**2
2 gnuplot> fit f(x) "daten.txt" using 1:2 via a
```

Code 6: Code zur Erzeugung von Datenmodellen

Man muss zunächst die Funktion angeben, die zur Modellierung erhalten soll. Danach gibt man die Daten und den Parameter zur Modellierung an. Bei dem Vorgehen wird eine Datei `fit.txt` erzeugt, die in Code 7 gezeigt wird.

```
1
2
3 *****
4 Thu Aug 26 15:19:15 2021
5
6
7 FIT:      data read from "daten.txt" using 1:2
8          format = x:z
9          #datapoints = 60
10         residuals are weighted equally (unit weight)
11
12 function used for fitting: f(x)
13   f(x)=0.5*a*x**2
14 fitted parameters initialized with current variable values
15
16 iter      chisq      delta/lim  lambda    a
17   0 1.0022152500e+07   0.00e+00  8.22e+02  1.000000e+00
18   3 1.0667620834e+04  -7.00e-04  8.22e-01  5.029212e-01
19
20 After 3 iterations the fit converged.
21 final sum of squares of residuals : 10667.6
22 rel. change during last iteration : -7.00364e-09
23
24 degrees of freedom      (FIT_NDF)                : 59
25 rms of residuals        (FIT_STDFIT) = sqrt(WSSR/ndf) : 13.4465
26 variance of residuals (reduced chisquare) = WSSR/ndf  : 180.807
27
28 Final set of parameters      Asymptotic Standard Error
29 =====
30 a = 0.502921                +/- 0.002112      (0.42%)
```

Code 7: `fit.txt` (Z.3 gekürzt)

⁸ Bei uns $f(x) = \frac{1}{2} \cdot a \cdot t^2$

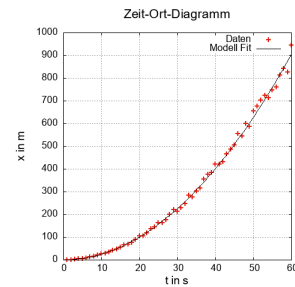


Abbildung 5: Abb. 4 mit Modellfit. Vergößert auf der Nächsten Seite.

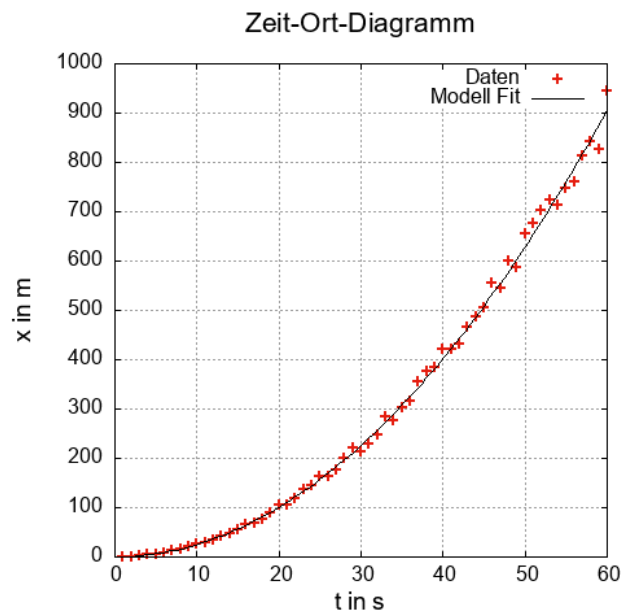


Abbildung 6: Plot 5 in groß.