

# gnuplot - in Kürze

## eps-Export

```
reset
set encoding iso_8859_1
set terminal postscript eps enhanced color linewidth 4 25
set output 'graph.eps'
...
set output
set terminal pop
```

## Beschriftung

```
set title "Schwingung ohne Dämpfung"
set xlabel "Zeit t [s]"
set ylabel "Amplitude {/Symbol D}x [beliebig]"

# zweite Achse
# set x2label "..."; set y2label "..."
```

```
plot sin(x) title "Schwingung", \
      cos(x) notitle
```

## Bereichseinschränkung

```
set xrange [-10:10]
set yrange [0:10]

... oder ...

plot [-10:10][0:10] sin(x)
```

- [Minimum:Maximum]
- man kann eine oder beide Grenzen weglassen ([Minimum:])
- Reihenfolge: [x-Achse] [y-Achse] [2. x-Achse] [2. y-Achse]

## Daten

```
# falls Dateien mit ',' als Dezimalzeichen:
# set decimalsign ','

# Spalte 3 über Spalte 1
plot 'data.dat' using 1:3

# Mittelwert der Spalten 2 und 3 über Spalte 1
plot 'data.dat' using 1:( 0.5*($2+$3) )

• mit $ auf Spalten referenzieren
• Zahl ohne Klammern → Spalte,
• Zahl innerhalb von Klammern, z.B. (1) → Wert
```

## Formatierungen / Beschriftung set ...

arrow	Pfeile zeichnen
border	Linien des Rahmens formatieren
boxwidth	Boxen für <code>linetype boxes</code> ... formatieren
decimalsign	Dezimalzeichen für Daten festlegen
encoding	Zeichensatz festlegen ( <code>set encoding iso_8859_1</code> )
fit errorvariables	Fehler beim Fitten in Variablen speichern
format	Achsenbeschriftung formatieren
grid	Gitternetz einstellen
key	Legende formatieren
label	Manuelle Beschriftungen setzen
logscale	Logarithmische Achsenkalibrierung setzen
multiplot	Mehrere Graphen gleichzeitig anzeigen
mxtics,mytics,...	Anzahl der Striche zwischen den Achsen-Skalenstrichen einstellen
output	Ausgabe-Datei festlegen
parametric	Parametrische Funktionen benutzen
pointsize	Symbolgröße einstellen
samples	Auflösung für Funktionen-Graphen einstellen
style	Verschiedene Formatierungen voreinstellen
terminal	Ausgabe-Format wählen
xtics,ytics,...	Achsen-Skalenstriche formatieren
title	Titel setzen
xlabel,ylabel,...	Achsenbeschriftung setzen
xzeroaxis, yzeroaxis,...	Ursprungslinien anzeigen

## png-Export

```
reset
set encoding iso_8859_1
set terminal png size 800,600 enhanced linewidth 4 20
set output 'graph.png'
...
set output
set terminal pop
```

## Skript laden

```
cd '<Skript-Verzeichnis>'
load '<Skript-Dateiname>'
```

## Griechische Symbole {/Symbol ...}, {/symbol ...}

abcdefghijklmnopqrstuvwxyz  
α β γ δ ε φ η ι φ κ λ μ ν ο π ϑ ρ σ τ υ ω ξ ψ ζ

ABCDEFGHIJKLMNOPQRSTUVWXYZ  
Α Β Χ Δ Ε Φ Γ Η Ι Θ Κ Λ Μ Ν Ο Π Θ Ρ Σ Τ Υ ς Ω Ξ Ψ Ζ

\305 Å \n neue Zeile

{...}^{...} Text hochstellen {...}\_{...} Text tiefstellen

## Funktionen

abs(x) acos(x) acosh(x) arg(x) asin(x)  
asinh(x) atan(x) atan2(y,x) atanh(x) EllipticK(k)  
EllipticE(k) EllipticPi(n,k) besj0(x) besj1(x)  
besy0(x) besy1(x) ceil(x) cos(x) cosh(x) erf(x)  
erfc(x) exp(x) floor(x) gamma(x) ibeta(p,q,x)  
inverf(x) igamma(a,x) imag(x) invnorm(x) int(x)  
lambertw(x) lgamma(x) log(x) log10(x) norm(x)  
rand(x) real(x) sgn(x) sin(x) sinh(x) sqrt(x)  
tan(x) tanh(x) x\*\*y ≐ x<sup>y</sup> pi ≐ π

## Fitten

```
# Funktion definieren
f(x) = exp(-x**2/w**2)
g(x) = a*x**2+b

# Startwerte vorgeben (optional) ->bessere Konvergenz!
w = 2

# Fitten ([...] und using sind optional)
fit [-10:10] A*f(x) 'sample.dat' using 1:2 via A,w
# Fitten mit Berücksichtigung des Fehlers:
fit A*f(x) 'sample.dat' using 1:2:3 via A,w
```

## Diagramm-Typen `plot ... with ...`

boxerrorbars boxes boxyerrorbars candlesticks  
circles dots filledcurves financebars fsteps  
histeps histograms newhistogram image rgbimage  
rgbalph impulses labels lines linespoints points  
vectors xerrorbars yerrorbars yerrorbars xerrorlines  
xyerrorlines yerrorlines pm3d

## Kurven formatieren `plot ...`

linetype Linientyp  
linewidth Linienbreite  
linecolor Linienfarbe (linecolor 1, linecolor rgb 'red')  
pointtype Symboltyp  
pointsize Symbolgröße

## Zweite Achse `plot ... axes ...`

x1y1 x1y2 x2y1 x2y2