

# Vorlesung Fehlerrechnung & Datenanalyse

Olaf Kaczmarek & Udo Werner

Heutiges Thema:

**gnuplot**

- Darstellung von Daten
- Lineare Skalen
- Darstellung von Fehlern
- Histogramme
- Beschriftungen
- Multiplots
- Parametrische Plots
- 3-dimensionale Plots
- Animationen

# Skripte = Plotdateien

## klima.gnu:

```
set xr [1880:2020]
set yr [:1.2]
set term windows font "Arial,14"
plot "klima.dat" u 1:2 title "Temperatur Land" w lp ls 1 lw 2 pt 5
rep "klima.dat" u 1:3 title "Temperatur Wasser" w lp ls 3 lw 2 pt 7
```

```
f(x) = a+b*x
g(x) = c+d*x
```

```
fit [1980:*] f(x) "klima.dat" u 1:2 via a,b
fit [1980:*] g(x) "klima.dat" u 1:3 via c,d
```

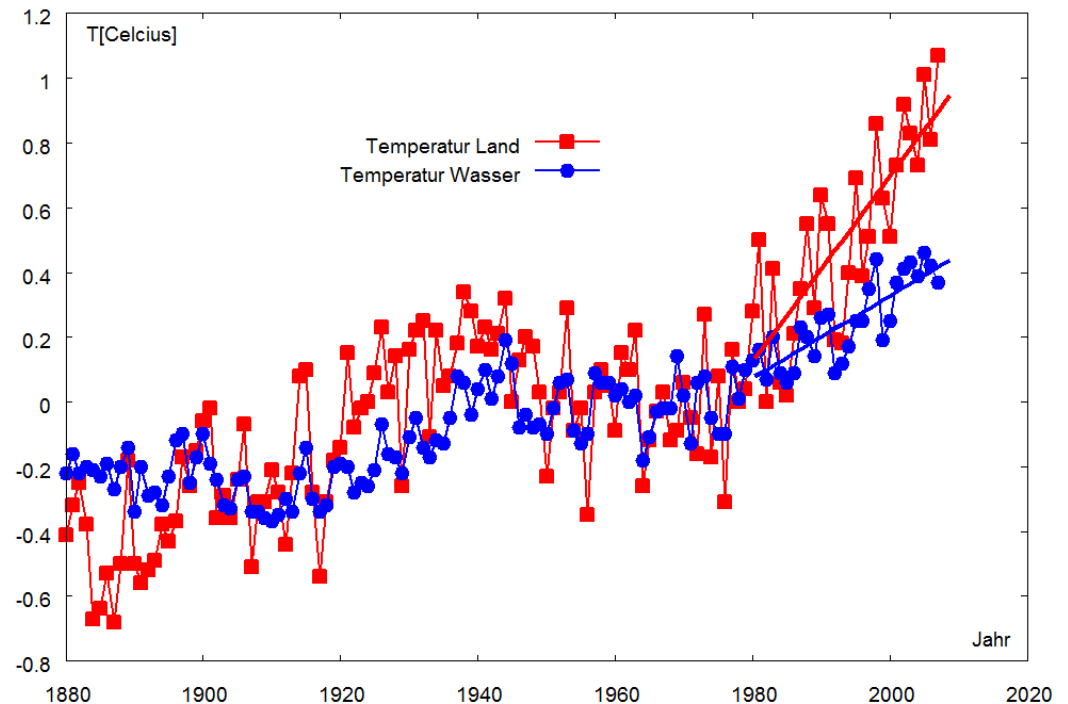
```
rep f(x)*((x>1980 && x<2010)?1:1/0) title "" w l ls 1 lw 4
rep g(x)*((x>1980 && x<2010)?1:1/0) title "" w l ls 3 lw 4
```

```
set key at 1960,0.85
set label 1 "Jahr" at 2012,-0.72
set label 2 "T[Celcius]" at 1883,1.15
set term emf 20
set out "klima.emf"
rep
set term windows
set out
```

Alle Befehle auf der Kommandozeile können in eine Textdatei geschrieben werden und mit

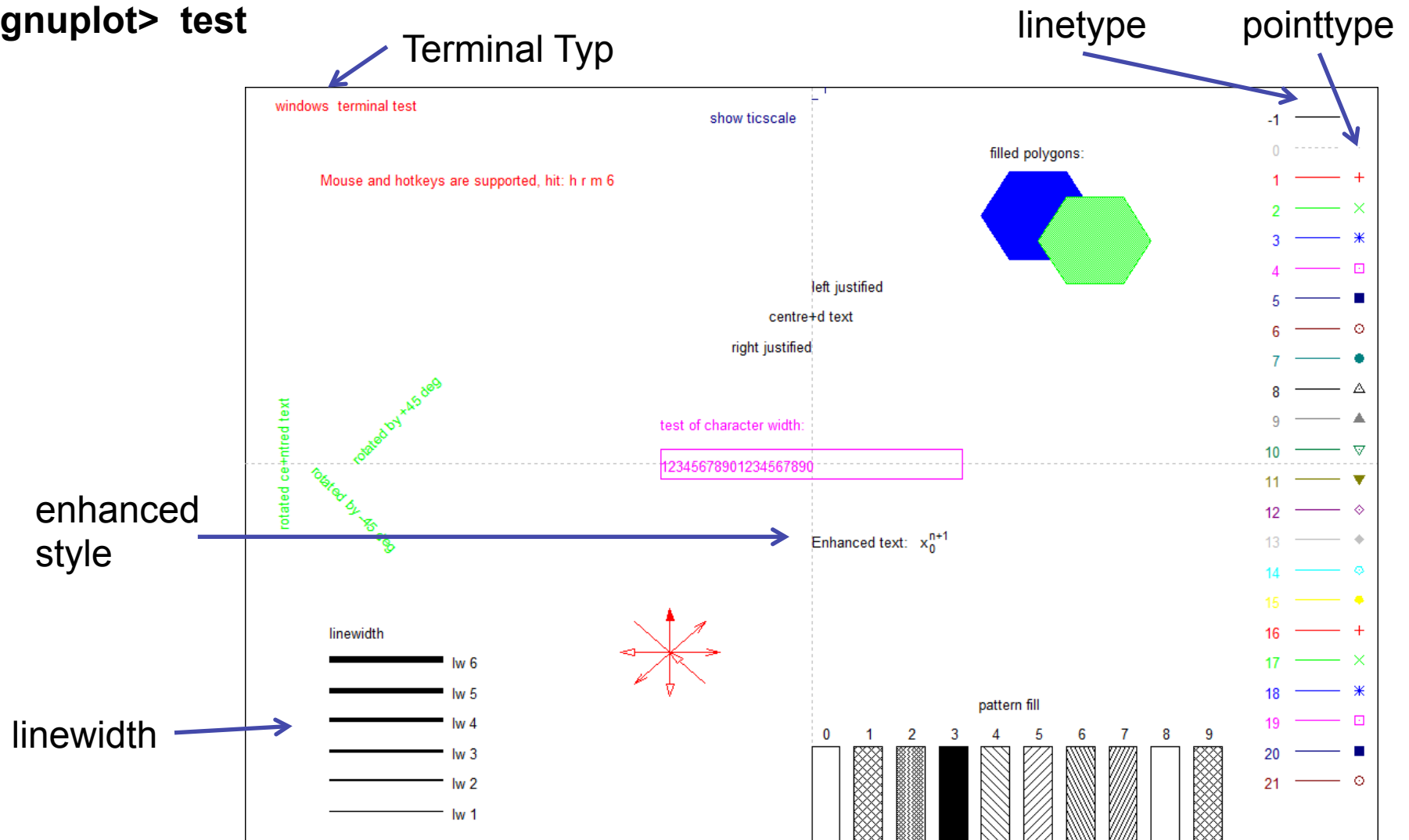
load "plotfile"

geladen und ausgeführt werden!



# Plot Styles

gnuplot> test



# Große Datenmengen

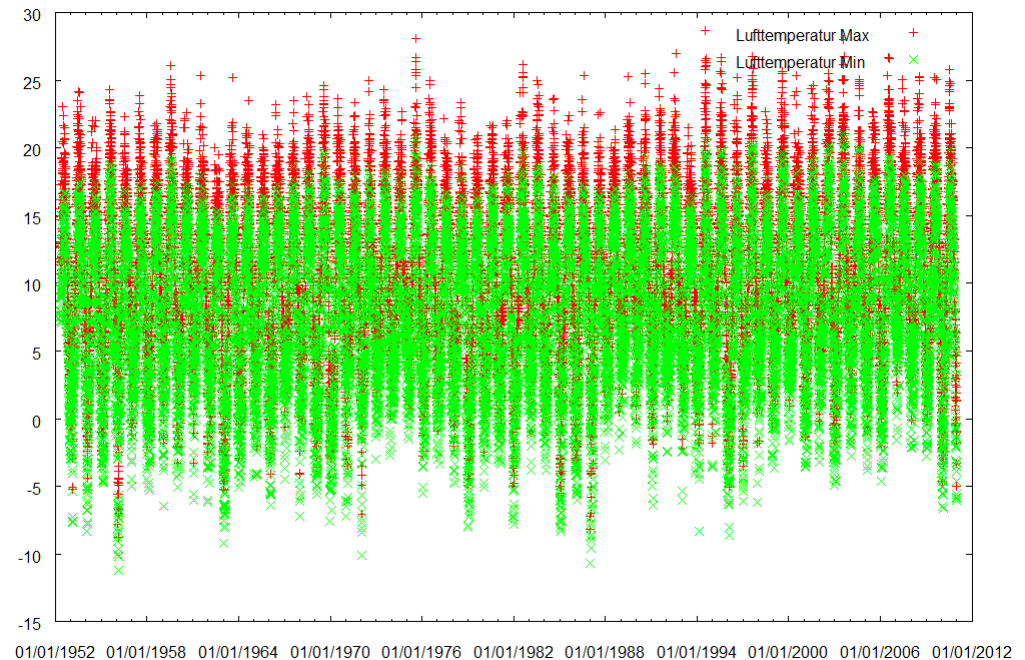
# Deutscher Wetterdienstes, Station Helgoland

#Mess\_Datum,REL\_FEUCHTE,LUFTTEMPERATUR,LUFTDRUCK\_STATIONSHOEHE,LUFTTEMPERATUR\_MINIMUM,LUFTTEMPERATUR\_MAXIMUM,NIEDERSCHLAGSHOEHE

19520501	80.00	10.6	1005.60	7.6	13.0	.0
19520502	80.00	10.2	1008.70	7.9	13.1	.0
19520503	75.00	11.1	1001.50	7.9	14.7	3.0
19520504	84.00	12.5	991.90	9.5	17.3	.3
19520505	86.00	11.5	992.30	9.5	16.2	4.7
19520506	88.00	12.4	1001.60	9.6	14.7	.7
19520507	90.00	10.0	1008.30	8.5	12.8	6.3
19520508	90.00	10.6	1014.00	7.5	11.8	.0
19520509	91.00	12.2	1010.00	9.5	15.5	21.4
19520510	97.00	9.9	1008.00	8.5	12.6	.9
19520511	85.00	11.7	1006.20	7.2	14.9	1.6
19520512	93.00	9.8	1008.80	9.0	12.9	.2
19520513	94.00	9.7	1011.20	8.5	11.4	.0
19520514	90.00	10.4	1011.60	8.5	11.6	.0
19520515	96.00	9.6	1012.20	8.0	11.6	.0
19520516	94.00	9.6	1014.00	8.9	10.7	.0
19520517	64.00	10.9	1018.40	8.9	12.7	.0
19520518	77.00	11.1	1011.20	9.8	15.0	.0
19520519	60.00	10.8	1013.40	9.2	13.4	.0
19520520	55.00	9.9	1021.00	7.7	12.0	.0
19520521	69.00	10.3	1023.40	8.4	12.8	.0
19520522	75.00	11.1	1022.30	9.5	12.9	.0
19520523	87.00	9.8	1021.50	9.0	11.8	.0
19520524	84.00	10.2	1017.90	8.4	11.6	.0
19520525	85.00	10.4	1015.40	8.9	11.9	.2
19520526	77.00	10.0	1010.00	9.1	11.7	.8
19520527	88.00	10.3	1007.50	8.4	11.5	1.3
19520528	80.00	10.1	997.60	8.9	10.7	1.6
19520529	76.00	8.9	997.90	7.2	10.3	3.1
19520530	75.00	10.4	1004.80	7.1	11.0	2.4
19520531	90.00	12.9	1004.20	9.3	14.9	1.5
19520601	90.00	15.2	1001.80	11.6	19.2	1.1
19520602	86.00	12.6	1005.80	12.0	15.3	5.6
19520603	87.00	10.7	1015.50	9.0	12.8	2.6
19520604	68.00	11.6	1022.30	7.6	13.5	.0
19520605	61.00	16.4	1013.60	10.6	19.3	.0
19520606	83.00	13.9	1003.00	12.3	17.6	10.7
19520607	76.00	11.0	1005.20	10.3	12.6	.0
19520608	78.00	11.6	1011.00	8.9	12.9	.3
19520609	75.00	10.9	1012.20	10.1	12.9	1.6
19520610	79.00	12.0	1008.80	9.8	14.0	1.0
19520611	86.00	11.2	1005.00	10.3	12.4	.0
19520612	85.00	11.1	1010.00	10.2	12.2	.0
19520613	86.00	14.2	1005.60	11.0	18.1	.0
19520614	88.00	12.4	999.40	10.8	15.7	3.2

Oft macht es keinen Sinn,  
alle Daten zu zeigen!

Interessiert mich wirklich das (tägliche)  
Wetter vor 50 Jahren?

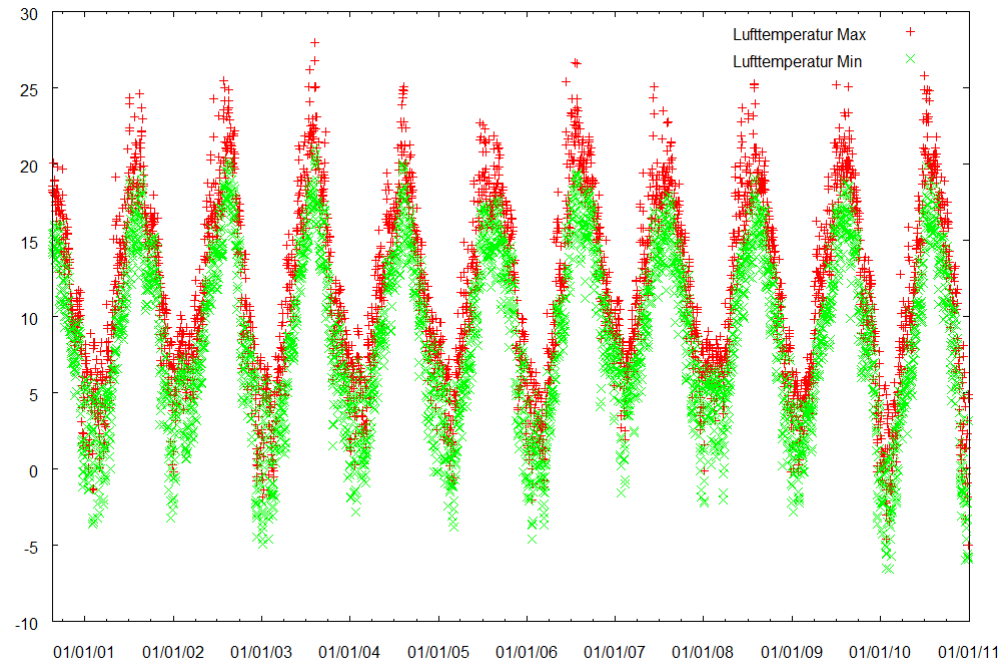


```
set xdata time
set timefmt "%Y%m%d"
plot "Helgoland.txt" u 1:6 title "Lufttemperatur Max" , \
      "Helgoland.txt" u 1:5 title "Lufttemperatur Min "
```

# Große Datenmengen

Was will ich mit den Daten aussagen?

Wie stelle ich das sinnvoll dar?

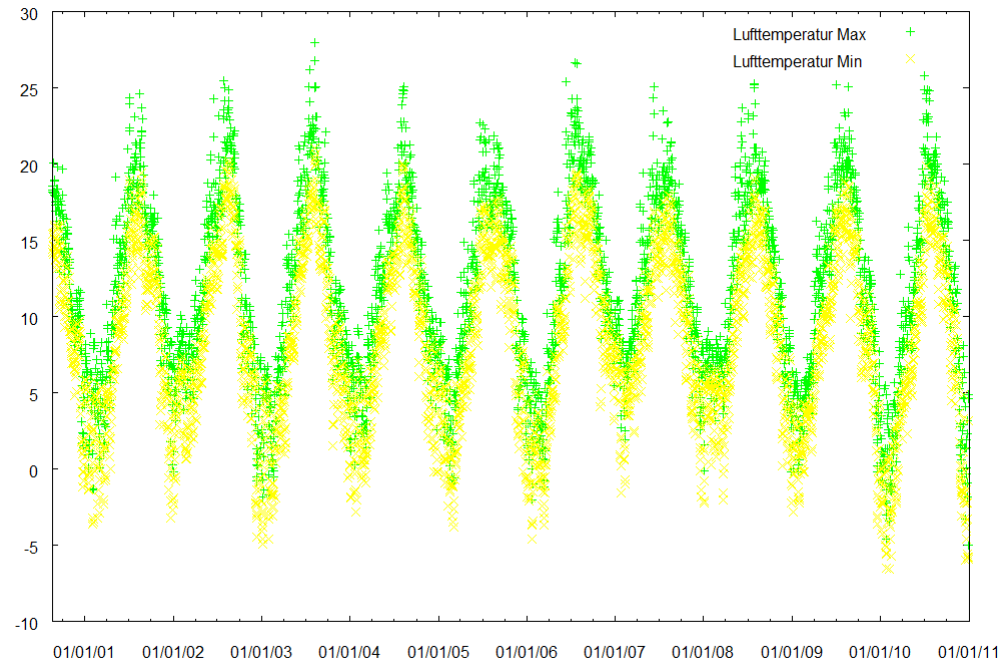


```
set xrange ["20010101":"20110101"]  
plot "Helgoland.txt" u 1:6 title "Lufttemperatur Max" with points linetype 1, \  
      "Helgoland.txt" u 1:5 title "Lufttemperatur Min " with points linetype 2
```

# Große Datenmengen

Was will ich mit den Daten aussagen?

Welche Farben benutze ich für welche Daten?



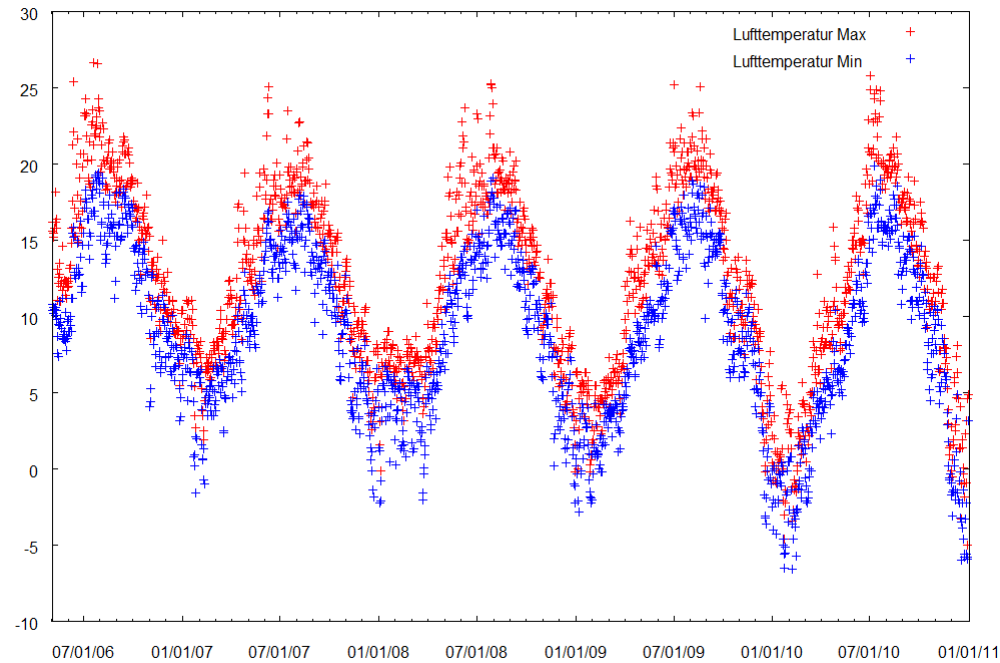
```
plot "Helgoland.txt" u 1:6 title "Lufttemperatur Max" w p lt 2, \
"Helgoland.txt" u 1:5 title "Lufttemperatur Min " w p lt 15
```

grün und gelb auf keinen Fall!!!

# Große Datenmengen

Was will ich mit den Daten aussagen?

Welche Farben benutze ich für welche Daten?

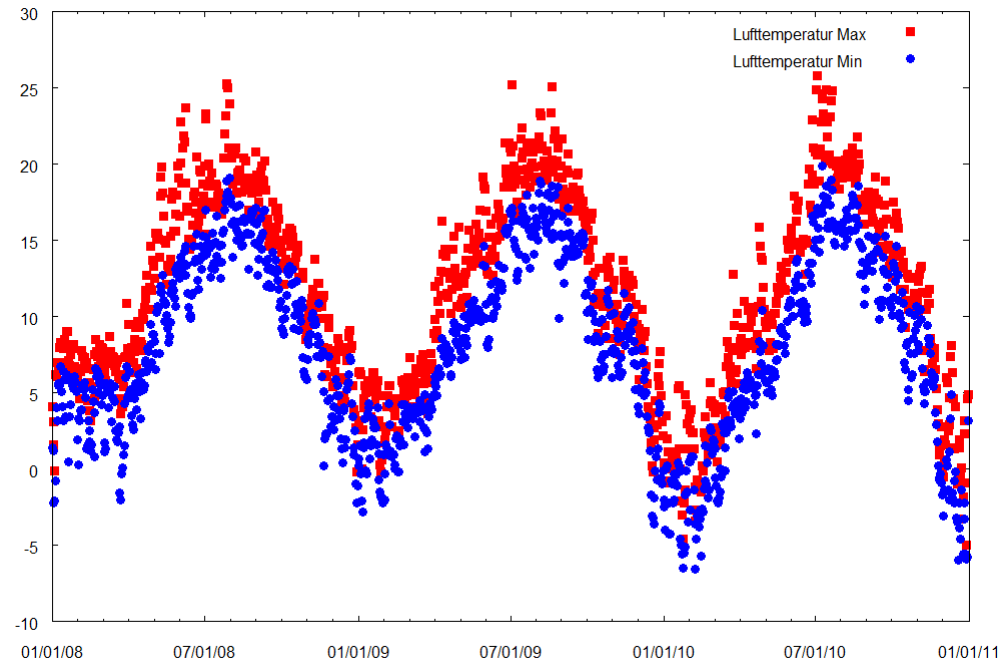


```
plot "Helgoland.txt" u 1:6 title "Lufttemperatur Max" w p lt 1, \  
"Helgoland.txt" u 1:5 title "Lufttemperatur Min " w p lt 3
```

# Große Datenmengen

Was will ich mit den Daten aussagen?

Welche Symbole benutze ich?



```
plot "Helgoland.txt" u 1:6 title "Lufttemperatur Max" w p pt 5 ps 1lt 1, \  
"Helgoland.txt" u 1:5 title "Lufttemperatur Min " w p pt 7 ps 1lt 3
```



## Skalen: Lineare Daten

Gegeben seien Messpunkte:

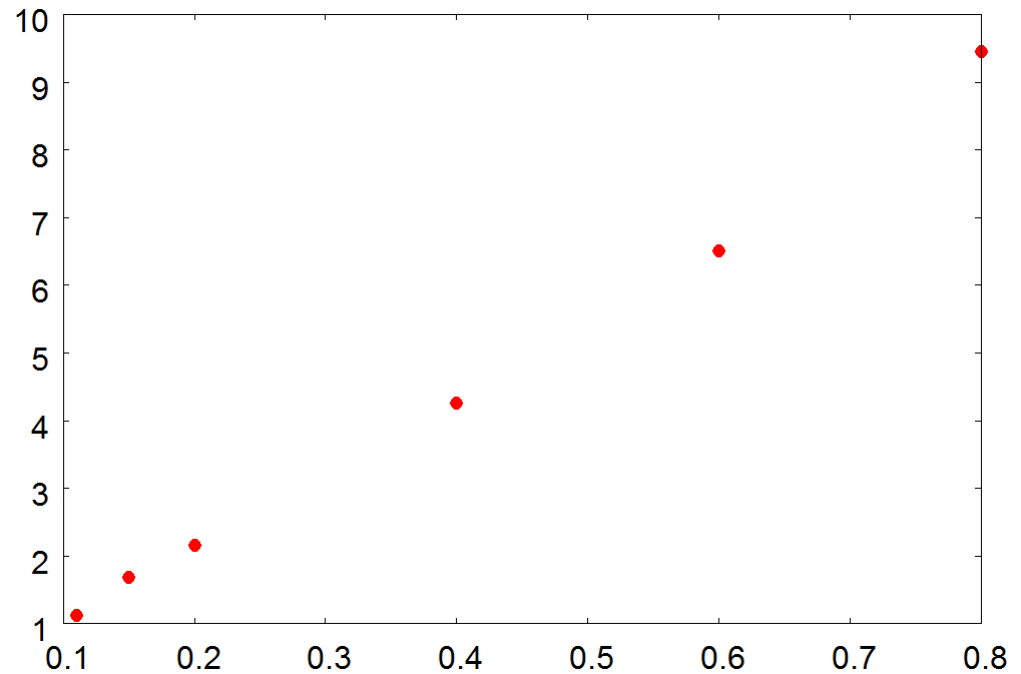
$(x_1, y_1), \dots, (x_N, y_N)$

und eine theoretische Vorhersage

$$y(x) = m \times x$$

d.h. ein linearer Zusammenhang von

x und y.



**Um zu sehen, dass die Daten wirklich einen linearen Verlauf zeigen (und nach  $y(0)=0$  extrapolieren), sollte der Ursprung mit geplottet werden!**

## Skalen: Lineare Daten

Gegeben seien Messpunkte:

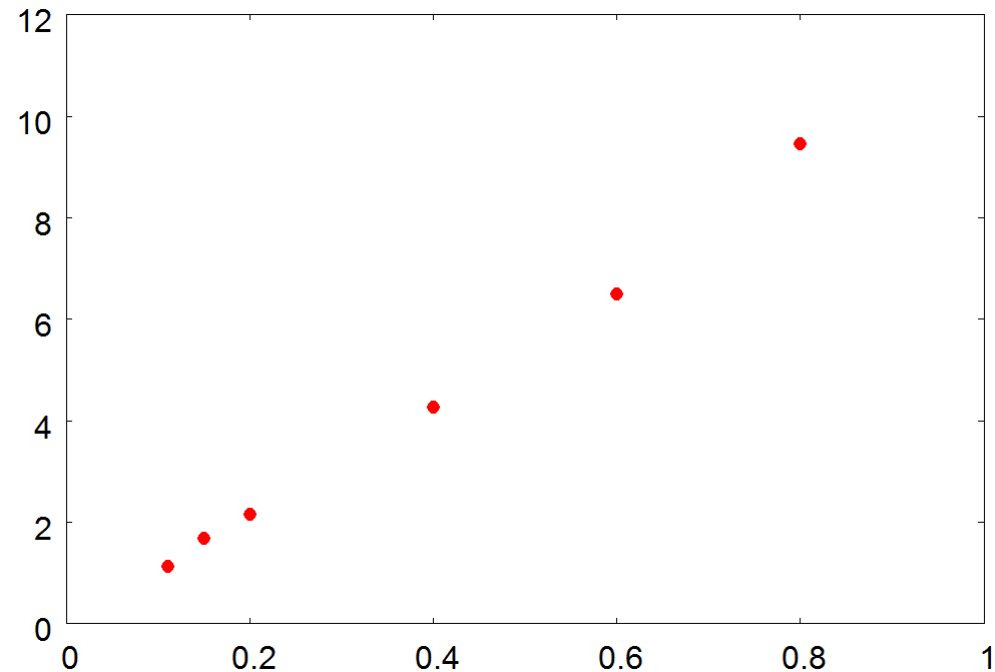
$(x_1, y_1), \dots, (x_N, y_N)$

und eine theoretische Vorhersage

$$y(x) = m \times x$$

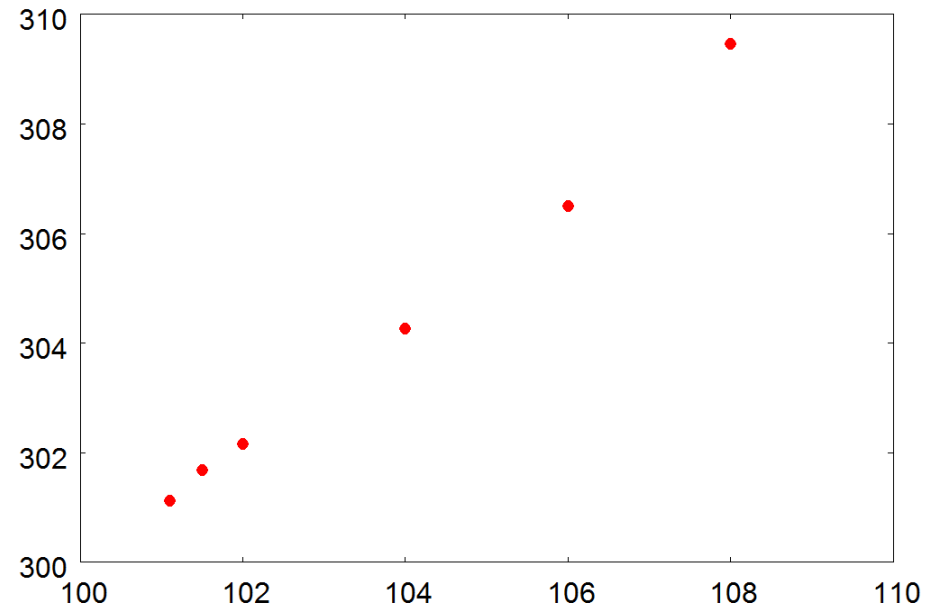
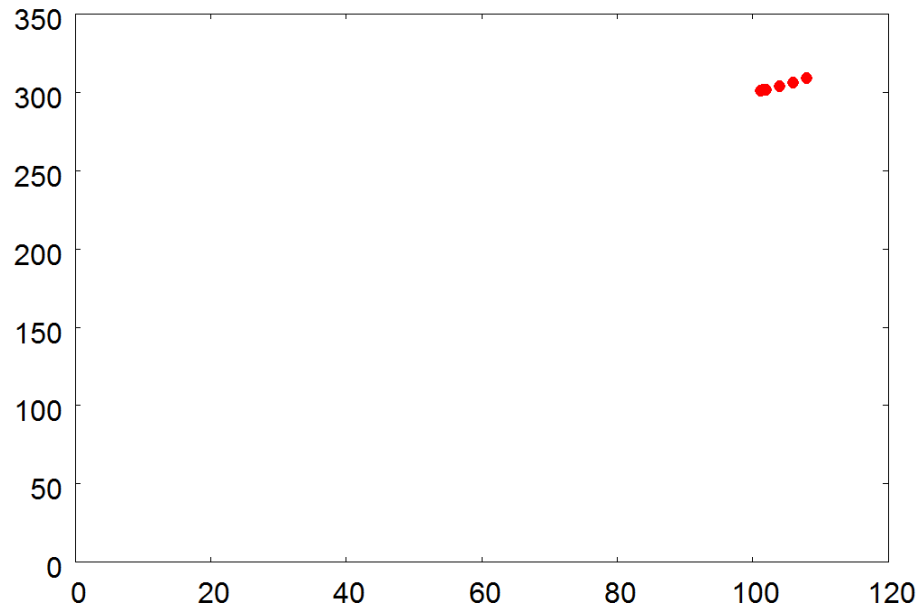
d.h. ein linearer Zusammenhang von

x und y.



**Um zu sehen, dass die Daten wirklich einen linearen Verlauf zeigen (und nach  $y(0)=0$  extrapolieren), sollte der Ursprung mit geplottet werden!**

## Skalen: Lineare Daten



**Um zu sehen, dass die Daten wirklich einen linearen Verlauf zeigen (und nach  $y(0)=0$  extrapolieren), sollte der Ursprung mit geplottet werden!**

**Das macht natürlich nicht immer Sinn (siehe Bild links)!**

# Skalen: Lineare Daten

Gegeben seien Messpunkte:

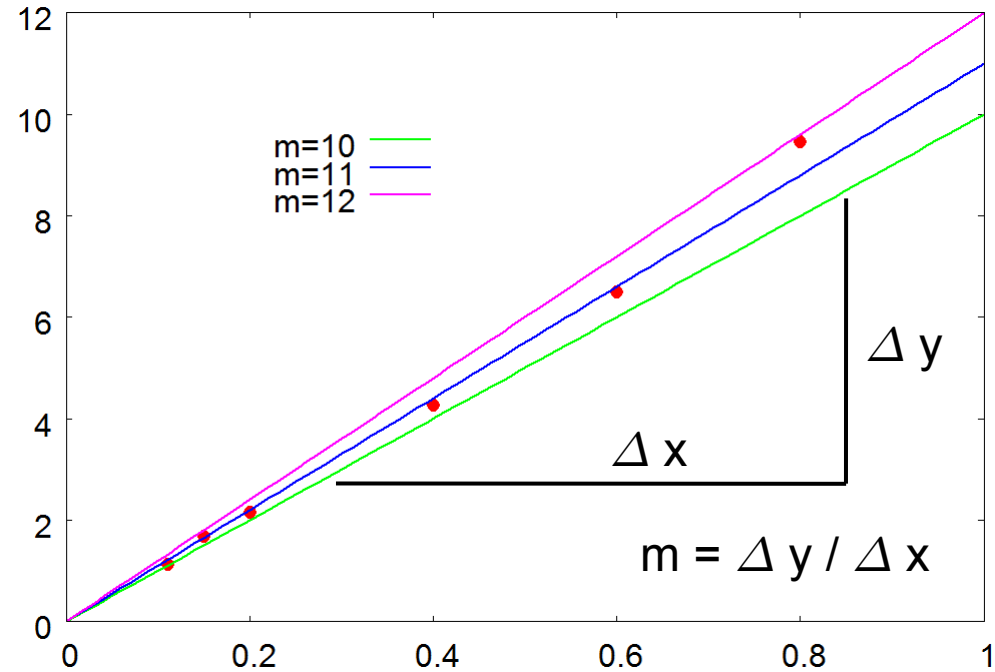
$(x_1, y_1), \dots, (x_N, y_N)$

und eine theoretische Vorhersage

$$y(x) = m \times x$$

d.h. ein linearer Zusammenhang von

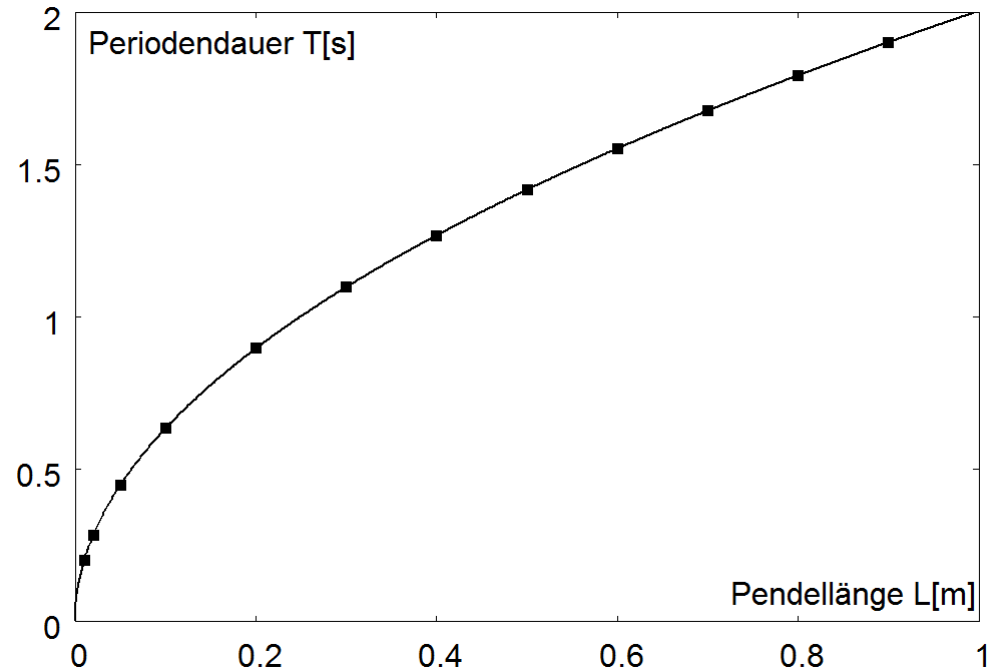
x und y.



## Beispiel: Fadenpendel

Periodendauer  $T$  eines Fadenpendels ist abhängig von der Pendellänge  $L$  und der Fallbeschleunigung  $g$ :

$$T = 2\pi \sqrt{\frac{L}{g}}$$



Messung der Periodendauer  $T$  für verschiedene Pendellängen  $L$

Auftragung zeigt eine gekrümmte Kurve

Ist die theoretisch erwartete funktionelle Abhängigkeit wirklich erfüllt ?

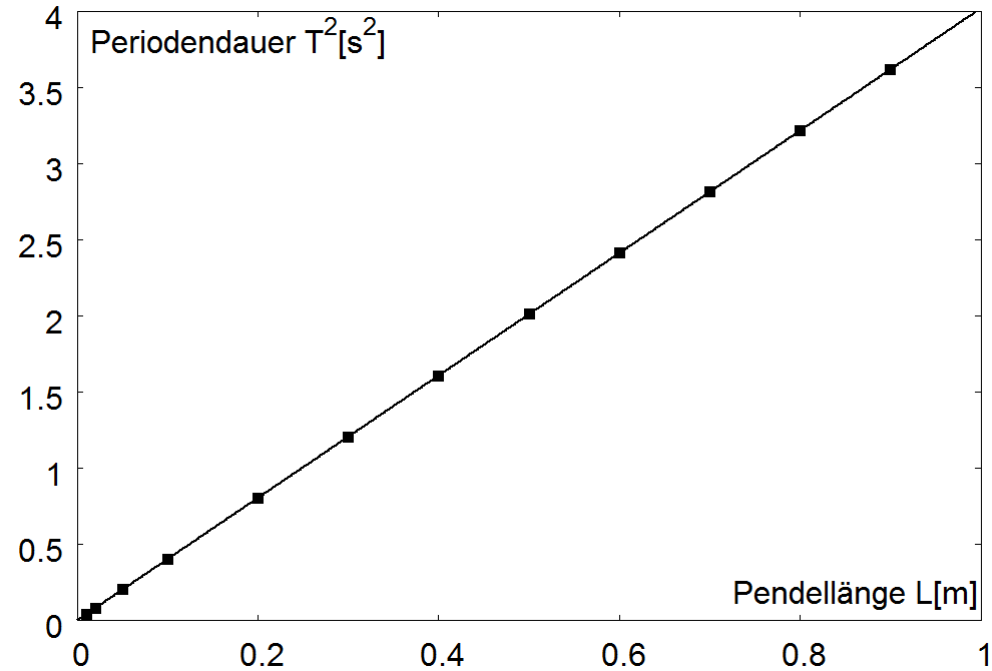
## Beispiel: Fadenpendel

Periodendauer  $T$  eines Fadenpendels  
ist abhängig von der Pendellänge  $L$   
und der Fallbeschleunigung  $g$ :

$$T = 2\pi \sqrt{\frac{L}{g}}$$

⇓

$$T^2 = \frac{4\pi^2}{g} L$$



Auftragung der Periodendauer  $T^2$  für verschiedene Pendellängen  $L$   
zeigt einen linearen Verlauf

Nun sieht man direkt, dass die theoretische Abhängigkeit erfüllt ist !

## Beispiel: Fadenpendel

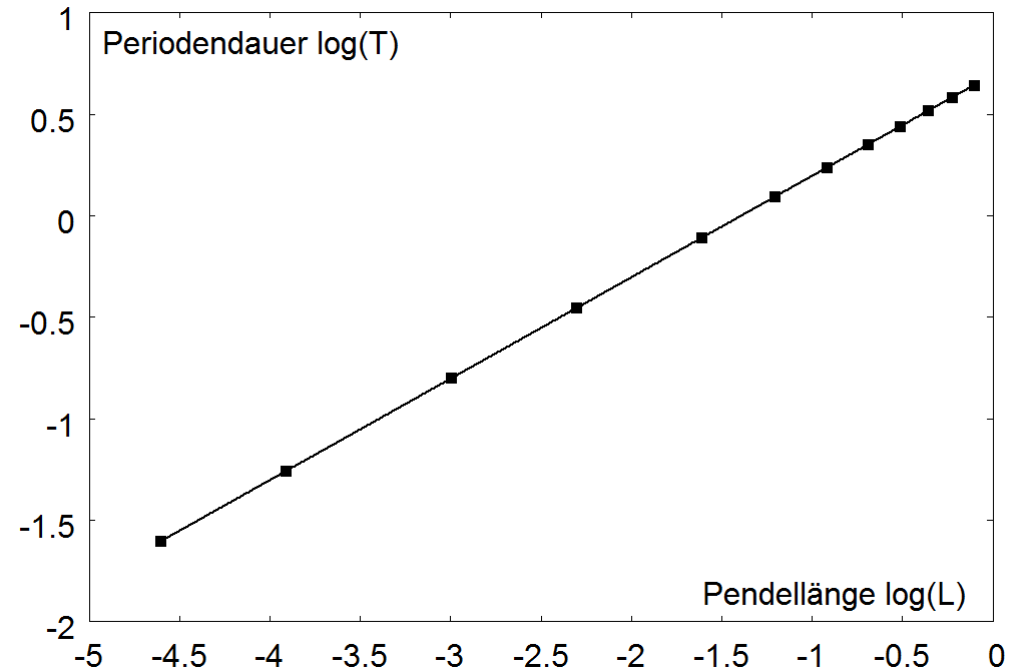
Periodendauer  $T$  eines Fadenpendels ist abhängig von der Pendellänge  $L$  und der Fallbeschleunigung  $g$ :

$$T = 2\pi \sqrt{\frac{L}{g}}$$

⇓

$$\log(T) = 0.5 \log(L) + \log(2\pi/\sqrt{g})$$

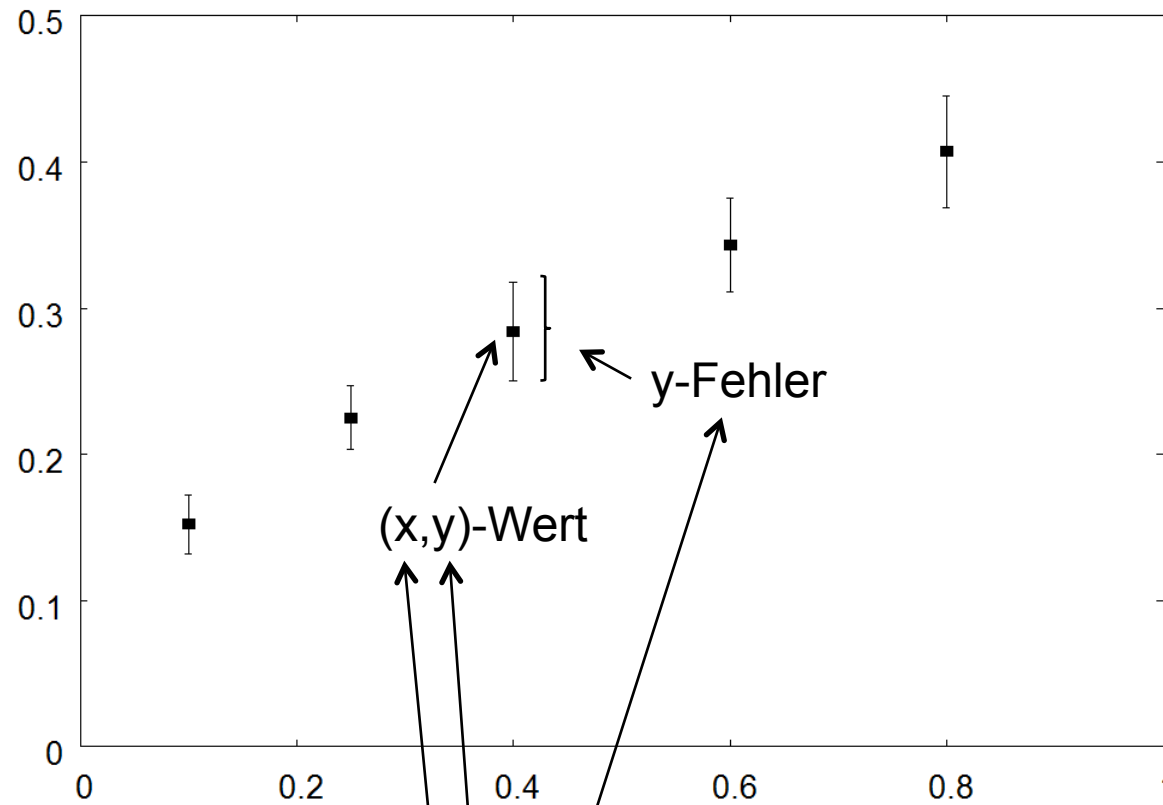
$$y(x') = a \cdot x' + c$$



Logarithmische Auftragung der Periodendauer  $T$  für verschiedene Pendellängen  $L$  zeigt auch einen linearen Verlauf

Die Steigung liefert  $L^a$  mit  $a=0.5$  falls der Zusammenhang  $T \sim \sqrt{L}$  erfüllt ist!

# Darstellung von Fehlern

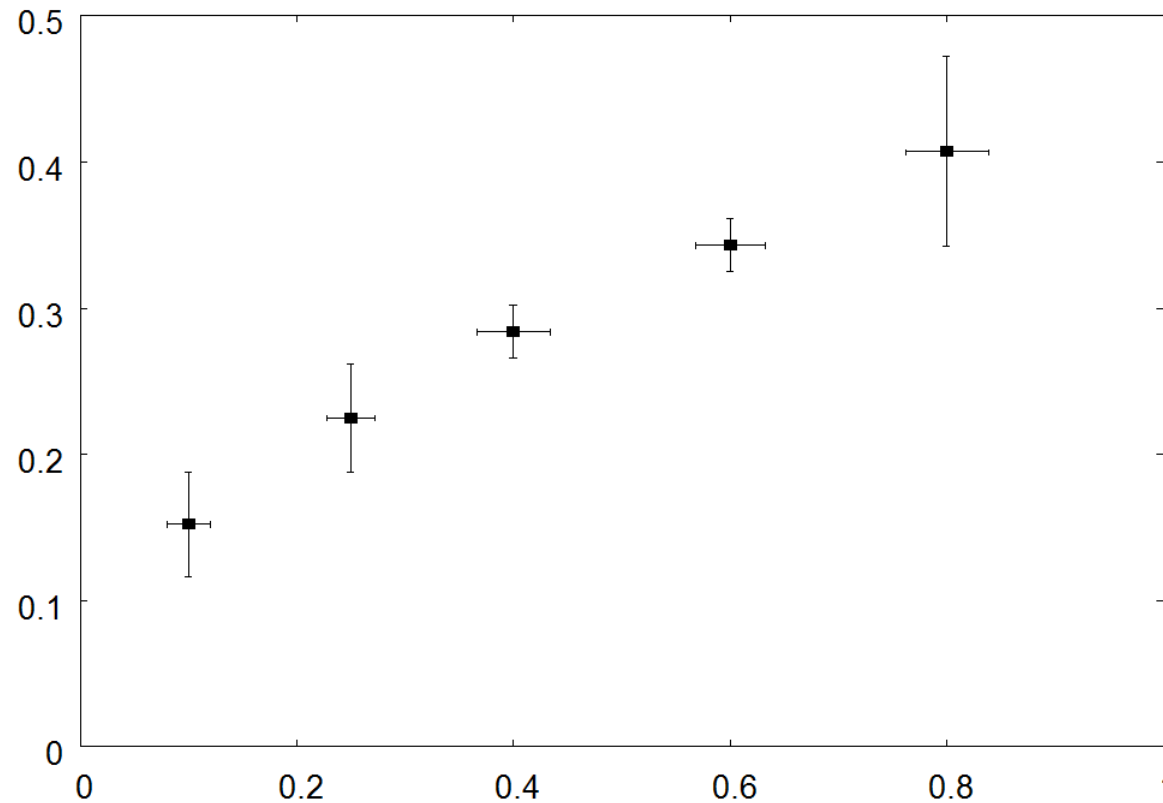


**Fehler auf den y-Werten:**

plot "datendatei" u 1:2:3 with errors



## Darstellung von Fehlern



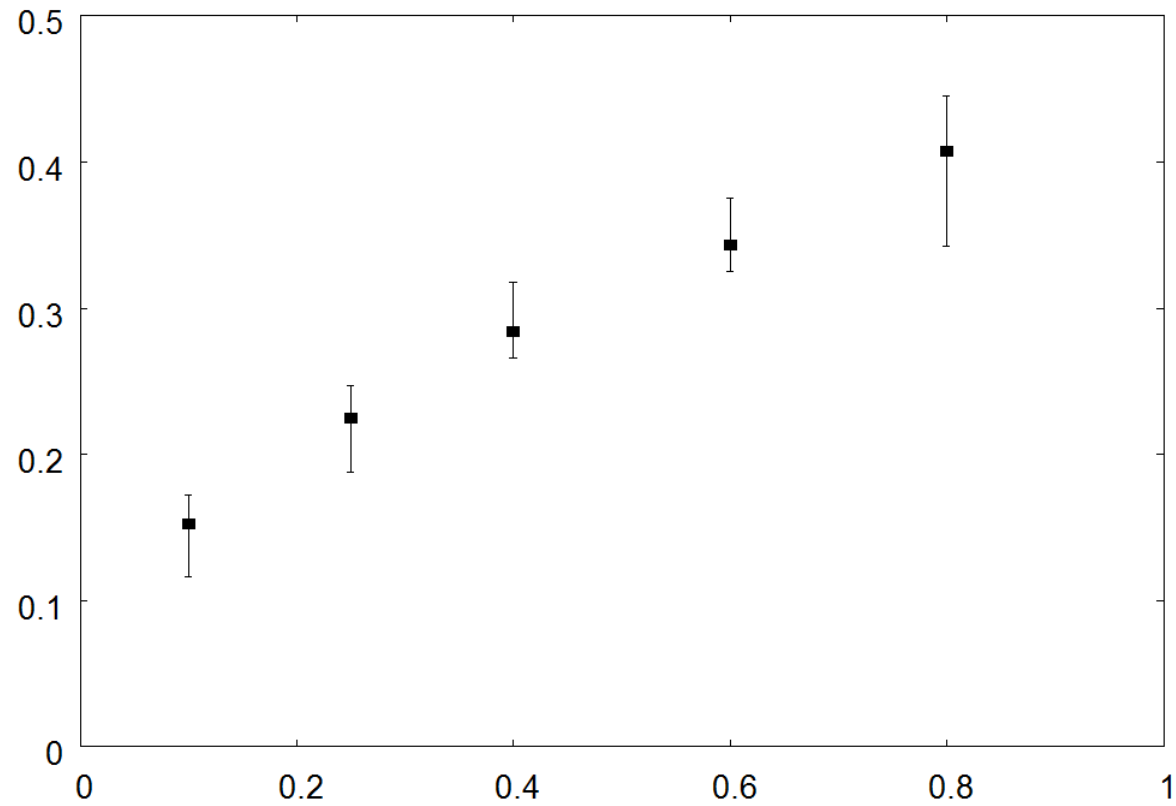
**Fehler auf den x- und y-Werten ( $x$ ,  $y$ ,  $\Delta x$ ,  $\Delta y$ ):**

plot "datendatei" u 1:2:3:4 with xyerrors

x-Fehler

y-Fehler

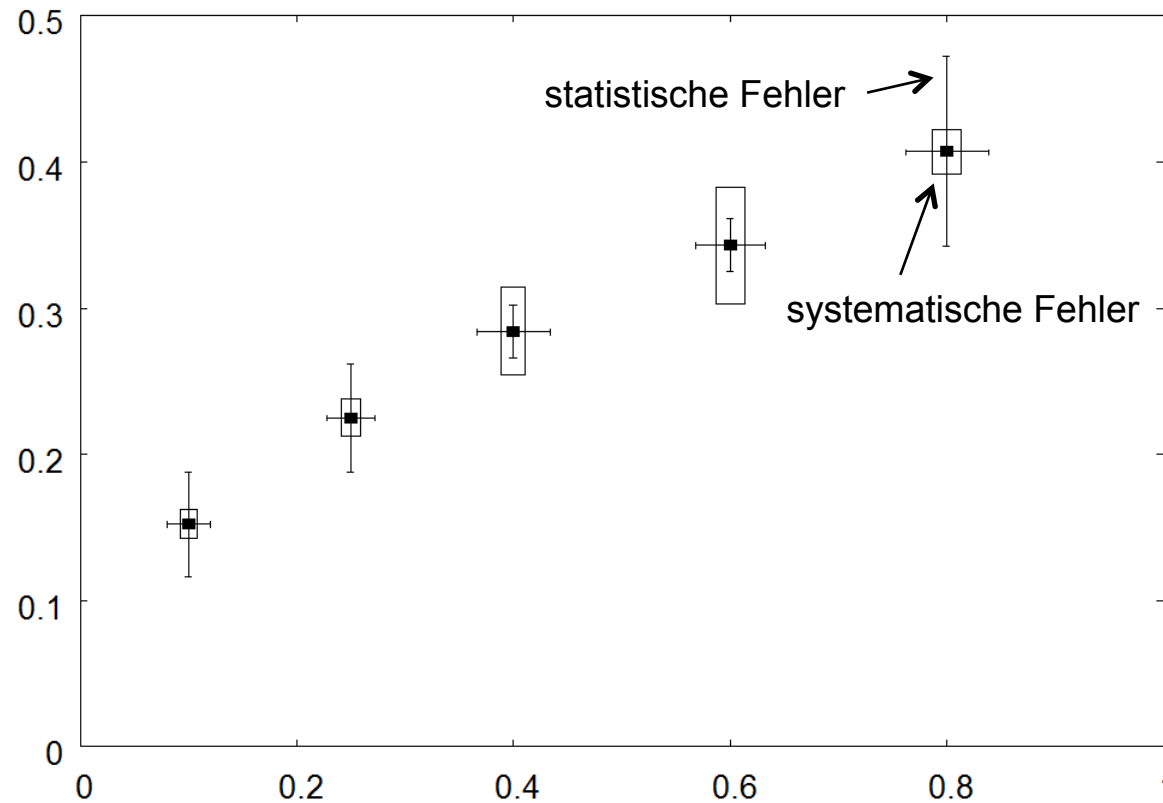
## Darstellung von Fehlern



**Fehler auf den y-Werten ( $x$ ,  $y$ ,  $y_{\min}$ ,  $y_{\max}$ ):**

plot "datendatei" u 1:2:(\$2-\$3):(\$2+\$4) with errors

# Darstellung von Fehlern



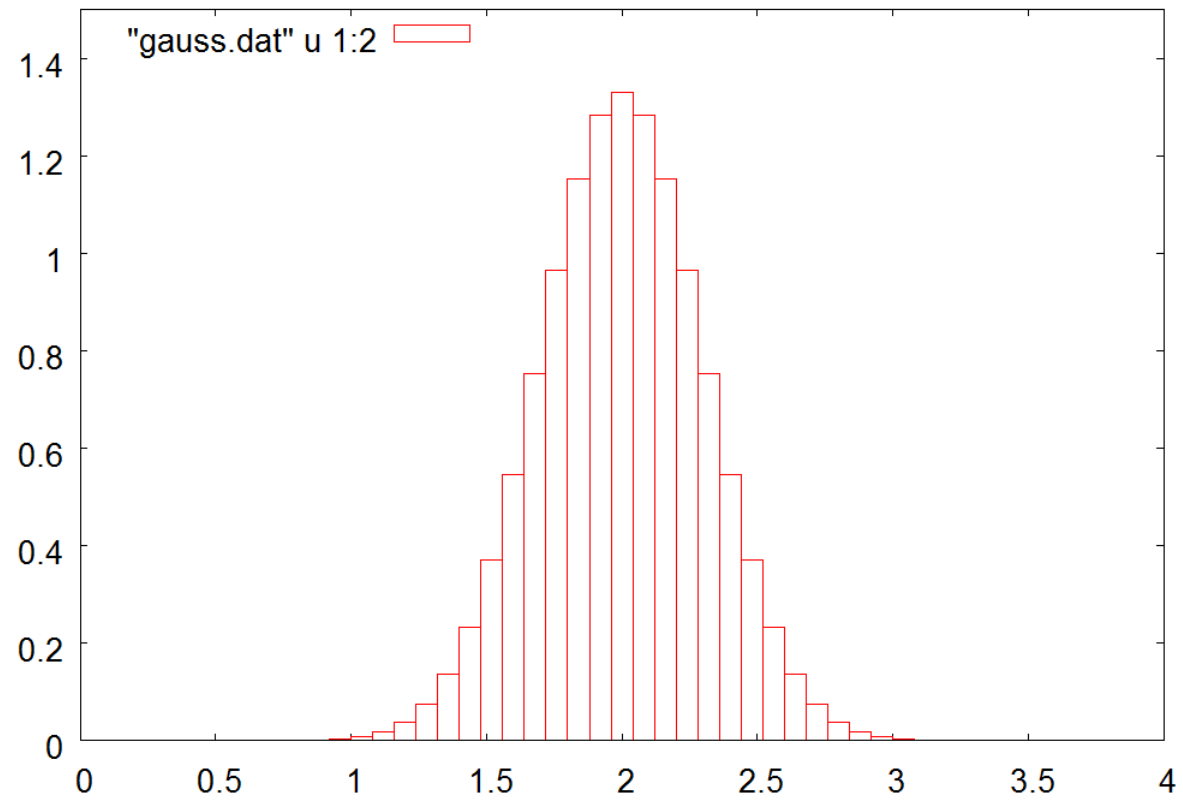
**Fehler auf den x- und y-Werten (x, y, Δx, Δy):**

plot "datendatei" u 1:2:3:4 with xyerrors

**und systematische Fehler:**

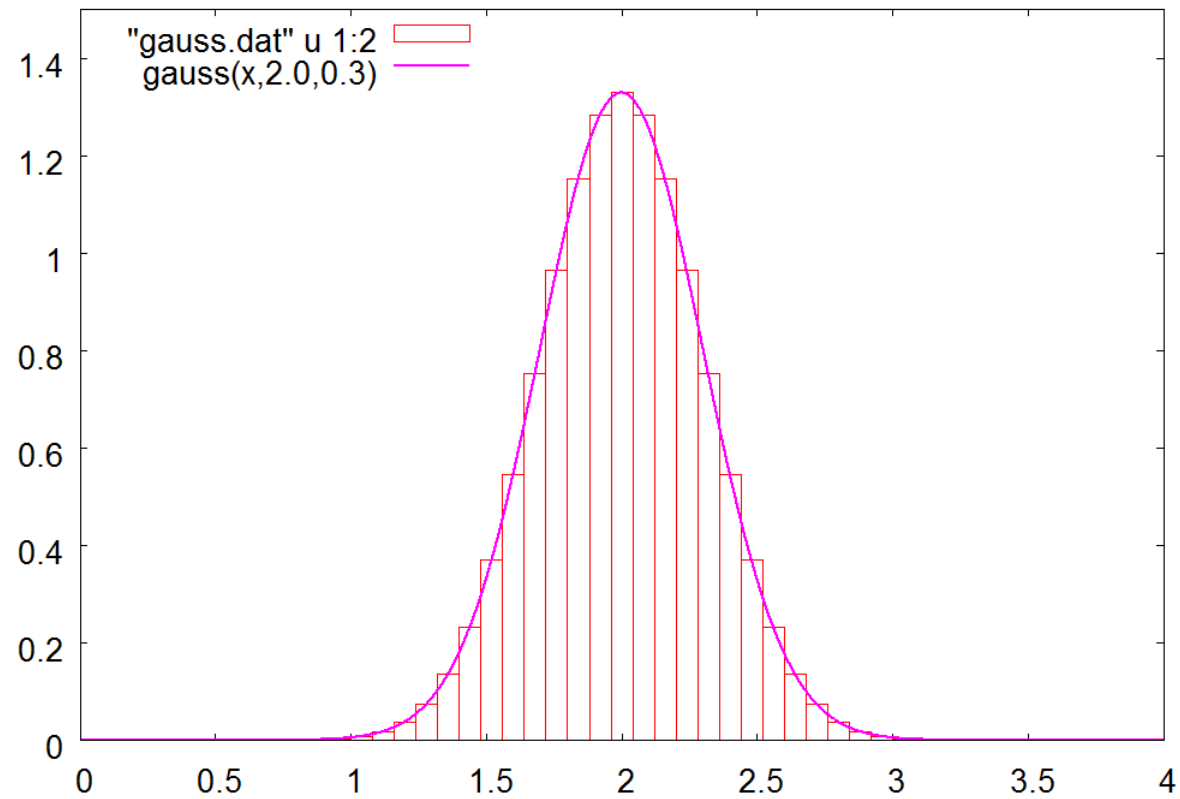
replot "datendatei" u 1:2:5:6 with boxxyerror

# Histogramme



plot "datendatei" u 1:2 with boxes

# Histogramme

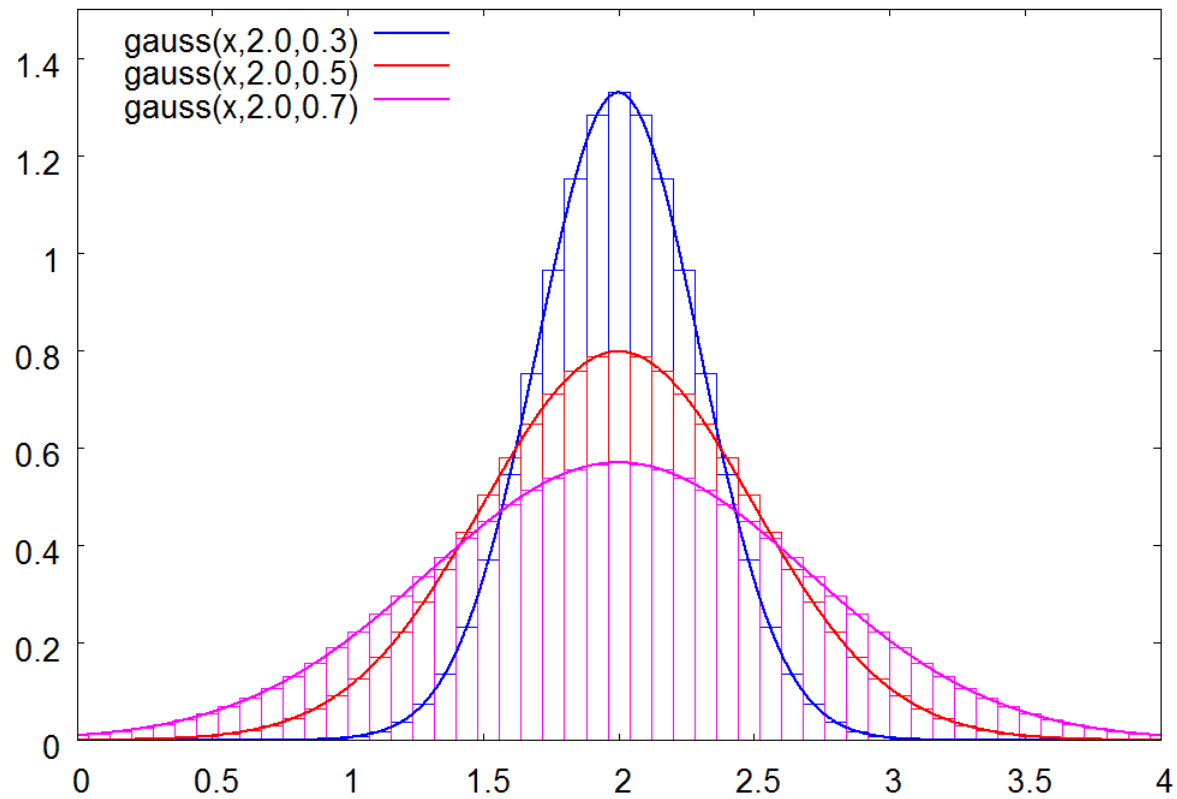


plot "datendatei" u 1:2 with boxes

$$\text{gauss}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

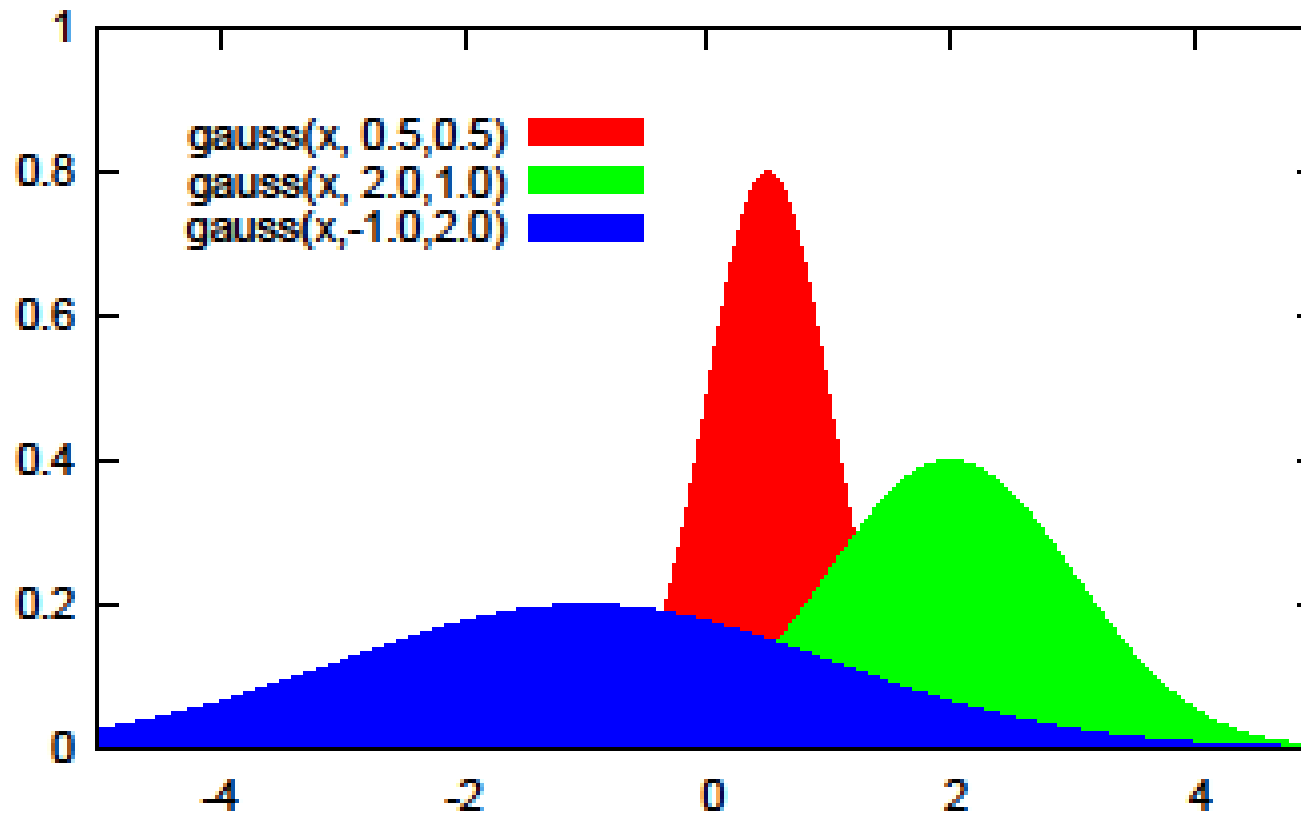
replot gauss(x,2.0,3.0) w l

# Histogramme



$$\text{gauss}(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

## gefüllte Kurven

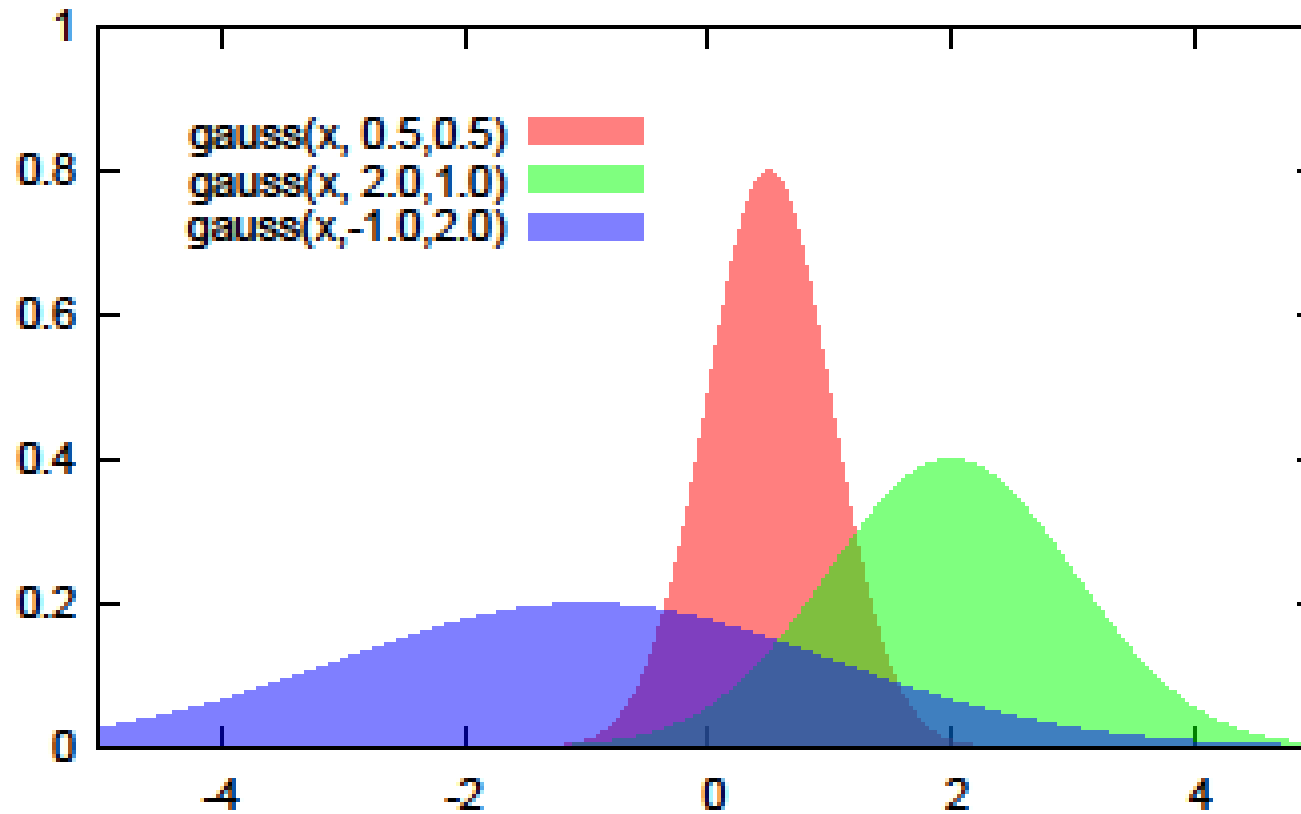


set style function filledcurves y1=0

plot gauss(x,0.5,0.5)

.....

## gefüllte Kurven



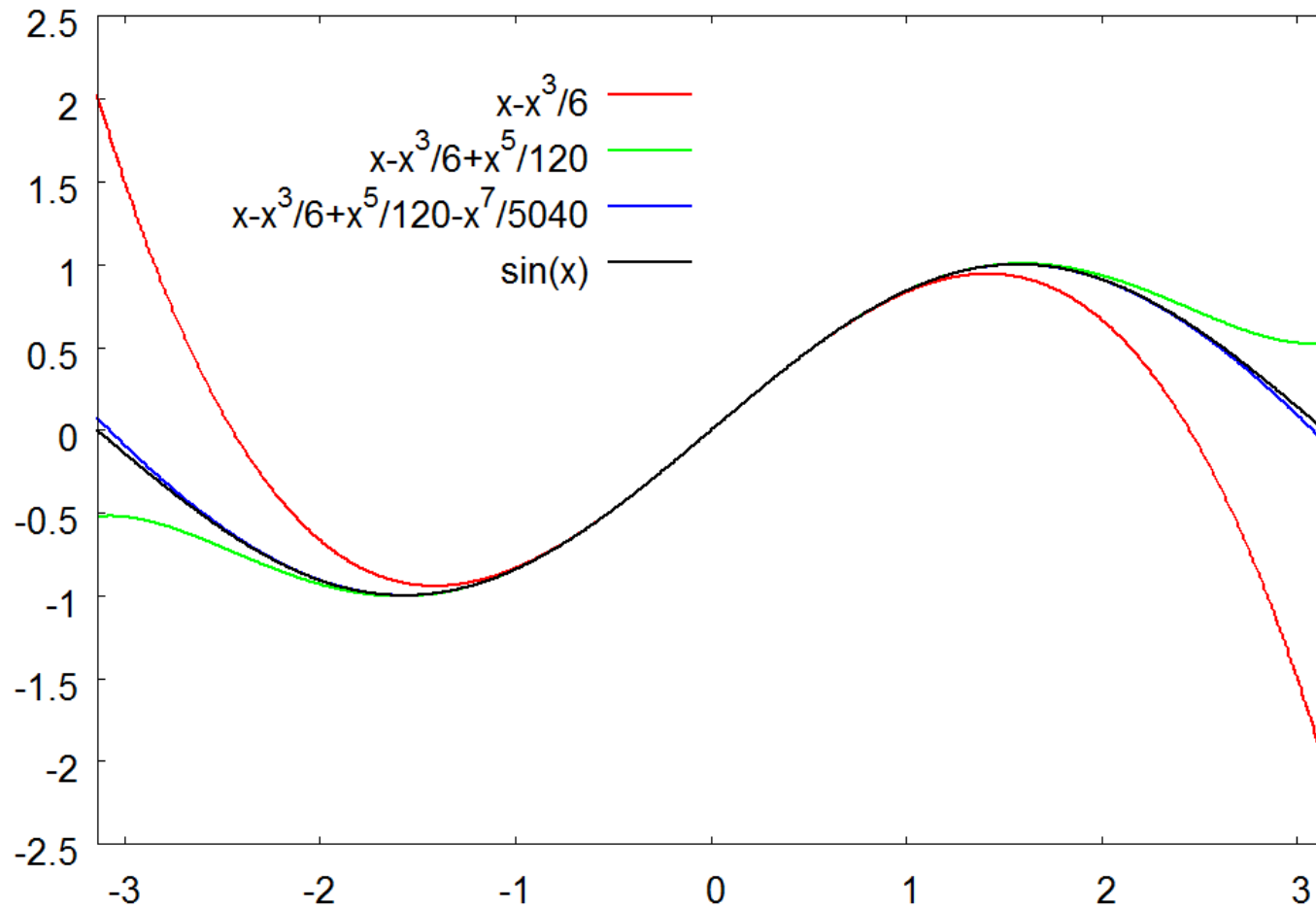
```
set style function filledcurves y1=0
```

```
set style fill transparent solid 0.5 noborder
```

```
replot
```

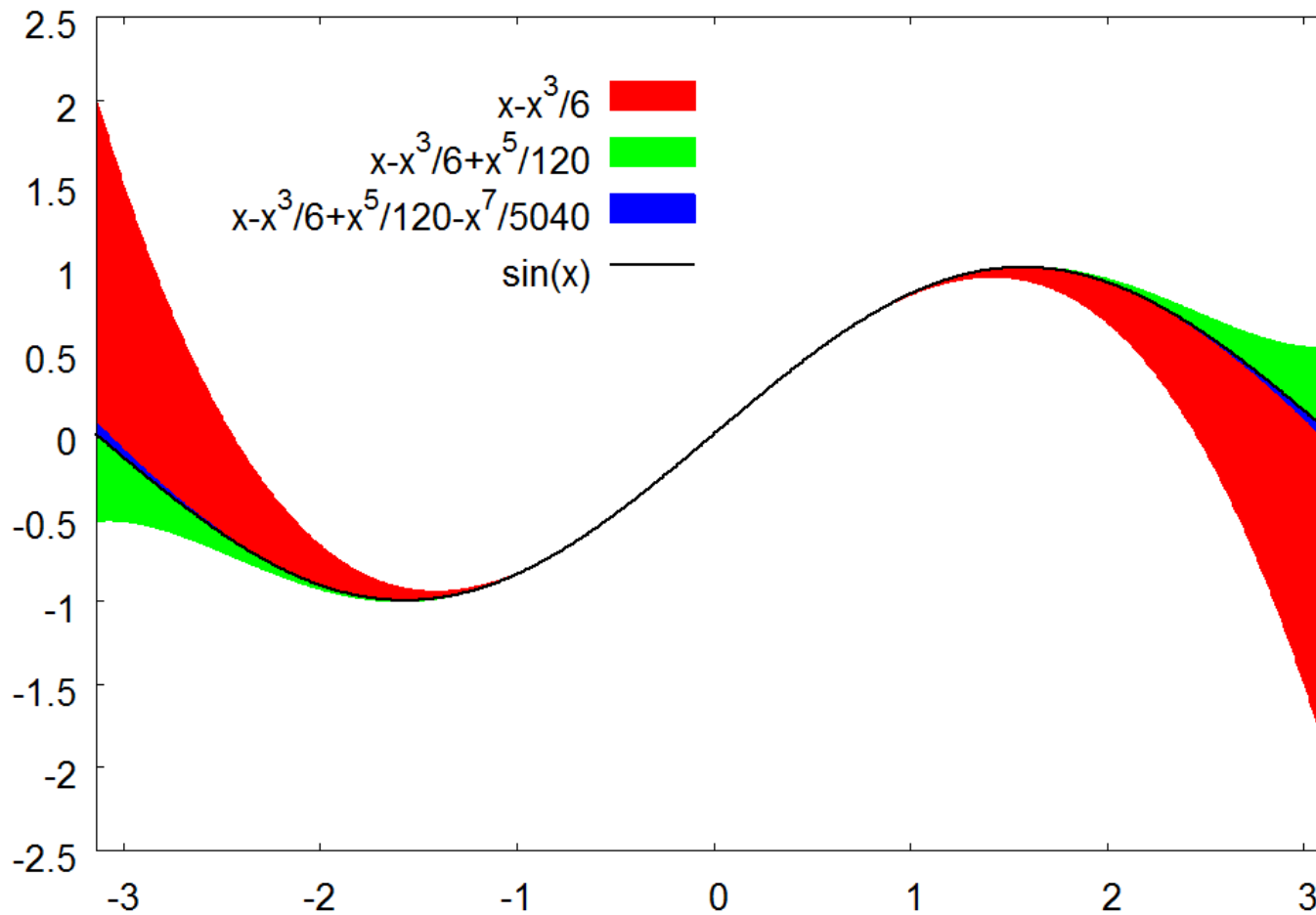


## gefüllte Kurven



```
plot "x" u 1:(t1($1)) title "x-x^3/6" with lines lw 2
replot "x" u 1:(t2($1)) title "x-x^3/6+x^5/120" with lines lw 2
replot "x" u 1:(t3($1)) title "x-x^3/6+x^5/120-x^7/5040" with lines lw 2
replot "x" u 1:(sin($1)) title "sin(x)" w l lw 2 lt -1
```

## gefüllte Kurven

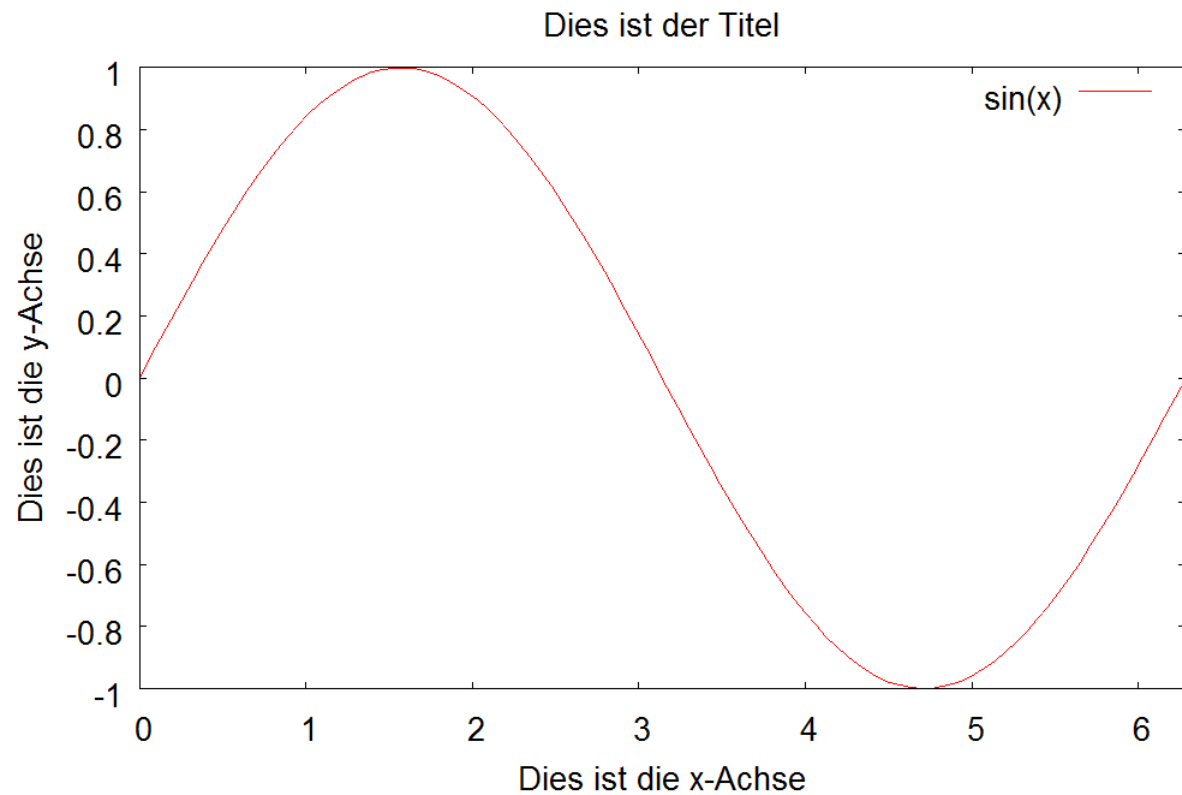


```
plot "x" u 1:(sin($1)):(t1($1)) title "x-x^3/6" with filledcurve
replot "x" u 1:(sin($1)):(t2($1)) title "x-x^3/6+x^5/120" with filledcurve
replot "x" u 1:(sin($1)):(t3($1)) title "x-x^3/6+x^5/120-x^7/5040" with filledcurve
replot "x" u 1:(sin($1)) title "sin(x)" w l lw 2 lt -1
```

# Achsenbeschriftungen

Achsenbeschriftungen über `title`, `xlabel`, `ylabel` werden ausserhalb des Plots angezeigt

```
set term windows 20  
  
set xr [0:2*pi]  
  
set title "Dies ist der Titel"  
set xlabel "Dies ist die x-Achse"  
set ylabel "Dies ist die y-Achse"  
  
set key right  
  
plot sin(x) w l lw 1.5 lt 1
```



# Beschriftungen

Beschriftungen über `label` können an beliebige Positionen geschrieben werden!

```
set term windows 20
```

```
set xr [0:2*pi]
```

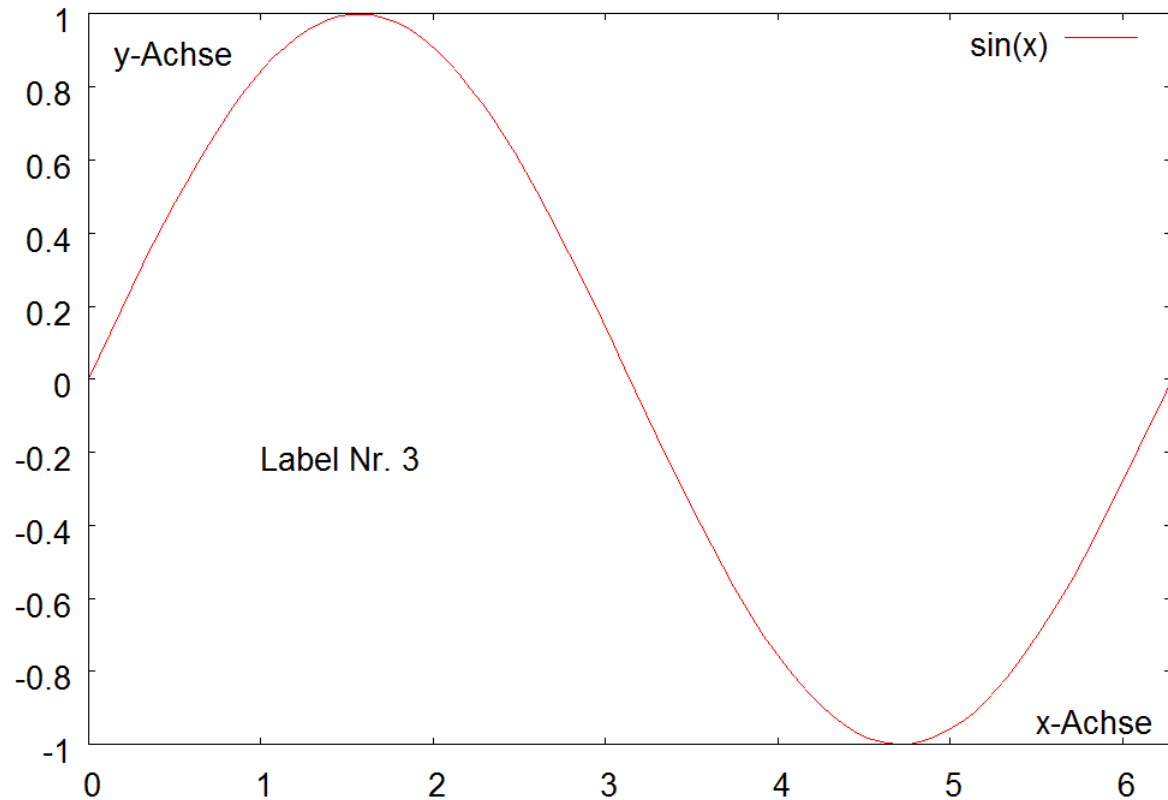
```
set label 1 "y-Achse" at 0.15,0.91
```

```
set label 2 "x-Achse" at 5.5,-0.92
```

```
set label 3 "Label Nr. 3" at 1,-0.2
```

```
set key right
```

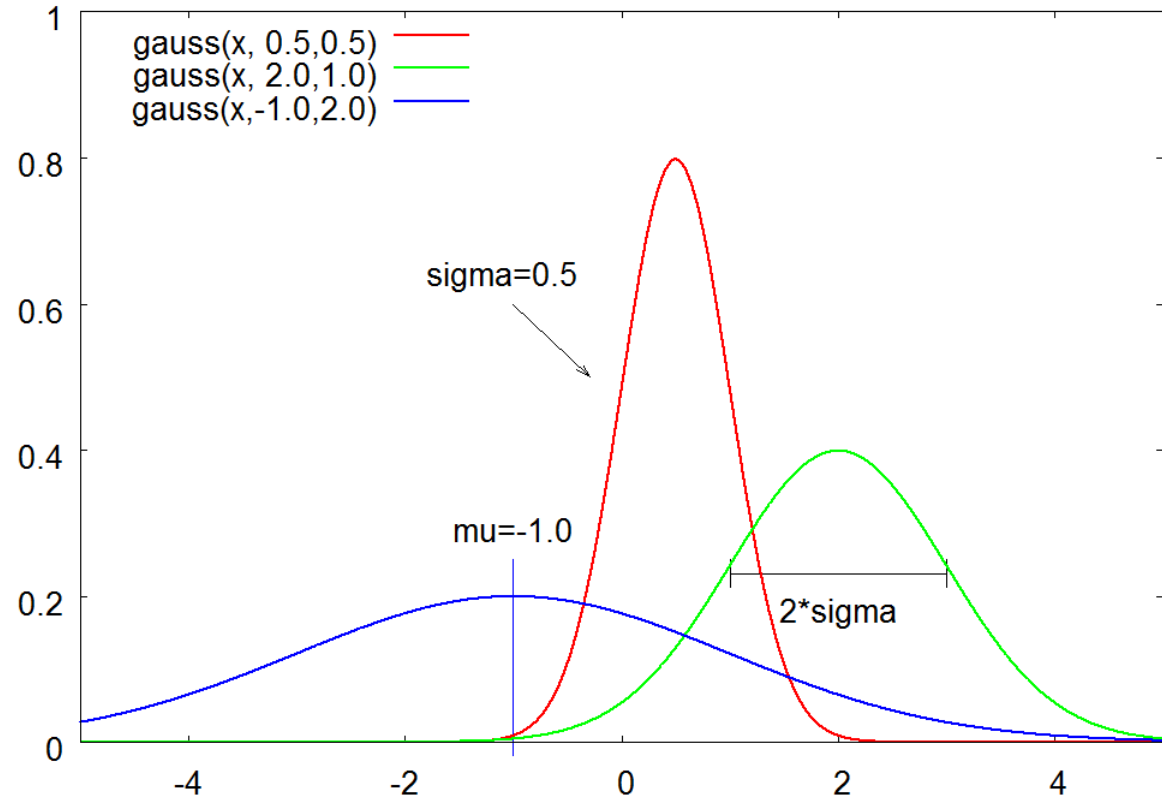
```
plot sin(x) w l lw 1.5 lt 1
```



# Beschriftungen

Beschriftungen über `label` können an beliebige Positionen geschrieben werden!

Pfeile und Linien können mit `arrow` eingefügt werden.



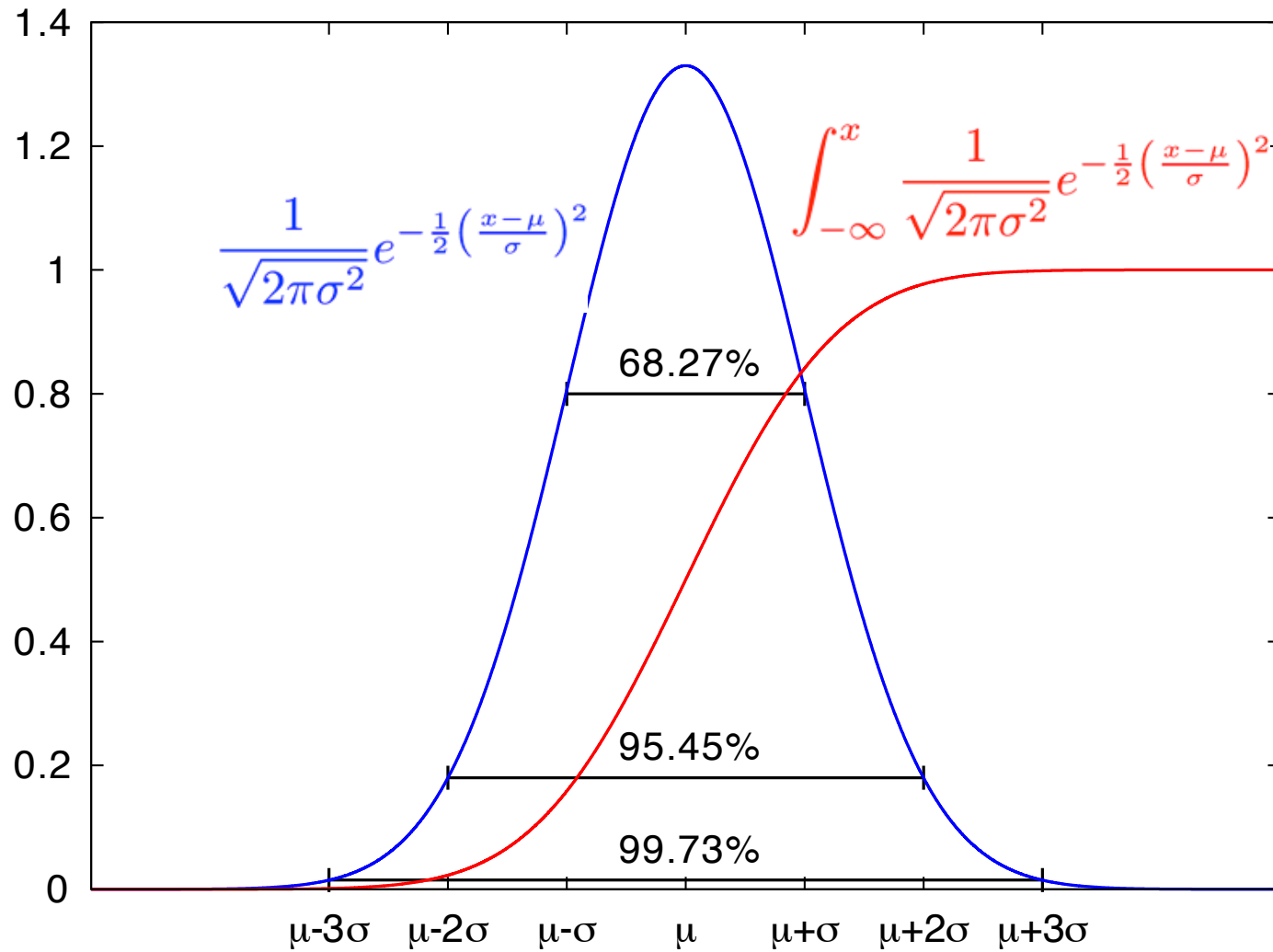
```
set arrow 1 from -1,0.6 to -0.3,0.5  
set label 1 "sigma=0.5" at -1.8,0.65
```

```
set arrow 2 from -1,-0.02 to -1,0.25 nohead linetype 3 lw 1.5  
set label 2 "mu=-1.0" at -1,0.3 center
```

```
set arrow 3 from 1.0,0.23 to 3.0,0.23 heads size 0.15,90  
set label 3 "2*sigma" at 2,0.19 center
```

# Skalen Beschriftungen

set xtics ( "Text" <Position> , ... , ... )



Bsp: set xtics ( "{/Symbol m}-3{/Symbol s}" 1.1, .... )

# Mehrere Skalen

```
set ylabel "y - Logarithmische Skala"  
set y2label "y2 - Lineare Skala (Power of 10)"
```

```
set x2tics ( "1/2" 0.5, "1" 1.0, "3/2" 1.5, "2" 2.0, "5/2" 2.5 )
```

```
set xlabel "x-Achse"  
set x2label "x2-Achse"
```

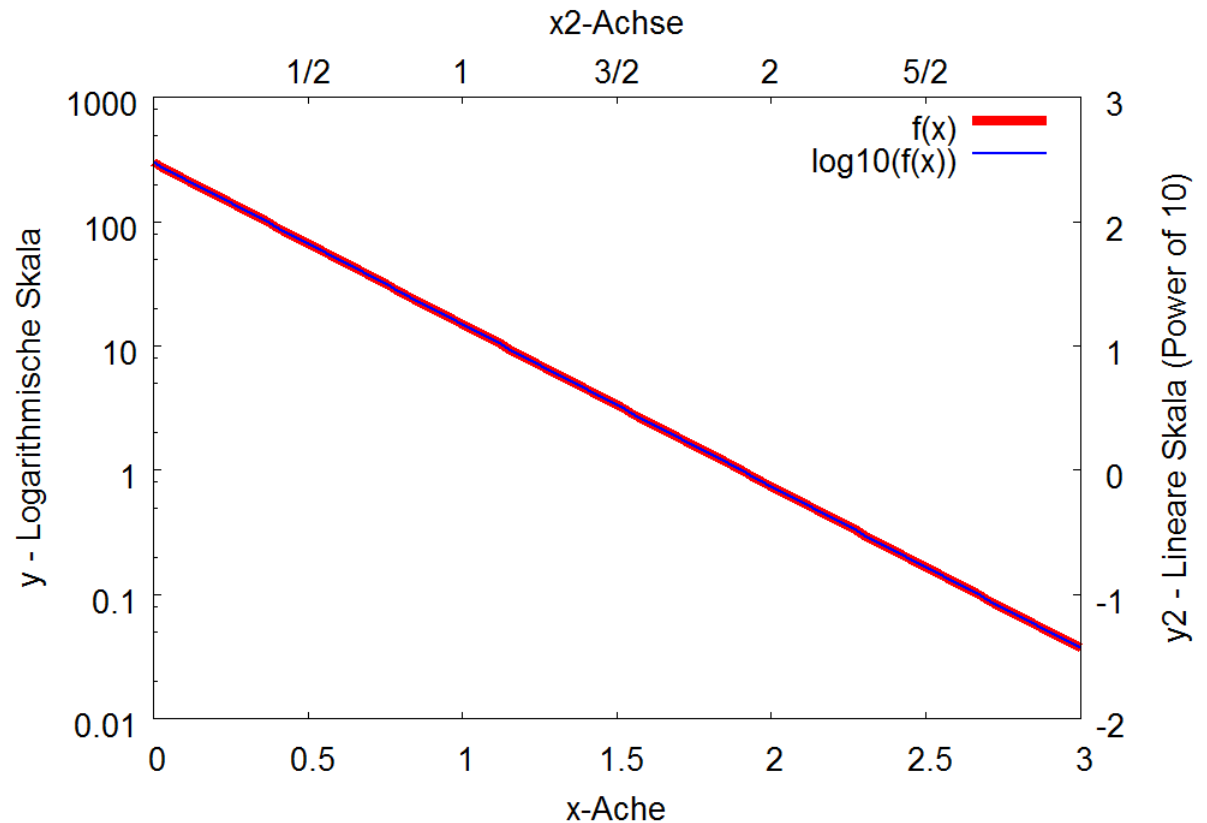
```
set log y  
unset log y2
```

```
set ytics nomirror  
set y2tics
```

```
set yrange [0.01:1000]  
set y2range [-2:3]
```

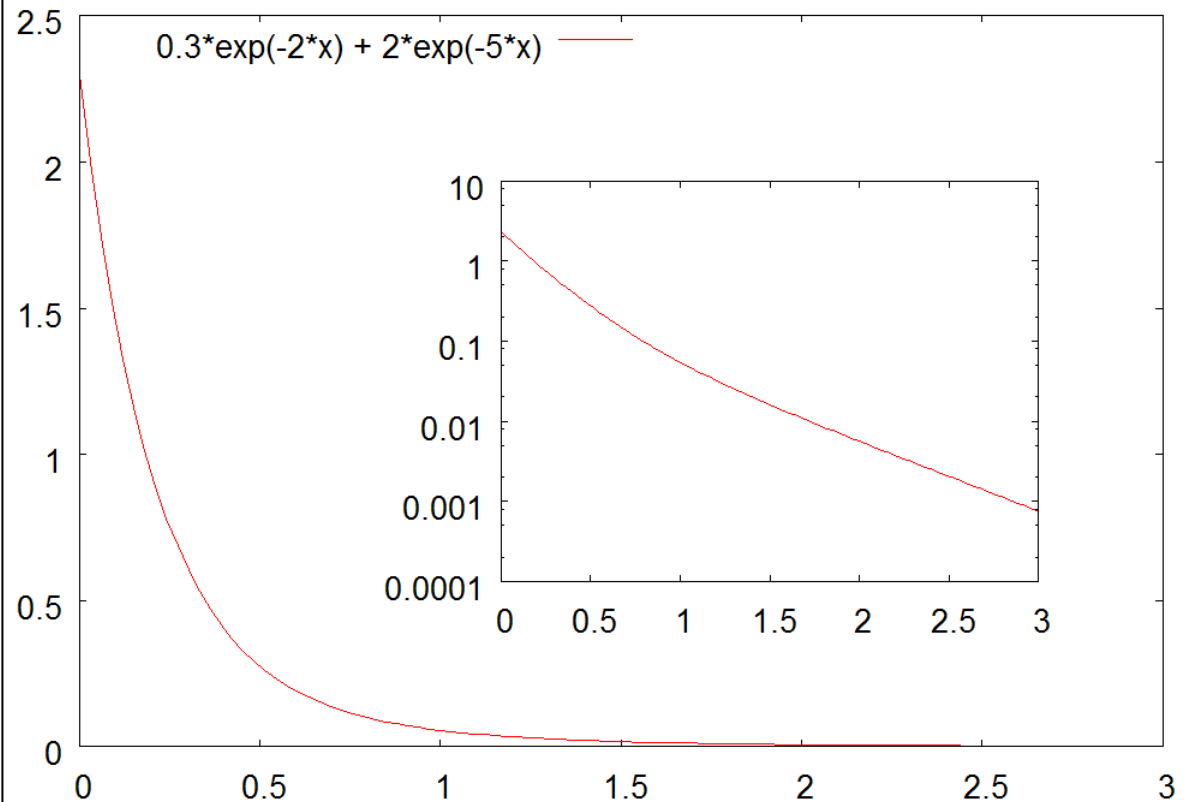
```
f(x) = 300*exp(-3*x)
```

```
plot f(x) axes x1y1 w l lw 8 lt 1  
rep log10(f(x)) axes x1y2 w l lw 2 lt 3
```



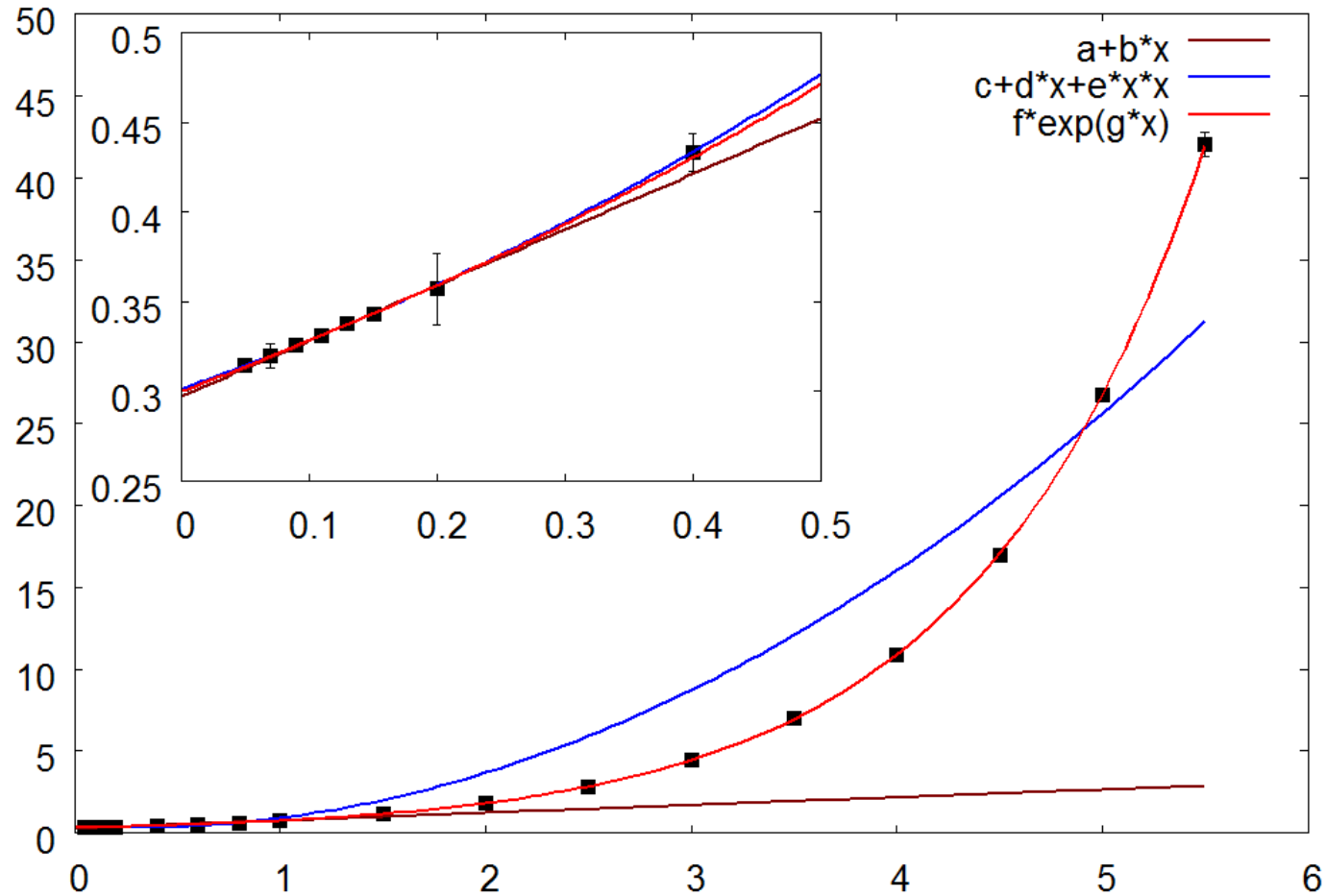
## Multiplot – mehrere Plots in einem Bild

```
set multiplot
# Plot 1 in gross
set key left
plot 0.3*exp(-2*x) + 2*exp(-5*x)
# Plot 2 in klein
set origin 0.3,0.2
set size 0.6,0.6
set log y
set nokey
plot 0.3*exp(-2*x) + 2*exp(-5*x)
```

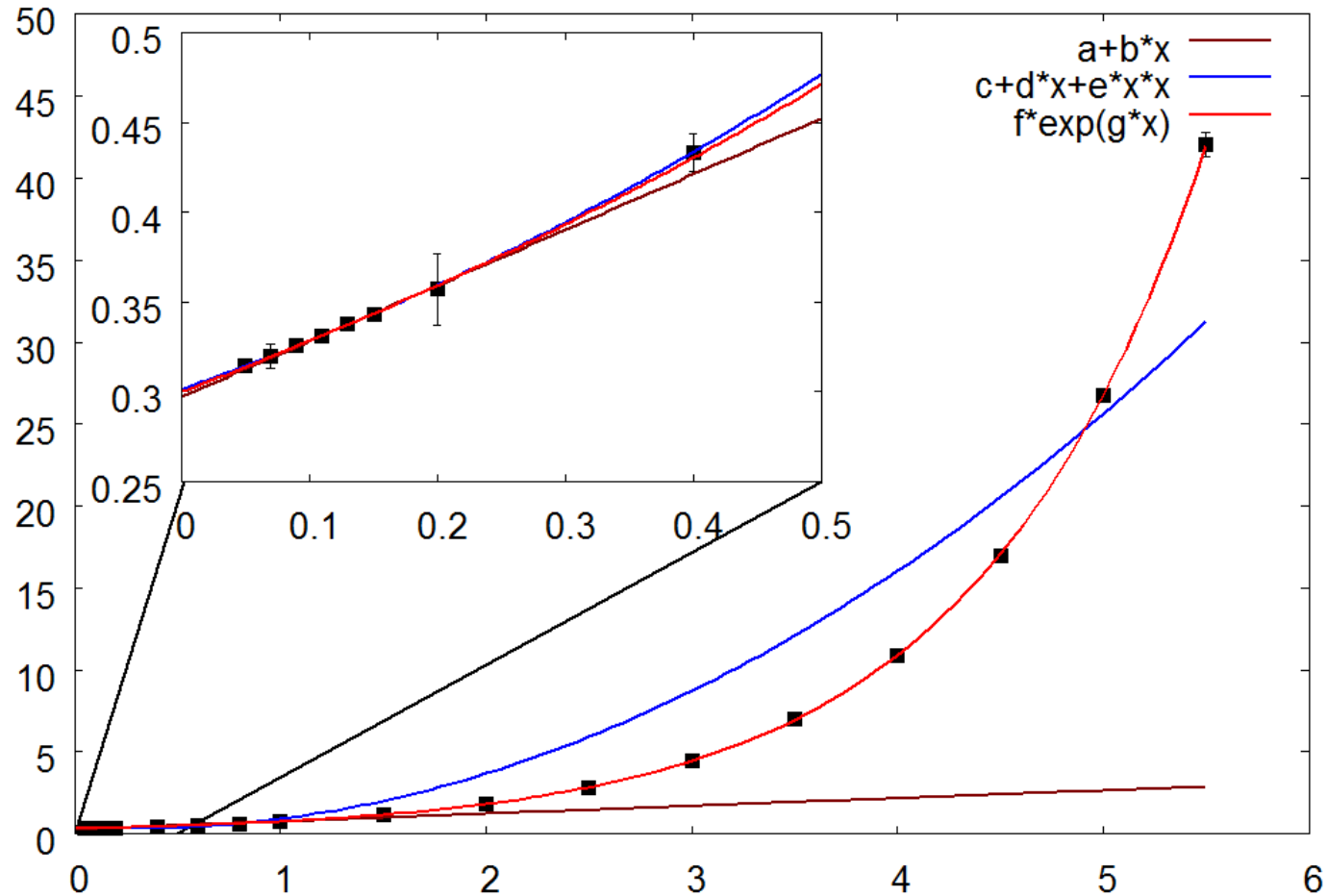




## Multiplot – mehrere Plots in einem Bild



## Multiplot – mehrere Plots in einem Bild



set arrow 1 from 0,0 to 0.53, 21.5 nohead lw 2  
set arrow 2 from 0.5,0 to 3.63, 21.5 nohead lw 2

## Multiplot – mehrere Plots in einem Bild

```
set multiplot title "Chebyshev Polynome" layout 3,2
```

```
t(x,n) = cos(n*acos(x))
```

```
set xr [-1:1]
```

```
set yr [-1:1]
```

```
set key left
```

```
plot t(x,1)
```

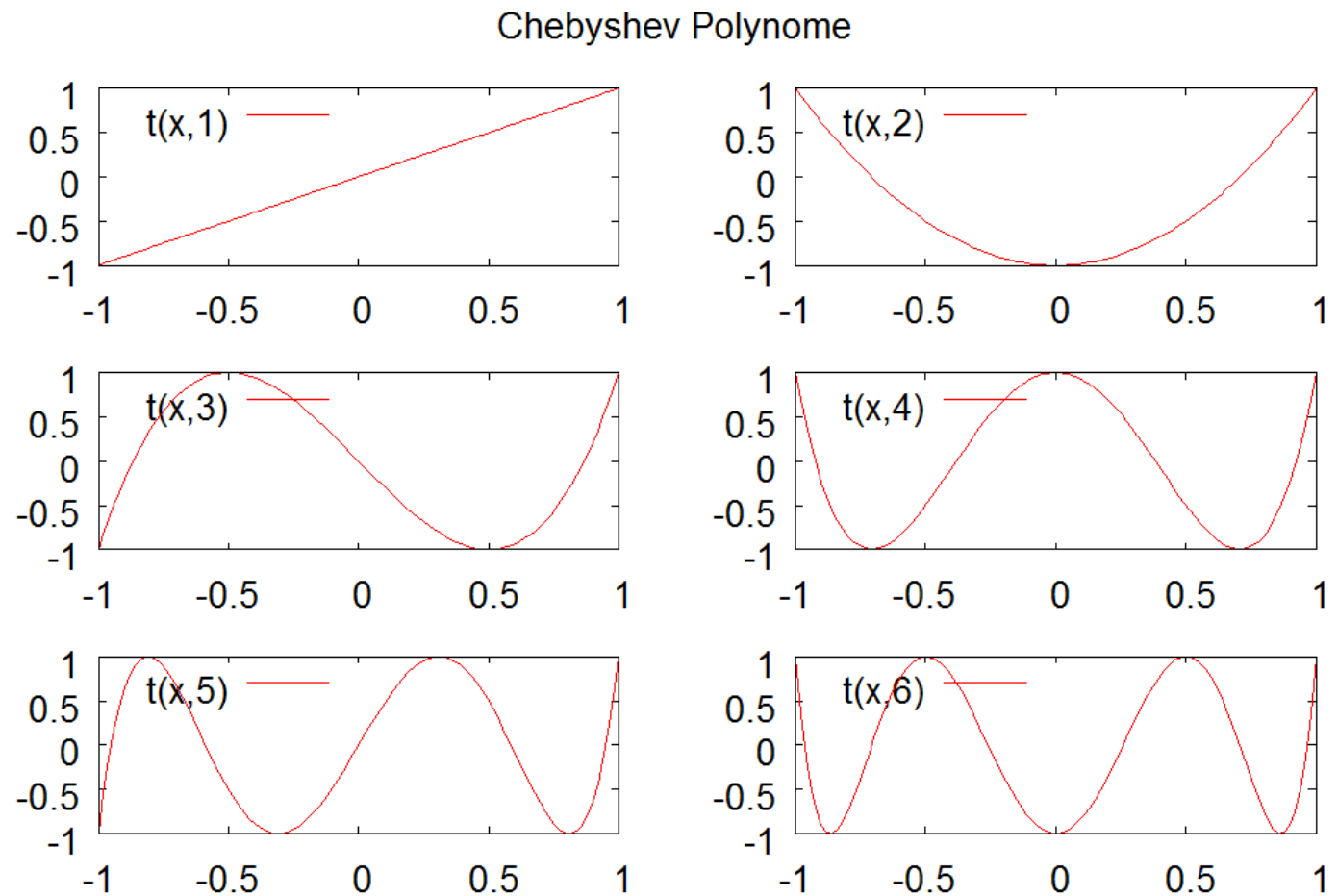
```
plot t(x,2)
```

```
plot t(x,3)
```

```
plot t(x,4)
```

```
plot t(x,5)
```

```
plot t(x,6)
```



# Multiplot – zwei Plots direkt untereinander

```
set multiplot title "Chebyshev Polynome" layout 2,1
```

```
t(x,n) = cos(n*acos(x))
```

```
set xr [-1:1]
```

```
set yr [-1:1]
```

```
set grid
```

```
# Plot oben
```

```
set bmargin 0
```

```
set format x ""
```

```
set ytics -0.5,0.5,1.0
```

```
set key bottom
```

```
plot t(x,1),t(x,2),t(x,3)
```

```
# Plot unten
```

```
set bmargin
```

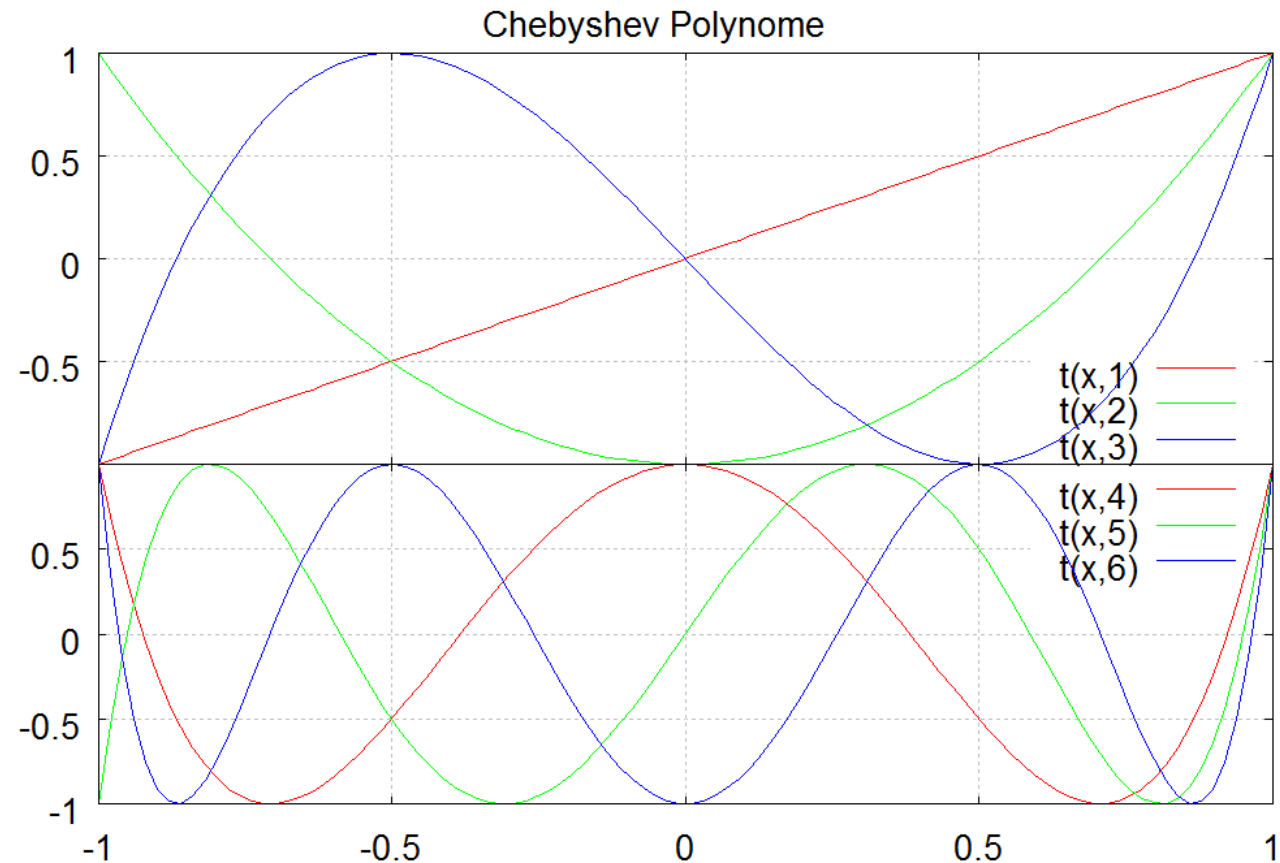
```
set tmargin 0
```

```
set format x "%g"
```

```
set ytics -1,0.5,0.5
```

```
set key top
```

```
plot t(x,4),t(x,5),t(x,6)
```



# Parametrische Plots

Anstatt einer Funktion  $y=f(x)$  wird hierbei eine Liste

$$(x,y) = (fx(t) ,fy(t)) , \text{ für } t=a\dots b$$

Mit getrennten Funktionen  $fx(t)$  und  $fy(t)$  für die x- und y-Koordinaten dargestellt

```
set size square
```

```
set parametric
```

```
set grid
```

```
set xr [-1.1:1.1]
```

```
set yr [-1.1:1.1]
```

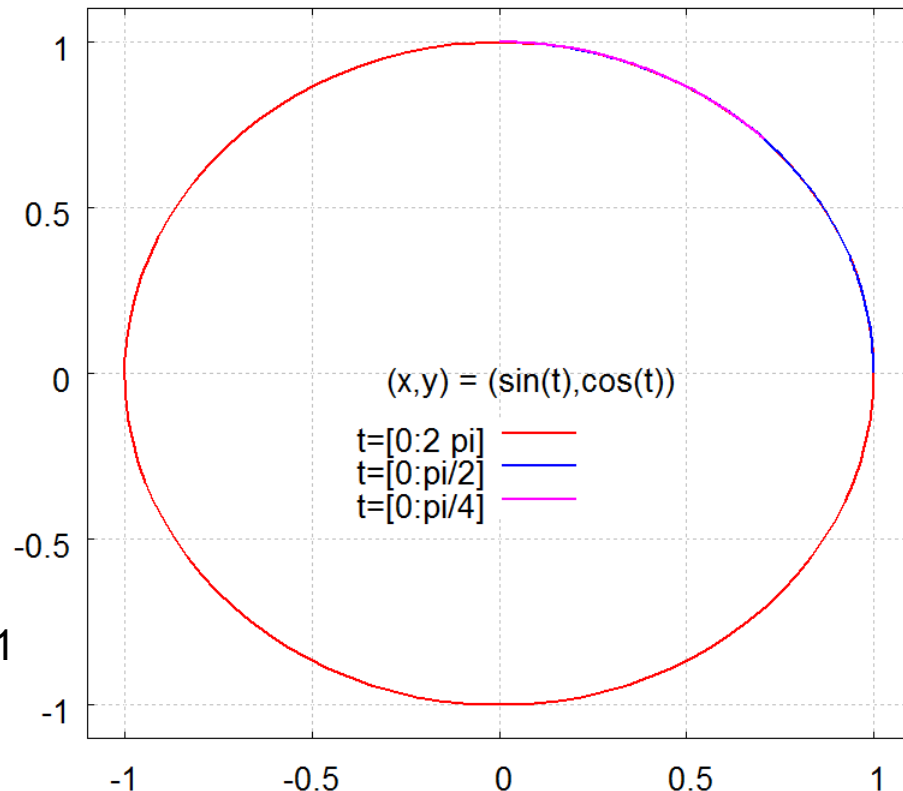
```
set label 1 "(x,y) = (sin(t),cos(t))" at -0.3,0.0
```

```
set key 0.25,-0.13
```

```
plot [0:2*pi] sin(t), cos(t) title "t=[0:2 pi]" w l lw 2 lt 1
```

```
rep sin(t/4.), cos(t/4.) title "t=[0:pi/2]" w l lw 2 lt 3
```

```
rep sin(t/8.), cos(t/8.) title "t=[0:pi/4]" w l lw 2 lt 4
```

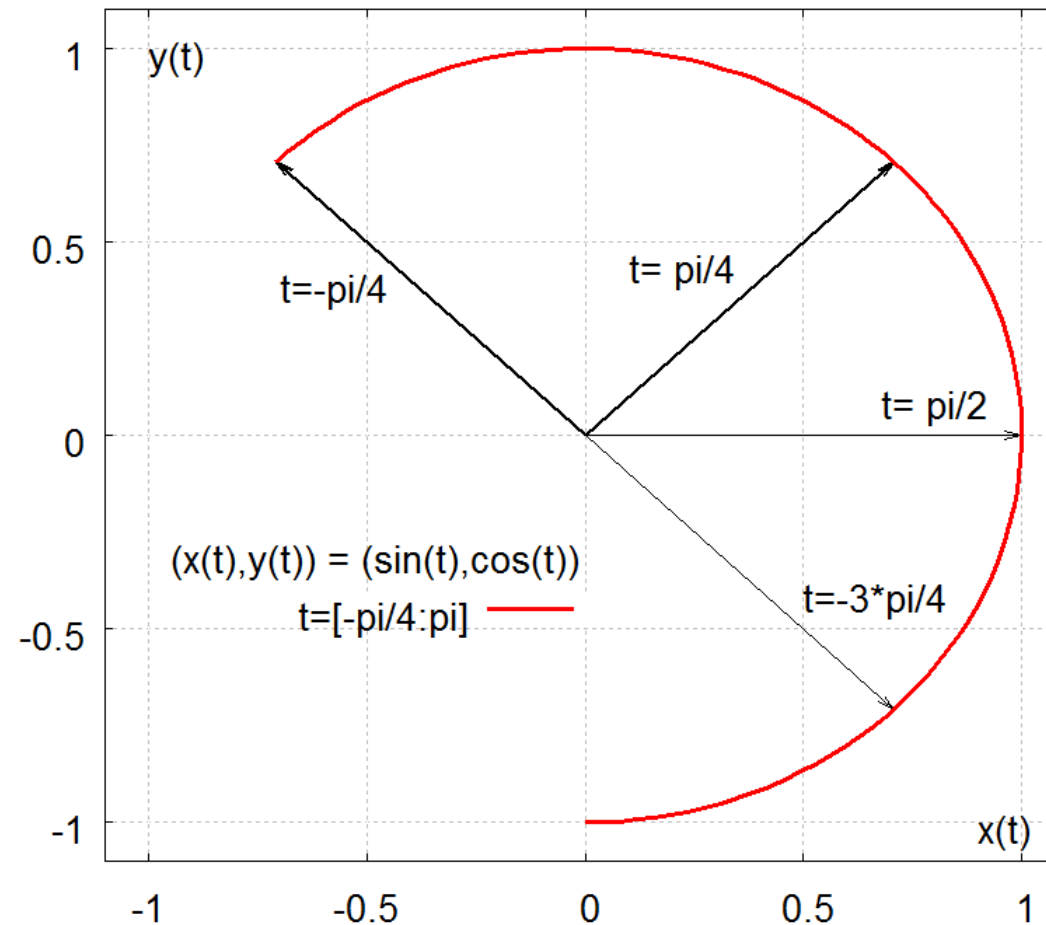


# Parametrische Plots

Anstatt einer Funktion  $y=f(x)$  wird hierbei eine Liste

$$(x,y) = (f_x(t), f_y(t)) , \text{ für } t=a\dots b$$

Mit getrennten Funktionen  $f_x(t)$  und  $f_y(t)$  für die x- und y-Koordinaten dargestellt



## Parametrische Plots

Anstatt einer Funktion  $y=f(x)$  wird hierbei eine Liste

$$(x,y) = (fx(t) ,fy(t)) , \text{ für } t=a\dots b$$

Mit getrennten Funktionen  $fx(t)$  und  $fy(t)$  für die x- und y-Koordinaten dargestellt

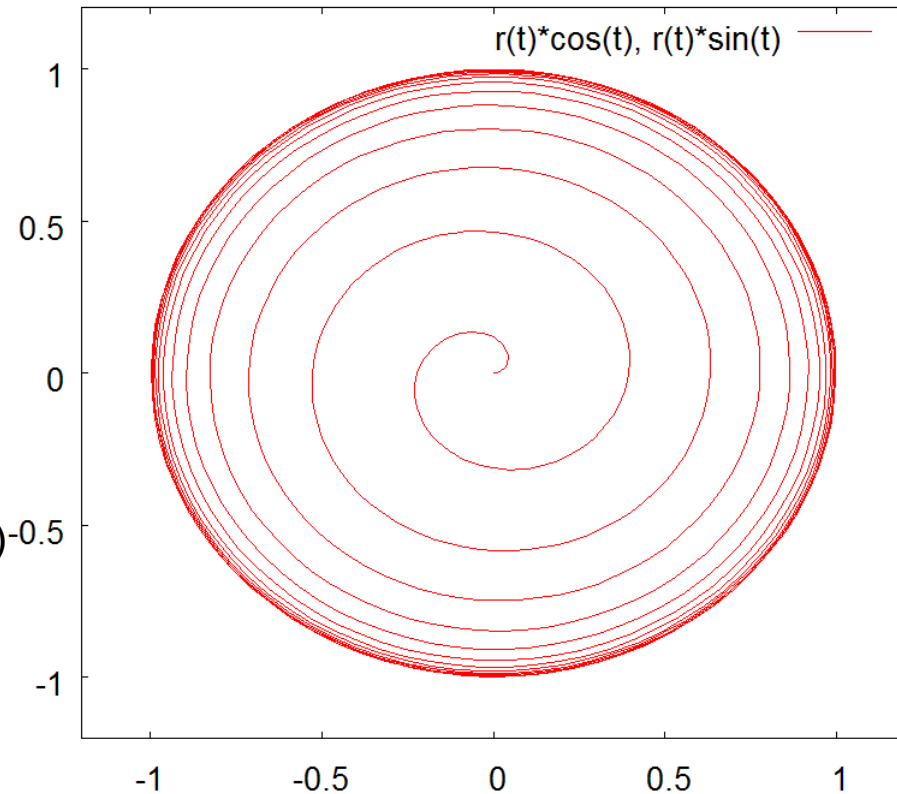
```
set size square
```

```
set parametric
```

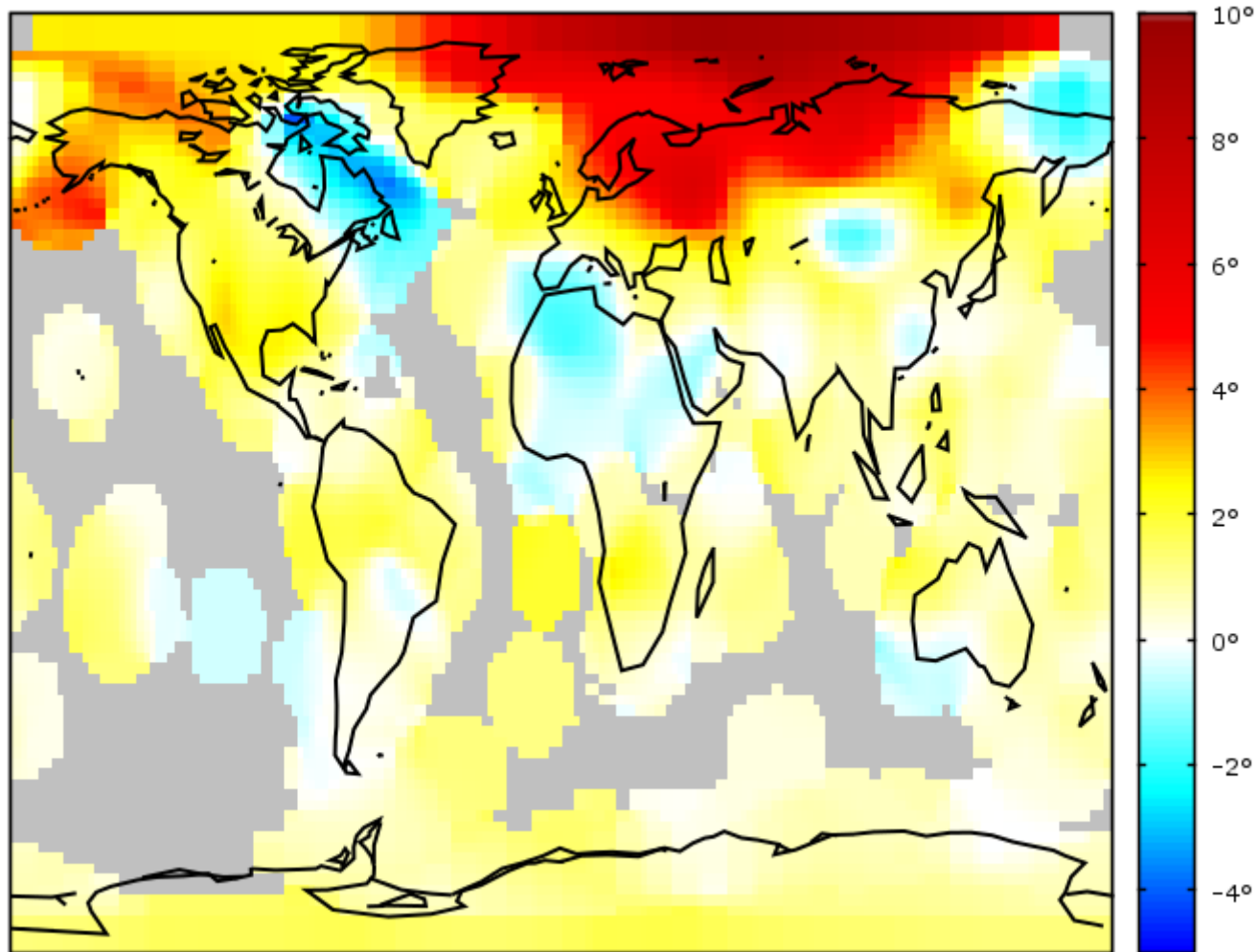
```
set samples 1000
```

```
r(t) = 1 - exp(-0.25*t/pi)
```

```
plot [0:25*pi][-1.2:1.2][-1.2:1.2] r(t)*cos(t), r(t)*sin(t)
```



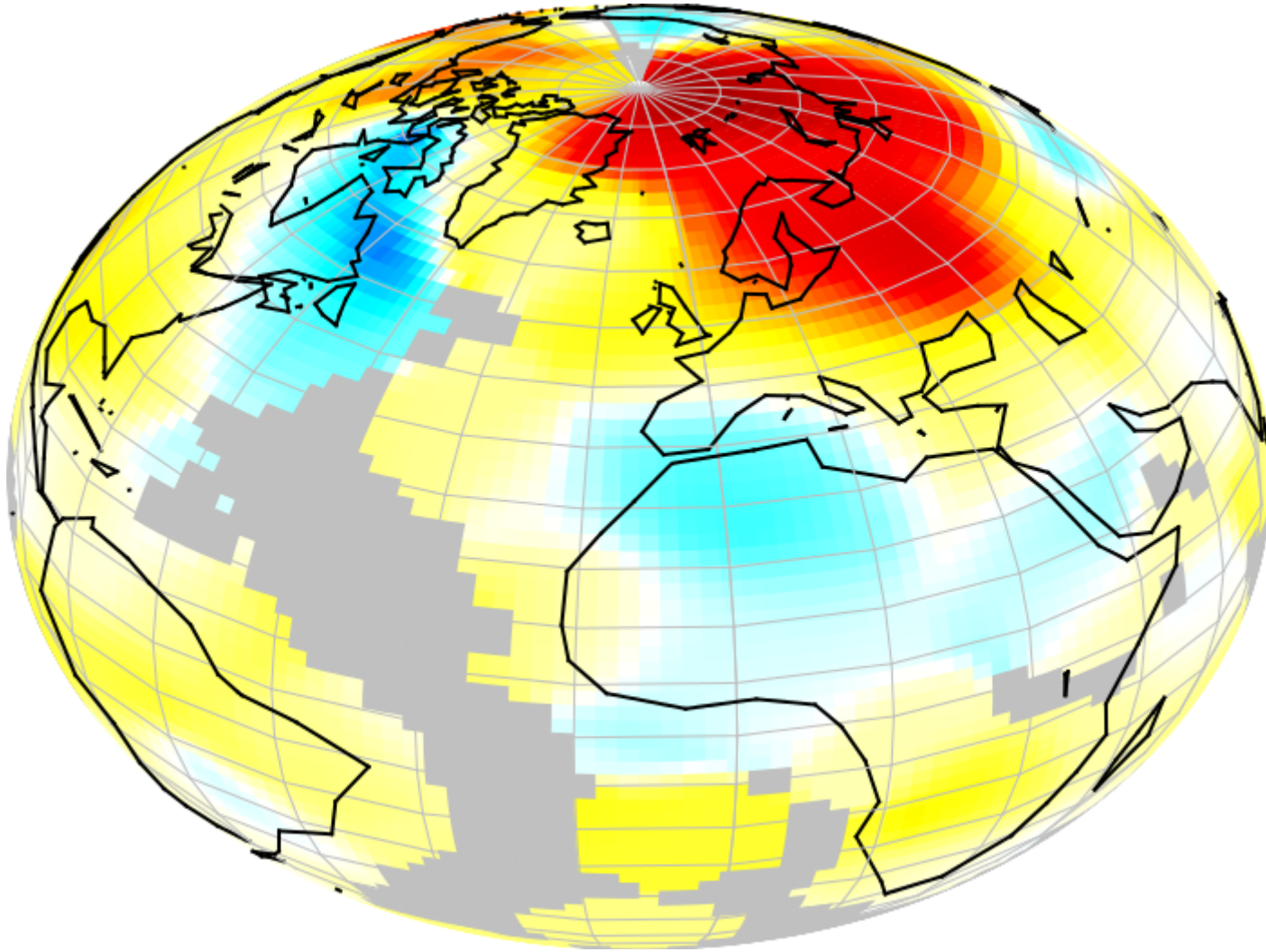
## 3d-Plots - Daten



Quelle: [www.gnuplotting.org](http://www.gnuplotting.org)



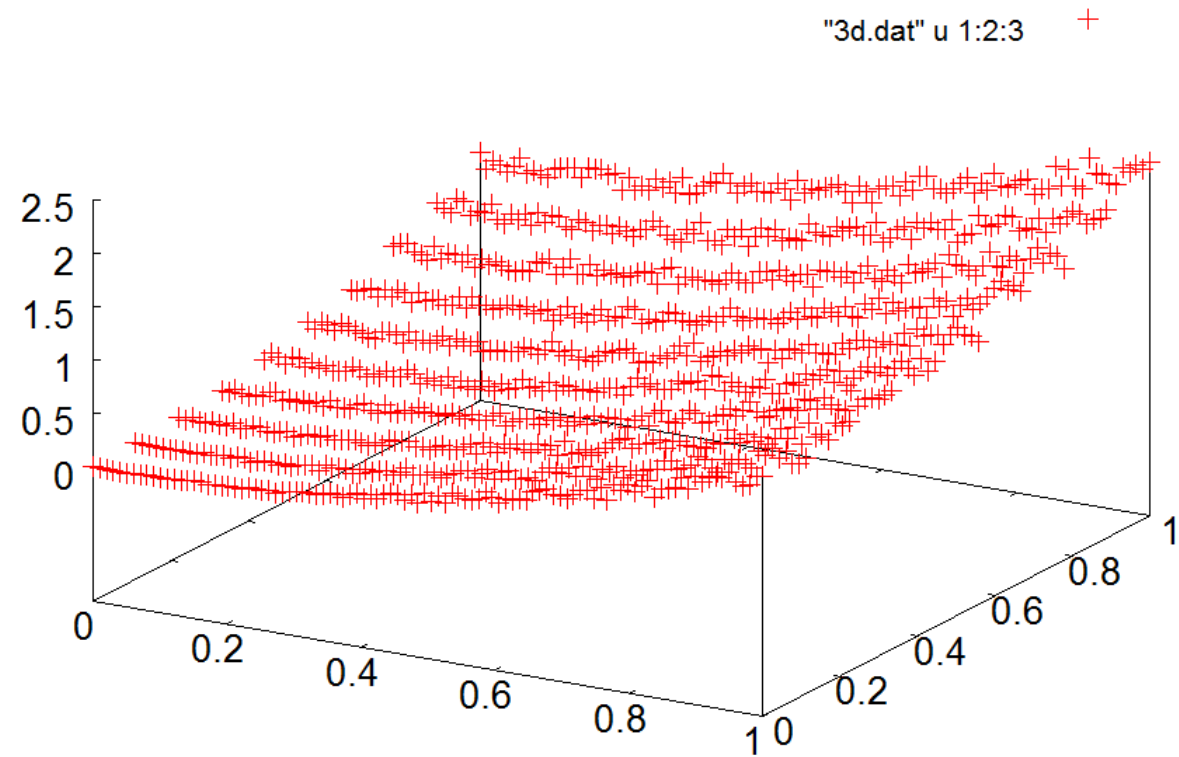
## 3d-Plots - Daten



Quelle: [www.gnuplotting.org](http://www.gnuplotting.org)

# 3d-Plots - Daten

splot "datafile" using 1:2:3



## 3d-Plots - Funktionen

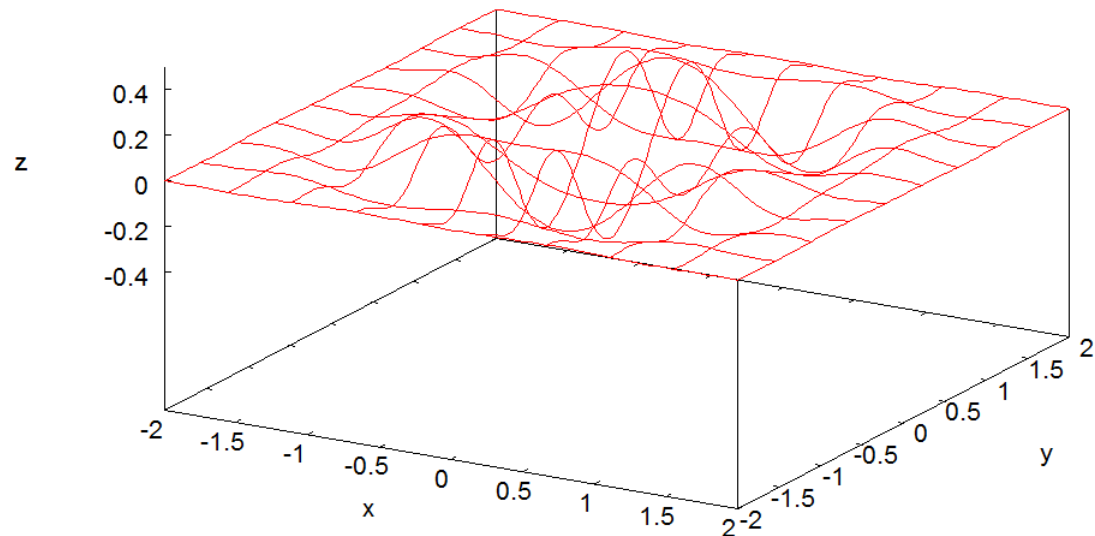
```
set xrange [-2:2]  
set yrange [-2:2]  
set zrange [-0.5:0.5]
```

```
set format cb "%.1f"
```

```
set xlabel "x"  
set ylabel "y"  
set zlabel "z"
```

```
splot exp(-(x**2 + y**2))*cos(x/4)*sin(y)*cos(2*(x**2+y**2))
```

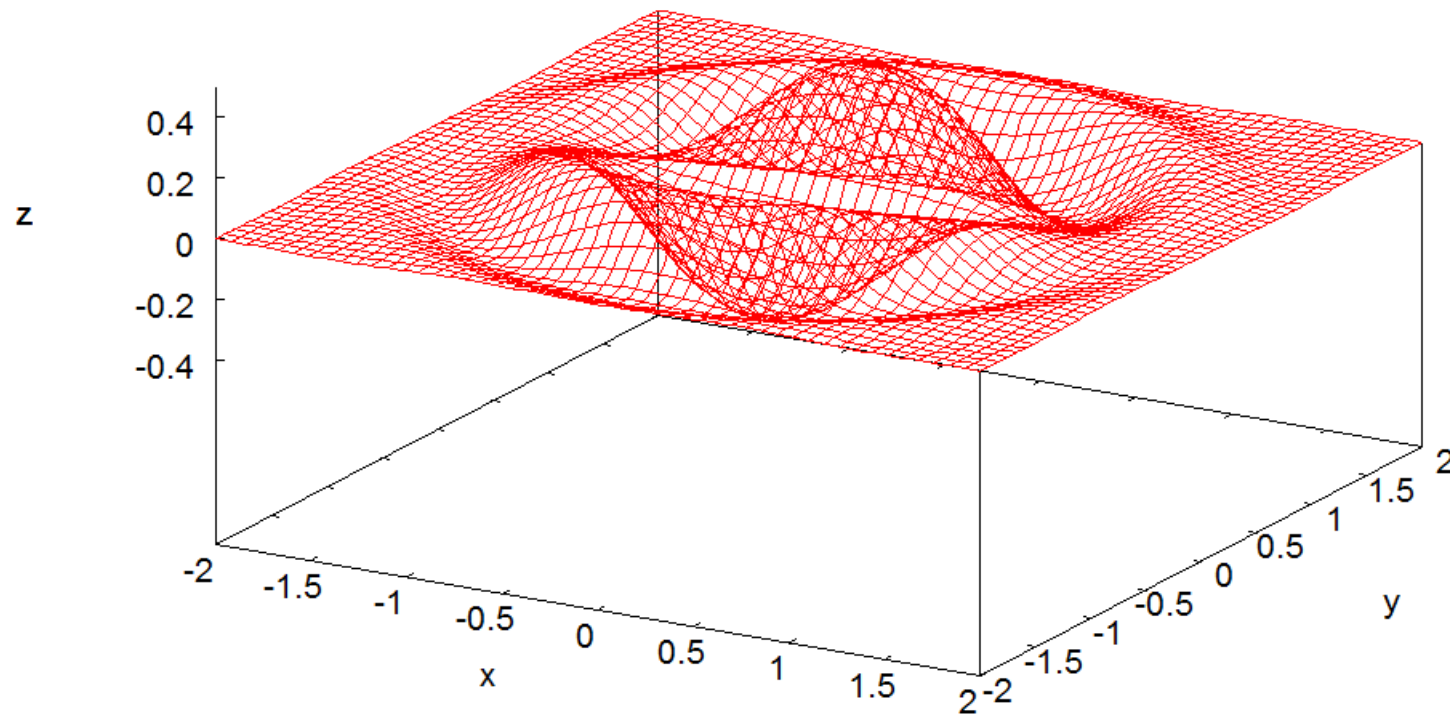
$\exp(-(x^2 + y^2)) \cdot \cos(x/4) \cdot \sin(y) \cdot \cos(2(x^2 + y^2))$  ———



# 3d-Plots - Funktionen

set isosamples 50 (Anzahl der Gitternetzlinien pro Richtung)

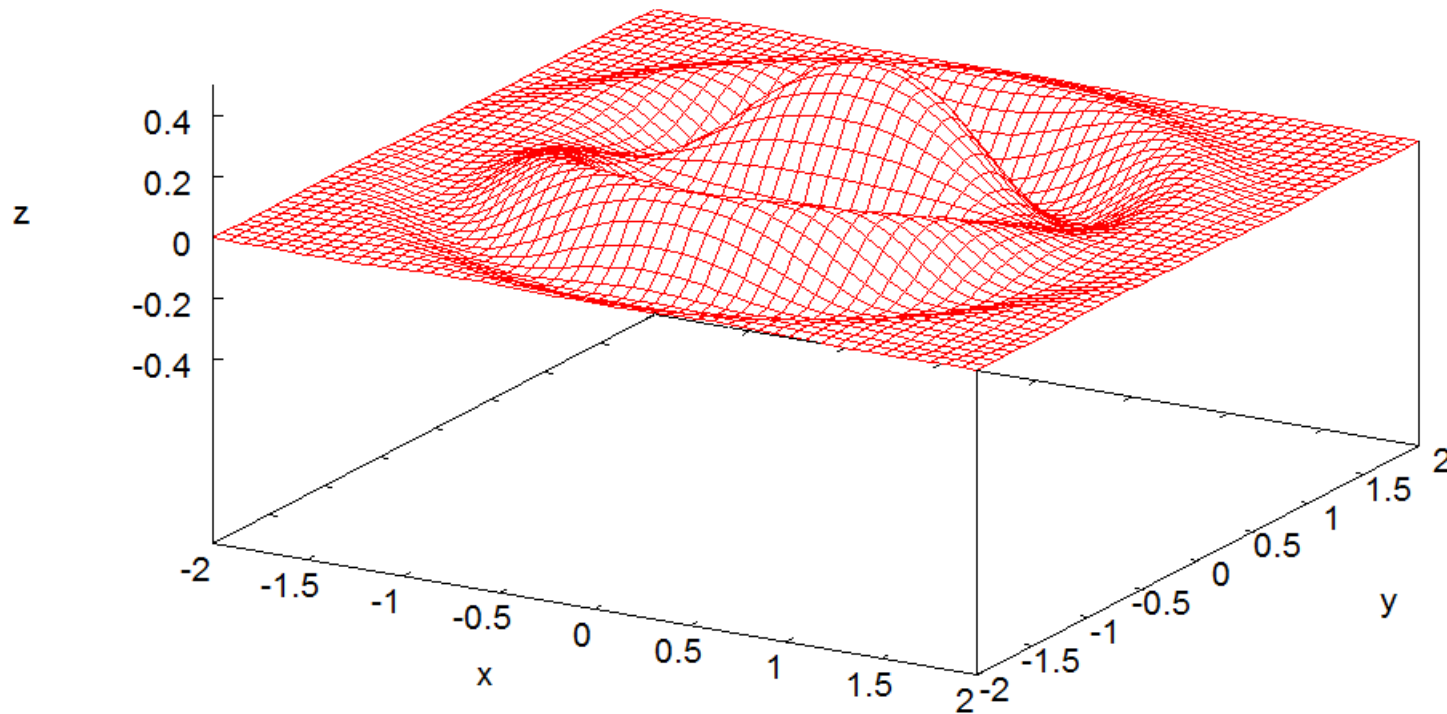
$$\exp(-(x^{**2} + y^{**2})) * \cos(x/4) * \sin(y) * \cos(2*(x^{**2} + y^{**2}))$$



# 3d-Plots - Funktionen

set hidden3d (Ausblenden der nicht-sichtbaren Gitternetzlinien)

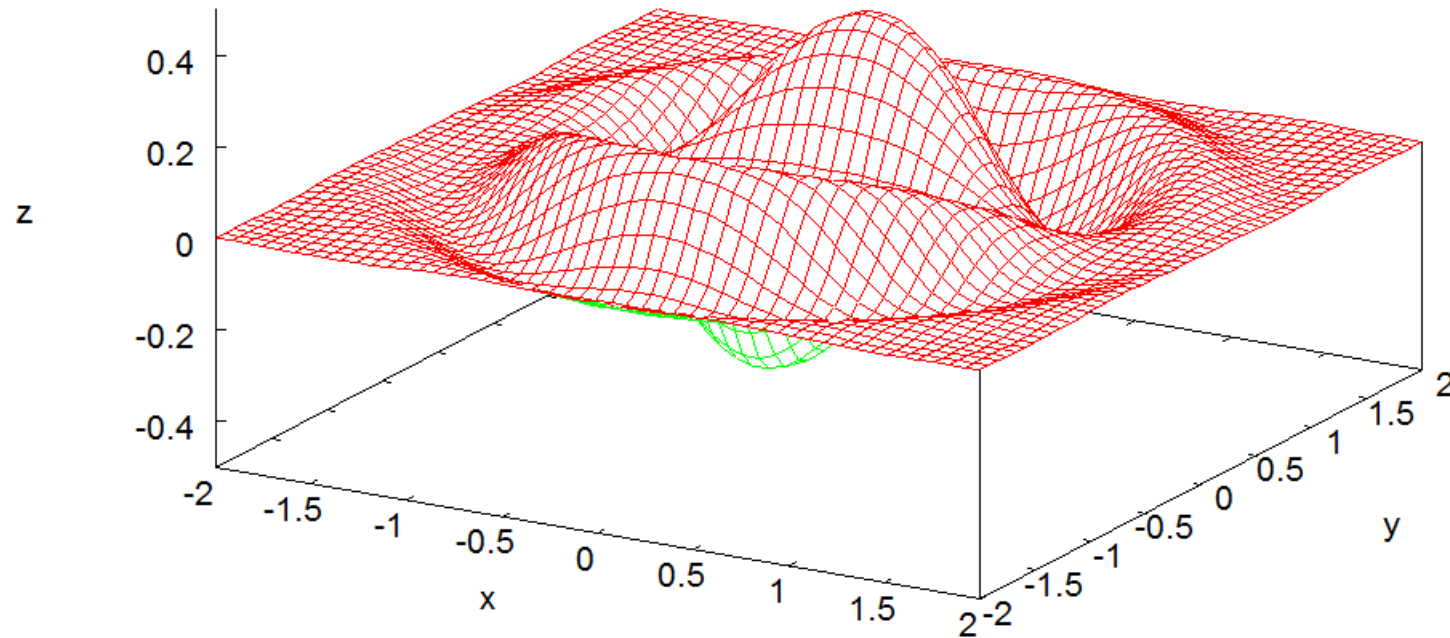
$$\exp(-(x^{**2} + y^{**2})) * \cos(x/4) * \sin(y) * \cos(2*(x^{**2} + y^{**2}))$$



# 3d-Plots - Funktionen

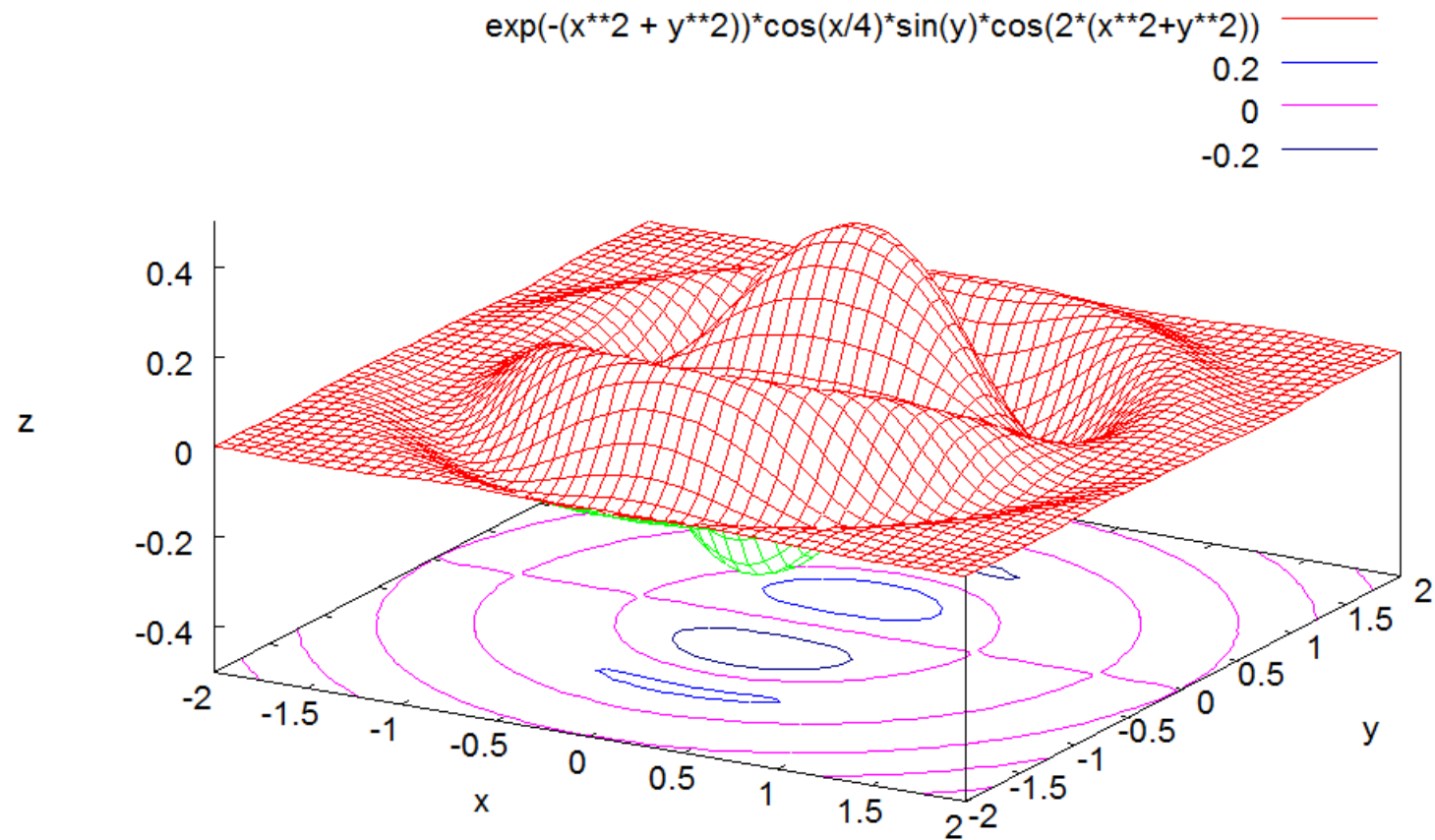
set xyplane at -0.5 (Position der xy-Ebene bei z=-0.5)

$$\exp(-(x^{**2} + y^{**2})) * \cos(x/4) * \sin(y) * \cos(2*(x^{**2} + y^{**2}))$$



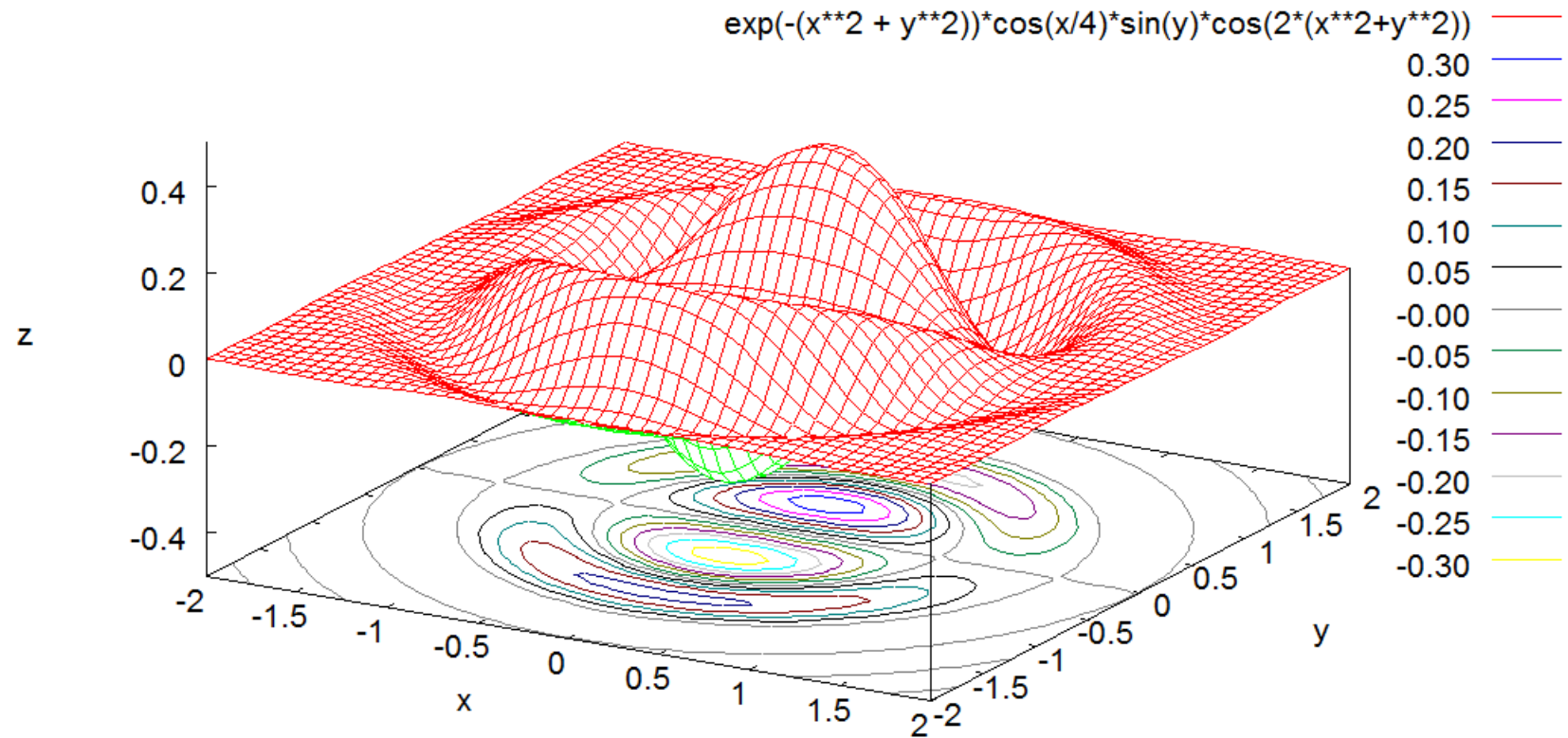
# 3d-Plots - Funktionen

set contour base (Anzeigen von Konturlinien auf der xy-Ebene)



# 3d-Plots - Funktionen

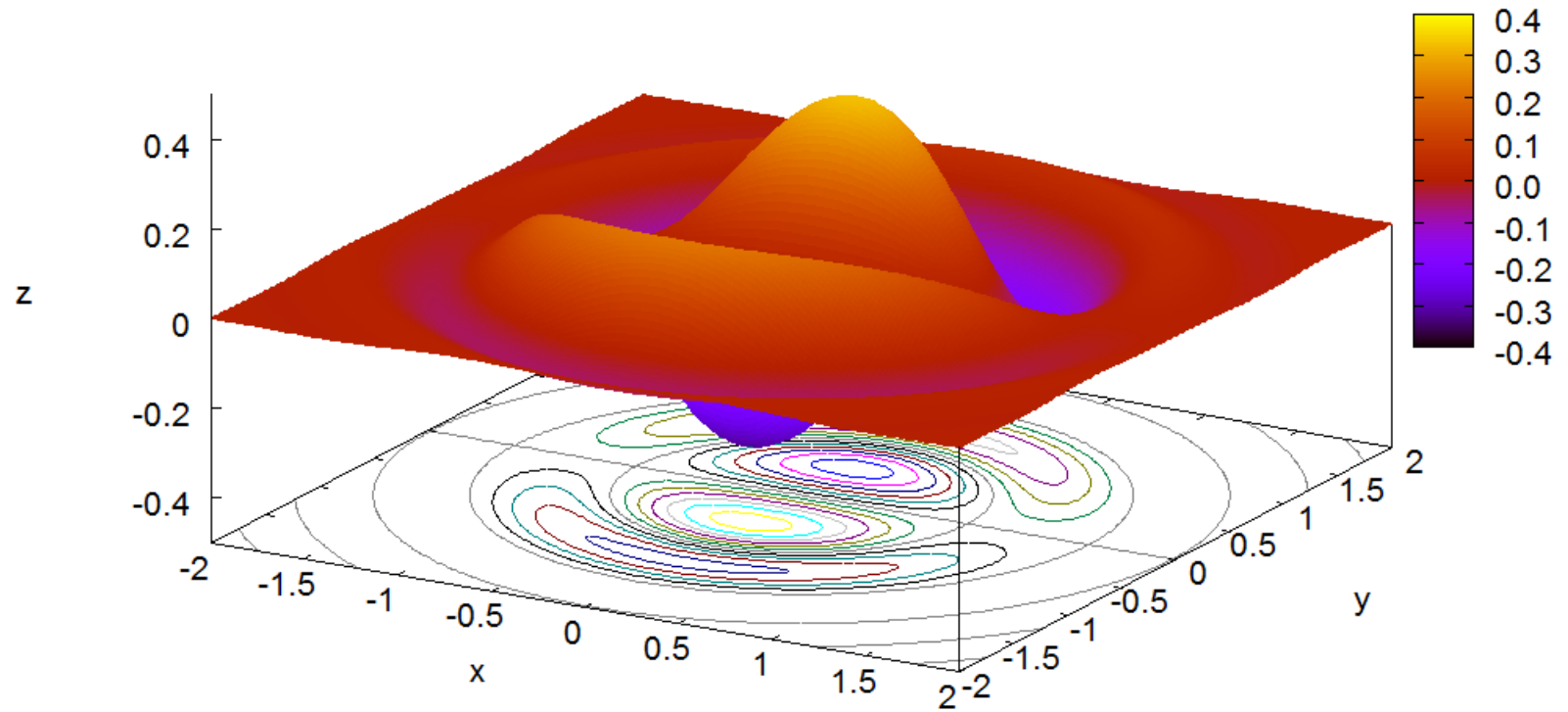
set cntrparam level 20 (Anzahl der Konturlinien)





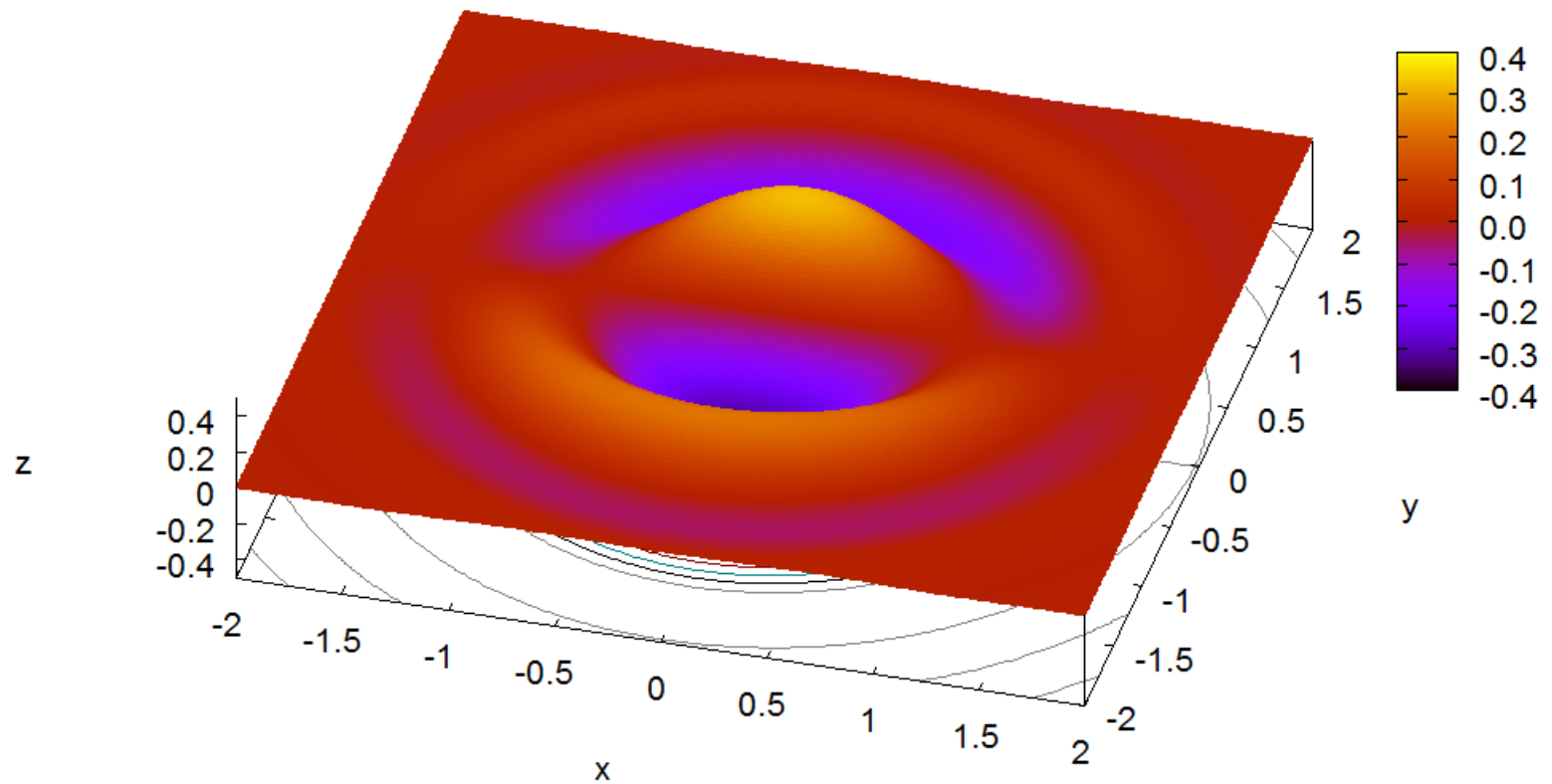
# 3d-Plots - Funktionen

set pm3d (Farbpalette anstatt Gitternetzlinien)



## 3d-Plots - Funktionen

set view 20,15 (Drehung der Ansicht um  $20^\circ$  um die x- und  $15^\circ$  um die y-Achse)

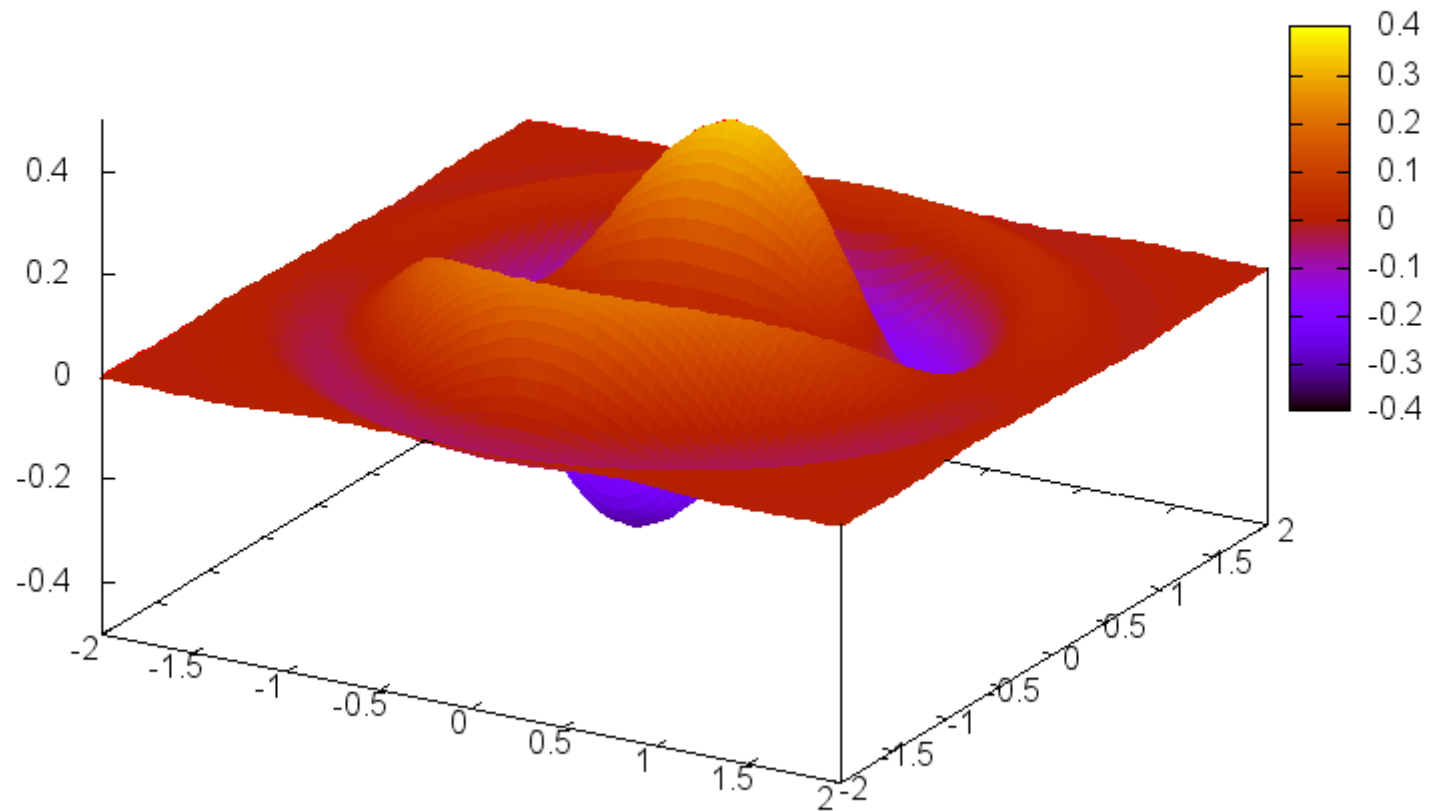


# Animationen

Hierbei wird jeder Plot Befehl als Einzelbild einer Sequenz gespeichert

```
set term gif animate delay 10 size 800,600  
set out "animation.gif"
```

```
splot ....  
splot ....  
....
```



# Animationen

Hierbei wird jeder Plot Befehl als Einzelbild einer Sequenz gespeichert

```
set term gif animate delay 10 size 800,600  
set out "animation.gif"
```

```
plot ....  
plot ....  
....
```

