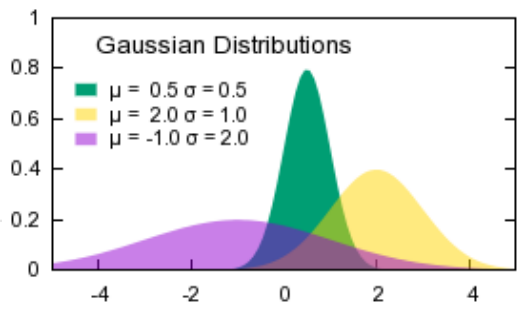
Gnuplot

Gnuplot ist ein kommandozeilen-orientiertes, interaktives wissenschaftliches Plotprogramm. Es ist klein, mächtig und macht all das, was man braucht, um Daten schnell darzustellen: Gnuplot kann sowohl Kurven (x/y-Datenpaare) als auch 3D-Objekte (Flächen) abbilden.

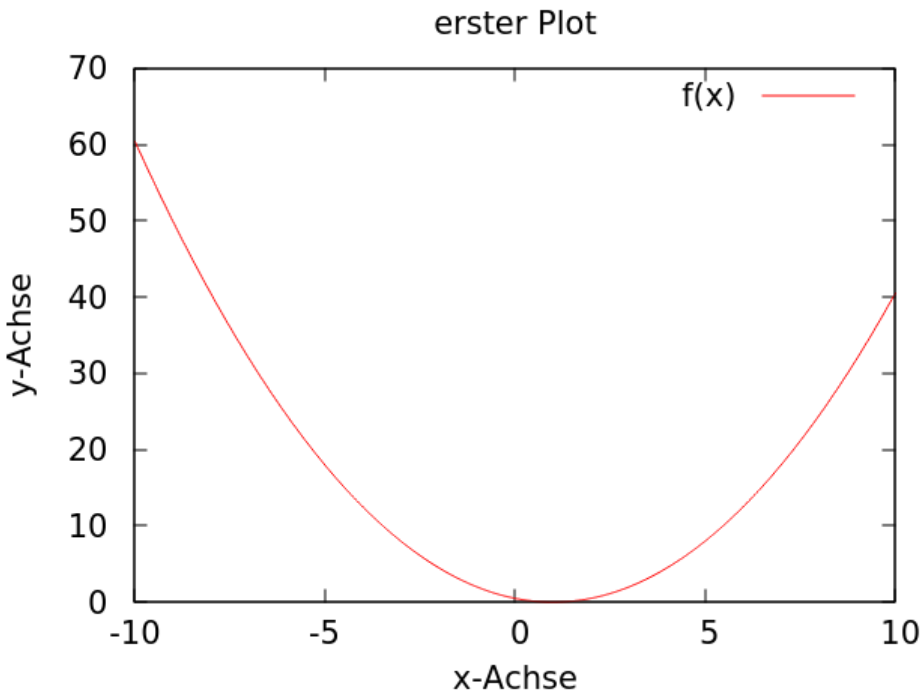


Im Internet findet man viele gute Einführungen. Aus diesem Grund sollen im folgenden nur die für uns wichtigsten Befehle kurz besprochen werden. Eine komplette Dokumentation sowie viele Demoprogramme findet man unter

`www.gnuplot.info`

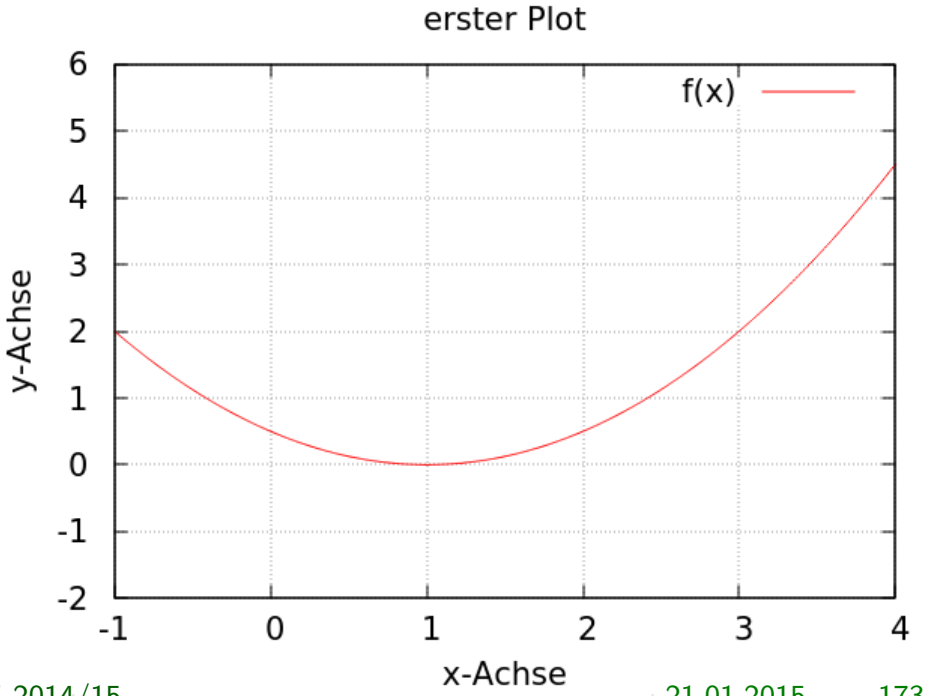
Nach dem Start des Programms in der Kommandozeile mit dem Befehl `gnuplot` ändert sich der Prompt: `gnuplot>`. Ab jetzt können direkt gnuplot-Befehle zum plotten eingegeben werden. Durch die Eingabe von `help` bzw. `help Befehl` erhält man eine ausführliche Hilfe. Mit dem Befehl `exit` oder `quit` wird Gnuplot beendet.
Die Befehle können auch untereinander in eine Datei (`bsp.gpl`) geschrieben werden. Durch den Aufruf `gnuplot bsp.gpl` oder innerhalb von Gnuplot mit `gnuplot> load 'bsp.gpl'` werden dann die Befehle nacheinander ausgeführt.

```
gnuplot> set title "erster Plot"
gnuplot> set xlabel "x-Achse"
gnuplot> set ylabel "y-Achse"
gnuplot> a = 1./2.
gnuplot> f(x) = a*(x-1)**2
gnuplot> plot f(x)
```

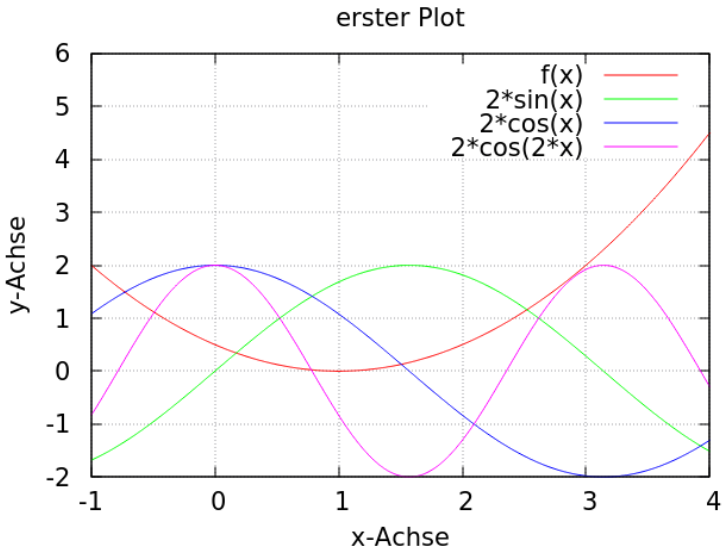


Möchte man einige Einstellungen ändern, so fährt man fort:

```
gnuplot> set xrange [-1:4]
gnuplot> set yrange [-2:6]
gnuplot> set grid
gnuplot> replot
```



```
gnuplot> replot 2*sin(x)
gnuplot> replot 2*con(x), 2*cos(2*x)
```

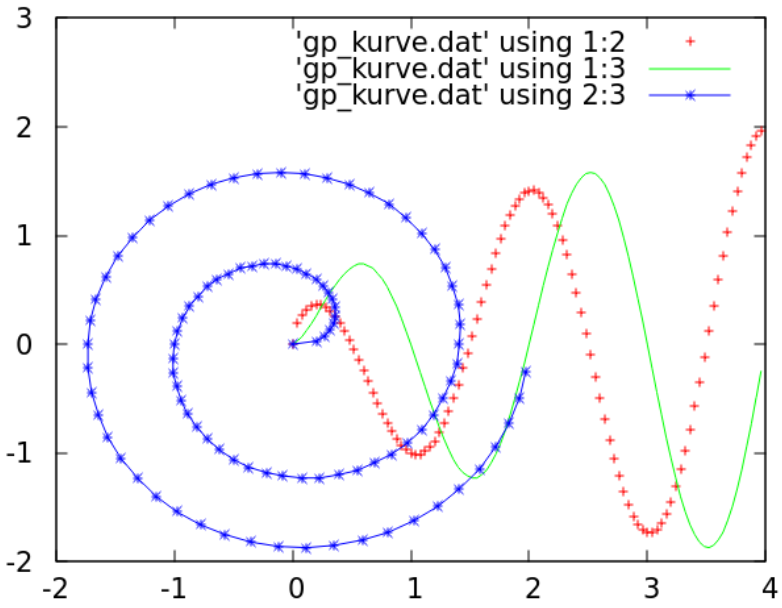


In den weiterführenden Vorlesungen wird Gnuplot meist dazu verwendet, um Daten zu visualisieren, die zuvor mit einem C-Programm berechnet wurden. Hierzu werden die Daten in einer Datei Spaltenweise abgespeichert und später mit Gnuplot eingelesen.
Die Datei kann auch Kommentare (beginnend mit #) beinhalten.

```
~/ModProg> head -10 gp_kurve.dat
#     Daten zum Visualisieren mit gnuplot
#       x                   y                f(x,y)
      0.00                0.00                0.00
      0.04                0.20                0.03
      0.08                0.27                0.07
      0.12                0.32                0.13
      0.16                0.35                0.19
      0.20                0.36                0.26
      0.24                0.36                0.34
      0.28                0.34                0.41
~/ModProg>
```

```
gnuplot> set yrange [-2:3]
gnuplot> plot 'gp_kurve.dat' using 1:2 with points
gnuplot> replot 'gp_kurve.dat' using 1:3 with lines
gnuplot> replot 'gp_kurve.dat' using 2:3 with linespoints
```

- mit `using` werden dabei die zu plottenden Spalten angegeben
- nach `with` folgt die Art der Visualisierung, also mit Punkten, Linien, Punkten und Linien, Balken, ...
- wird `with` verwendet, können nachfolgend noch die Eigenschaften angepasst werden: Liniendicke, Punktgröße, Farbe, gestrichelt, ...



Beispiel:

```
gnuplot> plot 'gp_kurve.dat' using 2:3 with linespoints linewidth 3 pointtype 7
```

Eine Auflistung und eine genaue Beschreibung der Optionen erhält man mit

```
gnuplot> help with
```

Hilfreich ist hierbei auch: `gnuplot> test`

Diese Anwendung von `using` beinhaltet noch weitere Möglichkeiten:

- Datenclipping: plotte Spalte 1 (x) gegen Spalte 3 (y), wenn y>5.

```
gnuplot> plot 'gp_kurve.dat' using 1:($3>5 ? $3 :1/0)
```

- Berechnen der Spalten: plotte Spalte 1 (x) gegen Spalte 3 (y), multipliziert mit -1.

```
gnuplot> plot 'gp_kurve.dat' using 1:($3*(-1))
```

Manchmal möchte man die Daten in einem logarithmischen Koordinatensystem darstellen. Hierzu kann man die Achsen logarithmisch skalieren mit:

```
gnuplot> set logscale x
gnuplot> set logscale y
```

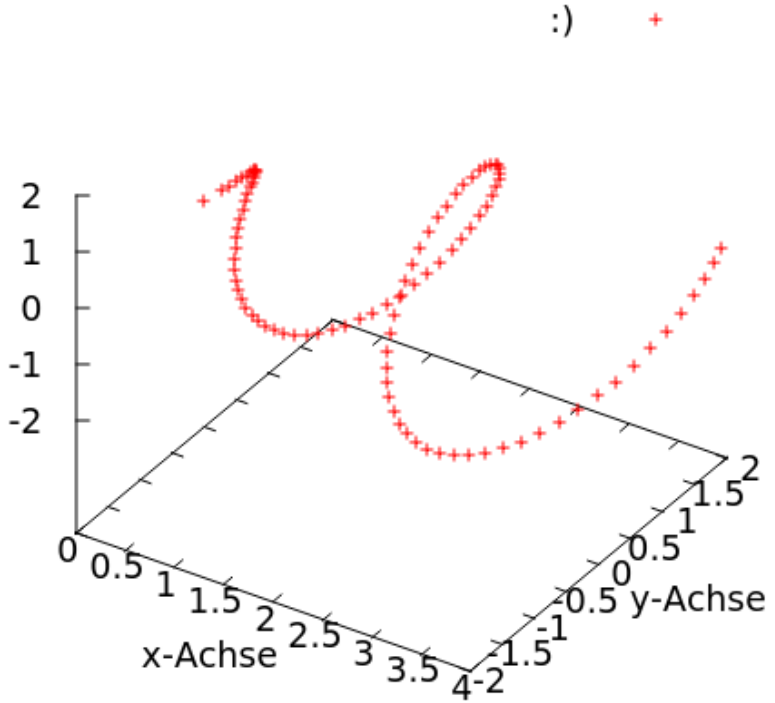
Ein Tipp zur Achsenbeschriftung: Mit

```
gnuplot> set xtics rotate 1
```

wird die Achsenbeschriftung der x-Achse um 90° gedreht, die 1 bewirkt, dass jeder `xtic` beschriftet wird (2 entsprechend jeder zweite . . .)

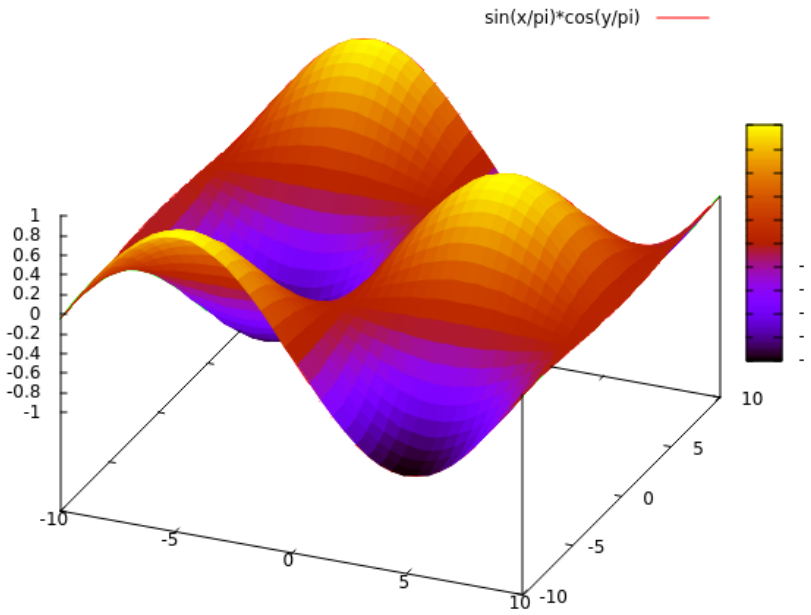
Es können auch 3D-Daten visualisiert werden:

```
gnuplot> set xlabel "x-Achse"
gnuplot> set ylabel "y-Achse"
gnuplot> set ztics 1
gnuplot> splot 'gp_kurve.dat' title ":)"
```



Und auch Funktionen:

```
gnuplot> set pm3d
gnuplot> set isosamples 40
gnuplot> set hidden3d
gnuplot> splot sin(x/pi)*cos(y/pi)
```



Um das Ergebnis als Bild abzuspeichern verwendet man z.B.:

```
gnuplot> set output "test.png"
gnuplot> set terminal png
gnuplot> replot
```