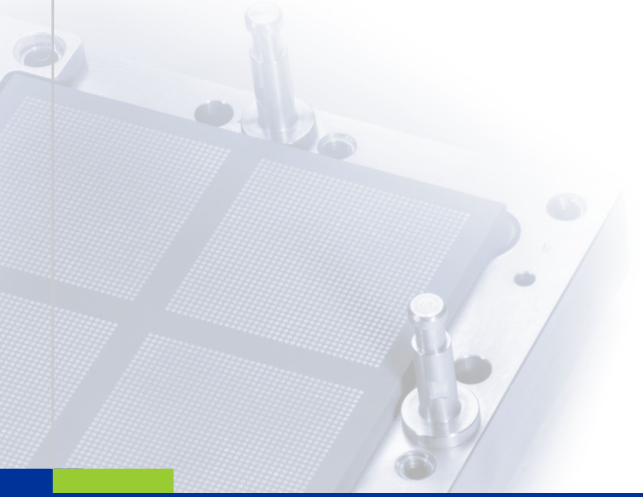
R|R|Z|N

Grafische Darstellung von Daten



gnuplot

Informationen

R|R|Z|N

- /9/ RRZN – Handbuch "gnuplot"

- Homepage; Download:
  - http://www.gnuplot.info/
- Tutorial:
  - http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html
  - http://t16web.lanl.gov/Kawano/gnuplot/index-e.html
  - http://www.rz.uni-osnabrueck.de/Zum_Nachlesen/Skripte_Tutorials/Gnuplot_Einfuehrung/pdf/gnuplot.pdf

gnuplot

- ... ist ein kommandozeilen-gesteuertes Plotprogramm
- ... ist ein Programm zur grafischen Darstellung von Funktionen und Daten.
- ... kann x / y-Paare sowohl als auch 3D-Objekte darstellen.
- ... nutzt Diagramme, um verschiedene mathematische Zusammenhänge darzustellen.
- ... ist ein freies Programm für UNIX, Linux, Windows etc.

Installation und Programmaufruf

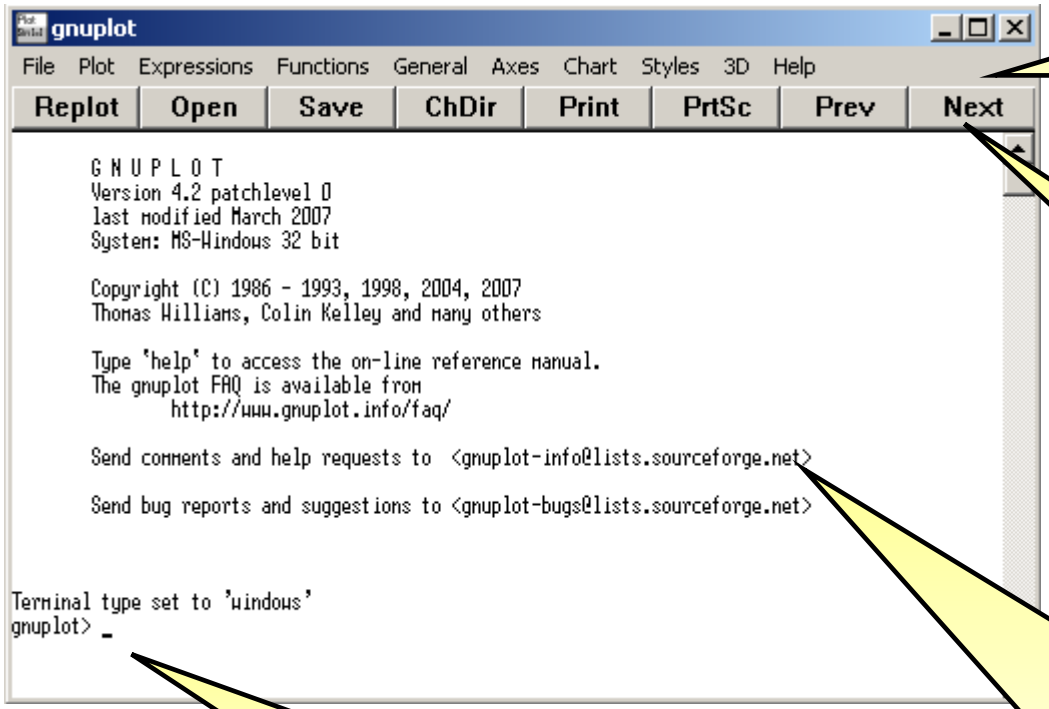
- Die Installation ist vom genutzten Betriebssystem abhängig.
- Häufig ist gnuplot als fertig kompiliertes Programm vorhanden.
- Teilweise muss das Programm entpackt werden.
- Die Installationsdatei bekommen Sie auf der Webseite http://www.gnuplot.info/.

Programmaufruf und -ende

- Der Programmaufruf hängt wiederum vom Betriebssystem ab.
  - Unter Linux wird das Programm mit Hilfe des Befehls *gnuplot* in der Konsole gestartet. Mit Hilfe des Befehls *man gnuplot* kann ein Manuel zu dem Programm aufgerufen werden.
  - Für das Betriebssystem Windows ist eine grafische Benutzeroberfläche vorhanden. Die grafische Benutzeroberfläche ist mit der MS-DOS-Eingabeaufforderung vergleichbar. Mit Hilfe des Befehls *wgnuplot.exe* wird das Programm gestartet.
- Um das Programm zu beenden, geben Sie `exit` oder `quit` ein.

Programm-Oberfläche unter Windows

R | R | Z | N

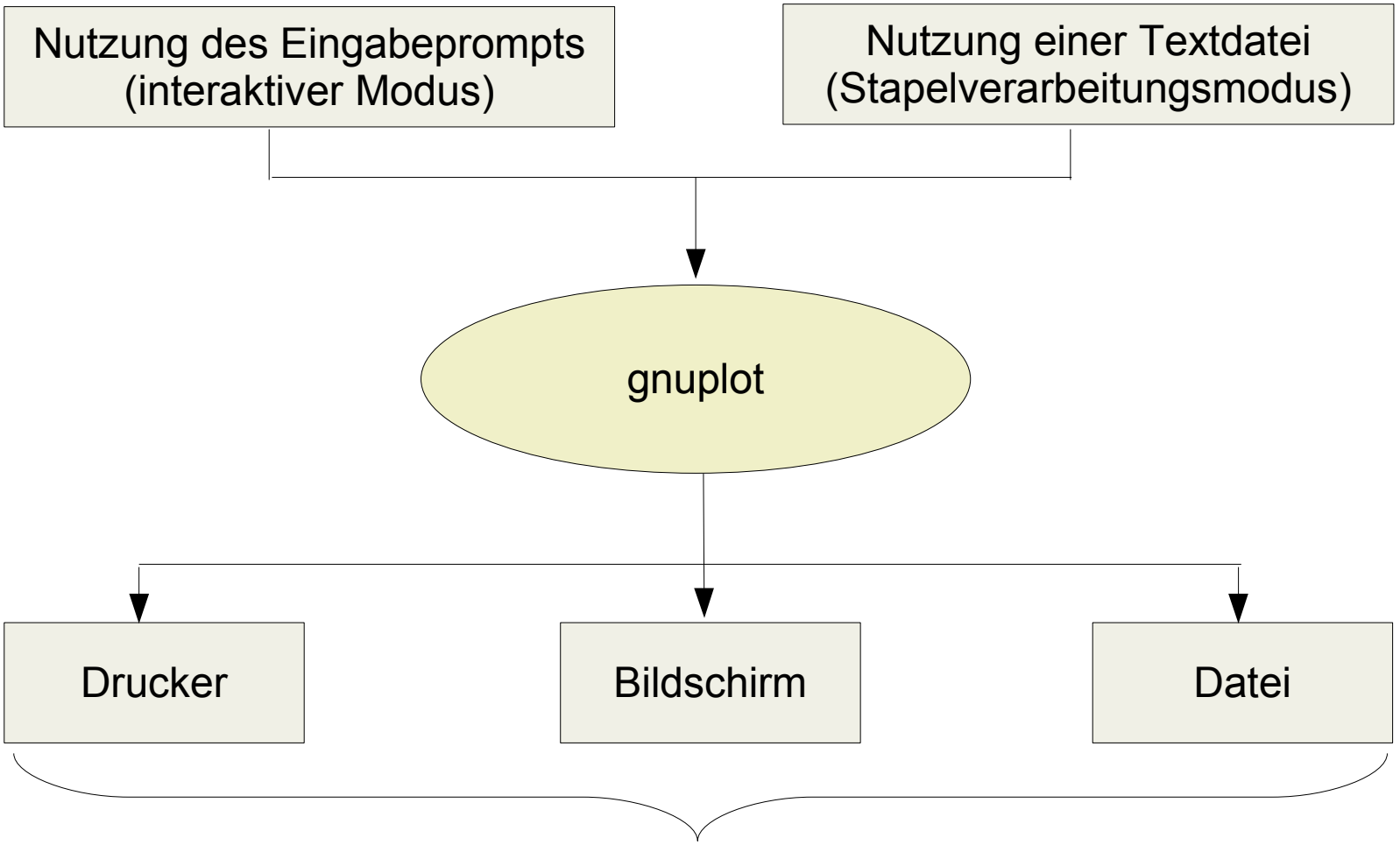


Eingebaute Funktionen etc.

Schaltflächen für die Bedienung von gnuplot.

Eingabeprompt.

Welche Version wird von gnuplot genutzt? Auf welchen Betriebssystem arbeitet gnuplot?


Nutzung des Eingabeprompts
(interaktiver Modus)
Nutzung einer Textdatei
(Stapelverarbeitungsmodus)
gnuplot
Drucker
Bildschirm
Datei
terminal
Terminal type set to 'window' = Ausgabe auf den Bildschirm

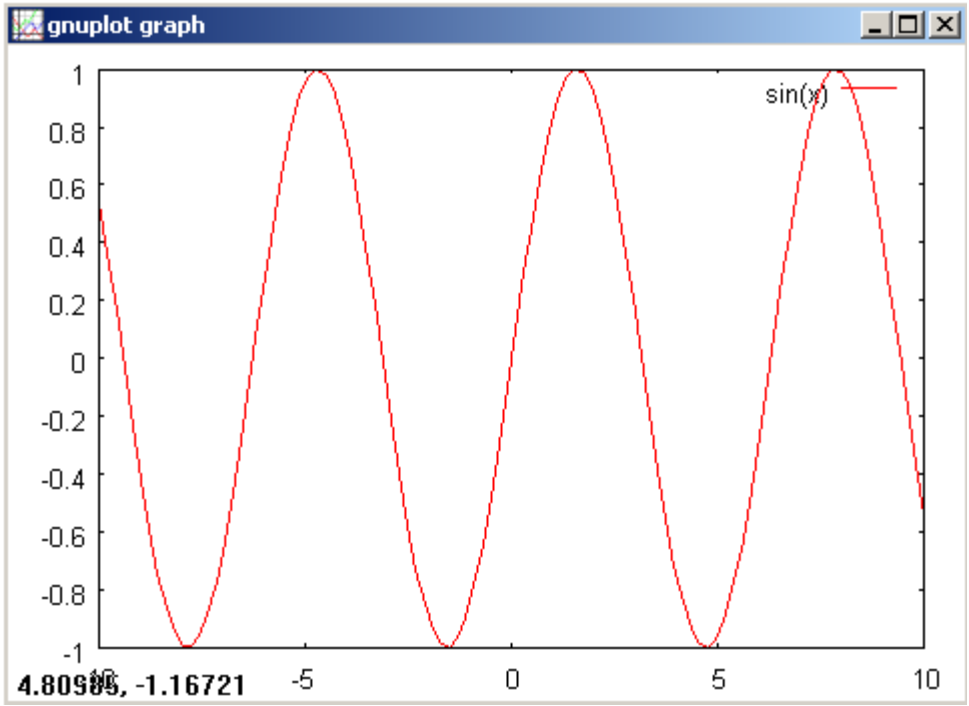
Funktion zeichnen

R|R|Z|N

- `plot` gibt eine 2-dimensionale Grafik aus.
- `splot` gibt eine 3-dimensionale Grafik aus.
- Übergabeparameter:
  - Dem Befehl wird eine vom Programm definierte Funktion übergeben.
  - Der Benutzer definiert eine Funktion und diese wird einen der beiden Befehle übergeben.
- Die Grafik wird in Abhängigkeit der Option `terminal` ausgegeben.
- Mit Hilfe von `replot` kann der letzte Plot wiederholt werden.

```
plot sin(x)
```

- Die Grafik wird mit einem Rahmen (border) umgeben.
- Die Koordinaten werden mit Hilfe der x- und y-Achse festgelegt.
- Die beiden Achsen schneiden sich im Ursprung.
- Jede Achse wird in bestimmten Schritten (minor tics) unterteilt.



Wertebereich für die Achsen setzen

R|R|Z|N

- Mit Hilfe von `xlabel` wird die x-Achse beschriftet.
- `xrange` setzt ein Intervall für die x-Achse. Das Intervall wird in Abhängigkeit des Funktionsverlaufs gewählt. Der Funktionsverlauf sollte vollständig dargestellt werden.

```
set xlabel "xAchse"
set xrange[0:5]
plot sin(x)
plot [0:5] sin(x)
```

- Mit Hilfe von `ylabel` kann die y-Achse beschriftet werden.
- `yrange` setzt ein Intervall für die y-Achse. gnuplot überprüft aber nicht, ob die Funktion in dem angegebenen Intervall liegt.

```
set ylabel "yAchse"
set yrange[0:5]
```

Skalenstriche


```
set xtics 5
set ytics 1, 2, 10
set xtics('niedrig' 0, 'mittel' 0.005, 'hoch' 0.01)
```

- Für die verschiedenen Skalenstriche kann die Einteilung festgelegt werden.
- In dem ersten Beispiel werden Skalenstriche im Abstand von fünf Längeneinheiten gesetzt.
- In dem zweiten Beispiel wird ein Startpunkt (1) und ein Endpunkt (10) angegeben. Die Skalenstriche werden im Abstand von zwei Längenheiten gesetzt.
- In dem dritten Beispiel wird an bestimmten Positionen ein Text ausgegeben.
- Desweitern kann man mit Hilfe von `set xtics font "schriftart, schriftgröße"` die Beschriftung der Achse formatiert werden.

Nebenskalenstriche

- Für die verschiedenen Nebenskalenstriche kann die Einteilung festgelegt werden.
- In dem ersten Beispiel werden Skalenstriche im Abstand von fünf Längeneinheiten gesetzt.
- In dem zweiten Beispiel wird ein Startpunkt (1) und ein Endpunkt (10) angegeben. Die Skalenstriche werden im Abstand von zwei Längenheiten gesetzt.

```
set mxtics 5
set mytics 3
plot sin(x)
```

Logarithmische Skalierung

R|R|Z|N

```
set logscale y
set format y '%.0e'
set grid
```

- `set logscale` schaltet eine logarithmische Skalierung für eine bestimmte Achse ein. Hier nutzt die y-Achse eine logarithmische Skalierung.
  - `unset logscale` schaltet die logarithmische Skalierung aus.
  - `set logscale xy 2` schaltet die lograrithmische Skalierung für die x- und y-Achse an. Es wird ein Logarithmus zur Basis 2 genutzt.
- `set format` beeinflusst die Beschriftung der Achsen. Die Formatierungen sind ähnlich wie in C. Die Möglichkeiten werden in der Hilfe von gnuplot beschrieben.
- `set grid` blendet ein Gitternetz für die Zeichnungsfläche ein.

Globale Optionen

R|R|Z|N

- `show Option` zeigt den aktuellen Wert der angegebenen Option an.
  - `show all` zeigt die eingestellten Optionen an.
- `set Option` setzt den Wert der angegebenen Option gesetzt.
  - `help set` zeigt eine Liste aller Einstellungsmöglichkeiten.
- `unset Option` setzt den Wert auf die Standardeinstellung zurück.
- `reset` setzt alle Optionen auf ihren Standardwert zurück. Die Optionen `terminal`, `output`, `loadpath` und `fontpath` werden nicht zurückgesetzt.

Variablen nutzen

R|R|Z|N

```
set xrange[-10:10]
a = 0.7
b = -2.0
c = -25
f(x)= a * x**3 + b * x**2 + c * x - 12
plot f(x)
```

- Den Variablen `a`, `b` und `c` werden mit Hilfe des Gleichheitszeichens ein Wert zugewiesen.
- Als Dezimaltrennzeichen wird der Punkt genutzt.

```
set xrange[-10:10]
a = 0.7
b = -2.0
c = -25
f(x)= a * x**3 + b * x**2 + c * x - 12
plot f(x)
```

- Mit Hilfe von `f(x)` wird eine Funktion definiert.
- Die Definition ist ähnlich wie in einem C-Programm.
- Mit Hilfe des Gleichheitszeichen wird der Funktion ein Ausdruck zugewiesen.
- Die Operatoren in dem Ausdruck sind ähnlich denen in der Programmiersprache C.
- Mit Hilfe von `printf f(Wert)` wird der berechnete Funktionswert ausgegeben.

Parametrischer Modus

R|R|Z|N

```
set parametric
f1(x) = cos(x)
f2(x) = sin(x)
plot [0.001 : 10*pi] f1(t), f2(t)
unset parametric
```

- Zwei eindimensionale Funktionen laufen über den gleichen Parameter.
- Darstellung von 2D-Kurven.
- Mit Hilfe von `set parametric` wird gnuplot in einen parametrischen Modus gesetzt. Der Nutzer muss diesen Modus manuell zurücksetzten.
- Die Funktionen werden über den Wert in den eckigen Klammern parametrisiert.

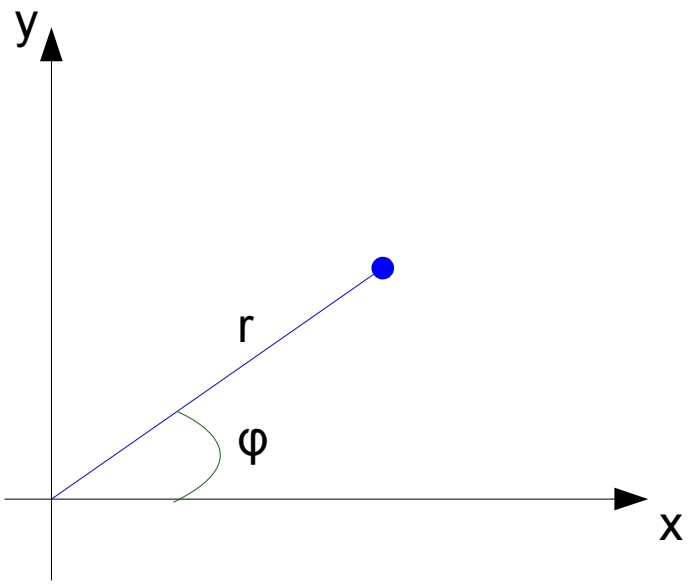
Parametrischer Modus

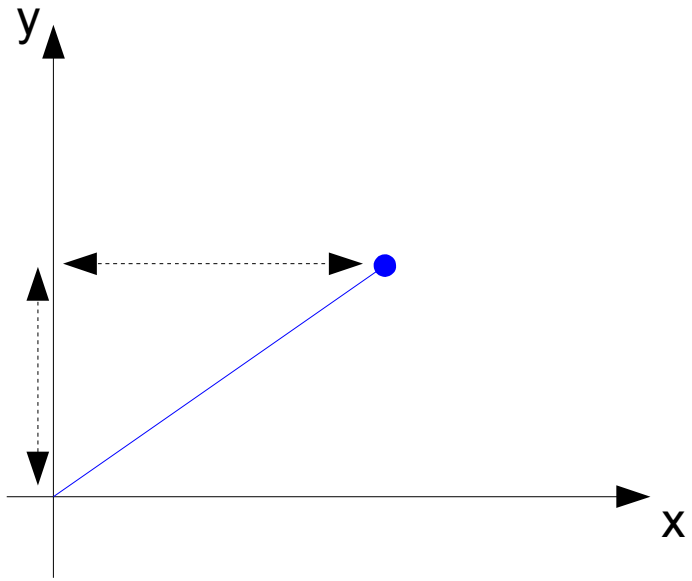
R|R|Z|N

```
set parametric
f1(x) = cos(x)
f2(x) = sin(x)
plot [0.001 : 10*pi] f1(t), f2(t)
unset parametric
```

- Zwei eindimensionale Funktionen laufen über den gleichen Parameter.
- Darstellung von 2-dimensionalen-Kurven.
- Mit Hilfe von `set parametric` wird gnuplot in einen parametrischen Modus gesetzt. Der Nutzer muss diesen Modus manuell zurücksetzten.
- Die Funktionen werden über den Wert in den eckigen Klammern parametrisiert.

Polarkoordinaten und kartesische Koordinaten





Polarkoordinaten nutzen

R|R|Z|N

```
set parametric
set polar
f1(x) = cos(x)
f2(x) = sin(x)
plot [0.001 : 10*pi] f1(t), f2(t)
unset parametric
unset polar
```

- 2-dimensionale Darstellung von Funktion mit Hilfe der Polarkoordinaten.

Eingaben speichern

R|R|Z|N

```
pwd
cd "C:\Eigene Dateien\myProgram"
save "datei.plt"
```

- Mit Hilfe des Befehls `pwd` wird das Arbeitsverzeichnis angezeigt.
- Dem Befehl `cd` wird ein Pfad zu einem Verzeichnis übergeben. Der Pfad des Arbeitsverzeichnisses wird verändert. Die Pfadangabe wird durch die Apostrophs begrenzt.
- `save` speichert die eingegebenen Zeilen in eine Datei. Der Dateiname wird durch Anführungszeichen begrenzt.
  - In der Textdatei stehen Angaben zur verwendeten Version, dem aktiven Ausgabemedium und dem Ziel der Ausgabe. Diesen Angaben folgt eine Liste von Variablen und Funktionen sowie der letzte plot-Befehl.
  - Der Name der Datei sowie die Dateiendung sind frei wählbar.

Befehle laden

```
pwd
cd "C:\Eigene Dateien\myProgram"
save "datei.plt"
load "datei.plt"
```

- `load` lädt die angegebene Datei in gnuplot.
- Die Befehle in der Datei werden automatisiert ausgeführt.
- Die Befehle werden im Stapelverarbeitungsmodus abgearbeitet. Das Ergebnis wird entsprechend der Angabe angezeigt oder gespeichert.

Modi in gnuplot

- Interaktiver Modus
  - Alle Befehle werden mit Hilfe des Prompts zeilenweise eingegeben.
  - Wenn das Programm beendet wird, gehen die Eingaben verloren.
- Stapelverarbeitungsmodus
  - Befehle werden zeilenweise in einer Textdatei abgelegt.
  - Die Textdatei kann mit Hilfe eines Editors oder des Befehls `save` im interaktiven Modus erstellt werden.
  - Die Textdatei wird in gnuplot geladen und die Befehle automatisch zeilenweise abgearbeitet. Das Ergebnis wird in Abhängigkeit von `terminal` ausgegeben.

- ... werden in Anführungszeichen oder Apostrophs gefasst.
- Falls ein Stringanfang und -ende durch Anführungszeichen gekennzeichnet sind, bleibt der Backslash als Steuerzeichen erhalten.
- Ein String in Apostrophs werden als Zeichenkette angesehen und im Speicher abgelegt.

Weitere Hinweise zu Stapeldateien

- Leerzeilen in Stapeldateien werden ignoriert.
- Ein Kommentarzeile beginnt mit einem Hash.
- gnuplot unterscheidet zwischen Groß- und Kleinschreibung.
- Eine vorhandene Datei wird ohne Nachfrage überschrieben.
- Mit Hilfe des Befehls `clear` kann das dargestellte Diagrammfenster geleert werden.

Datenpunkte ausgegeben

R|R|Z|N

- Die Datenpunkte der Linie werden in der Datei `output.dat` gespeichert.
- Die Datenpunkte werden in Tabellenform abgelegt.
- Der Dateiname ist frei wählbar.
- Als Dateiendung sollte ".dat" oder ".txt" gewählt werden.

```
set table "output.dat"
 plot f(x)
unset table
```

Datendatei mit Hilfe von C erstellen

R|R|Z|N

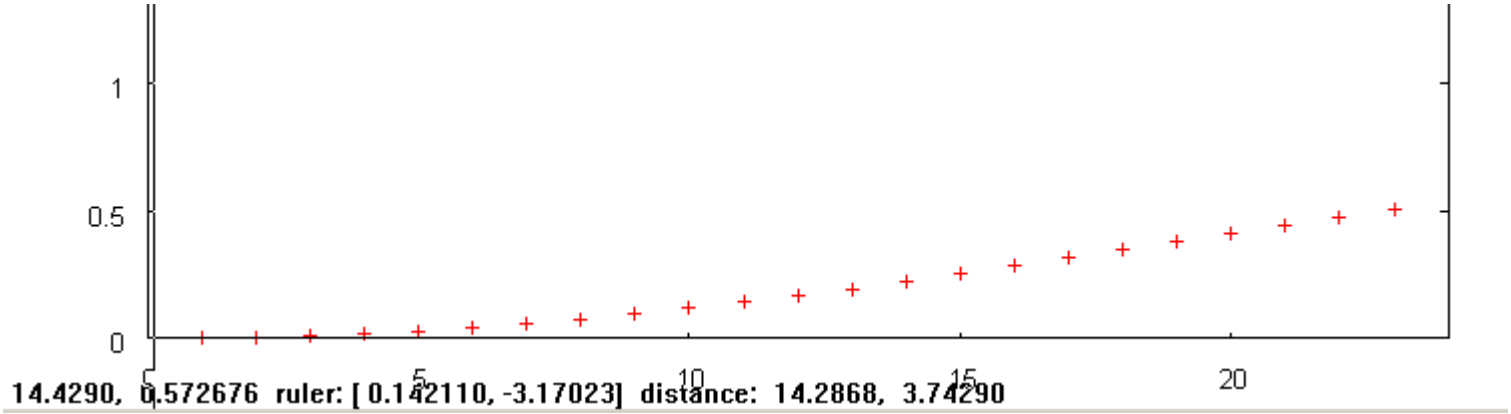
```
int main(){
    int person = 1;
    double q = 1;
    FILE *ptrDatei;
    ptrDatei = fopen("zufall.dat", "w");

   while (q >= 0.5){
         q *= (365.0 - person + 1.0) / 365.0;
         person++;
         fprintf(ptrDatei, "%d %f\n", person - 1, 1 - q);
   }

  fclose(ptrDatei);
  ptrDatei = NULL;
  return EXIT_SUCCESS;
}
```

Mit Hilfe von fprintf() werden die gewünschten Daten in eine Datei geschrieben. Diese Datei wird in gnuplot als Datengrundlage genutzt.

Datenpunkte anzeigen

- Die Datei `zufall.dat` enthält die darzustellenden Datenpunkte.
- Die Datenpunkte werden standardmäßig als Punkte dargestellt.

```
plot "zufall.dat"
```



Darstellung der Datenpunkte


```
plot "zufall.dat" with lines
```

- Die Datenpunkte werden mit einer Linie verbunden.
- Weitere Möglichkeiten:
  - `points`.
  - `linespoints` ist eine Kombination aus `points` und `lines`.
  - `impules` nutzt senkrechte Linien für die Darstellung der Datenpunkte.
  - `steps` erzeugt einen stufenförmigen Verlauf.

Spalten einer Tabelle

R|R|Z|N

```
plot "bahn.dat" using 2:3
```

- Mit Hilfe des Befehls `using` können die Spalten einer Tabelle ausgewählt und den Achsen zugeordnet werden.
- Zur Kennzeichnung der Spalten kann eine Zahl oder ein mathematischer Ausdruck genutzt werden.
- Die Spalten werden von 1 bis n nummeriert. Die Zeilen werden von 0 bis n nummeriert.
- Beispiele:
  - using 0:1 — Die Tabelle enthält eine Spalte.
  - using 2:3 — Die zweite Spalte wird auf der x-Achse aufgetragen. Die dritte Spalte wird auf der y-Achse aufgetragen.
  - using : : 4 — Die erste Spalte wird auf der x-Achse aufgetragen. Die zweite Spalte (der Wert des linken Nachbarn wird erhöht) wird auf der y-Achse aufgetragen. Die vierte Spalte wird auf der z-Achse aufgetragen.

Nährung von Funktionen

R|R|Z|N

```
f(x)= a * x**3 + b * x**2 + c * x - 12
fit f(x) "output.dat" via a, b, c
```

- Zuerst wird eine ein- oder zweidimensionale Funktion definiert.
- Diese Funktion soll an die, in der Datei vorgegebenen Datenpunkte angepasst werden. Es wird eine Kurve durch die eingelesenen Datenpunkte gelegt.
- `via` listet alle Parameter auf, die angenährt werden sollen. In diesem Beispiel sind es alle Variablen der Funktion.
- Die Nährung basiert auf der Methode der kleinsten Quadrate. Es wird die Kurve gesucht, bei der die Summe der Abweichungen zum Quadrat zwischen Kurve und Datenpunkten minimal ist.
- Der Befehl gibt die berechneten Werte sowie deren Fehler heraus.

Mehrere Linien zeichnen

```
plot "klima.dat" using 1:2 with lines, "klima.dat" \
     using 1:3 with lines
```

- In diesem Fall werden zwei Linien auf dem Bildschirm gezeichnet. Die erste Linie nutzt die Koordinaten der ersten und zweiten Spalte. Die zweite Linie nutzt die Daten der ersten und dritten Spalte.
- Hier werden mehrere Datensätze in einem Diagramm gezeichnet.
- Die einzelnen Datenblöcke werden durch ein Kommata getrennt und aneinander gereiht.

Fehlerbalken zeichnen

R|R|Z|N

```
plot "klimaError.dat" using 1:2:(sqrt($1)) with errorlines
```

- Hier werden die ersten zwei Spalten einer Datendatei eingelesen.
- Eine dritte Spalte wird für die Fehlerabweichungen aus der ersten Spalte berechnet (`sqrt($1)`).
- Die Datenpunkte werden mit einer Linie verbunden.
- In jedem Datenpunkt wird eine Fehlerlinie, die durch die Funktion berechnet wurde, gezeichnet.

```
plot "xxx.dat" using 1:2:4 with yerrorbars
```

- Ein Fehlerbalken wird für die y-Achse gezeichnet.
- Die Fehlergrenze wird aus der vierten Spalte der Datendatei berechnet.
- Statt `yerrorbars` kann auch `xerrorbars` genutzt werden.

```
plot "xxx.dat" using 1:2:($1 +$3):(sqrt($4)): 5:6 _
                                with xyerrorbars
```

- In der ersten und zweiten Spalte befinden sich die Messreihen.
- Für die x-Werte wird die minimale Fehlergrenze durch `($1 +$3)` sowie die maximale Fehlergrenze durch `(sqrt($4))` berechnet.
- Für die y-Werte wird die minimale Fehlergrenze durch die fünfte Spalte sowie die maximale Fehlergrenze durch die sechste Spalte berechnet.

Ausgabe in eine Datei

- Mit Hilfe der Option `terminal` wird das Ausgabeformat festgelegt. In diesem Beispiel wird das Ergebnis als Postscript-Datei ausgegeben.
- Die Ausgabe wird mit Hilfe der Option `output` in einer Datei gespeichert.
- Alle implementierten Ausgabeformate finden Sie in der Hilfe von gnuplot.

```
set terminal postscript
set output 'klima.ps'
```

```
set view 60, 60, 1, 1

fkt1(x, y)  = x * y
fkt2(x, y) = (x**2 + y**2) / 10
splot fkt1(x, y), fkt2(x, y)
```

- Mit Hilfe von `splot` werden die beiden Funktionen als dreidimensionale Grafik ausgegeben.
- Ein Datenpunkt besteht aus drei Koordinaten x, y, und z.
  - Falls eine einspaltige Tabelle genutzt wird, werden die Daten für die z-Achse genutzt.
  - Falls eine zweispaltige Tabelle genutzt wird, wird die erste Spalte für die z-Achse und der zweite Wert für die Farbgebung genutzt.
  - Mit Hilfe von `using` können die zu nutzenden Spalten angegeben werden.

```
set view 60, 60, 1, 1

fkt1(x, y)  = x * y
fkt2(x, y) = (x**2 + y**2) / 10
splot fkt1(x, y), fkt2(x, y)
```

- Mit Hilfe von `set view` kann der Blickwinkel angegeben werden.
- Folgende Parameter können an den Befehl übergeben werden:
  - α_x legt die Höhe, aus der das Diagramm betrachtet wird, fest. Rotation um die x-Achse in Grad.
  - β_z legt die Drehung der x, y-Ebene fest. Es wird in Drehwinkel in Grad angegeben. Rotation um die z-Achse in Grad.
  - Falls die Vergrößerung einen Wert größer eins besitzt, rückt die Position des Betrachters näher an das Diagramm.
  - Falls der Streckfaktor einen wert größer eins besitzt, wird die z-Achse überhöht.
- Mit Hilfe von `unset view` werden die Standardeinstellungen (60, 30, 1, 1) gesetzt.

Blickwinkel

```
set view 60, 60, 1, 1

set hidden3d
set contour both
```

- `contour` zeichnet Höhenlinien. Alle Punkte auf der Oberfläche, die einen bestimmten Wert auf der z-Achse besitzen, werden miteinander verbunden.
  - `base` erzeugt Konturen nur auf der x-y- Ebene.
  - `surface` zeichnet Höhenlinien auf die Oberfläche.
  - `both` ist eine Kombination aus beiden Eigenschaften.
- `hidden3D` schaltet die Transparenz der Oberfläche ab.
  - Es werden alle Linienstücke, die aus Sicht des Betrachters nicht sichtbar sind, ausgeblendet.

Parametrischer Modus

R|R|Z|N

```
set parametric
x(p, a) = 1 * sin(p) * cos(a)
y(p, a) = 1 * sin(p) * cos(a)
z(p)= 1 * cos(p)
splot [p=0:pi] [a=0:2*pi] x(p, a), y(p, a), z(p)
unset parametric
```

- Hier wird eine Fläche im Raum dargestellt, die über zwei Paramter läuft.
- Mit Hilfe von `set parametric` wird gnuplot in einen parametrischen Modus gesetzt. Der Nutzer muss diesen Modus manuell zurücksetzten.
- Die Funktionen werden über den Wert in den eckigen Klammern parametrisiert.
- Mit Hilfe des Gleichheitszeichen kann einem Parameter ein bestimmter Wertraum zugewiesen werden.