Gnuplot

Justin Simon Bayer

12. Januar 2006

Inhaltsverzeichnis

1 Gnuplot?

1.1 Was ist das eigentlich?

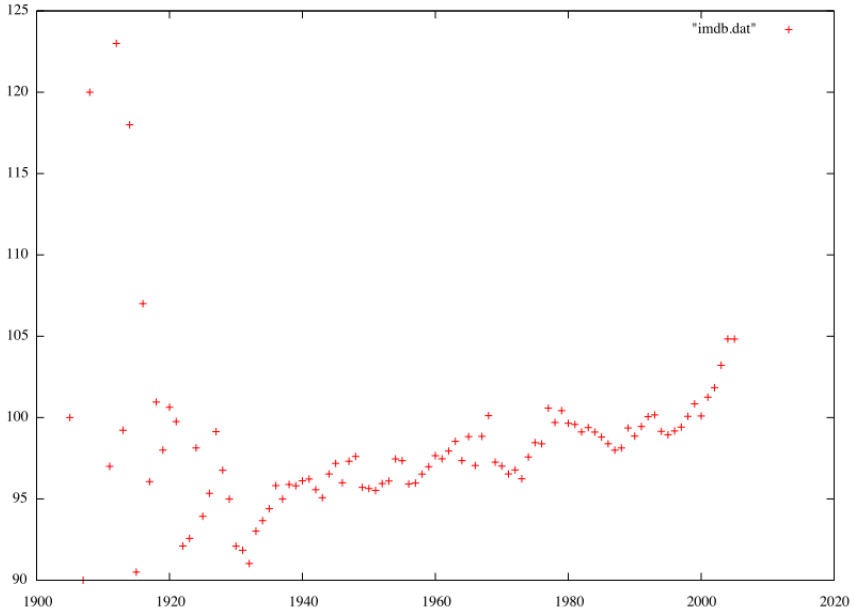
gnuplot.info Gnuplot is a portable command-line driven interactive data and function plotting utility.

- Lauffähig unter BSD, Linux, Mac OS, Windows, DOS, Atari, ...
- Eingaben werden in der Konsole entgegengenommen und in einer grafischen Oberfläche dargestellt
- Visualisierung von Daten und Funktionen

1.2 Ein Beispiel

Problem Hollywood interessiert sich für eine Visualisierung der durchschnittlichen Spielfilmlänge von Filmen pro Jahr.

Idee Gnuplot kann Dateien einlesen und entsprechende Graphen plotten. Über die IMDB-API bekommen wir Zugang zu den benötigten Daten.

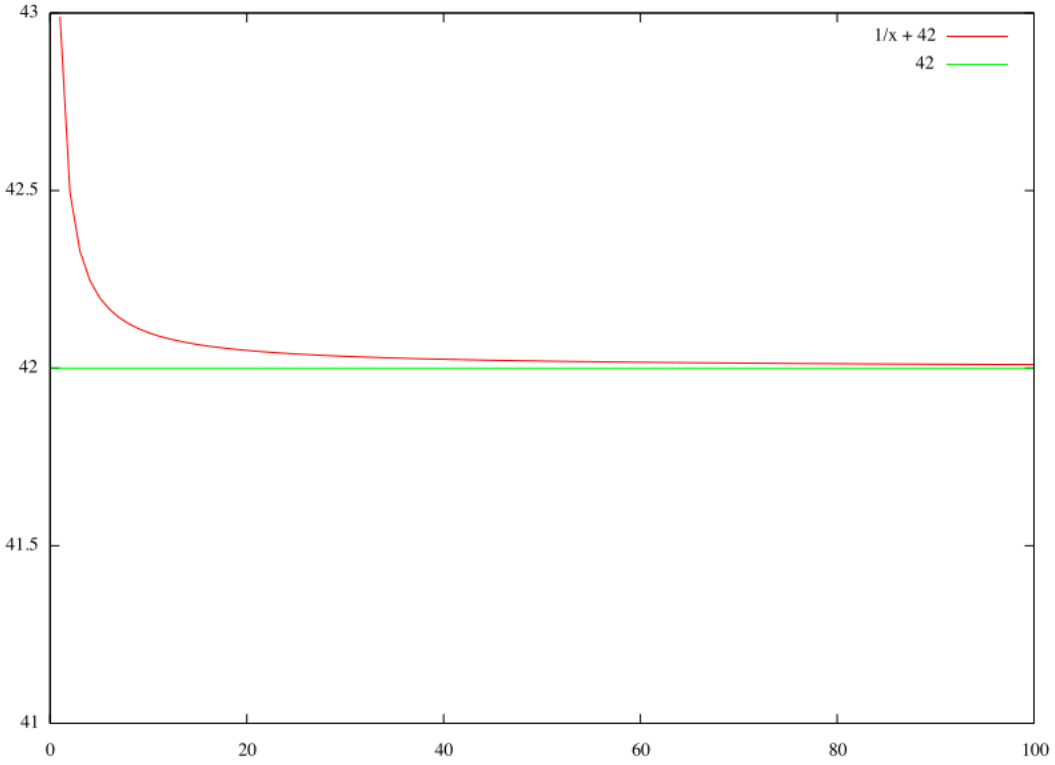


1.3 Und noch eins

Problem Grenzwert einer komplizierten Folge $a_n := \frac{1}{n} + 42$ ist für eine Diskrete Strukturen Hausaufgabe zu berechnen. Wir sind uns nicht völlig sicher ob unsere Lösung korrekt ist.

Gnuplot Konsole Dies erreicht man durch folende Eingabe:

```
gnuplot> plot[0:100][41:43] 1/x + 42
gnuplot> replot 42
```



2 Eintauchen...

2.1 Grundsätzliches

- Gnuplot kann auf verschiedene Weisen angesteuert werden: aus dem eigenen Interpreter, über Gnuplot-Skripte, Bibliotheken von Programmiersprachen, ...
- Drei Kommandos zum Plotten: `plot splot replot`
- Zahlreiche andere Kommandos zum Anpassen der Plots

2.2 Funktionen spezifizieren

Ein Plotbefehl hat als notwendiges Argument eine Funktion, z.B. `2*x/(x+1)`. Die Schreibweise ähnelt denen vieler Programmiersprachen.

- Bei zweidimensionalen Plots wird als Variable üblicherweise `x` verwendet
- Bei dreidimensionalen Plots kommt noch `y` hinzu
- Zur Verfügung stehen dabei die üblichen Operatoren
- Ausserdem zahlreiche *Builtins* die komplexere mathematische Funktionen ermöglichen

Ausserdem... Eine Funktion kann in Gnuplot auch fest definiert und später benutzt werden:

```
gnuplot> f(x) = x**2
gnuplot> print f(2)
4
```

2.2.1 Operatoren und Builtins

Unäre Operatoren `!`

Binäre Operatoren `** * / % + - == != < <= > >= ^ & | ||`

Ternäre Operator `?:`

Bultins `abs acos acosh arg asin asinh atan atan2 atanh besj0 besj1 besy0 besy1 ceil column cos ...`

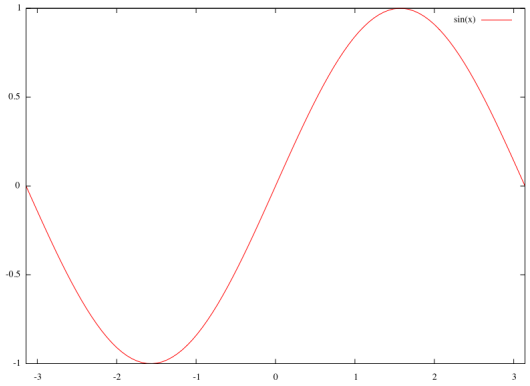
2.3 Die drei wichtigsten Befehle

2.3.1 plot

Der Befehl Plot ist zum Darstellen zweidimensionaler Daten.

Syntax Direkt nach dem Befehl kann man in eckigen Klammern die Intervalle angeben die dargestellt werden sollen. Dabei ist zuerst der Eingabe-Intervall anzugeben (also x), danach der Ausgabe-Intervall (also y). Direkt danach wird die zu plottende Funktion erwartet.

- `plot x**2`
- `plot[-pi:pi][-1:1] sin(x)`
- `plot[][0:3] sin(x)`

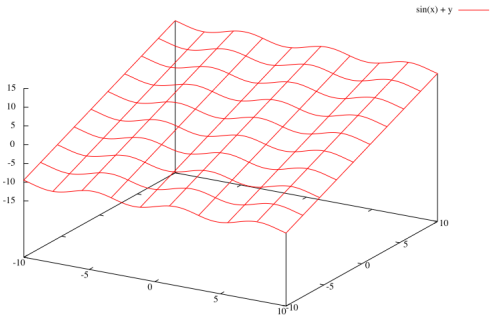


2.3.2 splot

Dieser Befehl ist zum Plotten dreidimensionaler Graphen.

Syntax Analog zu `plot`, mit dem Unterschied, dass es mit `y` eine weitere Laufvariable gibt. Deswegen lässt sich auch in eckigen Klammern ein dritter Intervall angeben.

- `splot sin(x) + y`
- `splot[-pi:pi] sin(x) * y`
- `splot[-pi:pi][-1:1][-1:1] sin(x) * y * cos(x)`



2.3.3 replot

Replot dient dazu, den letzten Plot nocheinmal auszuführen. Dies erweist sich in unterschiedlichsten Fällen als nützlich:

- Neusetzen von Umgebungsvariablen
- Überschreiben einer verwendeten Funktion
- In den aktuellen Plot wird ein weiterer Graph gezeichnet

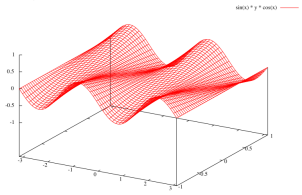
3 Erweiterte Funktionalität

3.1 Umgebungsvariablen

Weitere Funktionalität wird über Umgebungsvariablen ermöglicht. Dabei kann es sich um allerlei Datentypen handeln.
`set <variable> <wert>`
Dabei ist zu beachten, dass Wahrheitswerte einfach nur gesetzt werden müssen, danach nehmen sie den Wert wahr an. Um Urpsrungswerte wieder zu erlangen gibt es unset:
`unset <variable>`

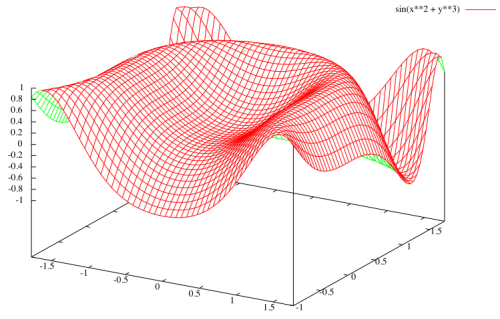
3.1.1 Genauigkeit

Die Umgebungsvariable `isosamples` gibt die Maschenweite oder Genauigkeit von 3D-Graphen an. Hohe Werte machen sich deutlich in Berechnungszeiten spürbar. Für 2-Graphen steht die Variable `samples` zur Verfügung. `set isosamples x [,y]`



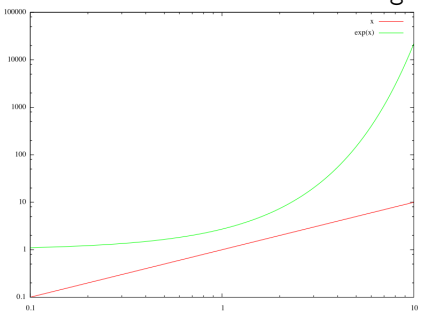
3.1.2 Verbergen von hinten Liegendem

In manchen Fällen sind durcheinander laufende Linien etwas verwirrend. Also kann man die Umgebungsvariable `hidden3d` setzen - dann wird immer nur die erste Fläche dargestellt und die Uner- und Oberseite in verschiedenen Farben.



3.1.3 Logarithmische Skala

Bei sehr stark wachsenden Funktionen fällt es z.T. sehr schwer Graphen zu interpretieren. In diesem Fall schafft üblicherweise eine logarithmische Skala abhilfe, die man erhält indem man die Variable `logscale` setzt.



3.1.4 Diverses

Es gibt noch viele weitere Umgebungsvariablen. Hier ein paar Beispiele

- `title` Titel des Plots
- `grid` Gitternetz im Koordinatensystem
- `xrange yrange zrange` Intervalle in denen zu Plotten ist
- `polar` Polare Darstellung der Graphen

4 Ausblick

4.1 Aus Dateien plotten

Gnuplot findet im wissenschaftlichen Bereich sehr viel Verwendung. Bei Messungen werden z.B. Daten direkt an Dateien angehangen und stündlich erstellt Gnuplot einen aktuellen Graph.

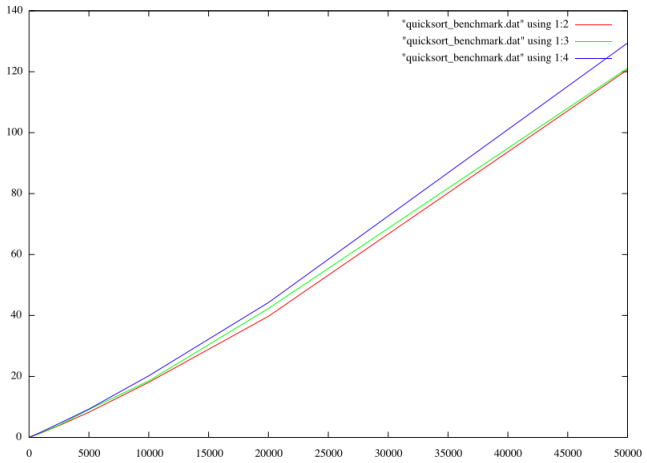
Syntax `gnuplot> plot '<dateiname>' using <x-Spalte>:<y-Spalte>`

Aufbau der Datei Spalten und Reihen - getrennt durch Whitespaces und Neue Zeilen.

#elements	first	median	random
10	0.01	0.01	0.01
100	0.12	0.12	0.14
1000	1.44	1.5	1.73
2500	3.85	3.97	4.52
5000	8.27	9.15	9.3
10000	18.09	18.54	20.22
20000	39.75	42.33	44.25
50000	120.69	121.18	129.45

Beispiel der Plotbefehle In diesem Beispiel wird der Quicksort-Algorithmus anhand der Wahl des Pivot-Elements verglichen.

```
gnuplot> set style data lines
gnuplot> plot 'quicksort_benchmark.dat' using 1:2
gnuplot> replot 'quicksort_benchmark.dat' using 1:3
gnuplot> replot 'quicksort_benchmark.dat' using 1:4
```



4.2 Skripte und Arbeitsabläufe

4.2.1 Gnuplot Skripte

Wenn man eine Reihe von Befehlen öfter ausführen will, kann man Gnuplot Skripte anlegen. Zeile für Zeile trägt man dort die Befehle ein, Kommentare erwirkt man durch die klassische Raute #.

```
   # author: trogdor at burnination dot net
set output 'latest_graph.ps'
set terminal postscript
plot 'dollars_on_my_account.dat'
```

4.2.2 Gnuplot Arbeitsabläufe

Speichern Die gegenwärtige Umgebung - aktuelle Plots und Umgebungsvariablen - kann man von der Konsole aus in eine Datei speichern:
`gnuplot> save 'MyWork.gnu'`

Laden Laden kann man seine Arbeit auf folgende Weise:
`gnuplot> load 'MyWork.gnu'`

4.3 Gnuplot von anderen Projekten aus ansprechen

Zahlreiche Programmiersprachen bieten eine API zu Gnuplot an. Z.B. kann man so direkt Diagramme für eine Website erstellen.

- Python http://gnuplot-py.sourceforge.net/
- Ruby http://rgplot.rubyforge.org/
- Falls für Sprache X nicht vorhanden: einfach über die Shell