

Computergrundlagen

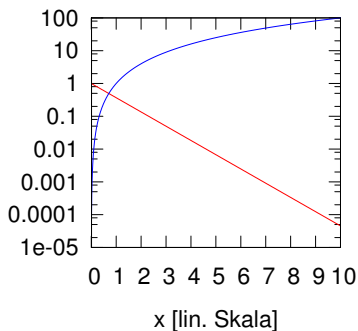
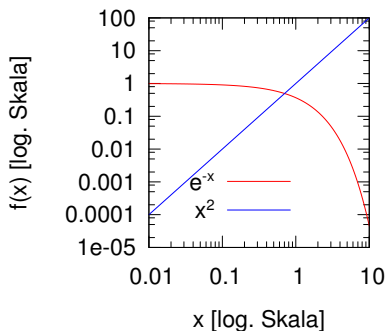
Visualisierung quick&dirty:

Gnuplot

Axel Arnold

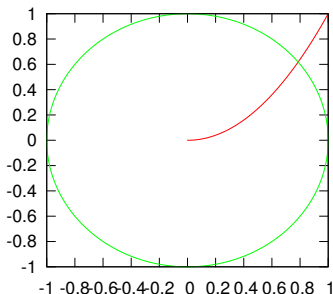
Institut für Computerphysik
Universität Stuttgart

Wintersemester 2011/12



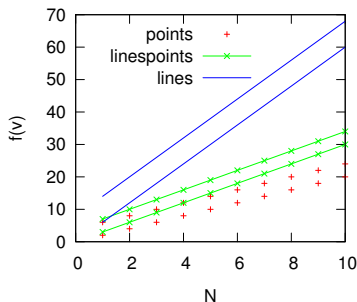
- Einfaches Kommandozeilentool zum Zeichnen von Funktionen
- Dateneingabe als Texttabellen
- Zahlreiche Ausgabeformate
- Sehr robuste und einfach zu bedienende Fitfunktion
- Befehl **help**: ausführliche Hilfe

```
plot x**2  
set parametric  
set trange [0:2*pi]  
plot sin(t), cos(t)
```



- Funktionsplots: Variable x
- Math. Funktionen wie in Python: $x**n$, **exp**, **sin**, **sqrt**, ...
- Parameterplots: **set parametric**, dann Funktion als $x(t)$, $y(t)$
- **set [x|t] range**: Zeichenbereich

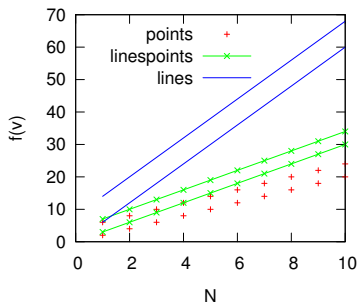
```
plot "test.dat" with points,\n      "" u 1:3 with linespoints
```



- Leerschritt-separierte Tabellen
- Leerzeilen trennen Blöcke (nicht durch Linien verbunden)
- Kommentare beginnen mit „#“
- Art der Darstellung:
 - **with** points, w p: Punkte (Default)
 - **with** linespoints, w lp: Punkte mit Linien verbunden
 - **with** lines, w l: nur Linien

#	N	f(N)	f2(N)
1	1	1	
2	2	4	
1	5	5	
2	6	8	

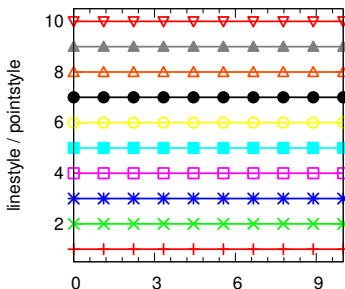
```
plot "test.dat" with points,\  
      "" u 1:3 with linespoints
```



- **using**, u: Spaltenauswahl
- Beispiele:
 - **using 1:2**
→ 2. Spalte als Funktion der 1.
(Default)
 - **using 3:4**
→ 4. Spalte als Funktion der 3.
 - **using 1:(2*\$4)**
→ 4. Spalte verdoppeln
 - **using (2*\$1):(\$4/\$1)**

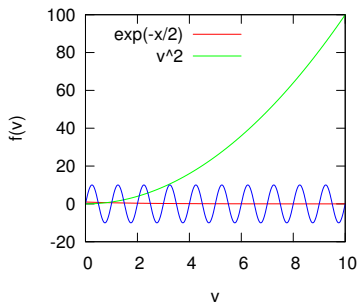
#	N	f(N)	f2(N)
1	1	1	
2	2	4	
1	5	5	
2	6	8	

```
set xtics 3
set mxtics 5
plot 1 w lp linestyle 1 lw 2 \
  pointtype 1 ps 2, \
  2 w lp linestyle 2 lw 2 \
  pointtype 2 ps 2
```



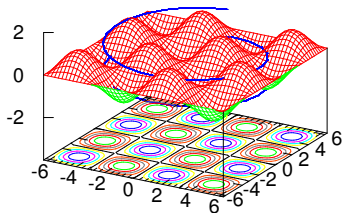
- **set [m][x|y]tics**: Achsenmarkierungen
- **linetype**, **lt**, **linecolor**, **lc** und **linewidth**, **lw**:
Linienstil, -farbe und -breite
- **pointtype**, **pt** und **pointsize**, **ps**: Form und Größe der Punkte,
Farbe entspricht immer der Linie
- Bedeutung der Zahlen hängt vom Ausgabegerät ab

```
set key left top
set xlabel "v"
set ylabel "f(v)"
set xrange [0:10]
plot exp(-x/2), \
    x**2 title "v^2", \
    10*sin(2*pi*x) notitle
```



- **set [x|y]label:** Achsenbeschriftung
- **set [x|y]range:** Achsenbereich
- **[no]title:** Name der Funktion in der Legende
- **set key [left|right] [top|bottom]:** Positionierung der Legende
- **set key off:** Legende ausschalten

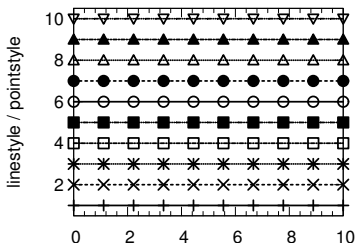
```
set isosamples 50
set hidden3d
set contour base
set cntrparam levels auto 10
splot [0:4*pi] [0:4*pi] \
      sin(x)*sin(y)
```



- **set hidden3d**: verdeckte Gitterlinien verbergen
- **set contour [base|surface|both]**: Kontourendarstellung
- **set cntrparam**: Auswahl der Kontourebenen
- **set isosamples**: Anzahl der Gitterlinien
- Linienstile usw. wie bei plot

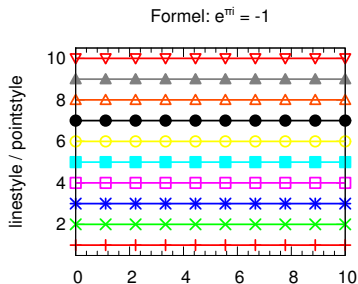
```
set term postscript eps
set out "test.eps"
...
unset out
```

Formel: $e^{\frac{1}{2} \rho} = -1$



- **set** out "name": Ausgabedatei
- **unset** out: Ausgabe schließen
- **set** term pdf: PDF-Ausgabe
- **set** term postscript: Postscript
- **set** term postscript eps: Encapsulated Postscript, zum Einbinden in Dokumente (\LaTeX)
- alternativ PDF aus EPS mit dem Kommandozeilentool `epstopdf`

```
set term postscript eps \  
  enhanced color solid \  
  rounded linewidth 4 \  
  font "Helvetica,18" size 12,5  
set out "test.eps"  
...
```

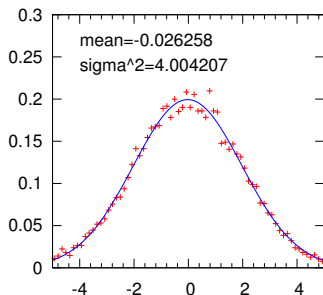


PDF- und Postscript-Optionen:

- color/mono: Farbig / Schwarz-Weiss
- solid/dashed: Durchgezogene / gestrichelte Linien
- size: Leinwand-Größe
- enhanced: Erweiterte Textlabels (hochstellen, Symbole)
- font: Auswahl des (Postscript-)Zeichensatzes

```
f(x,m,s)= 1./sqrt(2*pi*s) \  
          *exp(-0.5*(x-m)**2/s)
```

```
fit f(x,m,s) "histo.dat" via m, s  
plot "histo.dat", f(x,m,s)
```



- Fitten von beliebigen Funktionen an tabellierte Daten
- **using** funktioniert wie bei (s)**plot**
- Werte stehen anschließend als Variablen zur Verfügung
- Nützlich: Definition von Funktionen