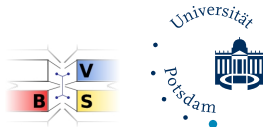


gnuplot-Vortrag auf dem BLIT 2010

Stefan Gasterstädt, Simon Kiertscher

Institut für Informatik
Universität Potsdam



6. November 2010

- Potsdam University
 - found. 1991
 - 5 Faculties



- Institute of Computer Science
 - Currently about 800 students in computer science (bachelor, master and diploma students)

- Research Topics

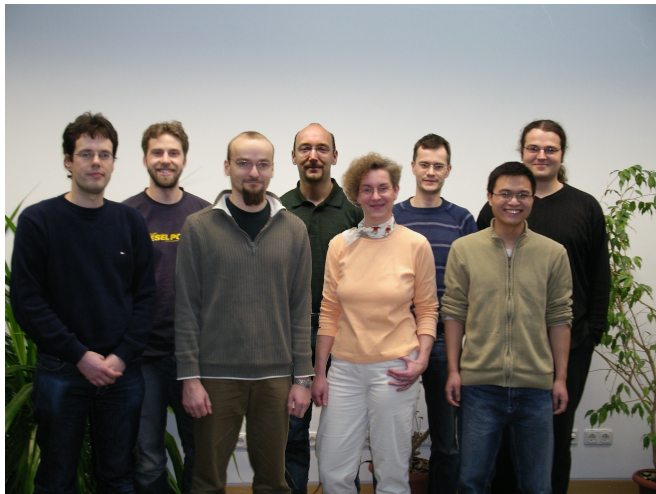
- Cluster Computing
Communication
Architectures for
High-Performance Cluster
Communication, Gigabit
Ethernet and InfiniBand
- Grid Computing, Grid
Services
Distributed Runs,
Job Resource Brokers,
Fault-Tolerant SOA

- Privacy & Security

- Privacy-Aware Data
Exchange in P2P
networks,
Secure VoIP Architecture



■ Our Team



Was ist gnuplot?

gnuplot is

- a command-driven interactive function plotting program
- can be used to plot functions and data points
- in both two- and three-dimensional plots
- in many different formats
- designed primarily for the visual display of scientific data
- copyrighted, but freely distributable

[gnuplot FAQ 1.1]

```
$ gnuplot
```

```
gnuplot> a=0.25
```

```
gnuplot> f(x)=sin(x)*a
```

```
gnuplot> plot f(x)
```

```
gnuplot> help plot
```

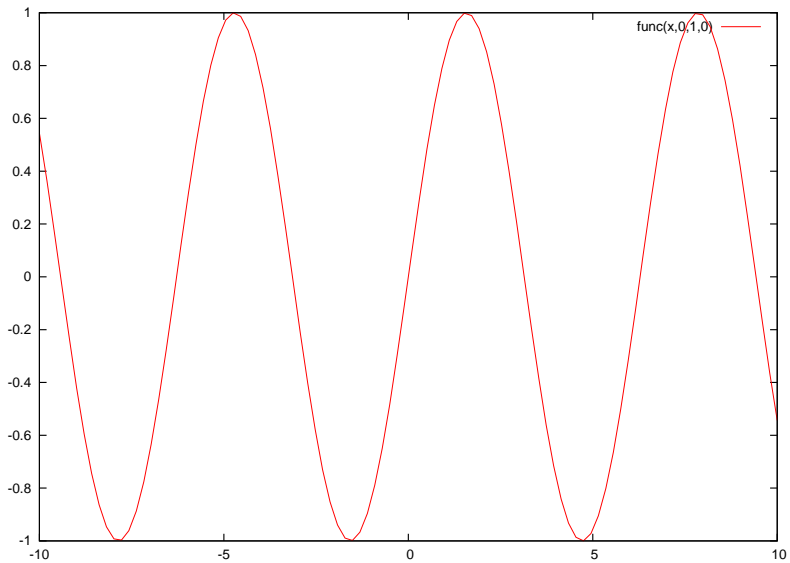
```
gnuplot> help style
```

```
$ cat function.plot
#!/usr/bin/env gnuplot

# set terminal environment
set term postscript eps enhanced color "Helvetica" 10

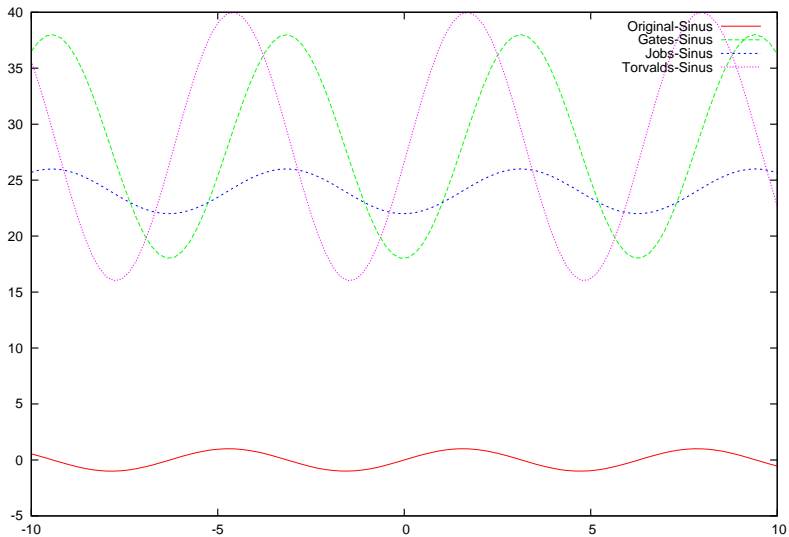
# define global parametrised function
func(x,s,a,f)=s+a*sin(x+f)

# specify output file
set output "function1.eps"
# plot function (default layout)
plot func(x,0,1,0)
```



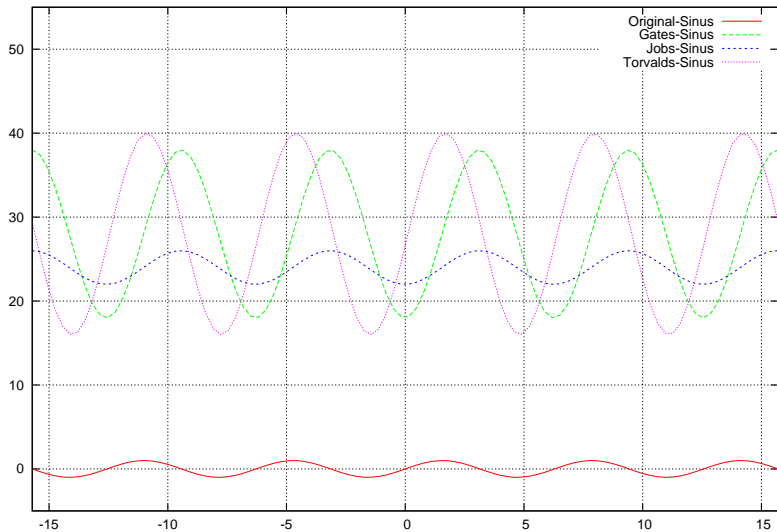

```
# set title
set title "Sinuskurven"
set output "function2.eps"
# plot three graphs in same figure
plot func(x,0,1,0) title "Original-Sinus", \
     func(x,28,10,55) title "Gates-Sinus", \
     func(x,24,2,55) title "Jobs-Sinus", \
     func(x,28,12,69) title "Torvalds-Sinus"
```

Sinuskurven



```
# show background grid
set grid
# modify ranges
set xrange[-5*pi:5*pi]
set yrange[-5:55]
set output "function3a.eps"
# plot same plot again
replot
```

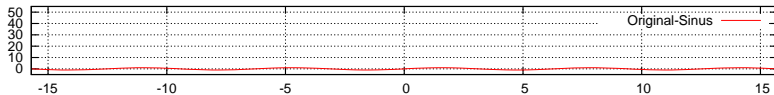
Sinuskurven



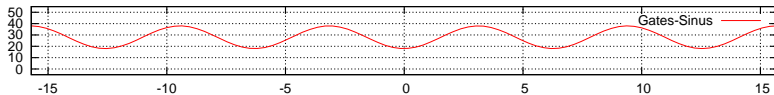
```
set output "function3b.eps"  
# plot three graphs using multiplot  
set multiplot layout 4,1 title "Sinus comparison"  
plot func(x,0,1,0) title "Original-Sinus"  
plot func(x,28,10,55) title "Gates-Sinus"  
plot func(x,24,2,55) title "Jobs-Sinus"  
plot func(x,28,12,69) title "Torvalds-Sinus"  
set nomultiplot
```

Sinus comparison

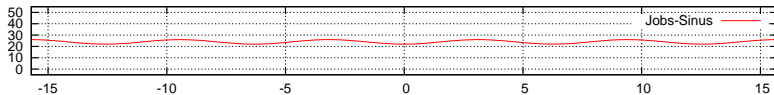
Sinuskurven



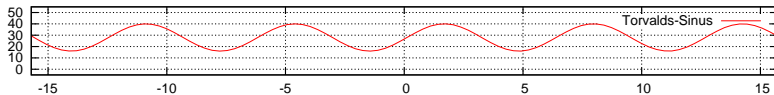
Sinuskurven



Sinuskurven

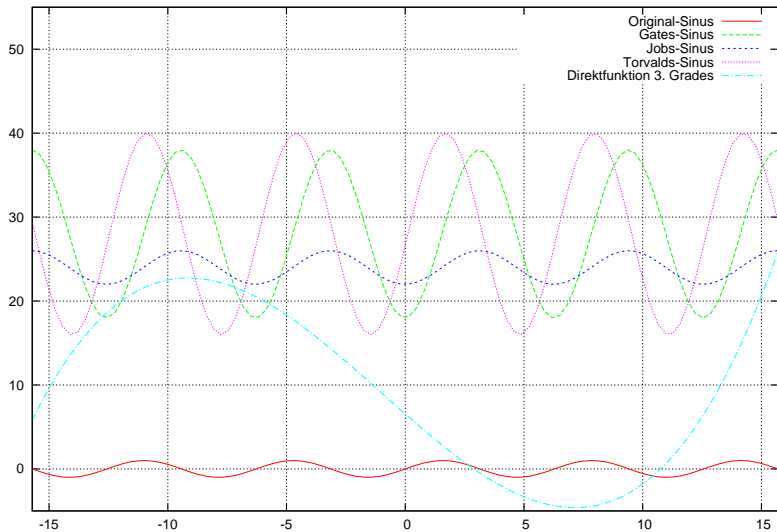


Sinuskurven



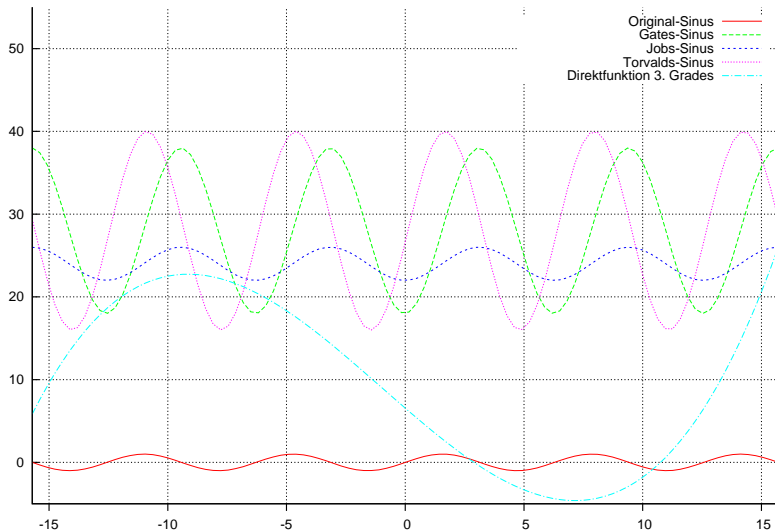
```
set output "function4.eps"
# plot three plus direct graph
plot func(x,0,1,0) title "Original-Sinus", \
      func(x,28,10,55) title "Gates-Sinus", \
      func(x,24,2,55) title "Jobs-Sinus", \
      func(x,28,12,69) title "Torvalds-Sinus", \
      (x-14)*(x+3)*(x+14)/79+14 title "Direktfunktion 3. Grades"
```

Sinuskurven



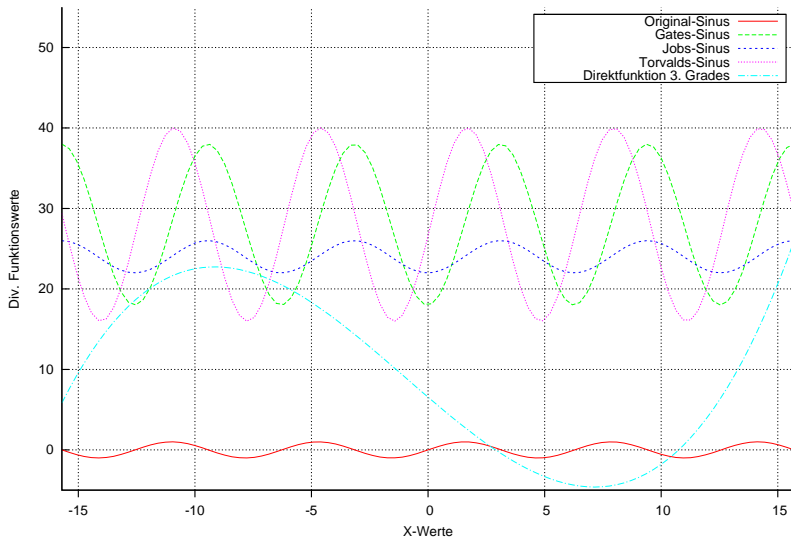

```
# modify tics
set tics out
set xtics nomirror
set ytics in nomirror
# modify border
set border 1+2
# set border 3
set output "function5.eps"
replot
```

Sinuskurven



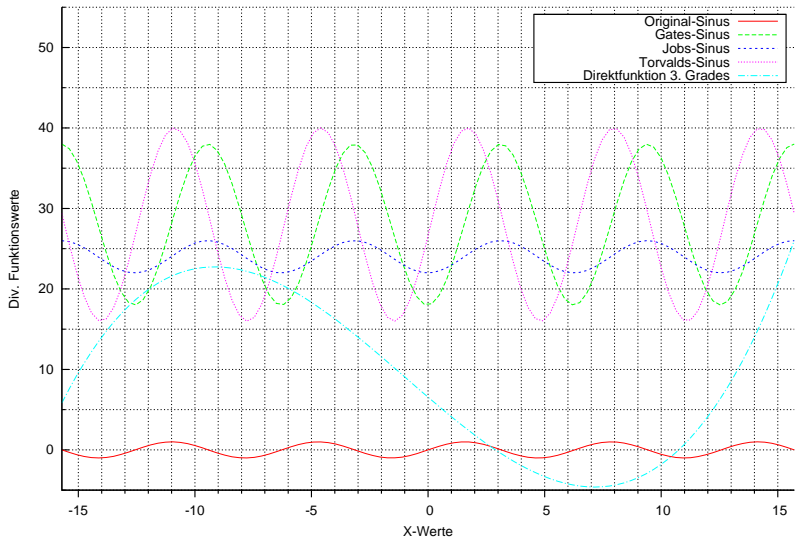
```
# set label
set xlabel "X-Werte"
set ylabel "Div. Funktionswerte"
# add border to legend
set key box linestyle -1
set output "function6.eps"
replot
```

Sinuskurven



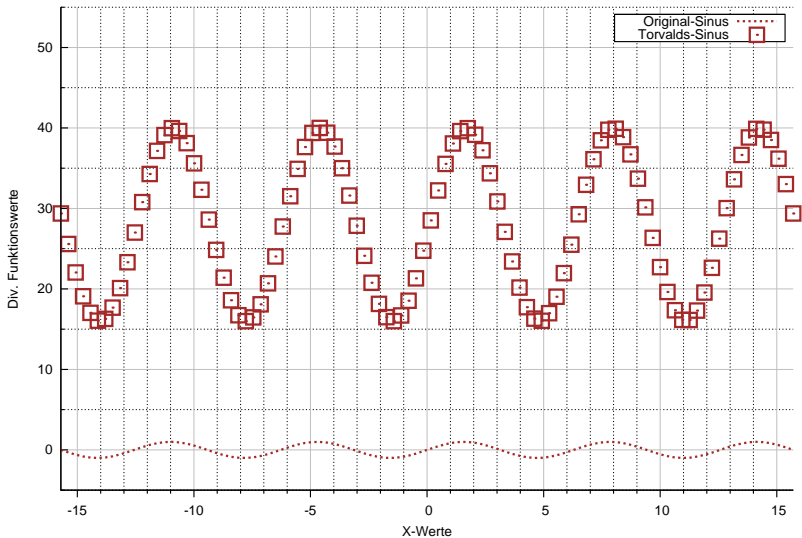
```
# add minor tics
set mxtics 5
set mytics 2
# modify grid
set grid xtics ytics mxtics mytics
set output "function7.eps"
replot
```

Sinuskurven



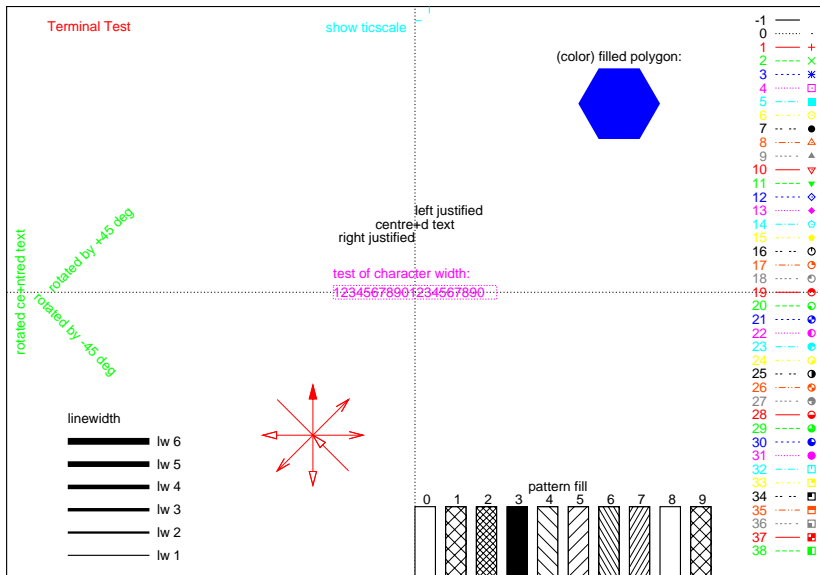
```
# define some line types
set style line 1 linetype rgb "#c0c0c0" linewidth 1
set style line 2 linetype 0 linewidth 1
set style line 3 linetype rgb "brown" linewidth 4 \
  pointtype 4 pointsize 2
# modify grid's lines
set grid xtics ytics mxtics mytics linestyle 1 ls 2
set output "function8.eps"
# plot using specific line/points types
plot func(x,0,1,0) title "Original-Sinus" with lines ls 3, \
  func(x,28,12,69) title "Torvalds-Sinus" with points linestyle
```

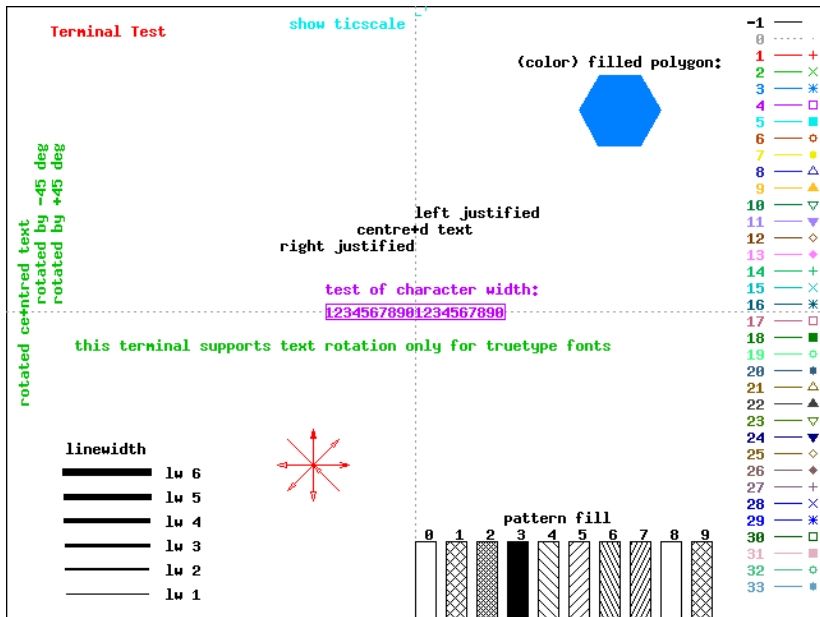
Sinuskurven




```
# Basiswerte  
set output "test.eps"  
test
```

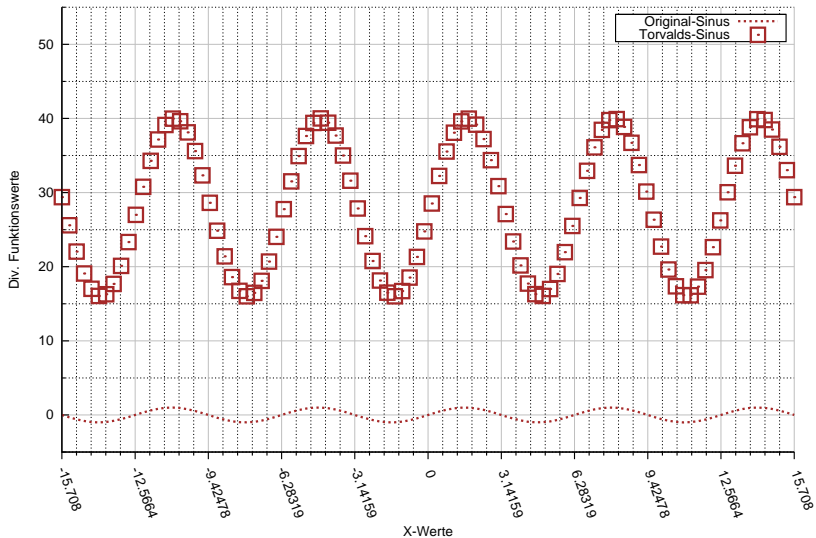
```
# Basiswerte  
set terminal png medium enhanced  
set output "test.png"  
test
```





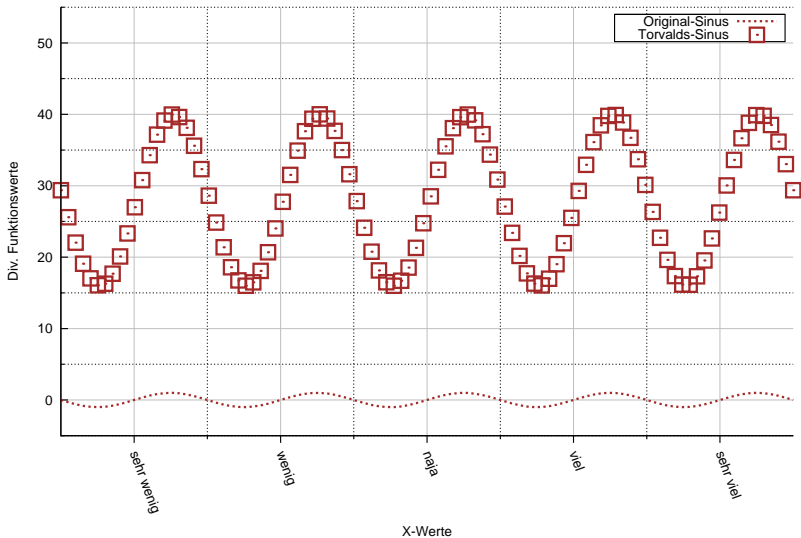
```
# rearrange tics
set xtics rotate by -70
set xtics pi
set output "function9.eps"
replot
```

Sinuskurven



```
# name tics
set xtics ("sehr wenig" -4*pi, "" -3*pi 1, "wenig" -2*pi, \
  "" -1*pi 1, "naja" 0, "" 1*pi 1, "viel" 2*pi, "" 3*pi 1, \
  "sehr viel" 4*pi)
set output "function10.eps"
replot
```

Sinuskurven



Gnuplot kann mathematische Berechnungen anstellen, die Funktionswerte plotten und es lassen sich dabei viele Darstellungsparameter anpassen:

- Achsen (Tiks, Mirror, ...)
- div. Beschriftungen
- Legende
- Positionierung
- Grid
- Benutzerspezifische Darstellung von Linien, Balken, Punkten, ...
- ...

Es existieren etliche weitere Darstellungsformen, die insbesondere für externe (Mess-) Daten geeignet sind.

- lines
- bars
- points (with lines (with errorbars))
- error bars, error lines, error boxen
- candle sticks
- ...


```
$ cat messung.data
```

```
1      10
```

```
2      20
```

```
4      40
```

```
5      50
```

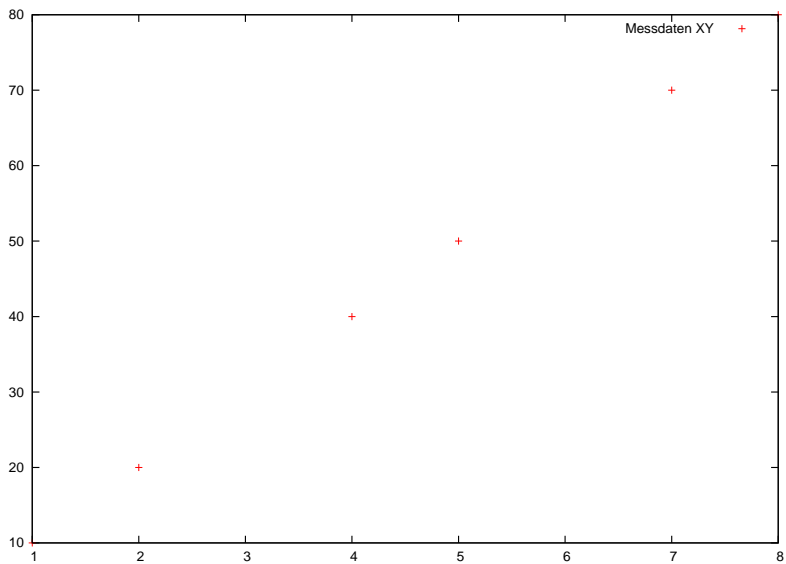
```
7      70
```

```
8      80
```

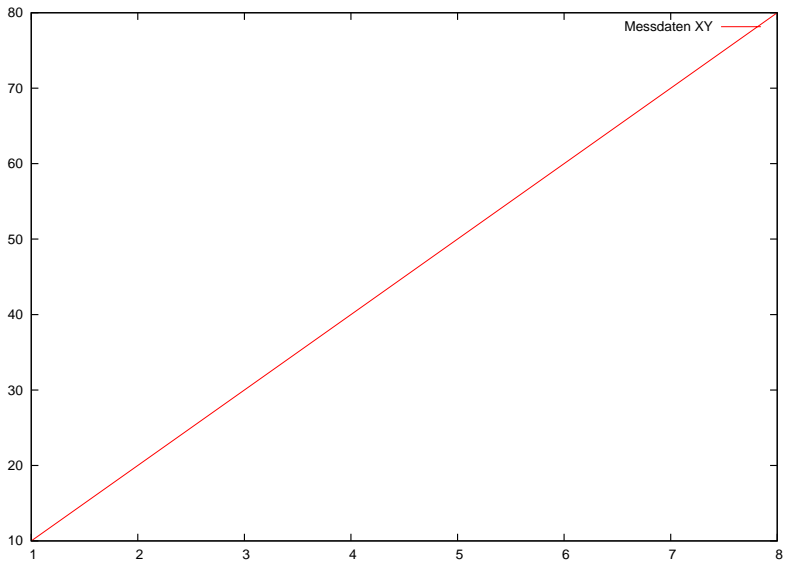
```
#!/usr/bin/env gnuplot

# set terminal environment
set term postscript eps enhanced color "Helvetica" 10

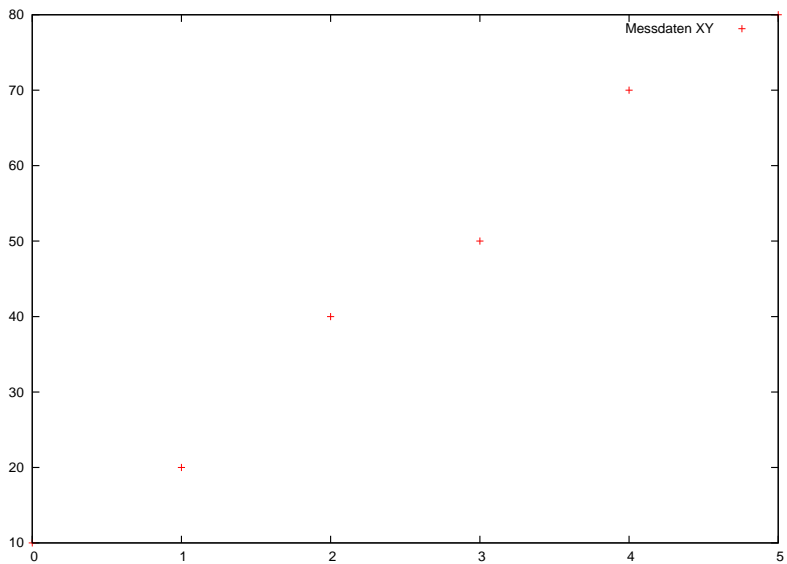
# specify output file
set output "messung1.eps"
# plot data using points
plot "messung.data" using 1:2 title "Messdaten XY" \
    with points
```



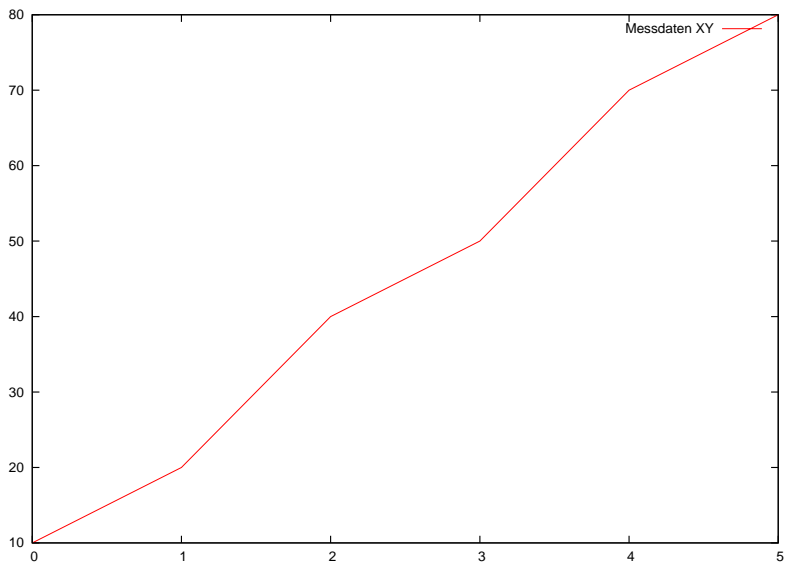
```
set output "messung2.eps"  
# plot data using lines  
plot "messung.data" using 1:2 title "Messdaten XY" \  
    with lines
```



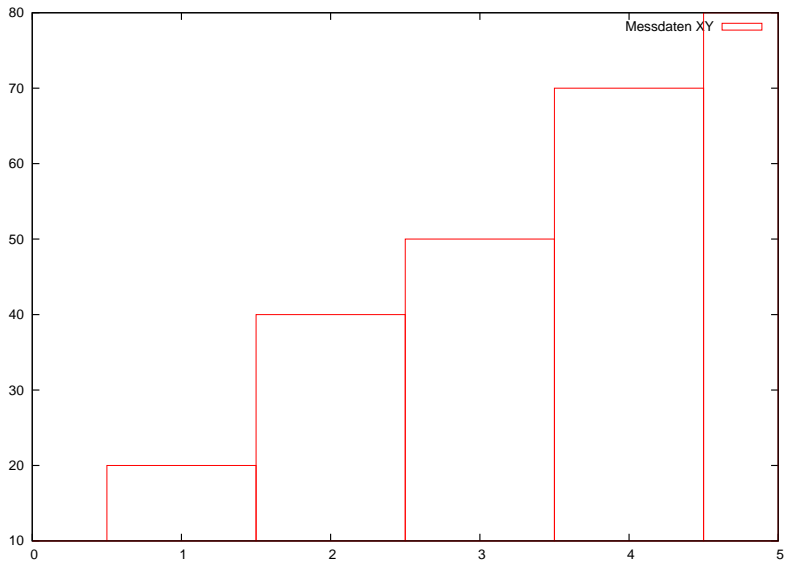
```
set output "messung3.eps"  
# plot data using index as x-value  
plot "messung.data" using ($0):2 title "Messdaten XY" \  
    with points
```



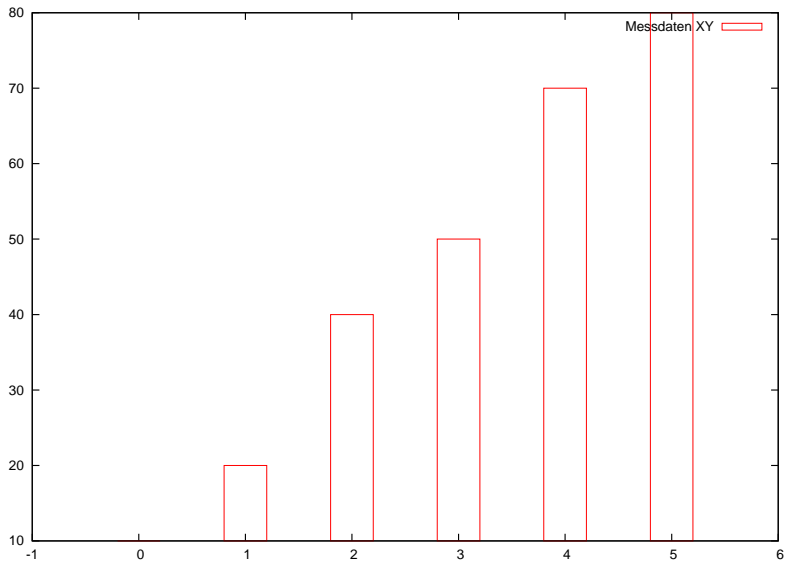
```
set output "messung4.eps"  
# plot data using lines  
plot "messung.data" using ($0):2 title "Messdaten XY" \  
    with lines
```

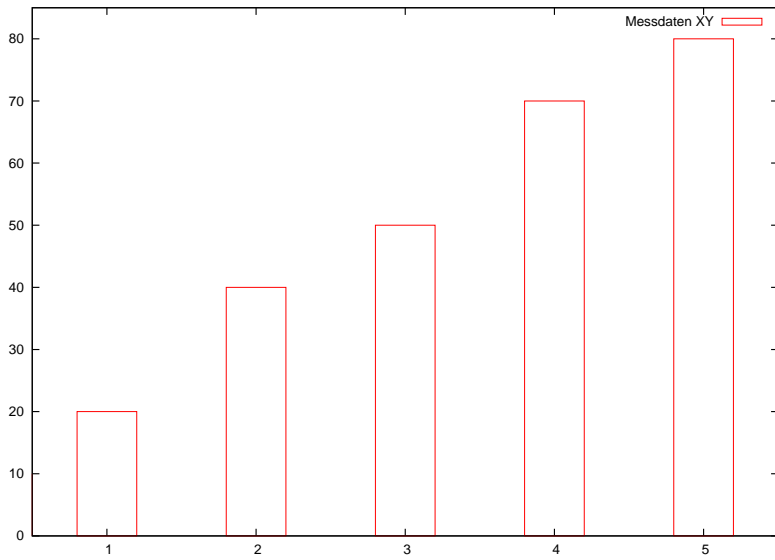
```
set output "messung5.eps"  
# plot data using boxes  
plot "messung.data" using ($0):2 title "Messdaten XY" \  
    with boxes
```



```
set output "messung6.eps"  
# plot data with explicit boxes' width  
plot "messung.data" using ($0):2:(0.4) title "Messdaten XY" \  
    with boxes
```



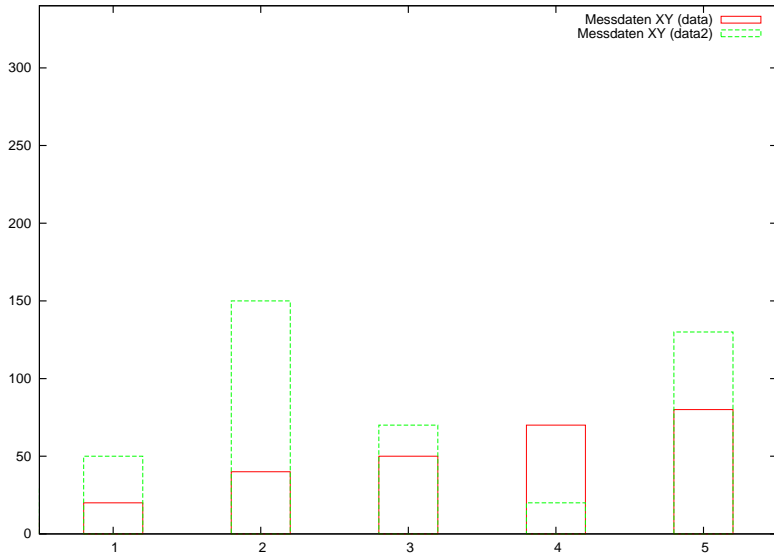
```
# modify ranges
set xrange [0.5:5.5]
set yrange [0:85]
set output "messung7.eps"
# plot same plot again
replot
```



```
$ cat messung.data2  
4      30  
7      50  
13     150  
11     70  
2      20  
9      130
```



```
# modify y-range
set yrange [0:340]
set output "messung8a.eps"
# plot data out of two different files
plot "messung.data" using ($0):2:(0.4) \
  title "Messdaten XY (data)" with boxes, \
  "messung.data2" using ($0):2:(0.4) \
  title "Messdaten XY (data2)" with boxes
```



```
$ cat messung.data3
```

```
1      10
```

```
2      20
```

```
4      40
```

```
5      50
```

```
7      70
```

```
8      80
```

```
4      30
```

```
7      50
```

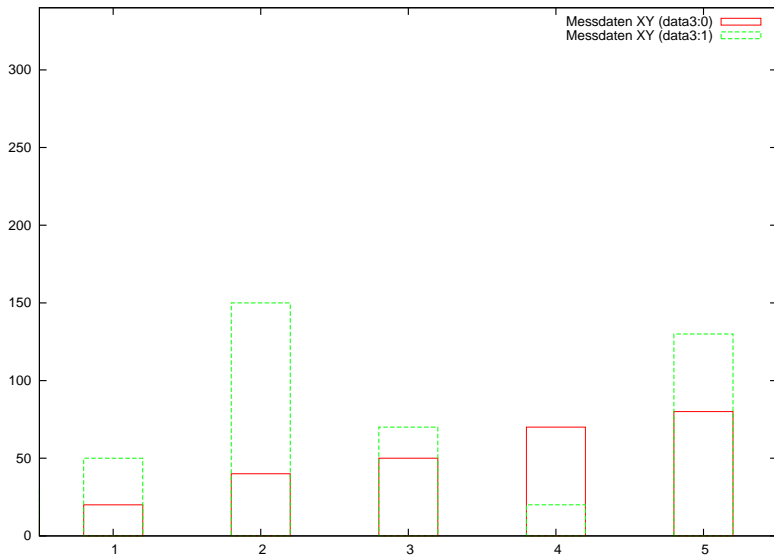
```
13     150
```

```
11     70
```

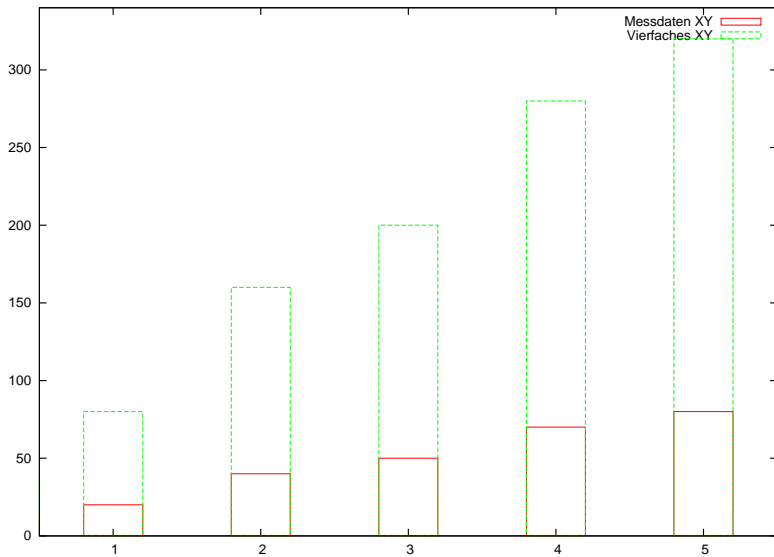
```
2      20
```

```
9      130
```

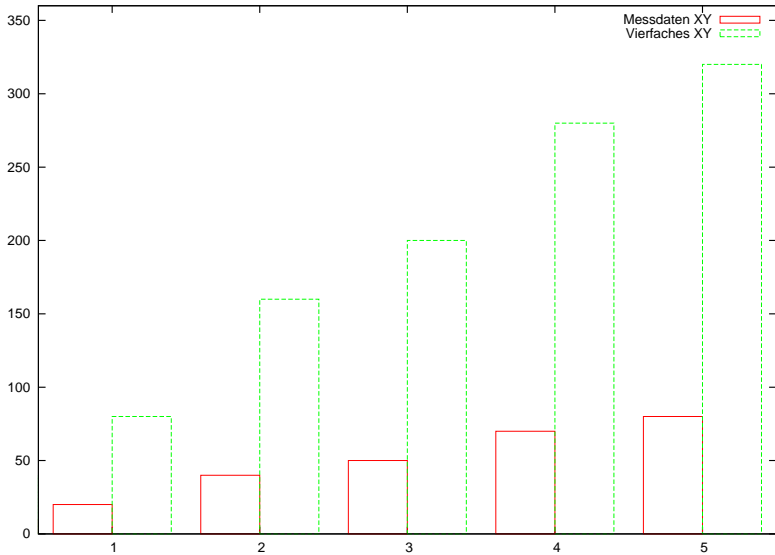
```
set output "messung8b.eps"  
# plot data out of one file using index  
plot "messung.data3" index 0 using ($0):2:(0.4) \  
    title "Messdaten XY (data3:0)" with boxes, \  
    "messung.data3" index 1 using ($0):2:(0.4) \  
    title "Messdaten XY (data3:1)" with boxes
```



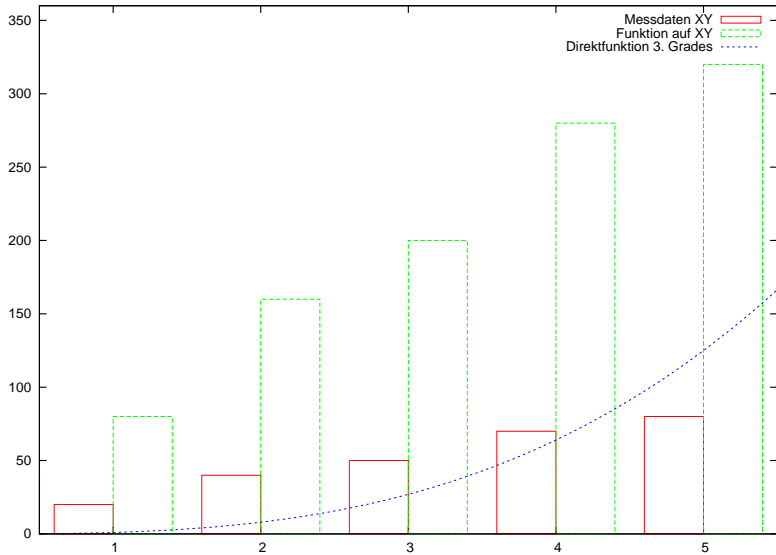
```
set output "messung9.eps"  
# plot data out of one file, plus enlarged data plot  
plot "messung.data3" index 0 using ($0):2:(0.4) \  
    title "Messdaten XY" with boxes, \  
    "messung.data3" index 0 using ($0):($2*4):(0.4) \  
    title "Vierfaches XY" with boxes
```



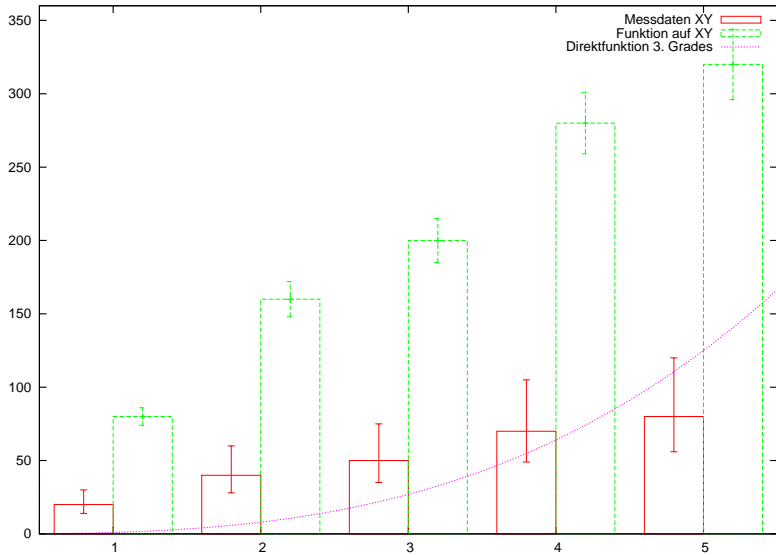
```
# modify y-range
set yrange [0:360]
set output "messung10.eps"
# plot two boxes next to each other by shifting its x-values
plot "messung.data3" index 0 using ($0-0.2):2:(0.4) \
    title "Messdaten XY" with boxes, \
    "messung.data3" index 0 using ($0+0.2):($2*4):(0.4) \
    title "Vierfaches XY" with boxes
```

```
# you still can use functions! (outside defined and inside)
fact=4
func(x)=x*fact
set output "messung11.eps"
plot "messung.data3" index 0 using ($0-0.2):2:(0.4) \
  title "Messdaten XY" with boxes, \
  "messung.data3" index 0 using ($0+0.2):(func($2)):(0.4) \
  title "Funktion auf XY" with boxes, \
  x*x*x title "Direktfunktion 3. Grades"
```



```
# same plot plus (calculated) boxerrorbars
# without additional legend entry
fact=4
func(x)=x*fact
set output "messung11.eps"
plot "messung.data3" index 0 \
    using ($0-0.2):2:($2-3*$1):($2+5*$1):(0.4) \
    title "Messdaten XY" with boxerrorbars, \
    "messung.data3" index 0 using ($0+0.2):(func($2)):(0.4) \
    title "Funktion auf XY" with boxes, \
    "messung.data3" index 0 using ($0+0.2):(func($2)):(3*$1) \
    notitle with errorbars linestyle 2, \
    x**3 title "Direktfunktion 3. Grades"
```



Is gnuplot suitable for scripting?

Yes! Gnuplot can

- read in files containing additional commands during an interactive session
- be run in batch mode by piping a pre-existing file or a stream of commands to stdin
- easily be wrapped in a cgi script for use as a web-driven plot generator

[gnuplot FAQ 1.4]

Wie übergibt man Werte (Parameter) zu Gnuplot?

Variante #1 Integration in ein Shell-Skript:

```
$ cat someplot.sh
#!/usr/bin/env bash
# synopsis ./someplot.sh datafile plotfile "Some Title"

gnuplot << EOF
set term postscript eps enhanced color "Helvetica" 12
set xrange [0.5:5.5]
set yrange [0:85]
set output "$2.eps"
plot "$1.data" using (\$0):2:(0.4) title "$3" \
    with boxes
EOF
```

Variante #2 Seit Version 4.2.5 gibt es einen speziellen Parameter

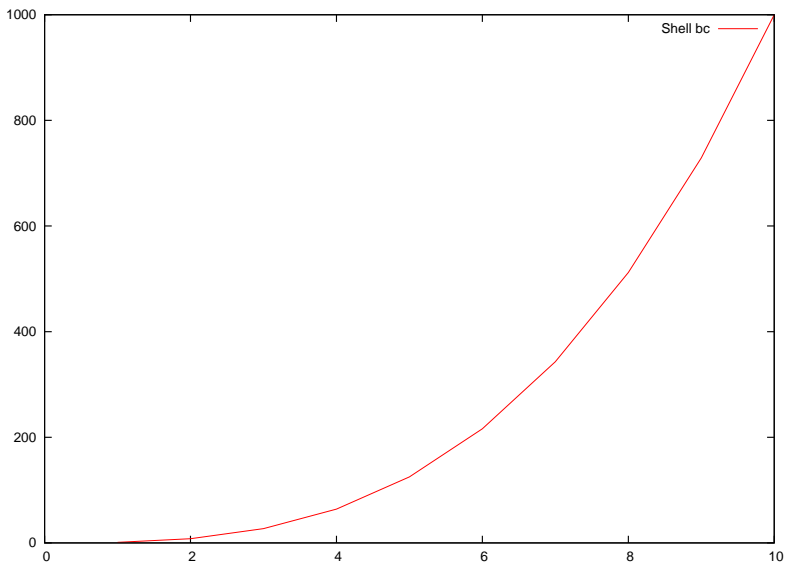
```
$ cat someother.plot
#!/usr/bin/env gnuplot

set term postscript eps enhanced color "Helvetica" 12
set xrange [0.5:5.5]
set yrange [0:85]
set output PLOT.".eps"
plot SOURCE.".data" using (\$0):2:(0.4) title "" .TITLE \
    with boxes
EOF

$ gnuplot \
> -e "SOURCE='messung';PLOT='inject-e';TITLE='Messung ABC'" \
> someother.plot
```


- Variante #3 Mittels Backticks können Sie an fast jeder Stelle beliebige Shell-Aufrufe starten!
 - Mit “!” kann ebenfalls ein Shell-Befehl gestartet werden.
 - Wie wäre es mit sort, awk, bc, python, perl, ...

```
# inline shell scripts
set xrange [0:10]
set yrange [0:1000]
set output "messung13.eps"
plot "< for i in 1 2 3 4 5 6 7 8 9 10; do \  
echo -n \"\$i \"; echo \"\$i^3\" | bc; done" using 1:2 \  
title "Shell bc" with lines
```



- plotiucs (<http://ploticus.sourceforge.net/>)
- R (<http://www.r-project.org/>)
- Octave (<http://www.gnu.org/software/octave/>)
- Yorick (<http://yorick.sourceforge.net/>)

- Gnuplot Homepage: <http://www.gnuplot.org/>
- Demo scripts for gnuplot:
<http://gnuplot.sourceforge.net/demo/>
- Gnuplot Manual:
<http://www.gnuplot.info/docs/gnuplot.html>
- gnuplot tips - not so Frequently Asked Questions:
<http://t16web.lanl.gov/Kawano/gnuplot/index-e.html>
- Dr. David Völker, Grundkurs Gnuplot: <http://userpage.fu-berlin.de/~voelker/gnuplotkurs/gnuplotkurs.html>
- Stephan ???, Crashkurs Gnuplot:
<http://www-stud.uni-due.de/~sdstscwe/index.php/Artikel/1-Sonstige%20Artikel/51-crashkurs-gnuplot>
- errorbar gnuplot demo: http://www.csse.uwa.edu.au/programming/gnuplot_demos/errorbar/errorbar.html
- HTML Help Books (u. a. Gnuplot): <http://code.google.com/p/htmlhelp/downloads/list?q=label:CHM>

Mehrere Messungdaten einer Messung in einer Datei:

```
$ cat eins_output.dat
# tcp messungen
#      paket size      latency      bandwidth
      1      56.568015      0.017678
      2      56.641127      0.035310
      3      56.958570      0.052670
      ...
1048576  89282.195221      11.744514

# sctp messungen
      1      60.016234      0.016662
      2      59.682547      0.033511
      ...
```

Setzen des Ausgabe Terminals:

```
set term postscript eps enhanced color "Helvetica" 10
```

Rahmen setzen:

```
set border 3
```

Grids anschalten:

```
set grid
```

Benutzerdefinierte Linien-Typen und Farben setzen:

```
set style line 1 linetype 1 linecolor rgb "dark-blue"  
set style line 2 linetype 1 linecolor rgb "dark-orange"  
set style line 3 linetype 1 linecolor rgb "dark-green"  
set style increment user
```

Tics an den Achsen nach außen verlagern:

```
set tics out
```

Spiegelung der Tics abschalten:

```
set xtics nomirror  
set ytics nomirror
```

Label der Achsen setzen:

```
set xlabel "packet size [byte]"  
set ylabel "bandwith [MB]"
```

Logarithmische x-Achse:

```
set logscale x 2
```


Ausgabe setzen:

```
set output "bandwith.eps"
```

Titel der Grafik:

```
set title "TCP/SCTP: bandwidth"
```

y-Range beeinflussen (um Legende sichtbar zu machen):

```
set yrange [:15]
```

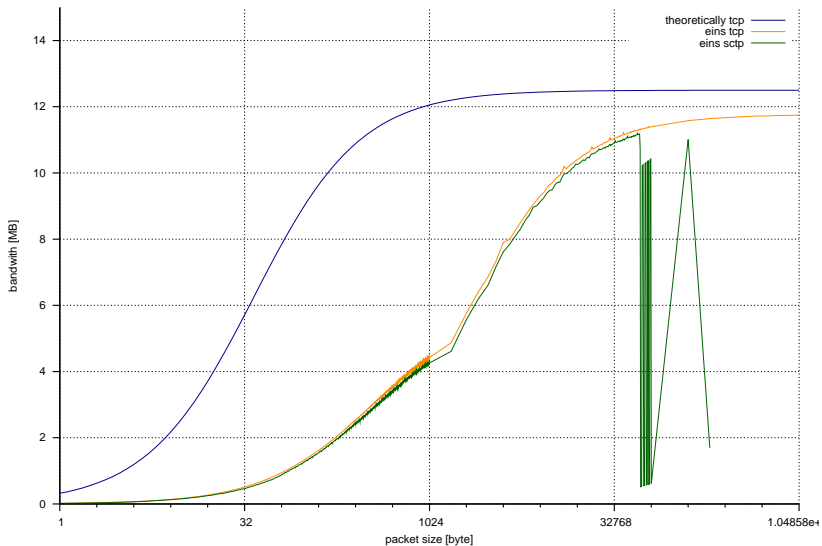
Plot style lines:

```
set data style lines
```

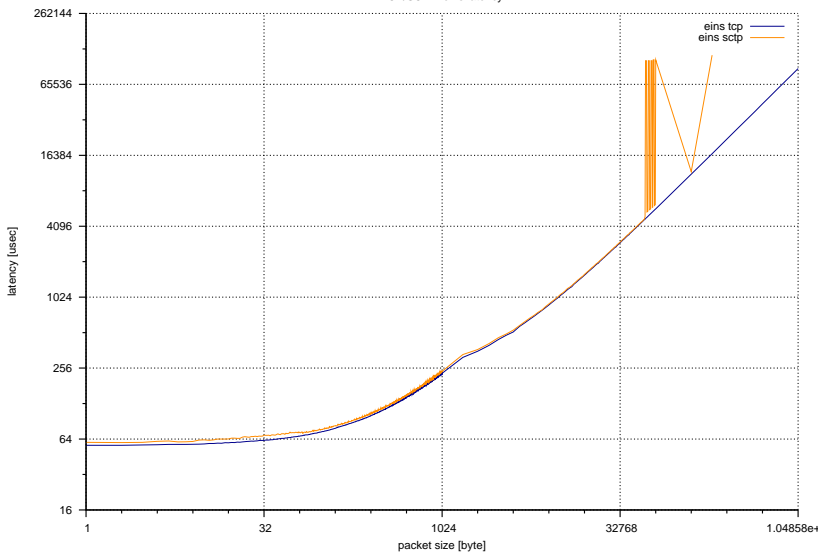
Plotten der eigentlichen Daten:

```
plot "output.dat" index 0 using 1:2 title "tcp", \  
      "output.dat" index 1 using 1:2 title "sctp"
```

TCP/SCTP: eins bandwidth



TCP/SCTP: eins latency



- Daten liegen als Notenliste vor (Matr.Nr., Punkte, Note)

```
100000  15,0  3,7
```

```
100004  0,0  5,0
```

```
...
```

```
100132  23,5  2,0
```

- Daten aufbereiten (Note, Anzahl)

```
# Note  Anzahl
```

```
1.0  1
```

```
1.3  2
```

```
...
```

```
5.0  3
```

- Darstellung der „Messdaten“ als Blockdiagramm und überlagerter Glättungskurve

Direkte Angabe für die Balkendarstellung (fs) und
ge"smooth"e Werte für die Kurve

```
plot "notenverteilung.dat" using 1:2:(0.2) title "Haeufigkeit" \  
with boxes fs solid 0.7, \  
"notenverteilung.dat" using 1:2 smooth bezier \  
title "Verteilungskurve" with lines
```

Notenverteilung der Klausur XY

