

GNU plot

GNU plot

Install it: `sudo apt install gnuplot`

Official website with user manual: <http://www.gnuplot.info/>

Simply type `gnuplot` in the terminal to use it

Simply type "help" in its environment will allow you to navigate through all its functions and usages

Simple function plot by `plot function(x)`

Function	Returns
abs(x)	absolute value of x, x
acos(x)	arc-cosine of x
asin(x)	arc-sine of x
atan(x)	arc-tangent of x
cos(x)	cosine of x, x is in radians.
cosh(x)	hyperbolic cosine of x, x is in radians
erf(x)	error function of x
exp(x)	exponential function of x, base e
inverf(x)	inverse error function of x
invnorm(x)	inverse normal distribution of x
log(x)	log of x, base e
log10(x)	log of x, base 10
norm(x)	normal Gaussian distribution function
rand(x)	pseudo-random number generator
sgn(x)	1 if x > 0, -1 if x < 0, 0 if x=0
sin(x)	sine of x, x is in radians
sinh(x)	hyperbolic sine of x, x is in radians
sqrt(x)	the square root of x
tan(x)	tangent of x, x is in radians
tanh(x)	hyperbolic tangent of x, x is in radians

Bessel, gamma, ibeta, igamma, and lgamma functions are also supported. Many functions can take complex arguments. Binary and unary operators are also supported.

More on using plot

Syntax:

```
plot {[ranges]}  
  {[function] | {"[datafile]" {datafile-modifiers}}}  
  {axes [axes] } { [title-spec] } {with [style] }  
  {, {definitions,} [function] ...}
```

```
gnuplot> plot [-3:3] abs(x)
```

```
gnuplot> plot [t=1:10] [-pi:pi*2] tan(t)
```

```
gnuplot> plot sin(x) title 'Sine Function', tan(x) title 'Tangent'
```

```
gnuplot> plot 'example.txt'          # plot file
```

```
gnuplot> plot "force.dat" using 1:2 title 'Column', "force.dat" using 1:3 title 'Beam' # multi column data
```

```
gnuplot> !ls                        # using terminal commands
```

```
gnuplot> plot "force.dat" using 1:2 title 'Column' with lines, \    #with one of these styles: lines, points, linespoints, impulses, dots,
```

```
    "force.dat" u 1:3 t 'Beam' w linespoints    #steps, fsteps, histeps, errorbars, xerrorbars, yerrorbars, xyerrorbars, vector
```

```
    # boxerrorbars, boxxyerrorbars, financebars, candlesticks
```

```
# This file is called  force.dat  
# Force-Deflection data for a beam and a bar  
# Deflection    Col-Force    Beam-Force  
0.000           0           0  
0.001          104          51  
0.002          202         101  
0.003          298         148  
0.0031          290         149  
0.004          289         201  
0.0041          291         209  
0.005          310         250  
0.010          311         260  
0.020          280         240
```

Setting

Customization of the axis ranges, axis labels, and plot title, as well as many other features, are specified using the set command at any time!

Create a title:	> set title "Force-Deflection Data"
Put a label on the x-axis:	> set xlabel "Deflection (meters)"
Put a label on the y-axis:	> set ylabel "Force (kN)"
Change the x-axis range:	> set xrange [0.001:0.005]
Change the y-axis range:	> set yrange [20:500]
Have Gnuplot determine ranges:	> set autoscale
Move the key:	> set key 0.01,100
Delete the key:	> unset key
Put a label on the plot:	> set label "yield point" at 0.003, 260
Remove all labels:	> unset label
Plot using log-axes:	> set logscale
Plot using log-axes on y-axis:	> unset logscale; set logscale y
Change the tic-marks:	> set xtics (0.002,0.004,0.006,0.008)
Return to the default tics:	> unset xtics; set xtics auto

Using scripts

```
# Gnuplot script file for plotting data in file "force.dat"
# This file is called force.p
set terminal png
set output "printme.png"
set autoscale          # scale axes automatically
unset log              # remove any log-scaling
unset label            # remove any previous labels
set xtic auto          # set xtics automatically
set ytic auto          # set ytics automatically
set title "Force Deflection Data for a Beam and a Column"
set xlabel "Deflection (meters)"
set ylabel "Force (kN)"
set label "Yield Point" at 0.003,260
set arrow from 0.0028,250 to 0.003,280
set xr [0.0:0.022]
set yr [0:325]
plot "force.dat" using 1:2 title 'Column' with linespoints , \
    "force.dat" using 1:3 title 'Beam' with points
```

Then the total plot can be generated with the command: `gnuplot> load 'force.p'`

Curve-fitting

To fit the data in force.dat with a function use the commands:

```
f1(x) = a1*tanh(x/b1)      # define the function to be fit
a1 = 300; b1 = 0.005;      # initial guess for a1 and b1
fit f1(x) 'force.dat' using 1:2 via a1, b1
f2(x) = a2 * tanh(x/b2)    # define the function to be fit
a2 = 300; b2 = 0.005;      # initial guess for a and b
fit f2(x) 'force.dat' using 1:3 via a2, b2
```

The curve-fit and data may now be plotted with the commands:

```
set key 0.018,150 title "F(x) = A tanh (x/B)"      # title to key!
set title "Force Deflection Data \n and curve fit"  # note newline!
set pointsize 1.5                                # larger point!
set xlabel 'Deflection, {/Symbol D}_x (m)'          # Greek symbols!
set ylabel 'Force, {/Times-Italic F}_A, (kN)'       # italics!
plot "force.dat" using 1:2 title 'Column data' with points 3, \
    "force.dat" using 1:3 title 'Beam data' with points 4, \
    a1 * tanh( x / b1 ) title 'Column-fit: A=309, B=0.00227', \
    a2 * tanh( x / b2 ) title 'Beam-fit: A=260, B=0.00415'
```