

GNU PLOT

- It's a command line driven plotting program
- It can plot 2D, 3D plots of functions, data and data fit
- It's plotting engine can be used by third party applications
- Completely open source and under active development since 1986

GNUPLOT Features

- Plotting engine can be used from various programming languages and by third party applications
- Supports piping
- Can be used interactively and well as in batch mode using scripts
- Can produce output directly on screen or in many standard graphics file format
- Capable of producing LATEX code that can be used directly in LATEX documents

Plotting

- **plot [initial x : final x] function(x)**

Plots `function(x)` with `x` ranging from `initial x` till `final x`

- **set xlabel "label"**
set ylabel "label"
set title "title"
- **set xrange [initial : final]**
set yrange [initial : final]

Plotting

- **plot** “data-file” using column#1:column#2, “data-file” using column#1:column#3

plots the data-file using column#1 as x-axis and column#2, column#3 as y-axis

- **plot function(x) with lines/points/steps/impulses**
- **replot**
Re-plots the graph
- **help command**

Fitting

- **plot “data-file” using 1:2:3 with yerrorbars**
- **f(x)=function(x;a,b,c....)**
where, a,b,c... are parameters
- **fit f(x) “data-file” using 1:2:3 via a,b,c...**
- **p f(x), “data-file” using 1:2:3 w yerr**

Postscript Output

- **set term postscript eps enhanced color**
set output “output-filename”
plot

Load / Save File

- **save** “filename”
- **load** “filename”

Parametric Plot

- **set parametric**
- **set trange [intial : final]**
- **y(t)=y_function(t)**
x(t)=x_function(t)
plot x(t),y(t)

Polar Plot

- **set polar**
set angle degrees/radians
plot $f(t)$

3D Plot

- **splot f(x,y)**
- **set isosample #x,#y**
sets the x and y coordinate sampling mesh density
- **set ticslevel 0**
sets the zero of z-axis in x-y plane
- **set view rot_x, rot_z, scale, scale_z**
sets the viewing angle
- **set hidden3d**
sets the mesh opaque

3D Plot

- **set pm3d**

draws colour mapped 3D plot

- **set palette defined (-3 “blue”, 0 “white”, 1 “red”)**

splot ... with pm3d

assigns colour to numerical values and plots accordingly with colour gradient

3D data plotting

- **splot** “data-file” **using**
column#1:column#2:column#3 **with lines**
- **set dgrid3d** **x_mesh, y_mesh**
generates 3D grid graph from data

Definition of Function

- Say $f(x) = f_a(x)$ when x *conditional a*
= $f_b(x)$ otherwise

$$f(x) = (x \text{ conditional } a) ? f_a(x) : f_b(x)$$

- If $f(xa)$ is related to $f(xb(xa))$ by a recursive relation

$$f(xa) = g(f(xb(xa)))$$

Then the function can be defined recursively in gnuplot. The recursive definition loop maybe terminated using conditional statements

Example of Recursive Function

- $N! = N * (N-1)!$

$$\mathbf{fac(x) = x*fac(x-1)}$$

this is an infinite loop

- To terminate the loop when $n=0$

$$\mathbf{fac(x) = (x==0) ? 1 : x*fac(x-1)}$$

when $n=0$, $fac(n)$'s value is given by 1 ie, the function immediately succeeding '?', otherwise $n*fac(n-1)$ is evaluated

- Since 'n' is integer

$$\mathbf{fac(x) = (int(x)==0) ? 1.0 : int(x)*fac(int(x)-1.0)}$$