gnuplot Quick Reference

Updated by Hans-Bernhard Bröker, April 2004

Starting gnuplot

to enter gnuplot	`gnuplot`
to enter batch gnuplot	`gnuplot macro_file`
to pipe commands to gnuplot	`application | gnuplot`

see below for environment variables you might want to change before entering gnuplot.

Exiting gnuplot

exit gnuplot	`quit`

All gnuplot commands can be abbreviated to the first few unique letters, usually three characters. This reference uses the complete name for clarity.

Getting Help

introductory help	`help plot`
help on a topic	`help <topic>`
list of all help available	`help or ?`
show current environment	`show all`

Command-line Editing

The UNIX, MS-DOS and VMS versions of gnuplot support command-line editing and a command history. EMACS style editing is supported.

Line Editing:

move back a single character	`^ B`
move forward a single character	`^ F`
moves to the beginning of the line	`^ A`
moves to the end of the line	`^ E`
delete the previous character	`^ H and DEL`
deletes the current character	`^ D`
deletes to the end of line	`^ K`
redraws line in case it gets trashed	`^ L,^ R`
deletes the entire line	`^ U`
deletes the last word	`^ W`

History:

moves back through history	`^ P`
moves forward through history	`^ N`

The following arrow keys may be used on most PC versions if READLINE is used.

IBM PC Arrow Keys:

Left Arrow	`same as ^ B`
Right Arrow	`same as ^ F`
Ctrl Left Arrow	`same as ^ A`
Ctrl Right Arrow	`same as ^ E`
Up Arrow	`same as ^ P`
Down Arrow	`same as ^ N`

Graphics Devices

All screen graphics devices are specified by names
a startup file (.gnuplot in UNIX). If you change
`replot` command or recreate it repeating the `loa`

get a list of valid devices	s

Graphics Terminals:

Mac OS X	s
AED 512 Terminal	s
AED 767 Terminal	s
Amiga	s
Adobe Illustrator 3.0 Format	s
Apollo graphics primitive, rescalable	s
Atari ST	s
BBN Bitgraph Terminal	s
SCO CGI Driver	s
Apollo graphics primitive, fixed window	s
SGI GL window	s
MS-DOS Kermit Tek4010 term - color	s
MS-DOS Kermit Tek4010 term - mono	s
NeXTstep window system	s
OS/2 Presentation Manager	s
REGIS graphics language	s
Selanar Tek Terminal	s
SunView window system	s
Tektronix 4106, 4107, 4109 & 420X	s
Tektronix 4010; most TEK emulators	s
VAX UIS window system	s
VT-like tek40xx terminal emulator	s
UNIX plotting (not always supplied)	s
AT&T 3b1 or 7300 UNIXPC	s
MS Windows	s
X11 default display device	s
X11 multicolor point default device	s

Turbo C PC Graphics Modes:

Hercules	s
Color Graphics Adaptor	s
Monochrome CGA	s
Extended Graphics Adaptor	s
VGA	s
Monochrome VGA	s
Super VGA - requires SVGA driver	s
AT&T 6300 Micro	s

Hardcopy Devices:

Unknown - not a plotting device	s
Dump ASCII table of X Y [Z] values	s
printer or glass dumb terminal	s
Roland DXY800A plotter	s

Dot Matrix Printers

Epson-style 60-dot per inch printers	s
Epson LX-800, Star NL-10	s
NX-1000, PROPRINTER	s
NEC printer CP6, Epson LQ-800	s
Star Color Printer	s
Tandy DMP-130 60-dot per inch	s
Vectrix 384 & Tandy color printer	s

Laser Printers

Talaris EXCL language	`set term excl`
Imagen laser printer	`set term imagen`
LN03-Plus in EGM mode	`set term ln03`
PostScript graphics language	`set term post [mode color ‘font’ size]`
CorelDraw EPS	`set term corel [mode color ‘font’ size]`
Prescribe - for the Kyocera Laser Printer	`set term prescribe`
Kyocera Laser Printer with Courier font	`set term kyo`
QMS/QUIC Laser (also Talaris 1200)	`set term qms`

Metafiles

AutoCAD DXF (120x80 default)	`set term dxf`
FIG graphics language: SunView or X	`set term fig`
FIG graphics language: Large Graph	`set term bfig`
SCO hardcopy CGI	`set term hcgi`
Frame Maker MIF 3.0	`set term mif [pentype curvetype help]`
Portable bitmap	`set term pbm [fontsize color]`
Uniplex Redwood Graphics Interface Protocol	`set term rgip`
TGIF language	`set term tgif`

HP Devices

HP2623A and maybe others	`set term hp2623A`
HP2648 and HP2647	`set term hp2648`
HP7580, & probably other HPs (4 pens)	`set term hp7580B`
HP7475 & lots of others (6 pens)	`set term hpgl`
HP Laserjet series II & clones	`set term hpljii [75 100 150 300]`
HP DeskJet 500	`set term hpdj [75 100 150 300]`
HP PaintJet & HP3630	`set term hppj [FNT5X9 FNT9X17 FNT13x25]`
HP laserjet III (HPGL plot vectors)	`set term pcl5 [mode font fontsize ]`

TeX picture environments

LaTeX picture environment	`set term latex`
EEPIC – extended LaTeX picture	`set term eepic`
LaTeX picture with emTeX specials	`set term emtex`
PSTricks macros for TeX or LaTeX	`set term pstricks`
TPIC specials for TeX or LaTeX	`set term tpic`
MetaFont font generation input	`set term mf`

Saving and restoring terminal

restore default or pushed terminal	`set term pop`
save (push) current terminal	`set term push`

Commands associated to interactive terminals

change mouse settings	`set mouse`
change hotkey bindings	`bind`

Files

plot a data file	`plot ‘fspec’`
load in a macro file	`load ‘fspec’`
save command buffer to a macro file	`save ‘fspec’`
save settings for later reuse	`save set ‘fpec’`

PLOT & SPLOT commands

plot and **splot** are the primary commands **plot**
splot plots 3-d surfaces and data.

Syntax:

plot {ranges} <function> {title}{style} {, <func

splot {ranges} <function> {title}{style} {, <fun

where <function> is either a mathematical expre
or a pair (**plot**) or triple (**splot**) of mathematical
User-defined functions and variables may also be

Plotting Data

Discrete data contained in a file can displayed by
in quotes) on the **plot** or **splot** command line. D
Lines beginning with # (or ! on VMS) will be
each data point represents an (x,y) pair. For **sp**
with error bars (see **plot errorbars**), each data
(x,y,xlow,xhigh), (x,y,xdelta,ydelta), or (x,y,xlow,
each line of a data file must be separated by blan
columns.

For **plot**s the x value may be omitted, and for **s**
either case the omitted values are assigned the cu
start at 0 and are incremented for each data poin

Surface Plotting

Implicitly, there are two types of 3-d datafiles. If a
assumed to be a grid data, i.e., the data has a grid
direction (the ith cross isoline passes thru the ith
drawn for grid data. (Note contouring is available
the same length, no cross isolines will be drawn a

Using Pipes

On some computer systems with a popen functio
piped through a shell command by starting the fi

pop(x) = 103*exp(x/10) plot "< awk '{ print $1-

would plot the same information as the first popu
x axis. Simple manipulations of this kind can also

Similarly, output can be piped to another applica

set out "|lpr -Pmy_laser_printer"

Plot Data Using

The format of data within a file can be selected with the **using** option. An explicit scanf string can be used, or simpler column choices can be made.

```
plot "datafile"                 { using {<ycol> |
                                <xcol>:<ycol> |
                                <xcol>:<ycol>:<ydelta> |
                                <xcol>:<ycol>:<width> |
                                <xcol>:<ycol>:<xdelta> |
                                <xcol>:<ycol>:<ylo>:<yhi> |
                                <xcol>:<ycol>:<xlo>:<xhi> |
                                <xcol>:<ycol>:<xdelta>:<ydelta> |
                                <xcol>:<ycol>:<ydelta>:<width> |
                                <xcol>:<ycol>:<ylo>:<yhi>:<width> |
                                <xc>:<yc>:<xlo>:<xhi>:<ylo>:<yhi>}
                                {"<scanf string>"}}...
splot "datafile"                { using {<xcol>:<ycol>:<zcol>}
                                {" <scanf string> "}}...
```

<xcol>, <ycol>, and <zcol> explicitly select the columns to plot from a space or tab separated multicolumn data file. If only <ycol> is selected for **plot**, <xcol> defaults to 1. If only <zcol> is selected for **splot**, then only that column is read from the file. An <xcol> of 0 forces <ycol> to be plotted versus its coordinate number. <xcol>, <ycol>, and <zcol> can be entered as constants or expressions. Expressions enclosed in parentheses can be used to compute a column data value from all numbers in the input record.

If errorbars (see also **plot errorbars**) are used for **plot**s, xdelta or ydelta (for example, a +/- error) should be provided as the third column, or (x,y)low and (x,y)high as third and fourth columns. These columns must follow the x and y columns. If errorbars in both directions are wanted then xdelta and ydelta should be in the third and fourth columns, respectively, or xlow, xhigh, ylow, yhigh should be in the third, fourth, fifth, and sixth columns, respectively.

Scanf strings override any <xcol>:<ycol>(:<zcol>) choices, except for ordering of input, e.g.,

```
plot "datafile"                 using 2:1 "%f%*f%f"
```

causes the first column to be y and the third column to be x.

If the scanf string is omitted, the default is generated based on the <xcol>:<ycol>(:<zcol>) choices. If the **using** option is omitted, "%f%f" is used for **plot** ("%f%f%f%f" or "%f%f%f%f%f%f" for **errorbar plot**s) and "%f%f%f" is used for **splot**.

```
plot "MyData"                   using "%*f%f%*20[^\n]%f" w lines
```

Data are read from the file "MyData" using the format "%*f%f%*20[^\n]%f". The meaning of this format is: "%*f" ignore the first number, "%f" then read in the second and assign to x, "%*20[^\n]" then ignore 20 non-newline characters, "%f" then read in the y value.

Plot With Errorbars

Error bars are supported for 2-d data file plots by r
ydelta, ylow and yhigh, xdelta, xlow and xhigh, xd
respectively. No support exists for error bars for **s**

In the default situation, gnuplot expects to see th
either (x, y, ydelta), (x, y, ylow, yhigh), (x, y, xd
or (x, y, xlow, xhigh, ylow, yhigh). The x coordina
must be exactly as given above. Data files in this

plot "data.dat" with errorbars (or yerrorbars)

plot "data.dat" with xerrorbars

plot "data.dat" with xyerrorbars

The error bar is a line plotted from (x, ylow) to (
specified instead of ylow and yhigh, ylow=y-ydelt
for xlow and xhigh are derived similarly from xde
yhigh and ylow are both set to y and xhigh and
between the data points, **plot** the data file twice,

If x or y autoscaling is on, the x or y range will b

Boxes may be drawn with y error bars using the **b**
be either set with the "set boxwidth" command,
automatically so each box touches the adjacent b
drawn for the **xyerrorbars** style by using the **bo**

x,y,ylow & yhigh from columns 1,2,3,4	p
x from third, y from second, xdelta from 6	p
x,y,xdelta & ydelta from columns 1,2,3,4	p

Plot Ranges

The optional range specifies the region of the plot

Ranges may be provided on the **plot** and **splot** co
set xrange, **set yrange**, etc., commands, to cha

[{<dummy-var>=}{<xmin>:<xmax>}]

where <dummy-var> is the independent variab
changed with **set dummy**) and the min and max

Both the min and max terms are optional. The ':'
is specified. This allows '[ ]' to be used as a null r

Specifying a range in the **plot** command line turns
one of the **set** range commands turns autoscaling
later. (See **set autoscale**).

This uses the current ranges	p
This sets the x range only	p
This sets both the x and y ranges	p
sets only y range, & turns off autoscaling on both axes	p
This sets xmax and ymin only	p
This sets the x, y, and z ranges	s

Plot With Style

Plots may be displayed in one of twelve styles: **lines**, **points**, **linespoints**, **impulses**, **dots**, **steps**, **errorbars** (or **yerrorbars**), **xerrorbars**, **xyerrorbars**, **boxes**, **boxerrorbars**, or **boxxyerrorbars**. The **lines** style connects adjacent points with lines. The **points** style displays a small symbol at each point. The **linespoints** style does both **lines** and **points**. The **impulses** style displays a vertical line from the x axis (or from the grid base for **splot**) to each point. The **dots** style plots a tiny dot at each point; this is useful for scatter plots with many points. The **steps** style is used for drawing stairstep-like functions. The **boxes** style may be used for barcharts.

The **errorbars** style is only relevant to 2-d data file plotting. It is treated like **points** for **splot**s and function **plot**s. For data **plot**s, **errorbars** is like **points**, except that a vertical error bar is also drawn: for each point (x,y), a line is drawn from (x,ylow) to (x,yhigh). A tic mark is placed at the ends of the error bar. The ylow and yhigh values are read from the data file's columns, as specified with the **using** option to plot. The **xerrorbars** style is similar except that it draws a horizontal error bar from xlow to xhigh. The **xyerrorbars** or **boxxyerrorbars** style is used for data with errors in both x and y. A barchart style may be used in conjunction with y error bars through the use of **boxerrorbars**. The See **plot errorbars** for more information.

Default styles are chosen with the **set function style** and **set data style** commands.

By default, each function and data file will use a different line type and point type, up to the maximum number of available types. All terminal drivers support at least six different point types, and re-use them, in order, if more than six are required. The LaTeX driver supplies an additional six point types (all variants of a circle), and thus will only repeat after twelve curves are plotted with points.

If desired, the style and (optionally) the line type and point type used for a curve can be specified.

with <style> {<linetype> {<pointtype>}}

where <style> is either **lines**, **points**, **linespoints**, **impulses**, **dots**, **steps**, **errorbars** (or **yerrorbars**), **xerrorbars**, **xyerrorbars**, **boxes**, **boxerrorbars**, **boxxyerrorbars**.

The <linetype> & <pointtype> are positive integer constants or expressions and specify the line type and point type to be used for the plot. Line type 1 is the first line type used by default, line type 2 is the second line type used by default, etc.

plots sin(x) with impulses	plot sin(x) with impulses
plots x*y with points, x**2 + y**2 default	splot x*y w points, x**2 + y**2
plots tan(x) with default function style	plot [ ] [-2:5] tan(x)
plots “data.1” with lines	plot "data.1" with l
plots “leastsq.dat” with impulses	plot 'leastsq.dat' w i
plots “exper.dat” with errorbars & lines connecting points	plot 'exper.dat' w l, 'exper.dat' w err

Here 'exper.dat' should have three or four data columns.

plots x**2 + y**2 and x**2 - y**2 with the same line type	splot x**2 + y**2 w l 1, x**2 - y**2 w l 1
plots sin(x) and cos(x) with linespoints, using the same line type but different point types	plot sin(x) w linesp 1 3, \ cos(x) w linesp 1 4
plots file “data” with points style 3	plot "data" with points 1 3

Note that the line style must be specified when specifying the point style, even when it is irrelevant. Here the line style is 1 and the point style is 3, and the line style is irrelevant.

See **set style** to change the default styles.

Plot Title

A title of each plot appears in the key. By default t
on the plot command line. The title can be change
precede any **with** option.

title "<title>"

where <title> is the new title of the plot and mu
shown in the key.

plots y=x with the title 'x'	p
plots the “glass.dat” file with the title 'revolution surface'	s
plots x squared with title “x^2” and “data.1” with title 'measured data'	p

Set-Show Commands

All commands below begin with either **set** or **unset**, and usually their state can be shown by passing their name to the **show** command.

unit any angles are given in	`angles [degrees|radians]`

arrows from point to	`arrow [<tag>][from <sx>,<sy>,<sz>]`
	`[to <ex>,<ey>,<ez>][head|nohead|heads]`

force autoscaling of an axis	`autoscale [<axes>]`
enter/exit parametric mode	`parametric`
display border	`border [<choice>] [<style>]`
clip points/line near boundaries	`clip <clip-type>`
specify parameters for contour plots	`cntrparam [spline][points][order][levels]`
enable splot contour plots	`contour [base|surface|both]`
default plotting style for data	`data style <style-choice>`
specify dummy variable	`dummy <dummy1>,<dummy2>...`
tic-mark label format specification	`format [<axes>]["format-string"]`
function plotting style	`function style <style-choice>`
draw a grid at tick positions	`grid [<which tics>...]  [<linestyle>]`
enables hiddenline removal	`hidden3d [...]`
specify number of isolines	`isosamples <n1>[,<n2>]`
enables key of curves in plot	`key [...]`
logscaling of axes (optionally giving base)	`logscale <axes> [<base>]`
mapping 3D coordinates	`mapping [cartesian|spherical|cylindrical]`
offsets from center of graph	`offsets <left>,<right>,<top>,<bottom>`
color-mapped plotting modes	`pm3d [...]`
mapping 2D coordinates	`polar`
set radial range	`rrange [<rmin>:<rmax>]`
set sampling rate of functions	`samples <expression>`
set scaling factors of plot	`size <xsize>,<ysize>`
control display of isolines of surface	`surface`
control graphics device	`terminal <device>`
change direction of tics	`tics <direction>`
adjust relative height of vertical axis	`ticslevel <level>`
adjust size of tick marks	`ticscale [<size>]`
turn on time/date stamp	`time`
set centered plot title	`title "title-text" <xoff>,<yoff>`
set parametric range	`trange [<tmin>:<tmax>]`
set surface parametric ranges	`urange or vrange`
sets the view point for **splot**	`view <rot_x>,<rot_z>,<scale>,<scale_z>`
sets the top view (map) for **splot**	`view map`
sets x-axis label	`xlabel "<label>" <xoff>,<yoff>`
set horizontal range	`xrange [<xmin>:<xmax>]`

change horizontal tics	`xtics <start>,<incr>,<end>,`
	`"<label>" <pos>`

adjust number of minor tick marks	`mxtics OR mytics [<freq>]`
draw x-axis	`xzeroaxis`
sets y-axis label	`ylabel "<label>" <xoff>,<yoff>`
set vertical range	`yrange [<ymin>:<ymax>]`

change vertical tics	`ytics <start>,<incr>,<end>,`
	`"<label>" <pos>`

draw y-axis	`yzeroaxis`
set default threshold for values near 0	`zero <expression>`
draw axes	`zeroaxis`
sets z-axis label	`zlabel "<label>" <xoff>,<yoff>`
set vertical range	`zrange [<zmin>:<zmax>]`

change vertical tics	`ztics <start>,<incr>,<end>,`
	`"<label>" <pos>`

draw z-axis	`zzeroaxis`

Contour Plots

Enable contour drawing for surfaces. This option

Syntax: set contour { base | surface | both } unse

If no option is provided to **set contour**, the def to draw the contours: **base** draws the contours **surface** draws the contours on the surfaces thems base and the surface.

See also **set cntrparam** for the parameters that

Contour Parameters

Sets the different parameters for the contouring p

set cntrparam	`p`
	`o`
	`l`
	`d`
	`i`

5 automatic levels	`s`
3 discrete levels at 10%, 37% and 90%	`s`
5 incremental levels at 0, .1, .2, .3 and .4	`s`
sets n = 10 retaining current setting of auto, incr., or discr.	`s`
set start = 100 and increment = 50, retaining old n	`s`

This command controls the way contours are plo pression and <z1>, <z2> any constant expressi

linear, **cubicspline**, **bspline** - Controls type of the contours are drawn piecewise linear, as extra then piecewise linear contours are interpolated to may undulate. The third option is the uniform **b** linear data but is guaranteed to be smoother.

points - Eventually all drawings are done with the number of points used to approximate a curve only.

order - Order of the bspline approximation to be resulting contour. (Of course, higher order bspline piecewise linear data.) This option is relevant for in the range from 2 (linear) to 10.

levels - Number of contour levels, 'n'. Selection 'discrete', and 'incremental'. For 'auto', if the surf will be generated from zmin+dz to zmax-dz in (levels + 1). For 'discrete', contours will be genera discrete levels is limited to MAX_DISCRETE_LEV contours are generated at <n> values of z beginni

Specifying Labels

Arbitrary labels can be placed on the plot using the **set label** command. If the z coordinate is given on a **plot** it is ignored; if it is missing on a **splot** it is assumed to be 0.

set label {<tag>}{" <label˙text> "}	`{at <x>,<y> {,<z>}}`
	`{<justification>}`

unset label {<tag>}	
show label	

The text defaults to "", and the position to 0,0,0. The <x>, <y>, and <z> values are in the graph's coordinate system. The tag is an integer that is used to identify the label. If no <tag> is given, the lowest unused tag value is assigned automatically. The tag can be used to delete or change a specific label. To change any attribute of an existing label, use the **set label** command with the appropriate tag, and specify the parts of the label to be changed.

By default, the text is placed flush left against the point x,y,z. To adjust the way the label is positioned with respect to the point x,y,z, add the parameter <justification>, which may be **left**, **right** or **center**, indicating that the point is to be at the left, right or center of the text. Labels outside the plotted boundaries are permitted but may interfere with axes labels or other text.

label at (1,2) to "y=x"	`set label "y=x" at 1,2`
label "y=xˆ2" w right of the text at (2,3,4), & tag the label number 3	`set label 3 "y=x^2" at 2,3,4 right`
change preceding label to center justification	`set label 3 center`
delete label number 2	`unset label 2`
delete all labels	`unset label`
show all labels (in tag order)	`show label`

(The EEPIC, Imagen, LaTeX, and TPIC drivers allow \\ in a string to specify a newline.)

Miscellaneous Commands

For further information on these commands, print out a copy of the gnuplot manual.

change working directory	`cd`
erase current screen or device	`clear`
exit gnuplot	`exit or quit or EOF`
display text and wait	`pause <time> ["<string>"]`
print the value of <expression>	`print <expression>`
print working directory	`pwd`
repeat last **plot** or **splot**	`replot`
spawn an interactive shell	`!  (UNIX) or $ (VMS)`

Environment Variables

A number of shell environment variables are unde but may be useful. See 'help environment' for the

If GNUTERM is defined, it is used as the name of terminal type sensed by gnuplot on start up, but is start-up file (see **start-up**), and of course by late

On Unix, OS/2, AmigaOS, and MS-DOS, GNUH HELP file (gnuplot.gih).

On VMS, the symbol GNUPLOT$HELP should gnuplot.

On Unix, HOME is used as the name of a directo in the current directory. On OS/2, AmigaOS a gnuplot.ini file. On VMS, SYS$LOGIN: is used. S

GNUPLOT_LIB may be used to define additional s files.

On Unix, PAGER is used as an output filter for h

GDFONTPATH is the directory where png termin GNUPLOT_FONTPATH is that for the postscrip

On Unix and AmigaOS, SHELL is used for the **sh** for the **shell** command.

On AmigaOS, GNUFONT is used for the screen phire/14".

On MS-DOS, if the BGI interface is used, the va the BGI drivers directory. Furthermore SVGA i 800x600 res., and its mode of operation as 'Name. C:\TC\BGI\SVGADRV.BGI and mode 3 is used and 'set SVGA=SVGADRV.3'.

GNUFITLOG holds the name of a directory or a

Expressions

In general, any mathematical expression accepte The precedence of these operators is determine language. White space (spaces and tabs) is ignore

Complex constants may be expressed as {<real> be numerical constants. For example, $\{3, 2\}$ repre curly braces are explicitly required here.

Functions

The functions in gnuplot are the same as the corresponding functions in the Unix math library, except that all functions accept integer, real, and complex arguments, unless otherwise noted. The **sgn** function is also supported, as in BASIC.

Function	Arguments	Returns
abs(x)	any	absolute value of `x`, $\|x\|$; same type
abs(x)	complex	length of `x`, $\sqrt{\text{real}(x)^2 + \text{imag}(x)^2}$
acos(x)	any	$\cos^{-1} x$ (inverse cosine) in radians
arg(x)	complex	the phase of x in radians
asin(x)	any	$\sin^{-1} x$ (inverse sin) in radians
atan(x)	any	$\tan^{-1} x$ (inverse tangent) in radians
besj0(x)	radians	j_0 Bessel function of x
besj1(x)	radians	j_1 Bessel function of x
besy0(x)	radians	y_0 Bessel function of x
besy1(x)	radians	y_1 Bessel function of x
ceil(x)	any	$\lceil x \rceil$, smallest integer not less than x (real part)
cos(x)	radians	$\cos x$, cosine of x
cosh(x)	radians	$\cosh x$, hyperbolic cosine of x
erf(x)	any	Erf(real(x)), error function of real(x)
erfc(x)	any	Erfc(real(x)), 1.0 $-$ error function of real(x)
exp(x)	any	e^x, exponential function of x
floor(x)	any	$\lfloor x \rfloor$, largest integer not greater than x (real part)
gamma(x)	any	Gamma(real(x)), gamma function of real(x)
ibeta(p,q,x)	any	Ibeta(real(p, q, x)), ibeta function of real(p,q,x)
igamma(a,x)	any	Igamma(real(a, x)), igamma function of real(a,x)
imag(x)	complex	imaginary part of x as a real number
int(x)	real	integer part of x, truncated toward zero
lgamma(x)	any	Lgamma(real(x)), lgamma function of real(x)
log(x)	any	$\log_e x$, natural logarithm (base e) of x
log10(x)	any	$\log_{10} x$, logarithm (base 10) of x
rand(x)	any	Rand(real(x)), pseudo random number generator
real(x)	any	real part of x
sgn(x)	any	1 if $x > 0$, -1 if $x < 0$, 0 if $x = 0$. imag(x) ignored
sin(x)	radians	$\sin x$, sine of x
sinh(x)	radians	$\sinh x$, hyperbolic sine x
sqrt(x)	any	$\sqrt{x}$, square root of x
tan(x)	radians	$\tan x$, tangent of x
tanh(x)	radians	$\tanh x$, hyperbolic tangent of x

Operators

The operators in gnuplot are the same as the corresponding operators in the C programming language, except that all operators accept integer, real, and complex arguments, unless otherwise noted. The ** operator (exponentiation) is supported, as in FORTRAN.

Parentheses may be used to change order of evaluation.