

A *plotxy* Reference

This document is meant to be a summary of commands and other features used in *plotxy*. For full descriptions, see the reference manual.

Commands by Topic

COLOR		LINES	
background ^c	color inside frame	weight ^c	thickness of pen
color ^c	standard color to be used next	OVERALL	
fill	fill in a polygon	frame ^c	type of frame and grid
palette ^c	custom colors	landscape ^c	reorient plot by 90°
CONTROL		plot	create and position plot
stop	end of file	POINTS	
DATA		symbol ^c	choose what to plot at each point
cancel	get rid of data series	SCALING	
save ^c	save data through plots	affine ^c	scale/translate series to plot units
HELP		logxy ^c	set log axis/axes, or equal scale
help	list commands	xlimit ^c	x-axis length, limits, and ticks
status	current series and settings	ylimit ^c	y-axis length, limits, and ticks
INPUT/OUTPUT		TEXT	
file ^c	name file to read from	character ^c	character size and angle
format ^c	use Fortran formatted read	nocomment ^c	allow use of % in text
node ^c	reading single values, pairs, triples	note ^c	place text on plot, arrows optional
output ^c	name file to write to	title ^c	title of plot
read	read data in	weight ^c	thickness of pen
skip	omit reading some lines	xlabel ^c	x-axis label
		ylabel ^c	y-axis label
LINE			
dash ^c	dash pattern for line		
smooth ^c	interpolation between values		

A ^c indicates that the command carries over from one plot to the next.

Command Summary

affine c_x x_0 c_y y_0	Scale data so $x_{pl} = c_x x + x_0$, $y_{pl} = c_y y + y_0$.
affine c_{xx} c_{xy} x_0 c_{yx} c_{yy} y_0	Scale data so $x_{pl} = c_{xx}x + c_{xy}y + x_0$, $y_{pl} = c_{yx}x + c_{yy}y + y_0$.
affine	Reset to 1 0 1 0.
background colorname	Set color to colorname inside plot frame.
background -	Reset to no color inside plot frame.
cancel [n]	Cancel the last n data series; default is 1.
character h [θ]	Set character height to h in inches and counterclockwise angle θ in °.
color [adjective] colorname	Set color by name, optionally prefixed by adjective (see table).
color n	Set color by number, usually after using palette .
dash [s_d s_k]	Set dash pattern as s_d inches drawn, s_k inches skipped.
dash 0 0	Reset to un-dashed line.
file filename	Read from file filename.
file *	Read from inside the plotxy file; read will need number of lines.
file	Go back to start of file
fill	Fill in next polygon read, which must have less than 1000 points.
format (format)	Read data using Fortran format (inside parentheses, no I formats).
format *	Read data using free-format.
format b	Read data as binary.
frame +-box	Add/remove lines around top and left of plot.
frame none	Take away frame entirely.
frame grid -grid grid-dash	Add/remove gridlines; change from light gray to dashed.
frame [+xnum] [+ymum]	Add/remove numbering on an axis; use char 0 to suppress ticks.
frame [+xaxis] [+yaxis]	Add/remove an axis.
frame [top bottom left right]	Move axis information (numbers, ticks, labels) to different side.

<code>frano [+xeven] [+yeven]</code>	Make ticks all the same height.
<code>help</code>	List all commands.
<code>landscape</code>	Rotate plot by 90°.
<code>logy mode</code>	Set axis modes. Choices are <code>loglin</code> (x-log, y linear), <code>linlog</code> , <code>loglog</code> , <code>linlogy+</code> (ignore $y \leq 0$ values), <code>logy+</code> , <code>linlin</code> , <code>equilin</code> (same scale on x and y, self-scaled to cover all data).
<code>node [-]1</code>	Read a single column of numbers; default is y-values, $\Delta x = 1$; - means x-values, $\Delta y = 1$.
<code>node 2</code>	Read two columns as x-y pairs.
<code>node [-]3</code>	Read three columns as x-y pair and error bar (or symbol size): default error in y, - for error in x.
<code>node 4</code>	Read from two files, first of x and then of y.
<code>node [-]10 c₁</code>	Read a single specified column; plot as with <code>node 1</code> .
<code>node [-]20 c₁ c₂</code>	Read two specified columns; plot as with <code>node 2</code> .
<code>node [-]30 c₁ c₂ c₃</code>	Read three specified columns; plot as with <code>node 3</code> .
<code>node [-]30 c₁ c₂ c₃ c₄</code>	Read four specified columns: x-y pair, then two error bar limits, default y, - x.
<code>nocoment</code>	Disables use of percent as comment.
<code>note (x y [in] [c r])text</code>	Put lower left corner of <code>text</code> at (x,y), in inches for <code>in</code> ; <code>r</code> is lower right corner, <code>c</code> bottom center.
<code>note (p q x y [in] [c r])text</code>	As previous, but arrow extends from (p,q) to text.
<code>note (p q x y [in] [c r] r)text</code>	Fill arrowhead.
<code>note (+)text</code>	Continue text from end of last note.
<code>note (v[c r])text</code>	Move down from last note, with optional centering or right justification.
<code>note filename</code>	Read coordinates and texts from filename.
<code>output filename</code>	Name output file filename.
<code>palette n h s b</code>	Set color number <code>n</code> to have hue <code>h</code> , saturation <code>s</code> and brightness <code>b</code> .
<code>plot</code>	Output plot at top of page, or automatically spaced down.
<code>plot [x₀ y₀ [abs]]</code>	Output plot at (x ₀ , y ₀) in inches relative to last, or in absolute location.
<code>plot [top center bottom] [left]</code>	Output plot at page location given, centered unless <code>left</code> used.
<code>plot [up down right left top]</code>	Output plot relative to previous, in direction given.
<code>read [n]</code>	Read <code>n</code> data values; all if number omitted.
<code>save</code>	Save data through a <code>plot</code> command.
<code>smooth [natural akima off]</code>	Interpolate between points linearly or using two possible splines.
<code>skip n</code>	Skip over <code>n</code> data values before reading more.
<code>status</code>	Print program settings and information about data read in.
<code>stop</code>	Stop.
<code>symbol [solid] name[h]</code>	Plot symbol named <code>name</code> , with height <code>h</code> , at each point; use <code>solid</code> to fill some symbols.
<code>symbol n [h]</code>	Plot symbol numbered <code>n</code> , with height <code>h</code> , at each point.
<code>symbol n 20</code>	Plot symbol numbered <code>n</code> , with variable height, at each point.
<code>symbol -1</code>	Reset to line plotting.
<code>title text</code>	Print text as title (above and centered).
<code>xlabel text</code>	Print text as label for x-axis.
<code>xlimit x_L [x_l x_r [δx]]</code>	Set x-axis length <code>x_L</code> , optionally limits <code>x_l</code> and <code>x_r</code> , and tick spacing <code>δx</code> ; <code>δx < 0</code> suppresses extra log ticks
<code>ylabel text</code>	Print text as label for y-axis.
<code>ylimit y_L [y_b y_t [δy]]</code>	Set y-axis length <code>y_L</code> , optionally limits <code>y_b</code> and <code>y_t</code> , and tick spacing <code>δy</code> ; <code>δy < 0</code> suppresses extra log ticks
<code>weight w [w₁]</code>	Set line thickness <code>w</code> in 0.001 st ; <code>w₁</code> sets separate thickness for lines in characters.

Mathematics in Text

Constructs are available for subscripts, superscripts, and decorations to variables. The syntax is close to, but not the same as, \TeX : `x\sup{2}` gives x^2 ; `x\sub{AB}` gives x_{AB} ; `x^` gives \hat{x} ; `x^~` gives \tilde{x} ; `x^-` gives \bar{x} .

Greek letters are accessed by surrounding the name with backslashes; for example, `\alpha`, or for uppercase characters `\DELTA`

alpha: α	beta: β	gamma: γ	delta: δ	epsilon: ϵ	zeta: ζ	eta: η
theta: θ	rgtheta: Θ	kappa: κ	lambda: λ	mu: μ	nu: ν	xi: ξ
pi: π	rho: ρ	sigma: σ	tau: τ	phi: ϕ	rgphi: Φ	chi: χ
psi: ψ	omega: ω	GAMMA: Γ	DELTA: Δ	THETA: Θ	LAMBDA: Λ	XI: Ξ
Pi: Π	SIGMA: Σ	UPSILON: Υ	PHI: Φ	PSI: Ψ	OMEGA: Ω	

There are also special codes for some special characters; `\grad` for ∇ , `\infi` for ∞ , `\time` for \times , `\part` for ∂ , `\degr` for $^\circ$, and `\sqrt` for $\sqrt{\quad}$. All of these, and more besides, can be referenced numerically (see below)

Text

Plotxy has four fonts of roman characters, accessed by surrounding the first three letters with backslashes; for example, `\sin\` gets the Simplex font, and `\ita\` the Italic.

```
Simplex: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,:;"/
Italic:  ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,:;"/
Duplex:  ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,:;"/
Complex: ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789.,:;"/
```

Symbols

The `symbol` command can use names or (for a wider range) numbers; `symbol star .1` gets a six-pointed star, and `symbol 24 .1` a five-pointed one. Putting `solid` before the name will fill the symbol.

```
0: □ (square)   1: * (triangle)  2: ◯ (octagon)  3: ◊ (diamond)  4: + (plus)     5: * (asterisk)  6: × (cross)   7: ×
8: +           9: ×          10: ×          11: ◊ (hexagon) 12: γ           13: |           14: * (star)   15: · (dot)
16: *         17: ◯ (circle) 18: ○          19: *          20: *          21: *          22: -          23: *
24: *
```

Symbols may be included in text strings by adding 2000 to their number and surrounding this number with backslashes; for example, `\2001\` in a text string would produce a triangle.

Special Characters

These special characters may be accessed by surrounding their numerical codes with backslashes; for example, `\1440\` is 1440.

```
1387: Ø      1397: ₣      1407: Ⓔ      1417: ?      1427: |      1437: ⊙      1447: €
1388: ∇      1398: ×      1408: ^      1418: %      1428: "      1438: ⚡      1448: ⚡
1389: €      1399: ÷      1409: [      1419: &      1429: °      1439: ♀      1449: *
1390: ≤      1400: ∏      1410: ]      1420: :      1430: →      1440: ⊕      1450: †
1391: ≥      1401: ·      1411: #      1421: ;      1431: ϕ      1441: ♂
1392: ∞      1402: √      1412: §      1422: !      1432: '      1442: ⚡
1393: ∫      1403: ∑      1413: †      1423: ?      1433: _      1443: ℏ
1394: ∫      1404: θ      1414: :      1424: ~      1434: ≠      1444: ⚡
1395: ∞      1405: {      1415: ;      1425: <      1435: ≈      1445: Ψ
1396: ±      1406: }      1416: !      1426: )      1436: III   1446: B
```

Colors

The table below shows all the standard colors except for `white`: one primary series, and three others that need adjectives. Note that `purple` and `magenta` are synonymous.

	red	green	blue	brown	orange	cyan	magenta	yellow	gray
pale									
light									
									
dark									

purple

Default Settings

All character strings (title and labels) are blank; font is set to `comp`.

```
character 0.11 0   dash 0 0       file *       format *     frame off    logxy linlin
mode 2           output mypost  smooth off  symbol -1    xlimit 3 0 0  ylimit 3 0 0
weight 6 6
```