

# MFPIC Quick Reference

(Copyright 2000–2004 by Daniel Luecking)

This information was prepared for version 0.8 of mfpic.

## Preamble commands

Load mfpic package ( $\LaTeX$ )	<code>\usepackage[<i>\langle options \rangle</i>]{mfpic}</code>
Options	<code>metafont   metapost, mplabels, overlaylabels, centeredcaptions, clip, truebbox, draft, final, nowrite, mfpreadlog</code>
Load mfpic; activate options (plain $\TeX$ )	<code>\input mfpic. \usemetafont   \usemetapost, \usemplabels, \overlaylabels \usecenteredcaptions, \clipmfpic, \usetruebbox, \mfpicdraft, \mfpicfinal, \mfpicnowrite, \mfpreadlog</code>
Turn off some options	<code>\nomplabels, \nooverlaylabels \nocenteredcaptions, \noclipmfpic, \notruebbox</code>
Set up/close the output file	<code>\opengraphsfile{<i>\langle base name \rangle</i>}...\closegraphsfile</code>

## The mfpic environment

Start an mfpic figure	<code>\mfpic[<i>\langle xscale \rangle</i>][<i>\langle yscale \rangle</i>]{<i>\langle x_{min} \rangle</i>}{<i>\langle x_{max} \rangle</i>}{<i>\langle y_{min} \rangle</i>}{<i>\langle y_{max} \rangle</i>}{<i>\langle mfpic commands \rangle</i>}</code> <code>\endmfpic</code>
$\LaTeX$ (optional)	<code>\begin{mfpic} \equiv \mfpic, \end{mfpic} \equiv \endmfpic</code>
The default <i>\langle xscale \rangle</i> is 1 ( <code>\mfpicunit</code> ), default <i>\langle yscale \rangle</i> is <i>\langle xscale \rangle</i> .	

## Dimensions (lengths)

<i>Purpose; where used:</i>	<i>Name and default value:</i>
Unit of length; <code>\mfpic</code>	<code>\mfpicunit, 1pt</code>
Size of a symbol; <code>\point</code> , <code>\plot</code> , and <code>\plotsymbol</code>	<code>\pointsize, 2pt</code>
Darkness of shading; <code>\shade</code>	<code>\shadespace, 1pt</code>
Space between dots; <code>\polkadot</code>	<code>\polkadotspace, 10pt</code>
Space between hatch lines; hatching macros	<code>\hatchspace, 3pt</code>
Size of arrowhead; <code>\arrow</code>	<code>\headlen, 3pt</code>
Size of x-, y-axis arrowhead; xy-axes macros	<code>\axisheadlen, 5pt</code>
Size of border axis arrowhead; side axis macros	<code>\sideheadlen, 0pt</code>
Size of marks on axes; axis marks	<code>\hashlen, 4pt</code>
Size of dashes; <code>\dashed</code>	<code>\dashlen, 4pt</code>
Space between dashes; <code>\dashed</code>	<code>\dashspace, 4pt</code>
Size of dots; <code>\dotted</code>	<code>\dotsize, 0.5pt</code>
Space between dots; <code>\dotted</code>	<code>\dotspace, 3pt</code>
Space between symbols; <code>\plot</code>	<code>\symbolspace, 5pt</code>

The following commands are used to change the size of some dimension parameters:

<i>Purpose (default):</i>	<i>Command:</i>
Set diameter of drawing pen (0.5pt)	<code>\penwd{<i>\langle dimen \rangle</i>}</code>
Set diameter of shading dots (0.5pt)	<code>\shadewd{<i>\langle dimen \rangle</i>}</code>
Set diameter of polkadot (5pt)	<code>\polkadotwd{<i>\langle dimen \rangle</i>}</code>
Set diameter of hatching pen (0.5pt)	<code>\hatchwd{<i>\langle dimen \rangle</i>}</code>
Multiply <code>\shadespace</code> by 1.2	<code>\lightershade</code>
Divide <code>\shadespace</code> by 1.2	<code>\darkershade</code>

## Colors

Set color for curves	<code>\drawcolor{<i>\langle color \rangle</i>}</code>
Set color for fills	<code>\fillcolor{<i>\langle color \rangle</i>}</code>
Set color for points, symbols	<code>\pointcolor{<i>\langle color \rangle</i>}</code>
Set color for hatching	<code>\hatchcolor{<i>\langle color \rangle</i>}</code>
Set color for arrowheads	<code>\headcolor{<i>\langle color \rangle</i>}</code>
Set color for tlabels	<code>\tlabelcolor{<i>\langle color \rangle</i>}</code>
Set color used by <code>\gcLEAR</code>	<code>\backgroundcolor{<i>\langle color \rangle</i>}</code>
$\LaTeX$ syntax	<code>\drawcolor[<i>\langle model \rangle</i>]{<i>\langle clr spec \rangle</i>}, etc.</code>
Define a color name	<code>\mpdefinecolor{<i>\langle name \rangle</i>}{<i>\langle model \rangle</i>}{<i>\langle clr spec \rangle</i>}</code>

## Common geometric figures

Drawing commands that operate on a variable length list in braces may be followed by `\datafile{\langle filename \rangle}` instead of the list.

### Points

Place a symbol at given point(s)	<code>\plotsymbol[<i>\langle size \rangle</i>]{<i>\langle name \rangle</i>}{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code>
Available symbol names	Triangle, Square, Circle, Diamond, Star, SolidTriangle, SolidSquare, SolidCircle, SolidDiamond, SolidStar, Plus, Cross, Asterisk

Points (filled or unfilled circles)	<code>\point[<i>\langle size \rangle</i>]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code>
Force filled/open circles in <code>\point</code> :	<code>\pointfilltrue/\pointfillfalse</code>

### Lines

Connect points with lines	<code>\polyline{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code> , or <code>(\lines)</code>
Closed polygon	<code>\polygon{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code>
Concatenate vectors	<code>\turtle{<i>\langle initial point \rangle</i>, <i>\langle v_1 \rangle</i>, <i>\langle v_2 \rangle</i>, ...}</code>
Rectangle (upright) with given corners	<code>\rect{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>}</code>

### Circles, arcs and ellipses

Circles	
polar form (default):	<code>\circle[p]{<i>\langle center \rangle</i>, <i>\langle radius \rangle</i>}</code>
three-point form:	<code>\circle[t]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, <i>\langle x_2, y_2 \rangle</i>}</code>
center-point form:	<code>\circle[c]{<i>\langle center \rangle</i>, <i>\langle point \rangle</i>}</code>
point-sweep form:	<code>\circle[s]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, <i>\langle angle \rangle</i>}</code>

### Arcs

polar form:	<code>\arc[p]{<i>\langle center \rangle</i>, <i>\langle \theta_1 \rangle</i>, <i>\langle \theta_2 \rangle</i>, <i>\langle radius \rangle</i>}</code>
three-point form:	<code>\arc[t]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, <i>\langle x_2, y_2 \rangle</i>}</code>
center-point-angle form:	<code>\arc[c]{<i>\langle center \rangle</i>, <i>\langle point \rangle</i>, <i>\langle angle \rangle</i>}</code>
point-sweep form (default):	<code>\arc[s]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, <i>\langle angle \rangle</i>}</code>
Ellipse, center <i>\langle x_0, y_0 \rangle</i> , radii <i>\langle r_x \rangle</i> , <i>\langle r_y \rangle</i> , angle <i>\langle \theta \rangle</i>	<code>\ellipse[<i>\langle \theta \rangle</i>]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle r_x \rangle</i>, <i>\langle r_y \rangle</i>}</code>

## General curves

A *\langle spec \rangle* can be **p** (for polyline) or **s** (for smooth) followed by a number for the tension.

Smooth curve through points	<code>\curve[<i>\langle tension \rangle</i>]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code>
Graph of $y = f(x)$	<code>\function[<i>\langle spec \rangle</i>]{<i>x_{min}</i>, <i>x_{max}</i>, <math>\Delta x</math>}{<i>f(x)</i>}</code>
Graph of parametric curve $(x(t), y(t))$	<code>\parafcn[<i>\langle spec \rangle</i>]{<i>t_{min}</i>, <i>t_{max}</i>, <math>\Delta t</math>}{<i>(x(t), y(t))</i>}</code>
Graph of $r = f(\theta)$	<code>\plrfcn[<i>\langle spec \rangle</i>]{<math>\theta_{min}</math>, <math>\theta_{max}</math>, <math>\Delta \theta</math>}{<i>f(t)</i>}</code>
Interpolate with a smooth <i>function</i>	<code>\fcncurve[<i>\langle tension \rangle</i>]{<i>\langle x_0, y_0 \rangle</i>, <i>\langle x_1, y_1 \rangle</i>, ...}</code>
Curve from data in a file	<code>\datafile[<i>\langle spec \rangle</i>]{<i>\langle file \rangle</i>}</code>
Set how <code>\datafile</code> processes a line	<code>\using{<i>\langle read-pattern \rangle</i>}{<i>\langle write-pattern \rangle</i>}</code>
Default is <code>\using{#1 #2 #3}{( #1, #2 )}</code>	

## Regions

Curves are not necessarily ‘closed’ even if the start and end are the same. The following are closed (may be filled), as are `\rect`, `\polygon`, `\circle`, and `\ellipse`.

Closed curve through given points	<code>\cyclic[⟨tension⟩]{(x<sub>1</sub>,y<sub>1</sub>),(x<sub>2</sub>,y<sub>2</sub>),...}</code>
Circular sector (pie slice)	<code>\sector{⟨center⟩,⟨radius⟩,⟨θ<sub>1</sub>⟩,⟨θ<sub>2</sub>⟩}</code>
Region between two functions	<code>\btwnfcn[⟨spec⟩]{x<sub>min</sub>,x<sub>max</sub>,Δx}{f(x)}{g(x)}</code>
Region in polar coordinates	<code>\plrregion[⟨spec⟩]{θ<sub>min</sub>,θ<sub>max</sub>,Δθ}{f(t)}</code>
Curves surrounding text	<code>\tlabelrect[⟨radius⟩](⟨x⟩,⟨y⟩){⟨text⟩}</code> <code>\tlabeloval[⟨mult⟩](⟨x⟩,⟨y⟩){⟨text⟩}</code> <code>\tlabelellipse[⟨ratio⟩](⟨x⟩,⟨y⟩){⟨text⟩}</code> <code>⟨radius⟩</code> : round corners. <code>⟨mult⟩</code> : stretch horizontally. <code>⟨ratio⟩</code> : width/height of ellipse

## Prefix macros

### Drawing curves

Dashed path	<code>\dashed[⟨length⟩,⟨gap⟩]...</code>
Dotted path	<code>\dotted[⟨size⟩,⟨gap⟩]...</code>
Trace a path with symbols	<code>\plot[⟨size⟩,⟨gap⟩]{⟨symbol⟩}...</code>
Generalized dashes	<code>\gendashed{⟨patname⟩}...</code>
Define a named dash pattern	<code>\dashpattern{⟨patname⟩}{⟨len<sub>1</sub>⟩,⟨len<sub>2</sub>⟩,...,⟨len<sub>2n</sub>⟩}</code>
Place a symbol at all nodes	<code>\plotnodes[⟨size⟩]{⟨symbol⟩}...</code>
Solid curve	<code>\draw[⟨color⟩]...</code>

### Closing a curve

These turn any path into a ‘closed’ path (result may be filled).

Close with a straight line,	<code>\lclosed...</code>
Close with a smooth join, like <code>\cycle</code> ,	<code>\sclosed...</code>
Close letting METAFONT choose	<code>\bclosed...</code>

### Filling closed curves

These filling prefixes turn off drawing of the curve.

Solid fill	<code>\gfill[⟨color⟩]...</code>
Unfill	<code>\gclear...</code>
Hatched fills	<code>\thatch[⟨space⟩,⟨angle⟩][⟨color⟩]...</code> <code>⟨angle⟩ = 45 deg</code> <code>\rhatch[⟨space⟩][⟨color⟩]...</code> <code>⟨angle⟩ = −45 deg</code> <code>\lhatch[⟨space⟩][⟨color⟩]...</code> crosshatching <code>\xhatch[⟨space⟩][⟨color⟩]...</code> <code>\hatch = \xhatch</code>
Shading	<code>\shade[⟨space⟩]...</code>
Polkadot fill	<code>\polkadot[⟨space⟩]...</code>
Fill with copies of a tile	<code>\tess{⟨tile⟩}...</code>
Define a tile*	<code>\tile{⟨name⟩,⟨unit⟩,⟨width⟩,⟨height⟩,⟨clip⟩}</code> <code>⟨drawing commands⟩ \endtile</code>

\* Creates a mini-mfpic, clipped if `⟨clip⟩ = true`.

### Storing and reusing a path

Store a path	<code>\store{⟨name⟩}...</code>
reusing a stored path	<code>\mobj{⟨name⟩}</code>

### Subpaths

Subpath by fraction of length	<code>\partpath{⟨frac1⟩,⟨frac2⟩}...</code>
Subpath by node number	<code>\subpath{⟨m⟩,⟨n⟩}...</code>
Cutting by another path	<code>\cutoffafter{⟨obj⟩}...</code> , <code>\cutoffbefore{⟨obj⟩}...</code> <code>⟨obj⟩</code> is a name created with <code>\store</code>
Trim the ends of a path	<code>\trimpath{⟨dim1⟩,⟨dim2⟩}...</code>

### Modifying a curve

Add arrowhead to the end	<code>\arrow[1⟨length⟩][r⟨angle⟩][b⟨backset⟩][c⟨color⟩]...</code>
Define arrowhead shape	<code>\headshape{⟨ratio⟩}{⟨tension⟩}{⟨filled⟩}</code>
Reverse a curve	<code>\reverse...</code>
Double arrow	<code>\arrow\reverse\arrow...</code>
Rotate around a point	<code>\rotatepath{(x<sub>0</sub>,y<sub>0</sub>),⟨angle⟩}...</code>
Reflect about a line	<code>\reflectpath{(x<sub>0</sub>,y<sub>0</sub>),(x<sub>1</sub>,y<sub>1</sub>)}...</code>
Shift	<code>\shiftpath{(dx,dy)}...</code>
Scale around a point	<code>\scalepath{(x<sub>0</sub>,y<sub>0</sub>),⟨scale⟩}...</code>
xscale about line $x = x_0$	<code>\xscalepath{x<sub>0</sub>,⟨scale⟩}...</code>
yscale about line $y = y_0$	<code>\yscalepath{y<sub>0</sub>,⟨scale⟩}...</code>
slant, pivoting on line $y = y_0$	<code>\slantpath{y<sub>0</sub>,⟨slant⟩}...</code>
yslant, pivoting on line $x = x_0$	<code>\yslantpath{x<sub>0</sub>,⟨slant⟩}...</code>
Swap x and y	<code>\xyswappath...</code>

## Axes

Draw x- and/or y-axes	<code>\axes[⟨headlen⟩], \xaxis[⟨headlen⟩], \yaxis[⟨headlen⟩]</code>
Draw various axes	<code>\axis[⟨headlen⟩]{⟨axis⟩}</code> , <code>⟨axis⟩</code> is one of x, y, l, b, r, or t.
Draw many axes	<code>\doaxes[⟨headlen⟩]{⟨list⟩}</code> , <code>⟨list⟩</code> of letters, no commas.
Shift border axis inward	<code>\axismargin{⟨axis⟩}{⟨amt⟩}</code> , <code>⟨amt⟩</code> is in graph units.
Add hashmarks to axes	<code>\axismarks{⟨axis⟩}[⟨len⟩]{c<sub>1</sub>,c<sub>2</sub>,...}</code> , <code>c<sub>j</sub></code> are positions. Abbrev. by <code>\xmarks</code> for <code>\axismarks{x}</code> , etc.
Change position of hash marks	<code>\setaxismarks{⟨axis⟩}{⟨pos⟩}</code> <code>⟨pos⟩</code> is one of inside, outside, centered, ontop, onbottom, onleft, or onright.

## Miscellaneous

Text labels	<code>\tlabel[⟨pos⟩⟨θ⟩](⟨x⟩,⟨y⟩){⟨T<sub>E</sub>X text⟩}</code> <code>\tlabels{⟨args<sub>1</sub>⟩⟨args<sub>2</sub>⟩}...</code> <code>\axislabels{⟨axis⟩}[⟨pos⟩⟨θ⟩]{⟨txt<sub>1</sub>⟩⟨n<sub>1</sub>⟩,⟨txt<sub>2</sub>⟩⟨n<sub>2</sub>⟩,...</code> <code>⟨pos⟩</code> is a two-letter sequence, <code>⟨θ⟩</code> the angle* of rotation in degrees; <code>⟨args<sub>j</sub>⟩</code> is an entire set of arguments as in <code>\tlabel</code> ; <code>⟨axis⟩</code> is a letter, <code>⟨txt<sub>j</sub>⟩</code> is label, <code>⟨n<sub>j</sub>⟩</code> is coordinate on axis
Clipping to a path	<code>\gclip...</code>
Polar conversion	<code>\plr{(r<sub>0</sub>,θ<sub>0</sub>),(r<sub>1</sub>,θ<sub>1</sub>),...}</code>
Connect paths	<code>\connect⟨path1⟩⟨path2⟩... \endconnect</code>
Draw many curves from one datafile	<code>\plotdata[⟨spec⟩]{⟨file⟩}</code> , <code>⟨spec⟩</code> is p or s⟨num⟩ where ⟨num⟩ is the (optional) tension in the smooth curve
Set how <code>\plotdata</code> draws curves <sup>†</sup>	<code>\dashedlines</code> (different dash patterns) <code>\coloredlines</code> (different colors, METAPOST only) <code>\pointedlines</code> (different symbols, like <code>\plot</code> ) <code>\datapointsonly</code> (different symbols, like <code>\plotnodes</code> )

\* The angle is optional, and ignored unless option `mplabels` is in effect.

<sup>†</sup> `\plotdata` also respects the `\using` setting (see `\datafile` in section **General curves**).