

MFPIC Quick Reference

(Copyright 2000–2009 by Daniel Luecking)

This information was prepared for version 0.99 of mfpic.

Preamble commands

Load mfpic package (L ^A T _E X)	<code>\usepackage[<i>options</i>]{mfpic}</code>
Options	<code>metafont metapost, mplabels, overlaylabels, centeredcaptions, raggedcaptions, clip, truebbox, draft, final, nowrite, mfpreadlog</code>
Load mfpic; activate options (plainT _E X)	<code>\input mfpic. \usemetafont \usemetapost, \usemplabels, \overlaylabels \usecenteredcaptions, \useraggedcaptions, \clipmfpic, \settruebbox, \mfpicdraft, \mfpicfinal, \mfpicnowrite, \mfpreadlog</code>
Turn off some options	<code>\nomplabels, \nooverlaylabels, \nocenteredcaptions, \noraggedcaptions, \noclipmfpic, \nottruebbox</code>
Set up/close the output file	<code>\opengraphsfile{<i>base name</i>}...\closegraphsfile</code>

The mfpic environment

Start an mfpic figure	<code>\mfpic[<i>xscale</i>][<i>yscale</i>]{<i>x_{min}</i>}{<i>x_{max}</i>}{<i>y_{min}</i>}{<i>y_{max}</i>}{<i>mfpic commands</i>}</code> <code>\endmfpic</code>
L ^A T _E X (optional)	<code>\begin{mfpic} ≡ \mfpic, \end{mfpic} ≡ \endmfpic</code>

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>dimen</i>}</code>
Set diameter of shading dots (0.5pt)	<code>\shadewd{<i>dimen</i>}</code>
Set diameter of polkadot (5pt)	<code>\polkadotwd{<i>dimen</i>}</code>
Set diameter of hatching pen (0.5pt)	<code>\hatchwd{<i>dimen</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>color</i>}</code>
Set color for fills	<code>\fillcolor{<i>color</i>}</code>
Set color for points, symbols	<code>\pointcolor{<i>color</i>}</code>
Set color for hatching	<code>\hatchcolor{<i>color</i>}</code>
Set color for arrowheads	<code>\headcolor{<i>color</i>}</code>
Set color for tlabels	<code>\tlabelcolor{<i>color</i>}</code>
Set color used by <code>\gc clear</code>	<code>\backgroundcolor{<i>color</i>}</code>
L ^A T _E X syntax	<code>\drawcolor[<i>model</i>]{<i>clr spec</i>}, etc.</code>
Define a color name	<code>\mfpdefinecolor{<i>name</i>}{<i>model</i>}{<i>clr spec</i>}</code>

Common geometric figures

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

Points

Place a symbol at given point(s)	<code>\plotsymbol[<i>size</i>]{<i>name</i>}{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</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>size</i>]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), ...}</code>
Force filled/open circles in <code>\point</code> :	<code>\pointfilltrue/\pointfillfalse</code>

Lines

Connect points with lines	<code>\polyline{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), ...}</code> , or <code>(\lines)</code>
Closed polygon	<code>\polygon{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), ...}</code>
Concatenate vectors	<code>\turtle{<i>initialpoint</i>}, (<i>v₁</i>), (<i>v₂</i>), ...}</code>
Rectangle (upright) with given corners	<code>\rect{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>)}</code>

Circles, arcs and ellipses

Circles	
polar form (default):	<code>\circle[p]{<i>center</i>}, (<i>radius</i>)}</code>
three-point form:	<code>\circle[t]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), (<i>x₂</i>, <i>y₂</i>)}</code>
center-point form:	<code>\circle[c]{<i>center</i>}, (<i>point</i>)}</code>
point-sweep form:	<code>\circle[s]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), (<i>angle</i>)}</code>
Arcs	
polar form:	<code>\arc[p]{<i>center</i>}, (<i>θ₁</i>), (<i>θ₂</i>), (<i>radius</i>)}</code>
three-point form:	<code>\arc[t]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), (<i>x₂</i>, <i>y₂</i>)}</code>
center-point-angle form:	<code>\arc[c]{<i>center</i>}, (<i>point</i>), (<i>angle</i>)}</code>
point-sweep form (default):	<code>\arc[s]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), (<i>angle</i>)}</code>
Ellipse, center (<i>x₀</i> , <i>y₀</i>), radii <i>r_x</i> , <i>r_y</i> , angle <i>θ</i>	<code>\ellipse[(<i>θ</i>)]{(<i>x₀</i>, <i>y₀</i>), (<i>r_x</i>), (<i>r_y</i>)}</code>

General curves

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

Smooth curve through points	<code>\curve[<i>tension</i>]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), ...}</code>
Graph of $y = f(x)$	<code>\function[<i>spec</i>]{<i>x_{min}</i>, <i>x_{max}</i>, <i>Δx</i>}{<i>f</i>(x)}</code>
Graph of parametric curve $(x(t), y(t))$	<code>\parafcn[<i>spec</i>]{<i>t_{min}</i>, <i>t_{max}</i>, <i>Δt</i>}{<i>x</i>(t), <i>y</i>(t)}</code>
Graph of $r = f(θ)$	<code>\plrfcn[<i>spec</i>]{<i>θ_{min}</i>, <i>θ_{max}</i>, <i>Δθ</i>}{<i>f</i>(t)}</code>
Interpolate with a smooth <i>function</i>	<code>\fcncurve[<i>tension</i>]{(<i>x₀</i>, <i>y₀</i>), (<i>x₁</i>, <i>y₁</i>), ...}</code>
Curve from data in a file	<code>\datafile[<i>spec</i>]{<i>file</i>}</code>
Set how <code>\datafile</code> processes a line	<code>\using{<i>read_pattern</i>}{<i>write_pattern</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 (can be filled), as are `\rect`, `\polygon`, `\circle`, and `\ellipse`.

Closed curve through given points	<code>\cyclic[$\langle tension \rangle$]{$(x_1, y_1), (x_2, y_2), \dots$}</code>
Circular sector (pie slice)	<code>\sector{$\langle center \rangle, \langle radius \rangle, \langle \theta_1 \rangle, \langle \theta_2 \rangle$}</code>
Region between two functions	<code>\btwnfcn[$\langle spec \rangle$]{$x_{\min}, x_{\max}, \Delta x$}{$f(x)$}{$g(x)$}</code>
Region in polar coordinates	<code>\plrregion[$\langle spec \rangle$]{$\theta_{\min}, \theta_{\max}, \Delta \theta$}{$f(t)$}</code>
Curves surrounding text	<code>\tlabelrect[$\langle radius \rangle$]($\langle x \rangle, \langle y \rangle$){$\langle text \rangle$}</code> <code>\tlabeloval[$\langle mult \rangle$]($\langle x \rangle, \langle y \rangle$){$\langle text \rangle$}</code> <code>\tlabelellipse[$\langle ratio \rangle$]($\langle x \rangle, \langle y \rangle$){$\langle text \rangle$}</code> $\langle radius \rangle$: round corners. $\langle mult \rangle$: stretch horizontally. $\langle ratio \rangle$: width/height of ellipse

Prefix macros

Drawing curves

Dashed path	<code>\dashed[$\langle length \rangle, \langle gap \rangle$]....</code>
Dotted path	<code>\dotted[$\langle size \rangle, \langle gap \rangle$]....</code>
Trace a path with symbols	<code>\plot[$\langle size \rangle, \langle gap \rangle$]{$\langle symbol \rangle$}....</code>
Generalized dashes	<code>\gendashed{$\langle patname \rangle$}....</code>
Define a named dash pattern	<code>\dashpattern{$\langle patname \rangle$}{$\langle len_1 \rangle, \langle len_2 \rangle, \dots, \langle len_{2n} \rangle$}</code>
Place a symbol at all nodes	<code>\plotnodes[$\langle size \rangle$]{$\langle symbol \rangle$}....</code>
Solid curve	<code>\draw[$\langle color \rangle$]....</code>

Closing a curve

These turn any path into a ‘closed’ path (result can then 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 automatic drawing of the curve.

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

* Creates a mini-mfpic, clipped if $\langle clip \rangle = \text{true}$.

Storing and reusing a path

Store a path	<code>\store{$\langle name \rangle$}....</code>
reusing a stored path	<code>\mfobj{$\langle name \rangle$}</code>

Subpaths

Subpath by fractions of length	<code>\partpath{$\langle frac1 \rangle, \langle frac2 \rangle$}....</code>
Subpath by node numbers	<code>\subpath{$\langle m \rangle, \langle n \rangle$}....</code>

Cutting by another path	<code>\cutoffafter{$\langle obj \rangle$}...., \cutoffbefore{$\langle obj \rangle$}....</code> $\langle obj \rangle$ is a name created with <code>\store</code>
Trim the ends of a path	<code>\trimpath{$\langle dim1 \rangle, \langle dim2 \rangle$}....</code>

Modifying a curve

Add arrowhead to the end	<code>\arrow[$\langle length \rangle$][$\langle angle \rangle$][$\langle b \rangle${$\langle backset \rangle$}[$\langle c \rangle${$\langle color \rangle$}....</code>
Define arrowhead shape	<code>\headshape{$\langle ratio \rangle$}{$\langle tension \rangle$}{$\langle filled \rangle$}</code>
Reverse a curve	<code>\reverse...</code>
Double arrow	<code>\arrow\reverse\arrow...</code>
Rotate around a point	<code>\rotatepath{(x_0, y_0), $\langle angle \rangle$}....</code>
Reflect about a line	<code>\reflectpath{(x_0, y_0), (x_1, y_1)}....</code>
Shift	<code>\shiftpath{(dx, dy)}....</code>
Scale around a point	<code>\scalepath{(x_0, y_0), $\langle scale \rangle$}....</code>
xscale about line $x = x_0$	<code>\xscalepath{x_0, $\langle scale \rangle$}....</code>
yscale about line $y = y_0$	<code>\yscalepath{y_0, $\langle scale \rangle$}....</code>
slant, pivoting on line $y = y_0$	<code>\slantpath{y_0, $\langle slant \rangle$}....</code>
yslant, pivoting on line $x = x_0$	<code>\yslantpath{x_0, $\langle slant \rangle$}....</code>
Swap x and y	<code>\xyswappath...</code>

Axes

Draw x- and/or y-axes	<code>\axes[$\langle headlen \rangle$], \xaxis[$\langle headlen \rangle$], \yaxis[$\langle headlen \rangle$]</code>
Draw various axes	<code>\axis[$\langle headlen \rangle$]{$\langle axis \rangle$}, $\langle axis \rangle$ is one of x, y, l, b, r, or t.</code>
Draw many axes	<code>\doaxes[$\langle headlen \rangle$]{$\langle list \rangle$}, $\langle list \rangle$ of letters, no commas.</code>
Shift border axis inward	<code>\axismargin{$\langle axis \rangle$}{$\langle amt \rangle$}, $\langle amt \rangle$ is in graph units.</code>
Add hashmarks to axes	<code>\axismarks{$\langle axis \rangle$}[$\langle len \rangle$]{c_1, c_2, \dots}, c_j are positions. Abbrev. by <code>\xmarks</code> for <code>\axismarks{x}</code>, etc.</code>
Change position of hash marks	<code>\setaxismarks{$\langle axis \rangle$}{$\langle pos \rangle$} $\langle pos \rangle$ is one of inside, outside, centered, ontop, onbottom, onleft, or onright.</code>

Miscellaneous

Text labels	<code>\tlabel[$\langle pos \rangle \langle \theta \rangle$]($\langle x \rangle, \langle y \rangle$){$\langle \text{\TeX } text \rangle$}</code> <code>\tlabels{$\langle args_1 \rangle \langle args_2 \rangle \dots$}</code> <code>\axislabels{$\langle axis \rangle$}[$\langle pos \rangle \langle \theta \rangle$]{$\langle \{ \langle txt_1 \rangle \langle n_1 \rangle, \langle \{ \langle txt_2 \rangle \langle n_2 \rangle, \dots \rangle$}</code> $\langle pos \rangle$ is a two-letter sequence, $\langle \theta \rangle$ the angle* of rotation in degrees; $\langle args_j \rangle$ is an entire set of arguments as in <code>\tlabel</code> ; $\langle axis \rangle$ is a letter, $\langle txt_j \rangle$ is label, $\langle n_j \rangle$ is coordinate on axis
Clipping to a path	<code>\gclip...</code>
Polar conversion	<code>\plr{$(r_0, \theta_0), (r_1, \theta_1), \dots$}</code>
Connect paths	<code>\connect{$\langle path1 \rangle \langle path2 \rangle \dots$} \endconnect</code>
Draw many curves from one datafile	<code>\plotdata[$\langle spec \rangle$]{$\langle file \rangle$}, $\langle spec \rangle$ is p or s$\langle num \rangle$ where $\langle num \rangle$ is the (optional) tension in the smooth curve</code>
Set how <code>\plotdata</code> draws curves [†]	<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.

[†] `\plotdata` also respects the `\using` setting (see `\datafile` in section **General curves**).