

MFPIC Quick Reference

(Copyright 2000–2011 by Daniel Luecking)

This information was prepared for version 1.06 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 (plainL ^A T _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, \notruebbox</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>}</code> <code>{<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>\gclear</code>	<code>\backgroundcolor{<i>color</i>}</code>
L ^A T _E X syntax	<code>\drawcolor[<i>model</i>]{<i>clrspec</i>}</code> , etc.
Define a color name	<code>\mfpdefinecolor{<i>name</i>}{<i>model</i>}{<i>clrspec</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, y₀)</i>, <i>(x₁, 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, y₀)</i>, <i>(x₁, y₁)</i>, ...}</code>
Force filled/open circles in <code>\point</code> :	<code>\pointfilltrue/\pointfillfalse</code>

Lines

Connect points with lines	<code>\polyline{<i>(x₀, y₀)</i>, <i>(x₁, y₁)</i>, ...}</code> , or <code>(\lines)</code>
Closed polygon	<code>\polygon{<i>(x₀, y₀)</i>, <i>(x₁, 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, y₀)</i>, <i>(x₁, 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, y₀)</i>, <i>(x₁, y₁)</i>, <i>(x₂, y₂)</i>}</code>
center-point form:	<code>\circle[c]{<i>center</i>}, <i>(point)</i>}</code>
point-sweep form:	<code>\circle[s]{<i>(x₀, y₀)</i>, <i>(x₁, 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, y₀)</i>, <i>(x₁, y₁)</i>, <i>(x₂, 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, y₀)</i>, <i>(x₁, y₁)</i>, <i>(angle)</i>}</code>
Ellipse, center (x_0, y_0) , radii $\langle r_x \rangle$, $\langle r_y \rangle$, angle $\langle \theta \rangle$	<code>\ellipse[<i>(θ)</i>]{<i>(x₀, 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, y₀)</i>, <i>(x₁, y₁)</i>, ...}</code>
Graph of $y = f(x)$	<code>\function[<i>spec</i>]{<i>x_{min}</i>, <i>x_{max}</i>, Δx}{<i>f(x)</i>}</code>
Graph of parametric curve $(x(t), y(t))$	<code>\parafcn[<i>spec</i>]{<i>t_{min}</i>, <i>t_{max}</i>, Δt}{<i>(x(t), y(t))</i>}</code>
Graph of $r = f(\theta)$	<code>\plrfcn[<i>spec</i>]{<i>θ_{min}</i>, <i>θ_{max}</i>, $\Delta \theta$}{<i>f(t)</i>}</code>
Interpolate with a smooth function	<code>\fncurve[<i>tension</i>]{<i>(x₀, y₀)</i>, <i>(x₁, 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[<i><tensor></i>]{(x₁,y₁),(x₂,y₂),...}</code>
Circular sector (pie slice)	<code>\sector{<center>,<radius>,<θ₁₂</code>
Region between two functions	<code>\btwnfcn[<spec>]{x_{min},x_{max},Δx}{f(x)}{g(x)}</code>
Region in polar coordinates	<code>\plrregion[<spec>]{θ_{min},θ_{max},Δθ}{f(t)}</code>
Curves surrounding text	<code>\labelrect[<radius>](x,y){text}</code> <code>\labeloval[<mult>](x,y){text}</code> <code>\labelellipse[<ratio>](x,y){text}</code>

<radius>: round corners. *<mult>*: stretch horizontally. *<ratio>*: 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_{122n}>}...</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 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[<color>]...</code>
Unfill	<code>\gclear...</code>
Hatched fills	<code>\thatch[<space>,<angle>][<color>]...</code>
<i><angle></i> = 45 deg	<code>\rhatch[<space>][<color>]...</code>
<i><angle></i> = -45 deg	<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>}<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>\mfobj{<name>}</code>

Subpaths

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

Cutting by another path	<code>\cutoffafter{<obj>}...,\cutoffbefore{<obj>}...<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[<length>][x<angle>][b<backset>][c<color>]...</code>
Define arrowhead shape	<code>\headshape{<ratio>}{<tensor>}{<filled>}</code>
Reverse a curve	<code>\reverse...</code>
Double arrow	<code>\arrow\reverse\arrow...</code>
Rotate around a point	<code>\rotatepath{(x₀,y₀),<angle>}...</code>
Reflect about a line	<code>\reflectpath{(x₀,y₀),(x₁,y₁)}...</code>
Shift	<code>\shiftpath{<dx>,<dy>}...</code>
Scale around a point	<code>\scalepath{(x₀,y₀),<scale>}...</code>
xscale about line <i>x</i> = <i>x</i> ₀	<code>\xscalepath{x₀,<scale>}...</code>
yscale about line <i>y</i> = <i>y</i> ₀	<code>\yscalepath{y₀,<scale>}...</code>
slant, pivoting on line <i>y</i> = <i>y</i> ₀	<code>\slantpath{y₀,<slant>}...</code>
yslant, pivoting on line <i>x</i> = <i>x</i> ₀	<code>\yslantpath{x₀,<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> <i><axis></i> is one of x, y, l, b, r, or t.
Draw many axes	<code>\doaxes[<headlen>]{<list>},</code> <i><list></i> of letters, no commas.
Shift border axis inward	<code>\axismargin{<axis>}{<amt>},</code> <i><amt></i> is in graph units.
Add hashmarks to axes	<code>\axismarks{<axis>}[<len>][c₁,c₂,...],</code> <i>c_j</i> are positions. Abbrev. by <code>\xmarks</code> for <code>\axismarks{x}</code> , etc.
Change position of hash marks	<code>\setaxismarks{<axis>}{<pos>}</code> <i><pos></i> is one of inside, outside, centered, ontop, onbottom, onleft, or onright.

Miscellaneous

Text labels	<code>\tlabel[<pos>{θ}](x,y){<i><TeX text></i>}</code> <code>\tlabels{<args_{12 <code>\axislabels{<axis>}[<pos>{θ}]{<{<txt₁>}<n₁>,<{<txt₂>}<n₂>,...}</code> <i><pos></i> is a two-letter sequence, <i><θ></i> the angle* of rotation in degrees; <i><args_j></i> is an entire set of arguments as in <code>\tlabel</code>; <i><axis></i> is a letter, <i><txt_j></i> is label, <i><n_j></i> is coordinate on axis}</code>
Clipping to a path	<code>\gclip...</code>
Polar conversion	<code>\plr{(r₀,θ₀),(r₁,θ₁),...}</code>
Connect paths	<code>\connect{<path1>}<path2>... \endconnect</code>
Draw many curves from one datafile	<code>\plotdata[<spec>]{<file>},</code> <i><spec></i> is p or s <i><num></i> where <i><num></i> is the (optional) tension in the smooth curve
Set how <code>\plotdata</code> draws curves [†]	<code>\dashedlines</code> (different dash patterns) <code>\coloredlines</code> (different colors, METAFONT only) <code>\pointedlines</code> (different symbols, like <code>\plot</code>) <code>\datapointonly</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**).