

MFPIC Quick Reference

(Copyright 2000–2012 by Daniel Luecking)

This information was prepared for version 1.08 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[⟨<i>tension</i>⟩]{(x₁,y₁),(x₂,y₂),...}</code>
Circular sector (pie slice)	<code>\sector{⟨<i>center</i>⟩,⟨<i>radius</i>⟩,⟨<i>θ</i>₁⟩,⟨<i>θ</i>₂⟩}</code>
Region between two functions	<code>\btwnfcn[⟨<i>spec</i>⟩]{x_{min},x_{max},Δ<i>x</i>}{<i>f</i>(<i>x</i>)}{<i>g</i>(<i>x</i>)}</code>
Region in polar coordinates	<code>\plrregion[⟨<i>spec</i>⟩]{θ_{min},θ_{max},Δ<i>θ</i>}{<i>f</i>(<i>t</i>)}</code>
Curves surrounding text	<code>\tlabelrect[⟨<i>radius</i>⟩](⟨<i>x</i>⟩,⟨<i>y</i>⟩){⟨<i>text</i>⟩}</code> <code>\tlabeloval[⟨<i>mult</i>⟩](⟨<i>x</i>⟩,⟨<i>y</i>⟩){⟨<i>text</i>⟩}</code> <code>\tlabelellipse[⟨<i>ratio</i>⟩](⟨<i>x</i>⟩,⟨<i>y</i>⟩){⟨<i>text</i>⟩}</code> <i>⟨radius⟩</i> : round corners. <i>⟨mult⟩</i> : stretch horizontally. <i>⟨ratio⟩</i> : width/height of ellipse

Prefix macros

Drawing curves

Dashed path	<code>\dashed[⟨<i>length</i>⟩,⟨<i>gap</i>⟩]...</code>
Dotted path	<code>\dotted[⟨<i>size</i>⟩,⟨<i>gap</i>⟩]...</code>
Trace a path with symbols	<code>\plot[⟨<i>size</i>⟩,⟨<i>gap</i>⟩]{⟨<i>symbol</i>⟩}...</code>
Generalized dashes	<code>\gendashed{⟨<i>patname</i>⟩}...</code>
Define a named dash pattern	<code>\dashpattern{⟨<i>patname</i>⟩}{⟨<i>len</i>₁⟩,⟨<i>len</i>₂⟩,...,⟨<i>len</i>_{2<i>n</i>⟩⟩}}</code>
Place a symbol at all nodes	<code>\plotnodes[⟨<i>size</i>⟩]{⟨<i>symbol</i>⟩}...</code>
Solid curve	<code>\draw[⟨<i>color</i>⟩]...</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[⟨<i>color</i>⟩]...</code>
Unfill	<code>\gclear...</code>
Hatched fills	<code>\thatch[⟨<i>space</i>⟩,⟨<i>angle</i>⟩][⟨<i>color</i>⟩]...</code>
<i>⟨angle⟩</i> = 45 deg	<code>\rhatch[⟨<i>space</i>⟩][⟨<i>color</i>⟩]...</code>
<i>⟨angle⟩</i> = −45 deg	<code>\lhatch[⟨<i>space</i>⟩][⟨<i>color</i>⟩]...</code>
crosshatching	<code>\xhatch[⟨<i>space</i>⟩][⟨<i>color</i>⟩]...</code>
	<code>\hatch = \xhatch</code>
Shading	<code>\shade[⟨<i>space</i>⟩]...</code>
Gradients [†]	<code>\gradient{⟨<i>clr</i>⟩,⟨<i>width</i>⟩,⟨<i>angle</i>⟩}...</code> <code>\areagradient{⟨<i>clr</i>⟩,⟨<i>h-wd</i>⟩,⟨<i>v-wd</i>⟩}...</code> <code>\radialgradient{⟨<i>clr</i>⟩,⟨<i>wd</i>⟩,⟨<i>center</i>⟩}...</code>
Polkadot fill	<code>\polkadot[⟨<i>space</i>⟩]...</code>
Fill with copies of a tile	<code>\tess{⟨<i>tile</i>⟩}...</code>
Define a tile*	<code>\tile{⟨<i>name</i>⟩,⟨<i>unit</i>⟩,⟨<i>width</i>⟩,⟨<i>height</i>⟩,⟨<i>clip</i>⟩}</code> <i>⟨drawing commands⟩</i> <code>\endtile</code>

[†] *⟨clr⟩* is a function that returns a color for parameter(s) in (0,1).

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

Storing and reusing a path

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

Subpaths

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

Modifying a curve

Add arrowhead to the end	<code>\arrow[1⟨<i>length</i>⟩][r⟨<i>angle</i>⟩][b⟨<i>backset</i>⟩][c⟨<i>color</i>⟩]...</code>
Define arrowhead shape	<code>\headshape{⟨<i>ratio</i>⟩}{⟨<i>tension</i>⟩}{⟨<i>filled</i>⟩}</code>
Reverse a curve	<code>\reverse...</code>
Double arrow	<code>\arrow\reverse\arrow...</code>
Rotate around a point	<code>\rotatepath{(x₀,y₀),⟨<i>angle</i>⟩}...</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),⟨<i>scale</i>⟩}...</code>
xscale about line <i>x</i> = <i>x</i> ₀	<code>\xscalepath{x₀,⟨<i>scale</i>⟩}...</code>
yscale about line <i>y</i> = <i>y</i> ₀	<code>\yscalepath{y₀,⟨<i>scale</i>⟩}...</code>
slant, pivoting on line <i>y</i> = <i>y</i> ₀	<code>\slantpath{y₀,⟨<i>slant</i>⟩}...</code>
yslant, pivoting on line <i>x</i> = <i>x</i> ₀	<code>\yslantpath{x₀,⟨<i>slant</i>⟩}...</code>
Swap x and y	<code>\xyswappath...</code>

Axes

Draw x- and/or y-axes	<code>\axes[⟨<i>headlen</i>⟩], \xaxis[⟨<i>headlen</i>⟩], \yaxis[⟨<i>headlen</i>⟩]</code>
Draw various axes	<code>\axis[⟨<i>headlen</i>⟩]{⟨<i>axis</i>⟩}</code> , <i>⟨axis⟩</i> is one of x , y , l , b , r , or t .
Draw many axes	<code>\doaxes[⟨<i>headlen</i>⟩]{⟨<i>list</i>⟩}</code> , <i>⟨list⟩</i> of letters, no commas.
Shift border axis inward	<code>\axismargin{⟨<i>axis</i>⟩}{⟨<i>amt</i>⟩}</code> , <i>⟨amt⟩</i> is in graph units.
Add hashmarks to axes	<code>\axismarks{⟨<i>axis</i>⟩}[⟨<i>len</i>⟩]{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{⟨<i>axis</i>⟩}{⟨<i>pos</i>⟩}</code> <i>⟨pos⟩</i> is one of inside , outside , centered , ontop , onbottom , onleft , or onright .

Miscellaneous

Text labels	<code>\tlabel[⟨<i>pos</i>⟩⟨<i>θ</i>⟩](⟨<i>x</i>⟩,⟨<i>y</i>⟩){⟨<i>T_EX text</i>⟩}</code> <code>\tlabels{⟨<i>args</i>₁⟩⟨<i>args</i>₂⟩}...</code> <code>\axislabels{⟨<i>axis</i>⟩}[⟨<i>pos</i>⟩⟨<i>θ</i>⟩]{⟨<i>txt</i>₁⟩⟨<i>n</i>₁⟩,⟨<i>txt</i>₂⟩⟨<i>n</i>₂⟩,...}</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
Clipping to a path	<code>\gclip...</code>
Polar conversion	<code>\plr{(r₀,θ₀),(r₁,θ₁),...}</code>
Connect paths	<code>\connect{⟨<i>path1</i>⟩⟨<i>path2</i>⟩}... \endconnect</code>
Draw many curves from one datafile	<code>\plotdata[⟨<i>spec</i>⟩]{⟨<i>file</i>⟩}</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, 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**).