MFPIC Quick Reference

This information was prepared for version 1.10 of mfpic.

Preamble commands

Load mfpic package (LaTeX)	`\usepackage[`⟨*options*⟩`]{mfpic}`
Options	`metafont` | `metapost`, `mplabels`, `overlaylabels`, `centeredcaptions`, `raggedcaptions`, `clip`, `truebbox`, `draft`, `final`, `nowrite`, `mfpreadlog`
Load mfpic; activate options (plainTeX)	`\input mfpic. \usemetafont` | `\usemetapost`, `\usemplabels`, `\overlaylabels` `\usecenteredcaptions`, `\useraggedcaptions`, `\clipmfpic`, `\usetruebbox`, `\mfpicdraft`, `\mfpicfinal`, `\mfpicnowrite`, `\mfpreadlog`
Turn off some options	`\nomplabels`, `\nooverlaylabels`, `\nocenteredcaptions`, `\noraggedcaptions`, `\noclipmfpic`, `\notruebbox`
Set up/close the output file	`\opengraphsfile{`⟨*base name*⟩`}`...`\closegraphsfile`

The mfpic environment

Start an mfpic figure	`\mfpic[`⟨*xscale*⟩`][`⟨*yscale*⟩`]{`⟨$x_{\min}$⟩`}{`⟨x_{max}⟩`}{`⟨y_{min}⟩`}{`⟨y_{max}⟩`}` ⟨*mfpic commands*⟩ `\endmfpic`
LaTeX (optional)	`\begin{mfpic}` ≡ `\mfpic`, `\end{mfpic}` ≡ `\endmfpic`

Dimensions (lengths)

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

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

Purpose (*default*):	*Command:*
Set diameter of drawing pen (0.5pt)	`\penwd{`⟨*dimen*⟩`}`
Set diameter of shading dots (0.5pt)	`\shadewd{`⟨*dimen*⟩`}`
Set diameter of polkadot (5pt)	`\polkadotwd{`⟨*dimen*⟩`}`
Set diameter of hatching pen (0.5pt)	`\hatchwd{`⟨*dimen*⟩`}`
Multiply `\shadespace` by 1.2	`\lightershade`
Divide `\shadespace` by 1.2	`\darkershade`

Colors

Set color for curves	`\drawcolor{`⟨*color*⟩`}`
Set color for fills	`\fillcolor{`⟨*color*⟩`}`
Set color for points, symbols	`\pointcolor{`⟨*color*⟩`}`
Set color for hatching	`\hatchcolor{`⟨*color*⟩`}`
Set color for arrowheads	`\headcolor{`⟨*color*⟩`}`
Set color for tlabels	`\tlabelcolor{`⟨*color*⟩`}`
Set color used by `\gclear`	`\backgroundcolor{`⟨*color*⟩`}`
LaTeX syntax	`\drawcolor[`⟨*model*⟩`]{`⟨*clrspec*⟩`}`, etc.
Define a color name	`\mfpdefinecolor{`⟨*name*⟩`}{`⟨*model*⟩`}{`⟨*clrspec*⟩`}`

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)	`\plotsymbol[`⟨*size*⟩`]{`⟨*name*⟩`}{`(x_0,y_0)`,`(x_1,y_1)`,...}`
Available symbol names	`Triangle`, `Square`, `Circle`, `Diamond`, `Star`, `SolidTriangle`, `SolidSquare`, `SolidCircle`, `SolidDiamond`, `SolidStar`, `Plus`, `Cross`, `Asterisk`
Points (filled or unfilled circles)	`\point[`⟨*size*⟩`]{`(x_0,y_0)`,`(x_1,y_1)`,...}`
Force filled/open circles in `\point`:	`\pointfilltrue`/`\pointfillfalse`

Lines

Connect points with lines	`\polyline{`(x_0,y_0)`,`(x_1,y_1)`,...}`, or (`\lines`)
Closed polygon	`\polygon{`(x_0,y_0)`,`(x_1,y_1)`,...}`
Concatenate vectors	`\turtle{`⟨*initialpoint*⟩`,`⟨v_1⟩`,`⟨v_2⟩`,...}`
Rectangle (upright) with given corners	`\rect{`(x_0,y_0)`,`(x_1,y_1)`}`

Circles, arcs and ellipses

Circles	
polar form (default):	`\circle[p]{`⟨*center*⟩`,`⟨*radius*⟩`}`
three-point form:	`\circle[t]{`(x_0,y_0)`,`(x_1,y_1)`,`(x_2,y_2)`}`
center-point form:	`\circle[c]{`⟨*center*⟩`,`⟨*point*⟩`}`
point-sweep form:	`\circle[s]{`(x_0,y_0)`,`(x_1,y_1)`,`⟨*angle*⟩`}`
Arcs	
polar form:	`\arc[p]{`⟨*center*⟩`,`⟨θ_1⟩`,`⟨θ_2⟩`,`⟨*radius*⟩`}`
three-point form:	`\arc[t]{`(x_0,y_0)`,`(x_1,y_1)`,`(x_2,y_2)`}`
center-point-angle form:	`\arc[c]{`⟨*center*⟩`,`⟨*point*⟩`,`⟨*angle*⟩`}`
point-sweep form (default):	`\arc[s]{`(x_0,y_0)`,`(x_1,y_1)`,`⟨*angle*⟩`}`
Ellipse, center (x_0,y_0), radii ⟨r_x⟩, ⟨r_y⟩, angle ⟨θ⟩	`\ellipse[`⟨θ⟩`]{`(x_0,y_0)`,`⟨r_x⟩`,`⟨r_y⟩`}`

General curves

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

Smooth curve through points	`\curve[`⟨*tension*⟩`]{`(x_0,y_0)`,`(x_1,y_1)`,...}`
Graph of $y=f(x)$	`\function[`⟨*spec*⟩`]{`$x_{\min}$`,`$x_{\max}$`,`Δx`}{`f`(x)}`
Graph of parametric curve $(x(t),y(t))$	`\parafcn[`⟨*spec*⟩`]{`$t_{\min}$`,`$t_{\max}$`,`Δt`}{(x(t), y(t))}`
Graph of $r=f(\theta)$	`\plrfcn[`⟨*spec*⟩`]{`$\theta_{\min}$`,`$\theta_{\max}$`,`$\Delta\theta$`}{`f`(t)}`
Interpolate with a smooth *function*	`\fcncurve[`⟨*tension*⟩`]{`(x_0,y_0)`,`(x_1,y_1)`,...}`
Curve from data in a file	`\datafile[`⟨*spec*⟩`]{`⟨*file*⟩`}`
Set how `\datafile` processes a line	`\using{`⟨*read_pattern*⟩`}{`⟨*write_pattern*⟩`}`
	Default is `\using{#1 #2 #3}{(#1,#2)}`

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	`\cyclic[`⟨*tension*⟩`]{`$(x_1,y_1),(x_2,y_2),\ldots$`}`
Circular sector (pie slice)	`\sector{`⟨*center*⟩`,`⟨*radius*⟩`,`⟨θ_1⟩`,`⟨θ_2⟩`}`
Region between two functions	`\btwnfcn[`⟨*spec*⟩`]{`$x_{\min}$`,`$x_{\max}$`,`Δx`}{`$f(\mathtt{x})$`}{`$g(\mathtt{x})$`}`
Region in polar coordinates	`\plrregion[`⟨*spec*⟩`]{`$\theta_{\min}$`,`$\theta_{\max}$`,`$\Delta\theta$`}{`$f(\mathtt{t})$`}`
Curves surrounding text	`\tlabelrect[`⟨*radius*⟩`](`⟨*x*⟩`,`⟨*y*⟩`){`⟨*text*⟩`}`
	`\tlabeloval[`⟨*mult*⟩`](`⟨*x*⟩`,`⟨*y*⟩`){`⟨*text*⟩`}`
	`\tlabelellipse[`⟨*ratio*⟩`](`⟨*x*⟩`,`⟨*y*⟩`){`⟨*text*⟩`}`
	⟨*radius*⟩: round corners. ⟨*mult*⟩: stretch horizontally. ⟨*ratio*⟩: width/height of ellipse

Prefix macros

Drawing curves

Dashed path	`\dashed[`⟨*length*⟩`,`⟨*gap*⟩`]...`
Dotted path	`\dotted[`⟨*size*⟩`,`⟨*gap*⟩`]...`
Trace a path with symbols	`\plot[`⟨*size*⟩`,`⟨*gap*⟩`]{`⟨*symbol*⟩`}...`
Generalized dashes	`\gendashed{`⟨*patname*⟩`}...`
Define a named dash pattern	`\dashpattern{`⟨*patname*⟩`}{`⟨len_1⟩`,`⟨len_2⟩`,...,`⟨len_{2n}⟩`}`
Place a symbol at all nodes	`\plotnodes[`⟨*size*⟩`]{`⟨*symbol*⟩`}...`
Solid curve	`\draw[`⟨*color*⟩`]...`

Closing a curve

These turn any path into a 'closed' path (result can then be filled).

Close with a straight line,	`\lclosed...`
Close with a smooth join, like `\cycle`,	`\sclosed...`
Close letting METAFONT choose	`\bclosed...`

Filling closed curves

These filling prefixes turn off automatic drawing of the curve.

Solid fill	`\gfill[`⟨*color*⟩`]...`
Unfill	`\gclear...`
Hatched fills	`\thatch[`⟨*space*⟩`,`⟨*angle*⟩`][`⟨*color*⟩`]...`
⟨*angle*⟩ = 45 deg	`\rhatch[`⟨*space*⟩`][`⟨*color*⟩`]...`
⟨*angle*⟩ = −45 deg	`\lhatch[`⟨*space*⟩`][`⟨*color*⟩`]...`
crosshatching	`\xhatch[`⟨*space*⟩`][`⟨*color*⟩`]...`
	`\hatch` = `\xhatch`
Shading	`\shade[`⟨*space*⟩`]...`
Gradients†	`\gradient{`⟨*clr*⟩`,`⟨*width*⟩`,`⟨*angle*⟩`}...`
	`\areagradient{`⟨*clr*⟩`,`⟨*h-wd*⟩`,`⟨*v-wd*⟩`}...`
	`\radialgradient{`⟨*clr*⟩`,`⟨*wd*⟩`,`⟨*center*⟩`}...`
Polkadot fill	`\polkadot[`⟨*space*⟩`]...`
Fill with copies of a tile	`\tess{`⟨*tile*⟩`}...`
Define a tile*	`\tile{`⟨*name*⟩`,`⟨*unit*⟩`,`⟨*width*⟩`,`⟨*height*⟩`,`⟨*clip*⟩`}`
	⟨*drawing commands*⟩ `\endtile`

† ⟨*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	`\store{`⟨*name*⟩`}...`
reusing a stored path	`\mfobj{`⟨*name*⟩`}`

Subpaths

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

Modifying a curve

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

Axes

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

Miscellaneous

Text labels	`\tlabel[`⟨*pos*⟩⟨θ⟩`](`⟨*x*⟩`,`⟨*y*⟩`){`⟨TEX *text*⟩`}`
	`\tlabels{`⟨$args_1$⟩⟨$args_2$⟩`...}`
	`\axislabels{`⟨*axis*⟩`}[`⟨*pos*⟩⟨θ⟩`]{{`⟨txt_1⟩`}`⟨n_1⟩`,{`⟨txt_2⟩`}`⟨n_2⟩`,...}`
	⟨*pos*⟩ is a two-letter sequence, ⟨θ⟩ the angle* of rotation in degrees; ⟨$args_j$⟩ is an entire set of arguments as in `\tlabel`; ⟨*axis*⟩ is a letter, ⟨txt_j⟩ is label, ⟨n_j⟩ is coordinate on axis
Clipping to a path	`\gclip...`
Polar conversion	`\plr{(`r_0`,`θ_0`),(`r_1`,`θ_1`),...}`
Connect paths	`\connect` ⟨*path1*⟩ ⟨*path2*⟩ ... `\endconnect`
Draw many curves from one datafile	`\plotdata[`⟨*spec*⟩`]{`⟨*file*⟩`}`, ⟨*spec*⟩ is `p` or `s`⟨*num*⟩ where ⟨*num*⟩ is the (optional) tension in the smooth curve
Set how `\plotdata` draws curves†	`\dashedlines` (different dash patterns)
	`\coloredlines` (different colors, METAPOST only)
	`\pointedlines` (different symbols, like `\plot`)
	`\datapointsonly` (different symbols, like `\plotnodes`)

* The angle is optional, and ignored unless option `mplabels` is in effect.
† `\plotdata` also respects the `\using` setting (see `\datafile` in section **General curves**).