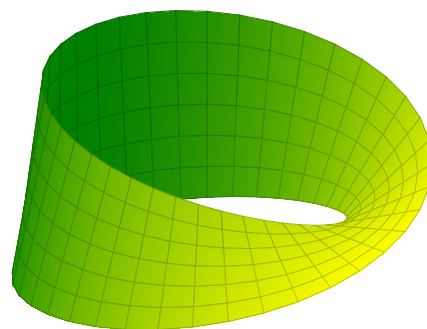


# Quick Reference

## PGFPLOTS

```
\begin{tikzpicture}
\begin{axis}[
  hide axis,
  view = {40}{40},
]
\addplot3[
  surf,
  colormap/greenyellow,
  shader = faceted interp,
  z buffer = sort,
  point meta = x,
  domain = 0:360,
  domain y = -0.5:0.5,
  samples = 40,
  samples y = 7,
]
({(1 + 0.5 * y * cos(x / 2)) * cos(x)},
{(1 + 0.5 * y * cos(x / 2)) * sin(x)},
{0.5 * y * sin(x/2)});
\end{axis}
\end{tikzpicture}
```



Contents

|                                |          |                            |          |
|--------------------------------|----------|----------------------------|----------|
| <b>1 General</b>               | <b>1</b> | <b>3.2 Line Plots</b>      | <b>3</b> |
| 1.1 Document Structure         | 1        | 3.3 Bar Plots              | 4        |
| 1.2 PGFPLOTS Options           | 1        | 3.4 Comb Plots             | 5        |
| 1.3 Key Handlers               | 1        | 3.5 Quiver Plots           | 5        |
| 1.4 Mathematical Expressions   | 1        |                            |          |
| <b>2 Axis Environments</b>     | <b>1</b> | <b>4 Lines and Markers</b> | <b>5</b> |
| <b>3 Plots</b>                 | <b>2</b> | 4.1 Line Width             | 5        |
| 3.1 Input Data                 | 2        | 4.2 Line Cap               | 6        |
| 3.1.1 Coordinates List         | 2        | 4.3 Line Join              | 6        |
| 3.1.2 Table Data               | 2        | 4.4 Dash Pattern           | 6        |
| 3.1.3 Mathematical Expressions | 3        | 4.5 Markers                | 6        |
|                                |          | 4.6 Colors                 | 7        |

Nomenclature

|  |   |
|--|---|
| <code>\foo</code>                          | T <sub>E</sub> X control sequence.  |
| <code>foo<sub>env</sub></code>             | L <sup>A</sup> T <sub>E</sub> X environment <code>foo</code> .  |
| <code>foo<sub>sty</sub></code>             | PGFPLOTS style with key <code>foo</code> .  |
| <code>foo</code>                           | Terminal symbol, literal text.  |
| <code>&lt;foo&gt;</code>                   | Non-terminal symbol, metasyntactic variable.  |
| <code>&lt;foo&gt; → &lt;bar&gt;</code>     | Production rule; <code>&lt;foo&gt;</code> can be replaced by <code>&lt;bar&gt;</code> ,<br><code>&lt;foo&gt;</code> and <code>&lt;bar&gt;</code> are implicit groups. |
| <code>&lt;foo&gt; &lt;bar&gt;</code>       | Sequence; <code>&lt;foo&gt;</code> followed by <code>&lt;bar&gt;</code> .   |
| <code>&lt;foo&gt;   &lt;bar&gt;</code>     | Choice; <code>&lt;foo&gt;</code> or <code>&lt;bar&gt;</code> .  |
| <code>&lt;foo&gt;*</code>                  | <code>&lt;foo&gt;</code> can occur zero or more times.  |
| <code>&lt;foo&gt;+</code>                  | <code>&lt;foo&gt;</code> can occur one or more times.   |
| <code>&lt;foo&gt;?</code>                  | <code>&lt;foo&gt;</code> is optional.   |
| <code>()</code>                            | Explicit group.   |
| <code>▷ &lt;key&gt; = &lt;value&gt;</code> | User option, <code>&lt;key&gt;</code> and <code>&lt;value&gt;</code> are implicit<br>groups.  |
| <code>▷ &lt;key&gt;</code>                 | User option without a value.  |
| <code>42</code>                            | Default value is 42.  |
| <code>␣</code>                             | Line continuation mark.   |
| <code>&lt;empty&gt;</code>                 | Nothing.  |
| <code>&lt;newline&gt;</code>               | Newline character, <code>^^M</code> in T <sub>E</sub> X.  |
| <code>&lt;dimension&gt;</code>             | A legitimate T <sub>E</sub> X dimension.  |
| <code>&lt;number&gt;</code>                | $(-\infty, \infty) \cap \mathbb{R}$ .   |
| <code>&lt;positive number&gt;</code>       | $(0, \infty) \cap \mathbb{R}$ .   |
| <code>&lt;non-negative number&gt;</code>   | $[0, \infty) \cap \mathbb{R}$ .   |
| <code>&lt;integer&gt;</code>               | $(-\infty, \infty) \cap \mathbb{Z}$ .   |
| <code>&lt;positive integer&gt;</code>      | $(0, \infty) \cap \mathbb{Z}$ .   |
| <code>&lt;non-negative integer&gt;</code>  | $[0, \infty) \cap \mathbb{Z}$ .   |

|                                       |   |   |                 |
|---------------------------------------|---|---|-----------------|
| <code>table data</code>               | 2 | <b>N</b>                                |                 |
| <b>J</b>                              |   | <code>none</code>                       | 2               |
| <code>jump</code>                     | 2 | <code>normal</code>                     | 2               |
| <b>K</b>                              |   | <b>P</b>                                |                 |
| key handler                           |   | <code>\pgfplotbarshift</code>           | 4               |
| <code>.append style</code>            | 1 | <code>\pgfplotbarwidth</code>           | 4               |
| <code>.cd</code>                      | 1 | <code>\pgfplotsset</code>               | 1               |
| <code>.code 2 args</code>             | 1 | <code>\pgfplotstableread</code>         | 3               |
| <code>.code</code>                    | 1 |   |                 |
| <code>.style</code>                   | 1 | <b>S</b>                                |                 |
| <b>L</b>                              |   | <code>scanline</code>                   | 2               |
| <code>line style</code>               | 6 | <code>semilogxaxis<sub>env</sub></code> | 1               |
| <code>line width</code>               | 5 | <code>semilogyaxis<sub>env</sub></code> | 1               |
| <code>linear</code>                   |   | sequence of coordinates                 |                 |
| <code>axis scaling</code>             | 2 | <code>input data</code>                 | 2               |
| <code>linear</code>                   | 2 | <code>.style key handler</code>         | 1               |
| <code>\lineno</code>                  | 2 | <code>style option</code>               | see key handler |
| list of coordinates                   |   | <b>T</b>                                |                 |
| <code>input data</code>               | 2 | <code>table</code>                      | 2               |
| <code>log</code>                      | 2 | <code>table data</code>                 |                 |
| logarithmic                           |   | <code>input data</code>                 | 2               |
| <code>axis scaling</code>             | 2 | <code>\thisrow</code>                   | 2               |
| <code>loglogaxis<sub>env</sub></code> | 1 | <code>\thisrowno</code>                 | 2               |

## Option Index

|   |   |   |   |
|---|---|---|---|
| <b>B</b>                                    |   | u value . . . . .                                       | 5 |
| bar cycle list <sub>sty</sub> . . . . .     | 4 | update limits . . . . .                                 | 5 |
| bar direction . . . . .                     | 4 | v . . . . .   | 5 |
| bar shift . . . . .                         | 4 | v value . . . . .                                       | 5 |
| bar shift auto <sub>sty</sub> . . . . .     | 4 | w . . . . .   | 5 |
| bar width . . . . .                         | 4 | w value . . . . .                                       | 5 |
| <b>C</b>                                    |   | <b>S</b>  |   |
| color . . . . .                             | 7 | samples . . . . .                                       | 3 |
| const plot . . . . .                        | 3 | samples at . . . . .                                    | 3 |
| const plot mark left . . . . .              | 3 | samples y . . . . .                                     | 3 |
| const plot mark mid . . . . .               | 3 | semithick <sub>sty</sub> . . . . .                      | 5 |
| const plot mark right . . . . .             | 3 | sharp plot . . . . .                                    | 3 |
| <b>D</b>                                    |   | smooth . . . . .  | 3 |
| dash pattern . . . . .                      | 6 | solid <sub>sty</sub> . . . . .                          | 6 |
| dash phase . . . . .                        | 6 | <b>T</b>  |   |
| dashdotted <sub>sty</sub> . . . . .         | 6 | table/  |   |
| dashed <sub>sty</sub> . . . . .             | 6 | col sep . . . . .                                       | 3 |
| densely dashdotted <sub>sty</sub> . . . . . | 6 | comment chars . . . . .                                 | 2 |
| densely dashdotted <sub>sty</sub> . . . . . | 6 | header . . . . .  | 2 |
| densely dashed <sub>sty</sub> . . . . .     | 6 | ignore chars . . . . .                                  | 2 |
| densely dotted <sub>sty</sub> . . . . .     | 6 | meta . . . . .  | 2 |
| domain . . . . .                            | 3 | read completely . . . . .                               | 3 |
| domain y . . . . .                          | 3 | row sep . . . . .                                       | 3 |
| dotted <sub>sty</sub> . . . . .             | 6 | search path . . . . .                                   | 3 |
| draw . . . . .                              | 7 | search path/  |   |
| <b>E</b>                                    |   | implicit . . . . .                                      | 3 |
| empty line . . . . .                        | 2 | skip first n . . . . .                                  | 2 |
| every axis plot . . . . .                   | 2 | white space chars . . . . .                             | 2 |
| every linear axis . . . . .                 | 2 | x . . . . .   | 2 |
| every loglog axis . . . . .                 | 2 | x error . . . . .                                       | 2 |
| every mark <sub>sty</sub> . . . . .         | 7 | x error minus . . . . .                                 | 2 |
| every semilogx axis . . . . .               | 2 | x error plus . . . . .                                  | 2 |
| every semilogy axis . . . . .               | 2 | y . . . . .   | 2 |
| <b>F</b>                                    |   | y error . . . . .                                       | 2 |
| fill . . . . .                              | 7 | y error minus . . . . .                                 | 2 |
| <b>J</b>                                    |   | y error plus . . . . .                                  | 2 |
| jump mark left . . . . .                    | 4 | z . . . . .   | 2 |
| jump mark mid . . . . .                     | 4 | z error . . . . .                                       | 2 |
| jump mark right . . . . .                   | 4 | z error minus . . . . .                                 | 2 |
| <b>L</b>                                    |   | z error plus . . . . .                                  | 2 |
| line cap . . . . .                          | 6 | tension . . . . .                                       | 3 |
| line join . . . . .                         | 6 | text mark . . . . .                                     | 7 |
| line width . . . . .                        | 5 | text mark as node . . . . .                             | 7 |
| log basis . . . . .                         | 2 | text mark style . . . . .                               | 7 |
| loosely dashdotted <sub>sty</sub> . . . . . | 6 | thick <sub>sty</sub> . . . . .                          | 5 |
| loosely dashdotted <sub>sty</sub> . . . . . | 6 | thin <sub>sty</sub> . . . . .                           | 5 |
| loosely dashed <sub>sty</sub> . . . . .     | 6 | <b>U</b>  |   |
| loosely dotted <sub>sty</sub> . . . . .     | 6 | ultra thick <sub>sty</sub> . . . . .                    | 5 |
| <b>M</b>                                    |   | ultra thin <sub>sty</sub> . . . . .                     | 5 |
| mark . . . . .                              | 7 | <b>V</b>  |   |
| mark color . . . . .                        | 7 | variable . . . . .                                      | 3 |
| mark indices . . . . .                      | 7 | variable y . . . . .                                    | 3 |
| mark options . . . . .                      | 7 | very thick <sub>sty</sub> . . . . .                     | 5 |
| mark phase . . . . .                        | 7 | very thin <sub>sty</sub> . . . . .                      | 5 |
| mark repeat . . . . .                       | 7 | <b>X</b>  |   |
| mark size . . . . .                         | 7 | xbar . . . . .  | 4 |
| <b>N</b>                                    |   | xbar <sub>sty</sub> . . . . .                           | 4 |
| no markers . . . . .                        | 7 | xbar interval . . . . .                                 | 4 |
| <b>P</b>                                    |   | xbar interval <sub>sty</sub> . . . . .                  | 4 |
| plot coordinates/                           |   | xcomb . . . . .   | 5 |
| math parser . . . . .                       | 2 | xmode . . . . .   | 2 |
| <b>Q</b>                                    |   | xticklabel interval boundaries <sub>sty</sub> . . . . . | 4 |
| quiver . . . . .                            | 5 | <b>Y</b>  |   |
| quiver/                                     |   | ybar . . . . .  | 4 |
| after arrow <sub>code</sub> . . . . .       | 5 | ybar <sub>sty</sub> . . . . .                           | 4 |
| before arrow <sub>code</sub> . . . . .      | 5 | ybar interval . . . . .                                 | 4 |
| colored . . . . .                           | 5 | ybar interval <sub>sty</sub> . . . . .                  | 4 |
| every arrow <sub>sty</sub> . . . . .        | 5 | ycomb . . . . .   | 5 |
| quiver legend <sub>sty</sub> . . . . .      | 5 | ymode . . . . .   | 2 |
| scale arrows . . . . .                      | 5 | yticklabel interval boundaries <sub>sty</sub> . . . . . | 4 |
| u . . . . .                                 | 5 | <b>Z</b>  |   |
|   |   | zmode . . . . .   | 2 |
|   |   | zticklabel interval boundaries <sub>sty</sub> . . . . . | 4 |

## Concept Index

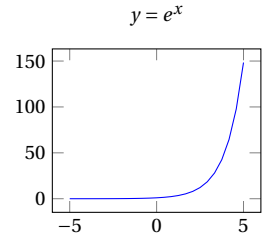
|                                     |                 |                            |                 |
|-------------------------------------|-----------------|----------------------------|-----------------|
| <b>@</b>                            |                 | coordinates list           |                 |
| + . . . . .                         | 2               | input data . . . . .       | 2               |
| <b>A</b>                            |                 | \coordindex . . . . .      | 2               |
| \addplot . . . . .                  | 2               | <b>D</b>                   |                 |
| .append style key handler . . . . . | 1               | dash pattern . . . . .     | 6               |
| auto . . . . .                      | 2               | dash phase . . . . .       | 6               |
| axisenv . . . . .                   | 1               | <b>E</b>                   |                 |
| axis scaling . . . . .              | 2               | expression . . . . .       | 3               |
| basis for logarithm . . . . .       | 2               | <b>H</b>                   |                 |
| <b>C</b>                            |                 | handler . . . . .          | see key handler |
| .cd key handler . . . . .           | 1               | <b>I</b>                   |                 |
| .code 2 args key handler . . . . .  | 1               | input data                 |                 |
| .code key handler . . . . .         | 1               | coordinates list . . . . . | 2               |
| code option . . . . .               | see key handler |                            |                 |
| coordinates . . . . .               | 2               |                            |                 |

## 1 General

### 1.1 Document Structure

```
\documentclass{standalone}
\usepackage{pgfplots}
\pgfplotsset{compat=1.16}

\begin{document}
\begin{tikzpicture}
\begin{axis}[title={y = e^x}]
\addplot+[no markers] {exp(x)};
\end{axis}
\end{tikzpicture}
\end{document}
```



### 1.2 PGFLOTS Options

```
\pgfplotsset{<key/value list>}
<key/value list> → (<key> = <value>,...)*
```

Options are supplied as a *<key/value list>*. The `/pgfplots/` and `/tikz/` prefixes in *<key>* can be omitted in the scope of PGFLOTS commands. Please note that a trailing comma in *<key/value list>* does no harm.

### 1.3 Key Handlers

```
\pgfplotsset{<key>/.style = {<key/value list>}}
```

Define or replace style *<key>*.

```
\pgfplotsset{<key>/.append style = {<key/value list>}}
```

Append to style *<key>*.

```
\pgfplotsset{<key>/.code = {<TeX code>}}
```

Define or replace *<key>* that – when run – takes one argument; *<TeX code>* can refer to the supplied argument as #1. Invoke as `\pgfplotsset{<key> = {<argument>}}`.

```
\pgfplotsset{<key>/.code 2 args = {<TeX code>}}
```

Like *<key>/.code* but with two arguments; *<TeX code>* can refer to the supplied arguments as #1 and #2. Invoke as `\pgfplotsset{<key> = {<first argument>}{<second argument>}}`.

```
\pgfplotsset{<key>/.cd}
```

Make *<key>* the default prefix.

### 1.4 Mathematical Expressions

See the TikZ/PGF manual for a detailed description.

Use parenthesis, ( and ), for grouping. Arguments and values of trigonometric functions are in degree angle.

*Arithmetic Operators:* +, − (also unary minus), \*, /, ^ (exponentiation), ! (factorial, postfix operator), r (radian, postfix operator, see deg).

*Relational Operators:* ==, !=, <, <=, >, >=.

*Logical Operators:* ! (not, prefix operator), || (or), && (and).

*Conditionals:* <condition>?<true>:<false>.

*Constants:* pi, e, false, true.

*Unary Functions:* abs, sign, int, frac (fractional part), round, floor, ceil, factorial (see !), iseven, isodd, isprime, sqrt, exp, ln, log10, log2, sin, cos, tan, cot, sec, cosec, asin, acos, atan, deg (degree from radian), rad (radian from degree), sinh, cosh, tanh.

*Binary Functions:* div (integer division), mod, Mod (unsigned result), gcd, pow (see ^), atan2, vecLen (vector length in  $\mathbb{R}^2$ ).

*n-ary Functions:* min, max.

*Pseudo-Random Number Functions (Uniform Distribution):* rnd ([0,1]  $\cap \mathbb{R}$ ), rand ([−1,1]  $\cap \mathbb{R}$ ), random(*n*) ([1,*n*]  $\cap \mathbb{N}$ ), random(*m*,*n*) ([*m*,*n*]  $\cap \mathbb{Z}$ ).

## 2 Axis Environments

```
\begin{axis}[<axis options>]?
<axis options> → <key/value list>
```

`axisenv` can also be `semilogxaxisenv`, `semilogyaxisenv`, or `loglogaxisenv`.

▷ `every <type>? axis` style  
`<type> → (linear|semilogx|semilogy|loglog)`  
 Define default axis options.

▷ `xmode|ymode|zmode = normal|linear|log` option  
 Customize axis scaling; `linear` is a synonym for `normal`.

▷ `log basis (x|y|z) = <empty>|<positive number>` option  
 The basis for logarithmic axis scaling. Empty means to apply the natural logarithm (base  $e$ ) to any input coordinate – if the axis scaling is logarithmic – and use the decadic/common logarithm (base 10) for displaying tick labels. Any non-empty value causes both, coordinates and tick labels, to use the logarithm with base `<number>`.

## 3 Plots

`\addplot[<plot options>]? <input data> <trailing TikZ path commands>;`  
`\addplot (without options) and \addplot+ [<plot options>]` utilize default options from the cycle list. `\addplot [<plot options>]` only use the manually provided options.

▷ `every axis plot (no  $n$ )?` style  
 Define `<plot options>` for all plots or for the  $n^{\text{th}}$  plot of every axis. Plot numbers are zero-based.

### 3.1 Input Data

▷ `empty line = auto|none|scanline|jump` option  
 How to handle empty lines in `<coordinates list>`, `none` means to do nothing, `jump` means to insert a discontinuity.

#### 3.1.1 Coordinates List

`<input data> → coordinates {<coordinates list>}`  
`<coordinates list> → <coordinates>*`  
`<coordinates> → (x, y, z) (+- (u, v, w))? [(meta data)]?`

Read input data from a sequence of coordinates.  $x$ ,  $y$ , and  $z$  are the point coordinates.  $u$ ,  $v$ , and  $w$  are the error coordinates (reliability bounds) for error bar plots. Coordinate  $z$  and  $w$  are only mandatory for 3D plots. Empty lines in the `<coordinates list>` indicate discontinuities; use `\\` when gathering coordinates in a `TikX` macro.

▷ `plot coordinates/math parser = true|false` option  
 Whether or not to enable mathematical expressions in every coordinate inside of a `<coordinates list>`.

#### 3.1.2 Table Data

`<input data> → table [<table options>]? {<table data>}`  
`<table data> → <file name>|<inline table>`

Read input data from table columns.

▷ `table/<coordinate> = <column name>` option  
 ▷ `table/<coordinate> index = <column index>` option  
 ▷ `table/<coordinate> expr = <expression>` option  
`<coordinate> → x|y|z|(x|y|z) error (plus|minus)?|meta`

Column names are case sensitive and have to exist. Use `{<column name>}` to quote non-trivial column names. The first column has index zero. Within `<expression>` `\thisrow{<column name>}` and `\thisrowno{<column index>}` yields the cell value of the specified column. Likewise, `\coordindex` yields the index of the current set of coordinates and `\lineno` yields the total line number. Both numbers start counting at zero.

▷ `table/header = true|false` option  
 Whether or not to check `<table data>` for column names. If enabled, the first non-comment line is checked for column names. That means if any element is not a number, all entries are treated as column names.

▷ `table/skip first n = 0|<non-negative integer>` option  
 Don't process the first  $n$  lines in `<table data>`.

All markers plotted with `'mark options = {draw = blue, fill = yellow}'` and `'mark color = pink'`. You can rotate makers with, e.g., `'mark options = {rotate = 90}'`.

▷ `/tikz/mark = *|<marker>` option  
 Use `<marker>`.

▷ `/tikz/mark size = 2pt|<dimension>` option  
 Marker size, `<dimension>` is either the radius or about half the width or height.

▷ `/tikz/mark repeat = 1|<integer>` option  
 Draw a marker at every `<integer>`<sup>th</sup> sample.

▷ `/tikz/mark phase = 1|<integer>` option  
 Draw the first marker at the `<integer>`<sup>th</sup> sample; `<integer>` is one based.

▷ `/tikz/mark indices = {}|{<comma-separated list of integers>}` option  
 Explicit sample indices for drawing markers.  
`<comma-separated list of integers>` can contain ... expressions, for example `'mark indices = {1, 2, ..., 7}'`.

▷ `/tikz/every mark` style  
 This style is applied before drawing a marker.

▷ `/tikz/mark options = {<options>}` option  
 Redefine 'every mark' so that it sets `<options>`.

▷ `/pgfplots/no markers` style  
 Disable markers; even for cycle lists that contain markers.

▷ `/pgf/mark color = white|<color>` option  
 Additional fill color for `halfcircle*`, `halfcircle*`, `halfdiamond*`, and `halfsquare*` markers.

▷ `/pgf/text mark = p|<text>` option  
 Define the text for 'mark = text'.

▷ `/pgf/text mark as node = false|true` option  
 Whether or not to draw text markers as nodes.

▷ `/pgf/text mark style = {<options>}` option  
 Customize the appearance of text markers. When 'text mark as node' is true, 'text mark style' are \node options. Otherwise, 'text mark style' are \pgftext options.

## 4.6 Colors

Color support is provided by the `xcolor` package. Standard color names:

|             |          |           |          |
|-------------|----------|-----------|----------|
| ■ black     | ■ red    | ■ green   | ■ blue   |
| ■ darkgray  | ■ cyan   | ■ magenta | ■ yellow |
| ■ gray      | ■ brown  | ■ lime    | ■ olive  |
| ■ lightgray | ■ orange | ■ pink    | ■ purple |
| □ white     | ■ teal   | ■ violet  | ■ none   |

▷ `/tikz/color = <color>` option  
 Set the color for drawing and filling. You can omit the option key if `<color>` is a color name.

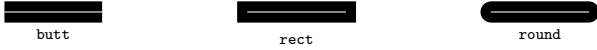
▷ `/tikz/draw = <color>` option  
 ▷ `/tikz/fill = <color>` option  
 Set the color for drawing or filling respectively. You can use `none` as `<color>` to disable drawing or filling.

`\definecolor{<name>}{<model>}{<spec>}`  
`<model> → rgb|cmy|cmyk|hsb|Hsb|tHsb|gray|RGB|HSB|Gray|HTML|wave`  
`<rgb spec> → x, x, x`  
`<cmy spec> → x, x, x`  
`<cmyk spec> → x, x, x, x`  
`<hsb spec> → x, x, x`  
`<Hsb spec> → H, x, x`  
`<tHsb spec> → H, x, x`  
`<gray spec> → x`  
`<RGB spec> → L, L, L`  
`<HSB spec> → M, M, M`  
`<Gray spec> → N`  
`<HTML spec> → [000000]16, FFFFFFFF16`  
`<wave spec> → [363, 814]`  
 $x = [0, 1]$ ,  $H = [0, 360]$ ,  $L = [0, 255] \cap \mathbb{Z}$ ,  $M = [0, 240] \cap \mathbb{Z}$ , and  $N = [0, 15] \cap \mathbb{Z}$ . All colors are defined in the sRGB color space. HSB is a synonym for HSL.

## 4.2 Line Cap

▷ /tikz/line cap = butt|rect|round  
Set the line cap style.

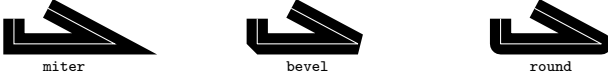
option



## 4.3 Line Join

▷ /tikz/line join = miter|bevel|round  
Set the line join style.

option



▷ /tikz/miter limit = 10|<number>

option

When the ratio of the miter length to the line width is greater than <number>, the miter join is replaced by a bevel. A miter limit  $\ell = 1/\sin(\alpha/2)$  for  $\alpha \in (0^\circ, 180^\circ)$  will create a bevel join for angles less than  $\alpha = 2 \cdot \arcsin(1/\ell)$ .

## 4.4 Dash Pattern

|                               |  |       |
|-------------------------------|--|-------|
| ▷ /tikz/solid                 |  | style |
| ▷ /tikz/dashed                |  | style |
| ▷ /tikz/dotted                |  | style |
| ▷ /tikz/dashdotted            |  | style |
| ▷ /tikz/dashdotdotted         |  | style |
| ▷ /tikz/densely dashed        |  | style |
| ▷ /tikz/densely dotted        |  | style |
| ▷ /tikz/densely dashdotted    |  | style |
| ▷ /tikz/densely dashdotdotted |  | style |
| ▷ /tikz/loosely dashed        |  | style |
| ▷ /tikz/loosely dotted        |  | style |
| ▷ /tikz/loosely dashdotted    |  | style |
| ▷ /tikz/loosely dashdotdotted |  | style |

Predefined line styles.

▷ /tikz/dash pattern = ((on|off) <dimension>)+  
Set the dash pattern (line style) for drawing lines, e.g.,  
'dash pattern = on 3.5mm off 0.7mm'.

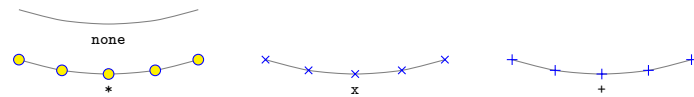
option

▷ /tikz/dash phase = Opt|<dimension>  
Start the dash pattern at offset <dimension>.

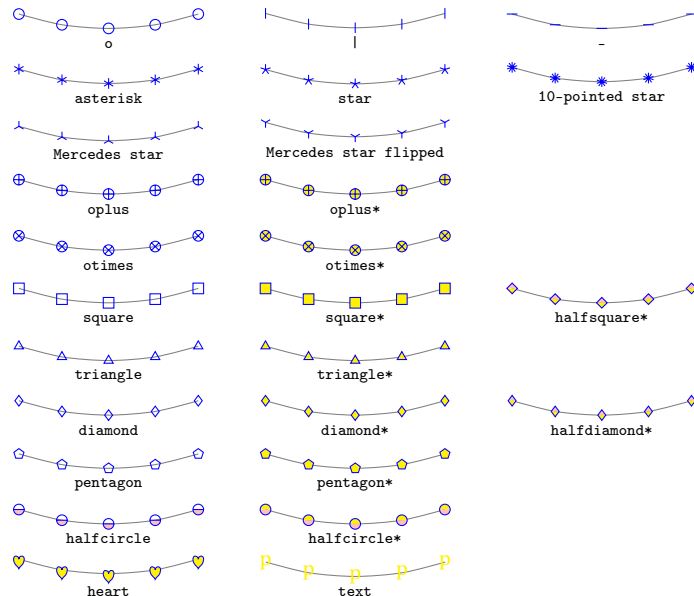
option

## 4.5 Markers

Standard markers:



With \usetikzlibrary{plotmarks}:



▷ table/ignore chars = { }|<comma-separated list> option  
▷ table/white space chars = { }|<comma-separated list> option  
▷ table/comment chars = { }|<comma-separated list> option

Extra characters to be ignored, treated like a whitespace character (beside space and tab), or treated like a comment start character (beside # and %).

▷ table/row sep = <newline>|\\ option

Use \\ as the row separator if you experience problems with <newline>, for example with inline table data or when gathering table data in a  $\text{\TeX}$  macro.

▷ table/col sep = space|tab|comma|semicolon|colon|\_|braces|&|ampersand option

A space column separator means one or more space or tab characters. With braces, every table cell looks like  $\langle \text{contents} \rangle$  and whitespace characters between adjacent table cells is ignored. A & column separator implies 'table/trim cells = true'.

▷ table/read completely = auto|true|false option  
Whether or not to read the whole table into memory. Use with care!

▷ table/search path = { }|<comma-separated list> option  
▷ table/search path/implicit . = true|false option

Search path for input files, . means to use the standard  $\text{\TeX}$  procedure.

\pgfplotstablerread{<file name>}\foo  
\addplot table [<table options>] {\foo};

Read table data once so that you can use it multiple times; \foo is a user-defined command sequence.

### 3.1.3 Mathematical Expressions

$\langle \text{input data} \rangle \rightarrow \text{expression}^? \{ \langle \text{expression} \rangle \}$   
 $\langle \text{input data} \rangle \rightarrow \langle \text{x-expression} \rangle, \langle \text{y-expression} \rangle, \langle \text{z-expression} \rangle$

Create input data by sampling a mathematical expression over an argument domain. The second form can be used to create parametric plots. Say  $\{ \langle \text{x-expression} \rangle \}$  if  $\langle \text{x-expression} \rangle$  contains parenthesis or commas. The  $\langle \text{z-expression} \rangle$  is only mandatory for 3D plots.

▷ domain = -5:5|<x<sub>1</sub>>:<x<sub>2</sub>> option  
▷ domain y = <empty>|<y<sub>1</sub>>:<y<sub>2</sub>> option

Define the argument domain for the x-axis to the closed interval  $[x_1, x_2]$ . Likewise for the y-axis for 3D plots. If domain y is empty, use the value of domain.

▷ samples = 25|<non-negative integer> option  
▷ samples y = <empty>|<non-negative integer> option

The number of samples to be generated. Samples are equally spaced over the corresponding argument domain. If 'samples y' is empty, use the value of samples.

▷ samples at = { }|<comma-separated list of numbers> option

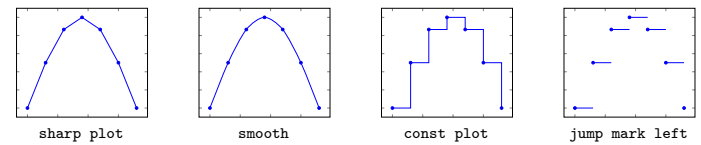
Explicit argument values for sampling <expression>. This option always overrides the domain and samples options.

<comma-separated list of numbers> can contain ... expressions, for example '-2, -1.8, ..., 2'.

▷ variable = x|<variable name> option  
▷ variable y = y|<variable name> option

The variable name containing the argument value when evaluating <expression>.

## 3.2 Line Plots



▷ /tikz/sharp plot option  
Connect points by straight lines. This is the default.

▷ /tikz/smooth option  
▷ /tikz/tension = 0.55|<number> option

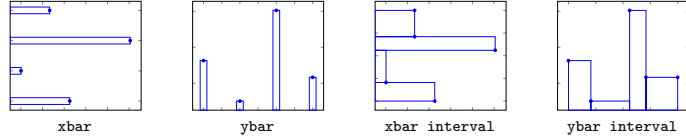
Connect points by a smooth curve. For best results, points should be equidistant and the bending angles should be less than about  $30^\circ$ . The tension option controls the sharpness of the corners; 0 yields sharp corners and 1 yields a circle if the path is a square.

`> /tikz/const plot` option  
`> /tikz/const plot mark (left|mid|right)` option

Connect points with horizontal and vertical line segments. ‘const plot’ is an alias for ‘const plot mark left’. Markers are placed on the left corner, in the middle, or on the right corner of the horizontal line segments. Use ‘const plot, no markers’ to omit the markers.

`> /tikz/jump mark (left|mid|right)` option  
 Like ‘const plot’ but omit the vertical line segments.

### 3.3 Bar Plots



`> /tikz/xbar` option  
`> /tikz/ybar` option  
 Render coordinates as horizontal or vertical bars respectively.

`> /pgf/bar width = 10pt|<dimension>|<number>` option  
 Width of a single bar. <dimension> is a TeX dimension and <number> is in axis units. Value can be a mathematical expression. The fully computed value is then available in `\pgfplotbarwidth`.

`> /pgf/bar shift = 0pt|<dimension>|<number>` option  
 Off-center distance for the bars. <dimension> is a TeX dimension and <number> is in axis units. Value can be a mathematical expression. The fully computed value is then available in `\pgfplotbarshift`.

`> xbar` style  
`> xbar( = 2pt|<dimension>|<number>)?` option  
`> ybar` style  
`> ybar( = 2pt|<dimension>|<number>)?` option

Predefined axis style for bar plots; implies `/tikz/xbar` or `/tikz/ybar` respectively, `bar shift autosty`, and `bar cycle liststy`. The default handler takes one optional argument which is passed on to `bar shift autosty`.

`> bar shift auto` style  
`> bar shift auto = 2pt|<dimension>|<number>` option

Predefined axis style setting `/pgf/bar shift` to the correct value based on the current plot number and the total number of plots. Argument is the distance between adjacent bars of a group.

When  $n$  bar plots are added to an axis, the total width of a group of bars is  $n \times \langle \text{bar width} \rangle + (n - 1) \times \langle \text{bar shift auto} \rangle$ .

`> bar cycle list` style  
 Predefined axis style installing a cycle list for bar plots.

`> bar direction = auto|x|y` option  
 Explicitly set the bar plot direction. Not needed if you say, for example ‘ybar’, `bar width = 1`, because the direction is clear from the context.

`> /tikz/xbar interval` option  
`> /tikz/ybar interval` option

Like `/tikz/xbar` or `/tikz/ybar` respectively, but draw the bar width as an interval from this point to the next point. You need one extra point to define the interval for the last bar.

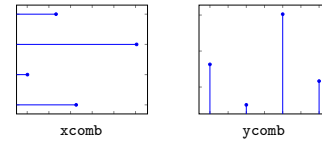
`> xbar interval` style  
`> xbar interval( = 1|<relative width>)?` option  
`> ybar interval` style  
`> ybar interval( = 1|<relative width>)?` option

Predefined axis style for interval bar plots; implies `/tikz/xbar interval` or `/tikz/ybar interval` respectively and `bar cycle liststy`. The default handler takes one optional argument to scale the intervals.

`> xticklabel interval boundaries` style  
`> yticklabel interval boundaries` style  
`> zticklabel interval boundaries` style

Axis style to display the interval bounds in the tick labels.

### 3.4 Comb Plots



`> /tikz/xcomb` option  
`> /tikz/ycomb` option  
 Render coordinates as horizontal or vertical lines respectively.

### 3.5 Quiver Plots

`> quiver = {<quiver options>}` option  
 Render coordinates as small arrows. The origin of the arrow is at the final point coordinates  $(x, y, z)$  and the direction and length of the arrow is defined by the direction coordinates  $(u, v, w)$ . The quiver/ prefix can be omitted within `<quiver options>`.

`> quiver/(u|v|w) = 0|<expression>` option  
 The direction coordinates of the arrows. Within `<expression>`,  $x$ ,  $y$ , and  $z$  are bound to the final point coordinates. For parametric plots use ‘variable = t’ and ‘quiver/u = f(t)’ and ‘quiver/v = g(t)’ to access the parameter.  

```
\addplot[
  variable = t,
  quiver = {u = {-sin(t)}, v = {cos(t)}},
]
{cos(t)}, {sin(t)};
```

`> quiver/(u|v|w) value = 0|<number>` option  
 Like `quiver/u`, `quiver/v`, and `quiver/w` respectively but without parsing mathematical expressions. However, `\thisrow{<column name>}` and similar code works.

`> quiver/colored` option  
`> quiver/colored = mapped color|<color>` option  
 Set a different color for each arrow. `quiver/colored` is an alias for ‘quiver/colored = mapped color’. Please note that ‘<color>, quiver = ...’ is more efficient if `<color>` is constant.

`> quiver/scale arrows = 1|<number>` option  
 Scale all arrows by a constant factor.

`> quiver/update limits = true|false` option  
 Whether or not the coordinates of the arrow heads shall be considered when determining the axis limits.

`> quiver/every arrow` style  
 Style to customize arrows individually at visualization time.

`> quiver/before arrow` code  
`> quiver/after arrow` code  
 Run `<TeX code>` before and after drawing a single arrow. Empty by default.

`> quiver/quiver legend` style  
 Style that redefines `legend image` code in order to produce a suitable legend for quiver plots.

## 4 Lines and Markers

### 4.1 Line Width

`> /tikz/ultra thin` style  
`> /tikz/very thin` style  
`> /tikz/thin` style  
`> /tikz/semithick` style  
`> /tikz/thick` style  
`> /tikz/very thick` style  
`> /tikz/ultra thick` style

Predefined line widths.

`> /tikz/line width = 0.4pt|<dimension>` option  
 Set the line width.