

The TikZ-Extensions Package
Manual for version 0.6.1 (9)
<https://github.com/Qrrbrbirlbel/tikz-extensions>

Qrrbrbirlbel

March 29, 2025

Contents

I	Introduction	5
1	Usage	5
2	Why do we need it?	5
3	Having problems?	5
4	Namespaces and <i>TikZ</i> -Extensions macros	5
5	Compatibility with older versions	6
II	<i>TikZ</i> Libraries	7
6	Arrow Pics	8
6.1	Arrow pic types	9
6.2	Arrow keys	9
6.3	Shifted and bended arrows for the <code>decorations.markings</code> library	10

7	Calendar	11
7.1	Value-keys and nestable if key	11
7.2	PGFmath functions	11
7.3	Week numbering (ISO 8601)	11
8	Layers	12
9	Node Families	13
9.1	Externalization	13
9.2	Text Box	13
9.3	Minimum Width/Height	14
9.4	More shapes that support the keys width and height	16
10	Nodes	17
10.1	Pic as a node	17
10.2	Nodes on paths	17
10.2.1	Nodes on Lines	17
10.2.2	Nodes on Curves	18
10.3	Automatic placement of nodes	18
10.3.1	More than left and right	18
10.3.2	Offset	18
10.3.3	Precise placement	19
11	Arc to a point	20
12	More Horizontal and Vertical Lines	22
12.1	Zig-Zag	22
12.2	Zig-Zig	24
12.3	Even more Horizontal and Vertical Lines	25
13	Extending the Path Timers	28
13.1	Rectangle	28
13.2	Parabola	29
13.3	Sine/Cosine	29
14	Using Images as a Pattern	31

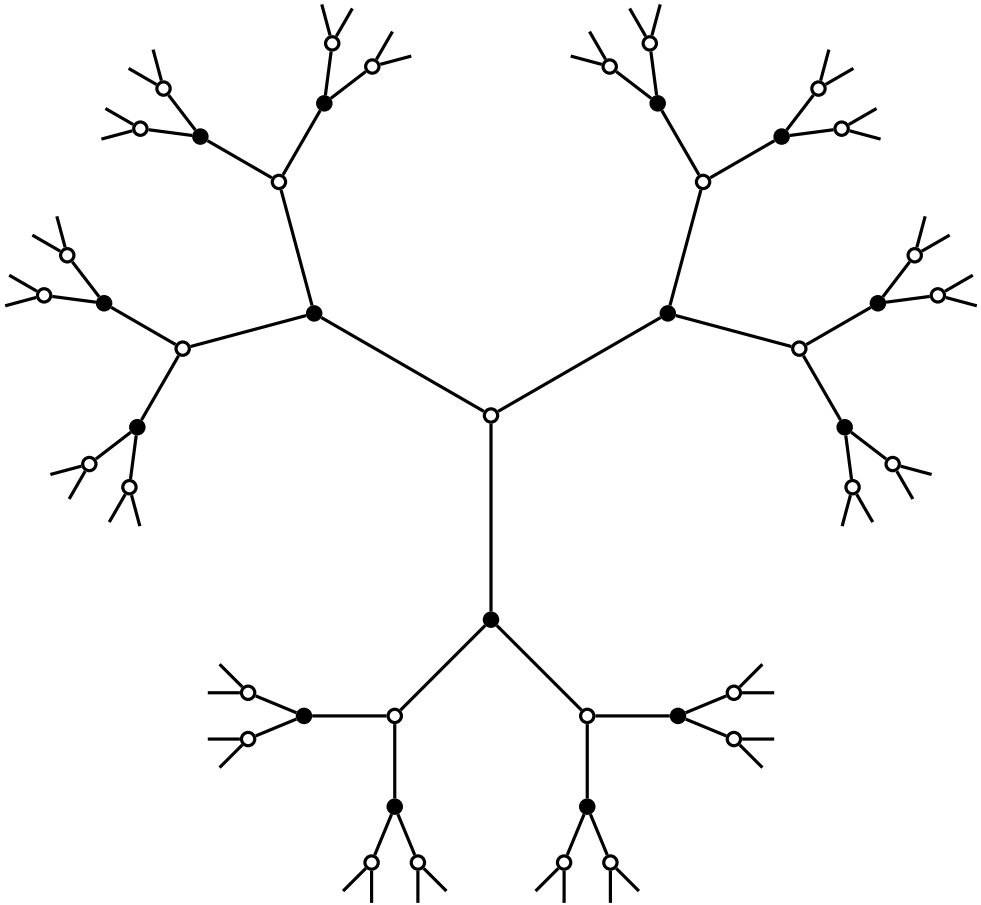
15	Positioning Plus	32
15.1	Useful corner anchors	32
15.2	Useful placement keys for vertical and horizontal alignment	33
16	Scaling Pictures to a Specific Size	35
16.1	Externalization	35
16.2	Keeping the aspect ratio	35
16.3	Changing the aspect ratio	36
17	Arcs through Three Points	37
18	Autobending	38
19	Mirror, Mirror on the Wall	40
19.1	Using the reflection matrix	40
19.2	Using built-in transformations	42
III	PGF Libraries	44
20	Arrow Tips	45
20.1	Centered	46
20.1.1	Barbed Arrow Tips	46
20.1.2	Geometric Arrow Tips	46
20.1.3	Special Arrow Tips	47
20.2	Untipped	47
20.2.1	Barbed Arrow Tips	47
20.2.2	Geometric Arrow Tips	47
20.3	Original Arrow Tips	48
21	Transformations: Mirroring	50
21.1	Using the reflection matrix	50
21.2	Using built-in transformations	50
22	Shape: Circle Arrow	52
23	Shape: Circle Cross Split	56
24	Shape: Heatmark	59

25	Shape: Rectangle with Rounded Corners	62
26	Shape: Superellipse	64
27	Shape: Uncentered Rectangle	67
IV	Utilities	70
28	Calendar: Weeknumbers and more conditionals	71
28.1	Extensions	71
28.2	Week numbering (ISO 8601)	72
29	Repeating Things and Other Things	73
30	And a little bit more	75
30.1	PGFmath	75
30.1.1	Postfix operator R	75
30.1.2	Functions	75
30.1.3	Functions: using coordinates	76
30.2	PGFfor	77
30.3	PGFkeys	77
30.3.1	Conditionals	77
30.3.2	Handlers	77
30.4	TikZ	80
V	Changelog, Index & References	81
	Changelog	81
	Index	83
	References	86

Part II

TikZ Libraries

These libraries only work with TikZ.



6 Arrow Pics

TikZ Library `ext.arrows-plus`

```
\usetikzlibrary{ext.arrows-plus} % LATEX and plain TEX
\usetikzlibrary[ext.arrows-plus] % ConTEXt
```

This library defines pics and keys that can be used to place (bended) arrow tips on paths.

The markings decoration already provides the functionality to place arrow tips along the path. The pics and keys provided by this library serve as an alternative.

Many of the pics and keys share various keys that specify where and how the arrow tips are placed.

`/tikz/ext/pos <=<value>` (no default, initially 0.0)

If the pic type supports it and a start arrow tip sequence is provided this specifies the position of that sequence.

`/tikz/ext/pos >=<value>` (no default, initially 0.5)

This is an alias for `/tikz/pos`, if an end arrow tip sequence is provided, it is placed at this position.

`/tikz/ext/angle pos <=<angle>` (no default)

For tips along an arc the angle along that arc can be specified for the start tip sequence.

`/tikz/ext/angle pos >=<angle>` (no default)

For tips along an arc the angle along that arc can be specified for the end tip sequence.

`/tikz/ext/arrow shift mode=<shift mode>` (no default, initially total length)

This key is used to set the *<shift mode>* for the arrow tip. It can be one of the following.

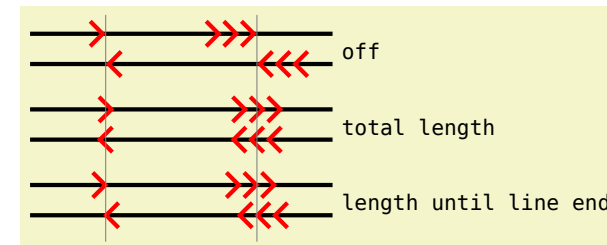
arrow shift mode=off This disables the shifting.

arrow shift mode=total length The total length of the whole arrow tip sequence will be used.

arrow shift mode=total This is an alias for total length.

arrow shift mode=length until line end The length of the whole arrow tip until the line end will be used – as reported by PGF which might not always be the expected one.

arrow shift mode=line end This is an alias for length until line end.



```
\usetikzlibrary {ext.arrows-plus}
\begin{tikzpicture}[>={Straight Barb[color=red]}, ultra thick]
\ttfamily
\foreach[count=\y] \shiftmode in {off, total length, length until line end}
\draw[ext/arrow shift mode=\shiftmode] (0, -\y)
-- pic {ext/arrow=>} ++(right:2)
-- pic {ext/arrow=>.>} ++(right:2) node[below right] {\shiftmode}
++(down:.4) -- pic {ext/arrow=>.>} ++(left:2)
-- pic {ext/arrow=>} ++(left:2);
\draw[thin, gray] (1,-.75) -- +(down:3) (3,-.75) -- +(down:3);
\end{tikzpicture}
```

For single arrow tips it might be better to use the Centered arrow tip variants of the `ext.arrows` library (see sec 20) and disabled `arrow shift mode`.

When an arrow tip sequence is to be drawn depending on the shift mode its total length or its length until the line end will be determined and multiplied with the `arrow shift` factor. The result of this evaluation is used to shift the arrow tip sequence in the tip's direction.

`/tikz/ext/arrow shift factor=<value>` (no default, initially 0.5)

This determines the shift factor.

The default value is probably good for most cases.

6.1 Arrow pic types

This library provides the following pics:

ext/arrow This is the simplest implementation to place an arrow tip along a path. It uses the current timer that is also used to place nodes.

It can be used without any adjustment for every path operation that provides such a timer. These do *not* include circle, ellipse, plot and grid. For rectangle, parabola, sin and cos, the `ext.paths.timer` library is recommended or even necessary (see section 13).

The arrow tips will never be bended. For this the following pic types or the `/tikz/ext/arc arrows` key will be necessary.

Due to [1] with an active transformation, the arrow tips won't be placed correctly in many cases. For this *and* bended arrow tips the following pics are necessary.

ext/softpath arrows This pic type places a possible bended arrow tip on the last segment of the path.

This won't work for arcs, for this the `/tikz/ext/arc arrows` key will be necessary.

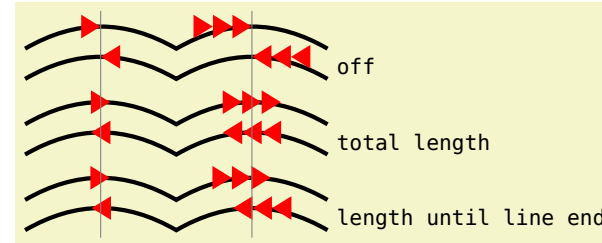
This pic type can place two tip specification, one at `pos >` and one at `pos <` in the reversed direction.

ext/softpath arrow This is an alias for `softpath arrows` with an empty start arrow tip specification.

Pic type ext/arrow=<arrow tip specification>

This pic draws the given *<arrow tip specification>* (defaults to the end tip specification of the path).

This obviously is best used as a pic along a path segment that supports it. It *does not* support bended arrow tips.



```
\usetikzlibrary {bending, ext.arrows-plus}
\begin{tikzpicture}[>={Triangle[color=red]}, arrows={[bend]}, ultra thick]
\ttfamily
\foreach[count=\y] \shiftmode in {off, total length, length until line end}
\draw[ext/arrow shift mode=\shiftmode] (0, -\y)
    to[bend left] pic {ext/arrow=>} ++(right:2)
    to[bend left] pic {ext/arrow=>>.>} ++(right:2)
    node[below right] {\shiftmode}
    ++(down:.4) to[bend right] pic {ext/arrow=>>.>} ++(left:2)
    to[bend right] pic {ext/arrow=>} ++(left:2);
\draw[thin, gray] (1,-.5) -- +(down:3) (3,-.5) -- +(down:3);
\end{tikzpicture}
```

Pic type ext/softpath arrows=<start tip specification>-<end tip specification>

This pic draws the given arrow tip specification (defaults to the already present tip specification of the path) along the previous path segment (a curve or a line). It supports the `pos <` key.

Note: For arcs with an angle greater than 90° this will not work as expected. Use the `arc arrows` key instead.

Pic type ext/softpath arrow=<end tip specification>

This pic type is an alias for `softpath arrows = -<end tip specification>`.

6.2 Arrow keys

The last pic type `softpath arrows` is also available as a key which is the preferred version.

`/tikz/ext/softpath arrows=<options>` (default ->)

This key adds arrow tips to the previous path segment (a curve or a line).

`/tikz/ext/every softpath arrows` (style, initially {})

This style will be applied for every instance of `softpath arrows` (key version, not the `pic`). It also sets up forwarding

- from `/tikz/pos >` to `/tikz/ext/pos >` and
- from `/tikz/pos <` to `/tikz/ext/pos >`.

For arcs the following key needs to be used directly after the arc path operation.

`/tikz/ext/arc arrows=<options>` (default ->)

This key adds arrow tips to the previous arc segment.

`/tikz/ext/every arc arrows` (style, initially {})

This style will be applied for every instance of `arc arrows`. It also sets up forwarding

- from `/tikz/pos >` to `/tikz/ext/pos >`,
- from `/tikz/pos <` to `/tikz/ext/pos >` as well as
- from `/tikz/angle pos >` to `/tikz/ext/angle pos >` and
- from `/tikz/angle pos <` to `/tikz/ext/angle pos >`.

Tip: Use an arc with the full 360° to place bended arrow tips along a circle or an ellipse.

6.3 Shifted and bended arrows for the `decorations.markings` library

Many paths are not properly accessible by the previous methods. If this library is loaded *after* the `decorations.markings` library, both the `\arrow` and the `\arrowreversed` macros are enhanced.

`\arrow**[<options>]{<arrow end tip>}`

This macro works the same as before but the one-starred version applies the shifting as specified by `arrow shift mode` and `arrow shift factor` where as the two-starred version also bends the arrow tip.

`\arrowreversed**[<options>]{<arrow end tip>}`

As above, only the arrow end tip is flipped and points in the other direction.

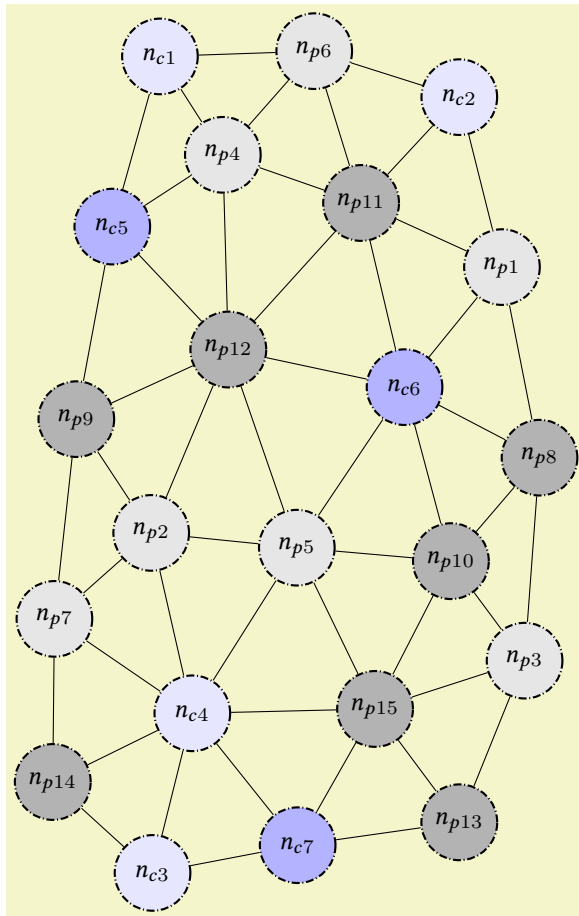


```
\usetikzlibrary {bending, decorations.markings, ext.arrows-plus}
\tikzset{
  arr/.style={
    postaction=decorate,
    decoration={
      name=markings,
      mark={between positions .25 and 1 step .25 with
        \arrow#1[red]{> _ < _ }}}
\tikz[y=1.5cm, >=Stealth, arrows={[round]}, nodes={circle, draw}]
\path node[arr= ]{Ti\emph kZ} % \arrow
(0,-1) node[arr=* ]{Ti\emph kZ} % \arrow*
(0,-2) node[arr=**]{Ti\emph kZ} % \arrow**
;
```


Part III

PGF Libraries

These libraries (should) work with both PGF and TikZ.



```
\usetikzlibrary {graphs,graphdrawing,ext.misc} \usegdlibrary {force}
\tikzset{
  mynode/.style={
    circle, minimum size=10mm, draw, densely dashdotted, thick,
    decide color/.expand once=#1,
    decide color/.style 2 args={
      /utils/ext/if=c#1
      {/utils/ext/ifnum={#2<5}{blue!light}{blue!dark}}
      {/utils/ext/ifnum={#2<8}{light}{dark}}},
    light/.style={fill=gray!20}, blue!light/.style={fill=blue!10},
    dark/.style={fill=gray!60}, blue!dark/.style={fill=blue!30}}
\tikz\graph[
  spring electrical layout, vertical=c2 to p13,
  node distance=1.5cm, typeset=$n_{\tikzgraphnodetext}$,
  nodes={mynode=\tikzgraphnodetext}] {
  % outer ring
  c2 -- {p1, p11, p6};
  p1 -- {p8, c6, p11};
  p8 -- {p3, p10, c6};
  p3 -- {p13, p15, p10};
  p13 -- {p15, c7};
  c7 -- {c3, c4, p15};
  c3 -- {p14, c4};
  p14 -- {p7, c4};
  p7 -- {p9, p2, c4};
  p9 -- {c5, p12, p2};
  c5 -- {c1, p4, p12};
  c1 -- {p6, p4};
  p6 -- {p11, p4};
  % inner ring
  p11 -- {c6, p12, p4};
  p5 -- {c6 -- {p10, p12}, p10 -- p15, p15 -- c4, c4 -- p2, p2 -- p12, p12 -- p4};
};
```

20 Arrow Tips

TikZ Library `ext.arrows`

```
\usepgflibrary{ext.arrows} % LATEX and plain TEX and pure pgf
\usepgflibrary[ext.arrows] % ConTEXt and pure pgf
\usetikzlibrary{ext.arrows} % LATEX and plain TEX when using TikZ
\usetikzlibrary[ext.arrows] % ConTEXt when using TikZ
```

This library adds arrows to PGF/TikZ.

























Q & A: [5, 3, 2, 4, 6] & [12, 10, 13, 11, 9]

The arrow tips of the `arrows.meta` library always just touch the end of original line – which is usually what you want.

But for some arrow tips (and when they lie along a path) it makes sense that these tips shoot a bit over the end of the line. This is why these arrow tips exist. They can be categorized into three groups:

1. Centered
2. Untipped
3. Overtipped⁶

Not all original arrow tips got all variants. For a summary, refer to table on the right side. As with the original tips of the `arrows.meta` library these can be organized in the following categories.

Group	Original	Centered	Untipped	Overtipped
Barbed	Arc Barb			–
	Parenthesis			–
	Hooks		–	–
	Straight Barb		–	–
	Tee Barb			–
	Bar			–
	Bracket			–
Geometric	Circle			–
	Ellipse			–
	Kite		–	–
	Diamond		–	–
	Turned Square		–	–
	LaTeX	–	–	–
	Square		–	–
	Rectangle		–	–
	Stealth		–	–
	Triangle		–	–
Rays	Rays		–	–

⁶The Overtipped arrow tips aren't yet implemented.

20.1 Centered

20.1.1 Barbed Arrow Tips

Arrow Tip Kind ext_Centered Arc Barb

pre 0.6 Centered Arc Barb

This is a variant of the Arc Barb tip. The center of the arc lies on the original end of the path.

Arrow Tip Kind ext_Centered Bar

pre 0.6 Centered Bar

A variant of the simple Bar tip. This is a simple instance of ext_Centered Tee Barb for length zero.

The middle of the line will lie on original end of the path.

Arrow Tip Kind ext_Centered Bracket

pre 0.6 Centered Bracket

This is a variant of the Bracket tip and therefore an instance of the ext_Centered Tee Barb arrow tip that results in something resembling a bracket.

The middle of the vertical part will lie on the original end of the path.

Arrow Tip Kind ext_Centered Hooks

pre 0.6 Centered Hooks

A variant of the Hooks tip. The starting point of the hooks will lie on the original end of the path.

Arrow Tip Kind ext_Centered Parenthesis

pre 0.6 Centered Parenthesis

This is a variant of the Parenthesis tip and thus an instance of the ext_Centered Arc Barb arrow tip.

Arrow Tip Kind ext_Centered Straight Barb

pre 0.6 Centered Straight Barb

A variant of the Straight Barb tip.

Arrow Tip Kind ext_Centered Tee Barb

pre 0.6 Centered Tee Barb

A variant of the Tee Barb tip.

The middle of the vertical part will lie on the original end of the path.

20.1.2 Geometric Arrow Tips

Arrow Tip Kind ext_Centered Circle

pre 0.6 Centered Circle

A variant of the Circle tip. The center of the circle will lie on the original end of the path.

Arrow Tip Kind ext_Centered Diamond

pre 0.6 Centered Diamond

This is a variant of the Diamond tip and thus an instance of ext_Centered Kite where the length is larger than the width.

Arrow Tip Kind ext_Centered Ellipse

pre 0.6 Centered Ellipse

This is a variant of the Ellipse tip and thus another name for the ext_Centered Circle tip that is twice as wide as high.

Arrow Tip Kind ext_Centered Kite

pre 0.6 Centered Kite

A variant of the Kite tip.

The widest part will lie on the original end of the path.

Arrow Tip Kind ext_Centered Rectangle

pre 0.6 Centered Rectangle

A variant of the Rectangle tip. By default, it is twice as long as high.

Arrow Tip Kind ext_Centered Square

pre 0.6 Centered Square

A variant of the Square tip.

Arrow Tip Kind ext_Centered Stealth

pre 0.6 Centered Stealth

This is a variant of the Stealth tip.

The weighted center will lie at the original end of the path.

Arrow Tip Kind ext_Centered Triangle

pre 0.6 Centered Triangle

This is a variant of the Triangle tip and thus an instance of the ext_Centered Kite tip with zero inset.

Arrow Tip Kind ext_Centered Turned Square

pre 0.6 Centered Turned Square

This is a variant of the Turned Square tip and thus an instance of the ext_Centered Kite tip with identical width and height and mid-inset.

20.1.3 Special Arrow Tips

Arrow Tip Kind ext_Centered Rays

pre 0.6 Centered Rays

A variant of the Rays tip. The origin of the rays will lie on the original end of the path.

20.2 Untipped

20.2.1 Barbed Arrow Tips

Arrow Tip Kind ext_Centered Arc Barb

pre 0.6 Centered Arc Barb

This is a variant of the Arc Barb tip. The arrow tip will protrude half its line width over the original end of the path.

Arrow Tip Kind ext_Untipped Bar

pre 0.6 Untipped Bar

A variant of the simple Bar tip. This is a simple instance of ext_Untipped Tee Barb for length zero.

The middle of the line will lie on original end of the path.

Arrow Tip Kind ext_Untipped Bracket

pre 0.6 Untipped Bracket

This is a variant of the Bracket tip and therefore an instance of the ext_Untipped Tee Barb arrow tip that results in something resembling a bracket.

The arrow tip will protrude half its line width over the original end of the path.

Arrow Tip Kind ext_Untipped Parenthesis

pre 0.6 Untipped Parenthesis

This is a variant of the Parenthesis tip and thus an instance of the ext_Untipped Arc Barb arrow tip.

Arrow Tip Kind ext_Untipped Tee Barb

pre 0.6 Untipped Tee Barb

A variant of the Tee Barb tip.

The middle of the vertical part will lie on the original end of the path.

20.2.2 Geometric Arrow Tips

Arrow Tip Kind ext_Untipped Circle

pre 0.6 Untipped Circle

A variant of the Circle tip. This tip will protrude half its line width over the original end of the path.

Arrow Tip Kind ext_Untipped Ellipse

pre 0.6 Untipped Ellipse

This is a variant of the Ellipse tip and thus another name for the ext_Untipped Circle tip that is twice as wide as high.

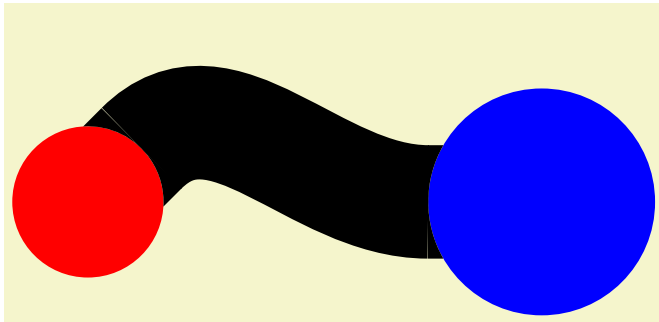
20.3 Original Arrow Tips

Arrow Tip Kind ext_Hug Cap

pre 0.6 Hug Cap

This arrow tips will hug a circle that would touch the end of the path.

Use the /pgf/arrow keys/length key to set up the radius of that circle.

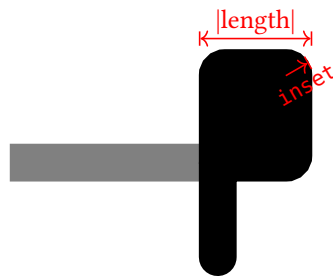


```
\usepgflibrary {ext.arrows}
\begin{tikzpicture}[
  dot/.style 2 args={
    shape=circle, outer sep=+0pt, fill={#1}, minimum size={#2}}]
\node[dot={red} {2cm}] (A) {};
\node[dot={blue}{3cm}] (B) at (6,0) {};
\draw[
  line width=1.5cm,
  arrows={ext_Hug Cap[length=1cm]-ext_Hug Cap[length=1.5cm]}
] (A) to[out=45, in=180] (B);
\end{tikzpicture}
```

Arrow Tip Kind ext_Loop

pre 0.6 Loop

This arrow tip attaches a one-sided loop to the end of the line. The length refers to the length of the whole tip while the inset specifies the radius of the three rounded corners. The width of the tip is twice the length (but can't specified independently).



Appearance of the below at line width

ext_Loop[]

ext_Loop[sep] ext_Loop[]

ext_Loop[sep] . ext_Loop[]

ext_Loop[open]

ext_Loop[open, swap]

ext_Loop[length=5pt,inset=0pt]

ext_Loop[reversed]

ext_Loop[slant=.3]

ext_Loop[red]

0.4pt

— thin

— thin

— thin

— thin

— thin

— thin

— thin

— thin

— thin

— thin

0.8pt

— thick

— thick

— thick

— thick

— thick

— thick

— thick

— thick

— thick

— thick

1.6pt

—

—

—

—

—

—

—

—

—

—

The following options have no effect: harpoon, round, line width.

On double lines, the arrow tip will not look correct.

Arrow Tip Kind ext_Double Stealth

This arrow tip is similar to the original Stealth, its back is left open so that it aligns neatly to a doubled path.

Arrow Tip Kind ext_Double Triangle

This arrow tip is similar to the original Triangle, its back is left open so that it aligns neatly to a doubled path.

Arrow Tip Kind ext_Double Cap

This arrow tip closes a doubled line so that it not left open.



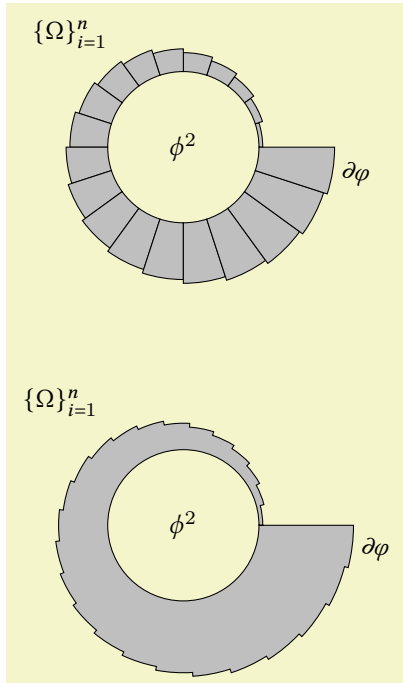
```

\usepgflibrary {ext.arrows}
\begin{tikzpicture}
\draw[
  ext_Double Cap-ext_Double Stealth,
  double distance=1cm,
  line width=3mm
] (0,0) to[bend left] (right:9);
\end{tikzpicture}

```

Part IV

Utilities



```
\usetikzlibrary {ext.misc}
\begin{tikzpicture}[
  declare function={bigR(\n)=smallR+.05*\n;},
  ext/declare constant={smallR=1; segments=20;},
  ext/full arc=segments]
\foreach \iN[evaluate={\endRadius=bigR(\iN+1);}, ext/use int=0 to segments-1]
\filldraw[fill=gray!50] (\iN R:\endRadius)
  arc [radius=\endRadius, start angle=\iN R, delta angle=+IR] -- (\iN R+1R:smallR)
  arc [radius=smallR, end angle=\iN R, delta angle=-IR] -- cycle;

\node                                {$\phi^2$};
\node at (north west:{sqrt 2 * bigR(segments/2)}) {$\{\Omega\}_{i=1}^n$};
\node[rotate=-.5R, right] at (-.5R: bigR segments) {$\partial \varphi$};

\tikzset{yshift=-.5cm, ext/declare constant={segments=25;}, ext/full arc=segments}
\filldraw[fill=gray!50] (right:smallR)
  \foreach \iN[evaluate={\endRadius=bigR(\iN+1);}, ext/use int=0 to segments-1] {
    -- (\iN R:\endRadius) arc[radius=\endRadius, start angle=\iN R, delta angle=IR]
    -- (right:smallR) arc[radius=smallR, start angle=0, delta angle=-360];

\node                                {$\phi^2$};
\node at (north west:{sqrt 2 * bigR(segments/2)}) {$\{\Omega\}_{i=1}^n$};
\node[rotate=-.5R, right] at (-.5R: bigR segments) {$\partial \varphi$};
\end{tikzpicture}
```

Part V

Changelog, Index & References

Changelog

Version 0.6.1 (2025-03-29)

- Added new tps `ext_Double Cap`, `ext_Double Stealth` and `ext_Double Triange`.
- Bugfix to `ext.arrows-plus`. [14]

Version 0.6 (2025-03-18)

- Added `\tikzextset`, `\tikzextversion` and `\tikzextversionnumber`
- Added six new auto placement mechanisms: `ext/above`, `ext/below`, `ext/west`, `ext/east`, `ext/north` and `ext/south`.
- Added `ext/auto offset` for auto placement.
- Added `ext/precise auto angle`.
- Added TikZ library `ext.arrows-plus`.
- Added TikZ library `ext.topaths.autobend`.
- Made `ext.node-families` and `ext.scalepicture` memoizable.

Version 0.5.1 (2023-04-02)

- Added PGF library `ext.arrows`.
- Bugfix to `ext.pgfkeys-plus`. [8]

Version 0.5 (2023-03-17)

- Added package `pgffor-ext`.
- Added TikZ library `ext.nodes`.
- Added TikZ library `ext.layers`.

- Bugfixes to `ext.calendar-plus`.
- Allow the original rectangle timer with `ext.paths.timer`.

Version 0.4.2 (2022-10-30)

- Added TikZ library `ext.scalepicture`.
- Bugfixes to `shapes.uncenteredrectangle`, `paths.ortho`, `positioning-plus` and `pgfcalender-ext`.

Version 0.4.1 (2022-10-23)

- Cleaned up directory structure of documentary.
- Added PGFkeys library `ext.pgfkeys-plus`.
- Added shape `uncentered rectangle` (PGF library `ext.shapes.uncenteredrectangle`).
- Fixed `ext.paths.arcto` – again [7].

Version 0.4 (2022-10-10)

- CTAN version of 0.3.1

Version 0.3.1 (2022-10-09)

- Fixed `ext.paths.ortho` keys only `vertical first` and only `horizontal first`.
- Moved all (except the `to paths`) to namespace `/tikz/ortho`. `/tikz/hvvh` and `/tikz/udlr` are considered deprecated.
- Fixed `\pgfcalendarjulianyeartoweek`.
- Added more calendar tests.

- Added directory structure.

Version 0.3 (2022-09-24)

- Added shape `circle arrow`
(PGF library `ext.shapes.circlearrow`).
- Added shape `circle cross split`
(PGF library `ext.shapes.circlecrosssplit`).
- Added shape `heatmark`
(PGF library `ext.shapes.heatmark`).
- Added shape `rectangle with rounded corners`
(PGF library `ext.shapes.rectangleroundedcorners`).
- Added shape `superellipse`
(PGF library `ext.shapes.superellipse`).
- Added TikZ library `ext.node-families.shapes.geometric`.
- Fixed `ext.node-families`' key size.
- Renamed internal macros to use custom namespace starting with `\tikzext@`.
- Added some references.

Version 0.2 (2022-08-21)

- Added TikZ library `ext.positioning-plus`.
- Added TikZ library `ext.node-families`.

Version 0.1 (2022-08-16)

- Added TikZ library `ext.calendar-plus`.
- Added TikZ library `ext.misc`.
- Added TikZ library `ext.paths.arcto`.
- Added TikZ library `ext.paths.ortho`.
- Added TikZ library `ext.paths.timer`.
- Added TikZ library `ext.patterns.images`.
- Added TikZ library `ext.topaths.arctthrough`.
- Added TikZ library `ext.transformations.mirror`.
- Added PGF library `ext.transformations.mirror`.

Index

This index contains automatically generated entries as well as [references](#) to original functionalities of `PGF/TikZ` and [references](#) to functionalities outside of `PGF/TikZ`.

- angle pos < key, [8](#)
- angle pos > key, [8](#)
- arc path operation, [9](#)
- arc arrows key, [9](#), [10](#)
- Arc Barb arrow tip, [46](#), [47](#)
- \arrow, [10](#)
- arrow shift factor key, [9](#)
- arrow shift mode key, [8](#)
- Arrow tips
 - Arc Barb, [46](#), [47](#)
 - Bar, [46](#), [47](#)
 - Bracket, [46](#), [47](#)
 - Circle, [46](#), [47](#)
 - Diamond, [46](#)
 - Ellipse, [46](#), [47](#)
 - ext_Centered Arc Barb, [46](#), [47](#)
 - ext_Centered Bar, [46](#)
 - ext_Centered Bracket, [46](#)
 - ext_Centered Circle, [46](#)
 - ext_Centered Diamond, [46](#)
 - ext_Centered Ellipse, [46](#)
 - ext_Centered Hooks, [46](#)
 - ext_Centered Kite, [46](#)
 - ext_Centered Parenthesis, [46](#)
 - ext_Centered Rays, [47](#)
 - ext_Centered Rectangle, [46](#)
 - ext_Centered Square, [46](#)
 - ext_Centered Stealth, [46](#)
 - ext_Centered Straight Barb, [46](#)
 - ext_Centered Tee Barb, [46](#)
 - ext_Centered Triangle, [46](#)
 - ext_Centered Turned Square, [47](#)
 - ext_Double Cap, [48](#)
 - ext_Double Stealth, [48](#)
 - ext_Double Triangle, [48](#)
 - ext_Hug Cap, [48](#)
 - ext_Loop, [48](#)
 - ext_Untipped Bar, [47](#)
 - ext_Untipped Bracket, [47](#)
 - ext_Untipped Circle, [47](#)
 - ext_Untipped Ellipse, [47](#)
 - ext_Untipped Parenthesis, [47](#)
 - ext_Untipped Tee Barb, [47](#)
 - Hooks, [46](#)
 - Kite, [46](#)
 - Parenthesis, [46](#), [47](#)
 - Rays, [47](#)
 - Rectangle, [46](#)
 - Square, [46](#)
 - Stealth, [46](#), [48](#)
 - Straight Barb, [46](#)
 - Tee Barb, [46](#), [47](#)
 - Triangle, [46](#), [48](#)
 - Turned Square, [47](#)
- \arrowreversed, [10](#)
- arrows.meta library, [45](#)
- Bar arrow tip, [46](#), [47](#)
- Bracket arrow tip, [46](#), [47](#)
- Circle arrow tip, [46](#), [47](#)
- circle path operation, [9](#)
- cos path operation, [9](#)
- Decorations
 - markings, [8](#)
- decorations.markings library, [10](#)
- Diamond arrow tip, [46](#)
- Ellipse arrow tip, [46](#), [47](#)
- ellipse path operation, [9](#)

- every arc arrows key, 10
- every softpath arrows key, 10
- ext.arrows library, 8, 45
- ext.arrows-plus library, 8
- ext.paths.timer library, 9
- ext/arrow pic type, 9
- ext/softpath arrow pic type, 9
- ext/softpath arrows pic type, 9
- ext_Centered Arc Barb arrow tip, 46, 47
- ext_Centered Bar arrow tip, 46
- ext_Centered Bracket arrow tip, 46
- ext_Centered Circle arrow tip, 46
- ext_Centered Diamond arrow tip, 46
- ext_Centered Ellipse arrow tip, 46
- ext_Centered Hooks arrow tip, 46
- ext_Centered Kite arrow tip, 46
- ext_Centered Parenthesis arrow tip, 46
- ext_Centered Rays arrow tip, 47
- ext_Centered Rectangle arrow tip, 46
- ext_Centered Square arrow tip, 46
- ext_Centered Stealth arrow tip, 46
- ext_Centered Straight Barb arrow tip, 46
- ext_Centered Tee Barb arrow tip, 46
- ext_Centered Triangle arrow tip, 46
- ext_Centered Turned Square arrow tip, 47
- ext_Double Cap arrow tip, 48
- ext_Double Stealth arrow tip, 48
- ext_Double Triangle arrow tip, 48
- ext_Hug Cap arrow tip, 48
- ext_Loop arrow tip, 48
- ext_Untipped Bar arrow tip, 47
- ext_Untipped Bracket arrow tip, 47
- ext_Untipped Circle arrow tip, 47
- ext_Untipped Ellipse arrow tip, 47
- ext_Untipped Parenthesis arrow tip, 47
- ext_Untipped Tee Barb arrow tip, 47
- grid path operation, 9
- Hooks arrow tip, 46
- Kite arrow tip, 46

- length key, 48
- Libraries
 - arrows.meta, 45
 - decorations.markings, 10
 - ext.arrows, 8, 45
 - ext.arrows-plus, 8
 - ext.paths.timer, 9
- markings decoration, 8
- parabola path operation, 9
- Parenthesis arrow tip, 46, 47
- Path operations
 - arc, 9
 - circle, 9
 - cos, 9
 - ellipse, 9
 - grid, 9
 - parabola, 9
 - plot, 9
 - rectangle, 9
 - sin, 9
- /pgf/
 - arrow keys/
 - length, 48
- Pic Types
 - ext/arrow, 9
 - ext/softpath arrow, 9
 - ext/softpath arrows, 9
- plot path operation, 9
- pos key, 8
- pos < key, 8
- pos > key, 8
- Rays arrow tip, 47
- Rectangle arrow tip, 46
- rectangle path operation, 9
- sin path operation, 9
- softpath arrows key, 9
- Square arrow tip, 46
- Stealth arrow tip, 46, 48

Straight Barb arrow tip, [46](#)

Tee Barb arrow tip, [46](#), [47](#)

/tikz/

pos, [8](#)

/tikz/ext/

angle pos <, [8](#)

angle pos >, [8](#)

arc arrows, [9](#), [10](#)

arrow shift factor, [9](#)

arrow shift mode, [8](#)

every arc arrows, [10](#)

every softpath arrows, [10](#)

pos <, [8](#)

pos >, [8](#)

softpath arrows, [9](#)

Triangle arrow tip, [46](#), [48](#)

Turned Square arrow tip, [47](#)

References

- [1] ‘sloped’ should consider the current transformation · Issue #1058 · pgf-tikz/pgf. URL: <https://github.com/pgf-tikz/pgf/issues/1058> (visited on 10/21/2023) (cit. on p. 9).
- [2] Christian. *TikZ arrow tip is displaced*. TeX - LaTeX Stack Exchange. Apr. 2013. URL: <https://tex.stackexchange.com/q/111051> (visited on 04/02/2023) (cit. on p. 45).
- [3] Alejandro DC. *Better fitting line to node in TiKZ*. TeX - LaTeX Stack Exchange. Apr. 2015. URL: <https://tex.stackexchange.com/q/241074> (visited on 04/01/2023) (cit. on p. 45).
- [4] Jan Hlavacek. *Modifying * and o style tikz arrows so that they are centered at the end of line*. TeX - LaTeX Stack Exchange. Feb. 2011. URL: <https://tex.stackexchange.com/q/11871> (visited on 04/02/2023) (cit. on p. 45).
- [5] Edgar A. Bering IV. *Set the color of a tikz-cd Glyph arrow tip with xelatex*. TeX - LaTeX Stack Exchange. Oct. 2020. URL: <https://tex.stackexchange.com/q/565010> (visited on 04/01/2023) (cit. on p. 45).
- [6] lucky1928. *add customize line header not smooth connected*. TeX - LaTeX Stack Exchange. ZSCC: NoCitationData[s0]. July 2024. URL: <https://tex.stackexchange.com/q/722350> (visited on 03/28/2025) (cit. on p. 45).
- [7] marmotghost. *clockwise/counter clockwise does not seem to work reliably*. Oct. 2022. URL: <https://github.com/Qrrbrbirlbel/tikz-extensions/issues/2> (visited on 10/23/2022) (cit. on p. 81).
- [8] marmotghost. *Latest version of ext.misc on CTAN appears to have a typo*. Mar. 2023. URL: <https://github.com/Qrrbrbirlbel/tikz-extensions/issues/6> (visited on 04/01/2023) (cit. on p. 81).
- [9] Qrrbrbirlbel. *Answer to “add customize line header not smooth connected”*. TeX - LaTeX Stack Exchange. ZSCC: NoCitationData[s0]. July 2024. URL: <https://tex.stackexchange.com/a/722413> (visited on 03/28/2025) (cit. on p. 45).
- [10] Qrrbrbirlbel. *Answer to “Better fitting line to node in TiKZ”*. TeX - LaTeX Stack Exchange. Apr. 2015. URL: <https://tex.stackexchange.com/a/241303> (visited on 04/01/2023) (cit. on p. 45).
- [11] Qrrbrbirlbel. *Answer to “Modifying * and o style tikz arrows so that they are centered at the end of line”*. TeX - LaTeX Stack Exchange. Sept. 2022. URL: <https://tex.stackexchange.com/a/656241> (visited on 04/02/2023) (cit. on p. 45).
- [12] Qrrbrbirlbel. *Answer to “Set the color of a tikz-cd Glyph arrow tip with xelatex”*. TeX - LaTeX Stack Exchange. Apr. 2023. URL: <https://tex.stackexchange.com/a/681474> (visited on 04/01/2023) (cit. on p. 45).
- [13] Qrrbrbirlbel. *Answer to “TikZ arrow tip is displaced”*. TeX - LaTeX Stack Exchange. Apr. 2013. URL: <https://tex.stackexchange.com/a/111053> (visited on 04/02/2023) (cit. on p. 45).
- [14] Qrrbrbirlbel. *arrow shifting for start tip sequence doesn’t work and shift amount needs to be expanded*. ZSCC: NoCitationData[s0] original-date: 2022-08-13T23:42:55Z. Mar. 2025. URL: <https://github.com/Qrrbrbirlbel/tikz-extensions> (visited on 03/29/2025) (cit. on p. 81).