

# The `overarrows` package\*

Julien Labbé  
Julien.Labbe@univ-grenoble-alpes.fr

July 11, 2024

## Abstract

A L<sup>A</sup>T<sub>E</sub>X package to create custom arrows over math expressions, mainly for vectors (but arrows can as well be drawn below). Arrows stretch with content, scale with math styles, and have a correct kerning when a subscript follows.

Short example:

```
\NewOverArrowCommand{\overrightarrow}{%  
  end=\rightharpoonup  
}  
  
\begin{align*}  
  &\overrightarrow{v} \quad \overrightarrow{v}_{\text{subscript}} \quad \overrightarrow{ABCD} \quad \overrightarrow{v}_{\text{subscript}} \\  
&\overrightarrow{ABCD} \quad \overrightarrow{v}_{\text{subscript}}  
\end{align*}
```

$$\begin{array}{cc} \overrightarrow{v} & \overrightarrow{v}_{\text{subscript}} \\ \overrightarrow{ABCD} & \overrightarrow{v}_{\text{subscript}} \end{array}$$

Predefined commands are also provided:

- to typeset vectors:

$$\vec{v} \quad \overrightarrow{AB},$$

- to draw arrows of various shapes above math expressions:

$$\overrightarrow{AB} \quad \overleftarrow{AB} \quad \overleftrightarrow{AB} \quad \overrightarrow{AB} \quad \overleftarrow{AB} \quad \overrightarrow{AB} \quad \overleftarrow{AB} \quad \overrightarrow{AB},$$

- to draw arrows of various shapes under math expressions:

$$\underline{\overrightarrow{AB}} \quad \underline{\overleftarrow{AB}} \quad \underline{\overleftrightarrow{AB}} \quad \underline{\overrightarrow{AB}} \quad \underline{\overleftarrow{AB}} \quad \underline{\overrightarrow{AB}} \quad \underline{\overleftarrow{AB}} \quad \underline{\overrightarrow{AB}}.$$

---

\*This document corresponds to `overarrows` v1.2, dated 2024/07/10.

# Contents

<b>1</b>	<b>Presentation of the package</b>	<b>4</b>
<b>2</b>	<b>Introduction</b>	<b>4</b>
2.1	Vector arrows . . . . .	4
2.2	Stack and arrow macros . . . . .	5
2.3	Extensible arrows . . . . .	5
<b>3</b>	<b>Quick start</b>	<b>6</b>
3.1	Loading the package <code>overarrows</code> . . . . .	6
3.2	Commands creation . . . . .	6
3.3	Start and end of the arrow . . . . .	6
3.4	Size and position of the arrow . . . . .	8
3.5	Symbols assemblage . . . . .	9
3.6	Drawing the arrow with TikZ . . . . .	11
3.7	Drawing the arrow with PSTricks . . . . .	12
3.8	Drawing the arrow with L <sup>A</sup> T <sub>E</sub> X picture environment . . . . .	12
<b>4</b>	<b>User interface</b>	<b>13</b>
4.1	Package options . . . . .	13
4.1.1	<code>esvect</code> configuration . . . . .	13
4.1.2	Predefined commands . . . . .	14
4.1.3	Other options . . . . .	16
4.2	Commands . . . . .	17
4.2.1	Macro for commands creation . . . . .	17
4.2.2	Useful macros for symbols assemblage . . . . .	19
4.2.3	Useful lengths for TikZ, PSTricks or <code>picture</code> environment . . . . .	19
4.2.4	Vectors macros . . . . .	20
4.2.5	Predefined commands . . . . .	21
4.3	Keys . . . . .	22
4.3.1	Arrow position and length settings . . . . .	22
4.3.2	Subscripts detection setting . . . . .	25
4.3.3	Symbols assemblage settings . . . . .	25
4.3.4	TikZ settings . . . . .	27
4.3.5	PSTricks settings . . . . .	28
4.3.6	Picture environment settings . . . . .	29
4.4	Advanced commands and keys . . . . .	30
4.4.1	Advanced commands . . . . .	30
4.4.2	Advanced keys . . . . .	30
<b>5</b>	<b>Complements</b>	<b>31</b>
5.1	Know issues . . . . .	31
5.1.1	Math font change . . . . .	31
5.1.2	Detection of non standard subscripts . . . . .	31
5.2	Package dependencies . . . . .	32
5.3	Alternatives . . . . .	32
5.4	Changelog . . . . .	33
<b>6</b>	<b>Implementation</b>	<b>33</b>



# 1 Presentation of the package

The `overarrows` package allows to create commands for drawing arrows over math expressions. These arrows:

- are fully customisable, at command definition, through a key-value interface;
- stretch with the content and can cover many characters, like in  $\overrightarrow{AB}$ ;
- scale with math styles<sup>1</sup>, like in  $\overrightarrow{v}_{\vec{u}\vec{v}}$ .

Commands created with the `overarrows` package are provided with a starred variant, that removes the extra end space generated by the arrow. This is particularly useful when the command is followed by a subscript. For example, the velocity of the center of mass can be written with exactly the same kerning when scalar  $v_{\text{cm}}$  or vector  $\vec{v}_{\text{cm}}$  (no extra space before the subscript, unlike the output of the unstarred variant:  $\vec{v}_{\text{cm}}$ ).

The `overarrows` package was primitively written for vectors, but in a highly customisable way. It can be used to define a large variety of arrows, using math symbols, or drawing commands from PGF/TikZ or PSTricks. It's also possible to create commands that draw the arrows under. Some predefined commands are provided, giving<sup>2</sup>, for arrow over:

$$\overrightarrow{\alpha + \beta} \quad \overleftarrow{\alpha + \beta} \quad \overleftrightarrow{\alpha + \beta} \quad \overrightarrow{\alpha + \beta} \quad \overleftarrow{\alpha + \beta} \quad \overrightarrow{\alpha + \beta} \quad \overleftarrow{\alpha + \beta} \quad \overrightarrow{\alpha + \beta}$$

and for arrow under :

$$\underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta}$$

## 2 Introduction

### 2.1 Vector arrows

Vectors are commonly typeset in bold face, or with an arrow above<sup>3</sup>. For this second convention,  $\text{\TeX/L\TeX}$  provides the command `\vec`, which accents its content (using the `\mathaccent` command) with the character  $\vec{\phantom{x}}$  (`\mathchar"017E` in Computer Modern font). But  $\vec{\phantom{x}}$  isn't extensible<sup>4</sup>, and gives:  $\vec{v}$ ,  $\vec{AB}$  or  $\vec{\text{grad}}$  (there's no command `\widevec` analogous to `\widehat`).

An extensible alternative is given by the command `\overrightarrow`, available in  $\text{\TeX/L\TeX}$ , and which is redefined by the commonly used `amsmath` package. But its arrow, built with the `\rightarrow` symbol  $\rightarrow$ , is too large, using the default *Computer Modern* font:  $\overrightarrow{AB}$ . Another alternative is the `esvect` package, which provides the `\vv` command and a set of custom arrows:  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{AB}$ .

<sup>1</sup>`\displaystyle`, `\textstyle`, `\scriptstyle` and `\scriptscriptstyle`.

<sup>2</sup>Displayed here with the `old-arrows`<sup>P.16</sup> option.

<sup>3</sup>See, for example: International Organization for Standardization. (2019). *Quantities and units – Part 2: Mathematics* (ISO Standard No. 80000-2:2019). <https://www.iso.org/standard/64973.html>.

<sup>4</sup>In fact, with the unicode engines  $\text{\LuaTeX}$  and  $\text{\XeTeX}$ , the command `\Umathaccent` can now define extensible accents. This is used by the `unicode-math` package, which also set the arrows displayed by `\vec` and `\overrightarrow` in a coherent manner.

## 2.2 Stack and arrow macros

It's worth looking at the definition of `amsmath \overrightarrow` command:

```
\long macro:->\mathpalette {\overarrow@ \rightarrowfill@ }
```

Three macros are used here:

`\mathpalette` adapts the output to the current math style;

`\overarrow@` is the *stack macro*, that puts the arrow above the content;

`\rightarrowfill@` is the *arrow macro*, that holds the content of the arrow.

The command `\vv` from `esvec` is defined with a very similar way, using its own stack macro (`\overvect@`) and arrow macro (`\vectfill@`).

The `overarrows` package uses the same mechanism. Arrow and stack macros are set, at command creation, through a key-value interface provided by the `pgfkeys` package (after creation, however, the command definition is static and the key-value interface is not used).

## 2.3 Extensible arrows

Arrows drawn by the commands `\overrightarrow` or `\vv` are built by joining math symbols, and made extensible by repetition of the central symbol<sup>5</sup>. Thus, the line of the macro `\overrightarrow` is made by repetition of command `\relbar` — (which simply corresponds to the minus sign), while `\vv` use its own command `\relbareda` -.

This method may generate some undesirable spacing issues, when symbols badly overlap. See, for example, the output of `amsmath \overrightarrow` (left) and `esvect \vv` (right) in `\scriptscriptstyle` math style (scaled by a factor 4):

$\overrightarrow{\text{long vector}}$        $\overrightarrow{\text{long vector}}$

While the arrow on the left lets guess where the symbols — overlap, the arrow on the right present unwanted spaces and show clearly its composition as association of the symbols —, - and →.

By default, the `overarrows` package uses the same mechanism to extend arrows according to their contents. Settings and tools are provided to perform fine tuning and avoid spacing issues. As example, see below the `\overrightarrow` and `\vv` commands, as redefined by `overarrows` (in `\scriptscriptstyle` and scaled by a factor 4):

$\overrightarrow{\text{long vector}}$        $\overrightarrow{\text{long vector}}$

The `overarrows` package also provides an alternative mechanism. When used, the length `\overarrowlength` is set, according to the arrow command content, and can be employed, for example, to draw arrows using PGF/TikZ, PSTricks or the  $\LaTeX$  picture environment.

<sup>5</sup>Using the  $\TeX$  `\cleaders` command.

## 3 Quick start

### 3.1 Loading the package `overarrows`

To load the `overarrows`, simply add in preamble, before the “`\begin{document}`”:

```
\usepackage{overarrows}
```

Options can be given, in a comma-separated list. For example, to use the predefined commands shown in the section 1, page 4, write:

```
\usepackage[allcommands, old-arrows]{overarrows}
```

This define the commands (described in section 4.2.5, page 21):

- |   |   |
|---|---|
| • <code>\overrightarrow</code> <sup>→P. 21</sup>            | • <code>\underrightarrow</code> <sup>→P. 22</sup>           |
| • <code>\overleftarrow</code> <sup>→P. 21</sup>             | • <code>\underleftarrow</code> <sup>→P. 22</sup>            |
| • <code>\overleftrightharpoonup</code> <sup>→P. 21</sup>    | • <code>\underleftrightharpoonup</code> <sup>→P. 22</sup>   |
| • <code>\overrightarrowharpoonup</code> <sup>→P. 21</sup>   | • <code>\underrightharpoonup</code> <sup>→P. 22</sup>       |
| • <code>\overrightarrowharpoondown</code> <sup>→P. 21</sup> | • <code>\underrightharpoondown</code> <sup>→P. 22</sup>     |
| • <code>\overleftarrowharpoonup</code> <sup>→P. 21</sup>    | • <code>\underleftarrowharpoonup</code> <sup>→P. 22</sup>   |
| • <code>\overleftarrowharpoondown</code> <sup>→P. 21</sup>  | • <code>\underleftarrowharpoondown</code> <sup>→P. 22</sup> |
| • <code>\overbar</code> <sup>→P. 21</sup>                   | • <code>\underbar</code> <sup>→P. 22</sup>                  |

Note that the `old-arrows`<sup>→P. 16</sup> option may give bad results, if math fonts have been changed. Simply remove the option in this case.

Many other options are available. See the complete list, page 13.

### 3.2 Commands creation

Commands are created with `\NewOverArrowCommand`<sup>→P. 17</sup>. This macro take two mandatory arguments : the name of the command and the arrow configuration as comma-separated list of key-values. By default, a right arrow is set:

```
\NewOverArrowCommand{\myovercmd}{  
$ \myovercmd{test}$
```



$$\overrightarrow{test}$$

Commands are defined with a starred variant, designed to handle subscripts:

```
$ v_{sub} \quad \backslashqqquad \backslashmyovercmd{v}_{sub} \quad \backslashqqquad \backslashmyovercmd*{v}_{sub} $
```

$$v_{sub} \quad \overrightarrow{v}_{sub} \quad \overrightarrow{v}_{sub}$$

### 3.3 Start and end of the arrow

Extremities of the arrow are set by the keys `start`<sup>→P. 25</sup> and `end`<sup>→P. 25</sup>. For example, an arrow starting with a hook (symbols `\hook` ) and ending with two heads (symbol `\twoheadrightarrow` ) is defined by:

```
\NewOverArrowCommand{\overhooktwoheadrightarrow}{%
  start=\lhook, end=\twoheadrightarrow,
}
```

Note that `\twoheadrightarrow` must be defined, as it is not in  $\text{\LaTeX}$ . This can be done with the package `amssymb`, by adding in preamble:

```
\usepackage{amssymb}
```

But with the previous definition, the result of the command `\overhooktwoheadrightarrow` is faulty:

```
$ \overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB} $
```

$$\overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB}$$

The problem comes from symbols junction and the trimming used to obtain their overlap. It can be solved with the keys `trim start`<sup>P.25</sup> and `trim end`<sup>P.26</sup>, which are numbers and set the corresponding trimming in math units (typically 1/18 em). Appropriate values gives better results:

```
\NewOverArrowCommand{\overhooktwoheadrightarrow}{%
  start=\lhook, end=\twoheadrightarrow,
  trim start=1.5, trim end=2,
}
$ \overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB} $
```

$$\overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB}$$

If the math font differs from the default *Computer Modern*, the central part of the arrow may have inappropriate position or line width. This is because the default symbol used for the arrow line is `\relbareda` from the `esvect` package<sup>6</sup>. If needed, try to set the `middle`<sup>P.25</sup> key with the symbol `\relbar`. The trimming should also be adapted:

```
\NewOverArrowCommand{\overhooktwoheadrightarrow}{%
  start=\lhook, end=\twoheadrightarrow, middle=\relbar,
  trim start=0, trim end=3, trim middle=5,
}
$ \overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB} $
```

$$\overhooktwoheadrightarrow{v} \quad \overhooktwoheadrightarrow{AB}$$

Finding the correct values for `trim start`<sup>P.25</sup>, `trim end`<sup>P.26</sup> and `trim middle`<sup>P.25</sup> may need many trials. For this purpose, the macro `\TestOverArrow`<sup>P.18</sup> displays the result of a command for different lengths and math styles:

<sup>6</sup>Except if the `unicode-math` package is used with a math font that provides the `\harrowextender` symbol (see the `middle config=auto` key).

`\TestOverArrow{\overhooktwoheadrightarrow}`

<code>\displaystyle</code>	<code>\textstyle</code>	<code>\scriptstyle</code>	<code>\scriptscriptstyle</code>
$\overrightarrow{v}$	$\overrightarrow{v}$	$\overrightarrow{v}$	$\overrightarrow{v}$
$\overrightarrow{AB}$	$\overrightarrow{AB}$	$\overrightarrow{AB}$	$\overrightarrow{AB}$
$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$
$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$

### 3.4 Size and position of the arrow

A command `\OverRightarrow`, built with the symbols `\Relbar`  $\Rightarrow$  and `\Rightarrow`  $\Rightarrow$ , gives:

```
\NewOverArrowCommand{\OverRightarrow}{%
  start=\Relbar,
  middle=\Relbar,
  end=\Rightarrow,
  trim=4,
}
$ \OverRightarrow{v} \quad \OverRightarrow{AB} $
```

$$\overrightarrow{v} \quad \overrightarrow{AB}$$

The key `trim`<sup>P. 26</sup> sets `trim start`<sup>P. 25</sup>, `trim middle`<sup>P. 25</sup> and `trim end`<sup>P. 26</sup> with the same value.

The previous arrow is visually too big. The macro `\smallermathstyle`<sup>P. 19</sup> allows to obtain a better result:

```
\NewOverArrowCommand{\OverRightarrow}{%
  start={\smallermathstyle\Relbar},
  middle={\smallermathstyle\Relbar},
  end=\Rightarrow,
  trim=4,
}
$ \OverRightarrow{v} \quad \OverRightarrow{AB} $
```

$$\overrightarrow{v} \quad \overrightarrow{AB}$$

Note that `\smallermathstyle`<sup>P. 19</sup> should not be used for `end`<sup>P. 25</sup>, because this last is formatted with the same math style as `start`<sup>P. 25</sup>.

It would be better to add an extra space between the arrow and the content of the command. This can be done with the key `space after arrow`<sup>P. 24</sup>:

```
\NewOverArrowCommand{\OverRightarrow}{%
  start={\smallermathstyle\Relbar},
  middle={\smallermathstyle\Relbar},
  end=\Rightarrow,
  trim=4,
  space after arrow=0.25ex,
}
$ \OverRightarrow{v} \quad \OverRightarrow{AB} $
```

$$\overrightarrow{v} \quad \overrightarrow{AB}$$



Default arrows are slightly shifted to the right. For a left arrow, this should be reversed, using the keys `shift left`<sup>→P.23</sup> and `shift right`<sup>→P.23</sup>. These keys set the corresponding shifts, in math units. Example:

```
\NewOverArrowCommand{\OverLeftarrow}{%
  start={\smallermathstyle\Leftarrow},
  middle={\smallermathstyle\Relbar},
  end=\Relbar,
  trim=4,
  space after arrow=0.25ex,
  shift left=0, shift right=2,
}
$ \OverLeftarrow{v} \quad \OverLeftarrow{AB} $
```




Finally, the key `arrow under`<sup>→P.23</sup> places the arrow below the content, instead of above (and `space before arrow`<sup>→P.24</sup> sets the space upon it):

```
\NewOverArrowCommand{\UnderLeftRightarrow}{%
  start={\smallermathstyle\Leftarrow},
  middle={\smallermathstyle\Relbar},
  end=\Rightarrow,
  trim=4,
  arrow under,
  space before arrow=0.5ex,
  shift left=0, shift right=0,
}
$ \UnderLeftRightarrow{v} \quad \UnderLeftRightarrow{AB} $
```



### 3.5 Symbols assemblage




Many  $\text{\LaTeX}$  math symbols are built by assemblage, using the macro `\joinrel`<sup>7</sup> which remove 3 math units of horizontal space. The `overarrows` package provides a flexible version of `\joinrel`, called `\xjoinrel`<sup>→P.19</sup>, which remove an arbitrary number of math units, given as optional argument.

Symbols association is then simple. As example, one can define a triple tail macro `\tttail` from the symbol `\succ` :

```
\newcommand*{\tttail}{\succ\xjoinrel[10]\succ\xjoinrel[10]\succ}
$ \tttail $
```



Thus defined, the macro `\tttail` can be used in arrow definition:

<sup>7</sup>For example, the symbol `\models`  is defined as `\mathrel{|\}\joinrel\Relbar` and corresponds to the assemblage of a vertical line  and the symbol `\Relbar` . The command `\mathrel` modifies the spacing according to the math relation class ; `\Relbar` corresponds to the equal sign (it's definition is `\mathrel{=}`).

```

\NewOverArrowCommand{\overttailrightarrow}{%
  start={\tttail},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overttailrightarrow{v} \quad \overttailrightarrow{AB} $

```



Here the `min length`<sup>P. 23</sup> key was added to ensure a minimum length (in math units) when the content of the command is small (as for a single character).

The previous arrow would be better with a smaller tail, and this can be done with the macro `\smallermathstyle`<sup>P. 19</sup>. But a small tail and a normal sized head are not aligned; as `{\smallermathstyle\tttail}\xjoinrel[8]\rightarrow` gives:



The solution comes from the command `\vcenter` which centers materials on math axis. The tail must then be wrapped in a `\hbox`:

```

\NewOverArrowCommand{\overttailrightarrow}{%
  start={\vcenter{\hbox{$\smallermathstyle\tttail$}}},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overttailrightarrow{v} \quad \overttailrightarrow{AB} $

```

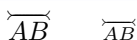


Text symbols, namely symbols that are not defined in math mode, can also be used. They should yet be enclosed in the `\text` macro, from the `amsmath` package, to be correctly displayed and correctly scaled according to math style. With, for example, the arrow heads given by the symbols 40 and 41 of the *lasy* font:

```

\newcommand*{\leftarrowhead}{\usefont{U}{lasy}{m}{n}\symbol{40}}
\newcommand*{\rightarrowhead}{\usefont{U}{lasy}{m}{n}\symbol{41}}
\NewOverArrowCommand{\overrightarrow}{%
  start=\text{\rightarrowhead},
  end=\text{\leftarrowhead},
  trim start=0.7, trim end=0.7,
  min length=20,
  shift leftright=-2,
}
$ \overrightarrow{AB} \quad \scriptstyle\overrightarrow{AB} $

```



### 3.6 Drawing the arrow with TikZ

In addition to the default method presented previously (assemblage of symbols, as described in section 2.3, page 5), the `overarrows` package has an alternative method to draw the arrow. This one allows the use of graphic languages such as PGF/TikZ.

Drawing arrows with TikZ requires to load the `tikz` package and its library `arrows.meta`. This can be simply done by passing the `tikz`<sup>→P.16</sup> option to the `overarrows` package<sup>8</sup>:

```
\usepackage[tikz]{overarrows}
```

To use PGF/TikZ language, the optional argument `tikz` must be passed to `\NewOverArrowCommand`<sup>→P.17</sup>. TikZ pictures are not extensible. That's why the `overarrows` package provides three lengths that can be used in TikZ commands:

- `\overarrowlength`<sup>→P.20</sup> for the arrow length,
- `\overarrowthickness`<sup>→P.20</sup> and `\overarrowsmallerthickness`<sup>→P.20</sup> for the arrow thickness.

These lengths are computed at each utilisation of a command created with the `tikz` optional argument.

Without any other configuration, a right arrow is drawn:

```
\NewOverArrowCommand[tikz]{\overtikzarrow}{%
$ \overtikzarrow{v} \quad \overtikzarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

Keys to use Tikz are described in section 4.3.4, page 27. Main keys are: `tikz options`<sup>→P.27</sup>, `path options`<sup>→P.27</sup> and `path`<sup>→P.27</sup>. It's also possible to append settings with `add tikz options`<sup>→P.27</sup> and `add path options`<sup>→P.27</sup>. The full TikZ command used to draw the arrow can as well be entirely redefined with the key `tikz command`<sup>→P.28</sup>

Here is a example of an arrow drawn with TikZ<sup>9</sup>:

```
\NewOverArrowCommand[tikz]{\overarchedleftrightharrow}{%
add tikz options={y=\overarrowlength},
add tikz options={line width={\overarrowsmallerthickness}},
path options={arrows={<[scale=0.5]->[scale=0.5]}},
path={(0,0) arc (-250:70:0.5 and 0.1)},
center arrow,
min length=25,
space after arrow=0.4ex,
}
$ \overarchedleftrightharrow{v} \quad \overarchedleftrightharrow{ABCD} $
```

$\overleftrightarrow{v}$        $\overleftrightarrow{ABCD}$

<sup>8</sup>Note that the `tikz`<sup>→P.16</sup> option isn't mandatory to use TikZ commands in `overarrows`. The `tikz` package and its library `arrows.meta` can be loaded independently.

<sup>9</sup>TikZ arrows are very powerfull, but much slower to draw than the default method using assemblage of math symbols.

### 3.7 Drawing the arrow with PSTricks

In addition to PGF/TikZ, the arrow can be drawn with PSTricks macros. For this, the optional argument `pstricks` must be passed to `\NewOverArrowCommand`<sup>→P.17</sup>. Like with `tikz`, the three lengths `\overarrowlength`<sup>→P.20</sup>, `\overarrowthickness`<sup>→P.20</sup> and `\overarrowsmallerthickness`<sup>→P.20</sup> can be used in PSTricks commands. By default, a right arrow is drawn:

```
\NewOverArrowCommand[pstricks]{\overpstarrow}{}
$ \overpstarrow{v} \quad \overpstarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

The `pstricks` package has to be loaded (for example, using the `pstricks`<sup>→P.16</sup> option of the `overarrows` package)

Keys to use PSTricks commands are described in section 4.3.5, page 28. The main keys are `pstricks command`<sup>→P.28</sup>, `psset`<sup>→P.28</sup>, `arrow`<sup>→P.28</sup>, `geometry`<sup>→P.28</sup> and `line thickness`<sup>→P.29</sup>. Examples:

```
\NewOverArrowCommand[pstricks]{\overreddisks}{%
  psset={linecolor=red}, arrow=***, center arrow,
}
$ \overreddisks{v} \quad \overreddisks{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

```
\NewOverArrowCommand[pstricks]{\ellipticarrow}{%
  pstricks command={%
    \psellipticarcn{->}%^A avoid space before coordinates
    (0.5\overarrowlength,0.2\overarrowlength)^%^A avoid space before coordinates
    (0.5\overarrowlength,0.2\overarrowlength)
    {170}{10}
  },
  geometry={(0,0.2\overarrowlength)(\overarrowlength,0.4\overarrowlength)},
  line thickness={\overarrowsmallerthickness},
  center arrow,
}
$ \ellipticarrow{v} \quad \ellipticarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

### 3.8 Drawing the arrow with L<sup>A</sup>T<sub>E</sub>X picture environment

Without any other package, arrows can also be drawn with the L<sup>A</sup>T<sub>E</sub>X `picture` environment. In this case, the optional argument `picture` must be passed to `\NewOverArrowCommand`<sup>→P.17</sup>. As with `tikz` or `pstricks`, the three lengths `\overarrowlength`<sup>→P.20</sup>, `\overarrowthickness`<sup>→P.20</sup> and `\overarrowsmallerthickness`<sup>→P.20</sup> are available and can be used in `picture` drawing commands. By default, a right vector is drawn:

```
\NewOverArrowCommand[picture]{\overpictarrow}{}
$ \overpictarrow{v} \quad \overpictarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

If `overarrows` is loaded with the option `pstarrows`<sup>→P.17</sup>, the package `pict2e` is used and a PSTricks style vector arrows is set. This gives:

```
\NewOverArrowCommand[picture]{\overpictarrow}{  
$ \overpictarrow{v} \quad \overpictarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

Keys to use L<sup>A</sup>T<sub>E</sub>X picture environment are described in section 4.3.6, page 29. The main keys are `picture` command<sup>→P.29</sup>, `geometry`<sup>→P.29</sup> and `line thickness`<sup>→P.29</sup>. Here is an example:

```
\NewOverArrowCommand[picture]{\overbandedarrow}{  
  picture command={%  
    \qbezier  
    (0.0\overarrowlength,0)  
    (0.5\overarrowlength,0)  
    (0.9\overarrowlength,0.2\overarrowlength)  
    \put(0.9\overarrowlength,0.2\overarrowlength)  
    {\vector(2,1){0.2\overarrowlength}}  
  },  
  geometry={(\overarrowlength,0.4\overarrowlength)(0,0)},  
  line thickness={\overarrowsmallerthickness},  
  center arrow,  
  space after arrow=0.4ex,  
}  
$ \overbandedarrow{v} \quad \overbandedarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

## 4 User interface

### 4.1 Package options



The `overarrows` package accepts many options, given as a comma-separated list  $\langle options \rangle$  at package loading: `\usepackage[ $\langle options \rangle$ ]{overarrows}`.









The option `esvect` is set by default. This can be overridden with `noesvect`.

#### 4.1.1 esvect configuration

##### esvect

Loads the `esvect` package and redefines its vector commands  $\backslash vv$ <sup>→P.20</sup> through the `overarrows` mechanism. Original `esvect`  $\backslash vv$  macro is still available with  $\backslash esvect vv$ <sup>→P.20</sup>. The `esvect` font description is fixed to allow any font sizes.

The `esvect` package provides the symbol  $\backslash relbareda$   which is smaller and often more flexible than the classic one  $\backslash relbar$  .  $\backslash relbareda$  fits with the standard *Computer Modern* math font, but can be unsuitable with other fonts.

The `esvect` package also provides the right arrow command  $\backslash fldr$ . The shape of the arrow depends on the option passed to the `esvect` package:  (option a),  (option b),  (option c),  (option d),  (option e),  (option f),  (option g) or  (option h). Note that by default `overarrows` loads the `esvect` package with the option f (while `esvect` default is d). This can be changed with one of the eight options described below: `esvecta`, `esvectb`, `esvectc`, `esvectd`, `esvecte`, `esvectf`, `esvectg` and `esvecth`.

This option is set by default and can be unset with `noesvect`.

#### **noesvect**

Prevents the loading of the `esvect` package and the definition of the command  $\backslash vv \rightarrow$  P. 20.

#### **esvecta**

Loads the `esvect` package with the `a` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvectb**

Loads the `esvect` package with the `b` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvectc**

Loads the `esvect` package with the `c` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvectd**

Loads the `esvect` package with the `d` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvecte**

Loads the `esvect` package with the `e` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvectf**

Loads the `esvect` package with the `f` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvectg**

Loads the `esvect` package with the `g` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

#### **esvecth**

Loads the `esvect` package with the `h` option.

$\backslash fldr$  corresponds the to the symbol  $\rightarrow$ .  $\backslash vv$  command gives :  $\vec{v}$   $\overrightarrow{AB}$   $\overrightarrow{\text{grad}}$ .

### **4.1.2 Predefined commands**

The `overarrows` package provides sixteen predefined commands, eight with the arrow over, and eight with the arrow under. By default, theses commands are not defined, and must be activated by the corresponding option. Beware that commands are created without checking if already defined by another package ( $\overleftarrow{\phantom{x}}$ ,  $\overrightarrow{\phantom{x}}$ ,  $\overleftrightarrow{\phantom{x}}$ ,  $\underleftarrow{\phantom{x}}$ ,  $\underrightarrow{\phantom{x}}$  and  $\underleftrightarrow{\phantom{x}}$  are, for example, part of the `ams-math` package).

Three options are also available to define set of commands.

#### **Set of commands**

### allcommands

Defines all sixteen predefined commands.

### overcommands

Defines all eight predefined commands with arrow over.

### undercommands

Defines all eight predefined commands with arrow under.

## Over arrows

### overrightarrow

Defines the `\overrightarrow`<sup>P.21</sup> command:  $\vec{v}$ ,  $\overrightarrow{AB}$ ,  $\overrightarrow{\text{grad}}$ .

### overleftarrow

Defines the `\overleftarrow`<sup>P.21</sup> command:  $\overleftarrow{v}$ ,  $\overleftarrow{AB}$ ,  $\overleftarrow{\text{grad}}$ .

### overleftrightarrow

Defines the `\overleftrightarrow`<sup>P.21</sup> command:  $\overleftrightarrow{v}$ ,  $\overleftrightarrow{AB}$ ,  $\overleftrightarrow{\text{grad}}$ .

### overrightarrowtharpoonup

Defines the `\overrightarrowtharpoonup`<sup>P.21</sup> command:  $\overrightarrow{\curvearrowright} v$ ,  $\overrightarrow{\curvearrowright} AB$ ,  $\overrightarrow{\curvearrowright} \text{grad}$ .

### overrightarrowtharpoondown

Defines the `\overrightarrowtharpoondown`<sup>P.21</sup> command:  $\overrightarrow{\curvearrowleft} v$ ,  $\overrightarrow{\curvearrowleft} AB$ ,  $\overrightarrow{\curvearrowleft} \text{grad}$ .

### overlefttharpoonup

Defines the `\overlefttharpoonup`<sup>P.21</sup> command:  $\overleftarrow{\curvearrowright} v$ ,  $\overleftarrow{\curvearrowright} AB$ ,  $\overleftarrow{\curvearrowright} \text{grad}$ .

### overlefttharpoondown

Defines the `\overlefttharpoondown`<sup>P.21</sup> command:  $\overleftarrow{\curvearrowleft} v$ ,  $\overleftarrow{\curvearrowleft} AB$ ,  $\overleftarrow{\curvearrowleft} \text{grad}$ .

### overbar

Defines the `\overbar`<sup>P.21</sup> command:  $\overline{v}$ ,  $\overline{AB}$ ,  $\overline{\text{grad}}$ .

## Under arrows

### underrightarrow

Defines the `\underrightarrow`<sup>P.22</sup> command:  $\underline{\rightarrow} v$ ,  $\underline{\rightarrow} AB$ ,  $\underline{\rightarrow} \text{grad}$ .

### underleftarrow

Defines the `\underleftarrow`<sup>P.22</sup> command:  $\underline{\leftarrow} v$ ,  $\underline{\leftarrow} AB$ ,  $\underline{\leftarrow} \text{grad}$ .

#### **underleftrightharrow**

Defines the `\underleftrightharrow`<sup>→P.22</sup> command:  $\overleftrightarrow{v}$ ,  $\overleftrightarrow{AB}$ ,  $\overleftrightarrow{\text{grad}}$ .

#### **underrightharpoonup**

Defines the `\underrightharpoonup`<sup>→P.22</sup> command:  $\underset{\curvearrowright}{v}$ ,  $\underset{\curvearrowright}{AB}$ ,  $\underset{\curvearrowright}{\text{grad}}$ .

#### **underrightharpoondown**

Defines the `\underrightharpoondown`<sup>→P.22</sup> command:  $\underset{\curvearrowleft}{v}$ ,  $\underset{\curvearrowleft}{AB}$ ,  $\underset{\curvearrowleft}{\text{grad}}$ .

#### **underleftharpoonup**

Defines the `\underleftharpoonup`<sup>→P.22</sup> command:  $\overleftarrow{\underset{\curvearrowright}{v}}$ ,  $\overleftarrow{\underset{\curvearrowright}{AB}}$ ,  $\overleftarrow{\underset{\curvearrowright}{\text{grad}}}$ .

#### **underleftharpoondown**

Defines the `\underleftharpoondown`<sup>→P.22</sup> command:  $\overleftarrow{\underset{\curvearrowleft}{v}}$ ,  $\overleftarrow{\underset{\curvearrowleft}{AB}}$ ,  $\overleftarrow{\underset{\curvearrowleft}{\text{grad}}}$ .

#### **underbar**

Defines the `\underbar`<sup>→P.22</sup> command:  $\underline{v}$ ,  $\underline{AB}$ ,  $\underline{\text{grad}}$ .

### **4.1.3 Other options**

#### **old-arrows**

Loads the `old-arrows` package with its option `old`. This provides the symbols `\varleftarrow`  $\xleftarrow{\hspace{0.5em}}$  and `\varrightarrow`  $\xrightarrow{\hspace{0.5em}}$ , used then by default for predefined command.

When the `old-arrows` option is set, the commands `\overrightarrow`<sup>→P.21</sup>, `\overleftarrow`<sup>→P.21</sup>, `\overleftrightharrow`<sup>→P.21</sup>, `\underrightarrow`<sup>→P.22</sup>, `\underleftarrow`<sup>→P.22</sup> and `\underleftrightharrow`<sup>→P.22</sup> give respectively :  $\overrightarrow{AB}$ ,  $\overleftarrow{AB}$ ,  $\overleftrightharrow{AB}$ ,  $\underset{\curvearrowright}{AB}$ ,  $\underset{\curvearrowleft}{AB}$  and  $\overleftrightarrow{AB}$ .

#### **tikz**

Loads the package `tikz` with its library `arrows.meta`.

Note that TikZ arrows, drawn with the `tikz` method, are always available, even if this option is not set, provided the `tikz` package and its library are loaded independently.

#### **pstricks**

Loads the package `pstricks-add`.

Note that, as it, this will compile with  $\text{\LaTeX}$ ,  $\text{\Lua\LaTeX}$  and  $\text{\Xe\LaTeX}$ , but not with  $\text{\pdf\LaTeX}$  (see the PSTricks documentation). PSTricks arrows, drawn with the `pstricks` method, are always available, even if this option is not set, provided the `pstricks` package is loaded independently.



### pstarrows

Loads the `pict2e` package, with its option `pstarrows`. Vectors using  $\LaTeX$  `picture` environment gives then  $\overrightarrow{AB}$  instead of  $\overline{AB}$ .

Note that this affect all vectors drawn in  $\LaTeX$  `picture` environments, and that this setting can be changed on the fly with the commands `\pstarrows` and `\ltxarrows` from the `pict2e` package.

### subscripts

Sets the default value of the key `detect subscripts`<sup>P. 25</sup> to `true`.

This option also impacts the command `\vv`<sup>P. 20</sup> and all predefined commands, so that they automatically use their starred variant when a subscript follows.

### subother

New: v1.1 2023/02/15

Sets to 12 (*other* catcode category) the catcode of the “\_” symbol used for subscript detection, when this is enabled by the key `detect subscripts`<sup>P. 25</sup> (see the section 5.1.2, page 31).

### subactive

New: v1.1 2023/02/15

Sets to 13 (*active* catcode category) the catcode of the “\_” symbol used for subscript detection, when this is enabled by the key `detect subscripts`<sup>P. 25</sup> (see the section 5.1.2, page 31).

### debug

Writes the meaning of defined commands in  $\LaTeX$  log.

## 4.2 Commands

### 4.2.1 Macro for commands creation

```
\NewOverArrowCommand[method]{command}{keys}
\RenewOverArrowCommand[method]{command}{keys}
\ProvideOverArrowCommand[method]{command}{keys}
\DeclareOverArrowCommand[method]{command}{keys}
```

Creates the command `<command>` and its starred variant `<command>*`. The starred variant `<command>*` removes the extra end space generated by the arrow, which is suitable, as example, when a subscript follows.

`<command>` can be given with or without backslash (prior to the version 1.2, only the name, without backslash, was accepted).

`\NewOverArrowCommand` raises an error if `<command>` is already defined.

`\RenewOverArrowCommand` raises an error if `<command>` is undefined.

`\ProvideOverArrowCommand` sets `<command>` if the command is undefined and does nothing if it is already defined, without raising any error.

`\DeclareOverArrowCommand` sets `<command>`, whether the command is already defined or not, without raising any error.

Updated: v1.2 2024/07/10

The  $\langle method \rangle$  used to draw the arrow must be:

**symb** to draw the arrow by symbols assemblage (default);

**tikz** to draw the arrow with PGF/TikZ;

**pstricks** to draw the arrow with PSTricks;

**picture** to draw the arrow with the L<sup>A</sup>T<sub>E</sub>X **picture** environment.

With no  $\langle method \rangle$  argument, the **symb** method is chosen.

$\langle keys \rangle$  is a comma-separated list of keys-values. Available keys depends of the  $\langle method \rangle$  chosen and are described in section 4.3, page 22.

```
\NewOverArrowCommand[tikz]{\myoverarrow}{arrows={Bar-Bar}, center arrow}
$ \myoverarrow{v} \quad \myoverarrow{ABCD} $
```

$$\overrightarrow{v} \quad \overrightarrow{ABCD}$$

**\TestOverArrow** $[\langle pattern \rangle]\{\langle command \rangle\}$

**\TestOverArrow\*** $[\langle pattern \rangle]\{\langle command \rangle\}$

Displays the result of the command  $\langle command \rangle$  for patterns of various lengths and for the four math styles. A custom  $\langle pattern \rangle$  can be added to the predefined ones.

The starred variant **\TestOverArrow\*** displays a full report, including kerning tests of the commands  $\langle command \rangle$  and  $\langle command \rangle^*$ .

$\langle command \rangle$  can be given with or without backslash (prior to the version 1.2, only the name, without backslash, was accepted).

```
\TestOverArrow*[my-pattern]{vv}
```

### Test of \vv and \vv\* macros

#### \vv for different math styles

$\backslash displaystyle$	$\backslash textstyle$	$\backslash scriptstyle$	$\backslash scriptscriptstyle$
$\overrightarrow{v}$	$\overrightarrow{v}$	$\overrightarrow{v}$	$\overrightarrow{v}$
$\overrightarrow{AB}$	$\overrightarrow{AB}$	$\overrightarrow{AB}$	$\overrightarrow{AB}$
$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$	$\overrightarrow{\text{grad}}$
$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$	$\overrightarrow{\text{my long vector}}$
$\overrightarrow{\text{my pattern}}$	$\overrightarrow{\text{my pattern}}$	$\overrightarrow{\text{my pattern}}$	$\overrightarrow{\text{my pattern}}$

#### \vv kerning

$$\overrightarrow{t} \overrightarrow{u} \overrightarrow{v} \quad \overrightarrow{i}_0 \quad \overrightarrow{v} = \overrightarrow{v}_x + \overrightarrow{v}_y + \overrightarrow{v}_z = v_x \overrightarrow{i} + v_y \overrightarrow{j} + v_z \overrightarrow{k}$$

#### \vv\* kerning

$$\overrightarrow{t} \overrightarrow{u} \overrightarrow{v} \quad \overrightarrow{i}_0 \quad \overrightarrow{v} = \overrightarrow{v}_x + \overrightarrow{v}_y + \overrightarrow{v}_z = v_x \overrightarrow{i} + v_y \overrightarrow{j} + v_z \overrightarrow{k}$$

### 4.2.2 Useful macros for symbols assemblage

Math symbols assemblage is the default method used to draw arrows. The macros `\xjoinrel` and `\smallermathstyle` are designed to help combine and format math symbols.

`\xjoinrel[⟨number⟩]`

Removes an horizontal space of `⟨number⟩` math units (3.5 mu by default). Must be used in math mode. Useful to assemble math symbols and create new ones.

```
\newcommand*{\triplebar}{\Relbar\xjoinrel[14]\relbar}
\newcommand*{\triplebararrow}{\Relbar\xjoinrel[15]\rightarrow}
\scalebox{2}{\triplebar \quad \triplebararrow} \par
\scalebox{2}{\triplebar\xjoinrel\triplebararrow}
```



`\smallermathstyle`

Applies the next math style, smaller than the current. That is:

- uses `\scriptstyle` if the current math style is `\displaystyle` or `\textstyle`;
- uses `\scriptscriptstyle` if the current math style is `\scriptstyle`;
- does nothing if the current math style is `\scriptscriptstyle`.

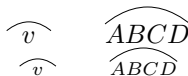
```
$ \displaystyle AB \quad \textstyle AB
\quad \scriptstyle AB \quad \scriptscriptstyle AB $ \par
$ \displaystyle AB \quad \smallermathstyle AB
\quad \smallermathstyle AB \quad \smallermathstyle AB $
```

$AB$   $AB$   $AB$   $AB$   
 $AB$   $AB$   $AB$   $AB$

### 4.2.3 Useful lengths for TikZ, PSTricks or picture environment

Arrows drawn with graphic languages, like PGF/TikZ, PSTricks or the  $\LaTeX$  `picture` environment, are not extensible. The three lengths `\overarrowlength`, `\overarrowthickness` and `\overarrowsmallerthickness` are computed at each utilisation of a command set with the `tikz`, `pstricks` or `picture` method, so they can be used in drawing commands.

```
\NewOverArrowCommand[tikz]{\overparabola}{%
  path options={x=\overarrowlength, line width=\overarrowsmallerthickness},
  path={(0,0) parabola[parabola height=0.2\overarrowlength] (1,0)},
  arrows={-}, center arrow, min length=30,
}
\displaystyle \overparabola{v} \quad \overparabola{ABCD} $ \par
\scriptstyle \overparabola{v} \quad \overparabola{ABCD} $ \par
```



### `\overarrowlength`

Is set to the width of the arrow command content, or, if larger, to the minimal arrow length set through the key `min length`<sup>→P.23</sup>.

### `\overarrowthickness`

Is set to the default rule thickness of the current math style. That is:

- `\fontdimen 8 \textfont 3 in \displaystyle` or `\textstyle`;
- `\fontdimen 8 \scriptfont 3 in \scriptstyle`;
- `\fontdimen 8 \scriptscriptfont 3 in \scriptscriptstyle`.

Updated: v1.2 2024/07/10

Theses settings are adapted when the package `unicode-math` is loaded (using `\Umathoverbarrule` with `LuaLaTeX` or `\fontdimen 54`, family 2 with `XYLaTeX` — see the manual of `unicode-math`).

### `\overarrowsmallerthickness`

Is set to the default rule thickness of the next smaller math style. That is:

- `\fontdimen 8 \scriptfont 3 in \displaystyle` or `\textstyle`;
- `\fontdimen 8 \scriptscriptfont 3 in \scriptstyle` or `\scriptscriptstyle`.

Updated: v1.2 2024/07/10

Theses settings are adapted when the package `unicode-math` is loaded (using `\Umathoverbarrule` with `LuaLaTeX` or `\fontdimen 54`, family 2 with `XYLaTeX` — see the manual of `unicode-math`).

## 4.2.4 Vectors macros

The macro `\vv`, dedicated to vectors, is automatically defined when the option `esvect`<sup>→P.13</sup> is set (which is the default). It is a clone of the `\vv` command provided by the `esvect` package, but its starred variant has a correct kerning when followed by a subscript.

`\vv{<content>}`

`\vv*{<content>}`

Draws a vector arrow upon math `<content>`. The shape of the arrow depends on the corresponding options described in section 4.1.1, page 13 : `esvecta`<sup>→P.14</sup>, `esvectb`<sup>→P.14</sup>, `esvectc`<sup>→P.14</sup>, `esvectd`<sup>→P.14</sup>, `esvecte`<sup>→P.14</sup>, `esvectf`<sup>→P.14</sup>, `esvectg`<sup>→P.14</sup>, `esvecth`<sup>→P.14</sup>.

The starred variant `\vv*` suppresses the end space created by the arrow.

```
$ \vv{\imath}_{0} \quad \vv{e}_r \quad \vv{L}_{\Delta} \quad \par
$ \vv*{\imath}_{0} \quad \vv*{e}_r \quad \vv*{L}_{\Delta} \quad
```

$$\begin{array}{ccc} \vec{i}_0 & \vec{e}_r & \vec{L}_{\Delta} \\ \vec{i}_0 & \vec{e}_r & \vec{L}_{\Delta} \end{array}$$

### `\esvectvv`

Is simply the backup of the original `esvect` `\vv` command.

<code>\$ \esvectvv{\imath}_{0}</code>	<code>\quad \esvectvv{e}_{r}</code>	<code>\quad \esvectvv{L}_{\Delta}</code>	<code>\$\par</code>
<code>\$ \esvectvv*{\imath}_{0}</code>	<code>\quad \esvectvv*{e}_{r}</code>	<code>\quad \esvectvv*{L}_{\Delta}</code>	<code>\$</code>
$\begin{array}{ccc} \vec{i}_0 & \vec{e}_r & \vec{L}_\Delta \\ \overrightarrow{i}_0 & \overrightarrow{e}_r & \overrightarrow{L}_\Delta \end{array}$			

#### 4.2.5 Predefined commands

Predefined commands are defined if the corresponding option is set (see section 4.1.2, page 14). The commands `\overrightarrow`, `\overleftarrow`, `\overleftrightarrow`, `\underrightarrow`, `\underleftarrow` and `\underleftrightarrow` are affected by the option `old-arrows`<sup>P.16</sup>.

##### Over arrows

###### `\overrightarrow`

$$\overrightarrow{v} \quad \overrightarrow{AB} \quad \overrightarrow{\text{grad}}$$

The shape of the arrow is smaller if the option `old-arrows`<sup>P.16</sup> is set.

###### `\overleftarrow`

$$\overleftarrow{v} \quad \overleftarrow{AB} \quad \overleftarrow{\text{grad}}$$

The shape of the arrow is smaller if the option `old-arrows`<sup>P.16</sup> is set.

###### `\overleftrightarrow`

$$\overleftrightarrow{v} \quad \overleftrightarrow{AB} \quad \overleftrightarrow{\text{grad}}$$

The shape of the arrows is smaller if the option `old-arrows`<sup>P.16</sup> is set.

###### `\overrightarrowtharpoonup`

$$\overrightarrowtharpoonup{v} \quad \overrightarrowtharpoonup{AB} \quad \overrightarrowtharpoonup{\text{grad}}$$

###### `\overrightarrowtharpoondown`

$$\overrightarrowtharpoondown{v} \quad \overrightarrowtharpoondown{AB} \quad \overrightarrowtharpoondown{\text{grad}}$$

###### `\overleftarrowtharpoonup`

$$\overleftarrowtharpoonup{v} \quad \overleftarrowtharpoonup{AB} \quad \overleftarrowtharpoonup{\text{grad}}$$

###### `\overleftarrowtharpoondown`

$$\overleftarrowtharpoondown{v} \quad \overleftarrowtharpoondown{AB} \quad \overleftarrowtharpoondown{\text{grad}}$$

###### `\overbar`

$$\overbar{v} \quad \overbar{AB} \quad \overbar{\text{grad}}$$

##### Under arrows

`\underrightarrow`

$$\underset{\rightarrow}{v} \quad \underset{\rightarrow}{AB} \quad \underset{\rightarrow}{\text{grad}}$$

The shape of the arrow is smaller if the option `old-arrows→P.16` is set.

`\underleftarrow`

$$\underset{\leftarrow}{v} \quad \underset{\leftarrow}{AB} \quad \underset{\leftarrow}{\text{grad}}$$

The shape of the arrow is smaller if the option `old-arrows→P.16` is set.

`\underleftrightharpoonup`

$$\underset{\leftrightharpoonup}{v} \quad \underset{\leftrightharpoonup}{AB} \quad \underset{\leftrightharpoonup}{\text{grad}}$$

The shape of the arrows is smaller if the option `old-arrows→P.16` is set.

`\underrightharpoonup`

$$\underset{\rightharpoonup}{v} \quad \underset{\rightharpoonup}{AB} \quad \underset{\rightharpoonup}{\text{grad}}$$

`\underrightharpoontdown`

$$\underset{\rightharpoonupdown}{v} \quad \underset{\rightharpoonupdown}{AB} \quad \underset{\rightharpoonupdown}{\text{grad}}$$

`\underleftharpoonup`

$$\underset{\leftharpoonup}{v} \quad \underset{\leftharpoonup}{AB} \quad \underset{\leftharpoonup}{\text{grad}}$$

`\underleftharpoontdown`

$$\underset{\leftharpoonupdown}{v} \quad \underset{\leftharpoonupdown}{AB} \quad \underset{\leftharpoonupdown}{\text{grad}}$$

`\underbar`

$$\underline{v} \quad \underline{AB} \quad \underline{\text{grad}}$$

## 4.3 Keys

The customisation of arrows is done at command creation through a key-value interface provided by the `pgfkeys` package (with `/overarrows/` as key path).

### 4.3.1 Arrow position and length settings

These keys are available whatever the method chosen at command creation (see section 4.2.1, page 17 for the documentation of commands creation).

#### Length

**min length**={ $\langle number \rangle$ } (no default, see below for the initial value)

Sets the minimal arrow length to  $\langle number \rangle$  math units. The arrow length is set from content width, or, if larger, to this value.

The initial value of **min length** depends on the  $\langle method \rangle$  chosen at command creation (see section 4.2.1, page 17 for the documentation of commands creation):

- $\langle number \rangle = 0$  for the **symp** method (method by default);
- $\langle number \rangle = 12$  for the **tikz** method;
- $\langle number \rangle = 12$  for the **pstricks** method;
- $\langle number \rangle = 18$  for the **picture** method.

```
\NewOverArrowCommand{\overlongarrow}{min length=50}
$ \overlongarrow{v} \qquad \overlongarrow{ABCDEF} $
```

$$\overrightarrow{v} \qquad \overrightarrow{ABCDEF}$$

## Placement

**arrow under** (default **autoconfig**, initially unset)

**arrow under**=autoconfig|noconfig

Places the arrow under, instead of over.

**arrow under** or **arrow under**=autoconfig also configures suitably the key **detect subscripts**<sup>→P. 25</sup> to **false** and the key **before arrow**<sup>→P. 24</sup> to get an additional space over the arrow.

**arrow under**=noconfig does not do any additional configuration.

```
\NewOverArrowCommand{\underhooks}{%
  start={\lhook}, end={\rhook}, trim=1,
  arrow under, shift leftright=-4,
}
$ \underhooks{v} \qquad \underhooks{AB} $
```

$$\underhook{v} \qquad \underhook{AB}$$

## Horizontal shifts

**shift left**={ $\langle number \rangle$ } (no default, initially 2)

Shifts the left side of the arrow by  $\langle number \rangle$  math units (positive number means a shift to the right).

**shift right**={ $\langle number \rangle$ } (no default, see below for the initial value)

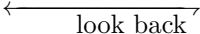
Shifts the right side of the arrow by  $\langle number \rangle$  math units (positive number means a shift to the left).

The initial value of **shift right** depends on the  $\langle method \rangle$  chosen at command creation (see section 4.2.1, page 17 for the documentation of commands creation):

- $\langle number \rangle = 0$  for the **symp** method (method by default);

- $\langle number \rangle = -2$  for the `tikz`, `pstricks` and `picture` methods.

```
\NewOverArrowCommand{\lookback}{%
  start={\leftarrow}, end={\rightharpoonowdown},
  shift left=-50, shift right=-10,
}
$ \lookback{\text{look back}} $
```



`shift leftright`=[ $\langle number \rangle$ ] (no default)

Sets `shift left`<sup>P.23</sup> and `shift right`<sup>P.23</sup> to the same  $\langle number \rangle$  value.

`center arrow`

Sets `shift left`<sup>P.23</sup> and `shift right`<sup>P.23</sup> to zero.

`left arrow` (default 2)

Sets `shift left`<sup>P.23</sup> to zero and `shift right`<sup>P.23</sup> to  $\langle number \rangle$ .

`right arrow` (default 2)

Sets `shift right`<sup>P.23</sup> to zero and `shift left`<sup>P.23</sup> to  $\langle number \rangle$ .

## Vertical adjunct

`before arrow`={ $\langle vertical material \rangle$ } (initially empty)

`after arrow`={ $\langle vertical material \rangle$ } (initially empty)

Adds the  $\langle vertical material \rangle$  before or after the arrow.

Over and under arrow commands are typeset through the  $\text{\TeX}$  `\ialign` command, which aligns contents, like a tabular. The  $\langle vertical material \rangle$  is inserted *between* the rows, with  $\text{\TeX}$  `\noalign` command.

These keys are essentially used to add some extra space between the arrow and the content of the command. They can be set in a handier way with the keys `space before arrow` and `space after arrow`.


`space before arrow`={ $\langle length \rangle$ } (no default)

Adds a space of  $\langle length \rangle$  before the arrow. This sets the keys `before arrow`.

`space after arrow`={ $\langle length \rangle$ } (no default)

Adds a space of  $\langle length \rangle$  after the arrow. This sets the keys `after arrow`.

```
\NewOverArrowCommand{\overharpoonowdown}{%
  start={\leftharpoonowdown}, end={\rightharpoonowdown}, center arrow,
  space before arrow=-0.2ex, space after arrow=0.3ex,
}
$ \dot{\overharpoonowdown{v}} \quad \ddot{\overharpoonowdown{AB}} $
```





### 4.3.2 Subscripts detection setting

This key is available whatever the method chosen at command creation (see section 4.2.1, page 17 for the documentation of commands creation).

**detect subscripts**=true|false (default true, see below for the initial value)

Removes automatically the extra end space created by the arrow, if a subscript immediately follows the command.

By default, the initial value of **detect subscripts** is false. When the option **subscripts**<sup>→P.17</sup> is set, the initial value of **detect subscripts** is true.

Note that the detection may fail when the standard subscript command is changed or altered (see the section 5.1.2, page 31).

```
\NewOverArrowCommand{\autosub}{detect subscripts}
$ \imath_0 \qquad \autosub{\imath}_0 \qquad
{\autosub{\imath}}_0 \qquad {\autosub*{\imath}}_0 $
```

$$\imath_0 \quad \overrightarrow{\imath}_0 \quad \overrightarrow{\imath}_0 \quad \overrightarrow{\imath}_0$$

### 4.3.3 Symbols assemblage settings

The following keys are available for arrows drawn with the default **sybm** method (see section 4.2.1, page 17 for the documentation of commands creation).

**start**=<command> (no default, initially \relbar)  
**middle**=<command> (no default, initially set by middle config=auto)  
**end**=<command> (no default, see below for the initial value)

Sets the <command> used to draw the start (left), middle (center) or end (right) part of the arrow. The **middle** one is repeated, if necessary, to extend the arrow. It is set, initially by **middle config=auto**. By default, the **end** symbols is initially \rightarrow →. When the option **old-arrows**<sup>→P.16</sup> is set, the initial value of **end** is \varrightarrow →.

**start** and **end** symbols are typeset in the same group. **middle** is typeset alone. This means that, if a command, like \smallermathstyle<sup>→P.19</sup>, is used to alter the symbols, it should be applied both to **start** and **middle** (but not to **end**).

```
\NewOverArrowCommand{\smalleroverrightarrow}{%
start={\smallermathstyle\relbar},
middle={\smallermathstyle\relbareda},
end={\rightarrow},
space after arrow={0.2ex},
}
$ \smalleroverrightarrow{v} \qquad \smalleroverrightarrow{AB} $
```

$$\overrightarrow{v} \quad \overrightarrow{AB}$$

**trim start**=<number> (no default, initially 7)

Trims <number> math units from the right side of the **start** symbol.

**trim middle**=<number> (no default, initially set by middle config=auto)

Trims <number> math units from both left and right sides of the **middle** symbol.

**trim end**={ $\langle number \rangle$ } (no default, initially 7)

Trims  $\langle number \rangle$  math units from the left side of the **end** symbol.

**trim**={ $\langle number \rangle$ } (no default)


Sets **trim start**<sup>→P.25</sup>, **trim middle**<sup>→P.25</sup> and **trim end** to the same  $\langle number \rangle$  value.


**no trimming**

Clears **trim start**<sup>→P.25</sup>, **trim middle**<sup>→P.25</sup> and **trim end**.

**middle config**=auto|relbar|relbareda|harrowextender (no default)

Sets a suitable configuration for the keys **middle**<sup>→P.25</sup> and **trim middle**<sup>→P.25</sup>:

For **middle config** = relbar, **middle**<sup>→P.25</sup> is set to `\relbar`  and **trim middle**<sup>→P.25</sup> to 2.5.

For **middle config** = relbareda, **middle**<sup>→P.25</sup> is set to `\relbareda`  and **trim middle**<sup>→P.25</sup> to 1.

For **middle config** = harrowextender, **middle**<sup>→P.25</sup> is set to `\harrowextender` and **trim middle**<sup>→P.25</sup> to 0.

For **middle config** = auto, if `\harrowextender` is provided by the math font<sup>10</sup>, **middle**<sup>→P.25</sup> is set with **middle config** = harrowextender. If `\harrowextender` isn't available, **middle**<sup>→P.25</sup> is set with **middle config** = relbareda if the option **esvect**<sup>→P.13</sup> is set (which is the default) and **middle config** = relabar if not.

**amsmath** (default mimic)

**amsmath**=mimic|strict

Loads a configuration coherent with **amsmath** `\overrightarrow` command.

**amsmath** or **amsmath**=mimic sets the corresponding keys suitably:

<b>start</b> ={ <code>\relbar</code> }	<b>middle</b> ={ <code>\relbar</code> }	<b>end</b> ={ <code>\rightarrow</code> }
<b>trim start</b> =7	<b>trim middle</b> =2	<b>trim end</b> =7
<b>shift leftright</b> =0	<b>after arrow</b> ={}	<b>before arrow</b> ={}

**amsmath**=strict makes, in addition, the command uses the internal macros of **amsmath** `\overrightarrow` (no trimming, fill macro={`\arrowfill@`}, stack macro={`\overarrow@`}). Note that many configuration keys becomes ineffective.

**esvect** (default mimic)

**esvect**=mimic|strict

Loads a configuration coherent with **amsmath** `\vv` command.

**esvect** or **esvect**=mimic sets the corresponding keys suitably:

<b>start</b> ={ <code>\relbared</code> }	<b>middle</b> ={ <code>\relbareda</code> }	<b>end</b> ={ <code>\fldr</code> }
<b>trim start</b> =1.5	<b>trim middle</b> =0	<b>trim end</b> =1.5
<b>space before arrow</b> =-.7pt	<b>space after arrow</b> =-.3pt	<b>right arrow</b> =2

**esvect**=strict makes, in addition, the command uses the internal macros of **esvect** `\vv` (no trimming, fill macro={`\traitfill@`}, stack macro={`\overvect@`}). Note that many configuration keys becomes ineffective.

<sup>10</sup>See the documentation of the package `unicode-math`.

### 4.3.4 TikZ settings

If, at command creation (see section 4.2.1, page 17 for the documentation of commands creation), the `tikz` method is chosen, then the arrow is drawn by the command:


$$\backslash tikz[tikz options]\{tikz command\}$$

where `tikz options` and `tikz command`<sup>P.28</sup> are two keys described below. When `tikz command` is let unset, the drawing command turns into:

$$\backslash tikz[tikz options]\{\draw[path options] path;\}$$

The best way to customise `tikz` arrows is then to set the keys `tikz options`, `path options` and `path`, preferably through the handy alternatives: `add tikz options`, `add path options`, `arrows`, `line thickness` or `thinner`<sup>P.28</sup>.

```
\NewOverArrowCommand[tikz]{\overdotteddoublearrow}{%
  add tikz options={blue}, add path options={densely dotted},
  arrows={->[scale=0.5]>[scale=0.5]}, thinner,
  min length=20, space after arrow={0.3ex},
}
$ \overdotteddoublearrow{v} \quad \overdotteddoublearrow{AB} $
```



The following keys are available when the `tikz` method is chosen.

**tikz options**={*TikZ options*}  
(no default, initially `x=\overarrowlength`, `line width=\overarrowthickness`)

Sets TikZ options to *TikZ options*.

**path options**={*path options*}  
(no default, initially `arrows=-Classical TikZ Rightarrow`, `cap=round`)

Sets TikZ path options to *path options*.

**path**={*path specification*} (no default, initially `(0,0)--(1,0)`)

Sets TikZ path specification to *path* (the ending semicolon is automatically appended).

**add tikz options**={*TikZ options*} (no default)

Appends the options *TikZ options* to the key `tikz options`.

**add path options**={*path options*} (no default)

Appends the options *path options* to the key `path options`.

**arrows**={*arrow specification*} (no default)

Appends the option `arrows={arrow specification}` to the key `path options`.

**line thickness**={*length*} (no default)

Appends the option `line width={length}` to the key `path options`.

**thinner**

Sets the keys `line thickness` with `\overarrowsmallerthickness`.

**tikz command**= $\langle TikZ\ command \rangle$  (initially unset)

Sets the  $\langle TikZ\ command \rangle$  used to draw the arrow. If left unset, the value `\draw[path options] path;` is used.

### 4.3.5 PStricks settings

New: v1.2 2024/07/10

If, at command creation (see section 4.2.1, page 17 for the documentation of commands creation), the `pstricks` method is chosen, then the arrow is drawn with by:

```
\begin{pspicture}geometry%
  \psset{linewidth=line thickness}%
  \psset{psset}%
  pstricks command%
\end{pspicture}%
```

where `geometry`, `line thickness`<sup>P. 29</sup>, `psset` and `pstricks command` are four keys described below.

```
\NewOverArrowCommand[pstricks]{\overloopandarrow}{
  pstricks command=%
  \pscurve{->}(0,0)
  (0.6\overarrowlength,0.05\overarrowlength)
  (0.5\overarrowlength,0.1\overarrowlength)
  (0.4\overarrowlength,0.05\overarrowlength)
  (\overarrowlength,0)
},
  geometry={(0,0)(\overarrowlength,0.2\overarrowlength)},
  space after arrow=2pt, min length=20,
  geometry={(0,0)(\overarrowlength,0.2\overarrowlength)},
}
$ \overloopandarrow{v} \quad \overloopandarrow{AB} $
```

$\vec{v}$        $\overrightarrow{AB}$

The following keys are available when the `pstricks` method is chosen.

**pstricks command**= $\langle pstricks\ command \rangle$   
(no default, initially `\psline{->}(0,0)(\overarrowlength,0)`)

Sets the `pspicture` command to  $\langle pstricks\ command \rangle$ .

**arrow**= $\langle arrow \rangle$  (no default, initially `->`)

Sets `pstricks command` with `\psline{ $\langle arrow \rangle$ }(0,0)(\overarrowlength,0)`.

**psset**= $\langle pstricks\ setting \rangle$  (no default, initially empty)

Sets  $\langle pstricks\ setting \rangle$  with `\psset`.

**geometry**= $\langle pstricks\ geometry\ specification \rangle$   
(no default, initially `(0,-0.5ex)(\overarrowlength,1ex)`)

Sets the `pspicture` geometry to  $\langle pstricks\ geometry\ specification \rangle$ .

**line thickness**= $\langle length \rangle$  (no default)

Sets the line thickness to  $\langle length \rangle$ .

**thinner**

Sets the keys **line thickness** with `\overarrowsmallerthickness`.

### 4.3.6 Picture environment settings

If, at command creation (see section 4.2.1, page 17 for the documentation of commands creation), the `picture` method is chosen, then the arrow is drawn with by:

```
\begin{picture}geometry%
  \linethickness{line thickness}%
  picture command%
\end{picture}%
```

where `geometry`, `line thickness` and `picture command` are three keys described below.

```
% ^^A \arc and \roundcap commands are from the pict2e package
% ^^A this example needs \usepackage{pict2e} in the preamble
\NewOverArrowCommand[picture]{\overarc}{%
  picture command={%
    \roundcap
    \put(0.5\overarrowlength,0){\arc[180,0]{0.6\overarrowlength}}
  },
  geometry={%
    (1.2\overarrowlength,0.5\overarrowlength)(-0.1\overarrowlength,0.2ex)
  },
  thinner, center arrow,
}
$ \overarc{v} \quad \overarc{AB} $
```



The following keys are available when the `picture` method is chosen.

**picture command**= $\langle picture command \rangle$   
(no default, initially `\put(0,0){\vector(1,0){\overarrowlength}}`)

Sets picture command to  $\langle picture command \rangle$ .

**geometry**= $\langle picture geometry specification \rangle$   
(no default, initially `(\overarrowlength,1ex)(0,-0.5ex)`)

Sets picture geometry to  $\langle picture geometry specification \rangle$ .

**line thickness**= $\langle length \rangle$  (no default)

Sets the picture line thickness to  $\langle length \rangle$ .

**thinner** (no default)

Sets the keys **line thickness** with `\overarrowsmallerthickness`.

## 4.4 Advanced commands and keys

The following commands and keys are used in the implementation of the `overarrows` package. They can also be employed for an advanced configuration of the commands created, although unnecessary in the vast majority of cases.

### 4.4.1 Advanced commands

`\SetOverArrowsSubscriptCommand{⟨command⟩}`

New: v1.1 2023/02/15

Sets to `⟨command⟩` the command used for subscript detection, when this is enabled by the key `detect subscripts`<sup>→P.25</sup> (see the section 5.1.2, page 31).

`\SetOverArrowsMethod[⟨stack mechanism⟩]{⟨name⟩}[⟨pre code⟩]{⟨keys def⟩}`

`\SetOverArrowsMethod*{⟨name⟩}[⟨pre code⟩]{⟨keys def⟩}`

Defines the method `⟨name⟩`, to be used in commands `\NewOverArrowCommand`<sup>→P.17</sup>, `\RenewOverArrowCommand`<sup>→P.17</sup>, `\ProvideOverArrowCommand`<sup>→P.17</sup> or `\DeclareOverArrowCommand`. When the `⟨name⟩` method is chosen, corresponding keys are defined by `⟨keys def⟩`. This must set, in particular, the keys `no stack macro hook`<sup>→P.31</sup> and `no arrow macro hook`<sup>→P.31</sup>. Optional code `⟨pre code⟩` is evaluated before the keys definition.

The unstarred variant automatically defines the key `no stack macro hook`<sup>→P.31</sup>, according to the value of the optional `⟨stack mechanism⟩`. This one must be:

**fill** if `arrow macro` creates extensible arrows (typically with `\cleaders`).

In this case, the arrow macro (defined by `no arrow macro hook`<sup>→P.31</sup>) is called with the math style, passed as argument (it can be, for example, the macro `\rightarrowfill@` used by `amsmath` `\overrightarrow`). `fill` is the mechanism used by the `symb` method.

**lens** if `arrow macro` creates fixed-length arrows, and needs the computation of lengths `\overarrowlength`<sup>→P.20</sup>, `\overarrowthickness`<sup>→P.20</sup> and `\overarrowsmallerthickness`<sup>→P.20</sup>. In this case, the arrow macro (defined by `no arrow macro hook`<sup>→P.31</sup>) is called without argument. `lens` is the mechanism used by the `tikz` and `picture` methods.

Without optional `⟨stack mechanism⟩`, `fill` is used. The starred variant does not set the key `no stack macro hook`<sup>→P.31</sup>.

### 4.4.2 Advanced keys

`stack macro={⟨stack definition⟩}` (no default, initially unset)

Defines the stack macro to be `⟨stack definition⟩`. Stack macro is a command which takes three arguments: the arrow macro set by `arrow macro`, the math style, and the command content (under or over the arrow). `⟨stack definition⟩` can be, for example, the macro `\overarrow@` used by `amsmath` `\overrightarrow` `\overrightarrow`.

`arrow macro={⟨arrow definition⟩}` (no default, initially unset)

Defines the arrow macro (used in the stack macro) by to be `⟨arrow definition⟩`.

**no stack macro hook**={ $\langle code \rangle$ } (no default)

Sets the  $\langle code \rangle$  executed if **stack macro** is left unset, after user evaluation of  $\langle keys \rangle$  in `\NewOverArrowCommand`<sup>P.17</sup>, `\RenewOverArrowCommand`<sup>P.17</sup>, `\ProvideOverArrowCommand`<sup>P.17</sup> or `\DeclareOverArrowCommand`<sup>P.17</sup>.

$\langle code \rangle$  must configure **stack macro**<sup>P.30</sup> accordingly to the user keys setting.

**no arrow macro hook**={ $\langle code \rangle$ } (no default)

Sets the  $\langle code \rangle$  executed if **arrow macro**<sup>P.30</sup> is left unset, after user evaluation of  $\langle keys \rangle$  in `\NewOverArrowCommand`<sup>P.17</sup>, `\RenewOverArrowCommand`<sup>P.17</sup>, `\ProvideOverArrowCommand`<sup>P.17</sup> or `\DeclareOverArrowCommand`<sup>P.17</sup>.

$\langle code \rangle$  must configure **arrow macro**<sup>P.30</sup> accordingly to the user keys setting.

**fill macro**={ $\langle definition \rangle$ } (no default, initially unset)

Defines the fill macro to be  $\langle definition \rangle$ . The fill macro is used by arrows created with the **symp** method, to set **arrow macro**<sup>P.30</sup> in **no arrow macro hook**. It is called with four arguments: start, middle and end symbols used to draw the arrow, and the math style.  $\langle definition \rangle$  can be, for example, the macro `\arrowfill@` used by `amsmath` `\overrightarrow`.

## 5 Complements

### 5.1 Know issues

#### 5.1.1 Math font change

If the math font differs from the default *Computer Modern*, arrow drawn with the **symp** method may have a central part of the arrow with inappropriate position or line width. This is because the default symbol used for the arrow line is `\relbareda` from the `esvect` package. This can be fixed with the **noesvect**<sup>P.14</sup> option.

Depending of the math font, predefined command may be faulty. For example, at the time of writing, the hook vertical position is incorrect with *Asana Math* or `\harrowextender` is badly positioned with *Stix two Math* (for the smaller math styles), *Libertinus Math* and *GFSNeohellenicMath*.

#### 5.1.2 Detection of non standard subscripts

The subscript detection enabled by the key **detect subscripts**<sup>P.25</sup> is based on the  $\text{\LaTeX}$  macro `\@ifnextchar`. The detection may fail if the standard subscript command is modified or altered. This is the case, as example:

- with the `spbookmark` package (<https://www.ctan.org/pkg/spbookmark>), by Qu Yi, which allows a complete customisation of subscripts, through the `\sub` command;
- with the `altsubsup` package (<https://www.ctan.org/pkg/altsubsup>), by Julien Labbé, which provides an alternative subscript format, and changes, for this purpose, the catcode of the underscore symbol “`_`” from 8 (*subscript* catcode category) to 12 (*other* catcode category).

To handle these cases, the command used for subscript detection can be re-defined with `\SetOverArrowsSubscriptCommand`<sup>P.30</sup>. Compatibility with the `spbookmark` package is then obtained by:

```
\SetOverArrowsSubscriptCommand{\sub}
```

In the same way, with the `altsubsup` package, add:

```
\SetOverArrowsSubscriptCommand{_{}}
```

after the `\begin{document}` (namely, after the catcode redefinition done by `altsubsup`).

Alternatively, two package options handle the cases where the catcode of the underscore “`_`” symbol is changed: `subother`<sup>P.17</sup> (for catcode 12, or *other*) and `subactive`<sup>P.17</sup> (for catcode 13, or *active*). Hence, setting the `subother`<sup>P.17</sup> option is sufficient for compatibility with the `altsubsup` package (no need of `\SetOverArrowsSubscriptCommand`<sup>P.30</sup>). Note, that with options `subother`<sup>P.17</sup> and `subactive`<sup>P.17</sup>, the command `\TestOverArrow*`<sup>P.18</sup> may give bad results for kerning test, as defined before the catcode redefinition.

## 5.2 Package dependencies

The following packages are used by `overarrows`:

- `amsmath`
- `etoolbox`
- `pgfkeys`
- `esvect` (unless the option `noesvect`<sup>P.14</sup> is used)
- `old-arrows` (when the option `old-arrows`<sup>P.16</sup> is used)
- `tikz` (when the `tikz` method or the option `tikz`<sup>P.16</sup> is used)
- `pict2e` (when the option `pstarrings`<sup>P.17</sup> is used)

L<sup>A</sup>T<sub>E</sub>X distributions prior to 2020/10/01 must load the `xparse` package before `overarrows`.

## 5.3 Alternatives

**esvect** package (<https://www.ctan.org/pkg/esvect>), by Eddie Sautrais, provides the fine vector macro `\vv`. This package is loaded by default by `overarrows`.

**letterswitharrows** package (<https://www.ctan.org/pkg/letterswitharrows>), by Max Teegen, provides left and right over arrows commands, which can extend to multiple characters.

**overrightarrow** package (<https://www.ctan.org/pkg/overrightarrow>), by Robin Fairbairns, provides the `\Overrightarrow` which is an amalgam of `\overrightarrow` and `\Rightarrow`.

**harpoon** package (<https://ctan.org/pkg/harpoon>), by Tobias Kuipers, provides over- and under-harpoon symbol commands.



## 5.4 Changelog

- v1.2
  - Fix compatibility issues with unicode-math.
  - Allow to draw the arrow with PSTricks.
  - Make esvect handle all font sizes.
  - Rewrite starred variant for better performances.
- v1.1 Support for non-standard subscripts.
- v1.0.1 Bug fix for under\* options.
- v1.0 Initial version.

## 6 Implementation

### Management of options

#### Declaration of conditionals

```
1 \newif\ifovar@option@oldarrows@
2 \newif\ifovar@option@esvect@ \ovar@option@esvect@true \PassOptionsToPackage{f}{esvect}
3 \newif\ifovar@option@tikz@
4 \newif\ifovar@option@pstricks@
5 \newif\ifovar@option@pstarrows@
6 \newif\ifovar@detectsubscripts@
7 \newif\ifovar@option@subother@
8 \newif\ifovar@option@subactive@
9 \newif\ifovar@option@debug@
```

Following conditionals are for predefined commands.

```
10 \newif\ifovar@option@overrightarrow@
11 \newif\ifovar@option@underrightarrow@
12 \newif\ifovar@option@overleftarrow@
13 \newif\ifovar@option@underleftarrow@
14 \newif\ifovar@option@overleftrightharpoon@
15 \newif\ifovar@option@underleftrightharpoon@
16 \newif\ifovar@option@overrightharpoonup@
17 \newif\ifovar@option@underrightharpoonup@
18 \newif\ifovar@option@overrightharpoondown@
19 \newif\ifovar@option@underrightharpoondown@
20 \newif\ifovar@option@overleftharpoonup@
21 \newif\ifovar@option@underleftharpoonup@
22 \newif\ifovar@option@overleftharpoondown@
23 \newif\ifovar@option@underleftharpoondown@
24 \newif\ifovar@option@overbar@
25 \newif\ifovar@option@underbar@
```

#### Declaration of options

```
26 \DeclareOption{esvect}{\ovar@option@esvect@true}
27 \DeclareOption{noesvect}{\ovar@option@esvect@false}
28 \DeclareOption{esvecta}{\ovar@option@esvect@true\PassOptionsToPackage{a}{esvect}}
29 \DeclareOption{esvectb}{\ovar@option@esvect@true\PassOptionsToPackage{b}{esvect}}
30 \DeclareOption{esvectc}{\ovar@option@esvect@true\PassOptionsToPackage{c}{esvect}}
```

```

31 \DeclareOption{esvectd}{\over@option@esvect@true\PassOptionsToPackage{d}{esvect}}
32 \DeclareOption{esvecte}{\over@option@esvect@true\PassOptionsToPackage{e}{esvect}}
33 \DeclareOption{esvectf}{\over@option@esvect@true\PassOptionsToPackage{f}{esvect}}
34 \DeclareOption{esvectg}{\over@option@esvect@true\PassOptionsToPackage{g}{esvect}}
35 \DeclareOption{esvecth}{\over@option@esvect@true\PassOptionsToPackage{h}{esvect}}
36 \DeclareOption{old-arrows}{\over@option@oldarrows@true}
37 \DeclareOption{tikz}{\over@option@tikz@true}
38 \DeclareOption{pstricks}{\over@option@pstricks@true}
39 \DeclareOption{pstarrows}{\over@option@pstarrows@true}
40 \DeclareOption{subscripts}{\over@detectsubscripts@true}
41 \DeclareOption{subother}{\over@option@subother@true}
42 \DeclareOption{subactive}{\over@option@subactive@true}
43 \DeclareOption{debug}{\over@option@debug@true}

```

Following options are for predefined commands.

```

44 \DeclareOption{overrightarrow}{\over@option@overrightarrow@true}
45 \DeclareOption{underrightarrow}{\over@option@underrightarrow@true}
46 \DeclareOption{overleftarrow}{\over@option@overleftarrow@true}
47 \DeclareOption{underleftarrow}{\over@option@underleftarrow@true}
48 \DeclareOption{overleftrightharpoonup}{\over@option@overleftrightharpoonup@true}
49 \DeclareOption{underleftrightharpoonup}{\over@option@underleftrightharpoonup@true}
50 \DeclareOption{overrightharpoonup}{\over@option@overrightharpoonup@true}
51 \DeclareOption{underrightharpoonup}{\over@option@underrightharpoonup@true}
52 \DeclareOption{overrightharpoondown}{\over@option@overrightharpoondown@true}
53 \DeclareOption{underrightharpoondown}{\over@option@underrightharpoondown@true}
54 \DeclareOption{overleftharpoonup}{\over@option@overleftharpoonup@true}
55 \DeclareOption{underleftharpoonup}{\over@option@underleftharpoonup@true}
56 \DeclareOption{overleftharpoondown}{\over@option@overleftharpoondown@true}
57 \DeclareOption{underleftharpoondown}{\over@option@underleftharpoondown@true}
58 \DeclareOption{overbar}{\over@option@overbar@true}
59 \DeclareOption{underbar}{\over@option@underbar@true}

```

Following options are for sets of predefined commands.

```

60 \DeclareOption{overcommands}{%
61   \over@option@overrightarrow@true
62   \over@option@overleftarrow@true
63   \over@option@overleftrightharpoonup@true
64   \over@option@overrightharpoonup@true
65   \over@option@overrightharpoondown@true
66   \over@option@overleftharpoonup@true
67   \over@option@overleftharpoondown@true
68   \over@option@overbar@true
69 }
70 \DeclareOption{undercommands}{%
71   \over@option@underrightarrow@true
72   \over@option@underleftarrow@true
73   \over@option@underleftrightharpoonup@true
74   \over@option@underrightharpoonup@true
75   \over@option@underrightharpoondown@true
76   \over@option@underleftharpoonup@true
77   \over@option@underleftharpoondown@true
78   \over@option@underbar@true
79 }
80 \DeclareOption{allcommands}{%
81   \over@option@overrightarrow@true
82   \over@option@underrightarrow@true
83   \over@option@overleftarrow@true
84   \over@option@underleftarrow@true
85   \over@option@overleftrightharpoonup@true
86   \over@option@underleftrightharpoonup@true
87   \over@option@overrightharpoonup@true
88   \over@option@underrightharpoonup@true

```

```

89 \ovar@option@overrightharpoon@true
90 \ovar@option@underrightharpoon@true
91 \ovar@option@overleftharpoon@true
92 \ovar@option@underleftharpoon@true
93 \ovar@option@overleftharpoondown@true
94 \ovar@option@underleftharpoondown@true
95 \ovar@option@overbar@true
96 \ovar@option@underbar@true
97 }

```

## Options processing

```

98 \DeclareOption*{\PackageWarning{overarrows}{Unknown option: '\CurrentOption'}}
99 \ProcessOptions\relax

```

## Package dependencies

L<sup>A</sup>T<sub>E</sub>X distributions prior to 2020/10/01 must add the xparse package.

```

100 \RequirePackage{amsmath}
101 \RequirePackage{etoolbox}

```

Option `old-arrows`<sup>→ P. 16</sup>. Configuration of arrows used for predefined commands.

```

102 \def\ovar@rightarrow{\rightarrow}
103 \def\ovar@leftarrow{\leftarrow}
104 \ifovar@option@oldarrows@
105 \RequirePackage[old]{old-arrows}
106 \def\ovar@rightarrow{\varrightarrow}
107 \def\ovar@leftarrow{\varleftarrow}
108 \fi

```

Option `esvect`<sup>→ P. 13</sup>.

```

109 \ifovar@option@esvect@
110 \RequirePackage{esvect}

```

Fix font description in `uesvect.fd` to allow any sizes (taken from Enrico Gregorio, <https://tex.stackexchange.com/a/689863/>)

```

111 \DeclareFontFamily{U}{esvect}{}
112 \DeclareFontShape{U}{esvect}{m}{n}{
113 <-5.5> vect5
114 <5.5-6.5> vect6
115 <6.5-7.5> vect7
116 <7.5-8.5> vect8
117 <8.5-9.5> vect9
118 <9.5-> vect10
119 }{}
120 \fi

```

Option `tikz`<sup>→ P. 16</sup>.

```

121 \ifovar@option@tikz@
122 \RequirePackage{tikz}
123 \usetikzlibrary{arrows.meta}
124 \fi

```

Option `pstricks`<sup>→ P. 16</sup>.

```

125 \ifovar@option@pstricks@
126 \RequirePackage{pstricks-add}
127 \fi

```

Option `pstarrows`<sup>→ P. 17</sup>.

```

128 \ifovar@option@pstarrows@
129   \RequirePackage[pstarrows]{pict2e}
130 \fi

```

Add hook rules to apply settings after unicode-math.

```

131 \DeclareHookRule{begindocument}{overarrows}{after}{unicode-math-luatex}
132 \DeclareHookRule{begindocument}{overarrows}{after}{unicode-math-xetex}

```

Set `\ovar@auto@middle` and `\ovar@auto@trim@middle`, used by configurations made with `middle config=auto`.

```

133 \AddToHook{begindocument}[overarrows]
134 {%
135   \ifdef{\relbareda}
136   {%
137     \gdef\ovar@auto@middle{\relbareda}
138     \gdef\ovar@auto@trim@middle{1}
139   }
140   {%
141     \gdef\ovar@auto@middle{\relbar}
142     \gdef\ovar@auto@trim@middle{2.5}
143   }%
144   \ifpackageloaded{unicode-math}
145   {%

```

Test of `\harrowextender` availability taken from Enrico Gregorio, (<https://tex.stackexchange.com/a/218407/>).

```

146   \check@mathfonts
147   \iffontchar\textfont\tw@\string"23AF
148   \gdef\ovar@auto@middle{\mathrel\harrowextender}
149   \gdef\ovar@auto@trim@middle{0}
150   \fi
151 }
152 {}%
153 }

```

## Configuration of subscripts detection

`\SetOverArrowsSubscriptCommand`

Sets the subscript command.

```

154 \newcommand{\SetOverArrowsSubscriptCommand}[1]{\global\let\ovar@subcmd=#1}

```

Initial configuration.

```

155 \SetOverArrowsSubscriptCommand{_{}}

```

Option `subother`<sup>P.17</sup> for *other* (catcode 12) subscript commands.

```

156 \ifovar@option@subother@
157   \begingroup
158   \catcode`_=12
159   \SetOverArrowsSubscriptCommand{_{}}%
160   \endgroup
161 \fi

```

Option `subactive`<sup>P.17</sup> for *active* (catcode 13) subscript commands.

```

162 \ifovar@option@subactive@
163   \begingroup
164   \catcode`_=13
165   \SetOverArrowsSubscriptCommand{_{}}%
166   \endgroup
167 \fi

```

## Management of keys

### Family declaration and setters

```

168 \RequirePackage{pgfkeys}
169 \pgfkeys{overarrows/.is family}
\ovar@set
170 \newcommand{\ovar@set}[1]{\pgfqkeys{/overarrows}{#1}}
\SetOverArrowsMethod
171 \NewDocumentCommand{\SetOverArrowsMethod}{s O{fill} m O{} m }{%
172   \IfBooleanTF{#1}{%
173     \csgdef{\ovar@set@#3}{#4\ovar@set{#5}}%
174   }{%
175     \csgdef{\ovar@set@#3}{#4\ovar@set{%
176       no stack macro hook/.code={%
177         \ovar@set{stack macro/.expanded={%
178           \expandafter\expandonce\csname ovar@stack@#2\endcsname%
179           {\expandonce\ovar@length@min}%
180           {\expandonce\ovar@before@arrow}{\expandonce\ovar@after@arrow}%
181         }}%
182       },#5}}%
183     }%
184   }

```

### Common keys

```

185 \SetOverArrowsMethod*{common}{\undef{\ovar@macro@stack}\undef{\ovar@macro@arrow}}{%

```

detect subscripts<sup>→ P. 25</sup>.

```

186   detect subscripts/.is if=ovar@detectsubscripts@,

```

stack macro<sup>→ P. 30</sup> and arrow macro<sup>→ P. 30</sup>.

```

187   stack macro/.store in=\ovar@macro@stack,
188   arrow macro/.store in=\ovar@macro@arrow,
189   stack macro/.value required,
190   arrow macro/.value required,

```

no stack macro hook<sup>→ P. 31</sup>, no arrow macro hook<sup>→ P. 31</sup>. These two keys must be redefined by the command `\ovar@set{method}`.

```

191   no stack macro hook/.code={%
192     \PackageError{overarrows}{Undefined stack macro}
193     {The requested method is perhaps misspelled}
194   },
195   no arrow macro hook/.code={%
196     \PackageError{overarrows}{Undefined arrow macro}
197     {The requested method is perhaps misspelled}
198   },

```

min length<sup>→ P. 23</sup>.

```

199   min length/.store in=\ovar@length@min,
200   min length/.value required,
201   min length=0,

```

before arrow<sup>→ P. 24</sup>, after arrow<sup>→ P. 24</sup>, space before arrow<sup>→ P. 24</sup>, space after arrow<sup>→ P. 24</sup>.

```

202   before arrow/.store in=\ovar@before@arrow,
203   after arrow/.store in=\ovar@after@arrow,
204   before arrow/.value required,
205   after arrow/.value required,
206   before arrow=\empty,
207   after arrow=\empty,
208   space before arrow/.code=\pgfkeysalso{before arrow={\kern ##1}},
209   space after arrow/.code=\pgfkeysalso{after arrow={\kern ##1}},

```

shift left<sup>→P.23</sup>, shift right<sup>→P.23</sup>, shift leftright<sup>→P.24</sup>, center arrow<sup>→P.24</sup>,  
left arrow<sup>→P.24</sup>, right arrow<sup>→P.24</sup>.

```

210 shift left/.store in=\ovar@shift@left,
211 shift right/.store in=\ovar@shift@right,
212 shift left/.value required,
213 shift right/.value required,
214 shift leftright/.code=\pgfkeysalso{%
215     shift left=##1, shift right=##1,
216 },
217 center arrow/.code=\pgfkeysalso{shift leftright=0},
218 shift leftright/.value required,
219 center arrow/.value forbidden,
220 left arrow/.code=\pgfkeysalso{%
221     shift left=0, shift right=##1,
222 },
223 right arrow/.code=\pgfkeysalso{%
224     shift left=##1, shift right=0,
225 },
226 left arrow/.default=2,
227 right arrow/.default=2,
228 right arrow,

```

arrow under<sup>→P.23</sup>.

```

229 arrow under/.is choice,
230 arrow under/noconfig/.code={
231     \def\ovar@stack@fill{\ovar@stackunder@fill}
232     \def\ovar@stack@lens{\ovar@stackunder@lens}
233 },
234 arrow under/autoconfig/.code={
235     \pgfkeysalso{%
236         arrow under=noconfig,
237         detect subscripts=false,
238         before arrow={\kern 1.3\ex@{\relax}},% like underarrow@ from amsmath
239     }
240 },
241 arrow under/.default=autoconfig,
242 }

```

## Keys for the symb method

```

243 \SetOverArrowsMethod{symb}[\undef{\ovar@macro@arrowfill}]{%

```

Fill macro.

```

244 fill macro/.store in=\ovar@macro@arrowfill,
245 fill macro/.value required,

```

Arrow macro.

```

246 no arrow macro hook/.code={%
247     \ifdef{\ovar@macro@arrowfill}{-}{%
248         \ovar@set{%
249             fill macro/.expanded={%
250                 \noexpand\ovar@arrow@fill%
251                 {\expandonce\ovar@shift@left}{\expandonce\ovar@shift@right}%
252             }
253         }
254     }
255     \ovar@set{%
256         arrow macro/.expanded={%
257             \expandonce{\ovar@macro@arrowfill}%
258             {\expandonce{\ovar@arrow@start}\expandonce{\ovar@trim@start}}%
259             {\expandonce{\ovar@trim@middle}\expandonce{\ovar@arrow@middle}%
260             \expandonce{\ovar@trim@middle}}%

```

```

261     {\expandonce{\ovar@trim@end}\expandonce{\ovar@arrow@end}}%
262   }
263 }
264 },

start→ P. 25, middle→ P. 25, end→ P. 25.

265   start/.store in=\ovar@arrow@start,
266   middle/.store in=\ovar@arrow@middle,
267   end/.store in=\ovar@arrow@end,
268   start/.value required,
269   middle/.value required,
270   end/.value required,

trim start→ P. 25, trim middle→ P. 25, trim end→ P. 26, trim→ P. 26, no trimming→ P. 26.

271   trim start/.code={\def\ovar@trim@start{\xjoinrel[##1]}},
272   trim middle/.code={\def\ovar@trim@middle{\xjoinrel[##1]}},
273   trim end/.code={\def\ovar@trim@end{\xjoinrel[##1]}},
274   trim start/.value required,
275   trim middle/.value required,
276   trim end/.value required,
277   trim/.code={\pgfkeysalso{trim start={##1}, trim middle={##1}, trim end={##1}}},
278   trim/.value required,
279   no trimming/.code={%
280     \let\ovar@trim@start\empty
281     \let\ovar@trim@middle\empty
282     \let\ovar@trim@end\empty
283   },
284   no trimming/.value forbidden,

middle config→ P. 26.

285   middle config/.is choice,
286   middle config/.value required,
287   middle config/relbar/.code=\pgfkeysalso{%
288     middle={\relbar},
289     trim middle={2.5},
290   },
291   middle config/relbareda/.code={%
292     \ifundef{\relbareda}{%
293       \PackageWarning{overarrows}{Key 'middle config=relbareda' used,
294         \MessageBreak%
295         but \protect\relbareda\space is undefined; ignored.
296       \MessageBreak%
297       Load 'esvect' package, or use 'esvect' option \MessageBreak%
298       to remove this warning}
299     }{%
300       \pgfkeysalso{%
301         middle={\relbareda},
302         trim middle={1},
303       }
304     }
305   },

306   middle config/harrowextender/.code={%
307     \pgfkeysalso{%
308       middle={\harrowextender},
309       trim middle={0},
310     }
311   },

```

Set middle config with (hopefully) a good configuration. It would be better to reuse the previous middle config settings, but we have to wait for the `begindocument` hook to know which one to use.

```

312 middle config/auto/.code={%
313   \pgfkeysalso{%
314     middle={\ovar@auto@middle},
315     trim middle={\ovar@auto@trim@middle},
316   }
317 },

```

amsmath → P. 26 .

```

318 amsmath/.is choice,%
319 amsmath/mimic/.code=\pgfkeysalso{%
320   start={\relbar}, middle={\relbar}, end={\rightarrow},
321   trim start=7,
322   trim middle=2,
323   trim end=7,
324   shift leftright=0,
325   after arrow={}, before arrow={},
326 },
327 amsmath/strict/.code=\pgfkeysalso{%
328   amsmath=mimic,
329   no trimming,
330   fill macro={\arrowfill@}, stack macro={\overarrow@},
331 },
332 amsmath/.default=mimic,

```

esvect → P. 26 .

```

333 esvect/.is choice,%
334 esvect/mimic/.code=\pgfkeysalso{%
335   start={\relbared}, middle={\relbareda}, end={\fldr},
336   trim start=1.5,
337   trim end=1.5,
338   trim middle=0,
339   right arrow=2,
340   space before arrow=-.7pt,
341   space after arrow=-.3pt,
342 },
343 esvect/strict/.code=\pgfkeysalso{%
344   esvect=mimic,
345   no trimming,
346   fill macro={\traitfill@}, stack macro={\overvect@},
347 },
348 esvect/.default=mimic,

```

Initial configuration.

```

349 amsmath, middle config=auto, end=\ovar@rightarrow, right arrow,
350 }

```

## Keys for the tikz method

```

351 \SetOverArrowsMethod[lens]{tikz}[\undef{\ovar@tikz@command}]{%

```

Arrow macro.

```

352 no arrow macro hook/.code={%
353   \ifdef{\ovar@tikz@command}{-}{%
354     \pgfkeysgetvalue{/overarrows/path options}{\ovar@tikz@pathoptions}
355     \ovar@set{%
356       tikz command/.expanded={%
357         \noexpand\draw[\expandonce\ovar@tikz@pathoptions]\expandonce\ovar@tikz@path;
358       }
359     }
360   }
361   \pgfkeysgetvalue{/overarrows/tikz options}{\ovar@tikz@options}
362   \ovar@set{%

```



```

363     arrow macro/.expanded={%
364         $\noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
365         \noexpand\tikz[\expandonce{\ovar@tikz@options}]{\expandonce{\ovar@tikz@command}}}%
366         $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
367     }
368 }
369 },

```

TikZ parts: tikz command<sup>→ P. 28</sup>, tikz options<sup>→ P. 27</sup>, path options<sup>→ P. 27</sup>, path<sup>→ P. 27</sup>.

```

370     tikz command/.store in=\ovar@tikz@command,
371     tikz options/.initial={x=\overarrowlength, line width=\overarrowthickness},
372     path options/.initial={arrows={-Classical TikZ Rightarrow}, cap=round},
373     path/.store in=\ovar@tikz@path,
374     path={ (0,0)--(1,0) },
375     tikz command/.value required,
376     tikz options/.value required,
377     path options/.value required,
378     path/.value required,

```

TikZ handy keys: add path options<sup>→ P. 27</sup>, add tikz options<sup>→ P. 27</sup>, arrows<sup>→ P. 27</sup>, line thickness<sup>→ P. 27</sup>, thinner<sup>→ P. 28</sup>.

```

379     add path options/.code=\pgfkeysalso{%
380         path options/.append={, ##1}},%
381     add tikz options/.code=\pgfkeysalso{%
382         tikz options/.append={, ##1}},%
383     arrows/.code=\pgfkeysalso{add path options={arrows={##1}}},%
384     line thickness/.code=\pgfkeysalso{add path options={line width=##1}}},%
385     thinner/.code=\pgfkeysalso{line thickness={\overarrowsmallerthickness}}},%
386     add path options/.value required,%
387     add tikz options/.value required,%
388     arrows/.value required,%
389     line thickness/.value required,%
390     thinner/.value forbidden,%

```

Initial configuration.

```

391     shift right=-2,
392     min length=12,
393 }

```

## Keys for the pstricks method

```

394 \SetOverArrowsMethod[lens]{pstricks}{%

```

Arrow macro.

```

395     no arrow macro hook/.code={%
396         \ovar@set{%
397             arrow macro/.expanded={%
398                 $\noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
399                 \noexpand\begin{pspicture}\expandonce{\ovar@pstricks@geometry}%
400                 \noexpand\psset{linewidth=\expandonce{\ovar@pstricks@linethickness}}%
401                 \noexpand\psset{\expandonce{\ovar@pstricks@psset}}}%
402                 \expandonce{\ovar@pstricks@command}}%
403                 \noexpand\end{pspicture}%
404                 $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
405             }
406         }
407     },

```

Pstricks parts: pstricks command<sup>→ P. 28</sup>, psset<sup>→ P. 28</sup>, geometry<sup>→ P. 28</sup>, line thickness<sup>→ P. 29</sup>.

```

408 pstricks command/.store in=\ovar@pstricks@command,
409 psset/.store in=\ovar@pstricks@psset,
410 geometry/.store in=\ovar@pstricks@geometry,
411 line thickness/.store in=\ovar@pstricks@linethickness,
412 pstricks command/.value required,
413 psset/.value required,
414 geometry/.value required,
415 line thickness/.value required,

```

Pstricks handy key: `arrow` <sup>→ P. 28</sup>, `thinner` <sup>→ P. 29</sup>.

```

416 arrow/.style={pstricks command=\psline{##1}(0,0)(\overarrowlength,0)}},%
417 arrow/.value required,%
418 thinner/.style={line thickness=\overarrowsmallerthickness}},%
419 thinner/.value forbidden,%

```

Initial configuration.

```

420 shift right=-2,
421 min length=12,
422 geometry={ (0,-0.5ex)(\overarrowlength,0.5ex)},%
423 line thickness={\overarrowthickness},%
424 arrow={->},%
425 psset={},%
426 }

```

## Keys for the picture method

```

427 \SetOverArrowsMethod[lens]{picture}{%

```

Arrow macro.

```

428 no arrow macro hook/.code={%
429 \ovar@set{%
430 arrow macro/.expanded={%
431 $noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
432 \noexpand\begin{picture}\expandonce{\ovar@picture@geometry}%
433 \noexpand\linethickness{\expandonce{\ovar@picture@linethickness}}}%
434 \expandonce{\ovar@picture@command}%
435 \noexpand\end{picture}%
436 $noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
437 }
438 }
439 },

```

Picture parts: `picture command` <sup>→ P. 29</sup>, `geometry` <sup>→ P. 29</sup>, `line thickness` <sup>→ P. 29</sup>.

```

440 picture command/.store in=\ovar@picture@command,
441 geometry/.store in=\ovar@picture@geometry,
442 line thickness/.store in=\ovar@picture@linethickness,
443 picture command/.value required,
444 geometry/.value required,
445 line thickness/.value required,

```

Picture handy key: `thinner` <sup>→ P. 29</sup>.

```

446 thinner/.code=\pgfkeysalso{line thickness=\overarrowsmallerthickness}},

```

Initial configuration.

```

447 shift right=-2,
448 min length=18,
449 geometry={(\overarrowlength,1ex)(0,-0.5ex)},%
450 line thickness={\overarrowthickness},%
451 picture command={\put(0,0){\vector(1,0){\overarrowlength}}},%
452 }

```

## Commands

### Macros for symbols assemblage

```
\xjoinrel
453 \ifdef{\xjoinrel}{%
454   \PackageWarning{overarrows}{Command \protect\xjoinrel\space already defined.
455   \MessageBreak%
456   Previous definition will be overridden}
457 }
```

Use a default value of 3.5 mu, as recommended by Enrico Gregorio (see <https://tex.stackexchange.com/a/471736>). `\xjoinrel` uses a value of 3 mu.

```
458 \DeclareRobustCommand{\xjoinrel}[1][3.5]{\mathrel{\mkern-#1mu}}
\smallermathstyle
459 \newcommand*{\smallermathstyle}{%
460   \mathchoice{\scriptstyle}{\scriptstyle}{\scriptscriptstyle}{}
461 }
```

`\ovar@arrow@fill`

Macro used for default fill macro<sup>→P.31</sup>.

#1: left shift  
 #2: right shift  
 #3: arrow start  
 #4: arrow middle  
 #5: arrow end  
 #6: math style

```
462 \def\ovar@arrow@fill#1#2#3#4#5#6{%
463   $\m@th\thickmuskip0mu\medmuskip\thickmuskip\thinmuskip\thickmuskip\relax%
464   \mkern #1 mu\relax#6#3%
465   \cleaders\hbox{#6#4}\hfill%
466   #5\mkern #2 mu\relax$%
467 }
```

### Macros for fixed length arrows

Lengths declaration.

```
468 \newlength{\overarrowlength}
469 \newlength{\overarrowthickness}
470 \newlength{\overarrowsmallerthickness}
471 \newlength{\ovar@tempdim}
```

`\ovar@set@arrowlength`

Sets `\overarrowlength`<sup>→P.20</sup>.

#1: min length, in math units  
 #2: math style  
 #3: content

```
472 \def\ovar@set@arrowlength#1#2#3{%
473   \settowidth{\ovar@tempdim}{$\m@th#2\mskip #1 mu\relax$}%
474   \settowidth{\overarrowlength}{$\m@th#2#3$}%
475   \ifdim \overarrowlength < \ovar@tempdim \overarrowlength=\ovar@tempdim\fi%
476 }
```

`\ovar@set@arrowthickness`

Sets `\overarrowthickness`<sup>→P.20</sup> and `\overarrowsmallerthickness`<sup>→P.20</sup>.

`\ovar@set@arrowthickness@UM@lua`

#1: math style

Set to the default rule thickness of the current math style, normally given by `\fontdimen 8 family 3`. With unicode-math, use instead:

- `\fontdimen 54 family 2` with XeTeX,
- `\Umathoverbarrule` with LuaTeX.

```

477 \def\ovar@rulethickness@fontdimen{8}
478 \def\ovar@rulethickness@family{3}
479 \def\ovar@set@arrowthickness#1{%
480   \ifx#1\displaystyle%
481     \overarrowthickness =
482       \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
483     \overarrowsmallerthickness =
484       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
485   \else\ifx#1\textstyle%
486     \overarrowthickness =
487       \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
488     \overarrowsmallerthickness =
489       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
490   \else\ifx#1\scriptstyle%
491     \overarrowthickness =
492       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
493     \overarrowsmallerthickness =
494       \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
495   \else%
496     \overarrowthickness =
497       \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
498     \overarrowsmallerthickness = \overarrowthickness%
499   \fi\fi\fi%
500 }

```

unicode-math with LuaTeX version.

```

501 \def\ovar@set@arrowthickness@UM@lua#1{%
502   \overarrowthickness = \Umathoverbarrule #1
503   \ifx#1\displaystyle%
504     \overarrowsmallerthickness = \Umathoverbarrule \textstyle%
505   \else\ifx#1\textstyle%
506     \overarrowsmallerthickness = \Umathoverbarrule \scriptstyle%
507   \else%
508     \overarrowsmallerthickness = \Umathoverbarrule \scriptscriptstyle%
509   \fi\fi%
510 }

```

Test which version to use.

```

511 \AddToHook{begindocument}[overarrows]
512 {
513   \@ifpackageloaded{unicode-math-luatex}
514   {
515     \global\let\ovar@set@arrowthickness\ovar@set@arrowthickness@UM@lua
516   }
517   {
518     \@ifpackageloaded{unicode-math-xetex}
519     {
520       \gdef\ovar@rulethickness@fontdimen{54}
521       \gdef\ovar@rulethickness@family{2}
522     }
523     {}
524   }
525 }

```

## Stack macros

`\ovar@stackover@@`

`\ovar@stackunder@@`

Bases of all stack macros.

#1: min length, in math units

#2: vertical mode material before arrow

#3: vertical mode material after arrow

#4: arrow  
#5: math style  
#6: content

```

526 \def\ovar@stackover@@#1#2#3#4#5#6{\vbox{\ialign{##\crrc%
527     $#5\mskip #1 mu\relax$\crrc%
528     \noalign{#2\nointerlineskip}#4\crrc%
529     \noalign{#3\nointerlineskip}%
530     $#6\th\hfil#5#6\hfil$\crrc%
531     }%
532     }%
533 }
534 \def\ovar@stackunder@@#1#2#3#4#5#6{\vtop{\ialign{##\crrc%
535     $#6\th\hfil#5#6\hfil$\crrc%
536     \noalign{#2\nointerlineskip}#4\crrc%
537     \noalign{#3\nointerlineskip}%
538     $#5\mskip #1 mu\relax$\crrc%
539     }%
540     }%
541 }

```

\ovar@stackover@

Stack macros without min arrow length.

\ovar@stackunder@

#1: vertical mode material before arrow  
#2: vertical mode material after arrow  
#3: arrow macro  
#4: math style  
#5: content

```

542 \def\ovar@stackover@#1#2#3#4#5{\ovar@stackover@@{0}{#1}{#2}{#3}{#4}{#5}}
543 \def\ovar@stackunder@#1#2#3#4#5{\ovar@stackunder@@{0}{#1}{#2}{#3}{#4}{#5}}

```

\ovar@stackover@fill

Stack macros for extensible arrows.

\ovar@stackunder@fill

\ovar@stack@fill

#1: min length, in math units  
#2: vertical mode material before arrow  
#3: vertical mode material after arrow  
#4: arrow filler macro  
#5: math style  
#6: content

```

544 \def\ovar@stackover@fill#1#2#3#4#5#6{\ovar@stackover@@{#1}{#2}{#3}{#4#5}{#5}{#6}}
545 \def\ovar@stackunder@fill#1#2#3#4#5#6{\ovar@stackunder@@{#1}{#2}{#3}{#4#5}{#5}{#6}}

```

\ovar@stack@fill matches the macro \ovar@stackover@fill by default, or  
\ovar@stackunder@fill with arrow under<sup>P. 23</sup>.

```

546 \def\ovar@stack@fill{\ovar@stackover@fill}

```

\ovar@stackover@lens

Stack macros for fixed-length arrows (these call \ovar@set@arrowlength and  
\ovar@set@arrowthickness).

\ovar@stackunder@lens

\ovar@stack@lens

#1: min length, in math units  
#2: vertical mode material before arrow  
#3: vertical mode material after arrow  
#4: arrow content macro  
#5: math style  
#6: content

```

547 \def\ovar@stackover@lens#1#2#3#4#5#6{%
548     \ovar@set@arrowlength{#1}{#5}{#6}%
549     \ovar@set@arrowthickness{#5}%
550     \ovar@stackover@{#2}{#3}{#4}{#5}{#6}%
551 }

```

```

552 \def\ovar@stackunder@lens#1#2#3#4#5#6{%
553   \ovar@set@arrowlength{#1}{#5}{#6}%
554   \ovar@set@arrowthickness{#5}%
555   \ovar@stackunder@{#2}{#3}{#4}{#5}{#6}%
556 }

```

\ovar@stack@lens matches the macro \ovar@stackover@lens by default, or \ovar@stackunder@lens with arrow under<sup>→ P. 23</sup>.

```

557 \def\ovar@stack@lens{\ovar@stackover@lens}

```

## Macro for commands creation

In the initial version, the commands names must be given as csname (without backslash). To harmonize the syntax with standard \NewDocumentCommand, define an argument processor so that both \NewOverArrowCommand{\myarrow} and \NewOverArrowCommand{myarrow} are accepted.

```

558 \ExplSyntaxOn
559 \cs_new_protected:Npn \__overarrows_processor_strip_escape_char:n #1
560 {
561   \regex_match:nnTF { ~\cC. } { #1 }
562   { \tl_set:Nx \ProcessedArgument { \cs_to_str:N #1 } }
563   { \tl_set:Nx \ProcessedArgument { #1 } }
564 }
565 \cs_new_eq:NN \ovar@cmdname@processor \__overarrows_processor_strip_escape_char:n
566 \ExplSyntaxOff

```

```

\DeclareOverArrowCommand
567 \NewDocumentCommand{\DeclareOverArrowCommand}{
568   0{sybm} >{\ovar@cmdname@processor} m m
569 }{%
570   \begingroup
571   \ovar@set@common
572   \ifcsdef\ovar@set@#1{%
573     \csuse\ovar@set@#1
574   }{%
575     \PackageError{overarrows}{Unknown method #1}
576     {Try with 'sybm', 'tikz', 'pstriks' or 'picture'}
577   }
578   \ovar@set@#3
579   \ifdef{\ovar@macro@arrow}{%
580     \ovar@set{no arrow macro hook}
581   }
582   \ifdef{\ovar@macro@stack}{%
583     \ovar@set{no stack macro hook}
584   }
585   \csxdef\ovar@#2@normal{%
586     \noexpand\mathpalette{%
587       \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}%
588     }
589   }
590   \csxdef\ovar@#2@starred{%
591     \noexpand\mathpalette{%
592       \noexpand\ovar@starversion{%
593         \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}%
594       }
595     }
596   }
597   \ifovar@detectsubscripts0%
598   \csgdef\ovar@#2@auto##1{%
599     \@ifnextchar \ovar@subcmd {%
600       \csuse\ovar@#2@starred}{##1}%

```

```

601     }{%
602     \csuse{ovar@#2@normal}{##1}%
603     }%
604   }
605   \csgdef{#2}{%
606     \ifstar{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@auto}}%
607   }
608   \else
609   \csgdef{#2}{%
610     \ifstar{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@normal}}%
611   }
612   \fi
613   \ifovar@option@debug@
614   \PackageInfo{overarrows}{%
615     Meaning of \protect\ovar@#2@normal\MessageBreak
616     used for \@backslashchar#2:\MessageBreak%
617     \expandafter\meaning\csname ovar@#2@normal\endcsname}
618   \fi
619   \endgroup
620 }

\ProvideOverArrowCommand
621 \NewDocumentCommand{\ProvideOverArrowCommand}{
622   0{symp} >{\ovar@cmdname@processor} m m
623 }{%
624   \ifcsdef{#2}{%
625     \DeclareOverArrowCommand[#1]{#2}{#3}
626   }
627 }

\NewOverArrowCommand
628 \NewDocumentCommand{\NewOverArrowCommand}{
629   0{symp} >{\ovar@cmdname@processor} m m
630 }{%
631   \ifcsdef{#2}{%
632     \PackageError{overarrows}{Command \csname #2\endcsname already defined}%
633     {You have used \protect\NewOverArrowCommand\space with a command that
634       already has a definition. \MessageBreak%
635       Choose another name, or use instead \protect\DeclareOverArrowCommand.}
636   }{%
637     \DeclareOverArrowCommand[#1]{#2}{#3}
638   }
639 }

\RenewOverArrowCommand
640 \NewDocumentCommand{\RenewOverArrowCommand}{
641   0{symp} >{\ovar@cmdname@processor} m m
642 }{%
643   \ifcsundef{#2}{%
644     \PackageError{overarrows}{Command \csname #2\endcsname undefined}%
645     {You have used \protect\RenewOverArrowCommand\space with a command that was
646       never defined. \MessageBreak%
647       Check the requested name, or use instead \protect\NewOverArrowCommand.}
648   }{%
649     \DeclareOverArrowCommand[#1]{#2}{#3}
650   }
651 }

```

## Starred variant

\ovar@starversion

#1: definition (stack macro + arrow macro)  
 #2: math style  
 #3: content

```
652 \newsavebox\ovar@tempbox
```

```

653 \def\ovar@starversion#1#2#3{%
654   \sbox{\ovar@tempbox}{\m@th #1#2{#3}$}%
655   \usebox{\ovar@tempbox}%

```

Remove the extra space added by the arrow.

```

656   \settowidth{\ovar@tempdim}{\m@th #2{#3}$}%
657   \kern\dimeval{0.5\ovar@tempdim - 0.5\wd\ovar@tempbox}%
658 }

```

## \vv vector command

\vv  
Backup and redefinition of esvect \vv<sup>→P. 20</sup> vector command.

```

\esvectvv
659 \ifovar@option@esvect@
660   \let\esvectvv\vv
661   \undef\vv
662 \NewOverArrowCommand{\vv}{esvect, middle config = relbareda}
663 \fi

```

## Predefined commands

Declare predefined commands after unicode-math settings.

```

664 \AddToHook{begindocument}[overarrows]
665 {
\overrightarrow
666   \ifovar@option@overrightarrow@
667   \DeclareOverArrowCommand{\overrightarrow}{%
668     amsmath, middle config=relbar,
669     end=\ovar@rightarrow,
670     right arrow,
671   }
672   \fi
\underrightarrow
673   \ifovar@option@underrightarrow@
674   \DeclareOverArrowCommand{\underrightarrow}{%
675     amsmath, middle config=relbar,
676     end=\ovar@rightarrow,
677     right arrow,
678     arrow under,
679   }
680   \fi
\overleftarrow
681   \ifovar@option@overleftarrow@
682   \DeclareOverArrowCommand{\overleftarrow}{%
683     amsmath, middle config=relbar,
684     start=\ovar@leftarrow,
685     end=\relbar,
686     left arrow,
687   }
688   \fi
\underleftarrow
689   \ifovar@option@underleftarrow@
690   \DeclareOverArrowCommand{\underleftarrow}{%
691     amsmath, middle config=relbar,
692     start=\ovar@leftarrow,
693     end=\relbar,
694     left arrow,
695     arrow under,
696   }
697   \fi
\overleftrightarrow

```



```

698 \ifovar@option@overleftrightharrow@
699 \DeclareOverArrowCommand{\overleftrightharrow}{%
700   amsmath, middle config=relbar,
701   start=\ovar@leftarrow,
702   end=\ovar@rightarrow,
703   center arrow,
704 }
705 \fi

\overleftrightharrow
706 \ifovar@option@overleftrightharrow@
707 \DeclareOverArrowCommand{\overleftrightharrow}{%
708   amsmath, middle config=relbar,
709   start=\ovar@leftarrow,
710   end=\ovar@rightarrow,
711   center arrow,
712   arrow under,
713 }
714 \fi

\overrightharpoonup
715 \ifovar@option@overrightharpoonup@
716 \DeclareOverArrowCommand{\overrightharpoonup}{%
717   amsmath, middle config=relbar,
718   end=\rightharpoonup,
719   right arrow,
720 }
721 \fi

\underrightharpoonup
722 \ifovar@option@underrightharpoonup@
723 \DeclareOverArrowCommand{\underrightharpoonup}{%
724   amsmath, middle config=relbar,
725   end=\rightharpoonup,
726   right arrow,
727   arrow under,
728 }
729 \fi

\overrightharpoondown
730 \ifovar@option@overrightharpoondown@
731 \DeclareOverArrowCommand{\overrightharpoondown}{%
732   amsmath, middle config=relbar,
733   end=\rightharpoondown,
734   right arrow,
735 }
736 \fi

\underrightharpoondown
737 \ifovar@option@underrightharpoondown@
738 \DeclareOverArrowCommand{\underrightharpoondown}{%
739   amsmath, middle config=relbar,
740   end=\rightharpoondown,
741   right arrow,
742   arrow under,
743 }
744 \fi

\overleftharpoonup
745 \ifovar@option@overleftharpoonup@
746 \DeclareOverArrowCommand{\overleftharpoonup}{%
747   amsmath, middle config=relbar,
748   start=\leftharpoonup,
749   end=\relbar,
750   left arrow,
751 }
752 \fi

\underleftharpoonup
753 \ifovar@option@underleftharpoonup@
754 \DeclareOverArrowCommand{\underleftharpoonup}{%
755   amsmath, middle config=relbar,

```

```

756         start=\leftharpoonup,
757         end=\relbar,
758         left arrow,
759         arrow under,
760     }
761 \fi

\overleftharpoonowdown
762 \ifovar@option@overleftharpoonowdown@
763 \DeclareOverArrowCommand{\overleftharpoonowdown}{%
764     amsmath, middle config=relbar,
765     start=\leftharpoonowdown,
766     end=\relbar,
767     left arrow,
768 }
769 \fi

\underleftharpoonowdown
770 \ifovar@option@underleftharpoonowdown@
771 \DeclareOverArrowCommand{\underleftharpoonowdown}{%
772     amsmath, middle config=relbar,
773     start=\leftharpoonowdown,
774     end=\relbar,
775     left arrow,
776     arrow under,
777 }
778 \fi

\overbar
779 \ifovar@option@overbar@
780 \DeclareOverArrowCommand{\overbar}{%
781     amsmath, middle config=relbar,
782     start={\std@minus}, end={\std@minus},% \relbar is defined with \mathsm@sh
783     shift leftright=0,
784     space after arrow=-0.3ex,
785 }
786 \fi

\underbar
787 \ifovar@option@underbar@
788 \DeclareOverArrowCommand{\underbar}{%
789     amsmath, middle config=relbar,
790     start={\vphantom{+}\std@minus}, end={\std@minus},% \relbar is defined with \mathsm@sh
791     shift leftright=0,
792     arrow under,
793     space before arrow=-0.3ex,
794 }
795 \fi

With unicode-math, add \vphantom{+} to get the correct position.

End of \AddToHook{begindocument} hook.
796 }
```

## Test macros

\ovar@testmathstyles

Tabular containing the output of a command for the four math styles and different patterns.

```

797 \newcommand{\ovar@testmathstyles}[2] [] {
798     \begingroup
799     \newcommand*{\ovar@row@teststyle}[1] {%
800         $\displaystyle ##1$
801         & $\textstyle ##1$
802         & $\scriptstyle ##1$
803         & $\scriptscriptstyle ##1$
804         \\

```

```

805 }
806 \renewcommand*{\arraystretch}{1.5}
807 \begin{tabular*}{0.95\linewidth}{@{\extracolsep{\fill}} cccc}
808 \hline
809 \footnotesize\texttt{\texttt{\textbackslash displaystyle}}
810 & \footnotesize\texttt{\texttt{\textbackslash textstyle}}
811 & \footnotesize\texttt{\texttt{\textbackslash scriptstyle}}
812 & \footnotesize\texttt{\texttt{\textbackslash scriptsstyle}}
813 \\
814 \hline
815 \ovar@row@teststyle{\csuse{#2}{v}}
816 \ovar@row@teststyle{\csuse{#2}{AB}}
817 \ovar@row@teststyle{\csuse{#2}{\mathrm{grad}}}
818 \ovar@row@teststyle{\csuse{#2}{my~long~vector}}
819 \IfValueT{#1}{\ovar@row@teststyle{\csuse{#2}{#1}}}
820 \hline
821 \end{tabular*}
822 \endgroup
823 }
\ovar@testkerning
824 \begingroup
825 \ifovar@option@subother@ \catcode \_ =12 \fi
826 \ifovar@option@subactive@ \catcode \_ =13 \fi
827 \gdef\ovar@testkerning#1{%
828 \begin{displaymath}
829 #1\{t\}_\{#1\{u\}_\{#1\{v\}\}
830 \quad
831 #1\{\mathrm{i}\}_0
832 \quad
833 #1\{v\}
834 = #1\{v\}_x + #1\{v\}_y + #1\{v\}_z
835 = v_x #1\{\mathrm{i}\} + v_y #1\{\mathrm{j}\} + v_z #1\{\mathrm{k}\}
836 \end{displaymath}
837 }
838 \endgroup
\TestOverArrow
839 \NewDocumentCommand{\TestOverArrow}{
840 s o >{\ovar@cmdname@processor} m
841 }{%
842 \ifcsdef{#3}{\{
843 \PackageWarning{overarrows}{Unknown name '#3' passed to
844 \protect\TestOverArrow}
845 }
846 \IfBooleanTF{#1}{\{
847 \noindent\framebox{
848 \begin{minipage}{0.95\linewidth}
849 \centering
850 \noindent\textbf{\large%
851 Test of \texttt{\textbackslash#3} and \texttt{\textbackslash#3*} macros}
852 \bigskip\par
853 \textbf{\texttt{\textbackslash#3} for different math styles}
854 \smallskip\par
855 \ovar@testmathstyles[#2]{#3}%
856 \bigskip\par
857 \textbf{\texttt{\textbackslash#3} kerning}
858 \ovar@testkerning{\csuse{#3}}
859 \textbf{\texttt{\textbackslash#3*} kerning}
860 \ovar@testkerning{\csuse{#3}*}
861 \end{minipage}%
862 }\bigskip\par
863 }{\{
864 \ovar@testmathstyles[#2]{#3}%
865 }
866 }

```

# Index

Entries listed in the categories “commands”, “lengths”, and “internal macros” also include references to package implementation.

## Package options

- `allcommands`, 15
- `debug`, 17
- `esvect`, 13
- `esvecta`, 14
- `esvectb`, 14
- `esvectc`, 14
- `esvectd`, 14
- `esvecte`, 14
- `esvectf`, 14
- `esvectg`, 14
- `esvecth`, 14
- `noesvect`, 14
- `old-arrows`, 16
- `overbar`, 15
- `overcommands`, 15
- `overleftarrow`, 15
- `overleftharpoonup`, 15
- `overleftrightharpoonup`, 15
- `overrightarrow`, 15
- `overrightarrowharpoonup`, 15
- `overrightarrowharpoondown`, 15
- `pstarrows`, 17
- `pstricks`, 16
- `subactive`, 17
- `subother`, 17
- `subscripts`, 17
- `tikz`, 16
- `underbar`, 16
- `undercommands`, 15
- `underleftarrow`, 15
- `underleftharpoonup`, 16
- `underleftharpoondown`, 16
- `underleftrightharpoonup`, 16
- `underrightharpoonup`, 15
- `underrightharpoondown`, 16
- `underrightharpoonup`, 16

- `add path options` key, 27
- `add tikz options` key, 27
- `after arrow` key, 24
- `allcommands` package option, 15
- `amsmath` key, 26
- `arrow` key, 28
- `arrow macro` key, 30
- `arrow under` key, 23

- `arrows` key, 27

- `before arrow` key, 24

- `center arrow` key, 24

## Commands

- `\DeclareOverArrowCommand`, 17, 46–50
- `\esvectvv`, 20, 48
- `\NewOverArrowCommand`, 17, 47, 48
- `\overbar`, 21, 34, 50
- `\overleftarrow`, 21, 34, 48
- `\overleftharpoonup`, 21, 34, 50
- `\overleftharpoondown`, 21, 34, 49
- `\overleftrightharpoonup`, 21, 34, 49
- `\overrightarrow`, 21, 34, 48
- `\overrightarrowharpoonup`, 21, 34, 49
- `\overrightarrowharpoondown`, 21, 34, 49
- `\overrightarrowharpoonup`, 21, 34, 49
- `\ProvideOverArrowCommand`, 17, 47
- `\RenewOverArrowCommand`, 17, 47
- `\SetOverArrowsMethod`, 30, 37, 38, 40–42
- `\SetOverArrowsMethod*`, 30
- `\SetOverArrowsSubscriptCommand`, 30, 36
- `\smallermathstyle`, 19, 43
- `\TestOverArrow`, 18, 51
- `\TestOverArrow*`, 18
- `\underbar`, 22, 34, 50
- `\underleftarrow`, 22, 34, 48
- `\underleftharpoonup`, 22, 34, 50
- `\underleftharpoondown`, 22, 34, 50
- `\underleftharpoonup`, 22, 34, 49
- `\underleftrightharpoonup`, 22, 34, 49
- `\underrightharpoonup`, 22, 34, 48
- `\underrightharpoondown`, 22, 34, 49
- `\underrightharpoonup`, 22, 34, 49
- `\vv`, 20, 48
- `\vv*`, 20
- `\xjoinrel`, 19, 39, 43

- `debug` package option, 17

- `\DeclareOverArrowCommand`, 17

- `detect subscripts` key, 25

- `end` key, 25

`esvect` key, 26  
`esvect` package option, 13  
`esvecta` package option, 14  
`esvectb` package option, 14  
`esvectc` package option, 14  
`esvectd` package option, 14  
`esvecte` package option, 14  
`esvectf` package option, 14  
`esvectg` package option, 14  
`esvecth` package option, 14  
`\esvectvv`, 20

`fill` macro key, 31

`geometry` key, 28, 29

Internal macros

`\ifovar@detectsubscripts@`, 33, 46  
`\ifovar@option@debug@`, 33, 47  
`\ifovar@option@esvect@`, 33, 35, 48  
`\ifovar@option@oldarrows@`, 33, 35  
`\ifovar@option@overbar@`, 33, 50  
`\ifovar@option@overleftarrow@`, 33, 48  
`\ifovar@option@overleftharpoondown@`, 33, 50  
`\ifovar@option@overleftharpoonup@`, 33, 49  
`\ifovar@option@overleftrightharpoon@`, 33, 49  
`\ifovar@option@overrightarrow@`, 33, 48  
`\ifovar@option@overrightharpoondown@`, 33, 49  
`\ifovar@option@overrightharpoonup@`, 33, 49  
`\ifovar@option@pstarrows@`, 33, 36  
`\ifovar@option@pstricks@`, 33, 35  
`\ifovar@option@subactive@`, 33, 36, 51  
`\ifovar@option@subother@`, 33, 36, 51  
`\ifovar@option@tikz@`, 33, 35  
`\ifovar@option@underbar@`, 33, 50  
`\ifovar@option@underleftarrow@`, 33, 48  
`\ifovar@option@underleftharpoondown@`, 33, 50  
`\ifovar@option@underleftharpoonup@`, 33, 49

`\ifovar@option@underleftrightharpoon@`, 33, 49  
`\ifovar@option@underrightharpoon@`, 33, 48  
`\ifovar@option@underrightharpoondown@`, 33, 49  
`\ifovar@option@underrightharpoonup@`, 33, 49  
`\ovar@after@arrow`, 37  
`\ovar@arrow@end`, 39  
`\ovar@arrow@fill`, 38, 43  
`\ovar@arrow@middle`, 38, 39  
`\ovar@arrow@start`, 38, 39  
`\ovar@auto@middle`, 36, 40  
`\ovar@auto@trim@middle`, 36, 40  
`\ovar@before@arrow`, 37  
`\ovar@cmdname@processor`, 46, 47, 51  
`\ovar@leftarrow`, 35, 48, 49  
`\ovar@length@min`, 37  
`\ovar@macro@arrow`, 37, 46  
`\ovar@macro@arrowfill`, 38  
`\ovar@macro@stack`, 37, 46  
`\ovar@picture@command`, 42  
`\ovar@picture@geometry`, 42  
`\ovar@picture@linethickness`, 42  
`\ovar@pstricks@command`, 41, 42  
`\ovar@pstricks@geometry`, 41, 42  
`\ovar@pstricks@linethickness`, 41, 42  
`\ovar@pstricks@psset`, 41, 42  
`\ovar@rightarrow`, 35, 40, 48, 49  
`\ovar@row@teststyle`, 50, 51  
`\ovar@rulethickness@family`, 44  
`\ovar@rulethickness@fontdimen`, 44  
`\ovar@set`, 37, 38, 40–42, 46  
`\ovar@set@arrowlength`, 43, 45, 46  
`\ovar@set@arrowthickness`, 44–46  
`\ovar@set@arrowthickness@UM@lua`, 44  
`\ovar@set@common`, 46  
`\ovar@shift@left`, 38, 41, 42  
`\ovar@shift@right`, 38, 41, 42  
`\ovar@stack@fill`, 38, 45  
`\ovar@stack@lens`, 38, 46  
`\ovar@stackover@`, 45  
`\ovar@stackover@@`, 45  
`\ovar@stackover@fill`, 45  
`\ovar@stackover@lens`, 45, 46  
`\ovar@stackunder@`, 45, 46  
`\ovar@stackunder@@`, 45  
`\ovar@stackunder@fill`, 38, 45

- `\ovar@stackunder@lens`, 38, 46
- `\ovar@starversion`, 46, 48
- `\ovar@subcmd`, 36, 46
- `\ovar@tempbox`, 47, 48
- `\ovar@tempdim`, 43, 48
- `\ovar@testkerning`, 51
- `\ovar@testmathstyles`, 50, 51
- `\ovar@tikz@command`, 40, 41
- `\ovar@tikz@options`, 40, 41
- `\ovar@tikz@path`, 40, 41
- `\ovar@tikz@pathoptions`, 40
- `\ovar@trim@end`, 39
- `\ovar@trim@middle`, 38, 39
- `\ovar@trim@start`, 38, 39

Keys

- add path options, 27
- add tikz options, 27
- after arrow, 24
- amsmath, 26
- arrow, 28
- arrow macro, 30
- arrow under, 23
- arrows, 27
- before arrow, 24
- center arrow, 24
- detect subscripts, 25
- end, 25
- esvect, 26
- fill macro, 31
- geometry, 28, 29
- left arrow, 24
- line thickness, 27, 29
- middle, 25
- middle config, 26
- min length, 23
- no arrow macro hook, 31
- no stack macro hook, 31
- no trimming, 26
- path, 27
- path options, 27
- picture command, 29
- psset, 28
- pstricks command, 28
- right arrow, 24
- shift left, 23
- shift leftright, 24
- shift right, 23
- space after arrow, 24
- space before arrow, 24
- stack macro, 30
- start, 25
- thinner, 28, 29
- tikz command, 28
- tikz options, 27
- trim, 26
- trim end, 26
- trim middle, 25
- trim start, 25

left arrow key, 24

Lengths

- `\overarrowlength`, 20, 41–43
- `\overarrowsmallerthickness`, 20, 41–44
- `\overarrowthickness`, 20, 41–44

line thickness key, 27, 29

middle key, 25

middle config key, 26

min length key, 23

`\NewOverArrowCommand`, 17

- no arrow macro hook key, 31
- no stack macro hook key, 31
- no trimming key, 26
- noesvect package option, 14

old-arrows package option, 16

`\overarrowlength` length, 20

`\overarrowsmallerthickness` length, 20

`\overarrowthickness` length, 20

`\overbar`, 21

overbar package option, 15

overcommands package option, 15

`\overleftarrow`, 21

overleftarrow package option, 15

`\overleftharpoondown`, 21

overleftharpoondown package option, 15

`\overleftharpoonup`, 21

overleftharpoonup package option, 15

`\overletrightarrow`, 21

overletrightarrow package option, 15

`\overrightarrow`, 21

overrightarrow package option, 15

`\overrightarrowharpoondown`, 21

overrightarrowharpoondown package option, 15

`\overrightarrowharpoonup`, 21

overrightarrowharpoonup package option, 15

path key, 27

path options key, 27

picture command key, 29

`\ProvideOverArrowCommand`, 17

psset key, 28

pstarrows package option, 17

- `pstricks` package option, 16
- `pstricks` command key, 28
- `\RenewOverArrowCommand`, 17
- `right arrow` key, 24
- `\SetOverArrowsMethod`, 30
- `\SetOverArrowsMethod*`, 30
- `\SetOverArrowsSubscriptCommand`, 30
- `shift left` key, 23
- `shift leftright` key, 24
- `shift right` key, 23
- `\smallermathstyle`, 19
- `space after arrow` key, 24
- `space before arrow` key, 24
- `stack macro` key, 30
- `start` key, 25
- `subactive` package option, 17
- `subother` package option, 17
- `subscripts` package option, 17
- `\TestOverArrow`, 18
- `\TestOverArrow*`, 18
- `thinner` key, 28, 29
- `tikz` package option, 16
- `tikz` command key, 28
- `tikz options` key, 27
- `trim` key, 26
- `trim end` key, 26
- `trim middle` key, 25
- `trim start` key, 25
- `\underbar`, 22
- `underbar` package option, 16
- `undercommands` package option, 15
- `\underleftarrow`, 22
- `underleftarrow` package option, 15
- `\underleftharpoondown`, 22
- `underleftharpoondown` package option, 16
- `\underleftharpoonup`, 22
- `underleftharpoonup` package option, 16
- `\underlefterightarrow`, 22
- `underlefterightarrow` package option, 16
- `\underrightarrow`, 22
- `underrightarrow` package option, 15
- `\underrightharpoondown`, 22
- `underrightharpoondown` package option, 16
- `\underrightharpoonup`, 22
- `underrightharpoonup` package option, 16
- `\vv`, 20
- `\vv*`, 20
- `\xjoinrel`, 19

## Change History

v1.0		
General: Initial version	.....	1
v1.0.1		
General: Bug fix for under* options		34
v1.1		
General: Support for non-standard subscripts	.....	36, 46, 51
v1.2		
General: Add option pstricks	....	33
Add the method pstricks	....	41
Allow backslash in command name	.....	46
Declare predefined commands		
		after <code>unicode-math</code> settings ... 48
		Fix esvect font sizes ..... 35
		Get the correct rule thickness
		with <code>unicode-math</code> ..... 44
		Remove useless
		<code>\AtBeginDocument</code> ..... 36
		Use <code>\def</code> instead of <code>\let</code> for
		<code>\ovar@rightarrow</code> and
		<code>\ovar@leftarrow</code> ..... 35
		Use <code>\harrowextender</code> , if
		available. .... 36
		Use boxes in starred variant for
		better performances ..... 47