

The `overarrows` package*

Julien Labbé

<https://github.com/julienlabbe/latex-packages>

October 19, 2025

Abstract

A \LaTeX 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}} \\
&\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:

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

- to draw arrows of various shapes above math expressions:

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

- to draw arrows of various shapes under math expressions:

$$\underline{AB} \quad \underline{AB} \quad \underline{AB} \quad \underline{AB} \quad \underline{AB} \quad \underline{AB} \quad \underline{AB}.$$

*This document corresponds to `overarrows` v1.5, dated 2025/10/19.

Contents

1	Presentation of the package	4
2	Introduction	4
2.1	Vector arrows	4
2.2	Stack and arrow macros	5
2.3	Extensible arrows	5
3	Quick start	6
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 ^A T _E X picture environment	12
4	User interface	13
4.1	Package options	13
4.1.1	<code>esvect</code> configuration	13
4.1.2	Predefined commands	15
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 .	20
4.2.4	Vectors macros	21
4.2.5	Predefined commands	22
4.3	Keys	24
4.3.1	Arrow position and length settings	24
4.3.2	Subscripts detection setting	26
4.3.3	Symbols assemblage settings	27
4.3.4	TikZ settings	28
4.3.5	PSTricks settings	30
4.3.6	Picture environment settings	31
4.4	Advanced commands and keys	31
4.4.1	Advanced commands	32
4.4.2	Advanced keys	32
5	Complements	33
5.1	Know issues	33
5.1.1	Math font change	33
5.1.2	Detection of non standard subscripts	33
5.2	Package dependencies	34
5.3	Alternatives	34
5.4	Changelog	35

6	Implementation	35
6.1	Management of options	35
6.1.1	Declaration of conditionals	35
6.1.2	Declaration of options	36
6.1.3	Options processing	37
6.2	Package dependencies	37
6.3	Configuration of subscripts detection	38
6.4	Management of keys	39
6.4.1	Family declaration and setters	39
6.4.2	Common keys	39
6.4.3	Keys for the <code>symb</code> method	41
6.4.4	Keys for the <code>tikz</code> method	43
6.4.5	Keys for the <code>pstricks</code> method	44
6.4.6	Keys for the <code>picture</code> method	44
6.5	Commands	45
6.5.1	Macros for symbols assemblage	45
6.5.2	Macros for fixed length arrows	46
6.5.3	Stack macros	47
6.5.4	Macro for commands creation	49
6.5.5	Starred variant	50
6.5.6	<code>\vv</code> vector command	51
6.5.7	Predefined commands	51
6.5.8	Test macros	53
	Index	55

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 styles¹, like in $\overrightarrow{v}_{\vec{a}\vec{b}}$.

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_{cm} or vector \vec{v}_{cm} (no extra space before the subscript, unlike the output of the unstarred variant: \vec{v}_{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, giving², 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 :

$$\underline{\alpha + \beta} \quad \overleftarrow{\alpha + \beta} \quad \overleftrightarrow{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta} \quad \underline{\alpha + \beta}.$$

2 Introduction

2.1 Vector arrows

Vectors are commonly typeset in bold face, or with an arrow above³. For this second convention, $\text{\TeX}/\text{\LaTeX}$ provides the command `\vec`, which accents its content (using the `\mathaccent` command) with the character $\vec{}$ (`\mathchar"017E` in Computer Modern font). But $\vec{}$ isn't extensible⁴, 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 $\text{\TeX}/\text{\LaTeX}$, 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} .

¹`\displaystyle`, `\textstyle`, `\scriptstyle` and `\scriptscriptstyle`.

²Displayed here with the `old-arrows`^{P.16} option.

³See, 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>.

⁴In fact, with the unicode engines \LuaTeX and \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 symbol⁵. 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):

$$\overrightarrow{\textit{long vector}} \quad \overrightarrow{\textit{long vector}}$$

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 \rightarrow .

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):

$$\overrightarrow{\textit{long vector}} \quad \overrightarrow{\textit{long vector}}$$

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.

⁵Using 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 22):

- `\overrightarrow`^{→P.22}
- `\overleftarrow`^{→P.22}
- `\overleftrightarrow`^{→P.23}
- `\overrightarrowharpoonup`^{→P.23}
- `\overrightarrowharpoondown`^{→P.23}
- `\overleftarrowharpoonup`^{→P.23}
- `\overleftarrowharpoondown`^{→P.23}
- `\overbar`^{→P.23}
- `\underrightarrow`^{→P.23}
- `\underleftarrow`^{→P.23}
- `\underleftrightarrow`^{→P.23}
- `\underrightharpoonup`^{→P.23}
- `\underrightharpoondown`^{→P.24}
- `\underleftarrowharpoonup`^{→P.24}
- `\underleftarrowharpoondown`^{→P.24}
- `\underbar`^{→P.24}

Note that the `old-arrows`^{→P.16} 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`^{→P.17}. 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 \myovercmd{v}_{sub} \quad \myovercmd*{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`^{→P.27} and `end`^{→P.27}. 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 \LaTeX . This can be done with the package `amssymb`, by adding in preamble:

```
\usepackage{amssymb}
```

But with the previous definition, the output of `\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`^{P.27} and `trim end`^{P.27}, 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` package⁶. If needed, try to set the `middle`^{P.27} 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`^{P.27}, `trim end`^{P.27} and `trim middle`^{P.27} may need many trials. For this purpose, the `\TestOverArrow`^{P.18} macro displays the result of a command for different lengths and math styles:

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

<code>\TestOverArrow{\overhooktwoheadrightarrow}</code>			
<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 symbol `\Relbar` \Rightarrow and the symbol `\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`^{P.27} sets `trim start`^{P.27}, `trim middle`^{P.27} and `trim end`^{P.27} with the same value.

The previous arrow is visually too big. The macro `\smallermathstyle`^{P.19} 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`^{P.19} should not be used for `end`^{P.27}, because this last is formatted with the same math style as `start`^{P.27}.

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`^{P.26}:

```
\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`^{P.25} and `shift right`^{P.25}. 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`^{P.25} places the arrow below the content, instead of above (and `space before arrow`^{P.26} 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 L^AT_EX math symbols are built by assemblage, using the macro `\joinrel`⁷ which remove 3 math units of horizontal space. The `overarrows` package provides a flexible version of `\joinrel`, called `\xjoinrel`^{P.19}, 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 `\ttail` from the symbol `\succ` :

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



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

⁷For 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{\overtttailrightarrow}{%
  start={\tttail},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overtttailrightarrow{v} \quad \overtttailrightarrow{AB} $

```



Here the `min lengthP.24` 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 `\smallermathstyleP.19`. 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{\overtttailrightarrow}{%
  start={\vcenter{\hbox{$\smallermathstyle\tttail$}}},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overtttailrightarrow{v} \quad \overtttailrightarrow{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`^{→P.16} option to the `overarrows` package⁸:

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

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

- `\overarrowlength`^{→P.21} for the arrow length,
- `\overarrowthickness`^{→P.21} and `\overarrowsmallerthickness`^{→P.21} 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} $
```

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

Keys to use Tikz are described in section 4.3.4, page 28. Main keys are: `tikz options`^{→P.29}, `path options`^{→P.29} and `path`^{→P.29}. It's also possible to append settings with `add tikz options`^{→P.29} and `add path options`^{→P.29}. The full TikZ command used to draw the arrow can as well be entirely redefined with the key `tikz command`^{→P.29}

Here is a example of an arrow drawn with TikZ⁹:

```
\NewOverArrowCommand[tikz]{\overarchedleftrigharrow}{%
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,
}
$ \overarchedleftrigharrow{v} \quad \overarchedleftrigharrow{ABCD} $
```

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

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

⁹TikZ 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`^{→P.17}. Like with `tikz`, the three lengths

- `\overarrowlength`^{→P.21};
- `\overarrowthickness`^{→P.21};
- `\overarrowsmallerthickness`^{→P.21}

can be used in PSTricks commands. By default, a right arrow is drawn:

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

\vec{v} \overrightarrow{AB}

The `pstricks` package has to be loaded (for example, using the `pstricks`^{→P.17} option of the `overarrows` package)

Keys to use PSTricks commands are described in section 4.3.5, page 30. The main keys are `pstricks command`^{→P.30}, `psset`^{→P.30}, `arrow`^{→P.30}, `geometry`^{→P.30} and `line thickness`^{→P.30}. 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^AT_EX picture environment

Without any other package, arrows can also be drawn with the L^AT_EX `picture` environment. In this case, the optional argument `picture` must be passed to `\NewOverArrowCommand`^{→P.17}. As with `tikz` or `pstricks`, the three lengths

- `\overarrowlength`^{→P.21};
- `\overarrowthickness`^{→P.21};
- `\overarrowsmallerthickness`^{→P.21}

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} $
```

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

If `overarrows` is loaded with the option `pstarrows`^{→P.17}, the package `pict2e` is used and a PSTricks style vector arrows is set. This gives:

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

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

Keys to use L^AT_EX `picture` environment are described in section 4.3.6, page 31. The main keys are `picture command`^{→P.31}, `geometry`^{→P.31} and `line thickness`^{→P.31}. 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} $
```

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

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 $\backslash vv$ ^{→P.22} through the `overarrows` mechanism. Original `esvect` $\backslash vv$ macro is still available with $\backslash esvectvv$ ^{→P.22}. 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 `\fldr`. The shape of the arrow depends on the option passed to the `esvect` package: \rightarrow (option a), \rightarrow (option b), \rightarrow (option c), \rightarrow (option d), \rightarrow (option e), \rightarrow (option f), \rightarrow (option g) or \rightarrow (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 `\vv`^{P.22}.

esvecta

Loads the `esvect` package with the a option.

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

esvectb

Loads the `esvect` package with the b option.

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

esvectc

Loads the `esvect` package with the c option.

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

esvectd

Loads the `esvect` package with the d option.

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

esvecte

Loads the `esvect` package with the e option.

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

esvectf

Loads the `esvect` package with the f option.

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

esvectg

Loads the `esvect` package with the g option.

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

esvecth

Loads the `esvect` package with the h option.

`\fldr` corresponds the to the symbol \rightarrow . `\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, these 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`, `\overrightarrow`, `\overleftrightarrow`, `\underleftarrow`, `\underrightarrow` and `\underleftrightarrow` are, for example, part of the `amsmath` 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`^{P.22} command: \vec{v} , \overrightarrow{AB} , $\overrightarrow{\text{grad}}$.

`overleftarrow`

Defines the `\overleftarrow`^{P.22} command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

`overleftrightarrow`

Defines the `\overleftrightarrow`^{P.23} command: \overleftrightarrow{v} , \overleftrightarrow{AB} , $\overleftrightarrow{\text{grad}}$.

`overrightarrowharpoonup`

Defines the `\overrightarrowharpoonup`^{P.23} command: \overrightarrow{v} , \overrightarrow{AB} , $\overrightarrow{\text{grad}}$.

`overrightarrowharpoondown`

Defines the `\overrightarrowharpoondown`^{P.23} command: \overrightarrow{v} , \overrightarrow{AB} , $\overrightarrow{\text{grad}}$.

`overleftarrowharpoonup`

Defines the `\overleftarrowharpoonup`^{P.23} command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

`overleftarrowharpoondown`

Defines the `\overleftarrowharpoondown`^{P.23} command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

`overbar`

Defines the `\overbar`^{P.23} command: \overline{v} , \overline{AB} , $\overline{\text{grad}}$.

Under arrows

underrightarrow

Defines the $\backslashunderrightarrow^{\rightarrow P.23}$ command: \underline{v} , \underline{AB} , $\underline{\text{grad}}$.

underleftarrow

Defines the $\backslashunderleftarrow^{\rightarrow P.23}$ command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

underleftrightharpoonup

Defines the $\backslashunderleftrightharpoonup^{\rightarrow P.23}$ command: \overleftrightarrow{v} , \overleftrightarrow{AB} , $\overleftrightarrow{\text{grad}}$.

underrightharpoonup

Defines the $\backslashunderrightharpoonup^{\rightarrow P.23}$ command: $\underline{\overrightarrow{v}}$, $\underline{\overrightarrow{AB}}$, $\underline{\overrightarrow{\text{grad}}}$.

underrightharpoondown

Defines the $\backslashunderrightharpoondown^{\rightarrow P.24}$ command: $\underline{\overleftarrow{v}}$, $\underline{\overleftarrow{AB}}$, $\underline{\overleftarrow{\text{grad}}}$.

underleftharpoonup

Defines the $\backslashunderleftharpoonup^{\rightarrow P.24}$ command: $\overleftarrow{\underline{v}}$, $\overleftarrow{\underline{AB}}$, $\overleftarrow{\underline{\text{grad}}}$.

underleftharpoondown

Defines the $\backslashunderleftharpoondown^{\rightarrow P.24}$ command: $\overleftarrow{\underline{\overrightarrow{v}}}$, $\overleftarrow{\underline{\overrightarrow{AB}}}$, $\overleftarrow{\underline{\overrightarrow{\text{grad}}}}$.

underbar

Defines the $\backslashunderbar^{\rightarrow P.24}$ 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 \backslashvarleftarrow $\overleftarrow{\hspace{1cm}}$ and \backslashvarrightarrow $\overrightarrow{\hspace{1cm}}$, used then by default for predefined command.

When the `old-arrows` option is set, the commands $\backslashoverrightarrow^{\rightarrow P.22}$, $\backslashoverleftarrow^{\rightarrow P.22}$, $\backslashoverleftrightharpoonup^{\rightarrow P.23}$, $\backslashunderrightarrow^{\rightarrow P.23}$, $\backslashunderleftarrow^{\rightarrow P.23}$ and $\backslashunderleftrightharpoonup^{\rightarrow P.23}$ give respectively : \overrightarrow{AB} , \overleftarrow{AB} , \overleftrightarrow{AB} , $\underline{\overrightarrow{AB}}$, $\underline{\overleftarrow{AB}}$ and $\underline{\overleftrightarrow{AB}}$

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 \LaTeX , \Lua\LaTeX and \Xe\LaTeX , but not with \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`^{P.26} to `true`.

This option also impacts the command `\vv`^{P.22} 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`^{P.26} (see the section 5.1.2, page 33).

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`^{P.26} (see the section 5.1.2, page 33).

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 $\langle command \rangle$ and its starred variant $\langle command \rangle^*$. The starred variant $\langle command \rangle^*$ removes the extra end space generated by the arrow, which is suitable, as example, when a subscript follows.

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

Updated: v1.2 2024/07/11

Since the version 1.4, the $\langle command \rangle$ is robust (for example, it can be used safely inside `\caption`).

Updated: v1.4 2025/04/30

`\NewOverArrowCommand` raises an error if $\langle command \rangle$ is already defined.

`\RenewOverArrowCommand` raises an error if $\langle command \rangle$ is undefined.

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

`\DeclareOverArrowCommand` sets $\langle command \rangle$, whether the command is already defined or not, without raising any error.

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

`symbol` to draw the arrow by symbols assemblage (default);

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

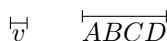
`pstricks` to draw the arrow with PSTricks;

`picture` to draw the arrow with the \LaTeX `picture` environment.

With no $\langle method \rangle$ argument, the `symbol` 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 24.

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

New: v1.2 2024/07/11

Updated: v1.2 2024/07/11

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

Test of `\vv` and `\vv*` macros

`\vv` for different math styles

<code>\displaystyle</code>	<code>\textstyle</code>	<code>\scriptstyle</code>	<code>\scriptscriptstyle</code>
\vec{v}	\vec{v}	\vec{v}	\vec{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

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

`\vv*` kerning

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

4.2.2 Useful macros for symbols assemblage

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

`\xjoinrel[⟨number⟩]`

Removes an horizontal space of $\langle number \rangle$ 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 \$}

```

$\equiv \quad \Rightarrow$
 $\equiv\equiv\Rightarrow$

`\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

`\arrowmathstyle`

New: v1.5 2025/10/19

Expands to the current math style. This could be useful inside box macros, such as, for example, `\raisebox`, `\rotatebox` or `\scalebox` (these last two are defined by the package `graphicx`). `\arrowmathstyle` is only available inside commands created with `\NewOverArrowCommand`^{P.17} and variants.

```

% ^A \rotatebox is defined with \usepackage{graphicx}
\NewOverArrowCommand{\heartarrow}{%
  start={\raisebox{\dimeval{-0.5\height+0.5\depth}}{\$\arrowmathstyle\gg$}},
  end={\rotatebox[origin=cB]{+90}{\$\arrowmathstyle\heartsuit$}},
  middle={\baselinebar},
  trim start=7,
  trim end=2.5,
  trim middle=4,
  shift leftright=-6,
}
$ \displaystyle \heartarrow{Love}
\qqquad \textstyle \heartarrow{Love}
\qqquad \scriptstyle \heartarrow{Love}
\qqquad \scriptscriptstyle \heartarrow{Love} $

```

`\baselinebar`

New: v1.5 2025/10/19

Draws a zero-height horizontal bar `—`, aligned with the baseline (analogous to `\relbar` defined by \LaTeX , and redefined by the `amsmath` package, but centered on the baseline). `\baselinebar` should work with most fonts. It is used in the definition of `\overbar`^{P.23} and `\underbar`^{P.24}.

For example, the `\overbar`^{P.23} command is defined with:

```

\DeclareOverArrowCommand{\overbar}{%
  amsmath, middle={\baselinebar},
  start={\baselinebar}, end={\baselinebar},
  space before arrow={0.5\overarrowthickness},
  space after arrow={3.5\overarrowthickness}
}

```

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.

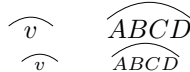
```

\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,
}

$$\overparabola{v} \quad \overparabola{ABCD}$$


$$\scriptstyle \overparabola{v} \quad \overparabola{ABCD}$$


```



\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`^{P.24}.

\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 2024/07/11

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

Updated: v1.5 2025/10/19

Since the version 1.5, the length `\overarrowthickness` is always available, whatever drawing method is used for the arrow (including arrows drawn by symbols assemblage).

$\overarrowsmallerthickness$

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 2024/07/11

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

Updated: v1.5 2025/10/19

Since the version 1.5, the length `\overarrowsmallerthickness` is always available, whatever drawing method is used for the arrow (including arrows drawn by symbols assemblage).

4.2.4 Vectors macros

The macro `\vv`, dedicated to vectors, is automatically defined when the option `esvect`^{P.13} 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`^{→P.14}, `esvectb`^{→P.14}, `esvectc`^{→P.14}, `esvectd`^{→P.14}, `esvecte`^{→P.14}, `esvectf`^{→P.14}, `esvectg`^{→P.14}, `esvecth`^{→P.14}.

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

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

$$\vec{i}_0 \quad \vec{e}_r \quad \vec{L}_\Delta$$

$$\vec{i}_0 \quad \vec{e}_r \quad \vec{L}_\Delta$$

`\esvectvv`

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

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

$$\vec{i}_0 \quad \vec{e}_r \quad \vec{L}_\Delta$$

$$\vec{i}_0 \quad \vec{e}_r \quad \vec{L}_\Delta$$

4.2.5 Predefined commands

Predefined commands are defined if the corresponding option is set (see section 4.1.2, page 15). The commands

- `\overrightarrow`;
- `\overleftarrow`;
- `\overleftrightarrow`;
- `\underrightarrow`;
- `\underleftarrow`;
- `\underleftrightarrow`

are affected by the option `old-arrows`^{→P.16}.

Over arrows

`\overrightarrow`

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

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

`\overleftarrow`

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

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

`\overleftrightharpoonarrow`

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

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

`\overrightarrow`

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

`\overleftarrow`

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

`\overleftarrowtharpoonup`

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

`\overleftarrowtharpoonordown`

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

`\overbar`

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

Updated: v1.5 2025/10/19

Since version 1.5, the `\overbar` command can be properly nested.

$$\overbar{\overbar{AB}} \quad x_{\overbar{\overbar{AB}}}$$

Under arrows

`\underrightarrow`

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

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

`\underleftarrow`

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

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

`\underleftrightharpoonarrow`

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

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

`\underrightharpoonup`

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

`\underrightharpoondown`

$\underset{\curvearrowright}{v}$ $\underset{\curvearrowright}{AB}$ $\underset{\curvearrowright}{grad}$

`\underleftharpoonup`

$\underset{\curvearrowleft}{v}$ $\underset{\curvearrowleft}{AB}$ $\underset{\curvearrowleft}{grad}$

`\underleftharpoondown`

$\underset{\curvearrowleft}{\curvearrowright}v$ $\underset{\curvearrowleft}{\curvearrowright}AB$ $\underset{\curvearrowleft}{\curvearrowright}grad$

`\underbar`

\underline{v} \underline{AB} \underline{grad}

Updated: v1.5 2025/10/19

Since version 1.5, the `\underbar` command can be properly nested.

$\underline{\underline{AB}}$ $x_{\underline{\underline{AB}}}$

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 `ymb` 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} \quad \overlongarrow{ABCDEF} $
```

\overlongarrow{v} \overlongarrow{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`^{→P.26} to `false` and the key before `arrow`^{→P.26} 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} \quad \underhooks{AB} $
```

\underline{v} \underline{AB}

Horizontal shifts

shift left={⟨number⟩} (no default, initially 2)

Shifts the left side of the arrow by ⟨number⟩ math units (positive number means a shift to the right).

shift right={⟨number⟩} (no default, see below for the initial value)

Shifts the right side of the arrow by ⟨number⟩ math units (positive number means a shift to the left).

The initial value of `shift right` depends on the ⟨method⟩ chosen at command creation (see section 4.2.1, page 17 for the documentation of commands creation):

- ⟨number⟩ = 0 for the `symb` method (method by default);
- ⟨number⟩ = -2 for the `tikz`, `pstricks` and `picture` methods.

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

$\overleftarrow{\text{look back}}$

shift leftright={⟨number⟩} (no default)

Sets `shift left` and `shift right` to the same ⟨number⟩ value.

center arrow

Sets `shift left` and `shift right` to zero.

left arrow={⟨number⟩} (default 2)

Sets `shift left` to zero and `shift right` to ⟨number⟩.

right arrow={*number*} (default 2)

Sets `shift right` to zero and `shift left` to *number*.

Vertical adjunct

before arrow={*vertical material*} (initially empty)

after arrow={*vertical material*} (initially empty)

Adds the *vertical material* before or after the arrow.

Over and under arrow commands are typeset through the \TeX `\ialign` command, which aligns contents, like a tabular. The *vertical material* is inserted *between* the rows, with \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={*length*} (no default)

Adds a space of *length* before the arrow. This sets the keys `before arrow`.

space after arrow={*length*} (no default)

Adds a space of *length* after the arrow. This sets the keys `after arrow`.

Updated: v1.5 2025/10/19

Since the version 1.5, the length `\overarrowthickness`^{P.21} can be used to set `space before arrow` or `space after arrow`, whatever drawing method is chosen, including the symbols assemblage. The main interest is that `\overarrowthickness`^{P.21} scale with the math style and the main font.

```
\NewOverArrowCommand{\overharpoonsdown}{%
  start=\leftharpoonsdown, end=\rightharpoonsdown, center arrow,
  space before arrow=-\overarrowthickness,
  space after arrow=3\overarrowthickness,
}
$ \dot{\overharpoonsdown{v}} \quad \ddot{\overharpoonsdown{AB}} \quad \quad
\scriptstyle \dot{\overharpoonsdown{v}} \quad \ddot{\overharpoonsdown{AB}} $
```

$$\dot{\overharpoonsdown{v}} \quad \ddot{\overharpoonsdown{AB}} \quad \scriptstyle \dot{\overharpoonsdown{v}} \quad \scriptstyle \ddot{\overharpoonsdown{AB}}$$

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`^{P.17} 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 33).

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

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

4.3.3 Symbols assemblage settings

The following keys are available for arrows drawn with the default `symp` 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` \rightarrow . When the option `old-arrows`^{P.16} is set, the initial value of `end` is `\varrightarrow` \rightarrow .

`start` and `end` symbols are typeset in the same group. `middle` is typeset alone. This means that, if a command, like `\smallermathstyle`^{P.19}, 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} \qqquad \smalleroverrightarrow{AB} $
```

\overrightarrow{v} \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`={*number*} (no default, initially 7)

Trims *number* math units from the left side of the `end` symbol.

`trim`={*number*} (no default)

Sets `trim start`, `trim middle` and `trim end` to the same *number* value.

no trimming


Clears `trim start`, `trim middle` and `trim end`.


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

Sets a suitable configuration for the keys `middle` and `trim middle`:

New: v1.2 2024/07/11

Updated: v1.2 2024/07/11

For `middle config = relbar`, `middle` is set to `\relbar`  and `trim middle` to 2.5.

For `middle config = relbareda`, `middle` is set to `\relbareda`  and `trim middle` to 1.

For `middle config = harrowextender`, `middle`^{P.27} is set to `\harrowextender` and `trim middle`^{P.27} to 0.

For `middle config = auto`, if `\harrowextender` is provided by the `math font`¹⁰, `middle`^{P.27} is set with `middle config = harrowextender`. If `\harrowextender` isn't available, `middle`^{P.27} is set with `middle config = relbareda` if the option `esvect`^{P.13} 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:

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

`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:

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

`esvect=strict` makes, in addition, the command uses the internal macros of the `esvect \vv` command. this imply

```
no trimming
fill macro={\traitfill@}
stack macro={\overvect@}
```

Note that many configuration keys becomes ineffective.

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:

¹⁰See the documentation of the package `unicode-math`.

`\tikz[<tikz options>]{<tikz command>}`

where `tikz options`^{P.29} and `tikz command`^{P.29} are two keys described below. When `tikz command` is let unset, the drawing command turns into:

`\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`.

```
\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 \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`={*TikZ command*} (initially unset)

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

4.3.5 PSTricks settings

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 by:

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

where `geometry`, `line thickness`, `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} \qqquad \overloopandarrow{AB} $
```

\vec{v} \overrightarrow{AB}

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

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

Sets the `pspicture` command to *<pstricks command>*.

`arrow`={*<arrow>*} (no default, initially `->`)

Sets `pstricks command` with `\psline{<arrow>}(0,0)(\overarrowlength,0)`.

`psset`={*<pstricks setting>*} (no default, initially empty)

Sets *<pstricks setting>* with `\psset`.

`geometry`={*<pstricks geometry specification>*}
(no default, initially `(0,-0.5ex)(\overarrowlength,1ex)`)

Sets the `pspicture` geometry to *<pstricks geometry specification>*.

`line thickness`={*<length>*} (no default)

Sets the line thickness to *<length>*.

`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 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} \qqquad \overarc{AB} $
```



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

picture command={*<picture command>*}
(no default, initially `\put(0,0){\vector(1,0){\overarrowlength}}`)

Sets picture command to *<picture command>*.

geometry={*<picture geometry specification>*}
(no default, initially `(\overarrowlength,1ex)(0,-0.5ex)`)

Sets picture geometry to *<picture geometry specification>*.

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

Sets the picture line thickness to *<length>*.

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⟩}`

Sets to `⟨command⟩` the command used for subscript detection, when this is enabled by the key `detect_subscripts`^{→P.26} (see the section 5.1.2, page 33).

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

Defines the method `⟨name⟩`, to be used with `\NewOverArrowCommand`^{→P.17}, with `\RenewOverArrowCommand`^{→P.17}, with `\ProvideOverArrowCommand`^{→P.17}, or with `\DeclareOverArrowCommand`^{→P.17}. When the `⟨name⟩` method is chosen, corresponding keys are defined by `⟨keys def⟩`. This must set, in particular, the keys `no stack macro hook` and `no arrow macro hook`^{→P.33}. Optional code `⟨pre code⟩` is evaluated before the keys definition.

The unstarred variant automatically defines the key `no stack macro hook`, 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`^{→P.33}) 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 `symp` method.

`lens` if `arrow macro` creates fixed-length arrows, and needs the computation of lengths `\overarrowlength`^{→P.21}, `\overarrowthickness`^{→P.21} and `\overarrowsmallerthickness`^{→P.21}. In this case, the arrow macro (defined by `no arrow macro hook`^{→P.33}) 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`.

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={⟨code⟩}` (no default)

Sets the `⟨code⟩` executed if `stack macro` is left unset, after user evaluation of `⟨keys⟩` in `\NewOverArrowCommand`^{→P.17}, `\RenewOverArrowCommand`^{→P.17}, `\ProvideOverArrowCommand`^{→P.17} or `\DeclareOverArrowCommand`^{→P.17}. `⟨code⟩` must configure `stack macro` accordingly to the user keys setting.

no arrow macro hook={*code*} (no default)

Sets the *code* executed if **arrow macro** is left unset, after user evaluation of *keys* in `\NewOverArrowCommand`^{P.17}, `\RenewOverArrowCommand`^{P.17}, `\ProvideOverArrowCommand`^{P.17} or `\DeclareOverArrowCommand`^{P.17}. *code* must configure **arrow macro**^{P.32} accordingly to the user keys setting.

fill macro={*definition*} (no default, initially unset)

Defines the fill macro to be *definition*. The fill macro is used by arrows created with the **symb** method, to set **arrow macro**^{P.32} 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. *definition* 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 **symb** 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`^{P.14} option.

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

5.1.2 Detection of non standard subscripts

The subscript detection enabled by the key `detect subscripts`^{P.26} is based on the \LaTeX macro `\@ifnextchar`. The detection may fail if the standard subscript command is modified or altered. This is the case, as example:

- with the `spbmark` package (<https://www.ctan.org/pkg/spbmark>), by Qu Yi, which allows a complete customisation of subscripts, through the `\sub` command;
- with the `altnumsub` package (<https://www.ctan.org/pkg/altnumsub>), 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 redefined with `\SetOverArrowsSubscriptCommand`^{P.32}. Compatibility with the `spbmark` package is then obtained by:

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

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

```
\SetOverArrowsSubscriptCommand{_}
```

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

Alternatively, two package options handle the cases where the catcode of the underscore “_” symbol is changed: `subother`^{P.17} (for catcode 12, or *other*) and `subactive`^{P.17} (for catcode 13, or *active*). Hence, setting the `subother`^{P.17} option is sufficient for compatibility with the `altnsubsup` package (no need of `\SetOverArrowsSubscriptCommand`^{P.32}). Note, that with options `subother`^{P.17} and `subactive`^{P.17}, the command `\TestOverArrow*`^{P.18} 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`^{P.14} is used)
- `old-arrows` (when the option `old-arrows`^{P.16} is used)
- `tikz` (when the `tikz` method or the option `tikz`^{P.16} is used)
- `pict2e` (when the option `pstarrows`^{P.17} is used)

L^AT_EX 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 Soudrais, 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` command 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

- v1.5
- Provide the command `\baselinebar`^{→P.20}.
 - Redefine `\overbar`^{→P.23} and `\underbar`^{→P.24} so that they can be nested properly.
 - Make always available the lengths `\overarrowthickness`^{→P.21} and `\overarrowsmallerthickness`^{→P.21}.
 - Provide the command `\arrowmathstyle`^{→P.20}.
- v1.4 Make commands robust.
- v1.3 Bug fix for `esvect` options (see <https://github.com/julienlabbe/latex-packages/issues/2>).
- v1.2
- Fix compatibility issues with `unicode-math`.
 - Allow to draw the arrow with `PSTricks`.
 - Make `esvect` handle all font sizes.
 - Allow backslash in command name for `\NewOverArrowCommand` and variants.
 - 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

```
1 \RequirePackage{etoolbox}
```

6.1 Management of options

6.1.1 Declaration of conditionals

```
2 \newif\ifovar@option@oldarrows@
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@overleftrightharrow@
15 \newif\ifovar@option@underleftrightharrow@
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@

```

6.1.2 Declaration of options

```

26 \def\ovar@option@esvect{f}
27 \DeclareOption{esvect}{\gdef\ovar@option@esvect{f}}
28 \DeclareOption{noesvect}{\gundef\ovar@option@esvect}
29 \DeclareOption{esvecta}{\gdef\ovar@option@esvect{a}}
30 \DeclareOption{esvectb}{\gdef\ovar@option@esvect{b}}
31 \DeclareOption{esvectc}{\gdef\ovar@option@esvect{c}}
32 \DeclareOption{esvectd}{\gdef\ovar@option@esvect{d}}
33 \DeclareOption{esvecte}{\gdef\ovar@option@esvect{e}}
34 \DeclareOption{esvectf}{\gdef\ovar@option@esvect{f}}
35 \DeclareOption{esvectg}{\gdef\ovar@option@esvect{g}}
36 \DeclareOption{esvecth}{\gdef\ovar@option@esvect{h}}
37 \DeclareOption{old-arrows}{\ovar@option@oldarrows@true}
38 \DeclareOption{tikz}{\ovar@option@tikz@true}
39 \DeclareOption{pstricks}{\ovar@option@pstricks@true}
40 \DeclareOption{pstarrows}{\ovar@option@pstarrows@true}
41 \DeclareOption{subscripts}{\ovar@option@detectsubscripts@true}
42 \DeclareOption{subother}{\ovar@option@subother@true}
43 \DeclareOption{subactive}{\ovar@option@subactive@true}
44 \DeclareOption{debug}{\ovar@option@debug@true}

```

Following options are for predefined commands.

```

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

```

Following options are for sets of predefined commands.

```

61 \DeclareOption{overcommands}{%
62   \ovar@option@overrightarrow@true
63   \ovar@option@overleftarrow@true
64   \ovar@option@overleftrightharrow@true
65   \ovar@option@overrightharpoonup@true
66   \ovar@option@overrightharpoondown@true
67   \ovar@option@overleftharpoonup@true
68   \ovar@option@overleftharpoondown@true

```

```

69 \ovar@option@overbar@true
70 }
71 \DeclareOption{undercommands}{%
72 \ovar@option@underrightarrow@true
73 \ovar@option@underleftarrow@true
74 \ovar@option@underleftrightharpoon@true
75 \ovar@option@underrightharpoonup@true
76 \ovar@option@underrightharpoondown@true
77 \ovar@option@underleftharpoonup@true
78 \ovar@option@underleftharpoondown@true
79 \ovar@option@underbar@true
80 }
81 \DeclareOption{allcommands}{%
82 \ovar@option@overrightarrow@true
83 \ovar@option@underrightarrow@true
84 \ovar@option@overleftarrow@true
85 \ovar@option@underleftarrow@true
86 \ovar@option@overleftrightharpoon@true
87 \ovar@option@underleftrightharpoon@true
88 \ovar@option@overrightharpoonup@true
89 \ovar@option@underrightharpoonup@true
90 \ovar@option@overrightharpoondown@true
91 \ovar@option@underrightharpoondown@true
92 \ovar@option@overleftharpoonup@true
93 \ovar@option@underleftharpoonup@true
94 \ovar@option@overleftharpoondown@true
95 \ovar@option@underleftharpoondown@true
96 \ovar@option@overbar@true
97 \ovar@option@underbar@true
98 }

```

6.1.3 Options processing

```

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

```

6.2 Package dependencies

L^AT_EX distributions prior to 2020/10/01 must add the xparse package.

etoolbox is loaded at the very start of the package, as `\gundef` is used at options processing.

```

101 \RequirePackage{amsmath}

```

Option `old-arrows`^{→P. 16}. 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`^{→P. 13}.

```

109 \ifdefined\ovar@option@esvect
110 \PassOptionsToPackage{\ovar@option@esvect}{esvect}
111 \RequirePackage{esvect}

```

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

```

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

```

Option `tikz` ^{→ P. 16}.

```

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

```

Option `pstricks` ^{→ P. 17}.

```

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

```

Option `pstarrows` ^{→ P. 17}.

```

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

```

Add hook rules to apply settings after `unicode-math`.

```

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

```

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

```

134 \AddToHook{begindocument}[overarrows]
135   {%
136   \ifdef{\relbareda}
137     {%
138     \gdef\ovar@auto@middle{\relbareda}%
139     \gdef\ovar@auto@trim@middle{1}%
140     }%
141   {%
142   \gdef\ovar@auto@middle{\relbar}%
143   \gdef\ovar@auto@trim@middle{2.5}%
144   }%
145   \@ifpackageloaded{unicode-math}
146   {%

```

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

```

147   \check@mathfonts
148   \iffontchar\textfont\tw@\string"23AF
149   \gdef\ovar@auto@middle{\mathrel\harrowextender}%
150   \gdef\ovar@auto@trim@middle{0}%
151   \fi
152   }%
153   {}%
154 }

```

6.3 Configuration of subscripts detection

`\SetOverArrowsSubscriptCommand`

Sets the subscript command.

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

Initial configuration.

```
156 \SetOverArrowsSubscriptCommand{}
```

Option `subother`^{→P.17} for *other* (catcode 12) subscript commands.

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

Option `subactive`^{→P.17} for *active* (catcode 13) subscript commands.

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

6.4 Management of keys

6.4.1 Family declaration and setters

```
169 \RequirePackage{pgfkeys}
170 \pgfkeys{overarrows/.is family}
```

```
\ovar@set 171 \newcommand{\ovar@set}[1]{\pgfkeys{/overarrows}#{1}}
```

```
\SetOverArrowsMethod 172 \NewDocumentCommand{\SetOverArrowsMethod}{ s O{fill} m O{ m } }{%
173   \IfBooleanTF{#1}{%
174     \csgdef{\ovar@set@#3}{#4\ovar@set{#5}}%
175   }{%
176     \csgdef{\ovar@set@#3}{#4\ovar@set{%
177       no stack macro hook/.code={%
178         \ovar@set{stack macro/.expanded={%
179           \expandafter\expandonce\csname ovar@stack@#2\endcsname%
180           {\expandonce\ovar@length@min}%
181           {\expandonce\ovar@before@arrow}{\expandonce\ovar@after@arrow}%
182         }}%
183       },#5}}%
184   }%
185 }
```

6.4.2 Common keys

```
186 \SetOverArrowsMethod*{common}[\undef{\ovar@macro@stack}\undef{\ovar@macro@arrow}]{%
```

`detect subscripts`^{→P.26}.

```
187   detect subscripts/.is if=ovar@detectsubscripts@,
```

`stack macro`^{→P.32} and `arrow macro`^{→P.32}.

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

`no stack macro hook`^{→P.32}, `no arrow macro hook`^{→P.33}. These two keys must be redefined by the command `\ovar@set@{method}`.

```

192 no stack macro hook/.code={%
193   \PackageError{overarrows}{Undefined stack macro}
194   {The requested method is perhaps misspelled}%
195 },
196 no arrow macro hook/.code={%
197   \PackageError{overarrows}{Undefined arrow macro}
198   {The requested method is perhaps misspelled}%
199 },

```

min length^{→P.24}.

```

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

```

before arrow^{→P.26}, after arrow^{→P.26}, space before arrow^{→P.26}, space after arrow^{→P.26}.

```

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

```

shift left^{→P.25}, shift right^{→P.25}, shift leftright^{→P.25}, center arrow^{→P.25}, left arrow^{→P.25}, right arrow^{→P.26}.

```

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

```

arrow under^{→P.25}.

```

230 arrow under/.is choice,
231 arrow under/noconfig/.code={%
232   \def\ovar@stack@fill{\ovar@stackunder@fill}%
233   \def\ovar@stack@lens{\ovar@stackunder@lens}%
234 },
235 arrow under/autoconfig/.code={%
236   \pgfkeysalso{%
237     arrow under=noconfig,
238     detect subscripts=false,
239     before arrow={\kern 1.3\ex@}\relax},% like underarrow@ from amsmath
240   }%
241 },
242 arrow under/.default=autoconfig,
243 }

```


6.4.3 Keys for the symb method

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

Fill macro.

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

Arrow macro.

```
247 no arrow macro hook/.code={%
248 \ifdef{\ovar@macro@arrowfill}{-}{%
249 \ovar@set{%
250 fill macro/.expanded={%
251 \noexpand\ovar@arrow@fill%
252 {\expandonce\ovar@shift@left}{\expandonce\ovar@shift@right%
253 }%
254 }%
255 }%
256 \ovar@set{%
257 arrow macro/.expanded={%
258 \expandonce{\ovar@macro@arrowfill}%
259 {\expandonce{\ovar@arrow@start}\expandonce{\ovar@trim@start}}%
260 {\expandonce{\ovar@trim@middle}\expandonce{\ovar@arrow@middle}%
261 \expandonce{\ovar@trim@middle}}%
262 {\expandonce{\ovar@trim@end}\expandonce{\ovar@arrow@end}}%
263 }%
264 }%
265 },
```

start^{→ P. 27}, *middle*^{→ P. 27}, *end*^{→ P. 27}.

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

trim start^{→ P. 27}, *trim middle*^{→ P. 27}, *trim end*^{→ P. 27}, *trim*^{→ P. 27}, *no trimming*^{→ P. 27}.

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

middle config^{→ P. 27}.

```
286 middle config/.is choice,
287 middle config/.value required,
288 middle config/relbar/.code=\pgfkeysalso{%
289 middle={\relbar},
290 trim middle={2.5},
291 },
292 middle config/relbareda/.code={%
```

```

293 \ifundef{\relbareda}{%
294   \PackageWarning{overarrows}{Key 'middle config=relbareda' used,
295     \MessageBreak%
296     but \protect\relbareda\space is undefined; ignored.
297     \MessageBreak%
298     Load 'esvect' package, or use 'esvect' option \MessageBreak%
299     to remove this warning}
300 }{%
301   \pgfkeysalso{%
302     middle={\relbareda},
303     trim middle={1},
304   }%
305 }%
306 },

307 middle config/harrowextender/.code={%
308   \pgfkeysalso{%
309     middle={\harrowextender},
310     trim middle={0},
311   }%
312 },

```

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.

```

313 middle config/auto/.code={%
314   \pgfkeysalso{%
315     middle={\ovar@auto@middle},
316     trim middle={\ovar@auto@trim@middle},
317   }%
318 },

```

amsmath → P.28.

```

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

```

esvect → P.28.

```

334 esvect/.is choice,%
335 esvect/mimic/.code=\pgfkeysalso{%
336   start={\relbared}, middle={\relbareda}, end={\fldr},
337   trim start=1.5,
338   trim end=1.5,
339   trim middle=0,
340   right arrow=2,
341   space before arrow=-.7pt,
342   space after arrow=-.3pt,
343 },
344 esvect/strict/.code=\pgfkeysalso{%
345   esvect=mimic,

```

```

346     no trimming,
347     fill macro={\traitfill@}, stack macro={\overvect@},
348   },
349   esvect/.default=mimic,

```

Initial configuration.

```

350   amsmath, middle config=auto, end=\ovar@rightarrow, right arrow,
351 }

```

6.4.4 Keys for the tikz method

```

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

```

Arrow macro.

```

353   no arrow macro hook/.code={%
354     \ifdef{\ovar@tikz@command}{%
355       \pgfkeysgetvalue{/overarrows/path options}{\ovar@tikz@pathoptions}%
356       \ovar@set{%
357         tikz command/.expanded={%
358           \noexpand\draw[\expandonce\ovar@tikz@pathoptions]\expandonce\ovar@tikz@path;
359         }%
360       }%
361     }%
362     \pgfkeysgetvalue{/overarrows/tikz options}{\ovar@tikz@options}%
363     \ovar@set{%
364       arrow macro/.expanded={%
365         $ \noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
366         \noexpand\tikz[\expandonce{\ovar@tikz@options}]{\expandonce{\ovar@tikz@command}}%
367         $ \noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
368       }%
369     }%
370   },

```

TikZ parts: `tikz command`^{→ P. 29}, `tikz options`^{→ P. 29}, `path options`^{→ P. 29}, `path`^{→ P. 29}.

```

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

```

TikZ handy keys: `add path options`^{→ P. 29}, `add tikz options`^{→ P. 29}, `arrows`^{→ P. 29}, `line thickness`^{→ P. 29}, `thinner`^{→ P. 29}.

```

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

```

Initial configuration.

```

392   shift right=-2,
393   min length=12,
394 }

```

6.4.5 Keys for the pstricks method

```

395 \SetOverArrowsMethod[lens]{pstricks}{%

```

Arrow macro.

```

396   no arrow macro hook/.code={%
397     \ovar@set{%
398       arrow macro/.expanded={%
399         $ \noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
400         \noexpand\begin{pspicture}\expandonce{\ovar@pstricks@geometry}%
401         \noexpand\psset{linewidth=\expandonce{\ovar@pstricks@linethickness}}%
402         \noexpand\psset{\expandonce{\ovar@pstricks@psset}}%
403         \expandonce{\ovar@pstricks@command}%
404         \noexpand\end{pspicture}%
405         $ \noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
406       }%
407     }%
408   },

```

Pstricks parts: pstricks command^{→P.30}, psset^{→P.30}, geometry^{→P.30}, line thickness^{→P.30}.

```

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

```

Pstricks handy key: arrow^{→P.30}, thinner^{→P.30}.

```

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

```

Initial configuration.

```

421   shift right=-2,
422   min length=12,
423   geometry={(0,-0.5ex)(\overarrowlength,0.5ex)},%
424   line thickness={\overarrowthickness},%
425   arrow={->},%
426   psset={},%
427 }

```

6.4.6 Keys for the picture method

```

428 \SetOverArrowsMethod[lens]{picture}{%

```

Arrow macro.

```

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

```

```

437     $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
438     }%
439   }%
440 },

```

Picture parts: `picture` command^{→P.31}, `geometry`^{→P.31}, `line thickness`^{→P.31}.

```

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

```

Picture handy key: `thinner`^{→P.31}.

```

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

```

Initial configuration.

```

448 shift right=-2,
449 min length=18,
450 geometry={({\overarrowlength,1ex}(0,-0.5ex)},%
451 line thickness={\overarrowthickness},%
452 picture command={\put(0,0){\vector(1,0){\overarrowlength}}},%
453 }

```

6.5 Commands

6.5.1 Macros for symbols assemblage

```

\joinrel
454 \ifdef{\xjoinrel}{%
455   \PackageWarning{overarrows}{Command \protect\xjoinrel\space already defined.
456   \MessageBreak%
457   Previous definition will be overridden}%
458 }{}

```

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

```

459 \DeclareRobustCommand{\xjoinrel}[1][3.5]{\mathrel{\mkern-#1mu}}

```

```

\smallermathstyle
460 \newcommand*{\smallermathstyle}{%
461   \mathchoice{\scriptstyle}{\scriptstyle}{\scriptscriptstyle}{}}%
462 }

```

`\arrowmathstyle` Math styles for arrows commands. #1: math style

```

\ovar@save@mathstyle
463 \def\ovar@save@mathstyle#1{%
464   \let\arrowmathstyle#1%
465 }%

```

`\ovar@bar` Draw a zero height horizontal bar aligned on the baseline.
#1: math style

```

466 \def\ovar@bar#1{%
467   \raisebox
468   {\dimeval[-0.5\height+0.5\depth]} [Opt] [Opt]
469   {${\m@th #1}\vphantom{+}\std@minus$}%
470 }

```

`\baselinebar` Draw a zero height horizontal bar aligned on the baseline with the current math style (use `\arrowmathstyle`, if defined).

```

471 \NewDocumentCommand{\baselinebar} { } {} {}
472 \mathrel {%
473   \ifdef \arrowmathstyle
474     {\ovar@bar\arrowmathstyle}

```

```

475   {\mathpalette\ovar@bar{}}}%
476   }%
477 }

```

`\ovar@arrow@fill` Macro used for default fill macro^{→P.33}.
#1: left shift
#2: right shift
#3: arrow start
#4: arrow middle
#5: arrow end
#6: math style

```

478 \def\ovar@arrow@fill#1#2#3#4#5#6{%
479   $\m@th\thickmuskip0\mu\medmuskip\thickmuskip\thinmuskip\thickmuskip\relax%
480   \mkern #1 \mu\relax#6#3%
481   \cleaders\hbox{#6#4$}\hfill%
482   #5\mkern #2 \mu\relax$%
483 }

```

6.5.2 Macros for fixed length arrows

Lengths declaration.

```

484 \newlength{\overarrowlength}
485 \newlength{\overarrowthickness}
486 \newlength{\overarrowsmallerthickness}
487 \newlength{\ovar@tempdim}

```

`\ovar@set@arrowlength` Sets `\overarrowlength`^{→P.21}.
#1: min length, in math units
#2: math style
#3: content

```

488 \def\ovar@set@arrowlength#1#2#3{%
489   \settowidth{\ovar@tempdim}{$\m@th#2\mskip #1 \mu\relax$}%
490   \settowidth{\overarrowlength}{$\m@th#2#3$}%
491   \ifdim \overarrowlength < \ovar@tempdim \overarrowlength=\ovar@tempdim\fi%
492 }

```

`\ovar@set@arrowthickness` Sets `\overarrowthickness`^{→P.21} and `\overarrowsmallerthickness`^{→P.21}.
#1: math style

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

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.

```

493 \def\ovar@rulethickness@fontdimen{8}
494 \def\ovar@rulethickness@family{3}
495 \def\ovar@set@arrowthickness#1{%
496   \ifx#1\displaystyle%
497     \overarrowthickness =
498     \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
499     \overarrowsmallerthickness =
500     \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
501   \else\ifx#1\textstyle%
502     \overarrowthickness =
503     \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
504     \overarrowsmallerthickness =
505     \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%

```

```

506 \else\ifx#1\scriptstyle%
507 \overarrowthickness =
508 \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
509 \overarrowsmallerthickness =
510 \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
511 \else%
512 \overarrowthickness =
513 \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
514 \overarrowsmallerthickness = \overarrowthickness%
515 \fi\fi\fi%
516 }

```

unicode-math with LuaTeX version.

```

517 \def\ovar@set@arrowthickness@UM@lua#1{%
518 \overarrowthickness = \Umathoverbarrule #1
519 \ifx#1\displaystyle%
520 \overarrowsmallerthickness = \Umathoverbarrule \textstyle%
521 \else\ifx#1\textstyle%
522 \overarrowsmallerthickness = \Umathoverbarrule \scriptstyle%
523 \else%
524 \overarrowsmallerthickness = \Umathoverbarrule \scriptscriptstyle%
525 \fi\fi%
526 }

```

Test which version to use.

```

527 \AddToHook{begindocument}[overarrows]
528 {%
529 \@ifpackageloaded{unicode-math-luatex}
530 {%
531 \global\let\ovar@set@arrowthickness\ovar@set@arrowthickness@UM@lua%
532 }%
533 {%
534 \@ifpackageloaded{unicode-math-xetex}
535 {%
536 \gdef\ovar@rulethickness@fontdimen{54}%
537 \gdef\ovar@rulethickness@family{2}%
538 }
539 }%
540 }%
541 }

```

6.5.3 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

```

542 \def\ovar@stackover@@#1#2#3#4#5#6{%
543 \ovar@save@mathstyle{#5}%
544 \ovar@set@arrowthickness{#5}%
545 \vbox{%
546 \ialign{##\crrc%
547 $#5\mskip #1 mu\relax$\crrc%
548 \noalign{#2\nointerlineskip}#4\crrc%
549 \noalign{#3\nointerlineskip}%
550 $\m@th\hfil#5#6\hfil$\crrc%

```

```

551 }%
552 }%
553 }
554 \def\ovar@stackunder@@#1#2#3#4#5#6{%
555   \ovar@save@mathstyle{#5}%
556   \ovar@set@arrowthickness{#5}%
557   \vtop{%
558     \ialign{##\cr
559       $\m@th\hfil#5#6\hfil$\cr
560       \noalign{#2\nointerlineskip}#4\cr
561       \noalign{#3\nointerlineskip}%
562       $#5\mskip #1 mu\relax$\cr
563     }%
564   }%
565 }

```

`\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

```

566 \def\ovar@stackover@#1#2#3#4#5{\ovar@stackover@@{0}{#1}{#2}{#3}{#4}{#5}}
567 \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`

#1: min length, in math units

`\ovar@stack@fill`

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

```

568 \def\ovar@stackover@fill#1#2#3#4#5#6{\ovar@stackover@@{#1}{#2}{#3}{#4#5}{#5}{#6}}
569 \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^{→ P. 25}.

```

570 \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

```

571 \def\ovar@stackover@lens#1#2#3#4#5#6{%
572   \ovar@set@arrowlength{#1}{#5}{#6}%
573   \ovar@stackover@{#2}{#3}{#4}{#5}{#6}%
574 }
575 \def\ovar@stackunder@lens#1#2#3#4#5#6{%
576   \ovar@set@arrowlength{#1}{#5}{#6}%
577   \ovar@stackunder@{#2}{#3}{#4}{#5}{#6}%
578 }

```

`\ovar@stack@lens` matches the macro `\ovar@stackover@lens` by default, or `\ovar@stackunder@lens` with arrow under^{→ P. 25}.


```
579 \def\ovar@stack@lens{\ovar@stackover@lens}
```

6.5.4 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.

```
580 \ExplSyntaxOn
581 \cs_new_protected:Npn __overarrows_processor_strip_escape_char:n #1
582 {
583   \regex_match:nnTF { ~\cC. } { #1 }
584   { \tl_set:Nx \ProcessedArgument { \cs_to_str:N #1 } }
585   { \tl_set:Nx \ProcessedArgument { #1 } }
586 }
587 \cs_new_eq:NN \ovar@cmdname@processor __overarrows_processor_strip_escape_char:n
588 \ExplSyntaxOff

\DeclareOverArrowCommand
589 \NewDocumentCommand{\DeclareOverArrowCommand}{
590   O{symb} >{\ovar@cmdname@processor} m m
591 }{%
592   \begingroup
593   \ovar@set@common
594   \ifcsdef{\ovar@set@#1}{%
595     \csuse{\ovar@set@#1}%
596   }{%
597     \PackageError{overarrows}{Unknown method #1
598       {Try with 'symb', 'tikz', 'pstriks' or 'picture'}}%
599   }%
600   \ovar@set@#3}%
601   \ifdef{\ovar@macro@arrow}{%
602     \ovar@set{no arrow macro hook}%
603   }%
604   \ifdef{\ovar@macro@stack}{%
605     \ovar@set{no stack macro hook}%
606   }%
607   \csxdef{\ovar@#2@normal}{%
608     \noexpand\mathpalette{%
609       \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}%
610     }%
611   }%
612   \csxdef{\ovar@#2@starred}{%
613     \noexpand\mathpalette{%
614       \noexpand\ovar@starversion{%
615         \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}%
616       }%
617     }%
618   }%
619   \ifovar@option@debug@
620   \PackageInfo{overarrows}{%
621     Meaning of \protect\ovar@#2@normal\MessageBreak
622     used for \@backslashchar#2:\MessageBreak%
623     \expandafter\meaning\csname \ovar@#2@normal\endcsname}%
624   \fi

625   \expandafter%
626   \endgroup%
627   \ifovar@detectsubscripts%
```

Expand `\ifovar@detectsubscripts@` before closing the group, then define the command.

```

628 \csgdef{ovar@#2@auto}##1{%
629 \@ifnextchar \ovar@subcmd {%
630 \csuse{ovar@#2@starred}{##1}%
631 }{%
632 \csuse{ovar@#2@normal}{##1}%
633 }%
634 }%
635 \expandafter\DeclareDocumentCommand\csname #2\endcsname { s }{%
636 \IfBooleanTF{##1}{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@auto}}%
637 }%
638 \else
639 \expandafter\DeclareDocumentCommand\csname #2\endcsname { s }{%
640 \IfBooleanTF{##1}{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@normal}}%
641 }%
642 \fi
643 }
\ProvideOverArrowCommand
644 \NewDocumentCommand{\ProvideOverArrowCommand}{
645 O{symb} >\ovar@cmdname@processor} m m
646 }{%
647 \ifcsdef{#2}{-}{%
648 \DeclareOverArrowCommand[#1]{#2}{#3}%
649 }%
650 }
\NewOverArrowCommand
651 \NewDocumentCommand{\NewOverArrowCommand}{
652 O{symb} >\ovar@cmdname@processor} m m
653 }{%
654 \ifcsdef{#2}{-}{%
655 \PackageError{overarrows}{Command \csname #2\endcsname already defined}%
656 {You have used \protect\NewOverArrowCommand\space with a command that
657 already has a definition. \MessageBreak%
658 Choose another name, or use instead \protect\DeclareOverArrowCommand.}%
659 }{%
660 \DeclareOverArrowCommand[#1]{#2}{#3}%
661 }%
662 }
\RenewOverArrowCommand
663 \NewDocumentCommand{\RenewOverArrowCommand}{
664 O{symb} >\ovar@cmdname@processor} m m
665 }{%
666 \ifcsundef{#2}{-}{%
667 \PackageError{overarrows}{Command \csname #2\endcsname undefined}%
668 {You have used \protect\RenewOverArrowCommand\space with a command that was
669 never defined. \MessageBreak%
670 Check the requested name, or use instead \protect\NewOverArrowCommand.}%
671 }{%
672 \DeclareOverArrowCommand[#1]{#2}{#3}%
673 }%
674 }

```

6.5.5 Starred variant

\ovar@starversion

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

```

675 \newsavebox\ovar@tempbox
676 \def\ovar@starversion#1#2#3{%
677 \sbox{\ovar@tempbox}{\m@th #1#2{#3}}%
678 \usebox{\ovar@tempbox}%

```

Remove the extra space added by the arrow.

```

679 \settowidth{\ovar@tempdim}{$\m@th #2{#3}$}%
680 \kern\dimeval{0.5\ovar@tempdim - 0.5\wd\ovar@tempbox}%
681 }

```

6.5.6 `\vv` vector command

`\vv`
`\esvectvv` Backup and redefinition of `esvect` `\vv`^{P. 22} vector command.

```

682 \ifdefined\ovar@option@esvect
683 \let\esvectvv\vv
684 \undef\vv

685 \NewOverArrowCommand{\vv}{esvect, middle config = relbar}
686 \fi

```

6.5.7 Predefined commands

Declare predefined commands after unicode-math settings.

```

687 \AddToHook{begindocument}[overarrows]
688 {%
\overrightarrow 689 \ifovar@option@overrightarrow@
690 \DeclareOverArrowCommand{\overrightarrow}{%
691 amsmath, middle config=relbar,
692 end=\ovar@rightarrow,
693 right arrow,
694 }%
695 \fi
\underrightarrow 696 \ifovar@option@underrightarrow@
697 \DeclareOverArrowCommand{\underrightarrow}{%
698 amsmath, middle config=relbar,
699 end=\ovar@rightarrow,
700 right arrow,
701 arrow under,
702 }%
703 \fi
\overleftarrow 704 \ifovar@option@overleftarrow@
705 \DeclareOverArrowCommand{\overleftarrow}{%
706 amsmath, middle config=relbar,
707 start=\ovar@leftarrow,
708 end=\relbar,
709 left arrow,
710 }%
711 \fi
\underleftarrow 712 \ifovar@option@underleftarrow@
713 \DeclareOverArrowCommand{\underleftarrow}{%
714 amsmath, middle config=relbar,
715 start=\ovar@leftarrow,
716 end=\relbar,
717 left arrow,
718 arrow under,
719 }%
720 \fi
\overleftrightarrow 721 \ifovar@option@overleftrightarrow@
722 \DeclareOverArrowCommand{\overleftrightarrow}{%
723 amsmath, middle config=relbar,
724 start=\ovar@leftarrow,
725 end=\ovar@rightarrow,

```

```

726         center arrow,
727     }%
728 \fi

\underrightarrow
729 \ifovar@option@underrightarrow@
730 \DeclareOverArrowCommand{\underrightarrow}{%
731     amsmath, middle config=relbar,
732     start=\over@leftarrow,
733     end=\over@rightarrow,
734     center arrow,
735     arrow under,
736 }%
737 \fi

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

\underrightharpoonup
745 \ifovar@option@underrightharpoonup@
746 \DeclareOverArrowCommand{\underrightharpoonup}{%
747     amsmath, middle config=relbar,
748     end=\rightharpoonup,
749     right arrow,
750     arrow under,
751 }%
752 \fi

\overrightarrow
753 \ifovar@option@overrightarrow@
754 \DeclareOverArrowCommand{\overrightarrow}{%
755     amsmath, middle config=relbar,
756     end=\rightharpoondown,
757     right arrow,
758 }%
759 \fi

\underrightharpoondown
760 \ifovar@option@underrightharpoondown@
761 \DeclareOverArrowCommand{\underrightharpoondown}{%
762     amsmath, middle config=relbar,
763     end=\rightharpoondown,
764     right arrow,
765     arrow under,
766 }%
767 \fi

\overleftarrow
768 \ifovar@option@overleftarrow@
769 \DeclareOverArrowCommand{\overleftarrow}{%
770     amsmath, middle config=relbar,
771     start=\leftarrow,
772     end=\relbar,
773     left arrow,
774 }%
775 \fi

\underleftarrow
776 \ifovar@option@underleftarrow@
777 \DeclareOverArrowCommand{\underleftarrow}{%
778     amsmath, middle config=relbar,
779     start=\leftarrow,
780     end=\relbar,
781     left arrow,
782     arrow under,
783 }%
784 \fi

```

```

\overleftharpoondown 785 \ifovar@option@overleftharpoondown@
786 \DeclareOverArrowCommand{\overleftharpoondown}{%
787 amsmath, middle config=relbar,
788 start=\leftharpoondown,
789 end=\relbar,
790 left arrow,
791 }%
792 \fi

\underleftharpoondown 793 \ifovar@option@underleftharpoondown@
794 \DeclareOverArrowCommand{\underleftharpoondown}{%
795 amsmath, middle config=relbar,
796 start=\leftharpoondown,
797 end=\relbar,
798 left arrow,
799 arrow under,
800 }%
801 \fi

\overbar 802 \ifovar@option@overbar@
803 \DeclareOverArrowCommand{\overbar}{%
804 amsmath, middle={\baselinebar},
805 start={\baselinebar}, end={\baselinebar},
806 space before arrow={0.5\overarrowthickness},
807 space after arrow={3.5\overarrowthickness}
808 }%
809 \fi

\underbar 810 \ifovar@option@underbar@
811 \DeclareOverArrowCommand{\underbar}{%
812 amsmath, middle={\baselinebar},
813 start={\baselinebar}, end={\baselinebar},
814 arrow under=noconfig,
815 space before arrow={3.5\overarrowthickness},
816 space after arrow={0.5\overarrowthickness}
817 }%
818 \fi

End of \AddToHook{begindocument} hook.
819 }

```

6.5.8 Test macros

\ovar@testmathstyles

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

```

820 \newcommand{\ovar@testmathstyles}[2] [] {%
821 \begin{group}
822 \newcommand*{\ovar@row@teststyle}[1] {%
823 $\displaystyle ##1$
824 & $\textstyle ##1$
825 & $\scriptstyle ##1$
826 & $\scriptscriptstyle ##1$
827 \\
828 }%
829 \renewcommand*{\arraystretch}{1.5}%
830 \begin{tabular*}{0.95\linewidth}{@{\extracolsep{\fill}} cccc}
831 \hline
832 \footnotesize\texttt{\texttt{\textbackslash displaystyle}}
833 & \footnotesize\texttt{\texttt{\textbackslash textstyle}}
834 & \footnotesize\texttt{\texttt{\textbackslash scriptstyle}}
835 & \footnotesize\texttt{\texttt{\textbackslash scriptscriptstyle}}

```

```

836     \\\
837     \hline
838     \ovar@row@teststyle{\cuse{#2}{v}}
839     \ovar@row@teststyle{\cuse{#2}{AB}}
840     \ovar@row@teststyle{\cuse{#2}{\mathrm{grad}}}}
841     \ovar@row@teststyle{\cuse{#2}{my~long~vector}}
842     \IfValueT{#1}{\ovar@row@teststyle{\cuse{#2}{#1}}}
843     \hline
844     \end{tabular*}%
845     \endgroup%
846 }
\ovar@testkerning
847 \begingroup
848 \ifovar@option@subother@ \catcode \_ =12 \fi
849 \ifovar@option@subactive@ \catcode \_ =13 \fi
850 \gdef\ovar@testkerning#1{%
851     \begin{displaymath}
852         #1\{t\}_{#1\{u\}_{#1\{v\}}}}
853         \quad
854         #1\{\imath\}_0
855         \quad
856         #1\{v\}
857         = #1\{v\}_x + #1\{v\}_y + #1\{v\}_z
858         = v_x #1\{\imath\} + v_y #1\{\jmath\} + v_z #1\{k\}
859     \end{displaymath}%
860 }
861 \endgroup
\TestOverArrow
862 \NewDocumentCommand{\TestOverArrow}{
863     s o >{\ovar@cmdname@processor} m
864 }{%
865     \ifcsdef{#3}{}{%
866         \PackageWarning{overarrows}{Unknown name '#3' passed to
867             \protect\TestOverArrow}%
868     }%
869     \IfBooleanTF{#1}{%
870         \noindent\framebox{%
871             \begin{minipage}{0.95\linewidth}
872                 \centering
873                 \noindent\textbf{\large%
874                     Test of \texttt{\textbackslash#3} and \texttt{\textbackslash#3*} macros}
875                 \bigskip\par
876                 \textbf{\texttt{\textbackslash#3} for different math styles}
877                 \smallskip\par
878                 \ovar@testmathstyles[#2]{#3}%
879                 \bigskip\par
880                 \textbf{\texttt{\textbackslash#3} kerning}
881                 \ovar@testkerning{\cuse{#3}}
882                 \textbf{\texttt{\textbackslash#3*} kerning}
883                 \ovar@testkerning{\cuse{#3}*}
884             \end{minipage}%
885         }\bigskip\par
886     }{%
887         \ovar@testmathstyles[#2]{#3}%
888     }%
889 }

```

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
- `overlefttharpoondown`, 15
- `overlefttharpoonup`, 15
- `overleftrightharpoonup`, 15
- `overrightarrow`, 15
- `overrightarrowtharpoondown`, 15
- `overrightarrowtharpoonup`, 15
- `pstarrows`, 17
- `pstricks`, 17
- `subactive`, 17
- `subother`, 17
- `subscripts`, 17
- `tikz`, 16
- `underbar`, 16
- `undercommands`, 15
- `underleftarrow`, 16
- `underlefttharpoondown`, 16
- `underlefttharpoonup`, 16
- `underleftrightharpoonup`, 16
- `underrightarrow`, 16
- `underrightharpoondown`, 16
- `underrightharpoonup`, 16

- `add path options` key, 29
- `add tikz options` key, 29
- `after arrow` key, 26
- `allcommands` package option, 15
- `amsmath` key, 28
- `arrow` key, 30
- `arrow macro` key, 32
- `arrow under` key, 25

- `\arrowmathstyle`, 20

- `arrows` key, 29

- `\baselinebar`, 20

- `before arrow` key, 26

- `center arrow` key, 25

Commands

- `\arrowmathstyle`, 20, 45
- `\baselinebar`, 20, 45, 53
- `\DeclareOverArrowCommand`, 17, 49–53
- `\esvectvv`, 22, 51
- `\NewOverArrowCommand`, 17, 50, 51
- `\overbar`, 23, 36, 53
- `\overleftarrow`, 22, 36, 51
- `\overlefttharpoondown`, 23, 36, 53
- `\overlefttharpoonup`, 23, 36, 52
- `\overleftrightharpoonup`, 23, 36, 51
- `\overrightarrow`, 22, 36, 51
- `\overrightarrowtharpoondown`, 23, 36, 52
- `\overrightarrowtharpoonup`, 23, 36, 52
- `\ProvideOverArrowCommand`, 17, 50
- `\RenewOverArrowCommand`, 17, 50
- `\SetOverArrowsMethod`, 32, 39, 41, 43, 44
- `\SetOverArrowsMethod*`, 32
- `\SetOverArrowsSubscriptCommand`, 32, 39
- `\smallermathstyle`, 19, 45
- `\TestOverArrow`, 18, 54
- `\TestOverArrow*`, 18
- `\underbar`, 24, 36, 53
- `\underleftarrow`, 23, 36, 51
- `\underlefttharpoondown`, 24, 36, 53
- `\underlefttharpoonup`, 24, 36, 52
- `\underleftrightharpoonup`, 23, 36, 52
- `\underrightarrow`, 23, 36, 51
- `\underrightharpoondown`, 24, 36, 52
- `\underrightharpoonup`, 23, 36, 52
- `\vv`, 22, 51
- `\vv*`, 22
- `\xjoinrel`, 19, 41, 45

- `debug` package option, 17

`\DeclareOverArrowCommand`, 17
`detect subscripts` key, 26

`end` key, 27
`esvect` key, 28
`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
`\esvectv`, 22

`fill macro` key, 33

`geometry` key, 30, 31

Internal macros

- `\ifovar@detectsubscripts@`, 35, 49
- `\ifovar@option@debug@`, 35, 49
- `\ifovar@option@oldarrows@`, 35, 37
- `\ifovar@option@overbar@`, 36, 53
- `\ifovar@option@overleftarrow@`, 35, 51
- `\ifovar@option@overleftharpoondown@`, 36, 53
- `\ifovar@option@overleftharpoonup@`, 36, 52
- `\ifovar@option@overleftrightharpoon@`, 36, 51
- `\ifovar@option@overrightarrow@`, 35, 51
- `\ifovar@option@overrightharpoondown@`, 36, 52
- `\ifovar@option@overrightharpoonup@`, 36, 52
- `\ifovar@option@pstarrows@`, 35, 38
- `\ifovar@option@pstricks@`, 35, 38
- `\ifovar@option@subactive@`, 35, 39, 54
- `\ifovar@option@subother@`, 35, 39, 54
- `\ifovar@option@tikz@`, 35, 38
- `\ifovar@option@underbar@`, 36, 53
- `\ifovar@option@underleftarrow@`, 36, 51
- `\ifovar@option@underleftharpoondown@`, 36, 53
- `\ifovar@option@underleftharpoonup@`, 36, 52
- `\ifovar@option@underleftrightharpoon@`, 36, 52
- `\ifovar@option@underrightharpoon@`, 35, 51
- `\ifovar@option@underrightharpoondown@`, 36, 52
- `\ifovar@option@underrightharpoonup@`, 36, 52
- `\ovar@after@arrow`, 39, 40
- `\ovar@arrow@end`, 41
- `\ovar@arrow@fill`, 41, 46
- `\ovar@arrow@middle`, 41
- `\ovar@arrow@start`, 41
- `\ovar@auto@middle`, 38, 42
- `\ovar@auto@trim@middle`, 38, 42
- `\ovar@bar`, 45, 46
- `\ovar@before@arrow`, 39, 40
- `\ovar@cmdname@processor`, 49, 50, 54
- `\ovar@leftarrow`, 37, 51, 52
- `\ovar@length@min`, 39, 40
- `\ovar@macro@arrow`, 39, 49
- `\ovar@macro@arrowfill`, 41
- `\ovar@macro@stack`, 39, 49
- `\ovar@option@esvect`, 36, 37, 51
- `\ovar@picture@command`, 44, 45
- `\ovar@picture@geometry`, 44, 45
- `\ovar@picture@linethickness`, 44, 45
- `\ovar@pstricks@command`, 44
- `\ovar@pstricks@geometry`, 44
- `\ovar@pstricks@linethickness`, 44
- `\ovar@pstricks@psset`, 44
- `\ovar@rightarrow`, 37, 43, 51, 52
- `\ovar@row@teststyle`, 53, 54
- `\ovar@rulethickness@family`, 46, 47
- `\ovar@rulethickness@fontdimen`, 46, 47
- `\ovar@save@mathstyle`, 45, 47, 48
- `\ovar@set`, 39, 41, 43, 44, 49
- `\ovar@set@arrowlength`, 46, 48
- `\ovar@set@arrowthickness`, 46–48
- `\ovar@set@arrowthickness@UM@lua`, 47
- `\ovar@set@common`, 49
- `\ovar@shift@left`, 40, 41, 43, 44
- `\ovar@shift@right`, 40, 41, 43–45
- `\ovar@stack@fill`, 40, 48
- `\ovar@stack@lens`, 40, 49
- `\ovar@stackover@`, 48

- `\ovar@stackover@`, 47, 48
- `\ovar@stackover@fill`, 48
- `\ovar@stackover@lens`, 48, 49
- `\ovar@stackunder@`, 48
- `\ovar@stackunder@`, 48
- `\ovar@stackunder@fill`, 40, 48
- `\ovar@stackunder@lens`, 40, 48
- `\ovar@starversion`, 49, 50
- `\ovar@subcmd`, 39, 50
- `\ovar@tempbox`, 50, 51
- `\ovar@tempdim`, 46, 51
- `\ovar@testkerning`, 54
- `\ovar@testmathstyles`, 53, 54
- `\ovar@tikz@command`, 43
- `\ovar@tikz@options`, 43
- `\ovar@tikz@path`, 43
- `\ovar@tikz@pathoptions`, 43
- `\ovar@trim@end`, 41
- `\ovar@trim@middle`, 41
- `\ovar@trim@start`, 41

Keys

- add path options, 29
- add tikz options, 29
- after arrow, 26
- amsmath, 28
- arrow, 30
- arrow macro, 32
- arrow under, 25
- arrows, 29
- before arrow, 26
- center arrow, 25
- detect subscripts, 26
- end, 27
- esvect, 28
- fill macro, 33
- geometry, 30, 31
- left arrow, 25
- line thickness, 29–31
- middle, 27
- middle config, 27
- min length, 24
- no arrow macro hook, 33
- no stack macro hook, 32
- no trimming, 27
- path, 29
- path options, 29
- picture command, 31
- psset, 30
- pstricks command, 30
- right arrow, 26
- shift left, 25
- shift leftright, 25
- shift right, 25

- space after arrow, 26
- space before arrow, 26
- stack macro, 32
- start, 27
- thinner, 29–31
- tikz command, 29
- tikz options, 29
- trim, 27
- trim end, 27
- trim middle, 27
- trim start, 27

left arrow key, 25

Lengths

- `\overarrowlength`, 21, 43–46
- `\overarrowsmallerthickness`, 21, 43–47
- `\overarrowthickness`, 21, 43–47, 53

line thickness key, 29–31

middle key, 27

middle config key, 27

min length key, 24

`\NewOverArrowCommand`, 17

no arrow macro hook key, 33

no stack macro hook key, 32

no trimming key, 27

noesvect package option, 14

old-arrows package option, 16

`\overarrowlength` length, 21

`\overarrowsmallerthickness` length, 21

`\overarrowthickness` length, 21

`\overbar`, 23

overbar package option, 15

overcommands package option, 15

`\overleftarrow`, 22

overleftarrow package option, 15

`\overleftharpoondown`, 23

overleftharpoondown package option, 15

`\overleftharpoonup`, 23

overleftharpoonup package option, 15

`\overleftrightarrow`, 23

overleftrightarrow package option, 15

`\overrightarrow`, 22

overrightarrow package option, 15

`\overrightarrowharpoondown`, 23

overrightarrowharpoondown package option, 15

`\overrightarrowharpoonup`, 23

overrightarrowharpoonup package option, 15

`path` key, 29
`path options` key, 29
`picture command` key, 31
`\ProvideOverArrowCommand`, 17
`psset` key, 30
`pstarrows` package option, 17
`pstricks` package option, 17
`pstricks command` key, 30

`\RenewOverArrowCommand`, 17
`right arrow` key, 26

`\SetOverArrowsMethod`, 32
`\SetOverArrowsMethod*`, 32
`\SetOverArrowsSubscriptCommand`, 32
`shift left` key, 25
`shift leftright` key, 25
`shift right` key, 25
`\smallermathstyle`, 19
`space after arrow` key, 26
`space before arrow` key, 26
`stack macro` key, 32
`start` key, 27
`subactive` package option, 17
`subother` package option, 17
`subscripts` package option, 17

`\TestOverArrow`, 18
`\TestOverArrow*`, 18
`thinner` key, 29–31
`tikz` package option, 16
`tikz command` key, 29
`tikz options` key, 29
`trim` key, 27
`trim end` key, 27
`trim middle` key, 27
`trim start` key, 27

`\underbar`, 24
`underbar` package option, 16
`undercommands` package option, 15
`\underleftarrow`, 23
`underleftarrow` package option, 16
`\underleftharpoondown`, 24
`underleftharpoondown` package option,
16
`\underleftharpoonup`, 24
`underleftharpoonup` package option, 16
`\underletrightarrow`, 23
`underletrightarrow` package option,
16
`\underrightarrow`, 23
`underrightarrow` package option, 16
`\underrightharpoondown`, 24

`underrightharpoondown` package
option, 16
`\underrightharpoonup`, 23
`underrightharpoonup` package option,
16

`\vv`, 22
`\vv*`, 22

`\xjoinrel`, 19

Change History

v1.0		v1.3	
General: Initial version	1	General: Process options in the	
v1.0.1		order specified by the user . . .	37
General: Bug fix for under* options	36	Replace esvect conditionnal by	
v1.1		the definition of a control	
General: Support for non-standard		sequence	51
subscripts	38, 50, 54	Use a control sequence to store	
v1.2		the esvect option	36
General: Add option pstricks	36	Use only one	
Add the method pstricks	44	<code>\PassOptionsToPackage</code> with	
Allow backslash in command		<code>esvect</code>	37
name	49	v1.4	
Declare predefined commands		General: Make commmands robust	49
after <code>unicode-math</code> settings . . .	51	v1.5	
Fix esvect font sizes	37	General: Make	
Get the correct rule thickness		<code>\overarrowthickness</code> ^{→P.21}	
with <code>unicode-math</code>	46	and	
Remove useless		<code>\overarrowsmallerthickness</code> ^{→P.21}	
<code>\AtBeginDocument</code>	39	always available	47
Use <code>\def</code> instead of <code>\let</code> for		Redefine <code>\overbar</code> ^{→P.23} so that	
<code>\ovar@rightarrow</code> and		it can be properly nested.	53
<code>\ovar@leftarrow</code>	37	Redefine <code>\underbar</code> ^{→P.24} so that	
Use <code>\harrowextender</code> , if		it can be properly nested.	53
availlable.	38	Save math style	45, 47
Use boxes in starred variant for			
better performances	50		