

The `overarrows` package*

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

July 8, 2024

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:

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

- to draw arrows of various shapes above math expressions:

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

- to draw arrows of various shapes under math expressions:

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

*This document corresponds to `overarrows` v1.1, dated 2023/02/15.

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	14
4.1.3	Other options	16
4.2	Commands	17
4.2.1	Macro for commands creation	17
4.2.2	Useful macros for symbols assemblage	18
4.2.3	Useful lengths for TikZ, PSTricks or <code>picture</code> environment	19
4.2.4	Vectors macros	20
4.2.5	Predefined commands	21
4.3	Keys	22
4.3.1	Arrow position and length settings	22
4.3.2	Subscripts detection setting	24
4.3.3	Symbols assemblage settings	25
4.3.4	TikZ settings	26
4.3.5	PSTricks settings	28
4.3.6	Picture environment settings	29
4.4	Advanced commands and keys	29
4.4.1	Advanced commands	30
4.4.2	Advanced keys	30
5	Complements	31
5.1	Know issues	31
5.1.1	Math font change	31
5.1.2	Detection of non standard subscripts	31
5.2	Package dependencies	32
5.3	Alternatives	32
5.4	Changelog	32
6	Implementation	33

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, \TeX/L\AA\TeX 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 extensible, 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 \TeX/L\AA\TeX , and which is redefined by the commonly used `amsmath` package. But its arrow, built with the `\rightarrow` symbol \rightarrow , is too large with 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>.

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 symbol⁴. Thus, the line of the macro `\overrightarrow` is made by repetition of command `\relbar` — (which simply corresponds to the minus sign), while `\vv` use its own command `\relbareda` -.

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

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

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

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

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

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

⁴Using the `TEX \cleaders` command.

3 Quick start

3.1 Loading the package `overarrows`

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

```
\usepackage{overarrows}
```

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

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

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

- | | |
|---|---|
| • <code>\overrightarrow</code> ^{→P. 21} | • <code>\underrightarrow</code> ^{→P. 21} |
| • <code>\overleftarrow</code> ^{→P. 21} | • <code>\underleftarrow</code> ^{→P. 22} |
| • <code>\overleftrightarrow</code> ^{→P. 21} | • <code>\underleftrightarrow</code> ^{→P. 22} |
| • <code>\overrightarrowharpoonup</code> ^{→P. 21} | • <code>\underrightharpoonup</code> ^{→P. 22} |
| • <code>\overrightarrowharpoondown</code> ^{→P. 21} | • <code>\underrightharpoondown</code> ^{→P. 22} |
| • <code>\overleftarrowharpoonup</code> ^{→P. 21} | • <code>\underleftarrowharpoonup</code> ^{→P. 22} |
| • <code>\overleftarrowharpoondown</code> ^{→P. 21} | • <code>\underleftarrowharpoondown</code> ^{→P. 22} |
| • <code>\overbar</code> ^{→P. 21} | • <code>\underbar</code> ^{→P. 22} |

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 (without backslash), 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:

```
$ \mathbf{v}_{sub} \quad \backslashqqquad \backslashmyovercmd{\mathbf{v}}_{sub} \quad \backslashqqquad \backslashmyovercmd*{\mathbf{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. 25} and `end`^{→P. 25}. 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 L^AT_EX. This can be done with the package `amssymb`, by adding in preamble:

```
\usepackage{amssymb}
```

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

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

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

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

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

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

If the math font differs from the default *Computer Modern*, the central part of the arrow may have inappropriate position or line width. This is because the default symbol used for the arrow line is `\relbareda` from the `esvect` package. If needed, try to set the `middle`^{P.25} 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.25}, `trim end`^{P.25} and `trim middle`^{P.25} may need many trials. For this purpose, the macro `\TestOverArrow`^{P.18} displays the result of a command for different lengths and math styles:

```
\TestOverArrow{\overhooktwoheadrightarrow}
```

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

3.4 Size and position of the arrow

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

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

\overrightarrow{v} \overrightarrow{AB}

The key `trim`^{→P.25} sets `trim start`^{→P.25}, `trim middle`^{→P.25} and `trim end`^{→P.25} 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} \overrightarrow{AB}

Note that `\smallermathstyle`^{→P.19} should not be used for `end`^{→P.25}, because this last is formatted with the same math style as `start`^{→P.25}.

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.24}:

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

\overrightarrow{v} \overrightarrow{AB}

Default arrows are slightly shifted to the right. For a left arrow, this should be reversed, using the keys `shift left`^{→P.23} and `shift right`^{→P.23}. 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} $

```

\overleftarrow{v} \overleftarrow{AB}

Finally, the key `arrow under`^{P.23} places the arrow below the content, instead of above (and `space before arrow`^{P.24} 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} $

```

\underleftrightarrow{v} \underleftrightarrow{AB}

3.5 Symbols assemblage

Many L^AT_EX math symbols are built by assemblage, using the macro `\joinrel`⁵ 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 `\tttail` from the symbol `\succ` :




```

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

```



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

⁵For example, the symbol `\models`  is defined as `\mathrel{||}\joinrel\Relbar` and corresponds to the assemblage of a vertical line  and the symbol `\Relbar` . The command `\mathrel` modifies the spacing according to the math relation class ; `\Relbar` corresponds to the equal sign (it's definition is `\mathrel{=}`).

```

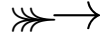
\NewOverArrowCommand{\overttailrightarrow}{%
  start={\tttail},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overttailrightarrow{v} \quad \overttailrightarrow{AB} $

```



Here the `min length`^{P. 22} key was added to ensure a minimum length (in math units) when the content of the command is small (as for a single character).

The previous arrow would be better with a smaller tail, and this can be done with the macro `\smallermathstyle`^{P. 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{\overttailrightarrow}{%
  start={\vcenter{\hbox{$\smallermathstyle\tttail$}}},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overttailrightarrow{v} \quad \overttailrightarrow{AB} $

```

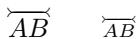


Text symbols, namely symbols that are not defined in math mode, can also be used. They should yet be enclosed in the `\text` macro, from the `amsmath` package, to be correctly displayed and correctly scaled according to math style. With, for example, the arrow heads given by the symbols 40 and 41 of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

```

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

```



3.6 Drawing the arrow with TikZ

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

Drawing arrows with TikZ requires to load the `tikz` package and its library `arrows.meta`. This can be simply done by passing the `tikz`^{→P.16} option to the `overarrows` package⁶:

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

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

- `\overarrowlength`^{→P.19} for the arrow length,
- `\overarrowthickness`^{→P.19} and `\overarrowsmallerthickness`^{→P.20} for the arrow thickness.

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

Without any other configuration, a right arrow is drawn:

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

\vec{v} \overrightarrow{AB}

Keys to use Tikz are described in section 4.3.4, page 26. Main keys are: `tikz options`^{→P.27}, `path options`^{→P.27} and `path`^{→P.27}. It's also possible to append settings with `add tikz options`^{→P.27} and `add path options`^{→P.27}. The full TikZ command used to draw the arrow can as well be entirely redefined with the key `tikz command`^{→P.27}

Here is a example of an arrow drawn with TikZ⁷:

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

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

⁶Note 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.

⁷TikZ 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

PSTricks commands can also be used to draw the arrow. For this, the optional argument `pstricks` must be passed to `\NewOverArrowCommand`^{→P.17}. Like for TikZ, the three lengths `\overarrowlength`^{→P.19}, `\overarrowthickness`^{→P.19} and `\overarrowsmallerthickness`^{→P.20} can be used in PSTricks commands. By default, a right arrow is drawn:

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



The `pstricks` package has to be loaded (for example, using the `pstricks`^{→P.16} option to the `overarrows` package)

Keys to use PSTricks commands are described in section 4.3.5, page 28. The main keys are `pstricks command`^{→P.28}, `psset`^{→P.28}, `arrow`^{→P.28}, `geometry`^{→P.28} and `line thickness`^{→P.28}. Examples:

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



```
\NewOverArrowCommand[pstricks]{\ellipticarrow}{%
  pstricks command={%
    \psellipticarcn{->}%^A strip spaces
    (0.5\overarrowlength,0.2\overarrowlength)^%^A strip spaces
    (0.5\overarrowlength,0.2\overarrowlength)^%^A strip spaces
    {170}{10}
  },
  geometry={(0,0.2\overarrowlength)(\overarrowlength,0.4\overarrowlength)},
  line thickness={\overarrowsmallerthickness},
  center arrow,
}
$ \ellipticarrow{v} \quad \ellipticarrow{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 for TikZ or PSTricks, the three lengths `\overarrowlength`^{→P.19}, `\overarrowthickness`^{→P.19} and `\overarrowsmallerthickness`^{→P.20} are available and can be used with `picture` drawing commands. By default, a right vector is drawn:

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

Keys to use L^AT_EX picture environment are described in section 4.3.6, page 29. The main keys are `picture` command^{→P.29}, `geometry`^{→P.29} and `line thickness`^{→P.29}. Here is an example:

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

\vec{v} \overrightarrow{AB}

4 User interface

4.1 Package options



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









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

4.1.1 esvect configuration

esvect

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

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

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


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

noesvect

Prevents the loading of the `esvect` package and the definition of the command `\vv→P.20`.


esvecta

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

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

4.1.2 Predefined commands

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

Three options are also available to define set of commands.

Set of commands

allcommands

Defines all sixteen predefined commands.

overcommands

Defines all eight predefined commands with arrow over.

undercommands

Defines all eight predefined commands with arrow under.

Over arrows

overrightarrow

Defines the `\overrightarrow`^{P.21} command: \vec{v} , \overrightarrow{AB} , $\overrightarrow{\text{grad}}$.

overleftarrow

Defines the `\overleftarrow`^{P.21} command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

overleftrightarrow

Defines the `\overleftrightarrow`^{P.21} command: \overleftrightarrow{v} , \overleftrightarrow{AB} , $\overleftrightarrow{\text{grad}}$.

overrightarrowtharpoonup

Defines the `\overrightarrowtharpoonup`^{P.21} command: $\overrightarrow{\text{v}}$, $\overrightarrow{\text{AB}}$, $\overrightarrow{\text{grad}}$.

overrightarrowtharpoondown

Defines the `\overrightarrowtharpoondown`^{P.21} command: $\overrightarrow{\text{v}}$, $\overrightarrow{\text{AB}}$, $\overrightarrow{\text{grad}}$.

overlefttharpoonup

Defines the `\overlefttharpoonup`^{P.21} command: \overlefttharpoonup{v} , \overlefttharpoonup{AB} , $\overlefttharpoonup{\text{grad}}$.

overlefttharpoondown

Defines the `\overlefttharpoondown`^{P.21} command: \overlefttharpoondown{v} , $\overlefttharpoondown{AB}$, $\overlefttharpoondown{\text{grad}}$.

overbar

Defines the `\overbar`^{P.21} command: \overline{v} , \overline{AB} , $\overline{\text{grad}}$.

Under arrows

underrightarrow

Defines the `\underrightarrow`^{P.21} command: $\underline{\text{v}}$, $\underline{\text{AB}}$, $\underline{\text{grad}}$.

underleftarrow

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

underleftrightharrow

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

underrightharpoonup

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

underrightharpoondown

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

underleftharpoonup

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

underleftharpoondown



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

underbar

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

4.1.3 Other options

old-arrows

Loads the `old-arrows` package with its option `old`. This provides the symbols `\varleftarrow`  and `\varrightarrow` , used then by default for predefined command.

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

tikz

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

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

pstricks

Loads the package `pstricks-add`.

Note that, as it, this will compile with \LaTeX , \Lua\LaTeX and \Xe\LaTeX , but not compile \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. 24} to `true`.

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

subother

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. 24} (see the section 5.1.2, page 31).

subactive

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. 24} (see the section 5.1.2, page 31).

debug

Writes the meaning of defined commands in \LaTeX log.

4.2 Commands

4.2.1 Macro for commands creation

```
\NewOverArrowCommand[method]{\langle name \rangle}{\langle keys \rangle}  
\RenewOverArrowCommand[method]{\langle name \rangle}{\langle keys \rangle}  
\ProvideOverArrowCommand[method]{\langle name \rangle}{\langle keys \rangle}  
\DeclareOverArrowCommand[method]{\langle name \rangle}{\langle keys \rangle}
```

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

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

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

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

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

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

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

`tikz` to draw the arrow with PGF/TikZ;
`pstricks` to draw the arrow with PSTricks;
`picture` to draw the arrow with the L^AT_EX `picture` environment.

With no $\langle method \rangle$ argument, the `symp` method is chosen.

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

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

\vec{v} \overrightarrow{ABCD}

`\TestOverArrow[$\langle pattern \rangle$]{ $\langle name \rangle$ }`

`\TestOverArrow*[$\langle pattern \rangle$]{ $\langle name \rangle$ }`

Displays the result of the command $\langle name \rangle$ 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 $\langle name \rangle$ and $\langle name \rangle^*$.

```
\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

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

`\vv*` kerning

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

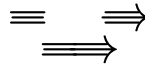
4.2.2 Useful macros for symbols assemblage

Math symbols assemblage is the default method used to draw arrows. The macros `\xjoinrel` and `\smallermathstyle` 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} $
```



`\smallermathstyle`

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

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

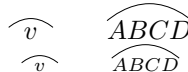
```
$ \displaystyle AB \quad \textstyle AB
\quad \scriptstyle AB \quad \scriptscriptstyle AB $ \par
$ \displaystyle AB \quad \smallermathstyle AB
\quad \smallermathstyle AB \quad \smallermathstyle AB $
```

AB AB AB AB
 AB AB AB AB

4.2.3 Useful lengths for TikZ, PSTricks or picture environment

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

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



`\overarrowlength`

Is set to the width of the arrow command content, or, if larger, to the minimal arrow length set through the key `min length`^{P.22}.

`\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`.

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

`\overarrowsmallerthickness`

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

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

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

4.2.4 Vectors macros

The macro `\vv`, dedicated to vectors, is automatically defined when the option `esvect→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} $
```

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

`\esvectvv`

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

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

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

4.2.5 Predefined commands

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

Over arrows

`\overrightarrow`

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

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.

`\overleftrightarrow`

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

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

`\overrightarrow{\hspace{0.5em}}`

$$\overrightarrow{\hspace{0.5em}}v \quad \overrightarrow{\hspace{0.5em}}AB \quad \overrightarrow{\hspace{0.5em}}\text{grad}$$

`\overrightarrow{\hspace{0.5em}}`

$$\overrightarrow{\hspace{0.5em}}v \quad \overrightarrow{\hspace{0.5em}}AB \quad \overrightarrow{\hspace{0.5em}}\text{grad}$$

`\overleftarrow{\hspace{0.5em}}`

$$\overleftarrow{\hspace{0.5em}}v \quad \overleftarrow{\hspace{0.5em}}AB \quad \overleftarrow{\hspace{0.5em}}\text{grad}$$

`\overleftarrow{\hspace{0.5em}}`

$$\overleftarrow{\hspace{0.5em}}v \quad \overleftarrow{\hspace{0.5em}}AB \quad \overleftarrow{\hspace{0.5em}}\text{grad}$$

`\overbar`

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

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.

`\underleftrightharpoonup`

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

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

`\underrightharpoonup`

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

`\underrightharpoonowdown`

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

`\underleftharpoonup`

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

`\underleftharpoonowdown`

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

`\underbar`

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

4.3 Keys

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

4.3.1 Arrow position and length settings

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

Length

`min length={⟨number⟩}` (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 `symb` method (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} $
```



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. 24} to `false` and the key `before arrow`^{→P. 24} 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} $
```



Horizontal shifts

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

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

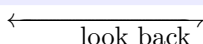
`shift right={\langle number \rangle}` (no default, see below for the initial value)

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

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

- $\langle number \rangle = 0$ for the `symb` method (default);
- $\langle number \rangle = -2$ for the `tikz`, `pstricks` and `picture` methods.

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



shift leftright=[*<number>*] (no default)

Sets **shift left** and **shift right** to the same *<number>* value.

center arrow

Sets **shift left**^{→P.23} and **shift right**^{→P.23} to zero.

left arrow (default 2)

Sets **shift left**^{→P.23} to zero and **shift right**^{→P.23} to *<number>*.

right arrow (default 2)

Sets **shift right**^{→P.23} to zero and **shift left**^{→P.23} 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**.

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

$\dot{\overrightarrow{v}}$ $\ddot{\overrightarrow{AB}}$

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 31).

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

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

4.3.3 Symbols assemblage settings

The following keys are available for arrows drawn with the default `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} \qquad \smalleroverrightarrow{AB} $
```

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

trim start={*number*} (no default, initially 7)

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

trim middle={*number*} (no default, initially set by `middle config=auto`)

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

trim end={*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`:

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

For `middle config = relbareda`, `middle`^{P.25} is set to `\relbareda`  and `trim middle`^{P.25} to 1.

For `middle config = harrowextender`, `middle`^{P.25} is set to `\harrowextender` and `trim middle`^{P.25} to 0.

For `middle config = auto`, if `\harrowextender` is provided by the `math font`⁸, `middle`^{P.25} is set with `middle config = harrowextender`. If `\harrowextender` isn't available, `middle`^{P.25} 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={\relbared}</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 `esvect \vv` (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:

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


⁸See the documentation of the package `unicode-math`.

where `tikz options`^{→P. 27} and `tikz command`^{→P. 27} 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 \overdotteddoublearrow{AB} $
```



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

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

Sets TikZ options to $\langle TikZ options \rangle$.

path options= $\{\langle path options \rangle\}$
 (no default, initially `arrows=--Classical TikZ Rightarrow`, `cap=round`)

Sets TikZ path options to $\langle path options \rangle$.

path= $\{\langle path specification \rangle\}$ (no default, initially `(0,0)--(1,0)`)

Sets TikZ path specification to $\langle path \rangle$ (the ending semicolon is automatically appended).

add tikz options= $\{\langle TikZ options \rangle\}$ (no default)

Appends the options $\langle TikZ options \rangle$ to the key `tikz options`.

add path options= $\{\langle path options \rangle\}$ (no default)

Appends the options $\langle path options \rangle$ to the key `path options`.

arrows= $\{\langle arrow specification \rangle\}$ (no default)

Appends the option `arrows= $\{\langle arrow specification \rangle\}$` to the key `path options`.

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

Appends the option `line width= $\{\langle length \rangle\}$` to the key `path options`.

thinner

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

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

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

4.3.5 PSTricks settings

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

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

where `geometry`, `line thickness` `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 with by:

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

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

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



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

picture command={*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.24} (see the section 5.1.2, page 31).

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

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

Defines the method `⟨name⟩`, to be used in commands `\NewOverArrowCommand`^{→P.17}, `\RenewOverArrowCommand`^{→P.17}, `\ProvideOverArrowCommand`^{→P.17} or `\DeclareOverArrowCommand`. When the `⟨name⟩` method is chosen, corresponding keys are defined by `⟨keys def⟩`. This must set, in particular, the keys `no stack macro hook` and `no arrow macro hook`^{→P.31}. 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.31}) is called with the math style, passed as argument (it can be, for example, the macro `\rightarrowfill@` used by `amsmath \overrightarrow`). **fill** is the mechanism used by the `symb` method.

lens if `arrow macro` creates fixed-length arrows, and needs the computation of lengths `\overarrowlength`^{→P.19}, `\overarrowthickness`^{→P.19} and `\overarrowsmallerthickness`^{→P.20}. In this case, the arrow macro (defined by `no arrow macro hook`^{→P.31}) 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` and `\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={ $\langle code \rangle$ } (no default)

Sets the $\langle code \rangle$ executed if **arrow macro** is left unset, after user evaluation of $\langle keys \rangle$ in `\NewOverArrowCommand`^{P.17}, `\RenewOverArrowCommand`^{P.17}, `\ProvideOverArrowCommand`^{P.17} or `\DeclareOverArrowCommand`^{P.17}.

$\langle code \rangle$ must configure **arrow macro**^{P.30} accordingly to the user keys setting.

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

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

5 Complements

5.1 Know issues

5.1.1 Math font change

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

5.1.2 Detection of non standard subscripts

The subscript detection enabled by the key `detect subscripts`^{P.24} 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 `spbookmark` package (<https://www.ctan.org/pkg/spbookmark>), by Qu Yi, which allows a complete customisation of subscripts, through the `\sub` command;
- with the `altsubsup` package (<https://www.ctan.org/pkg/altsubsup>), by Julien Labbé, which provides an alternative subscript format, and changes, for this purpose, the catcode of the underscore symbol “`_`” from 8 (*subscript* catcode category) to 12 (*other* catcode category).

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

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

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

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

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

Alternatively, two package options handle the cases where the catcode of the underscore “_” symbol is changed: `subother`^{→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 `altnumsub` package (no need of `\SetOverArrowsSubscriptCommand`^{→P.30}). 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 Sautrais, provides the fine vector macro `\vv`. This package is loaded by default by `overarrows`.

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

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

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

5.4 Changelog

- v1.2**
- Fix compatibility issues with `unicode-math`.
 - Allow to draw the arrow with `PSTricks`.
 - Make `esvect` handle all font sizes.

v1.1 Support for non-standard subscripts.

v1.0.1 Bug fix for `under*` options.

v1.0 Initial version.

6 Implementation

Management of options

Declaration of conditionals

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

Following conditionals are for predefined commands.

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

Declaration of options

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

Following options are for predefined commands.

```
44 \DeclareOption{overrightarrow}{\ovar@option@overrightarrow@true}
45 \DeclareOption{underrightarrow}{\ovar@option@underrightarrow@true}
46 \DeclareOption{overleftarrow}{\ovar@option@overleftarrow@true}
47 \DeclareOption{underleftarrow}{\ovar@option@underleftarrow@true}
48 \DeclareOption{overleftrightharpoon}{\ovar@option@overleftrightharpoon@true}
```

```

49 \DeclareOption{underleftrightarrow}{\var@option@underleftrightarrow@true}
50 \DeclareOption{overrightharpoonup}{\var@option@overrightharpoonup@true}
51 \DeclareOption{underrightharpoonup}{\var@option@underrightharpoonup@true}
52 \DeclareOption{overrightharpoondown}{\var@option@overrightharpoondown@true}
53 \DeclareOption{underrightharpoondown}{\var@option@underrightharpoondown@true}
54 \DeclareOption{overleftharpoonup}{\var@option@overleftharpoonup@true}
55 \DeclareOption{underleftharpoonup}{\var@option@underleftharpoonup@true}
56 \DeclareOption{overleftharpoondown}{\var@option@overleftharpoondown@true}
57 \DeclareOption{underleftharpoondown}{\var@option@underleftharpoondown@true}
58 \DeclareOption{overbar}{\var@option@overbar@true}
59 \DeclareOption{underbar}{\var@option@underbar@true}

```

Following options are for sets of predefined commands.

```

60 \DeclareOption{overcommands}{%
61   \var@option@overrightarrow@true
62   \var@option@overleftarrow@true
63   \var@option@overleftrightarrow@true
64   \var@option@overrightharpoonup@true
65   \var@option@overrightharpoondown@true
66   \var@option@overleftharpoonup@true
67   \var@option@overleftharpoondown@true
68   \var@option@overbar@true
69 }
70 \DeclareOption{undercommands}{%
71   \var@option@underrightarrow@true
72   \var@option@underleftarrow@true
73   \var@option@underleftrightarrow@true
74   \var@option@underrightharpoonup@true
75   \var@option@underrightharpoondown@true
76   \var@option@underleftharpoonup@true
77   \var@option@underleftharpoondown@true
78   \var@option@underbar@true
79 }
80 \DeclareOption{allcommands}{%
81   \var@option@overrightarrow@true
82   \var@option@underrightarrow@true
83   \var@option@overleftarrow@true
84   \var@option@underleftarrow@true
85   \var@option@overleftrightarrow@true
86   \var@option@underleftrightarrow@true
87   \var@option@overrightharpoonup@true
88   \var@option@underrightharpoonup@true
89   \var@option@overrightharpoondown@true
90   \var@option@underrightharpoondown@true
91   \var@option@overleftharpoonup@true
92   \var@option@underleftharpoonup@true
93   \var@option@overleftharpoondown@true
94   \var@option@underleftharpoondown@true
95   \var@option@overbar@true
96   \var@option@underbar@true
97 }

```

Options processing

```

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

```

Package dependencies

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

```

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

```

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

```

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

```

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

```

Option `tikz`^{→ P. 16}.

```

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

```

Option `pstricks`^{→ P. 16}.

```

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

```

Option `pstarrows`^{→ P. 17}.

```

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

```

Configuration of subscripts detection

`\SetOverArrowsSubscriptCommand`

Sets the subscript command.

```

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

```

Initial configuration.

```

132 \SetOverArrowsSubscriptCommand{_{}}

```

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

```

133 \ifovar@option@subother@
134   \begingroup
135     \catcode `_=12
136     \SetOverArrowsSubscriptCommand{_{}}%
137   \endgroup
138 \fi

```

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

```

139 \ifovar@option@subactive@
140   \begingroup
141   \catcode `_=13
142   \SetOverArrowsSubscriptCommand{_}%
143   \endgroup
144 \fi

```

Management of keys

Family declaration and setters

```

145 \RequirePackage{pgfkeys}
146 \pgfkeys{overarrows/.is family}
\ovar@set
147 \newcommand{\ovar@set}[1]{\pgfqkeys{/overarrows}{#1}}
\SetOverArrowsMethod
148 \NewDocumentCommand{\SetOverArrowsMethod}{s O{fill} m O{} m }{%
149   \IfBooleanTF{#1}{%
150     \csgdef{\ovar@set@#3}{#4\ovar@set{#5}}%
151   }{%
152     \csgdef{\ovar@set@#3}{#4\ovar@set{%
153       no stack macro hook/.code={%
154         \ovar@set{stack macro/.expanded={%
155           \expandafter\expandonce\csname ovar@stack@#2\endcsname%
156           {\expandonce\ovar@length@min}%
157           {\expandonce\ovar@before@arrow}{\expandonce\ovar@after@arrow}%
158         }}%
159       },#5}}%
160     }%
161   }

```

Common keys

```

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

```

`detect subscripts`^{→P.24}.

```

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

```

`stack macro`^{→P.30} and `arrow macro`^{→P.30}.

```

164   stack macro/.store in=\ovar@macro@stack,
165   arrow macro/.store in=\ovar@macro@arrow,
166   stack macro/.value required,
167   arrow macro/.value required,

```

`no stack macro hook`^{→P.30}, `no arrow macro hook`^{→P.31}. These two keys must be redefined by the command `\ovar@set@{method}`.

```

168   no stack macro hook/.code={%
169     \PackageError{overarrows}{Undefined stack macro}
170     {The requested method is perhaps misspelled}
171   },
172   no arrow macro hook/.code={%
173     \PackageError{overarrows}{Undefined arrow macro}
174     {The requested method is perhaps misspelled}
175   },

```

`min length`^{→P.22}.

```

176   min length/.store in=\ovar@length@min,
177   min length/.value required,
178   min length=0,

```

before arrow^{→P.24}, after arrow^{→P.24}, space before arrow^{→P.24}, space after arrow^{→P.24}.

```

179 before arrow/.store in=\ovar@before@arrow,
180 after arrow/.store in=\ovar@after@arrow,
181 before arrow/.value required,
182 after arrow/.value required,
183 before arrow=\empty,
184 after arrow=\empty,
185 space before arrow/.code=\pgfkeysalso{before arrow={\kern ##1}},
186 space after arrow/.code=\pgfkeysalso{after arrow={\kern ##1}},

```

shift left^{→P.23}, shift right^{→P.23}, shift leftright^{→P.24}, center arrow^{→P.24}, left arrow^{→P.24}, right arrow^{→P.24}.

```

187 shift left/.store in=\ovar@shift@left,
188 shift right/.store in=\ovar@shift@right,
189 shift left/.value required,
190 shift right/.value required,
191 shift leftright/.code=\pgfkeysalso{%
192   shift left=##1, shift right=##1,
193 },
194 center arrow/.code=\pgfkeysalso{shift leftright=0},
195 shift leftright/.value required,
196 center arrow/.value forbidden,
197 left arrow/.code=\pgfkeysalso{%
198   shift left=0, shift right=##1,
199 },
200 right arrow/.code=\pgfkeysalso{%
201   shift left=##1, shift right=0,
202 },
203 left arrow/.default=2,
204 right arrow/.default=2,
205 right arrow,

```

arrow under^{→P.23}.

```

206 arrow under/.is choice,
207 arrow under/noconfig/.code={
208   \def\ovar@stack@fill{\ovar@stackunder@fill}
209   \def\ovar@stack@lens{\ovar@stackunder@lens}
210 },
211 arrow under/autoconfig/.code={
212   \pgfkeysalso{%
213     arrow under=noconfig,
214     detect subscripts=false,
215     before arrow={\kern 1.3\ex@}\relax},% like underarrow@ from amsmath
216   }
217 },
218 arrow under/.default=autoconfig,
219 }

```

Keys for the symb method

```

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

```

Fill macro.

```

221 fill macro/.store in=\ovar@macro@arrowfill,
222 fill macro/.value required,

```

Arrow macro.

```

223 no arrow macro hook/.code={%
224   \ifdef{\ovar@macro@arrowfill}{-}{%
225     \ovar@set{%

```

```

226     fill macro/.expanded={%
227         \noexpand\ovar@arrow@fill%
228         {\expandonce\ovar@shift@left}{\expandonce\ovar@shift@right}%
229     }
230 }
231 }
232 \ovar@set{%
233     arrow macro/.expanded={%
234         \expandonce{\ovar@macro@arrow@fill}%
235         {\expandonce{\ovar@arrow@start}\expandonce{\ovar@trim@start}}%
236         {\expandonce{\ovar@trim@middle}\expandonce{\ovar@arrow@middle}%
237         \expandonce{\ovar@trim@middle}}%
238         {\expandonce{\ovar@trim@end}\expandonce{\ovar@arrow@end}}%
239     }
240 }
241 },

```

start^{→P. 25}, middle^{→P. 25}, end^{→P. 25}.

```

242     start/.store in=\ovar@arrow@start,
243     middle/.store in=\ovar@arrow@middle,
244     end/.store in=\ovar@arrow@end,
245     start/.value required,
246     middle/.value required,
247     end/.value required,

```

trim start^{→P. 25}, trim middle^{→P. 25}, trim end^{→P. 25}, trim^{→P. 25}, no trimming^{→P. 25}.

```

248     trim start/.code={\def\ovar@trim@start{\xjoinrel[##1]}},
249     trim middle/.code={\def\ovar@trim@middle{\xjoinrel[##1]}},
250     trim end/.code={\def\ovar@trim@end{\xjoinrel[##1]}},
251     trim start/.value required,
252     trim middle/.value required,
253     trim end/.value required,
254     trim/.code={\pgfkeysalso{trim start={##1}, trim middle={##1}, trim end={##1}}},
255     trim/.value required,
256     no trimming/.code={%
257         \let\ovar@trim@start\empty
258         \let\ovar@trim@middle\empty
259         \let\ovar@trim@end\empty
260     },
261     no trimming/.value forbidden,

```

middle config^{→P. 26}.

```

262     middle config/.is choice,
263     middle config/.value required,
264     middle config/relbar/.code=\pgfkeysalso{%
265         middle={\relbar},
266         trim middle={2.5},
267     },
268     middle config/relbareda/.code={%
269         \ifundef{\relbareda}{%
270             \PackageWarning{overarrows}{Key 'middle config=relbareda' used,
271                 \MessageBreak%
272                 but \protect\relbareda\space is undefined; ignored.
273                 \MessageBreak%
274                 Load 'esvect' package, or use 'esvect' option \MessageBreak%
275                 to remove this warning}
276         }{%
277             \pgfkeysalso{%
278                 middle={\relbareda},
279                 trim middle={1},
280             }

```

```

281     }
282 },

283 middle config/harrowextender/.code={%
284     \pgfkeysalso{%
285         middle={\harrowextender},
286         trim middle={0},
287     }
288 },

289 middle config/auto/.code={%
290     \ifundef{\harrowextender}{%
291         \ifovar@option@esvect@
292             \pgfkeysalso{middle config=relbareda}
293         \else
294             \pgfkeysalso{middle config=relbar}
295         \fi
296     }{%
297         \pgfkeysalso{middle config=harrowextender}
298     }
299 },

```

`amsmath` → P. 26.

```

300 amsmath/.is choice,%
301 amsmath/mimic/.code=\pgfkeysalso{%
302     start={\relbar}, middle={\relbar}, end={\rightarrow},
303     trim start=7,
304     trim middle=2,
305     trim end=7,
306     shift leftright=0,
307     after arrow={}, before arrow={},
308 },
309 amsmath/strict/.code=\pgfkeysalso{%
310     amsmath=mimic,
311     no trimming,
312     fill macro={\arrowfill@}, stack macro={\overarrow@},
313 },
314 amsmath/.default=mimic,

```

`esvect` → P. 26.

```

315 esvect/.is choice,%
316 esvect/mimic/.code=\pgfkeysalso{%
317     start={\relbared}, middle={\relbareda}, end={\fldr},
318     trim start=1.5,
319     trim end=1.5,
320     trim middle=0,
321     right arrow=2,
322     space before arrow=-.7pt,
323     space after arrow=-.3pt,
324 },
325 esvect/strict/.code=\pgfkeysalso{%
326     esvect=mimic,
327     no trimming,
328     fill macro={\traitfill@}, stack macro={\overvect@},
329 },
330 esvect/.default=mimic,

```

Initial configuration.

```

331 amsmath, middle config=auto, end=\ovar@rightarrow, right arrow,
332 }

```

Keys for the `tikz` method

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

Arrow macro.

```
334 no arrow macro hook/.code={%
335 \ifdef{\ovar@tikz@command}{%
336 \pgfkeysgetvalue{/overarrows/path options}{\ovar@tikz@pathoptions}
337 \ovar@set{%
338 tikz command/.expanded={%
339 \noexpand\draw[\expandonce\ovar@tikz@pathoptions]\expandonce\ovar@tikz@path;
340 }
341 }
342 }
343 \pgfkeysgetvalue{/overarrows/tikz options}{\ovar@tikz@options}
344 \ovar@set{%
345 arrow macro/.expanded={%
346 $ \noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
347 \noexpand\tikz[\expandonce{\ovar@tikz@options}]{\expandonce{\ovar@tikz@command}}%
348 $ \noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
349 }
350 }
351 },
```

TikZ parts: `tikz command` ^{→ P. 27}, `tikz options` ^{→ P. 27}, `path options` ^{→ P. 27}, `path` ^{→ P. 27}.

```
352 tikz command/.store in=\ovar@tikz@command,
353 tikz options/.initial={x=\overarrowlength, line width=\overarrowthickness},
354 path options/.initial={arrows={-Classical TikZ Rightarrow}, cap=round},
355 path/.store in=\ovar@tikz@path,
356 path={(0,0)--(1,0)},
357 tikz command/.value required,
358 tikz options/.value required,
359 path options/.value required,
360 path/.value required,
```

TikZ handy keys: `add path options` ^{→ P. 27}, `add tikz options` ^{→ P. 27}, `arrows` ^{→ P. 27}, `line thickness` ^{→ P. 27}, `thinner` ^{→ P. 27}.

```
361 add path options/.code=\pgfkeysalso{%
362 path options/.append={, ##1}},%
363 add tikz options/.code=\pgfkeysalso{%
364 tikz options/.append={, ##1}},%
365 arrows/.code=\pgfkeysalso{add path options={arrows={##1}}},%
366 line thickness/.code=\pgfkeysalso{add path options={line width=##1}},%
367 thinner/.code=\pgfkeysalso{line thickness={\overarrowsmallerthickness}},%
368 add path options/.value required,%
369 add tikz options/.value required,%
370 arrows/.value required,%
371 line thickness/.value required,%
372 thinner/.value forbidden,%
```

Initial configuration.

```
373 shift right=-2,
374 min length=12,
375 }
```

Keys for the pstricks method

```
376 \SetOverArrowsMethod[lens]{pstricks}{%
```

Arrow macro.

```
377 no arrow macro hook/.code={%
378 \ovar@set{%
```



```

379     arrow macro/.expanded={%
380       $\noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
381       \noexpand\begin{pspicture}\expandonce{\ovar@pstricks@geometry}%
382       \noexpand\psset{linewidth=\expandonce{\ovar@pstricks@linethickness}}%
383       \noexpand\psset{\expandonce{\ovar@pstricks@psset}}%
384       \expandonce{\ovar@pstricks@command}%
385       \noexpand\end{pspicture}%
386       $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
387     }
388   }
389 },

```

Pstricks parts: pstricks command^{→P.28}, psset^{→P.28}, geometry^{→P.28}, line thickness^{→P.28}.

```

390   pstricks command/.store in=\ovar@pstricks@command,
391   psset/.store in=\ovar@pstricks@psset,
392   geometry/.store in=\ovar@pstricks@geometry,
393   line thickness/.store in=\ovar@pstricks@linethickness,
394   pstricks command/.value required,
395   psset/.value required,
396   geometry/.value required,
397   line thickness/.value required,

```

Pstricks handy key: arrow^{→P.28}, thinner^{→P.28}.

```

398   arrow/.style={pstricks command={\psline{##1}(0,0)(\overarrowlength,0)}},%
399   arrow/.value required,%
400   thinner/.style={line thickness={\overarrowsmallerthickness}},%
401   thinner/.value forbidden,%

```

Initial configuration.

```

402   shift right=-2,
403   min length=12,
404   geometry={(0,-0.5ex)(\overarrowlength,0.5ex)},%
405   line thickness={\overarrowthickness},%
406   arrow={->},%
407   psset={},%
408 }

```

Keys for the picture method

```

409 \SetOverArrowsMethod[lens]{picture}{%

```

Arrow macro.

```

410   no arrow macro hook/.code={%
411     \ovar@set{%
412       arrow macro/.expanded={%
413         $\noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
414         \noexpand\begin{picture}\expandonce{\ovar@picture@geometry}%
415         \noexpand\linethickness{\expandonce{\ovar@picture@linethickness}}%
416         \expandonce{\ovar@picture@command}%
417         \noexpand\end{picture}%
418         $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
419       }
420     }
421   },

```

Picture parts: picture command^{→P.29}, geometry^{→P.29}, line thickness^{→P.29}.

```

422   picture command/.store in=\ovar@picture@command,
423   geometry/.store in=\ovar@picture@geometry,
424   line thickness/.store in=\ovar@picture@linethickness,
425   picture command/.value required,
426   geometry/.value required,
427   line thickness/.value required,

```

Picture handy key: `thinner`^{→ P. 29}.

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

Initial configuration.

```
429 shift right=-2,
430 min length=18,
431 geometry={(\overarrowlength,1ex)(0,-0.5ex)},%
432 line thickness={\overarrowthickness},%
433 picture command={\put(0,0){\vector(1,0){\overarrowlength}}},%
434 }
```

Commands

Macros for symbols assemblage

```
\xjoinrel
435 \ifdef{\xjoinrel}{%
436 \PackageWarning{overarrows}{Command \protect\xjoinrel\space already defined.
437 \MessageBreak%
438 Previous definition will be overridden}
439 }
```

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.

```
440 \DeclareRobustCommand{\xjoinrel}[1][3.5]{\mathrel{\mkern-#1mu}}
\smallermathstyle
441 \newcommand*{\smallermathstyle}{%
442 \mathchoice{\scriptstyle}{\scriptstyle}{\scriptscriptstyle}{}}
443 }
```

`\ovar@arrow@fill`

Macro used for default fill macro^{→ P. 31}.

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

```
444 \def\ovar@arrow@fill#1#2#3#4#5#6{%
445 $ \mkern #1 mu\relax#6#3%
446 \mkern #1 mu\relax#6#3%
447 \cleaders\hbox{$#6#4$}\hfill%
448 #5\mkern #2 mu\relax$%
449 }
```

Macros for fixed length arrows

Lengths declaration.

```
450 \newlength{\overarrowlength}
451 \newlength{\overarrowthickness}
452 \newlength{\overarrowsmallerthickness}
453 \newlength{\ovar@extralength}
454 \newlength{\ovar@tempdim}
```

`\ovar@set@arrowlength`

Sets `\overarrowlength`^{→ P. 19}.

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

```
455 \def\ovar@set@arrowlength#1#2#3{%
456 \settowidth{\ovar@tempdim}{$\mkern#2\mskip #1 mu\relax$}%
457 \settowidth{\overarrowlength}{$\mkern#2#3$}%

```

```

458 \ifdim \overarrowlength < \ovar@tempdim \overarrowlength=\ovar@tempdim\fi%
459 }

```

```

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

```

Sets `\overarrowthickness`^{→P.19} and `\overarrowsmallerthickness`^{→P.20}.

#1: math style

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

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

```

460 \def\ovar@rulethickness@fontdimen{8}
461 \def\ovar@rulethickness@family{3}
462 \def\ovar@set@arrowthickness#1{%
463   \ifx#1\displaystyle%
464     \overarrowthickness =
465       \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
466     \overarrowsmallerthickness =
467       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
468   \else\ifx#1\textstyle%
469     \overarrowthickness =
470       \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
471     \overarrowsmallerthickness =
472       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
473   \else\ifx#1\scriptstyle%
474     \overarrowthickness =
475       \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
476     \overarrowsmallerthickness =
477       \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
478   \else%
479     \overarrowthickness =
480       \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
481     \overarrowsmallerthickness = \overarrowthickness%
482   \fi\fi\fi%
483 }

```

`unicode-math` with LuaTeX version.

```

484 \def\ovar@set@arrowthickness@UM@lua#1{%
485   \overarrowthickness = \Umathoverbarrule #1
486   \ifx#1\displaystyle%
487     \overarrowsmallerthickness = \Umathoverbarrule \textstyle%
488   \else\ifx#1\textstyle%
489     \overarrowsmallerthickness = \Umathoverbarrule \scriptstyle%
490   \else%
491     \overarrowsmallerthickness = \Umathoverbarrule \scriptscriptstyle%
492   \fi\fi%
493 }

```

Test which version to use.

```

494 \AtBeginDocument{%
495   \@ifpackageloaded{unicode-math-luatex}
496   {%
497     \global\let\ovar@set@arrowthickness\ovar@set@arrowthickness@UM@lua
498   }
499   {%
500     \@ifpackageloaded{unicode-math-xetex}
501     {%
502       \gdef\ovar@rulethickness@fontdimen{54}
503       \gdef\ovar@rulethickness@family{2}
504     }

```

```

505     {}
506   }
507 }

```

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

```

508 \def\ovar@stackover@@#1#2#3#4#5#6{\vbox{\ialign{##\crrc%
509     $#5\mskip #1 mu\relax$\crrc%
510     \noalign{#2\nointerlineskip}#4\crrc%
511     \noalign{#3\nointerlineskip}%
512     $m@th\hfil#5#6\hfil$\crrc%
513   }%
514 }%
515 }
516 \def\ovar@stackunder@@#1#2#3#4#5#6{\vtop{\ialign{##\crrc%
517     $m@th\hfil#5#6\hfil$\crrc%
518     \noalign{#2\nointerlineskip}#4\crrc%
519     \noalign{#3\nointerlineskip}%
520     $#5\mskip #1 mu\relax$\crrc%
521   }%
522 }%
523 }

```

`\ovar@stackover@`
`\ovar@stackunder@`

Stack macros without min arrow length.
#1: vertical mode material before arrow
#2: vertical mode material after arrow
#3: arrow macro
#4: math style
#5: content

```

524 \def\ovar@stackover@#1#2#3#4#5{\ovar@stackover@@{0}{#1}{#2}{#3}{#4}{#5}}
525 \def\ovar@stackunder@#1#2#3#4#5{\ovar@stackunder@@{0}{#1}{#2}{#3}{#4}{#5}}

```

`\ovar@stackover@fill`
`\ovar@stackunder@fill`
`\ovar@stack@fill`

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

```

526 \def\ovar@stackover@fill#1#2#3#4#5#6{\ovar@stackover@@{#1}{#2}{#3}{#4#5}{#5}{#6}}
527 \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. 23}.

```

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

```

`\ovar@stackover@lens`
`\ovar@stackunder@lens`
`\ovar@stack@lens`

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

#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

```

529 \def\ovar@stackover@lens#1#2#3#4#5#6{%
530   \ovar@set@arrowlength{#1}{#5}{#6}%
531   \ovar@set@arrowthickness{#5}%
532   \ovar@stackover@{#2}{#3}{#4}{#5}{#6}%
533 }
534 \def\ovar@stackunder@lens#1#2#3#4#5#6{%
535   \ovar@set@arrowlength{#1}{#5}{#6}%
536   \ovar@set@arrowthickness{#5}%
537   \ovar@stackunder@{#2}{#3}{#4}{#5}{#6}%
538 }

```

`\ovar@stack@lens` matches the macro `\ovar@stackover@lens` by default, or `\ovar@stackunder@lens` with arrow under^{→ P. 23}.

```

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

```

Macro for commands creation

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

```

540 \ExplSyntaxOn
541 \cs_new_protected:Npn \__overarrows_processor_strip_escape_char:n #1
542 {
543   \regex_match:nnTF { \cC. } { #1 }
544   { \tl_set:Nx \ProcessedArgument { \cs_to_str:N #1 } }
545   { \tl_set:Nx \ProcessedArgument { #1 } }
546 }
547 \cs_new_eq:NN \ovar@cmdname@processor \__overarrows_processor_strip_escape_char:n
548 \ExplSyntaxOff

```

```

\DeclareOverArrowCommand
549 \NewDocumentCommand{\DeclareOverArrowCommand}{
550   0{symp} >{\ovar@cmdname@processor} m m
551 }{%
552   \begingroup
553   \ovar@set@common
554   \ifcsdef\ovar@set@#1{%
555     \csuse\ovar@set@#1
556   }{%
557     \PackageError{overarrows}{Unknown method #1}
558     {Try with 'symp', 'tikz' or 'picture'}
559   }
560   \ovar@set@#3
561   \ifdef{\ovar@macro@arrow}{%
562     \ovar@set{no arrow macro hook}
563   }
564   \ifdef{\ovar@macro@stack}{%
565     \ovar@set{no stack macro hook}
566   }
567   \csxdef\ovar@#2@normal{%
568     \noexpand\mathpalette{
569       \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}%

```

```

570 }
571 }
572 \csxdef{ovar@#2@starred}{%
573   \noexpand\mathpalette{%
574     \noexpand\ovar@starversion{%
575       \expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}}%
576     }
577   }
578 }
579 \ifovar@detectsubscripts@%
580 \csgdef{ovar@#2@auto}##1{%
581   \@ifnextchar \ovar@subcmd {%
582     \csuse{ovar@#2@starred}{##1}%
583   }{%
584     \csuse{ovar@#2@normal}{##1}%
585   }%
586 }
587 \csgdef{#2}{%
588   \@ifstar{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@auto}}%
589 }
590 \else
591 \csgdef{#2}{%
592   \@ifstar{\csuse{ovar@#2@starred}}{\csuse{ovar@#2@normal}}%
593 }
594 \fi
595 \ifovar@option@debug@
596 \PackageInfo{overarrows}{%
597   Meaning of \protect\ovar@#2@normal\MessageBreak
598   used for \@backslashchar#2:\MessageBreak%
599   \expandafter\meaning\csname ovar@#2@normal\endcsname}
600 \fi
601 \endgroup
602 }
\ProvideOverArrowCommand
603 \NewDocumentCommand{\ProvideOverArrowCommand}{%
604   O{symp} >{\ovar@cmdname@processor} m m
605 }{%
606   \ifcsdef{#2}{}{
607     \DeclareOverArrowCommand[#1]{#2}{#3}
608   }
609 }
\NewOverArrowCommand
610 \NewDocumentCommand{\NewOverArrowCommand}{%
611   O{symp} >{\ovar@cmdname@processor} m m
612 }{%
613   \ifcsdef{#2}{%
614     \PackageError{overarrows}{Command \csname #2\endcsname already defined}%
615     {You have used \protect\NewOverArrowCommand\space with a command that
616       already has a definition. \MessageBreak%
617       Choose another name, or use instead \protect\DeclareOverArrowCommand.}
618   }{%
619     \DeclareOverArrowCommand[#1]{#2}{#3}
620   }
621 }
\RenewOverArrowCommand
622 \NewDocumentCommand{\RenewOverArrowCommand}{%
623   O{symp} >{\ovar@cmdname@processor} m m
624 }{%
625   \ifcsundef{#2}{%
626     \PackageError{overarrows}{Command \csname #2\endcsname undefined}%
627     {You have used \protect\RenewOverArrowCommand\space with a command that was
628       never defined. \MessageBreak%
629       Check the requested name, or use instead \protect\NewOverArrowCommand.}
630   }{%

```

```

631 \DeclareOverArrowCommand[#1]{#2}{#3}
632 }
633 }

```

Starred variant

\ovar@starversion

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

```

634 \def\ovar@starversion#1#2#3{%
635   #1#2{#3}%
636   \settowidth{\ovar@extralength}{\m@th#1#2{#3}}
637   \settowidth{\ovar@tempdim}{\m@th#2{#3}}
638   \deflength{\ovar@extralength}{0.5\ovar@extralength-0.5\ovar@tempdim}%
639   \kern-\ovar@extralength%
640 }

```

\vv vector command

\vv

\esvectvv

Backup and redefinition of esvect \vv^{→P.20} vector command.

```

641 \ifovar@option@esvect@
642 \let\esvectvv\vv
643 \undef\vv
644 \NewOverArrowCommand{\vv}{esvect, middle config = relbareda}
645 \fi

```

Predefined commands

Declare predefined commands after unicode-math settings.

```

646 \DeclareHookRule{begindocument}{overarrows}{after}{unicode-math-luatex}
647 \DeclareHookRule{begindocument}{overarrows}{after}{unicode-math-xetex}
648 \AddToHook{begindocument}[overarrows]
649 {

```

\overrightarrow

```

650 \ifovar@option@overrightarrow@
651 \DeclareOverArrowCommand{\overrightarrow}{%
652   amsmath, middle config=relbar,
653   end=\ovar@rightarrow,
654   right arrow,
655 }
656 \fi

```

\underrightarrow

```

657 \ifovar@option@underrightarrow@
658 \DeclareOverArrowCommand{\underrightarrow}{%
659   amsmath, middle config=relbar,
660   end=\ovar@rightarrow,
661   right arrow,
662   arrow under,
663 }
664 \fi

```

\overleftarrow

```

665 \ifovar@option@overleftarrow@
666 \DeclareOverArrowCommand{\overleftarrow}{%
667   amsmath, middle config=relbar,
668   start=\ovar@leftarrow,
669   end=\relbar,
670   left arrow,

```

```

671     }
672     \fi
\underrightarrow
673     \ifovar@option@underrightarrow@
674     \DeclareOverArrowCommand{\underrightarrow}{%
675         amsmath, middle config=relbar,
676         start=\ovar@leftarrow,
677         end=\relbar,
678         left arrow,
679         arrow under,
680     }
681     \fi
\overleftrightharpoonup
682     \ifovar@option@overleftrightharpoonup@
683     \DeclareOverArrowCommand{\overleftrightharpoonup}{%
684         amsmath, middle config=relbar,
685         start=\ovar@leftarrow,
686         end=\ovar@rightarrow,
687         center arrow,
688     }
689     \fi
\underrleftrightharpoonup
690     \ifovar@option@underrleftrightharpoonup@
691     \DeclareOverArrowCommand{\underrleftrightharpoonup}{%
692         amsmath, middle config=relbar,
693         start=\ovar@leftarrow,
694         end=\ovar@rightarrow,
695         center arrow,
696         arrow under,
697     }
698     \fi
\overrightharpoonup
699     \ifovar@option@overrightharpoonup@
700     \DeclareOverArrowCommand{\overrightharpoonup}{%
701         amsmath, middle config=relbar,
702         end=\rightharpoonup,
703         right arrow,
704     }
705     \fi
\underrightharpoonup
706     \ifovar@option@underrightharpoonup@
707     \DeclareOverArrowCommand{\underrightharpoonup}{%
708         amsmath, middle config=relbar,
709         end=\rightharpoonup,
710         right arrow,
711         arrow under,
712     }
713     \fi
\overrightharpoonupdown
714     \ifovar@option@overrightharpoonupdown@
715     \DeclareOverArrowCommand{\overrightharpoonupdown}{%
716         amsmath, middle config=relbar,
717         end=\rightharpoonupdown,
718         right arrow,
719     }
720     \fi
\underrightharpoonupdown
721     \ifovar@option@underrightharpoonupdown@
722     \DeclareOverArrowCommand{\underrightharpoonupdown}{%
723         amsmath, middle config=relbar,
724         end=\rightharpoonupdown,
725         right arrow,
726         arrow under,
727     }
728     \fi
\overlefttharpoonup

```



```

729 \ifovar@option@overleftharpoonup@
730 \DeclareOverArrowCommand{\overleftharpoonup}{%
731   amsmath, middle config=relbar,
732   start=\leftharpoonup,
733   end=\relbar,
734   left arrow,
735 }
736 \fi

\underleftharpoonup
737 \ifovar@option@underleftharpoonup@
738 \DeclareOverArrowCommand{\underleftharpoonup}{%
739   amsmath, middle config=relbar,
740   start=\leftharpoonup,
741   end=\relbar,
742   left arrow,
743   arrow under,
744 }
745 \fi

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

\underleftharpoondown
754 \ifovar@option@underleftharpoondown@
755 \DeclareOverArrowCommand{\underleftharpoondown}{%
756   amsmath, middle config=relbar,
757   start=\leftharpoondown,
758   end=\relbar,
759   left arrow,
760   arrow under,
761 }
762 \fi

\overbar
763 \ifovar@option@overbar@
764 \DeclareOverArrowCommand{\overbar}{%
765   amsmath, middle config=relbar,
766   start={\std@minus}, end={\std@minus},% \relbar is defined with \mathsm@sh
767   shift leftright=0,
768   space after arrow=-0.3ex,
769 }
770 \fi

\underbar
771 \ifovar@option@underbar@
772 \DeclareOverArrowCommand{\underbar}{%
773   amsmath, middle config=relbar,
774   start={\vphantom{+}\std@minus}, end={\std@minus},% \relbar is defined with \mathsm@sh
775   shift leftright=0,
776   arrow under,
777   space before arrow=-0.3ex,
778 }
779 \fi

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

780 }
```

Test macros

\ovar@testmathstyles

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

```

781 \newcommand{\ovar@testmathstyles}[2][]{
782   \begin{group}
783     \newcommand*{\ovar@row@teststyle}[1]{%
784       $\displaystyle ##1$
785       & $\textstyle ##1$
786       & $\scriptstyle ##1$
787       & $\scriptscriptstyle ##1$
788       \\
789     }
790     \renewcommand*{\arraystretch}{1.5}
791     \begin{tabular*}{0.95\linewidth}{@{\extracolsep{\fill}} cccc}
792       \hline
793       \footnotesize\texttt{\texttt{\textbackslash displaystyle}}
794       & \footnotesize\texttt{\texttt{\textbackslash textstyle}}
795       & \footnotesize\texttt{\texttt{\textbackslash scriptstyle}}
796       & \footnotesize\texttt{\texttt{\textbackslash scriptscriptstyle}}
797       \\
798       \hline
799       \ovar@row@teststyle{\csuse{#2}{v}}
800       \ovar@row@teststyle{\csuse{#2}{AB}}
801       \ovar@row@teststyle{\csuse{#2}{\mathrm{grad}}}
802       \ovar@row@teststyle{\csuse{#2}{my~long~vector}}
803       \IfValueT{#1}{\ovar@row@teststyle{\csuse{#2}{#1}}}
804       \hline
805     \end{tabular*}
806   \end{group}
807 }
```

\ovar@testkerning

```

808 \begin{group}
809 \ifovar@option@subother@ \catcode \_ =12 \fi
810 \ifovar@option@subactive@ \catcode \_ =13 \fi
811 \gdef\ovar@testkerning#1{%
812   \begin{displaymath}
813     #1\{t\}_\{#1\{u\}_\{#1\{v\}\}
814     \quad
815     #1\{\imath\}_0
816     \quad
817     #1\{v\}
818     = #1\{v\}_x + #1\{v\}_y + #1\{v\}_z
819     = v_x #1\{\imath\} + v_y #1\{\jmath\} + v_z #1\{k\}
820   \end{displaymath}
821 }
822 \end{group}
```

\TestOverArrow

```

823 \NewDocumentCommand{\TestOverArrow}{
824   s o >{\ovar@cmdname@processor} m
825 }{%
826   \ifcsdef{#3}{}%
827   \PackageWarning{overarrows}{Unknown name '#3' passed to
828     \protect\TestOverArrow}
829 }
830 \IfBooleanTF{#1}{%
831   \noindent\framebox{%
832     \begin{minipage}{0.95\linewidth}
833       \centering
834       \noindent\textbf{\large%
835         Test of \texttt{\textbackslash#3} and \texttt{\textbackslash#3*} macros}
836       \bigskip\par
837       \textbf{\texttt{\textbackslash#3} for different math styles}
```

```

838     \smallskip\par
839     \ovar@testmathstyles[#2]{#3}%
840     \bigskip\par
841     \textbf{\texttt{\textbackslash#3} kerning}
842     \ovar@testkerning{\csuse{#3}}
843     \textbf{\texttt{\textbackslash#3*} kerning}
844     \ovar@testkerning{\csuse{#3}*}
845     \end{minipage}%
846   }\bigskip\par
847   }{%
848     \ovar@testmathstyles[#2]{#3}%
849   }
850 }

```

Index

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

Package options

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

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

- `arrows` key, 27

- `before arrow` key, 24

- `center arrow` key, 24

Commands

- `\DeclareOverArrowCommand`, 17, 45–49
- `\esvectvv`, 20, 47
- `\NewOverArrowCommand`, 17, 46, 47
- `\overbar`, 21, 34, 49
- `\overleftarrow`, 21, 33, 47
- `\overleftharpoonup`, 21, 34, 49
- `\overleftharpoondown`, 21, 34, 49
- `\overleftrightharpoonup`, 21, 33, 48
- `\overrightarrow`, 21, 33, 47
- `\overrightarrowharpoonup`, 21, 34, 48
- `\overrightarrowharpoondown`, 21, 34, 48
- `\ProvideOverArrowCommand`, 17, 46
- `\RenewOverArrowCommand`, 17, 46
- `\SetOverArrowsMethod`, 30, 36, 37, 40, 41
- `\SetOverArrowsMethod*`, 30
- `\SetOverArrowsSubscriptCommand`, 30, 35, 36
- `\smallermathstyle`, 19, 42
- `\TestOverArrow`, 18, 50
- `\TestOverArrow*`, 18
- `\underbar`, 22, 34, 49
- `\underleftarrow`, 22, 33, 48
- `\underleftharpoonup`, 22, 34, 49
- `\underleftharpoondown`, 22, 34, 49
- `\underleftrightharpoonup`, 22, 34, 48
- `\underrightharpoonup`, 21, 33, 47
- `\underrightharpoondown`, 22, 34, 48
- `\underrightharpoonup`, 22, 34, 48
- `\vv`, 20, 47
- `\vv*`, 20
- `\xjoinrel`, 19, 38, 42

- `debug` package option, 17

- `\DeclareOverArrowCommand`, 17

- `detect subscripts` key, 24

- `end` key, 25

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

`fill` macro key, 31

`geometry` key, 28, 29

Internal macros

`\ifovar@detectsubscripts@`, 33, 46
`\ifovar@option@debug@`, 33, 46
`\ifovar@option@esvect@`, 33, 35, 39, 47
`\ifovar@option@oldarrows@`, 33, 35
`\ifovar@option@overbar@`, 33, 49
`\ifovar@option@overleftarrow@`, 33, 47
`\ifovar@option@overleftharpoondown@`, 33, 49
`\ifovar@option@overleftharpoonup@`, 33, 49
`\ifovar@option@overleftrightarrow@`, 33, 48
`\ifovar@option@overrightarrow@`, 33, 47
`\ifovar@option@overrightharpoondown@`, 33, 48
`\ifovar@option@overrightharpoonup@`, 33, 48
`\ifovar@option@pstarrows@`, 33, 35
`\ifovar@option@subactive@`, 33, 36, 50
`\ifovar@option@subother@`, 33, 35, 50
`\ifovar@option@tikz@`, 33, 35
`\ifovar@option@underbar@`, 33, 49
`\ifovar@option@underleftarrow@`, 33, 48
`\ifovar@option@underleftharpoondown@`, 33, 49
`\ifovar@option@underleftharpoonup@`, 33, 49
`\ifovar@option@underleftrightarrow@`, 33, 48

`\ifovar@option@underrightarrow@`, 33, 47
`\ifovar@option@underrightharpoondown@`, 33, 48
`\ifovar@option@underrightharpoonup@`, 33, 48
`\ovar@after@arrow`, 36, 37
`\ovar@arrow@end`, 38
`\ovar@arrow@fill`, 38, 42
`\ovar@arrow@middle`, 38
`\ovar@arrow@start`, 38
`\ovar@before@arrow`, 36, 37
`\ovar@extralength`, 42, 47
`\ovar@leftarrow`, 35, 47, 48
`\ovar@length@min`, 36
`\ovar@macro@arrow`, 36, 45, 46
`\ovar@macro@arrowfill`, 37, 38
`\ovar@macro@stack`, 36, 45, 46
`\ovar@picture@command`, 41
`\ovar@picture@geometry`, 41
`\ovar@picture@linethickness`, 41
`\ovar@rightarrow`, 35, 39, 47, 48
`\ovar@row@teststyle`, 50
`\ovar@rulethickness@family`, 43
`\ovar@rulethickness@fontdimen`, 43
`\ovar@set`, 36–38, 40, 41, 45
`\ovar@set@`, 36, 45
`\ovar@set@arrowlength`, 42, 45
`\ovar@set@arrowthickness`, 43, 45
`\ovar@set@arrowthickness@UM@lua`, 43
`\ovar@set@common`, 45
`\ovar@shift@left`, 37, 38, 40, 41
`\ovar@shift@right`, 37, 38, 40, 41
`\ovar@stack@fill`, 37, 44
`\ovar@stack@lens`, 37, 45
`\ovar@stackover@`, 44, 45
`\ovar@stackover@@`, 44
`\ovar@stackover@fill`, 44
`\ovar@stackover@lens`, 45
`\ovar@stackunder@`, 44, 45
`\ovar@stackunder@@`, 44
`\ovar@stackunder@fill`, 37, 44
`\ovar@stackunder@lens`, 37, 45
`\ovar@starversion`, 46, 47
`\ovar@subcmd`, 35, 46
`\ovar@tempdim`, 42, 43, 47
`\ovar@testkerning`, 50, 51
`\ovar@testmathstyles`, 50, 51
`\ovar@tikz@command`, 40
`\ovar@tikz@options`, 40
`\ovar@tikz@path`, 40
`\ovar@tikz@pathoptions`, 40

- `\ovar@trim@end`, 38
- `\ovar@trim@middle`, 38
- `\ovar@trim@start`, 38

Keys

- add path options, 27
- add tikz options, 27
- after arrow, 24
- amsmath, 26
- arrow, 28
- arrow macro, 30
- arrow under, 23
- arrows, 27
- before arrow, 24
- center arrow, 24
- detect subscripts, 24
- end, 25
- esvect, 26
- fill macro, 31
- geometry, 28, 29
- left arrow, 24
- line thickness, 27–29
- middle, 25
- middle config, 26
- min length, 22
- no arrow macro hook, 31
- no stack macro hook, 30
- no trimming, 25
- path, 27
- path options, 27
- picture command, 29
- psset, 28
- pstricks command, 28
- right arrow, 24
- shift left, 23
- shift leftright, 24
- shift right, 23
- space after arrow, 24
- space before arrow, 24
- stack macro, 30
- start, 25
- thinner, 27–29
- tikz command, 27
- tikz options, 27
- trim, 25
- trim end, 25
- trim middle, 25
- trim start, 25

left arrow key, 24

Lengths

- `\overarrowlength`, 19, 40–43
- `\overarrowsmallerthickness`, 20, 40–43
- `\overarrowthickness`, 19, 40–43
- line thickness key, 27–29
- middle key, 25
- middle config key, 26
- min length key, 22

`\NewOverArrowCommand`, 17

- no arrow macro hook key, 31
- no stack macro hook key, 30
- no trimming key, 25
- noesvect package option, 14

old-arrows package option, 16

`\overarrowlength` length, 19

`\overarrowsmallerthickness` length, 20

`\overarrowthickness` length, 19

`\overbar`, 21

overbar package option, 15

overcommands package option, 15

`\overleftarrow`, 21

overleftarrow package option, 15

`\overleftharpoondown`, 21

overleftharpoondown package option, 15

`\overleftharpoonup`, 21

overleftharpoonup package option, 15

`\overletrightarrow`, 21

overletrightarrow package option, 15

`\overrightarrow`, 21

overrightarrow package option, 15

`\overrightarrowharpoondown`, 21

overrightarrowharpoondown package option, 15

`\overrightarrowharpoonup`, 21

overrightarrowharpoonup package option, 15

path key, 27

path options key, 27

picture command key, 29

`\ProvideOverArrowCommand`, 17

psset key, 28

pstarrows package option, 17

pstricks package option, 16

pstricks command key, 28

`\RenewOverArrowCommand`, 17

right arrow key, 24

`\SetOverArrowsMethod`, 30

`\SetOverArrowsMethod*`, 30

`\SetOverArrowsSubscriptCommand`, 30

shift left key, 23

shift leftright key, 24

- `shift right` key, 23
- `\smallermathstyle`, 19
- `space after arrow` key, 24
- `space before arrow` key, 24
- `stack macro` key, 30
- `start` key, 25
- `subactive` package option, 17
- `subother` package option, 17
- `subscripts` package option, 17

- `\TestOverArrow`, 18
- `\TestOverArrow*`, 18
- `thinner` key, 27–29
- `tikz` package option, 16
- `tikz command` key, 27
- `tikz options` key, 27
- `trim` key, 25
- `trim end` key, 25
- `trim middle` key, 25
- `trim start` key, 25

- `\underbar`, 22
- `underbar` package option, 16
- `undercommands` package option, 15
- `\underleftarrow`, 22
- `underleftarrow` package option, 15
- `\underleftharpoondown`, 22
- `underleftharpoondown` package option, 16
- `\underleftharpoonup`, 22
- `underleftharpoonup` package option, 16
- `\underletrightarrow`, 22
- `underletrightarrow` package option, 16
- `\underrightarrow`, 21
- `underrightarrow` package option, 15
- `\underrightharpoondown`, 22
- `underrightharpoondown` package option, 16
- `\underrightharpoonup`, 22
- `underrightharpoonup` package option, 16

- `\vv`, 20
- `\vv*`, 20

- `\xjoinrel`, 19

Change History

v1.0		with unicode-math	43
General: Initial version	1	Remove useless	
v1.0.1		<code>\AtBeginDocument</code>	35
General: Bug fix for under* options	33	Use <code>\def</code> instead of <code>\let</code> for	
v1.1		<code>\ovar@rightarrow</code> and	
General: Support for non-standard		<code>\ovar@leftarrow</code>	35
subscripts	35, 46, 50	Use <code>\harrowextender</code> , if	
v1.2		available.	39
General: Add key <code>??^P??</code>	39	Use <code>??^P??</code> instead of <code>??^P??</code>	47
Declare Declare predefined		v1.4	
commands after <code>unicode-math</code>		General: Add option <code>pstricks</code>	33
settings.	47	Add the method <code>pstricks</code>	40
Fix esvect font sizes	35	Allow backslash in command	
Get the correct rule thickness		name	45