

The `overarrows` package*

Julien Labbé

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

February 2, 2025

Abstract

A L^AT_EX 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*}
```



Predefined commands are also provided:

- to typeset vectors:

\vec{v} \overrightarrow{AB} ,

- to draw arrows of various shapes above math expressions:

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

- to draw arrows of various shapes under math expressions:

\underline{AB} \underline{AB} \underline{AB} \underline{AB} \underline{AB} \underline{AB} \underline{AB} \underline{AB} .

*This document corresponds to `overarrows` v1.3, dated 2025/02/02.

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	19
4.2.3	Useful lengths for TikZ, PSTricks or <code>picture</code> environment	19
4.2.4	Vectors macros	20
4.2.5	Predefined commands	21
4.3	Keys	22
4.3.1	Arrow position and length settings	22
4.3.2	Subscripts detection setting	25
4.3.3	Symbols assemblage settings	25
4.3.4	TikZ settings	27
4.3.5	PSTricks settings	28
4.3.6	Picture environment settings	29
4.4	Advanced commands and keys	30
4.4.1	Advanced commands	30
4.4.2	Advanced keys	30
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	33
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 $\overrightarrow{v}_{\vec{u}}$.

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:

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

and for arrow under :

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

2 Introduction

2.1 Vector arrows

Vectors are commonly typeset in bold face, or with an arrow above³. For this second convention, $\text{\TeX}/\text{\LaTeX}$ provides the command `\vec`, which accents its content (using the `\mathaccent` command) with the character $\vec{}$ (`\mathchar"017E` in Computer Modern font). But $\vec{}$ isn't extensible⁴, and gives: \vec{v} , \vec{AB} or $\vec{\text{grad}}$ (there's no command `\widevec` analogous to `\widehat`).

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

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

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

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

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

2.2 Stack and arrow macros

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

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

Three macros are used here:

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

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

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

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

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

2.3 Extensible arrows

Arrows drawn by the commands `\overrightarrow` or `\vv` are built by joining math symbols, and made extensible by repetition of the central symbol⁵. Thus, the line of the macro `\overrightarrow` is made by repetition of command `\relbar` — (which simply corresponds to the minus sign), while `\vv` use its own command `\relbareda` — .

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

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

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

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

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

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

⁵Using the \TeX `\cleaders` command.

3 Quick start

3.1 Loading the package `overarrows`

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

```
\usepackage{overarrows}
```

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

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

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

- `\overrightarrow`^{→P.21}
- `\overleftarrow`^{→P.21}
- `\overleftrightarrow`^{→P.21}
- `\overrightarrowharpoonup`^{→P.21}
- `\overrightarrowharpoondown`^{→P.21}
- `\overleftarrowharpoonup`^{→P.21}
- `\overleftarrowharpoondown`^{→P.21}
- `\overbar`^{→P.21}
- `\underrightarrow`^{→P.22}
- `\underleftarrow`^{→P.22}
- `\underleftrightarrow`^{→P.22}
- `\underrightharpoonup`^{→P.22}
- `\underrightharpoondown`^{→P.22}
- `\underleftarrowharpoonup`^{→P.22}
- `\underleftarrowharpoondown`^{→P.22}
- `\underbar`^{→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 and the arrow configuration as comma-separated list of key-values. By default, a right arrow is set:

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

$$\overrightarrow{test}$$

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

```
$ v_{sub} \quad \myovercmd{v}_{sub} \quad \myovercmd*{v}_{sub} $
```

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

3.3 Start and end of the arrow

Extremities of the arrow are set by the keys `start`^{→P.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 \LaTeX . This can be done with the package `amssymb`, by adding in preamble:

```
\usepackage{amssymb}
```

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



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

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

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

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

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

If the math font differs from the default *Computer Modern*, the central part of the arrow may have inappropriate position or line width. This is because the default symbol used for the arrow line is `\relbareda`  from the `esvect` package⁶. If needed, try to set the `middle`^{P.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.26} 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:

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

<code>\TestOverArrow{\overhooktwoheadrightarrow}</code>			
<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` \Rightarrow and `\Rightarrow` \Rightarrow , gives:

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

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

The key `trim`^{→P.26} sets `trim start`^{→P.25}, `trim middle`^{→P.25} and `trim end`^{→P.26} with the same value.

The previous arrow is visually too big. The macro `\smallermathstyle`^{→P.19} allows to obtain a better result:

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

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

Note that `\smallermathstyle`^{→P.19} should not be used for `end`^{→P.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} $
```

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

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



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} $
```



3.5 Symbols assemblage

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

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

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



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

⁷For example, the symbol `\models` \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{=}`).

```

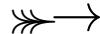
\NewOverArrowCommand{\overttailrightarrow}{%
  start={\ttail},
  end={\rightarrow},
  trim start=12,
  shift left=0, shift right=0,
  space after arrow=.2ex,
  min length=24,
}
$ \overttailrightarrow{v} \quad \overttailrightarrow{AB} $

```



Here the `min length`^{P.23} 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\ttail}\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{$\smallermathstyle\ttail$}}},
  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` package⁸:

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

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

- `\overarrowlength`^{→P.20} for the arrow length,
- `\overarrowthickness`^{→P.20} 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} $
```

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

Keys to use Tikz are described in section 4.3.4, page 27. 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.28}

Here is a example of an arrow drawn with TikZ⁹:

```
\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} $
```

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

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

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

3.7 Drawing the arrow with PSTricks

In addition to PGF/TikZ, the arrow can be drawn with PSTricks macros. For this, the optional argument `pstricks` must be passed to `\NewOverArrowCommand`^{→P.17}. Like with `tikz`, the three lengths `\overarrowlength`^{→P.20}, `\overarrowthickness`^{→P.20} 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} $
```

\vec{v} \overrightarrow{AB}

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

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

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

\vec{v} \overrightarrow{AB}

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

\vec{v} \overrightarrow{AB}

3.8 Drawing the arrow with L^AT_EX picture environment

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

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

\vec{v} \overrightarrow{AB}

If `overarrows` is loaded with the option `pstarrows`^{→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} $
```



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} $
```



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 $\backslash esvectvv$ ^{P.20}. The `esvect` font description is fixed to allow any font sizes.

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

The `esvect` package also provides the right arrow command `\fldr`. The shape of the arrow depends on the option passed to the `esvect` package:  (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 \rightarrow . `\vv` command gives : \vec{v} \overrightarrow{AB} $\overrightarrow{\text{grad}}$.

esvectb

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

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

esvectc

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

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

esvectd

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

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

esvecte

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

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

esvectf

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

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

esvectg

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

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

esvecth

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

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

4.1.2 Predefined commands

The `overarrows` package provides sixteen predefined commands, eight with the arrow over, and eight with the arrow under. By default, 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`, `\overleftrightharrow`, `\underleftarrow`, `\underrightharrow` and `\underleftrightharrow` 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 $\overrightarrow{}$ command: \vec{v} , \overrightarrow{AB} , $\overrightarrow{\text{grad}}$.

overleftarrow

Defines the $\overleftarrow{}$ command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

overleftrightarrow

Defines the $\overleftrightarrow{}$ command: \overleftrightarrow{v} , \overleftrightarrow{AB} , $\overleftrightarrow{\text{grad}}$.

overrightarrow

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

overrightarrow

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

overleftarrow

Defines the $\overleftarrow{}$ command: \overleftarrow{v} , \overleftarrow{AB} , $\overleftarrow{\text{grad}}$.

overleftrightarrow

Defines the $\overleftrightarrow{}$ command: \overleftrightarrow{v} , \overleftrightarrow{AB} , $\overleftrightarrow{\text{grad}}$.

overbar

Defines the $\overline{}$ command: \bar{v} , \overline{AB} , $\overline{\text{grad}}$.

Under arrows

underrightarrow

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

underleftarrow

Defines the $\underleftarrow{}$ command: $\underline{\overleftarrow{v}}$, $\underline{\overleftarrow{AB}}$, $\underline{\overleftarrow{\text{grad}}}$.

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` \leftarrow and `\varrightarrow` \rightarrow , 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}, `\underrightharpoonup`^{→P.22}, `\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 with \pdf\LaTeX (see the PSTricks documentation). PSTricks arrows, drawn with the `pstricks` method, are always available, even if this option is not set, provided the `pstricks` package is loaded independently.

pstarrows

Loads the `pict2e` package, with its option `pstarrows`. Vectors using \LaTeX `picture` environment gives then \overrightarrow{AB} instead of \vec{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.25} 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

New: v1.1 2023/02/15

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

subactive

New: v1.1 2023/02/15

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

debug

Writes the meaning of defined commands in \LaTeX log.

4.2 Commands

4.2.1 Macro for commands creation

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

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

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

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

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

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

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

Updated: v1.2 2024/07/11

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

- `symbol` to draw the arrow by symbols assemblage (default);
- `tikz` to draw the arrow with PGF/TikZ;
- `pstricks` to draw the arrow with PSTricks;
- `picture` to draw the arrow with the L^AT_EX `picture` environment.

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

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

```

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

```

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

```

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

```

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

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

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

```

\TestOverArrow* [my-pattern] {vv}

```

Test of \vv and \vv* macros

\vv for different math styles

<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

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

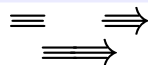
4.2.2 Useful macros for symbols assemblage

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 *number* math units (3.5 mu by default). Must be used in math mode. Useful to assemble math symbols and create new ones.

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

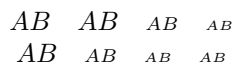


`\smallermathstyle`

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

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

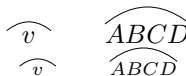
```
\displaystyle AB \quad \textstyle AB
\quad \scriptstyle AB \quad \scriptscriptstyle AB \par
\displaystyle AB \quad \smallermathstyle AB
\quad \smallermathstyle AB \quad \smallermathstyle AB
```



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}
```



`\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.23}.

`\overarrowthickness`

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

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

Updated: v1.2 2024/07/11

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

`\overarrowsmallerthickness`

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

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

Updated: v1.2 2024/07/11

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

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

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

`\esvectvv`

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

```

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

```

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

4.2.5 Predefined commands

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

Over arrows

`\overrightarrow`

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

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`

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

`\overrightarrow`

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

`\overleftrightarrow`

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

`\overleftrightarrow`

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

`\overbar`

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

Under arrows

`\underrightarrow`

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

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

`\underleftarrow`

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

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

`\underleftrightharpoonrightarrow`

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

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

`\underrightharpoonup`

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

`\underrightharpoondown`

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

`\underleftharpoonup`

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

`\underleftharpoondown`

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

`\underbar`

$$\underbar v \quad \underbar AB \quad \underbar grad$$

4.3 Keys

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

4.3.1 Arrow position and length settings

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

Length

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

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

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

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

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

\overrightarrow{v} \overrightarrow{ABCDEF}

Placement

arrow under (default autoconfig, initially unset)

arrow under=autoconfig|noconfig

Places the arrow under, instead of over.

arrow under or arrow under=autoconfig also configures suitably the key **detect subscripts**^{→P.25} 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} $
```

\underhook{v} \underhook{AB}

Horizontal shifts

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

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

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

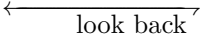
Shifts the right side of the arrow by $\langle number \rangle$ math units (positive number means a shift to the left).

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

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

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

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



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

Sets `shift left`^{P.23} and `shift right`^{P.23} to the same $\langle number \rangle$ 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 $\langle number \rangle$.

`right arrow` (default 2)

Sets `shift right`^{P.23} to zero and `shift left`^{P.23} to $\langle number \rangle$.

Vertical adjunct

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

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

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

Over and under arrow commands are typeset through the \TeX `\align` command, which aligns contents, like a tabular. The $\langle vertical material \rangle$ 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`={ $\langle length \rangle$ } (no default)

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

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

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

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



4.3.2 Subscripts detection setting

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

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

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

By default, the initial value of `detect subscripts` is `false`. When the option `subscripts→P.17` is set, the initial value of `detect subscripts` is `true`.

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

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

$$i_0 \quad \vec{i}_0 \quad \overrightarrow{i}_0 \quad \overrightarrow{i}_0$$

4.3.3 Symbols assemblage settings

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

start={command} (no default, initially `\relbar`)
middle={command} (no default, initially set by `middle config=auto`)
end={command} (no default, see below for the initial value)

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

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

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

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

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

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

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

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

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

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

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


Sets **trim start**^{→P.25}, **trim middle**^{→P.25} and **trim end** to the same $\langle number \rangle$ value.


no trimming

Clears **trim start**^{→P.25}, **trim middle**^{→P.25} and **trim end**.

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

Sets a suitable configuration for the keys **middle**^{→P.25} and **trim middle**^{→P.25}:

For **middle config** = relbar, **middle**^{→P.25} is set to `\relbar`  and **trim middle**^{→P.25} 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 font¹⁰, **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={\relbaredd}</code>	<code>middle={\relbareda}</code>	<code>end={\fldr}</code>
<code>trim start=1.5</code>	<code>trim middle=0</code>	<code>trim end=1.5</code>
<code>space before arrow=-.7pt</code>	<code>space after arrow=-.3pt</code>	<code>right arrow=2</code>

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

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

4.3.4 TikZ settings

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


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

where `tikz options` and `tikz command`^{P.28} are two keys described below. When `tikz command` is let unset, the drawing command turns into:

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

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

```
\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},
}
```

$$\$ \overdotteddoublearrow{v} \quad \backslash quad \quad \overdotteddoublearrow{AB} \$$$


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

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

Sets TikZ options to *<TikZ options>*.

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

Sets TikZ path options to *<path options>*.

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

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

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

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

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

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

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

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

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

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

thinner

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

tikz command={*TikZ command*} (initially unset)

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

4.3.5 PSTricks settings

New: v1.2 2024/07/11

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

```

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

```

where `geometry`, `line thickness`^{P.29}, `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} \vec{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 $\langle length \rangle$.

`thinner`

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

4.3.6 Picture environment settings

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

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

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

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



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

`picture command={⟨picture command⟩}`
(no default, initially `\put(0,0){\vector(1,0){\overarrowlength}}`)

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

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

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

`line thickness={⟨length⟩}` (no default)

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

`thinner` (no default)

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

4.4 Advanced commands and keys

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

4.4.1 Advanced commands

`\SetOverArrowsSubscriptCommand{⟨command⟩}`

New: v1.1 2023/02/15

Sets to `⟨command⟩` the command used for subscript detection, when this is enabled by the key `detect_subscripts`^{→P.25} (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`^{→P.31} 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`^{→P.31}, 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 `symp` method.

`lens` if `arrow macro` creates fixed-length arrows, and needs the computation of lengths `\overarrowlength`^{→P.20}, `\overarrowthickness`^{→P.20} 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`^{→P.31}.

4.4.2 Advanced keys

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

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

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

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

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

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

code must configure **stack macro**^{→P.30} accordingly to the user keys setting.

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

Sets the *code* executed if **arrow macro**^{→P.30} is left unset, after user evaluation of *keys* in `\NewOverArrowCommand`^{→P.17}, `\RenewOverArrowCommand`^{→P.17}, `\ProvideOverArrowCommand`^{→P.17} or `\DeclareOverArrowCommand`^{→P.17}.

code must configure **arrow macro**^{→P.30} accordingly to the user keys setting.

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

Defines the fill macro to be *definition*. The fill macro is used by arrows created with the **ymb** 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. *definition* can be, for example, the macro `\arrowfill@` used by `amsmath \overrightarrow`.

5 Complements

5.1 Know issues

5.1.1 Math font change

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

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

5.1.2 Detection of non standard subscripts

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

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

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

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

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

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

Alternatively, two package options handle the cases where the catcode of the underscore “_” symbol is changed: `subother`^{→P.17} (for catcode 12, or *other*) and `subactive`^{→P.17} (for catcode 13, or *active*). Hence, setting the `subother`^{→P.17} option is sufficient for compatibility with the `altnsubsup` package (no need of `\SetOverArrowsSubscriptCommand`^{→P.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

- v1.3 Bug fix for `esvect` options (see <https://github.com/julienlabbe/latex-packages/issues/2>).
- v1.2
- Fix compatibility issues with `unicode-math`.
 - Allow to draw the arrow with `PSTricks`.
 - Make `esvect` handle all font sizes.
 - Allow backslash in command name for `\NewOverArrowCommand` and variants.
 - Rewrite starred variant for better performances.
- v1.1 Support for non-standard subscripts.
- v1.0.1 Bug fix for `under*` options.
- v1.0 Initial version.

6 Implementation

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

Management of options

Declaration of conditionals

```
2 \newif\ifovar@option@oldarrows@
3 \newif\ifovar@option@tikz@
4 \newif\ifovar@option@pstricks@
5 \newif\ifovar@option@pstarrows@
6 \newif\ifovar@detectsubscripts@
7 \newif\ifovar@option@subother@
8 \newif\ifovar@option@subactive@
9 \newif\ifovar@option@debug@
```

Following conditionals are for predefined commands.

```
10 \newif\ifovar@option@overrightarrow@
11 \newif\ifovar@option@underrightarrow@
12 \newif\ifovar@option@overleftarrow@
13 \newif\ifovar@option@underleftarrow@
14 \newif\ifovar@option@overleftrightharpoon@
15 \newif\ifovar@option@underleftrightharpoon@
16 \newif\ifovar@option@overrightarrowharpoonup@
17 \newif\ifovar@option@underrightharpoonup@
18 \newif\ifovar@option@overrightarrowharpoondown@
19 \newif\ifovar@option@underrightharpoondown@
20 \newif\ifovar@option@overleftarrowharpoonup@
21 \newif\ifovar@option@underleftarrowharpoonup@
22 \newif\ifovar@option@overleftarrowharpoondown@
23 \newif\ifovar@option@underleftarrowharpoondown@
24 \newif\ifovar@option@overbar@
25 \newif\ifovar@option@underbar@
```

Declaration of options

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

Following options are for predefined commands.

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

Following options are for sets of predefined commands.

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

```

82 \ovar@option@overrightarrow@true
83 \ovar@option@underrightarrow@true
84 \ovar@option@overleftarrow@true
85 \ovar@option@underleftarrow@true
86 \ovar@option@overleftrightharpoon@true
87 \ovar@option@underleftrightharpoon@true
88 \ovar@option@overrightharpoonup@true
89 \ovar@option@underrightharpoonup@true
90 \ovar@option@overrightharpoondown@true
91 \ovar@option@underrightharpoondown@true
92 \ovar@option@overlefttharpoonup@true
93 \ovar@option@underlefttharpoonup@true
94 \ovar@option@overlefttharpoondown@true
95 \ovar@option@underlefttharpoondown@true
96 \ovar@option@overbar@true
97 \ovar@option@underbar@true
98 }

```

Options processing

```

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

```

Package dependencies

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

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

```

101 \RequirePackage{amsmath}

```

Option `old-arrows`^{→P. 16}. Configuration of arrows used for predefined commands.

```

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

```

Option `esvect`^{→P. 13}.

```

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

```

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

```

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

```

Option `tikz`^{→P. 16}.

```

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

```

Option `pstricks` ^{→ P. 16.}

```

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

```

Option `pstarrows` ^{→ P. 17.}

```

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

```

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

```

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

```

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

```

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

```

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

```

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

```

Configuration of subscripts detection

`\SetOverArrowsSubscriptCommand`

Sets the subscript command.

```

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

```

Initial configuration.

```

156 \SetOverArrowsSubscriptCommand{_{}}

```

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

```

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

```

```
161 \endgroup
162 \fi
```

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

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

Management of keys

Family declaration and setters

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

Common keys

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

`detect subscripts`^{→P.25}.

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

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

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

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

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

`min length`^{→P.23}.

```

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

```

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

```

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

```

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

```

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

```

arrow under^{→P.23}.

```

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

```

Keys for the symb method

```

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

```

Fill macro.

```

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

```

Arrow macro.

```

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

```

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

```

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

```

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

```

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

```

middle config^{→P.26}.

```

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

```

```

302     middle={\relbareda},
303     trim middle={1},
304   }
305 }
306 },

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

```

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

```

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

```

`amsmath` → P.26 .

```

319 amsmath/.is choice,%
320 amsmath/mimic/.code=\pgfkeysalso{%
321   start={\relbar}, middle={\relbar}, end={\rightarrow},
322   trim start=7,
323   trim middle=2,
324   trim end=7,
325   shift leftright=0,
326   after arrow={}, before arrow={},
327 },
328 amsmath/strict/.code=\pgfkeysalso{%
329   amsmath=mimic,
330   no trimming,
331   fill macro={\arrowfill0}, stack macro={\overarrow0},
332 },
333 amsmath/.default=mimic,

```

`esvect` → P.26 .

```

334 esvect/.is choice,%
335 esvect/mimic/.code=\pgfkeysalso{%
336   start={\relbaredd}, middle={\relbareda}, end={\fldr},
337   trim start=1.5,
338   trim end=1.5,
339   trim middle=0,
340   right arrow=2,
341   space before arrow=-.7pt,
342   space after arrow=-.3pt,
343 },
344 esvect/strict/.code=\pgfkeysalso{%
345   esvect=mimic,
346   no trimming,
347   fill macro={\traitfill0}, stack macro={\overvect0},
348 },
349 esvect/.default=mimic,

```

Initial configuration.

```

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

```


Keys for the tikz method

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

Arrow macro.

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

TikZ parts: `tikz command`^{→P.28}, `tikz options`^{→P.27}, `path options`^{→P.27}, `path`^{→P.27}.

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

TikZ handy keys: `add path options`^{→P.27}, `add tikz options`^{→P.27}, `arrows`^{→P.27}, `line thickness`^{→P.27}, `thinner`^{→P.28}.

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

Initial configuration.

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

Keys for the pstricks method

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

Arrow macro.

```

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

```

Pstricks parts: pstricks command^{→P.28}, psset^{→P.28}, geometry^{→P.28}, line thickness^{→P.29}.

```

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

```

Pstricks handy key: arrow^{→P.28}, thinner^{→P.29}.

```

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

```

Initial configuration.

```

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

```

Keys for the picture method

```

428 \SetOverArrowsMethod[lens]{picture}{%

```

Arrow macro.

```

429 no arrow macro hook/.code={%
430   \ovar@set{%
431     arrow macro/.expanded={%
432       $\noexpand\mkern \expandonce{\ovar@shift@left} mu\noexpand\relax$%
433       \noexpand\begin{picture}\expandonce{\ovar@picture@geometry}%
434         \noexpand\linethickness{\expandonce{\ovar@picture@linethickness}}%
435         \expandonce{\ovar@picture@command}%
436       \noexpand\end{picture}%
437       $\noexpand\mkern \expandonce{\ovar@shift@right} mu\noexpand\relax$%
438     }
439   }
440 },

```

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

```

441 picture command/.store in=\ovar@picture@command,
442 geometry/.store in=\ovar@picture@geometry,
443 line thickness/.store in=\ovar@picture@linethickness,

```

```

444 picture command/.value required,
445 geometry/.value required,
446 line thickness/.value required,

```

Picture handy key: `thinner` → P. 29.

```

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

```

Initial configuration.

```

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

```

Commands

Macros for symbols assemblage

```

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

```

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

```

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

```

```

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

```

`\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

```

463 \def\ovar@arrow@fill#1#2#3#4#5#6{%
464   $\m@th\thickmuskip0mu\medmuskip\thickmuskip\thinmuskip\thickmuskip\relax%
465   \mkern #1 mu\relax#6#3%
466   \cleaders\hbox{#6#4$}\hfill%
467   #5\mkern #2 mu\relax$%
468 }

```

Macros for fixed length arrows

Lengths declaration.

```

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

```

`\ovar@set@arrowlength`

Sets `\overarrowlength` → P. 20.

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

```

473 \def\ovar@set@arrowlength#1#2#3{%
474   \settowidth{\ovar@tempdim}{$\m@th#2\mskip #1 mu\relax$}%
475   \settowidth{\overarrowlength}{$\m@th#2#3$}%
476   \ifdim \overarrowlength < \ovar@tempdim \overarrowlength=\ovar@tempdim\fi%
477 }

```

`\ovar@set@arrowthickness`

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

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

#1: math style

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

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

```

478 \def\ovar@rulethickness@fontdimen{8}
479 \def\ovar@rulethickness@family{3}
480 \def\ovar@set@arrowthickness#1{%
481   \ifx#1\displaystyle%
482     \overarrowthickness =
483     \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
484     \overarrowsmallerthickness =
485     \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
486   \else\ifx#1\textstyle%
487     \overarrowthickness =
488     \fontdimen \ovar@rulethickness@fontdimen \textfont \ovar@rulethickness@family%
489     \overarrowsmallerthickness =
490     \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
491   \else\ifx#1\scriptstyle%
492     \overarrowthickness =
493     \fontdimen \ovar@rulethickness@fontdimen \scriptfont \ovar@rulethickness@family%
494     \overarrowsmallerthickness =
495     \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
496   \else%
497     \overarrowthickness =
498     \fontdimen \ovar@rulethickness@fontdimen \scriptscriptfont \ovar@rulethickness@family%
499     \overarrowsmallerthickness = \overarrowthickness%
500   \fi\fi\fi%
501 }

```

`unicode-math` with LuaTeX version.

```

502 \def\ovar@set@arrowthickness@UM@lua#1{%
503   \overarrowthickness = \Umathoverbarrule #1
504   \ifx#1\displaystyle%
505     \overarrowsmallerthickness = \Umathoverbarrule \textstyle%
506   \else\ifx#1\textstyle%
507     \overarrowsmallerthickness = \Umathoverbarrule \scriptstyle%
508   \else%
509     \overarrowsmallerthickness = \Umathoverbarrule \scriptscriptstyle%
510   \fi\fi%
511 }

```

Test which version to use.

```

512 \AddToHook{begindocument}[overarrows]
513 {%
514   \@ifpackageloaded{unicode-math-luatex}
515   {%
516     \global\let\ovar@set@arrowthickness\ovar@set@arrowthickness@UM@lua
517   }
518   {%
519     \@ifpackageloaded{unicode-math-xetex}

```

```

520     {%
521       \gdef\ovar@rulethickness@fontdimen{54}
522       \gdef\ovar@rulethickness@family{2}
523     }
524     {}
525   }
526 }

```

Stack macros

`\ovar@stackover@@`

Bases of all stack macros.

`\ovar@stackunder@@`

#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

```

527 \def\ovar@stackover@@#1#2#3#4#5#6{\vbox{\ialign{##\crrc%
528   $%5\mskip #1 mu\relax$\crrc%
529   \noalign{#2\nointerlineskip}#4\crrc%
530   \noalign{#3\nointerlineskip}%
531   $%m@th\hfil#5#6\hfil$\crrc%
532   }%
533 }%
534 }
535 \def\ovar@stackunder@@#1#2#3#4#5#6{\vtop{\ialign{##\crrc%
536   $%m@th\hfil#5#6\hfil$\crrc%
537   \noalign{#2\nointerlineskip}#4\crrc%
538   \noalign{#3\nointerlineskip}%
539   $%5\mskip #1 mu\relax$\crrc%
540   }%
541 }%
542 }

```

`\ovar@stackover@`

Stack macros without min arrow length.

`\ovar@stackunder@`

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

```

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

```

`\ovar@stackover@fill`

Stack macros for extensible arrows.

`\ovar@stackunder@fill`

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

`\ovar@stack@fill`

```

545 \def\ovar@stackover@fill#1#2#3#4#5#6{\ovar@stackover@@{#1}{#2}{#3}{#4#5}{#5}{#6}}
546 \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}.

```

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

```

```

\ovar@stackover@lens      Stack macros for fixed-length arrows (these call \ovar@set@arrowlength and
\ovar@stackunder@lens    \ovar@set@arrowthickness).
\ovar@stack@lens         #1: min length, in math units
                        #2: vertical mode material before arrow
                        #3: vertical mode material after arrow
                        #4: arrow content macro
                        #5: math style
                        #6: content
548 \def\ovar@stackover@lens#1#2#3#4#5#6{%
549   \ovar@set@arrowlength{#1}{#5}{#6}%
550   \ovar@set@arrowthickness{#5}%
551   \ovar@stackover@{#2}{#3}{#4}{#5}{#6}%
552 }
553 \def\ovar@stackunder@lens#1#2#3#4#5#6{%
554   \ovar@set@arrowlength{#1}{#5}{#6}%
555   \ovar@set@arrowthickness{#5}%
556   \ovar@stackunder@{#2}{#3}{#4}{#5}{#6}%
557 }

\ovar@stack@lens matches the macro \ovar@stackover@lens by default, or
\ovar@stackunder@lens with arrow under→P. 23.
558 \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.

```

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

\DeclareOverArrowCommand
568 \NewDocumentCommand{\DeclareOverArrowCommand}{
569   0{symb} >{\ovar@cmdname@processor} m m
570 }{%
571   \begingroup
572   \ovar@set@common
573   \ifcsdef\ovar@set@#1{%
574     \csuse\ovar@set@#1
575   }{%
576     \PackageError{overarrows}{Unknown method #1}
577     {Try with 'symb', 'tikz', 'pstriks' or 'picture'}
578   }
579   \ovar@set@#3
580   \ifdef\ovar@macro@arrow{}{%
581     \ovar@set{no arrow macro hook}
582   }
583   \ifdef\ovar@macro@stack{}{%
584     \ovar@set{no stack macro hook}
585   }

```

```

586 \csxdef{ovar#2@normal}{%
587 \noexpand\mathpalette{%
588 \expandonce{\ovar@macro@stack}\expandonce{\ovar@macro@arrow}}%
589 }
590 }
591 \csxdef{ovar#2@starred}{%
592 \noexpand\mathpalette{%
593 \noexpand\ovar@starversion{%
594 \expandonce{\ovar@macro@stack}\expandonce{\ovar@macro@arrow}}%
595 }
596 }
597 }
598 \ifovar@detectsubscripts%
599 \csgdef{ovar#2@auto}##1{%
600 \ifnextchar \ovar@subcmd {%
601 \csuse{ovar#2@starred}{##1}%
602 }{%
603 \csuse{ovar#2@normal}{##1}%
604 }%
605 }
606 \csgdef{#2}{%
607 \@ifstar{\csuse{ovar#2@starred}}{\csuse{ovar#2@auto}}%
608 }
609 \else
610 \csgdef{#2}{%
611 \@ifstar{\csuse{ovar#2@starred}}{\csuse{ovar#2@normal}}%
612 }
613 \fi
614 \ifovar@option@debug@
615 \PackageInfo{overarrows}{%
616 Meaning of \protect\ovar#2@normal\MessageBreak
617 used for \@backslashchar#2:\MessageBreak%
618 \expandafter\meaning\csname ovar#2@normal\endcsname}
619 \fi
620 \endgroup
621 }
\ProvideOverArrowCommand
622 \NewDocumentCommand{\ProvideOverArrowCommand}{
623 O{symp} >{\ovar@cmdname@processor} m m
624 }{%
625 \ifcsdef{#2}{%
626 \DeclareOverArrowCommand[#1]{#2}{#3}
627 }
628 }
\NewOverArrowCommand
629 \NewDocumentCommand{\NewOverArrowCommand}{
630 O{symp} >{\ovar@cmdname@processor} m m
631 }{%
632 \ifcsdef{#2}{%
633 \PackageError{overarrows}{Command \csname #2\endcsname already defined}%
634 {You have used \protect\NewOverArrowCommand\space with a command that
635 already has a definition. \MessageBreak%
636 Choose another name, or use instead \protect\DeclareOverArrowCommand.}
637 }{%
638 \DeclareOverArrowCommand[#1]{#2}{#3}
639 }
640 }
\RenewOverArrowCommand
641 \NewDocumentCommand{\RenewOverArrowCommand}{
642 O{symp} >{\ovar@cmdname@processor} m m
643 }{%
644 \ifcsundef{#2}{%
645 \PackageError{overarrows}{Command \csname #2\endcsname undefined}%
646 {You have used \protect\RenewOverArrowCommand\space with a command that was

```

```

647     never defined. \MessageBreak%
648     Check the requested name, or use instead \protect\NewOverArrowCommand.}
649   }{%
650     \DeclareOverArrowCommand[#1]{#2}{#3}
651   }
652 }

```

Starred variant

`\ovar@starversion`

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

```

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

```

Remove the extra space added by the arrow.

```

657   \settowidth{\ovar@tempdim}{\m@th #2{#3}$}%
658   \kern\dimeval{0.5\ovar@tempdim - 0.5\wd\ovar@tempbox}%
659 }

```

`\vv` vector command

`\vv`

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

`\esvectvv`

```

660 \ifdefined\ovar@option@esvect
661   \let\esvectvv\vv
662   \undef\vv
663
664 \NewOverArrowCommand{\vv}{esvect, middle config = relbareda}
665 \fi

```

Predefined commands

Declare predefined commands after unicode-math settings.

```

665 \AddToHook{begindocument}[overarrows]
666 {

```

`\overrightarrow`

```

667   \ifovar@option@overrightarrow@
668     \DeclareOverArrowCommand{\overrightarrow}{%
669       amsmath, middle config=relbar,
670       end=\ovar@rightarrow,
671       right arrow,
672     }
673   \fi

```

`\underrightarrow`

```

674   \ifovar@option@underrightarrow@
675     \DeclareOverArrowCommand{\underrightarrow}{%
676       amsmath, middle config=relbar,
677       end=\ovar@rightarrow,
678       right arrow,
679       arrow under,
680     }
681   \fi

```

`\overleftarrow`


```

682 \ifovar@option@overleftarrow@
683 \DeclareOverArrowCommand{\overleftarrow}{%
684 amsmath, middle config=relbar,
685 start=\ovar@leftarrow,
686 end=\relbar,
687 left arrow,
688 }
689 \fi

\underleftarrow 690 \ifovar@option@underleftarrow@
691 \DeclareOverArrowCommand{\underleftarrow}{%
692 amsmath, middle config=relbar,
693 start=\ovar@leftarrow,
694 end=\relbar,
695 left arrow,
696 arrow under,
697 }
698 \fi

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

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

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

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

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

\underrightharpoondown 738 \ifovar@option@underrightharpoondown@
739 \DeclareOverArrowCommand{\underrightharpoondown}{%

```

```

740     amsmath, middle config=relbar,
741     end=\rightharpoondown,
742     right arrow,
743     arrow under,
744   }
745 \fi

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

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

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

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

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

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

```

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

End of `\AddToHook{begindocument}` hook.

```
797 }
```

Test macros

`\ovar@testmathstyles`

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

```
798 \newcommand{\ovar@testmathstyles}[2][]{
799   \begingroup
800   \newcommand*{\ovar@row@teststyle}[1]{%
801     $\displaystyle ##1$
802     & $\textstyle ##1$
803     & $\scriptstyle ##1$
804     & $\scriptscriptstyle ##1$
805     \\
806   }
807   \renewcommand*\arraystretch{1.5}
808   \begin{tabular*}{0.95\linewidth}{@{\extracolsep{\fill}} cccc}
809     \hline
810     \footnotesize\texttt{\texttt{\textbackslash displaystyle}}
811     & \footnotesize\texttt{\texttt{\textbackslash textstyle}}
812     & \footnotesize\texttt{\texttt{\textbackslash scriptstyle}}
813     & \footnotesize\texttt{\texttt{\textbackslash scriptscriptstyle}}
814     \\
815     \hline
816     \ovar@row@teststyle{\csuse{#2}{v}}
817     \ovar@row@teststyle{\csuse{#2}{AB}}
818     \ovar@row@teststyle{\csuse{#2}{\mathrm{grad}}}
819     \ovar@row@teststyle{\csuse{#2}{\my~long~vector}}
820     \IfValueT{#1}{\ovar@row@teststyle{\csuse{#2}{#1}}}
821     \hline
822   \end{tabular*}
823   \endgroup
824 }
```

`\ovar@testkerning`

```
825 \begingroup
826 \ifovar@option@subother@ \catcode \_ =12 \fi
827 \ifovar@option@subactive@ \catcode \_ =13 \fi
828 \gdef\ovar@testkerning#1{%
829   \begin{displaymath}
830     #1\{t\}_\{#1\{u\}\}_\{#1\{v\}\}
831     \quad
832     #1\{\imath\}_0
833     \quad
834     #1\{v\}
835     = #1\{v\}_x + #1\{v\}_y + #1\{v\}_z
836     = v_x #1\{\imath\} + v_y #1\{\jmath\} + v_z #1\{k\}
837   \end{displaymath}
838 }
839 \endgroup
```

`\TestOverArrow`

```
840 \NewDocumentCommand{\TestOverArrow}{
841   s o >{\ovar@cmdname@processor} m
842 }{%
843   \ifcsdef{#3}{-}{%
844     \PackageWarning{overarrows}{Unknown name '#3' passed to
845       \protect\TestOverArrow}
846   }
847   \IfBooleanTF{#1}{%
848     \noindent\framebox{%
849       \begin{minipage}{0.95\linewidth}
```

```

850     \centering
851     \noindent\textbf{\large%
852         Test of \texttt{\textbackslash#3} and \texttt{\textbackslash#3*} macros}
853     \bigskip\par
854     \textbf{\texttt{\textbackslash#3} for different math styles}
855     \smallskip\par
856     \ovar@testmathstyles[#2]{#3}%
857     \bigskip\par
858     \textbf{\texttt{\textbackslash#3} kerning}
859     \ovar@testkerning{\cuse{#3}}
860     \textbf{\texttt{\textbackslash#3*} kerning}
861     \ovar@testkerning{\cuse{#3}*}
862     \end{minipage}%
863     }\bigskip\par
864     }{%
865     \ovar@testmathstyles[#2]{#3}%
866     }
867 }

```

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
- `overleftharpoondown`, 15
- `overleftharpoonup`, 15
- `overleftrightarrow`, 15
- `overrightarrow`, 15
- `overrightharpoondown`, 15
- `overrightharpoonup`, 15
- `pstarrows`, 17
- `pstricks`, 16
- `subactive`, 17
- `subother`, 17
- `subscripts`, 17
- `tikz`, 16
- `underbar`, 16
- `undercommands`, 15
- `underleftarrow`, 15
- `underleftharpoondown`, 16
- `underleftharpoonup`, 16
- `underleftrightarrow`, 16
- `underrightarrow`, 15
- `underrightharpoondown`, 16
- `underrightharpoonup`, 16

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

- `arrows` key, 27

- `before arrow` key, 24

- `center arrow` key, 24

Commands

- `\DeclareOverArrowCommand`, 17, 46–50
- `\esvectvv`, 20, 48
- `\NewOverArrowCommand`, 17, 47, 48
- `\overbar`, 21, 34, 50
- `\overleftarrow`, 21, 34, 49
- `\overleftharpoondown`, 21, 34, 50
- `\overleftharpoonup`, 21, 34, 50
- `\overleftrightarrow`, 21, 34, 49
- `\overrightarrow`, 21, 34, 48
- `\overrightharpoondown`, 21, 34, 49
- `\overrightharpoonup`, 21, 34, 49
- `\ProvideOverArrowCommand`, 17, 47
- `\RenewOverArrowCommand`, 17, 47
- `\SetOverArrowsMethod`, 30, 37, 38, 41, 42
- `\SetOverArrowsMethod*`, 30
- `\SetOverArrowsSubscriptCommand`, 30, 36, 37
- `\smallermathstyle`, 19, 43
- `\TestOverArrow`, 18, 51
- `\TestOverArrow*`, 18
- `\underbar`, 22, 34, 50
- `\underleftarrow`, 22, 34, 49
- `\underleftharpoondown`, 22, 34, 50
- `\underleftharpoonup`, 22, 34, 50
- `\underleftrightarrow`, 22, 34, 49
- `\underrightarrow`, 22, 34, 48
- `\underrightharpoondown`, 22, 34, 49
- `\underrightharpoonup`, 22, 34, 49
- `\vv`, 20, 48
- `\vv*`, 20
- `\xjoinrel`, 19, 39, 43

- `debug` package option, 17

- `\DeclareOverArrowCommand`, 17

- `detect subscripts` key, 25

- `end` key, 25

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

`fill` macro key, 31

`geometry` key, 28, 29

Internal macros

- `\ifovar@detectsubscripts@`, 33, 47
- `\ifovar@option@debug@`, 33, 47
- `\ifovar@option@oldarrows@`, 33, 35
- `\ifovar@option@overbar@`, 33, 50
- `\ifovar@option@overleftarrow@`, 33, 49
- `\ifovar@option@overleftharpoondown@`, 33, 50
- `\ifovar@option@overleftharpoonup@`, 33, 50
- `\ifovar@option@overleftrightharpoon@`, 33, 49
- `\ifovar@option@overrightarrow@`, 33, 48
- `\ifovar@option@overrightharpoondown@`, 33, 49
- `\ifovar@option@overrightharpoonup@`, 33, 49
- `\ifovar@option@pstarrows@`, 33, 36
- `\ifovar@option@pstricks@`, 33, 56
- `\ifovar@option@subactive@`, 33, 37, 51
- `\ifovar@option@subother@`, 33, 36, 51
- `\ifovar@option@tikz@`, 33, 36
- `\ifovar@option@underbar@`, 33, 50
- `\ifovar@option@underleftarrow@`, 33, 49
- `\ifovar@option@underleftharpoondown@`, 33, 50
- `\ifovar@option@underleftharpoonup@`, 33, 50
- `\ifovar@option@underleftrightharpoon@`, 33, 49
- `\ifovar@option@underrightharpoon@`, 33, 48
- `\ifovar@option@underrightharpoondown@`, 33, 49
- `\ifovar@option@underrightharpoonup@`, 33, 49
- `\ovar@after@arrow`, 37, 38
- `\ovar@arrow@end`, 39
- `\ovar@arrow@fill`, 39, 43
- `\ovar@arrow@middle`, 39
- `\ovar@arrow@start`, 39
- `\ovar@auto@middle`, 36, 40
- `\ovar@auto@trim@middle`, 36, 40
- `\ovar@before@arrow`, 37, 38
- `\ovar@cmdname@processor`, 46, 47, 51
- `\ovar@leftarrow`, 35, 49
- `\ovar@length@min`, 37, 38
- `\ovar@macro@arrow`, 37, 46, 47
- `\ovar@macro@arrowfill`, 38, 39
- `\ovar@macro@stack`, 37, 46, 47
- `\ovar@option@esvect`, 34, 35, 48
- `\ovar@picture@command`, 42
- `\ovar@picture@geometry`, 42
- `\ovar@picture@linethickness`, 42
- `\ovar@pstricks@command`, 42
- `\ovar@pstricks@geometry`, 42
- `\ovar@pstricks@linethickness`, 42
- `\ovar@pstricks@psset`, 42
- `\ovar@rightarrow`, 35, 40, 48, 49
- `\ovar@row@teststyle`, 51
- `\ovar@rulethickness@family`, 44, 45
- `\ovar@rulethickness@fontdimen`, 44, 45
- `\ovar@set`, 37, 39, 41, 42, 46
- `\ovar@set@arrowlength`, 44, 46
- `\ovar@set@arrowthickness`, 44, 46
- `\ovar@set@arrowthickness@UM@lua`, 44
- `\ovar@set@common`, 46
- `\ovar@shift@left`, 38, 39, 41, 42
- `\ovar@shift@right`, 38, 39, 41, 42
- `\ovar@stack@fill`, 38, 45
- `\ovar@stack@lens`, 38, 46
- `\ovar@stackover@`, 45, 46
- `\ovar@stackover@@@`, 45
- `\ovar@stackover@fill`, 45
- `\ovar@stackover@lens`, 46
- `\ovar@stackunder@`, 45, 46
- `\ovar@stackunder@@@`, 45
- `\ovar@stackunder@fill`, 38, 45
- `\ovar@stackunder@lens`, 38, 46

- `\ovar@starversion`, 47, 48
- `\ovar@subcmd`, 36, 47
- `\ovar@tempbox`, 48
- `\ovar@tempdim`, 43, 44, 48
- `\ovar@testkerning`, 51, 52
- `\ovar@testmathstyles`, 51, 52
- `\ovar@tikz@command`, 41
- `\ovar@tikz@options`, 41
- `\ovar@tikz@path`, 41
- `\ovar@tikz@pathoptions`, 41
- `\ovar@trim@end`, 39
- `\ovar@trim@middle`, 39
- `\ovar@trim@start`, 39

Keys

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

- trim, 26
- trim end, 26
- trim middle, 25
- trim start, 25

left arrow key, 24

Lengths

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

line thickness key, 27, 29

middle key, 25

middle config key, 26

min length key, 23

`\NewOverArrowCommand`, 17

no arrow macro hook key, 31

no stack macro hook key, 31

no trimming key, 26

noesvect package option, 14

old-arrows package option, 16

`\overarrowlength` length, 20

`\overarrowsmallerthickness` length, 20

`\overarrowthickness` length, 20

`\overbar`, 21

overbar package option, 15

overcommands package option, 15

`\overleftarrow`, 21

overleftarrow package option, 15

`\overleftharpoondown`, 21

overleftharpoondown package option, 15

`\overleftharpoonup`, 21

overleftharpoonup package option, 15

`\overleftrightarrow`, 21

overleftrightarrow package option, 15

`\overrightarrow`, 21

overrightarrow package option, 15

`\overrightarrowharpoondown`, 21

overrightarrowharpoondown package option, 15

`\overrightarrowharpoonup`, 21

overrightarrowharpoonup package option, 15

path key, 27

path options key, 27

picture command key, 29

`\ProvideOverArrowCommand`, 17

psset key, 28

pstarrows package option, 17

pstricks package option, 16

`pstricks` command key, 28

`\RenewOverArrowCommand`, 17

`right arrow` key, 24

`\SetOverArrowsMethod`, 30

`\SetOverArrowsMethod*`, 30

`\SetOverArrowsSubscriptCommand`, 30

`shift left` key, 23

`shift leftright` key, 24

`shift right` key, 23

`\smallermathstyle`, 19

`space after arrow` key, 24

`space before arrow` key, 24

`stack macro` key, 30

`start` key, 25

`subactive` package option, 17

`subother` package option, 17

`subscripts` package option, 17

`\TestOverArrow`, 18

`\TestOverArrow*`, 18

`thinner` key, 28, 29

`tikz` package option, 16

`tikz` command key, 28

`tikz options` key, 27

`trim` key, 26

`trim end` key, 26

`trim middle` key, 25

`trim start` key, 25

`\underbar`, 22

`underbar` package option, 16

`undercommands` package option, 15

`\underleftarrow`, 22

`underleftarrow` package option, 15

`\underleftharpoondown`, 22

`underleftharpoondown` package option, 16

`\underleftharpoonup`, 22

`underleftharpoonup` package option, 16

`\underletrightarrow`, 22

`underletrightarrow` package option, 16

`\underrightarrow`, 22

`underrightarrow` package option, 15

`\underrightharpoondown`, 22

`underrightharpoondown` package option, 16

`\underrightharpoonup`, 22

`underrightharpoonup` package option, 16

`\vv`, 20

`\vv*`, 20

`\xjoinrel`, 19

