The \textbf{overarrows} package\textsuperscript{*}

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

February 15, 2023

Abstract

A \LaTeX{} package to create custom arrows over math expressions, mainly for vectors (but arrows can as well be drawn below). Arrows stretch with content, scale with math styles, and have a correct kerning when a subscript follows.

Short example:

\begin{align*}
\NewOverArrowCommand{overrightharpoon}{% \\
\text{end}\texttt{\textbackslash rightharpoonup}}
\begin{align*}
& \overrightharpoon{v} && \overrightharpoon{v}_{\text{subscript}} \\
& \overrightharpoon{ABCD} && \overrightharpoon{\star v}_{\text{subscript}}
\end{align*}
\end{align*}

Predefined commands are also provided:

\begin{itemize}
  \item to typeset vectors: \( \overrightarrow{v} \) \( \overrightarrow{AB} \),
  \item to draw arrows of various shapes above math expressions:
    \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \) \( \overrightarrow{AB} \),
  \item to draw arrows of various shapes under math expressions:
    \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \) \( \overleftarrow{AB} \).
\end{itemize}

\textsuperscript{*}This document corresponds to \texttt{overarrows} v1.1, dated 2023/02/15.
Contents

1 Presentation of the package 3

2 Introduction 3
   2.1 Vector arrows 3
   2.2 Stack and arrow macros 4
   2.3 Extensible arrows 4

3 Quick start 5
   3.1 Loading the package overarrows 5
   3.2 Commands creation 5
   3.3 Start and end of the arrow 5
   3.4 Size and position of the arrow 7
   3.5 Symbols assemblage 8
   3.6 Drawing the arrow with TikZ 10
   3.7 Drawing the arrow with \LaTeX picture environment 11

4 User interface 11
   4.1 Package options 11
      4.1.1 \vvecs configuration 11
      4.1.2 Predefined commands 13
      4.1.3 Other options 14
   4.2 Commands 15
      4.2.1 Macro for commands creation 15
      4.2.2 Useful macros for symbols assemblage 17
      4.2.3 Useful lengths for TikZ or picture environment 17
      4.2.4 Vectors macros 18
      4.2.5 Predefined commands 19
   4.3 Keys 20
      4.3.1 Arrow position and length settings 20
      4.3.2 Subscripts detection setting 22
      4.3.3 Symbols assemblage settings 23
      4.3.4 TikZ settings 24
      4.3.5 Picture environment settings 26
   4.4 Advanced commands and keys 26
      4.4.1 Advanced commands 27
      4.4.2 Advanced keys 27

5 Complements 28
   5.1 Know issues 28
      5.1.1 Math font change 28
      5.1.2 Detection of non standard subscripts 28
   5.2 Package dependencies 29
   5.3 Alternatives 29
   5.4 Changelog 29

6 Implementation 30

Index 46
1 Presentation of the package

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

- are fully customisable, at command definition, through a key-value interface;
- stretch with the content and can cover many characters, like in $\overrightarrow{AB}$;
- scale with math styles\(^1\), like in $\overrightarrow{uv}$.

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 $\overrightarrow{v}_{cm}$ (no extra space before the subscript, unlike the output of the unstarred variant: $\overrightarrow{v}_{cm}$).

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 PGF/TikZ commands. It’s also possible to create commands that draw the arrows under. Some predefined commands are provided, giving\(^2\), for arrow over:

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

and for arrow under:

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

2 Introduction

2.1 Vector arrows

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

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

\(^1\)\texttt{\textstyle, \scriptstyle, \scriptscriptstyle} and \texttt{\textbackslash scriptscriptstyle}.

\(^2\)Displayed here with the \texttt{old-arrows} \texttt{\textbackslash P.14} option.

2.2 Stack and arrow macros

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

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

Three macros are used here:

\mathpalette adapts the output to the current math style;
\overarrow@ is the stack macro, that puts the arrow above the content;
\rightarrowfill@ is the arrow macro, that holds the content of the arrow.

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

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

2.3 Extensible arrows

Arrows drawn by the commands \overrightarrow or \vv are built by joining math symbols, and made extensible by repetition of the central symbol\(^4\). 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 \(^5\).

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

long vector \hspace{24pt} 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 \(-\), \(\equiv\) 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):

long vector \hspace{24pt} long vector

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

\(^4\)Using the \TeX \cleaders command.

\(^5\)Using the Ti\TeX \cleaders command.
3 Quick start

3.1 Loading the package \texttt{overarrows}

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

\begin{verbatim}
\usepackage{overarrows}
\end{verbatim}

Options can be given, in a comma-separated list. For example, to use the predefined commands shown in the section 1, page 3, write:

\begin{verbatim}
\usepackage[allcommands, old-arrows]{overarrows}
\end{verbatim}

This define the commands (described in section 4.2.5, page 19):

\begin{itemize}
\item \texttt{\overrightarrow}\textsuperscript{P.19}
\item \texttt{\overleftarrow}\textsuperscript{P.19}
\item \texttt{\overleftrightarrow}\textsuperscript{P.19}
\item \texttt{\overrightharpoonup}\textsuperscript{P.19}
\item \texttt{\overrightharpoondown}\textsuperscript{P.19}
\item \texttt{\overleftharpoonup}\textsuperscript{P.19}
\item \texttt{\overleftharpoondown}\textsuperscript{P.19}
\item \texttt{\overbar}\textsuperscript{P.19}
\item \texttt{\underrightarrow}\textsuperscript{P.19}
\item \texttt{\underleftarrow}\textsuperscript{P.20}
\item \texttt{\underleftrightarrow}\textsuperscript{P.20}
\item \texttt{\underrightharpoonup}\textsuperscript{P.20}
\item \texttt{\underrightharpoondown}\textsuperscript{P.20}
\item \texttt{\underleftharpoonup}\textsuperscript{P.20}
\item \texttt{\underleftharpoondown}\textsuperscript{P.20}
\item \texttt{\underbar}\textsuperscript{P.20}
\end{itemize}

Note that the \texttt{old-arrows}\textsuperscript{P.14} 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 11.

3.2 Commands creation

Commands are created with \texttt{\NewOverArrowCommand}\textsuperscript{P.15}. This macro take two mandatory arguments: the name of the command (without backslash), and the arrow configuration as comma-separated list of key-values. By default, a right arrow is set:

\begin{verbatim}
\NewOverArrowCommand{myovercmd}{}
$\myovercmd{test}$
\end{verbatim}

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

\begin{verbatim}
$ v_{(\text{sub})} \quad \myovercmd(v)_{(\text{sub})} \quad \myovercmd*(v)_{(\text{sub})}$
\end{verbatim}

3.3 Start and end of the arrow

Extremities of the arrow are set by the keys \texttt{start}\textsuperscript{P.23} and \texttt{end}\textsuperscript{P.23}. For example, an arrow starting with a hook (symbols \texttt{\lhook}) and ending with two heads (symbol \texttt{\twoheadrightarrow}) is defined by:
Note that 	woheadrightarrow must be defined, as it is not in \LaTeX. This can be done with the package \texttt{amssymb}, by adding in preamble:

\begin{verbatim}
\usepackage{amssymb}
\end{verbatim}

With the previous definition, the result of the command \texttt{\overhooktwoheadrightarrow} is faulty:

\begin{verbatim}
\overhooktwoheadrightarrow(v) \quad \overhooktwoheadrightarrow(AB)
\end{verbatim}

The problem comes from symbols junction and the trimming used to obtain their overlap. It can be solved with the keys \texttt{trim start}\textsuperscript{P.23} and \texttt{trim end}\textsuperscript{P.23}, which are numbers and set the corresponding trimming in math units (typically 1/18 em). Appropriate values gives better results:

\begin{verbatim}
\overhooktwoheadrightarrow(v) \quad \overhooktwoheadrightarrow(AB)
\end{verbatim}

If the math font differs from the default \textit{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 \texttt{relbareda} from the \texttt{esvect} package. If needed, try to set the \texttt{middle}\textsuperscript{P.23} key with the symbol \texttt{relbar =}. The trimming should also be adapted:

\begin{verbatim}
\overhooktwoheadrightarrow(v) \quad \overhooktwoheadrightarrow(AB)
\end{verbatim}

Finding the correct values for \texttt{trim start}\textsuperscript{P.23}, \texttt{trim end}\textsuperscript{P.23} and \texttt{trim middle}\textsuperscript{P.23} may need many trials. For this purpose, the macro \texttt{\TestOverArrow}\textsuperscript{P.16} displays the result of a command for different lengths and math styles:

\begin{verbatim}
\TestOverArrow{overhooktwoheadrightarrow}
\end{verbatim}
3.4 Size and position of the arrow

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

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

The key trim \textsuperscript{•}P.23 sets trim start \textsuperscript{•}P.23, trim middle \textsuperscript{•}P.23 and trim end \textsuperscript{•}P.23 with the same value.

The previous arrow is visually too big. The macro \smallermathstyle \textsuperscript{•}P.17 allows to obtain a better result:

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

Note that \smallermathstyle \textsuperscript{•}P.17 should not be used for end \textsuperscript{•}P.23, because this last is formatted with the same math style as start \textsuperscript{•}P.23.

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 \textsuperscript{•}P.22:

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

Default arrows are slightly shifted to the right. For a left arrow, this should be reversed, using the keys shift left \textsuperscript{•}P.21 and shift right \textsuperscript{•}P.21. These keys set the corresponding shifts, in math units. Example:
Finally, the key arrow under \cite{P.21} places the arrow below the content, instead of above (and space before arrow \cite{P.22} sets the space upon it):

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

3.5 Symbols assemblage

Many \LaTeX math symbols are built by assemblage, using the macro \texttt{\LaTeX\texttt{\joinrel}} which remove 3 math units of horizontal space. The overarrows package provides a flexible version of \texttt{\joinrel}, called \texttt{xjoinrel} \cite{P.17}, 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 \texttt{\tttail} from the symbol \texttt{\succ}:

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

Thus defined, the macro \texttt{\tttail} can be used in arrow definition:

\texttt{\models} is defined as \texttt{\mathrel{\models}} and corresponds to the assemblage of a vertical line and the symbol \texttt{\Relbar}. The command \texttt{\mathrel} modifies the spacing according to the math relation class ; \texttt{\Relbar} corresponds to the equal sign (its definition is \texttt{\mathrel{=}}).

\footnote{For example, the symbol \texttt{\models \models \models} is defined as \texttt{\mathrel{\models}} and corresponds to the assemblage of a vertical line and the symbol \texttt{\Relbar \Relbar \Relbar}. The command \texttt{\mathrel{\models \models \models}} modifies the spacing according to the math relation class ; \texttt{\Relbar \Relbar \Relbar} corresponds to the equal sign (its definition is \texttt{\mathrel{\Relbar \Relbar \Relbar}}).}
Here the \texttt{min length} 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 \texttt{\smallermathstyle}. But a small tail and a normal sized head are not aligned; as {\smallermathstyle\tttail}\xjoinrel\rightarrow gives:

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

Text symbols, namely symbols that are not defined in math mode, can also be used. They should yet be enclosed in the \texttt{\text} macro, from the \texttt{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 \texttt{lasy} font:
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 4), 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 \texttt{tikz} option to the overarrows package:

\begin{verbatim}
\usepackage{overarrows}
\end{verbatim}

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

- \texttt{overarrowlength} for the arrow length,
- \texttt{overarrowthickness} and \texttt{overarrowsmallerthickness} for the arrow thickness.

These lengths are computed at each utilisation of a command created with the \texttt{tikz} optional argument.

Without any other configuration, a right arrow is drawn:

\begin{verbatim}
\NewOverArrowCommand[tikz]{overtikzarrow}{}
$\overtikzarrow{v} \quad \overtikzarrow{AB}$
\end{verbatim}

Keys to use Tikz are described in section 4.3.4, page 24. Main keys are: \texttt{tikz options}, \texttt{path options} and \texttt{path}. It's also possible to append settings with \texttt{add tikz options} and \texttt{add path options}. The full TikZ command used to draw the arrow can as well be entirely redefined with the key \texttt{tikz command}.

Here is an example of an arrow drawn with TikZ:

\begin{verbatim}
\NewOverArrowCommand[tikz]{overarchedleftarrow}{% 
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, }
$\overarchedleftarrow{v} \quad \overarchedleftarrow{ABCD} $
\end{verbatim}

\footnote{Note that the \texttt{tikz} option isn't mandatory to use TikZ commands in overarrows. The tikz package and its library arrows.meta can be loaded independently.}

\footnote{TikZ arrows are very powerful, but much slower to draw than the default method using assemblage of math symbols.}
3.7 Drawing the arrow with \LaTeX picture environment

As well as TikZ, the \LaTeX picture environment can be used to draw the arrow. For this, the optional argument picture must be passed to \(\texttt{NewOverArrowCommand}\). Like for TikZ, the three lengths \(\overarrowlength\) and \(\overarrowsmallerthickness\) can be used in picture commands. By default, a right vector is drawn:

\[
\overpictarrow{v} \quad \overpictarrow{AB}
\]

If overarrows is loaded with the option \texttt{pstars}, the package \texttt{pict2e} is used and a PSTricks style vector arrows is set. This gives:

\[
\overbandedarrow{v} \quad \overbandedarrow{AB}
\]

Keys to use \LaTeX picture environment are described in section 4.3.5, page 26. The main keys are picture command, geometry and line thickness. Here is an example:

\[
\overbandedarrow{v} \quad \overbandedarrow{AB}
\]

4 User interface

4.1 Package options

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

The option \texttt{esvect} is set by default. This can be overridden with \texttt{noesvect}.

4.1.1 esvect configuration

\texttt{esvect}
Loads the `esvect` package and redefines its vector commands through the `overarrows` mechanism. Original `esvect` `\vv` macro is still available with `\esvectvv`.

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: `\relbar` (option `a`), `\relbar` (option `b`), `\relbar` (option `c`), `\relbar` (option `d`), `\relbar` (option `e`), `\relbar` (option `f`), `\relbar` (option `g`) or `\relbar` (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`.

`esvecta`
Loads the `esvect` package with the `a` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvectb`
Loads the `esvect` package with the `b` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvectc`
Loads the `esvect` package with the `c` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvectd`
Loads the `esvect` package with the `d` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvecte`
Loads the `esvect` package with the `e` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvectf`
Loads the `esvect` package with the `f` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.

`esvectg`
Loads the `esvect` package with the `g` option.
`\fldr` corresponds to the symbol `\relbar`. `\vv` command gives: \( \vec{AB} \) `grad`.
esvecth

Loads the esvect package with the h option.

\vector \mathfrak{fdr} corresponds to the symbol \( \vec{v} \). \vector \mathfrak{vv} command gives: \( \vec{v} \ \vec{AB} \ \vec{\nabla} \).

4.1.2 Predefined commands

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

Three options are also available to define sets 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: \( \vec{v} \), \( \vec{AB} \), \( \vec{\nabla} \).

overleftarrow

Defines the \( \overleftarrow \) command: \( \overleftarrow{v} \), \( \overleftarrow{AB} \), \( \overleftarrow{\nabla} \).

overleftrightarrow

Defines the \( \overleftrightarrow \) command: \( \overleftarrow{v} \), \( \overleftarrow{AB} \), \( \overleftarrow{\nabla} \).

overrightharpoonup

Defines the \( \overrightharpoonup \) command: \( \overrightarrow{v} \), \( \overrightarrow{AB} \), \( \overrightarrow{\nabla} \).

overrightharpoondown

Defines the \( \overrightharpoondown \) command: \( \overrightarrow{v} \), \( \overrightarrow{AB} \), \( \overrightarrow{\nabla} \).

overleftharpoonup

Defines the \( \overleftharpoonup \) command: \( \overrightarrow{v} \), \( \overrightarrow{AB} \), \( \overrightarrow{\nabla} \).

overleftharpoondown

Defines the \( \overleftharpoondown \) command: \( \overrightarrow{v} \), \( \overrightarrow{AB} \), \( \overrightarrow{\nabla} \).
overleftharpoondown

Defines the \overleftharpoondown command: \( \overleftharpoondown \mathbf{v}, \overleftharpoondown \mathbf{AB}, \overleftharpoondown \mathbf{grad} \).

overbar

Defines the \overbar command: \( \overbar \mathbf{v}, \overbar \mathbf{AB}, \overbar \mathbf{grad} \).

Under arrows

underrightarrow

Defines the \underrightarrow command: \( \underrightarrow \mathbf{v}, \underrightarrow \mathbf{AB}, \underrightarrow \mathbf{grad} \).

underleftarrow

Defines the \underleftarrow command: \( \underleftarrow \mathbf{v}, \underleftarrow \mathbf{AB}, \underleftarrow \mathbf{grad} \).

underlefttrightarrow

Defines the \underlefttrightarrow command: \( \underlefttrightarrow \mathbf{v}, \underlefttrightarrow \mathbf{AB}, \underlefttrightarrow \mathbf{grad} \).

underrightharpoonup

Defines the \underrightharpoonup command: \( \underrightharpoonup \mathbf{v}, \underrightharpoonup \mathbf{AB}, \underrightharpoonup \mathbf{grad} \).

underleftharpoonup

Defines the \underleftharpoonup command: \( \underleftharpoonup \mathbf{v}, \underleftharpoonup \mathbf{AB}, \underleftharpoonup \mathbf{grad} \).

underbar

Defines the \underbar command: \( \underbar \mathbf{v}, \underbar \mathbf{AB}, \underbar \mathbf{grad} \).

4.1.3 Other options

old-arrows

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

When the old-arrows option is set, the commands \overrightarrow, \overleftarrow, \underleftarrow, \underrightarrow, \underrightarrow and \underlefttrightarrow give respectively: \( \overrightarrow \mathbf{AB}, \underleftarrow \mathbf{AB}, \underrightarrow \mathbf{AB}, \underleftrightarrow \mathbf{AB}, \underlefttrightarrow \mathbf{AB} \) and \( \overleftarrow \mathbf{AB} \).
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.

pstarrows
Loads the pict2e package, with its option pstarrows. Vectors using \LaTeX
picture environment gives then $\overrightarrow{AB}$ instead of $\overrightarrow{AB}$.
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\textsuperscript{-P.22} to true.
This option also impacts the command \vv\textsuperscript{-P.18} and all predefined com-
mands, so that they automatically use their starred variant when a subscript
follows.

subactive
Sets to 13 (active catcode category) the catcode of the \_ symbol used for
subscript detection, when this is enabled by the key detect subscripts\textsuperscript{-P.22}
(see the section 5.1.2, page 28).

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

4.2 Commands
4.2.1 Macro for commands creation
\NewOverArrowCommand{⟨method⟩}{⟨name⟩}{⟨keys⟩}
\RenewOverArrowCommand{⟨method⟩}{⟨name⟩}{⟨keys⟩}
\ProvideOverArrowCommand{⟨method⟩}{⟨name⟩}{⟨keys⟩}
\DeclareOverArrowCommand{⟨method⟩}{⟨name⟩}{⟨keys⟩}

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

\NewOverArrowCommand raises an error if \⟨name⟩ is already defined.
\RenewOverArrowCommand raises an error if \⟨name⟩ is undefined.
\ProvideOverArrowCommand sets \⟨name⟩ if the command is undefined and
does nothing if it is already defined, without raising any error.
\DeclareOverArrowCommand sets \( \langle \text{name} \rangle \), whether the command is already defined or not, without raising any error.

The \( \langle \text{method} \rangle \) used to draw the arrow must be:

- \text{\texttt{symb}} to draw the arrow by symbols assemblage (default);
- \text{\texttt{tikz}} to draw the arrow with PGF/TikZ;
- \text{\texttt{picture}} to draw the arrow with the \LaTeX\ picture environment.

With no \( \langle \text{method} \rangle \) argument, the \texttt{symb} method is chosen.

\( \langle \text{keys} \rangle \) is a comma-separated list of keys-values. Available keys depends of the \( \langle \text{method} \rangle \) chosen and are described in section 4.3, page 20.

\begin{verbatim}
\NewOverArrowCommand[tikz]{myoverarrow}{arrows={Bar-Bar}, center arrow}\$
\myoverarrow{v} \qquad \myoverarrow{ABCD} \$
\end{verbatim}

\begin{verbatim}
\TestOverArrow[\langle \text{pattern} \rangle]{\langle \text{name} \rangle}
\TestOverArrow*[\langle \text{pattern} \rangle]{\langle \text{name} \rangle}
\end{verbatim}

Displays the result of the command \( \langle \text{name} \rangle \) for patterns of various lengths and for the four math styles. A custom \( \langle \text{pattern} \rangle \) can be added to the predefined ones.

The starred variant \TestOverArrow* displays a full report, including kerning tests of the commands \( \langle \text{name} \rangle \) and \( \langle \text{name} \rangle^* \).

\begin{verbatim}
\TestOverArrow*[my~pattern]{vv}
\end{verbatim}

\begin{verbatim}
Test of \texttt{vv} and \texttt{vv*} macros

\begin{tabular}{|c|c|c|c|}
\hline
\texttt{\displaystyle} & \texttt{textstyle} & \texttt{scriptstyle} & \texttt{scriptscriptstyle} \\
\hline
\texttt{v} & \texttt{v} & \texttt{v} & \texttt{v} \\
\texttt{AB} & \texttt{AB} & \texttt{AB} & \texttt{AB} \\
\texttt{my long vector} & \texttt{my long vector} & \texttt{my long vector} & \texttt{my long vector} \\
\texttt{my pattern} & \texttt{my pattern} & \texttt{my pattern} & \texttt{my pattern} \\
\hline
\end{tabular}

\texttt{vv} kerning
\begin{align*}
\vec{v}_0 \cdot \vec{v} & = \vec{v}_x + \vec{v}_y + \vec{v}_z = v_x \vec{\imath} + v_y \vec{\jmath} + v_z \vec{k} \\
\end{align*}

\texttt{vv*} kerning
\begin{align*}
\vec{v}_0 \cdot \vec{v} & = \vec{v}_x + \vec{v}_y + \vec{v}_z = v_x \vec{\imath} + v_y \vec{\jmath} + v_z \vec{k} \\
\end{align*}
\end{verbatim}
4.2.2 Useful macros for symbols assemblage

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

`\xjoinrel[(number)]`

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

\[
\text{\scalebox{2}{$\triplebar \quad \triplebararrow$}}
\]

\[
\text{\scalebox{2}{$\triplebar \xjoinrel \triplebararrow$}}
\]

`\smallermathstyle`

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

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

\[
\text{\scriptstyle \overparabola{v}} \quad \text{\scriptstyle \overparabola{ABCD}}
\]

\[
\text{\scriptstyle \overparabola{v}} \quad \text{\scriptstyle \overparabola{ABCD}}
\]

4.2.3 Useful lengths for TikZ or picture environment

Arrows drawn with graphic languages, like PGF/TikZ 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` or `picture` method, so they can be used in drawing commands.

\[
\text{\scriptstyle \overparabola{v}} \quad \text{\scriptstyle \overparabola{ABCD}}$
\]

\[
\text{\scriptstyle \overparabola{v}} \quad \text{\scriptstyle \overparabola{ABCD}}$
\]
\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.\footnote{p.\textsuperscript{20}}.

\overarrowthickness
Is set to the default rule thickness of the current math style. That is:
- \texttt{\fontdimen 8 \textfont 3} in \texttt{\displaystyle} or \texttt{\textstyle};
- \texttt{\fontdimen 8 \scriptfont 3} in \texttt{\scriptstyle};
- \texttt{\fontdimen 8 \scriptscriptfont 3} in \texttt{\scriptscriptstyle}.

\overarrowsmallerthickness
Is set to the default rule thickness of the next smaller math style. That is:
- \texttt{\fontdimen 8 \scriptfont 3} in \texttt{\displaystyle} or \texttt{\textstyle};
- \texttt{\fontdimen 8 \scriptscriptfont 3} in \texttt{\scriptstyle} or \texttt{\scriptscriptstyle}.

4.2.4 Vectors macros
The macro \vv, dedicated to vectors, is automatically defined when the option esvect\footnote{p.\textsuperscript{11}} is set (which is the default). It is a clone of the esvect command provided by the esvect package, but its starred variant has a correct kerning when followed by a subscript.

\vv{\langle content\rangle}
\vv*{\langle content\rangle}

Draws a vector arrow upon math \langle content\rangle. The shape of the arrow depends on the corresponding options described in section 4.1.1, page 11: esvecta\footnote{p.\textsuperscript{12}}, esvectb\footnote{p.\textsuperscript{12}}, esvectc\footnote{p.\textsuperscript{12}}, esvectd\footnote{p.\textsuperscript{12}}, esvekte\footnote{p.\textsuperscript{12}}, esvectf\footnote{p.\textsuperscript{12}}, esvectg\footnote{p.\textsuperscript{12}}, esvecth\footnote{p.\textsuperscript{13}}.

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

\esvectvv
Is simply the backup of the original esvect \vv command.
4.2.5 Predefined commands

Predefined commands are defined if the corresponding option is set (see section 4.1.2, page 13). The commands \overrightarrow, \overleftarrow, \overleftrightarrow, \underrightarrow, \underleftarrow and \underleftrightarrow are affected by the option old-arrows ° P. 14.

Over arrows

\overrightarrow

\overleftarrow

\overleftrightarrow

\overrightharpoonup

\underrightarrow

The shape of the arrow is smaller if the option old-arrows ° P. 14 is set.
4.3 Keys
The customisation of arrows is done at command creation through a key-value interface provided by the \texttt{pgfkeys} package (with \texttt{/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 15 for the documentation of commands creation).

Length

\texttt{min length}={⟨number⟩} (no default, see below for the initial value)

Sets the minimal arrow length to ⟨number⟩ math units. The arrow length is set from content width, or, if larger, to this value.

The initial value of \texttt{min length} depends on the ⟨method⟩ chosen at command creation (see section 4.2.1, page 15 for the documentation of commands creation):
\NewOverArrowCommand\overlongarrow{\text{v}} \quad \overlongarrow{ABCDEF}

Placement

\arrowunder \quad \text{(default autoconfig, initially unset)}
\arrowunder=\text{autoconfig|noconfig}

Places the arrow under, instead of over.

\arrowunder \text{or} \arrowunder=\text{autoconfig} also configures suitably the key
\text{detection subscripts} \text{F.22} to \text{false} and the key \text{before arrow} \text{F.22} to
get an additional space over the arrow.

\arrowunder=\text{noconfig} does not do any additional configuration.

\NewOverArrowCommand\underhooks{\text{v}} \quad \underhooks{AB}

Horizontal shifts

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

Shifts the left side of the arrow by \text{number} \text{math units} (positive number
means a shift to the right).

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

Shifts the right side of the arrow by \text{number} \text{math units} (positive number
means a shift to the left).

The initial value of \text{shift right} depends on the \text{method} chosen at com-
mand creation (see section 4.2.1, page 15 for the documentation of commands
creation):

\begin{itemize}
\item \langle\text{number}\rangle = 0 \quad \text{for the symb method (default)};
\item \langle\text{number}\rangle = -2 \quad \text{for the tikz and picture methods}.
\end{itemize}
shift leftright=\langle number \rangle  
Sets shift left and shift right to the same \langle number \rangle value.

center arrow  
Sets shift left and shift right to zero.

left arrow  
Sets shift left \cdot P.^{21} to zero and shift right \cdot P.^{21} to \langle number \rangle.

right arrow  
Sets shift right \cdot P.^{21} to zero and shift left \cdot P.^{21} to \langle number \rangle.

Vertical adjunct

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

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

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

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

4.3.2 Subscripts detection setting

This key is available whatever the method chosen at command creation (see section 4.2.1, page 15 for the documentation of commands creation).

detect subscripts=true|false  
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 \cdot P.^{15} 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 28).

4.3.3 Symbols assemblage settings

The following keys are available for arrows drawn with the default symb method (see section 4.2.1, page 15 for the documentation of commands creation).

\NewOverArrowCommand{autosub}{detect subscripts}
$\imath_0 \qquad \text{}\begin{array}{l}
\{\text{\imath}_0\} \\
\{\text{\imath}_0\} \\
\{\text{\imath}_0\}
\end{array}$
$
o_{t_0} \rightarrow o_{t_0} \rightarrow o_{t_0}$

\NewOverArrowCommand{smalleroverrightarrow}{
\langle\text{\imath}\rangle_0 \quad \text{\begin{array}{l}
\{\text{\imath}_0\} \\
\{\text{\imath}_0\} \\
\{\text{\imath}_0\}
\end{array}}$

4.3.3 Symbols assemblage settings

The following keys are available for arrows drawn with the default symb method (see section 4.2.1, page 15 for the documentation of commands creation).

\NewOverArrowCommand{smalleroverrightarrow}{
\langle\text{\imath}\rangle_0 \quad \text{\begin{array}{l}
\{\text{\imath}_0\} \\
\{\text{\imath}_0\} \\
\{\text{\imath}_0\}
\end{array}}$

$\smalleroverrightarrow{v} \quad \smalleroverrightarrow{AB}$

sets the \langle\text{\imath}\rangle_0 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.14 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 \rightarrow, is used to alter the symbols, it should be applied both to start and middle (but not to end).

\NewOverArrowCommand{smalleroverrightarrow}{
\langle\text{\imath}\rangle_0 \quad \text{\begin{array}{l}
\{\text{\imath}_0\} \\
\{\text{\imath}_0\} \\
\{\text{\imath}_0\}
\end{array}}$

\NewOverArrowCommand{smalleroverrightarrow}{
\langle\text{\imath}\rangle_0 \quad \text{\begin{array}{l}
\{\text{\imath}_0\} \\
\{\text{\imath}_0\} \\
\{\text{\imath}_0\}
\end{array}}$

$\smalleroverrightarrow{v} \quad \smalleroverrightarrow{AB}$

trim start={(\langle\text{number}\rangle)}

Trims \langle\text{number}\rangle math units from the right side of the start symbol.

trim middle={(\langle\text{number}\rangle)}

Trims \langle\text{number}\rangle math units from both left and right sides of the middle symbol.

trim end={(\langle\text{number}\rangle)}

Trims \langle\text{number}\rangle math units from the left side of the end symbol.

trim={(\langle\text{number}\rangle)}

Sets trim start, trim middle and trim end to the same \langle\text{number}\rangle value.

no trimming

Clears trim start, trim middle and trim end.
middle config=auto|relbar|relbareda (no default)

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

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

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

For middle config = auto, middle is set with middle config = relbareada if the option esvect is set (which is the default) and middle config = relbar 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:

\begin{itemize}
  \item start=\relbar
  \item middle=\relbar
  \item end=\rightarrow
  \item trim start=7
  \item trim middle=2
  \item trim end=7
  \item shift leftright=0
  \item after arrow=
  \item before arrow=
\end{itemize}

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 \vec command.

esvect or esvect=mimic sets the corresponding keys suitably:

\begin{itemize}
  \item start=\relbaredd
  \item middle=\relbareda
  \item end=\vec
  \item trim start=1.5
  \item trim middle=0
  \item trim end=1.5
  \item space before arrow=-.7pt
  \item space after arrow=-.3pt
  \item right arrow=2
\end{itemize}

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

4.3.4 TikZ settings

If, at command creation (see section 4.2.1, page 15 for the documentation of commands creation), the tikz method is chosen, then the arrow is drawn by the command:

\begin{verbatim}
\tikz[tikz options]{tikz command}
\end{verbatim}

where tikz options and tikz command are two keys described below. When tikz command is let unset, the drawing command turns into:

\begin{verbatim}
\tikz[tikz options]{\draw[path options] path;}
\end{verbatim}
The best way to customise \texttt{tikz} arrows is then to set the keys \texttt{tikz options P. 25}, \texttt{path options P. 25} and \texttt{path P. 25}, preferably through the handy alternatives: \texttt{add tikz options P. 25}, \texttt{add path options P. 25}, \texttt{arrows P. 25}, \texttt{line thickness P. 25} or \texttt{thinner P. 25}.

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

The following keys are available when the \texttt{tikz} method is chosen.

- \texttt{tikz options}={\langle \text{TikZ options} \rangle}  
  (no default, initially x=\overarrowlength, line width=\overarrowthickness)
  
  Sets TikZ options to \langle \text{TikZ options} \rangle.

- \texttt{path options}={\langle \text{path options} \rangle}  
  (no default, initially arrows=-Classical TikZ Rightarrow, cap=round)
  
  Sets TikZ path options to \langle \text{path options} \rangle.

- \texttt{path}={\langle \text{path specification} \rangle}  
  (no default, initially (0,0)--(1,0))
  
  Sets TikZ path specification to \langle \text{path} \rangle (the ending semicolon is automatically appended).

- \texttt{add tikz options}={\langle \text{TikZ options} \rangle}  
  (no default)
  
  Appends the options \langle \text{TikZ options} \rangle to the key \texttt{tikz options}.

- \texttt{add path options}={\langle \text{path options} \rangle}  
  (no default)
  
  Appends the options \langle \text{path options} \rangle to the key \texttt{path options}.

- \texttt{arrows}={\langle \text{arrow specification} \rangle}  
  (no default)
  
  Appends the option \texttt{arrows}={\langle \text{arrow specification} \rangle} to the key \texttt{path options}.

- \texttt{line thickness}={\langle \text{length} \rangle}  
  (no default)
  
  Appends the option \texttt{line width}={\langle \text{length} \rangle} to the key \texttt{path options}.

- \texttt{thinner}
  
  Sets the keys \texttt{line thickness} with \texttt{\overarrowssmallerthickness}.

- \texttt{tikz command}={\langle \text{TikZ command} \rangle}  
  (initially unset)
  
  Sets the \langle \text{TikZ command} \rangle used to draw the arrow. If left unset, the value \texttt{\draw[\text{path options}] \text{path};} is used.
4.3.5 Picture environment settings

If, at command creation (see section 4.2.1, page 15 for the documentation of commands creation), the \texttt{picture} method is chosen, then the arrow is drawn with by:

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

where \texttt{geometry}, \texttt{line thickness} and \texttt{picture command} are three keys described below.

\begin{verbatim}
% ^^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(0.5overarrowlength,0){\arc[180,0]{0.6overarrowlength}}
 },
 \geometry=,%
 (1.2\overarrowlength,0.5\overarrowlength)(-0.1\overarrowlength,0.2ex)
 },
 \thinner, center arrow,
 $\overarc{v}$ \quad $\overarc{AB}$ $\overarc{v}$ \quad $\overarc{AB}$
\end{verbatim}

The following keys are available when the \texttt{picture} method is chosen.

\texttt{picture command}={\langle\texttt{picture command}\rangle}

(no default, initially \texttt{\put(0,0){\vector(1,0){\overarrowlength}}})

Sets picture command to \langle\texttt{picture command}\rangle.

\texttt{geometry}={\langle\texttt{picture geometry specification}\rangle}

(no default, initially \texttt{(\overarrowlength,1ex)(0,-0.5ex)})

Sets picture geometry to \langle\texttt{picture geometry specification}\rangle.

\texttt{line thickness}={\langle\texttt{length}\rangle}

(no default)

Sets the picture line thickness to \langle\texttt{length}\rangle.

\texttt{thinner}

(no default)

Sets the keys \texttt{line thickness} with \texttt{\overarrowsmallerthickness}.

4.4 Advanced commands and keys

The following commands and keys are used in the implementation of the \texttt{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{\textit{(command)}}

Sets to \textit{(command)} the command used for subscript detection, when this is enabled by the key detect subscripts \(^{P.22}\) (see the section 5.1.2, page 28).

\SetOverArrowsMethod{\textit{(stack mechanism)}}{\textit{(name)}}{\textit{(pre code)}}{\textit{(keys def)}}

\SetOverArrowsMethod*{\textit{(name)}}{\textit{(pre code)}}{\textit{(keys def)}}

Defines the method \textit{(name)}, to be used in commands \texttt{\NewOverArrowCommand} \(^{P.15}\), \texttt{\RenewOverArrowCommand} \(^{P.15}\), \texttt{\ProvideOverArrowCommand} \(^{P.15}\) or \texttt{\DeclareOverArrowCommand} \(^{P.15}\). When the \textit{(name)} method is chosen, corresponding keys are defined by \textit{(keys def)}. This must set, in particular, the keys no stack macro hook and no arrow macro hook \(^{P.28}\). Optional code \textit{(pre code)} is evaluated before the keys definition.

The unstarred variant automatically defines the key no stack macro hook, according to the value of the optional \textit{(stack mechanism)}. This one must be:

- **fill** if arrow macro creates extensible arrows (typically with \texttt{\cleaders}). In this case, the arrow macro (defined by no arrow macro hook \(^{P.28}\)) is called with the math style, passed as argument (it can be, for example, the macro \texttt{\rightarrowfill@} used by amsmath \texttt{\overrightarrow}).

- **fill** is the mechanism used by the symb method.

- **lens** if arrow macro creates fixed-length arrows, and needs the computation of lengths \texttt{\overarrowlength} \(^{P.18}\), \texttt{\overarrowthickness} \(^{P.18}\) and \texttt{\overarrowsmallerthickness} \(^{P.18}\). In this case, the arrow macro (defined by no arrow macro hook \(^{P.28}\)) is called without argument. lens is the mechanism used by the tikz and picture methods.

Without optional \textit{(stack mechanism)}, fill is used. The starred variant does not set the key no stack macro hook.

4.4.2 Advanced keys

**stack macro**={\textit{(stack definition)}}

(no default, initially unset)

Defines the stack macro to be \textit{(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). \textit{(stack definition)} can be, for example, the macro \texttt{\overarrow@} used by amsmath \texttt{\overrightarrow}.

**arrow macro**={\textit{(arrow definition)}}

(no default, initially unset)

Defines the arrow macro (used in the stack macro) by to be \textit{(arrow definition)}.

**no stack macro hook**={\textit{(code)}}

(no default)

Sets the \textit{(code)} executed if stack macro is left unset, after user evaluation of \textit{(keys)} in \texttt{\NewOverArrowCommand} \(^{P.15}\), \texttt{\RenewOverArrowCommand} \(^{P.15}\), \texttt{\ProvideOverArrowCommand} \(^{P.15}\) or \texttt{\DeclareOverArrowCommand} \(^{P.15}\). \textit{(code)} must configure stack macro accordingly to the user keys setting.
no arrow macro hook={⟨code⟩}  \[\text{(no default)}\]

Sets the ⟨code⟩ executed if arrow macro is left unset, after user evaluation of ⟨keys⟩ in \NewOverArrowCommand P.15, \RenewOverArrowCommand P.15, \ProvideOverArrowCommand P.15 or \DeclareOverArrowCommand P.15. ⟨code⟩ must configure arrow macro P.27 accordingly to the user keys setting.

fill macro={⟨definition⟩}  \[\text{(no default, initially unset)}\]

Defines the fill macro to be ⟨definition⟩. The fill macro is used by arrows created with the symb method, to set arrow macro P.27 in no arrow macro hook. It is called with fours arguments: start, middle and end symbols used to draw the arrow, and the math style. ⟨definition⟩ can be, for example, the macro \arrowfill@ used by amsmath \overrightarrow.

5 Complements

5.1 Know issues

5.1.1 Math font change

If the math font differs from the default Computer Modern, arrow drawn with the symb method may have a central part of the arrow with inappropriate position or line width. This is because the default symbol used for the arrow line is \relbareda from the esvect package. This can be fixed with the noesvect P.12 option.

5.1.2 Detection of non standard subscripts

The subscript detection enabled by the key detect subscripts P.22 is based on the \LaTeX macro \@ifnextchar. The detection may fail if the standard subscript command is modified of 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 (subscribe catcode category) to 12 (other catcode category).

To handle these cases, the command used for subscript detection can be redefined with \SetOverArrowsSubscriptCommand P.27. Compatibility with the spbmark package is then obtained by:

\SetOverArrowsSubscriptCommand{\sub}

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

\SetOverArrowsSubscriptCommand{_{}}

after the \begin{document} (namely, after the catcode redefinition done by altsubsup).
Alternatively, two package options handle the cases where the catcode of the underscore “_” symbol is changed: \texttt{subother}^{P.15} (for catcode 12, or \textit{other}) and \texttt{subactive}^{P.15} (for catcode 13, or \textit{active}). Hence, setting the \texttt{subother}^{P.15} option is sufficient for compatibility with the \texttt{altsubsup} package (no need of \texttt{\SetOverArrowsSubscriptCommand}^{P.27}). Note, that with options \texttt{subother}^{P.15} and \texttt{subactive}^{P.15}, the command \texttt{\TestOverArrow*}^{P.16} may give bad results for kerning test, as defined before the catcode redefinition.

### 5.2 Package dependencies

The following packages are used by \texttt{overarrows}:

- \texttt{amsmath}
- \texttt{etoolbox}
- \texttt{pgfkeys}
- \texttt{esvect} (unless the option \texttt{novesvect}^{P.12} is used)
- \texttt{old-arrows} (when the option \texttt{old-arrows}^{P.14} is used)
- \texttt{tikz} (when the \texttt{tikz} method or the option \texttt{tikz}^{P.15} is used)
- \texttt{pict2e} (when the option \texttt{pstars}^{P.15} is used)

\LaTeX{} distributions prior to 2020/10/01 must load the \texttt{xparse} package before \texttt{overarrows}.

### 5.3 Alternatives

- \texttt{esvect package} (https://www.ctan.org/pkg/esvect), by Eddie Saudrais, provides the fine vector macro \texttt{\vv}. This package is loaded by default by \texttt{overarrows}.

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

- \texttt{overrightarrow package} (https://www.ctan.org/pkg/overrightarrow), by Robin Fairbairns, provides the \texttt{\Overrightarrow} which is an amalgam of \texttt{\overrightarrow} and \texttt{\Rightarrow}.

- \texttt{harpoon package} (https://ctan.org/pkg/harpoon), by Tobias Kuipers, provides over- and under-harpoon symbol commands.

### 5.4 Changelog

- **v1.1** Support for non-standard subscripts
- **v1.0.1** Bug fix for under* options.
- **v1.0** Initial version.
6 Implementation

Management of options

Declaration of conditionals

\newif\ifovar@option@oldarrows@
\newif\ifovar@option@esvect@
\ovar@option@esvect@true\PassOptionsToPackage{f}{esvect}
\newif\ifovar@option@tikz@
\newif\ifovar@option@pstarrows@
\newif\ifovar@detectsubscripts@
\newif\ifovar@option@subother@
\newif\ifovar@option@subactive@
\newif\ifovar@option@debug@

Following conditionals are for predefined commands.

\newif\ifovar@option@overrightarrow@
\newif\ifovar@option@underrightarrow@
\newif\ifovar@option@overleftarrow@
\newif\ifovar@option@underleftarrow@
\newif\ifovar@option@overleftrightarrow@
\newif\ifovar@option@underleftrightarrow@
\newif\ifovar@option@overrightharpoonup@
\newif\ifovar@option@underrightharpoonup@
\newif\ifovar@option@overleftharpoonup@
\newif\ifovar@option@underleftharpoonup@
\newif\ifovar@option@overleftharpoondown@
\newif\ifovar@option@underleftharpoondown@
\newif\ifovar@option@overbar@
\newif\ifovar@option@underbar@

Declaration of options

\DeclareOption{esvect}{\ovar@option@esvect@true}
\DeclareOption{noesvect}{\ovar@option@esvect@false}
\DeclareOption{esvecta}{\ovar@option@esvect@true\PassOptionsToPackage{a}{esvect}}
\DeclareOption{esvectb}{\ovar@option@esvect@true\PassOptionsToPackage{b}{esvect}}
\DeclareOption{esvectc}{\ovar@option@esvect@true\PassOptionsToPackage{c}{esvect}}
\DeclareOption{esvectd}{\ovar@option@esvect@true\PassOptionsToPackage{d}{esvect}}
\DeclareOption{esvecte}{\ovar@option@esvect@true\PassOptionsToPackage{e}{esvect}}
\DeclareOption{esvectf}{\ovar@option@esvect@true\PassOptionsToPackage{f}{esvect}}
\DeclareOption{esvectg}{\ovar@option@esvect@true\PassOptionsToPackage{g}{esvect}}
\DeclareOption{esvecth}{\ovar@option@esvect@true\PassOptionsToPackage{h}{esvect}}
\DeclareOption{old-arrows}{\ovar@option@oldarrows@true}
\DeclareOption{tikz}{\ovar@option@tikz@true}
\DeclareOption{pstarrows}{\ovar@option@pstarrows@true}
\DeclareOption{subscripts}{\ovar@detectsubscripts@true}
\DeclareOption{subother}{\ovar@option@subother@true}
\DeclareOption{subactive}{\ovar@option@subactive@true}
\DeclareOption{debug}{\ovar@option@debug@true}

Following options are for predefined commands.

\DeclareOption{overrightarrow}{\ovar@option@overrightarrow@true}
\DeclareOption{underrightarrow}{\ovar@option@underrightarrow@true}
\DeclareOption{overleftarrow}{\ovar@option@overleftarrow@true}
\DeclareOption{underleftarrow}{\ovar@option@underleftarrow@true}
\DeclareOption{overleftrightarrow}{\ovar@option@overleftrightarrow@true}
\DeclareOption{underleftrightarrow}{\ovar@option@underleftrightarrow@true}
\DeclareOption{overrightharpoonup}{\ovar@option@overrightharpoonup@true}
\DeclareOption{underrightharpoonup}{\ovar@option@underrightharpoonup@true}
\DeclareOption{overleftharpoonup}{\ovar@option@overleftharpoonup@true}
\DeclareOption{underleftharpoonup}{\ovar@option@underleftharpoonup@true}
\DeclareOption{overleftharpoondown}{\ovar@option@overleftharpoondown@true}
\DeclareOption{underleftharpoondown}{\ovar@option@underleftharpoondown@true}
\DeclareOption{overbar}{\ovar@option@overbar@true}
\DeclareOption{underbar}{\ovar@option@underbar@true}
Options processing

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

Package dependencies

\RequirePackage{amsmath}
\RequirePackage{etoolbox}

Option \texttt{old-arrows} \footnote{P. 14}. Configuration of arrows used for predefined commands.
\let\ovar@rightarrow\rightarrow
\let\ovar@leftarrow\leftarrow
\ifovar@option@oldarrows@
    \RequirePackage[old]{old-arrows}
    \let\ovar@rightarrow\varrightarrow
    \let\ovar@leftarrow\varleftarrow
\fi

Option \texttt{esvect} \textsuperscript{\textit{P. 11}}.
\ifovar@option@esvect@
    \RequirePackage{esvect}
\fi

Option \texttt{tikz} \textsuperscript{\textit{P. 15}}.
\ifovar@option@tikz@
    \RequirePackage{tikz}
    \usetikzlibrary{arrows.meta}
\fi

Option \texttt{pstarrows} \textsuperscript{\textit{P. 15}}.
\ifovar@option@pstarrows@
    \RequirePackage[pstarrows]{pict2e}
\fi

Configuration of subscripts detection

Sets the subscript command.
\newcommand{\SetOverArrowsSubscriptCommand}[1]{\global\let\ovar@subcmd=#1}

Initial configuration.
\SetOverArrowsSubscriptCommand{_}

Option \texttt{subother} \textsuperscript{\textit{P. 15}} for \textit{other} (catcode 12) subscript commands.
\begin{group}
    \catcode`_=12
    \AddToHook{begindocument/end}{\SetOverArrowsSubscriptCommand{_}}
\end{group}

Option \texttt{subactive} \textsuperscript{\textit{P. 15}} for \textit{active} (catcode 13) subscript commands.
\begin{group}
    \catcode`_=13
    \AddToHook{begindocument/end}{\SetOverArrowsSubscriptCommand{_}}
\end{group}

Management of keys

Family declaration and setters
\RequirePackage{pgfkeys}
\pgfkeys{/overarrows/.is family}
\newcommand{\ovar@set}[1]{\pgfkeys{/overarrows}(#1)}
\NewDocumentCommand{\SetOverArrowsMethod}{ s O{fill} m O{} m }{% 
\IfBooleanTF{#1}{% 
\csgdef{ovar@set@#3}{#4 \ovar@set{% 
\no stack macro hook/.code={% 
\ovar@set(stack macro/.expanded=({% 
\expandafter{\expandonce\csname ovar@stack@#2\endcsname}% 
\expandonce{ovar@length@min}}}% 
\expandonce{ovar@before@arrow}{\expandonce{ovar@after@arrow}}}% 
}%,#5}% 
}% 
\csgdef{ovar@set@#3}{#4 \ovar@set{no stack macro hook/.code={% 
\ovar@set(stack macro/.expanded=({% 
\expandafter{\expandonce\csname ovar@stack@#2\endcsname}% 
\expandonce{ovar@length@min}}}% 
\expandonce{ovar@before@arrow}{\expandonce{ovar@after@arrow}}}% 
}%,#5}}%
}

Common keys

\SetOverArrowsMethod*(common)\undef{ovar@macro@stack}\undef{ovar@macro@arrow}{}% 
detect subscripts \^P. 22.
\detect subscripts/.is if=ovar@detectsubscripts%,

stack macro \^P. 27 and arrow macro \^P. 27.
stack macro/.store in=ovar@macro@stack,
arrow macro/.store in=ovar@macro@arrow,
stack macro/.value required,
arow macro/.value required,

no stack macro hook \^P. 27, no arrow macro hook \^P. 28. These two keys must
be redefined by the command \ovar@set@(method).
no stack macro hook/.code=\PackageError{overarrows}{Undefined stack macro}{
(The requested method is perhaps mispelled)
},
nos arrow macro hook/.code=\PackageError{overarrows}{Undefined arrow macro}{
(The requested method is perhaps mispelled)
},

min length \^P. 20.
min length/.store in=ovar@length@min,
min length/.value required,
min length=0,

before arrow \^P. 22, after arrow \^P. 22, space before arrow \^P. 22, space after
arrow \^P. 22.
before arrow/.store in=ovar@before@arrow,
after arrow/.store in=ovar@after@arrow,
before arrow/.value required,
after arrow/.value required,
before arrow=\empty,
after arrow=\empty,
space before arrow/.code=\pgfkeysalso{before arrow={$\kern #1$}},
space after arrow/.code=\pgfkeysalso{after arrow={$\kern #1$}},

shift left \^P. 21, shift right \^P. 21, shift left right \^P. 21, center arrow \^P. 22,
left arrow \^P. 22, right arrow \^P. 22.
shift left/.store in=ovar@shift@left,
shift right/.store in=ovar@shift@right,
shift left/.value required,
shift right/.value required,
shift leftright/.code=\pgfkeysalso{\pgfmathsetmacro{\shiftleft}{##1}, shift right=##1,},
center arrow/.code=\pgfkeysalso{shift leftright=0},
left arrow/.code=\pgfkeysalso{shift left=##1, shift right=##1,},
right arrow/.code=\pgfkeysalso{shift left=0, shift right=##1,},
left arrow/.default=2,
right arrow/.default=2,

arrow under/.is choice,
arow under/noconfig/.code={
\def\ovar@stack@fill\ovar@stackunder@fill
\def\ovar@stack@lens\ovar@stackunder@lens
},
arow under/autoconfig/.code={
\pgfkeysalso{
arrow under=noconfig,
detect subscripts=false,
before arrow={\kern 1.3\ex\relax},% like underarrow@ from amsmath
}
},
arow under/.default=autoconfig,

Keys for the symb method
\SetOverArrowsMethod{symb}{\undef\ovar@macro@arrowfill}{% Fill macro.
fill macro/.store in=\ovar@macro@arrowfill,
fill macro/.value required,

Arrow macro.
\ifdef\ovar@macro@arrowfill{%
\ovar@set{%
\setmacro\ovar@shift@left{\expandonce\ovar@shift@left}\%}
\ovar@set{%
\setmacro\ovar@shift@right{\expandonce\ovar@shift@right}\%}
%

\ovar@set{%
\setmacro\ovar@arrow@fill{\expandonce\ovar@arrow@fill}\%
\setmacro\ovar@arrow@start{\expandonce\ovar@arrow@start}\%
\setmacro\ovar@trim@start{\expandonce\ovar@trim@start}\%
\setmacro\ovar@arrow@middle{\expandonce\ovar@arrow@middle}\%
\setmacro\ovar@trim@end{\expandonce\ovar@trim@end}\%
\setmacro\ovar@arrow@end{\expandonce\ovar@arrow@end}\%
}
%
start/.P. 23, middle/.P. 23, end/.P. 23 .}
trim start/.code={\def\ovar@trim@start\{\xjoinrel\[##1\]}},
trim middle/.code={\def\ovar@trim@middle\{\xjoinrel\[##1\]}},
trim end/.code={\def\ovar@trim@end\{\xjoinrel\[##1\]}},
trim/.code={\pgfkeysalso{trim start={##1}, trim middle={##1}, trim end={##1}}},
trim/.value required,
nof trimming/.code={\let\ovar@trim@start\empty\let\ovar@trim@middle\empty\let\ovar@trim@end\empty},
nof trimming/.value forbidden,

middle config/.is choice,
middle config/.value required,
middle config/relbar/.code={\pgfkeysalso{\middle={\relbar}, trim middle={2.5},}},
middle config/relbareda/.code={\ifundef{\relbareda}{\PackageWarning{overarrows}{Key 'middle config=relbareda' used, but \protect\relbareda\space is undefined; ignored. \MessageBreak\PackageWarning{overarrows}{Load 'esvect' package, or use 'esvect' option \MessageBreak to remove this warning}}}{\pgfkeysalso{\middle={\relbareda}, trim middle={1},}},
middle config/auto/.code={\ifovar@option@esvect@\pgfkeysalso{middle config=relbareda}\else\pgfkeysalso{middle config=relbar}\fi},

amsmath/.is choice,
amsmath/mimic/.code={\pgfkeysalso{start={\relbar}, middle={\relbar}, end={\rightarrow}, trim start=7, trim middle=2, trim end=7, shift leftright=0},
after arrow={}, before arrow={},
},
\amsmath/strict/.code=\pgfkeysalso{
\amsmath=mimic,
no trimming,
fill macro={\arrowfill@}, stack macro={\overarrow@},
},
\amsmath/.default=mimic,

esvect \textsuperscript{P. 24}.
esvect/.is choice,%
esvect/mimic/.code=\pgfkeysalso{
\start={\relbaredd}, \middle={\relbareda}, \end={\fldr},
trim start=1.5, 
trim end=1.5, 
trim middle=0, 
right arrow=2, 
space before arrow=-.7pt, 
space after arrow=-.3pt,
},
esvect/strict/.code=\pgfkeysalso{
\esvect=mimic, 
no trimming, 
fill macro={\traitfill@}, stack macro={\overvect@},
},
esvect/.default=mimic,
Initial configuration.
\amsmath, middle config=auto, end=\overarrowrightarrow, right arrow,}

Keys for the \textit{tikz} method

\SetOverArrowsMethod\[lens\]{tikz}\[\undef{\ovar@tikz@command}\]{%
Arrow macro.

\no arrow macro hook/.code=\%
\ifdef{\ovar@tikz@command}{\pgfkeysgetvalue{/overarrows/path options}{\ovar@tikz@pathoptions} \ovar@set{\tikz command/.expanded={\noexpand\draw[\expandonce\ovar@tikz@pathoptions]\expandonce\ovar@tikz@path;}}}\%
\pgfkeysgetvalue{/overarrows/tikz options}{\ovar@tikz@options} \ovar@set{\arrow macro/.expanded={\noexpand\draw[\expandonce\ovar@tikz@options]\expandonce\ovar@tikz@command]\noexpand\relax\}%
\noexpand\draw[\expandonce\ovar@tikz@options]{\expandonce\ovar@tikz@command]\noexpand\relax\%}
}

TikZ parts: \textit{tikz command} \textsuperscript{P. 25}, \textit{tikz options} \textsuperscript{P. 25}, \textit{path options} \textsuperscript{P. 25}, \textit{path} \textsuperscript{P. 25}. 

\tikz command/.store in=\ovar@tikz@command, 
\tikz options/.initial={\textbackslash overarrowlength, \textbackslash overarrowthickness}, \path options/.initial={\textbackslash arrows=-\textbackslash Classical \textbackslash TikZ \textbackslash Rightarrow, \textbackslash cap=\textbackslash round}, \path/.store in=\ovar@tikz@path, \path=((0,0)\textasciitilde (1,0)),

36

Initial configuration.

Keys for the picture method

Initial configuration.

Keys for the picture method

Initial configuration.
Commands

Macros for symbols assemblage

\usepackage{overarrows}

\DeclareRobustCommand{\joinrel}{\mathrel{\mkern-#1mu}}

Use a default value of 3.5\,mu, as recommended by egreg (see \url{https://tex.stackexchange.com/a/471736}). \texttt{\joinrel} uses a value of 3\,mu.

\newcommand*{\smallermathstyle}{\mathchoice{\scriptstyle}{\scriptstyle}{\scriptscriptstyle}{}}

\newcommand{\ovar@arrow@fill}{#1#2#3#4#5#6}{\relax\thinmuskip0mu
\medmuskip\thickmuskip
\thickmuskip \thickmuskip \relax\mkern#1\relax#6#3
\leaders$#6#4$\hfill#5\mkern#2\relax}$

Macros for fixed length arrows

Lengths declaration.

\newlength{\overarrowlength}
\newlength{\overarrowthickness}
\newlength{\overarrowsmallerthickness}
\newlength{\ovar@extralength}
\newlength{\ovar@tempdim}

Sets \texttt{\overarrowlength} \textsuperscript{\textit{P.\,18}}.
#1: min length, in math units
#2: math style
#3: content

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

Sets \texttt{\overarrowthickness} \textsuperscript{\textit{P.\,18}} and \texttt{\overarrowsmallerthickness} \textsuperscript{\textit{P.\,18}}.
#1: arrow length
#2: math style

\def\ovar@set@arrowthickness#1{% use rule thickness=\fontdimen 8 font family 3
\if#1displaystyle\fontdimen 8\textfont 3\else\fontdimen 8\scriptfont 3\fi}

38
\overarrowthickness = \fontdimen 8 \textfont 3\%
\overarrowsmallerthickness = \fontdimen 8 \scriptfont 3\%
\else\ifx\scriptstyle\overarrowthickness = \fontdimen 8 \scriptfont 3\%
\overarrowsmallerthickness = \fontdimen 8 \scriptscriptfont 3\%
\else
\overarrowthickness = \fontdimen 8 \scriptscriptfont 3\%
\overarrowsmallerthickness = \overarrowthickness\%
\fi\fi\fi

\overarrow\& \overarrow\& \overarrow\& \overarrow

Stack macros

Bases of all stack macros.

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

\def\ovar@stackover@@ #1#2#3#4#5#6{\vbox{\ialign{##\crcr
$#5$\mskip #1 mu\relax$
\noalign{\nointerlineskip}$#4$
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr
\noalign{\nointerlineskip}$%$
\mskip #1 mu\relax$
\noalign{\nointerlineskip}$#3$
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr}}}%

\def\ovar@stackunder@@ #1#2#3#4#5#6{\vtop{\ialign{##\crcr
$#5$
\noalign{\nointerlineskip}$#4$
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr
\noalign{\nointerlineskip}$%$
\m@th\hfil #5#6\hfil $
\crcr}}}%

\def\ovar@stackover@ #1#2#3#4#5#6{\ovar@stackover@@ {0}{#1}{#2}{#3}{#4}{#5}}
\def\ovar@stackunder@ #1#2#3#4#5#6{\ovar@stackunder@@ {0}{#1}{#2}{#3}{#4}{#5}}

\ovar@stack@fill
\ovar@stackunder@fill
\ovar@stack@fill

Stack macros without min arrow length.

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

\def\ovar@stackover@ #1#2#3#4#5{\ovar@stackover@@ (#1)(#2)(#3)(#4)(#5)}
\def\ovar@stackunder@ #1#2#3#4#5{\ovar@stackunder@@ (#1)(#2)(#3)(#4)(#5)}

\ovar@stack@fill
\ovar@stackunder@fill
\ovar@stack@fill

Stack macros for extensible arrows.

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

\def\ovar@stackover@fill #1#2#3#4#5#6{\ovar@stackover@@ {#1}{#2}{#3}{#4#5}{#5}{#6}}
\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\textit{P. 21}.

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

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

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

\def\ovar@stackover@lens#1#2#3#4#5#6{% 
\ovar@set@arrowlength{#1}{#5}{#6}\
\ovar@set@arrowthickness{#5}\
\ovar@stackover@{#2}{#3}{#4}{#5}{#6}\
}%

\def\ovar@stackunder@lens#1#2#3#4#5#6{% 
\ovar@set@arrowlength{#1}{#5}{#6}\
\ovar@set@arrowthickness{#5}\
\ovar@stackunder@{#2}{#3}{#4}{#5}{#6}\
}%

\ovar@stack@lens matches the macro \ovar@stackover@lens by default, or \ovar@stackunder@lens with arrow \under\textit{P. 21}.

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

\begin{Verbatim}
\texttt{Macro for commands creation}
\end{Verbatim}

\NewDocumentCommand{\DeclareOverArrowCommand}{O{symb} m m}{% 
\begingroup 
\ovar@set@common 
\ifcsdef{ovar@set@#1}{% 
\csuse{ovar@set@#1} 
\PackageError{overarrows}{Unknown method \textit{#1}}{Try with 'symb', 'tikz' or 'picture'} 
}% 
\ovar@set{#3} 
\ifndef{\ovar@macro@arrow}{}{\ovar@set{no arrow macro hook}} 
\ifndef{\ovar@macro@stack}{}{\ovar@set{no stack macro hook}} 
\csxdef{ovar@#2@normal}{% 
\noexpand\mathpalette{% 
\expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}}% 
} 
\csxdef{ovar@#2@starred}{% 
\noexpand\mathpalette{% 
\noexpand\ovar@starversion{% 
\expandonce{\ovar@macro@stack}{\expandonce{\ovar@macro@arrow}}}% 
} 
}\ifndef{\ovar@detectsubscripts}{}{\csxdef{ovar@#2@auto}##1{% 
\expandonce{\textstyle}{\expandonce{\ovar@#2@normal}{##1}} 
} 
\expandonce{\scriptstyle}{\expandonce{\ovar@#2@normal}{##1}} 
\expandonce{\scriptscriptstyle}{\expandonce{\ovar@#2@normal}{##1}} 
}} 
\endgroup}
Starred variant

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

\def\ovar@starversion#1#2#3{% #1#2(#3)\% \settowidth\ovar@extralength{$\math@th@matrix\#1#2#3\matrix$} \settowidth\ovar@tempdim{$\math@th@matrix#2\matrix$} \deflength{\ovar@extralength}{0.5\ovar@extralength-0.5\ovar@tempdim} \kern-\ovar@extralength\}%
\vv vector command

Backup and redefinition of \texttt{esvect} \vv vector command.

\begin{verbatim}
\ifovar@option@esvect@
  \let\esvectvv\vv
  \undef\vv
  \NewOverArrowCommand{\vv}{esvect, middle config=auto}
\fi
\end{verbatim}

Predefined commands

\begin{verbatim}
\overrightarrow
\ifovar@option@overrightarrow@
  \DeclareOverArrowCommand{overrightarrow}{%
    amsmath, middle config=relbar,
    end=\ovar@rightarrow,
    right arrow,
  }%
\fi

\underrightarrow
\ifovar@option@underrightarrow@
  \DeclareOverArrowCommand{underrightarrow}{%
    amsmath, middle config=relbar,
    start=\ovar@rightarrow,
    end=\relbar,
    right arrow,
    arrow under,
  }%
\fi

\overleftarrow
\ifovar@option@overleftarrow@
  \DeclareOverArrowCommand{overleftarrow}{%
    amsmath, middle config=relbar,
    start=\ovar@leftarrow,
    end=\relbar,
    left arrow,
    arrow under,
  }%
\fi

\underleftarrow
\ifovar@option@underleftarrow@
  \DeclareOverArrowCommand{underleftarrow}{%
    amsmath, middle config=relbar,
    start=\ovar@leftarrow,
    end=\relbar,
    left arrow,
    arrow under,
  }%
\fi

\overleftrightarrow
\ifovar@option@overleftrightarrow@
  \DeclareOverArrowCommand{overleftrightarrow}{%
    amsmath, middle config=relbar,
    start=\ovar@leftarrow,
    end=\ovar@rightarrow,
    center arrow,
  }%
\fi

\underleftrightarrow
\ifovar@option@underleftrightarrow@
  \DeclareOverArrowCommand{underleftrightarrow}{%
    amsmath, middle config=relbar,
    start=\ovar@leftarrow,
    end=\ovar@rightarrow,
    center arrow,
    arrow under,
  }%
\fi
\end{verbatim}

42
\overrightharpoonup \fi
\ifovar@option@overrightharpoonup@
\DeclareOverArrowCommand{overrightharpoonup}{% 
  amsmath, middle config=relbar, 
  end=\rightharpoonup, 
  right arrow, 
} \fi
\underrightharpoonup
\ifovar@option@underrightharpoonup@
\DeclareOverArrowCommand{underrightharpoonup}{% 
  amsmath, middle config=relbar, 
  end=\rightharpoonup, 
  right arrow, 
  arrow under, 
} \fi
\overrightharpoondown
\ifovar@option@overrightharpoondown@
\DeclareOverArrowCommand{overrightharpoondown}{% 
  amsmath, middle config=relbar, 
  end=\rightharpoonup, 
  right arrow, 
} \fi
\underrightharpoondown
\ifovar@option@underrightharpoondown@
\DeclareOverArrowCommand{underrightharpoondown}{% 
  amsmath, middle config=relbar, 
  end=\rightharpoonup, 
  right arrow, 
  arrow under, 
} \fi
\overleftharpoonup
\ifovar@option@overleftharpoonup@
\DeclareOverArrowCommand{overleftharpoonup}{% 
  amsmath, middle config=relbar, 
  start=\leftharpoonup, 
  end=\relbar, 
  left arrow, 
} \fi
\underleftharpoonup
\ifovar@option@underleftharpoonup@
\DeclareOverArrowCommand{underleftharpoonup}{% 
  amsmath, middle config=relbar, 
  start=\leftharpoonup, 
  end=\relbar, 
  left arrow, 
  arrow under, 
} \fi
\overleftharpoondown
\ifovar@option@overleftharpoondown@
\DeclareOverArrowCommand{overleftharpoondown}{% 
  amsmath, middle config=relbar, 
  start=\leftharpoonup, 
  end=\relbar, 
  left arrow, 
} \fi
\underleftharpoondown
\ifovar@option@underleftharpoondown@
Test macros

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

\newcommand{\avar@testmathstyles}{2}[]{\begingroup
\newcommand*{\avar@row@teststyle}{1}{% $\displaystyle #1$ & $\textstyle #1$ & $\scriptstyle #1$ & $\scriptscriptstyle #1$ \\}
\renewcommand*{\arraystretch}{1.5}
\begin{tabular}{cccc}
\hline
\footnotesize \texttt{\textbackslash displaystyle}\{##1\} & \footnotesize \texttt{\textbackslash textstyle}\{##1\} & \footnotesize \texttt{\textbackslash scriptstyle}\{##1\} & \footnotesize \texttt{\textbackslash scriptscriptstyle}\{##1\} \\
\hline
\avar@row@teststyle{v} & \avar@row@teststyle{AB} & \avar@row@teststyle{\texttt{grad}} & \avar@row@teststyle{my~long~vector} & \IfValueT{##1}{\avar@row@teststyle{##1}} \\hline
\end{tabular}
\endgroup
}
\begin{displaymath}
\text{#1}_{\text{#1}_{#1(v)}}
\end{displaymath}
\begin{align}
#1_\mathbf{i}_0
v &= #1_\mathbf{v}_x + #1_\mathbf{v}_y + #1_\mathbf{v}_z
&= v_x #1_\mathbf{i} + v_y #1_\mathbf{j} + v_z #1_\mathbf{k}
\end{align}
\endgroup

\ifcsdef{#3}{}{\PackageWarning{overarrows}{Unknown name '#3' passed to \protect\TestOverArrow}}
\IfBooleanTF{#1}{{
\noindent\framebox{%
\centering
{\textbf{\large Test of \texttt{#3} and \texttt{#3*} macros}}
\bigskip
\textbf{\texttt{#3} for different math styles}}
\ovar@testmathstyles[#2]{#3}%
\bigskip
\textbf{\texttt{#3} kerning}}
\ovar@testkerning{\csuse{#3}}
\textbf{\texttt{#3*} kerning}}
\ovar@testkerning{\csuse{#3}*}
\end{minipage}%
}{%}
\begin{minipage}{0.95\linewidth}
\centering
{\textbf{\large Test of \texttt{#3} and \texttt{#3*} macros}}
\bigskip
\textbf{\texttt{#3} for different math styles}}
\ovar@testmathstyles[#2]{#3}%
\bigskip
\textbf{\texttt{#3} kerning}}
\ovar@testkerning{\csuse{#3}}
\textbf{\texttt{#3*} kerning}}
\ovar@testkerning{\csuse{#3}*}
\end{minipage}%
\bigskip
}{%}
\begin{minipage}{\linewidth}
\centering
{\textbf{\large Test of \texttt{#3} and \texttt{#3*} macros}}
\bigskip
\textbf{\texttt{#3} for different math styles}}
\ovar@testmathstyles[#2]{#3}%
\bigskip
\textbf{\texttt{#3} kerning}}
\ovar@testkerning{\csuse{#3}}
\textbf{\texttt{#3*} kerning}}
\ovar@testkerning{\csuse{#3}*}
\end{minipage}
Index

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

Package options

- **allcommands**, 13
- **debug**, 15
- **esvecta**, 11
- **esvectb**, 12
- **esvectc**, 12
- **esvectd**, 12
- **esvecte**, 12
- **esvectf**, 12
- **esvectg**, 12
- **esvecth**, 13
- **noesvect**, 12
- **old-arrows**, 14
- **overbar**, 14
- **overcommands**, 13
- **overleftarrow**, 13
- **overleftharpoonup**, 14
- **overleftarrow**, 13
- **overrightarrow**, 13
- **overrightarrow**, 13
- **underbar**, 14
- **underleftarrow**, 14
- **underleftharpoonup**, 14
- **underrightarrow**, 13
- **underrightarrow**, 14
- **underleftarrow**, 14
- **underrightarrow**, 14
- **underrightarrow**, 14
- **underleftarrow**, 14

- **add path options** key, 25
- **add tikz options** key, 25
- **after arrow** key, 22
- **allcommands** package option, 13
- **amsmath** key, 24
- **arrow macro** key, 27
- **arrow under** key, 21
- **arrows** key, 25
- **before arrow** key, 22

- **center arrow** key, 22

Commands

- \texttt{\DeclareOverArrowCommand}, 15
- \texttt{\esvectv}, 18
- \texttt{\NewOverArrowCommand}, 15, 41
- \texttt{\overbar}, 19, 31, 44
- \texttt{\overleftarrow}, 19, 30, 42
- \texttt{\overleftharpoonup}, 19, 31, 43
- \texttt{\overleftarrow}, 19, 30, 42
- \texttt{\ProvideOverArrowCommand}, 15, 41
- \texttt{\RenewOverArrowCommand}, 15, 41
- \texttt{\SetOverArrowsMethod}, 27, 33, 34, 36, 37
- \texttt{\SetOverArrowsMethod*}, 27
- \texttt{\SetOverArrowsSubscriptCommand}, 27, 32
- \texttt{\smallermathstyle}, 17, 38
- \texttt{\TestOverArrow}, 16, 45
- \texttt{\TestOverArrow*}, 16
- \texttt{\underbar}, 20, 31, 44
- \texttt{\underleftarrow}, 20, 30, 42
- \texttt{\overleftarrow}, 20, 31, 44
- \texttt{\underleftarrow}, 20, 31, 43
- \texttt{\underrightarrow}, 20, 30, 42
- \texttt{\underrightarrow}, 20, 30, 42
- \texttt{\underleftarrow}, 20, 31, 43
- \texttt{\vv}, 18, 42
- \texttt{\vv*}, 18
- \texttt{\xjoinrel}, 17, 35, 38

- **debug** package option, 15
- \texttt{\DeclareOverArrowCommand}, 15
- **detect subscripts** key, 25

- **end** key, 23
- **esvect** key, 24
- **esvect** package option, 11
- **esvect** package option, 12
- **esvectb** package option, 12

46
\texttt{esvectc} package option, 12
\texttt{esvectd} package option, 12
\texttt{esvecte} package option, 12
\texttt{esvectf} package option, 12
\texttt{esvectg} package option, 12
\texttt{esvecth} package option, 13
\texttt{esvectv}, 18

\texttt{fill} macro key, 28

\texttt{geometry} key, 26

Internal macros
\begin{verbatim}
  \ifovar@detectsubscripts@, 30, 40
  \ifovar@option@debug@, 30, 41
  \ifovar@option@esvect@, 30, 32, 35, 42
  \ifovar@option@goldarrows@, 30, 32
  \ifovar@option@overbar@, 30, 44
  \ifovar@option@overleftarrow@, 30, 42
  \ifovar@option@overlefttharpoondown@, 30, 43
  \ifovar@option@overlefttharpoonup@, 30, 43
  \ifovar@option@overleftrightarrow@, 30, 42
  \ifovar@option@overrightarrow@, 30, 42
  \ifovar@option@overrighttharpoondown@, 30, 43
  \ifovar@option@overrighttharpoonup@, 30, 43
  \ifovar@option@pstarrows@, 30, 32
  \ifovar@option@subactive@, 30, 32, 44
  \ifovar@option@subother@, 30, 32, 44
  \ifovar@option@stikz@, 30, 32
  \ifovar@option@stunderbar@, 30, 44
  \ifovar@option@stunderleftarrow@, 30, 42
  \ifovar@option@stunderlefttharpoondown@, 30, 44
  \ifovar@option@stunderlefttharpoonup@, 30, 43
  \ifovar@option@stunderrightarrow@, 30, 42
  \ifovar@option@stunderrightarrow@, 30, 42
  \ifovar@option@stunderrighttharpoondown@, 30, 43
  \ifovar@option@stunderrighttharpoonup@, 30, 43
  \ifovar@option@stunderlefttharpoondown@, 30, 43
  \ifovar@option@stunderlefttharpoonup@, 30, 43
  \ifovar@option@stunderrighttharpoondown@, 30, 43
  \ifovar@option@stunderrighttharpoonup@, 30, 43
\end{verbatim}

\texttt{\ifovar@option@underrightharpoonup@}, 30, 43
\texttt{\ovar@after@arrow}, 33
\texttt{\ovar@arrow@end}, 34, 35
\texttt{\ovar@arrow@fill}, 34, 38
\texttt{\ovar@arrow@middle}, 34, 35
\texttt{\ovar@arrow@start}, 34, 35
\texttt{\ovar@before@arrow}, 33
\texttt{\ovar@extralength}, 38, 41
\texttt{\ovar@leftarrow}, 32, 42
\texttt{\ovar@length@min}, 33
\texttt{\ovar@macro@arrow}, 33, 40
\texttt{\ovar@macro@arrowfill}, 34
\texttt{\ovar@macro@stack}, 33, 40
\texttt{\ovar@picture@command}, 37
\texttt{\ovar@picture@geometry}, 37
\texttt{\ovar@picture@linethickness}, 37
\texttt{\ovar@rightarrow}, 32, 36, 42
\texttt{\ovar@row@teststyle}, 44
\texttt{\ovar@set}, 32–34, 36, 37, 40
\texttt{\ovar@set@}, 33, 40
\texttt{\ovar@set@arrowlength}, 38, 40
\texttt{\ovar@set@arrowthickness}, 38, 40
\texttt{\ovar@set@common}, 40
\texttt{\ovar@shift@left}, 33, 34, 36, 37
\texttt{\ovar@shift@right}, 33, 34, 36, 37
\texttt{\ovar@stack@fill}, 34, 40
\texttt{\ovar@stack@lens}, 34, 40
\texttt{\ovar@stack@over@}, 34, 40
\texttt{\ovar@stack@over@fill}, 39, 40
\texttt{\ovar@stack@over@lens}, 34, 40
\texttt{\ovar@stack@over@over@}, 39, 40
\texttt{\ovar@stack@over@over@fill}, 39, 40
\texttt{\ovar@stack@over@over@lens}, 39, 40
\texttt{\ovar@stack@under@}, 39, 40
\texttt{\ovar@stack@under@over@}, 39, 40
\texttt{\ovar@stack@under@over@fill}, 34, 39
\texttt{\ovar@stack@under@over@lens}, 34, 40
\texttt{\ovar@starversion}, 40, 41
\texttt{\ovar@subcmd}, 32, 42
\texttt{\ovar@tempdim}, 38, 41
\texttt{\ovar@testkerning}, 45
\texttt{\ovar@testmathstyles}, 44, 45
\texttt{\ovar@tikz@command}, 36
\texttt{\ovar@tikz@options}, 36
\texttt{\ovar@tikz@path}, 36
\texttt{\ovar@tikz@pathoptions}, 36
\texttt{\ovar@trim@end}, 34, 35
\texttt{\ovar@trim@middle}, 34, 35
\texttt{\ovar@trim@start}, 34, 35

\texttt{Keys}
\begin{verbatim}
  add path options, 25
  add tikz options, 25
  after arrow, 22
  amsmath, 24
\end{verbatim}

47
arrow macro, 27
arrow under, 21
arrows, 25
before arrow, 22
center arrow, 22
detect subscripts, 22
detect subscripts, 22
end, 23
esvect, 24
fill macro, 28
geometry, 26
left arrow, 22
line thickness, 25, 26
middle, 23
middle config, 24
min length, 20
no arrow macro hook, 28
no stack macro hook, 27
no trimming, 23
path, 25
path options, 25
picture command, 26
right arrow, 22
shift left, 21
shift leftright, 22
shift right, 21
space after arrow, 22
space before arrow, 22
stack macro, 27
start, 23
thinner, 25, 26
tikz command, 25
tikz options, 25
trim, 23
trim end, 23
trim middle, 23
trim start, 23

left arrow key, 22
Lengths
\overarrowlength, 18, 36–38
\overarrowsmallerthickness, 18, 37–39
\overarrowthickness, 18, 36–39
line thickness key, 25, 26

middle key, 23
middle config key, 24
min length key, 20

\NewOverArrowCommand, 15
no arrow macro hook key, 28
no stack macro hook key, 27
no trimming key, 23
noesvect package option, 12

old–arrows package option, 14
\overarrowlength length, 18
\overarrowsmallerthickness length, 18
\overarrowthickness length, 18
\overbar, 19
overbar package option, 14
overcommands package option, 13
\overleftarrow, 19
overleftarrow package option, 13
overlefharpoon down, 19
overlefharpoon down package option, 14
\overlefharpoonup, 19
overlefharpoonup package option, 13
\overleftuparrow, 19
overleftuparrow package option, 13
\overleftrightarrow, 19
overleftrightarrow package option, 13
\overleftarrow, 19
overrightarrow package option, 13
\overrightarrow, 19
overrightharpoon down package option, 13
\overrightharpoonup, 19
overrightharpoonup package option, 13

path key, 25
path options key, 25
picture command key, 26
\ProvideOverArrowCommand, 15
pstarrows package option, 15

\RenewOverArrowCommand, 15
right arrow key, 22

\SetOverArrowsMethod, 27
\SetOverArrowsMethod*, 27
\SetOverArrowsSubscriptCommand, 27

shift left key, 21
shift leftright key, 22
shift right key, 21

\smallermathstyle, 17
space after arrow key, 22
space before arrow key, 22
stack macro key, 27
start key, 23
subactive package option, 15
subother package option, 15
subscripts package option, 15

\TestOverArrow, 16
\TestOverArrow*, 16
thinner key, 25, 26
tikz package option, 15
tikz command key, 25
tikz options key, 25

48
trim key, 23
trim end key, 23
trim middle key, 23
trim start key, 23

\underbar, 20
underbar package option, 14
undercommands package option, 13
\underleftarrow, 20
underleftarrow package option, 14
\underleftharpoondown, 20
underleftharpoondown package option, 14
\underleftarrow, 20
underleftarrow package option, 14
\underleftharpoondown, 20
underleftharpoondown package option, 14
\underleftarrow, 20
underleftarrow package option, 14
\underleftharpoondown, 20
underleftharpoondown package option, 14
\underrightarrow, 19
underrightarrow package option, 14
\underrightharpoondown, 20
underrightharpoondown package option, 14
\underrightarrow, 20
underrightarrow package option, 14
\underleftrightarrow, 20
underrightarrow package option, 14
\underleftrightarrow, 20
underrightarrow package option, 14
\xjoinrel, 17
Change History

v1.0
   General: Initial version ............ 1

v1.0.1
   General: Bug fix for under* options 30

v1.1
   General: Support for non-standard subscripts ............ 32, 40, 44