The mattens package*

Danie Els

2009/09/01

Abstract

The mattens package contains the definitions to typeset vectors and tensors such as $\overrightarrow{\boldsymbol{e}}_i$, $\dot{\boldsymbol{x}}^r$, $\underline{\overline{\boldsymbol{E}}}_r^s$, etc., for the representation of common vectors and tensors such as forces, velocities, moments of inertia, etc. These symbols (and variants thereof) are found on the black boards of engineering schools and in many journals and books, particularly those concerned with dynamics and kinematics.

This package adheres to the well defined and documented notation of Hassenpflug.[1,2]

Keywords: *vector, matrix, tensor, notation.*

*This file has version number v1.3, last revised 2009/09/01.

[1]Hassenpflug, W.C., Matrix Tensor Notation Part I. Rectilinear Orthogonal Coordinates. *Comput. Math. Applic.,* vol. 26, no. 3, 1993, pp. 55–93.

[2]Hassenpflug, W.C., Matrix Tensor Notation Part II. Skew and Curved Coordinates. *Comput. Math. Appl.*, vol. 29, no. 11, 1993, pp. 1–103.

Contents

1 Background

A browse through journals and handbooks, in particular those concerned with dynamics, reveals an amazing array of private notations for vectors and tensors. Every author has his or her own notation, making it very difficult to comprehend what is going on in complex multi reference axes environments.

This package is based on the notation of Hassenpflug[1, 2]. It was developed and refined by Dr. Hassenpflug over many years of teaching engineering, and as stated by him[1]:

> *"It is designed particulary to distinguish between vectors and tensors and their representation as vectors and matrices in different coordinate systems. The main purpose of this notation is that it can be used in the teaching situation, therefore, it conveys all the information explicitly in the symbols, and it can be used in handwriting."*

Hassenpflug[1] identifies the following list of requirements for a good notation for tensor quantities and operations, to which his notation conforms. A notation must:

- be easily written by hand;
- distinguish between vector and scalar quantities;
- distinguish between (second order) tensors and vectors;
- distinguish between physical vectors and their representation by vector arrays, and between physical (second order) tensors and their representation by matrices;
- distinguish between row and column vectors;
- use the same symbol as name for the same vector or tensor in either its physical sense or its representation by a vector array or matrix in different coordinate axes;
- distinguish between matrix/vector representation of the same vector/tensor in different coordinate axes;
- be equally valid in orthonormal and skew coordinate axes;
- indicate all intended operations uniquely;
- be equally valid in all dimensions;
- be equally valid for algebraic vector/matrix algebra which has no connection to any metric space;
- be applicable to differentials;
- allow for defaults to avoid repetitive elaborate symbols, i.e., not all the symbols need to be written down explicitly if it is clear from the context.
- It must be well documented (own addition).

The mattens package was developed to typeset the Hassenpflug matrix tensor symbols in a consistent manner.

An appeal is made to the engineering and scientific community to adopt this notation, because of its consistency and ease of use.

2 Usage of **mattens** package

The mattens package is loaded in the document preamble with:

```
    ⋮
\usepackage[⟨options⟩]{mattens}   % amsmath loaded internally
    ⋮
\usepackage{⟨font packages⟩}      %
\DeclareSymbolFont...             % All the font changes
\SetSymbolFont...                 %
\DeclareSymbolFontAlphabet...     %
    ⋮
\usepackage{bm}       %\boldsymbol redirected to call \bm
    ⋮
```

When mattens is loaded, the amsmath package is loaded automatically, because it is needed for the redefined `\overrightarrow` and `\underrightarrow` commands, as well as the `\boldsymbol` command. It must be loaded before any font packages that redefine some of the amsmath symbols or commands.

On the other hand, the preferred method for obtaining bold italic math symbols is the `\bm` command of the bm package.[3] The bm package must be loaded after all the font packages in order for it to recognize the bold math versions. The bm package reroutes the `\boldsymbol` command to point to `\bm`. If `\boldsymbol` is called after the bm package is loaded, it is equivalent to `\bm`. If the bm package is not loaded, mattens defaults to the `\boldsymbol` command.

The mattens package by default sets bold italics symbols. This choice stems from the ISO standards for typesetting of vectors and tensors. The formatting of symbols then indicates the fact that it is a vector/tensor and the lines, arrows and sub- and superscripts indicate the specific type and reference axes.

The following options are recognized by mattens:

noformat: No symbol formatting is performed, otherwise symbols are set by default in bold italics with the `\boldsymbol` or `\bm` command.

It is important to note that the Hassenpflug requirement of easily written by hand is not fulfilled if the symbols are formatted by anything else than normal math fonts.

mathstrut: A mathstrut is inserted with the symbol to force all the lines and arrows to the same height and depth. The default is no mathstrut.

[3] In the amsmath documentation, $\mathcal{AMS}$ recommends the `\bm` command of the bm package instead of the `\boldsymbol` command for bold italic math symbols. The `\boldsymbol` puts its contents in a box, `\mbox{\boldmath$⟨contents⟩$}`, while `\bm` is a font changing command that uses the appropriate bold math font.

3 List of **mattens** commands

Table 1: List of Matrix Tensor typing commands

Type	Command	Description	Output
Physical column vector	`\aS[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–Symbol*	$\overrightarrow{\boldsymbol{x}}$
Physical row vector	`\Sa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*Symbol–arrow*	$\underrightarrow{\boldsymbol{x}}$
Column vector	`\bS[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–Symbol*	$\overline{\boldsymbol{x}}$
Row vector	`\Sb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*Symbol–bar*	$\underline{\boldsymbol{x}}$
Physical tensor	`\aSa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–Symbol–arrow*	$\underrightarrow{\overrightarrow{\boldsymbol{E}}}$
Tensor (mixed base)	`\aSb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–Symbol–bar*	$\overrightarrow{\underline{\boldsymbol{E}}}$
Tensor (mixed base)	`\bSa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–Symbol–arrow*	$\underrightarrow{\overline{\boldsymbol{E}}}$
Tensor	`\bSb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–Symbol–bar*	$\overline{\underline{\boldsymbol{E}}}$
Cross-product tensor[†]	`\aCSa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–CSymbol–arrow*	$\underrightarrow{\overrightarrow{\underset{\sim}{\boldsymbol{w}}}}$
Cross-product tensor	`\bCSb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–CSymbol–bar*	$\overline{\underline{\boldsymbol{w}}}$

[†] It is defined as the tensor $\overline{\underline{\boldsymbol{a}}}$ associated with the vector $\overline{\boldsymbol{a}}$, where $\overline{\boldsymbol{a}} \times \overline{\boldsymbol{c}} = \overline{\underline{\boldsymbol{a}}} \cdot \overline{\boldsymbol{c}}$

3.1 General syntax

The general syntax of the mattens commands is

`\`□`S`□ `[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	`\`□`CS`□ `[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`
`\`□`S`□ `*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	`\`□`CS`□ `*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`

with command names `\bS`, `\aS`, `\aSa`, `\aSb`, etc. See table 1 for a full list of all the commands. The "starred" form is used to set the symbol in normal math. This can be used for compound tensors or for pre-declared symbols (see bm documentation). An example of the usage is

`\aS{e}`	`\aS*{e}`	$\overrightarrow{\boldsymbol{e}}$	$\overrightarrow{e}$
`\Sb{x}`	`\Sb*{x}`	$\underline{\boldsymbol{x}}$	$\underline{x}$
`\aCSa{z}`	`\aCSa*{z}`	$\underrightarrow{\underset{\sim}{\boldsymbol{z}}}$	$\underrightarrow{\underset{\sim}{z}}$

The optional argument ⟨*accent*⟩ is intended for LaTeX accent commands such as `\dot`, `\ddot`, etc., or their $A_{\mathcal{M}}S$ equivalents, `\Dot`, `\Ddot`.

`\bS[\Ddot]{x}`	$\overline{\ddot{\boldsymbol{x}}}$

The mattens commands look ahead for sub- and superscripts (including primes) in order to place them at the correct horizontal and vertical positions.

`\bS[\Dot]{x}^s_i`	$\overline{\dot{\boldsymbol{x}}}^{s}_{i}$
`\bS[\Ddot]{x}_i^s`	$\overline{\ddot{\boldsymbol{x}}}^{s}_{i}$
`\bS{x}''^s_i`	$\overline{\boldsymbol{x}}^{\prime\prime s}_{i}$
`\bSa{E}^k`	$\underrightarrow{\overline{\boldsymbol{E}}}^{k}$
`\aSb[\Dot]{E}_s`	$\underline{\overrightarrow{\dot{\boldsymbol{E}}}}_{s}$

The commands are also robust and can be used in moving commands such as footnotes[4], headers, etc.

```
\footnote{A vector $\aS{e}_i$ in a footnote}
```

The symbols scale to the appropriate sizes if used in sub- and superscripts. For example, for an integration path parameterized by the vector $\overline{\boldsymbol{r}}^s(\xi)$, the equation for a line integral

```
\begin{equation*}
    \oint\limits_{\bS{r}^s(\xi)}  \dotsi
\end{equation*}
```

gives

$$\oint\limits_{\overline{\boldsymbol{r}}^s(\xi)} \dots$$

4 Symbol formatting commands

4.1 Bold italic symbols

The symbol format can be set with the package options

```
\usepackage{mattens}                % Uses \boldsymbol as default
```

or

```
\usepackage[noformat]{mattens}      % No symbol formatting
```

or anywhere in the document with the command

`\SetSymbFont{`⟨*font-command*⟩`}`

The bm package reroutes the `\boldsymbol` command to point to `\bm`. If `\SetSymbFont{\boldsymbol}` is called after the bm package is loaded, it is equivalent to `\SetSymbFont{\bm}`. Bold formatting of individual tensors can also be switched off with the use of the starred forms of the tensor commands.

If the symbols are interpreted as tensors, then according to the ISO, it can be typeset in a slanted sans serif font (if you are fond of fonts). For the Computer Modern fonts with an OT1 encoding, you can put in the preamble

```
\DeclareMathAlphabet{\mathsfsl}{OT1}{cmss}{m}{sl}
```

Examples of formats are

`\SetSymbFont{\bm}`	$\underline{\overrightarrow{\boldsymbol{E}}}_s$
`\SetSymbFont{\relax}`	$\underline{\overrightarrow{E}}_s$
`\SetSymbFont{\mathsfsl}`	$\underline{\overrightarrow{\mathsfsl{E}}}_s$

[4] A vector $\overrightarrow{\boldsymbol{e}}_i$ in a footnote

Only the first symbol (or group) in multi-symbol constructions is formatted. This can be used to obtain

```
\bS{{}xy} \bS*{xy}
\bS{xy}
\bS{{xy}}
```

$\overline{xy}$ $\overline{xy}$
$\overline{\boldsymbol{xy}}$
$\overline{\boldsymbol{xy}}$

or

```
\bCSb*{\bS{x}+\bS{y}}
\bSb{E_{313}}^s_r
```

$\overbrace{\overline{\boldsymbol{x}+\boldsymbol{y}}}$
$\overline{\underline{\boldsymbol{E}_{313}}^{s}_{r}}$

When a font does not have bold italic symbols and is properly configured, the `\bm` command constructs the symbols with the "poor man's bold" method. This results in the loss of the subscript kerning. This is the case for the mathptm package for Times fonts. If bold italic symbols are needed for Times fonts, it is advisable to use the txfonts package or one of the commercial fonts.

4.2 Struts

A strut can be inserted inside the tensor construction to force all the lines to the same height. This can be given in the package options

```
\usepackage{mattens}                % No strut as default
```

or

```
\usepackage[mathstrut]{mattens}     % Uses \mathstrut
```

or anywhere in the document with the command

`\SetSymbStrut{`⟨*strut*⟩`}`

For example

```
\SetSymbStrut{\relax}
\bSb{E}, \bS{f}, \Sb{y}
```

$\overline{\boldsymbol{E}}$, $\overline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$

```
\SetSymbStrut{\mathstrut}
\bSb{E}, \bS{f}, \Sb{y}
```

$\overline{\boldsymbol{E}}$, $\overline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$

```
\SetSymbStrut{\vphantom{E}}
\bSb{E}, \bS{f}, \Sb{y}
```

$\overline{\boldsymbol{E}}$, $\overline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$

4.3 Additional sub- and superscript spaces

The placing of the sub- and superscripts was fine-tuned for Computer Modern fonts. Other fonts may require the sub- and superscript to shift closer or further away from the lines and the symbols. Additional spaces can be inserted before the sub- and superscripts with the following commands:

`\SetArrowSkip{`⟨*muskip length*⟩`}`
`\SetBarSkip{`⟨*muskip length*⟩`}`
`\SetSymSubSkip{`⟨*muskip length*⟩`}`
`\SetSymSupSkip{`⟨*muskip length*⟩`}`

The length units must be in math units (*mu*), where $18\,mu = 1\,em$ (a little less than the width of the letter "M").

5 Other packages and classes

bm: The bm package is preferred for bold/heavy symbols in math mode. It can also be used to predeclare bold symbols for use with the starred form of the tensor commands, for example:

```
\bmdefine{\bO}{\mathit{\Omega}}
$\bSb*{\bO_i}$
```

$\overline{\underline{\boldsymbol{\Omega}_i}}$

hyperref: When tensor symbols are set in chapter and section headers, hyperref crashes if the `\texorpdfstring` command is not used.

```
\section{A header with \texorpdfstring{$\bS{x}^i_j$}{xij} in it}
```

color: To change the colour of a symbol the `\color` command must be grouped two levels deep to survive all the expansions if the bm package is loaded.

```
$\aSb{{{\color{red}E}}}_i$
```

$\overrightarrow{\underline{\boldsymbol{E}}}_i$

slide, foils: The Euler math fonts are definitely compulsory. The ratio of the sub- and superscript sizes to normal math font sizes is too big inside the slide class. For tensors such as $\overline{\boldsymbol{x}}^j_k$, you may find that the sub- and superscripts overlap. Choose your script characters carefully.

eulervm: The Hassenpflug ideal of symbols that are easily written by hand is satisfied with the beautiful letters of the Euler fonts. This package is a must for presentation material, but *please do not use bold Euler symbols*, they just do not look right!

accents: For the creation of alternative accents the mattens package is fully compatible with the accents package. As an example of its usage, the equation in Hassenpflug[1], §10.1, p.82

```
\SetSymbFont{\relax}
\SetSymbStrut{\mathstrut}
\newcommand{\dotr}[1]{%
    \accentset{\phantom{r}{\displaystyle.}r}{#1}}
\begin{equation*}
    \text{apparent velocity}
            = \frac{\partial_r}{\mathrm{d}t} \aS{r}
            = \aS[\dotr]{r}
            = \aSb{E}_s \tdot \bS[\dotr]{r}^s
       \equiv \aS{v}_{\mathrm{app}}
            = \aS{v}_{\mathrm{rel}}
\end{equation*}
```

which gives

$$\text{apparent velocity} = \frac{\partial_r}{\mathrm{d}t}\overrightarrow{r} = \overrightarrow{\dot{r}} = \overrightarrow{\underline{E}}_s \cdot \overrightarrow{\dot{r}}^s \equiv \overrightarrow{v}_{\mathrm{app}} = \overrightarrow{v}_{\mathrm{rel}}$$

6 Typesetting the Hassenpflug notation in LaTeX

6.1 Why?

The Hassenpflug notation contains symbols such as, $\overrightarrow{\boldsymbol{e}}_i$, $\dot{\overline{\boldsymbol{x}}}^r_\alpha$, $\underrightarrow{\dot{\boldsymbol{E}}}^s$, etc. These symbols are quite common and variants thereof are found on many blackboards of engineering schools. Based on the reputation of TeX it would seem trivial to typeset them, but to the contrary ...

`\bS[\dot]{f}^a_b`	$\dot{\overline{f}}^a_b$ (correct typesetting)
`$\dot{\overline{f}}^a_b$`	$\dot{\overline{f}}^a_b$
`$\dot{\overline{f}}{}^a_b$`	$\dot{\overline{f}}^a_b$
`$\dot{\overline{f}^a_b}$`	$\dot{\overline{f}^a_b}$

When a subscript is inserted for slanted characters, e.g., f^a_b, the subscript is kerned closer to the symbol. This subscript kerning as well as the superscript height is retained for math accents, e.g., $\dot{f}^a_b$. When the `\overline` or `\overrightarrow` commands are used together with math accents, the problems begin[5]:

- For an `\overline`, $\overline{f}^a_b$, the superscript moves up to the height of the overline, which is fine for vector notation, but the subscript kerning is lost, because the symbol is set inside a box with italic correction added (App. G of the TeXbook). If the subscript is added inside the `\overline` construct, $\overline{f_b}^a$, the overline is stretched, which is acceptable only when the subscript is part of the symbol itself. The subscript kerning can be corrected by attaching the subscript to the symbol and the superscript to the overline and then overtyping (`\rlap` or `\llap`) them while using a phantom symbol (without subscript) for the overline,

  ```
  \leavevmode\rlap{$\overline{\phantom{f}}^a$}%
                  {$f^{}_b$}
  ```

  which gives $\overline{f}^a_b$.

  The `\leavevmode` command is nessesary to prevent the symbol being spread over two lines when it is the first token in a new paragraph (TeX in vertical mode).

  Care must be taken with the sequence in which the two parts overlap when the one is wider than the other, $\overline{f}^{abcde}_a$, because the superscript will then run into the following text.

- If a math accent is added to the overline, $\dot{\overline{f}}^a_b$, the superscript goes to the level of the accent (except if the accents package is loaded). This can be remedied by controlling the height of the accented overline construction with a `\vphantom` commands as a strut,

[5] If you are using Equation Editor or MathType inside word processors in the Windows environment, you end up with the same problems and have to fiddle *every* sub- and superscript individually. A browse through the accompanying sample document, mtsample.tex, will show that this can amount to hundreds of little equations. I challenge anyone to rewrite that document with one of the popular wordprocessors and still be sane afterwards :-)

```
\leavevmode\rlap%
   {$\dot{\overline{\phantom{f}}}{\vphantom{\overline{f}}}^a$}%
   {$f^{}_b$}
```

which gives $\dot{\overline{f}}{}^a_b$.

- For an `\overrightarrow` in vector notation, ($\overrightarrow{i}_a$, $\overrightarrow{iiii}_a$), the construction of the stretchable arrow results in an arrow wider than most symbols, with the symbol centred below it. The subscript ended up far removed from the symbol. For multiple symbols, the arrow width is equal to the box containing the letters, resulting in better-looking subscripts (but still without the subscript kerning). The previous procedure can be used to correct the subscript kerning, except that additional white space must be inserted before the symbol to centre it beneath the arrow.

```
\newlength{\Wspace}\settowidth{\Wspace}{$\overrightarrow{i}$}
\newlength{\SymWdth}\settowidth{\SymWdth}{$i$}
\addtolength{\Wspace}{-\SymWdth}
\leavevmode\rlap{$\dot{\overrightarrow{\phantom{i}}}$}%
                {\hspace*{.5\Wspace}$i^{}_b$}
```

which gives $\dot{\overrightarrow{i}}_b$.

The commands can be optimized by using saveboxes, *but* the symbols will not scale correctly when used as sub- or superscripts or in the `\tfrac` macro. A solution is to nest the commands inside a `\mathpalette` environment, but be warned, this is very expensive! The commands `\phantom`, `\smash`, `\overrightarrow` (in amsmath) are already inside `\mathpalette` macros, and TEX calls the arguments of a `\mathpalette` macro four times before deciding on the dimensions.

- The shape of the arrow tip of the `\overrightarrow` command was probably not designed for this type of application and is much too broad in the final CM font version. This broad arrow shape is incidentally one of last changes by Prof. Knuth to the CM font symbols. The PostScript version of the CM fonts typesets the arrow much better, but it is highly likely that it is still the old outdated version of the symbol.

- The subscript heights differ for the cases with and without superscripts, x_b, x^a_b. This is not very satisfactory for a consistent matrix-tensor notation, but it can be remedied by inserting a dummy superscript, `$x^{}_b$`, which then gives x_b, x^a_b.

The mattens package was developed to overcome (some of) the above-mentioned problems and to typeset the Hassenpflug matrix tensor symbols in a consistent manner.

6.2 To do's

- The vertical spacing between the symbols and the lines and arrows differs, $\overrightarrow{\underline{\boldsymbol{E}}}$, $\underrightarrow{\overline{\boldsymbol{E}}}$. This problem cannot be fixed easily and would need some additional struts or even a rewriting of the arrows and lines commands.

- For the purists: The ends of the `\overrightarrow` are rounded (ligature of symbols), while the ends of the `\overline` are squared (TEX line drawing).

References

[1] Hassenpflug, W. C., "Matrix Tensor Notation Part I. Rectilinear Orthogonal Coordinates," *Comput. Math. Appl.*, **26**(3), 1993, pp. 55–93.

[2] Hassenpflug, W. C., "Matrix Tensor Notation Part II. Skew and Curved Coordinates," *Comput. Math. Appl.*, **29**(11), 1993, pp. 1–103.

7 The Code: mattens.sty

```
⟨*package⟩
```

7.1 Identification

```
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{mattens}[2009/09/01
                          v1.3
                          Matrix/Tensors (DNJ Els)]
```

7.2 Options

\MT@SymbStrt
\SetSymbStrut

Struts to set all the lines and arrows at predetermined heights and depths.

```
\newcommand*{\MT@SymbStrt}{}
\newcommand*{\SetSymbStrut}[1]{\renewcommand*{\MT@SymbStrt}{#1}}
\SetSymbStrut{\relax}
\DeclareOption{mathstrut}{\SetSymbStrut{\mathstrut}}
```

\MT@SymbFnt
\SetSymbFont

Initialize the symbol font formatting commands.

```
\newcommand*{\MT@SymbFnt}{}
\newcommand*{\SetSymbFont}[1]{\renewcommand*{\MT@SymbFnt}{#1}}
\SetSymbFont{\boldsymbol}
\DeclareOption{noformat}{\SetSymbFont{\relax}}
```

Process the options

```
\DeclareOption*{%
    \PackageWarning{mattens}{Unknown option \CurrentOption}}
\ProcessOptions\relax
```

7.3 Packages

The amsmath package is loaded to provide the scalable `\overrightarrow` and `\underrightarrow` commands, as well as the `\boldsymbol` command for setting bold math symbols.

```
\RequirePackage{amsmath}
```

7.4 Workaround commands

\MT@Overarrow
\MT@Underarrow

We define over- and under-arrows that bypass the `\mathpalette` part of the amsmath macros `\overrightarrow` and `\underrightarrow`. It uses the amsmath internal macros `\overarrow@`, `\underarrow@` and `\rightarrowfill@`. The first parameter `#1` consists of math styles `\displaystyle`, `\textstyle`, etc. The second parameter `#2` is the symbol or character.

```
\newcommand{\MT@Overarrow}[2]{\overarrow@{\rightarrowfill@}{#1}{#2}}
\newcommand{\MT@Underarrow}[2]{\underarrow@{\rightarrowfill@}{#1}{#2}}
```

\MT@Overline
\MT@Underline

Make over- and underlines with the same calling syntax as the arrows.

```
\newcommand{\MT@Overline}[2]{{#1\overline{#2}}}
\newcommand{\MT@Underline}[2]{{#1\underline{#2}}}
```

\xusebox The \usebox command does not function properly when the pdftex.def driver is loaded, because pdftex does not implement a colour stack such as in the dvips driver, but simulate it at TEX macro level. The \xusebox is a workaround where the \usebox command is grouped.[6] A \mathord is added around the box to regain its height in the pdftex case.

```
\AtBeginDocument{%
   \@ifl@aded{def}{pdftex}%
      {\newcommand*{\xusebox}[1]{\mathord{{\usebox{#1}}}}}%
      {\let\xusebox\usebox}%
}
```

7.5 Initialize

Define skip lengths for insertion in front of sub- and superscripts.

```
\newmuskip{\MT@Askip}
\newmuskip{\MT@Bskip}
\newmuskip{\MT@SPskip}
\newmuskip{\MT@SBskip}
```

\SetArrowSkip \SetBarSkip \SetSymSupSkip \SetSymSubSkip Define commands to set or change the skip lengths and set initial values.

```
\newcommand*{\SetArrowSkip}[1]{\MT@Askip#1}
\newcommand*{\SetBarSkip}[1]{\MT@Bskip#1}
\newcommand*{\SetSymSupSkip}[1]{\MT@SPskip#1}
\newcommand*{\SetSymSubSkip}[1]{\MT@SBskip#1}
\SetArrowSkip{0mu}
\SetBarSkip{1mu}
\SetSymSubSkip{0mu}
\SetSymSupSkip{0mu}
```

\MT@SubSkip \MT@SupSkip Define math skip lengths to insert in front of the sub- and superscripts. The values are set inside the main mattens commands according to the type of symbol.

```
\newmuskip\MT@SubSkip
\newmuskip\MT@SupSkip
```

7.6 Main mattens commands

Setup command templates and lengths to function as global variables and pointers.

\MT@accent The \MT@accent command points to the math accent that are inserted as the optional argument inside the main mattens commands.

```
\newcommand*{\MT@accent}{}
```

\MT@cmd \MT@@cmd The commands \MT@cmd and \MT@cmd do the actual typesetting of the symbols.

```
\newcommand*{\MT@cmd}{}
\newcommand*{\MT@@cmd}{}
```

[6] Thanks to Heiko Oberdiek for this workaround

They can be seen as function pointer that are set with `\let` commands inside the main mattens commands to point to specific commands.

Cmd	*Primary command*	*Secondary command*
`\aS:`	`\MT@cmd` $\mapsto$ `\MT@OverAandB`,	`\MT@@cmd` $\mapsto$ `\MT@Overarrow`
`\bS:`	`\MT@cmd` $\mapsto$ `\MT@OverAandB`,	`\MT@@cmd` $\mapsto$ `\MT@Overline`
`\Sa:`	`\MT@cmd` $\mapsto$ `\MT@UnderAandB`,	`\MT@@cmd` $\mapsto$ `\MT@Underarrow`
`\Sb:`	`\MT@cmd` $\mapsto$ `\MT@UnderAandB`,	`\MT@@cmd` $\mapsto$ `\MT@Underline`
`\bSb:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@bSb`
`\aSb:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@aSb`
`\bSa:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@bSa`
`\aSa:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@aSa`
`\bCSb:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@bCSb`
`\aCSa:`	`\MT@cmd` $\mapsto$ `\MT@DoubleAandB`,	`\MT@@cmd` $\mapsto$ `\MT@@aCSa`

`\MT@bold` The `\MT@bold` command is used internally and set by the "starred" command option.

```
\newcommand*{\MT@bold}{}
```

`\aS` Type the tensor: $\overrightarrow{\boldsymbol{x}}$

```
\DeclareRobustCommand*{\aS}{%
    \let\MT@cmd=\MT@OverAandB%
    \let\MT@@cmd=\MT@Overarrow%
    \MT@SupSkip=\MT@Askip%
    \MT@SubSkip=\MT@SBskip%
    \MT@Tensor}
```

`\bS` Type the tensor: $\overline{\boldsymbol{x}}$

```
\DeclareRobustCommand*{\bS}{%
    \let\MT@cmd=\MT@OverAandB%
    \let\MT@@cmd=\MT@Overline%
    \MT@SupSkip=\MT@Bskip%
    \MT@SubSkip=\MT@SBskip%
    \MT@Tensor}%
```

`\Sa` Type the tensor: $\underrightarrow{\boldsymbol{x}}$

```
\DeclareRobustCommand*{\Sa}{%
    \let\MT@cmd=\MT@UnderAandB%
    \let\MT@@cmd=\MT@Underarrow%
    \MT@SupSkip=\MT@SPskip%
    \MT@SubSkip=\MT@Askip%
    \MT@Tensor}
```

`\Sb` Type the tensor: $\underline{\boldsymbol{x}}$

```
\DeclareRobustCommand*{\Sb}{%
    \let\MT@cmd=\MT@UnderAandB%
    \let\MT@@cmd=\MT@Underline%
    \MT@SupSkip=\MT@SPskip%
    \MT@SubSkip=\MT@Bskip%
    \MT@Tensor}
```

`\bSb` Type the tensor: $\overline{\underline{\boldsymbol{E}}}$

```
\DeclareRobustCommand*{\bSb}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@bSb%
    \MT@SupSkip=\MT@Bskip%
    \MT@SubSkip=\MT@Bskip%
    \MT@Tensor}
```

`\aSb` Type the tensor: $\overrightarrow{\underline{\boldsymbol{E}}}$

```
\DeclareRobustCommand*{\aSb}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@aSb%
    \MT@SupSkip=\MT@Askip%
    \MT@SubSkip=\MT@Bskip%
    \MT@Tensor}
```

`\bSa` Type the tensor: $\overline{\underrightarrow{\boldsymbol{E}}}$

```
\DeclareRobustCommand*{\bSa}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@bSa%
    \MT@SupSkip=\MT@Bskip%
    \MT@SubSkip=\MT@Askip%
    \MT@Tensor}
```

`\aSa` Type the tensor: $\overrightarrow{\underrightarrow{\boldsymbol{E}}}$

```
\DeclareRobustCommand*{\aSa}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@aSa%
    \MT@SupSkip=\MT@Askip%
    \MT@SubSkip=\MT@Askip%
    \MT@Tensor}
```

`\bCSb` Type the tensor: $\overline{\underset{\sim}{\boldsymbol{\omega}}}$

```
\DeclareRobustCommand*{\bCSb}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@bCSb%
    \MT@SupSkip=\MT@Bskip%
    \MT@SubSkip=\MT@Bskip%
    \MT@Tensor}
```

`\aCSa` Type the tensor: $\overrightarrow{\underset{\sim}{\boldsymbol{\omega}}}$

```
\DeclareRobustCommand*{\aCSa}{%
    \let\MT@cmd=\MT@DoubleAandB%
    \let\MT@@cmd=\MT@@aCSa%
    \MT@SupSkip=\MT@Askip%
    \MT@SubSkip=\MT@Askip%
    \MT@Tensor}
```

`\MT@Tensor` `\MT@@Tensor` General tensor commands to look for starred form and and initiate script extraction.

```
\newcommand*{\MT@Tensor}{%
   \@ifstar{\let\MT@bold=\@firstofone\MT@@Tensor}
           {\let\MT@bold=\MT@SymbFnt\MT@@Tensor}}

\newcommand*{\MT@@Tensor}[2][\@firstofone]{%
   \let\MT@accent=#1\relax%
   \MT@GetScripts{#2}}
```

7.7 Sub- and superscripts

`\MT@GetScripts` This part of the code looks ahead for sub- and superscripts.

```
\newcommand*\MT@GetScripts[1]{%
   \@ifnextchar'%
      {\MT@GetPrimes{#1}{\prime}}%
      {\MT@UnprimedScripts{#1}}}
```

`\MT@GetPrimes` `\MT@GetPrimedSuper` `\MT@GetPrimedSub` Extract primes and look ahead for superscripts `^`. Note that the sequence of operators for a primed symbol is: $\langle symb\rangle$`'`$\cdot$`'^`$\langle sup\rangle$`_`$\langle sub\rangle$

```
\newcommand*\MT@GetPrimes[3]{%
   \@ifnextchar'%
      {\MT@GetPrimes{#1}{#2\prime}}%
      {\@ifnextchar^%
          {\MT@GetPrimedSuper{#1}{#2}}%
          {\@ifnextchar_%
              {\MT@GetPrimedSub{#1}{#2}}%
              {\MT@SetScripts{#1}{#2}{\@empty}}%
          }%
      }%
   }

\def\MT@GetPrimedSuper#1#2^#3{%
   \@ifnextchar_{\MT@GetPrimedSub{#1}{#2#3}}%
                {\MT@SetScripts{#1}{#2#3}{\@empty}}}

\def\MT@GetPrimedSub#1#2_#3{%
   \MT@SetScripts{#1}{#2}{#3}}
```

`\MT@UnprimedScripts` The first extraction command for symbols without primes. It looks ahead for `^` or `_`, if not present, then pass `\@empty` flags forward, otherwise it passes the tokens on to the next extraction commands.

```
\newcommand*\MT@UnprimedScripts[1]{%
   \@ifnextchar^%
      {\MT@GetSuper{#1}}%
         {\@ifnextchar_%
             {\MT@GetSub{#1}}%
             {\MT@SetScripts{#1}{\@empty}{\@empty}}%
         }%
   }
```

`\MT@GetSuper` Extract scripts of the form $\langle Sym\rangle$`^`$\langle sup\rangle$ if there are no further `_` tokens available, otherwise pass the tokens on to the next extraction command.

```
\def\MT@GetSuper#1^#2{%
   \@ifnextchar_{\MT@GetSuperSub{#1}{#2}}%
                {\MT@SetScripts{#1}{#2}{\@empty}}}
```

`\MT@GetSub` Extract scripts of the form ⟨*Sym*⟩`_`⟨*sub*⟩ if there are no further `^` tokens available, otherwise pass the tokens on to the next extraction command.

```
\def\MT@GetSub#1_#2{%
   \@ifnextchar^{\MT@GetSubSuper{#1}{#2}}%
                {\MT@SetScripts{#1}{\@empty}{#2}}}
```

`\MT@GetSuperSub` Extract scripts of the form ⟨*Sym*⟩`^`⟨*sup*⟩`_`⟨*sub*⟩.

```
\def\MT@GetSuperSub#1#2_#3{%
   \MT@SetScripts{#1}{#2}{#3}}
```

`\MT@GetSubSuper` Extract scripts of the form ⟨*Sym*⟩`_`⟨*sub*⟩`^`⟨*sup*⟩.

```
\def\MT@GetSubSuper#1#2^#3{%
   \MT@SetScripts{#1}{#3}{#2}}
```

`\MT@SetSup` Define global variables commands that contain the extracted sub- and superscripts.
`\MT@SetSub` These commands are redefined inside the the final `\MT@SetScripts` command.

```
\newcommand*{\MT@SetSup}{}
\newcommand*{\MT@SetSub}{}
\newcommand*{\MT@SetSubS}{}
```

```
\newcommand*\MT@SetScripts[3]{%
  \let\MT@SetSup\relax%
  \let\MT@SetSub\relax%
  \let\MT@SetSubS\relax%
  \ifx\@empty#2\@empty\else
     \def\MT@SetSup{^{\mskip\MT@SupSkip\relax#2}}%
  \fi
  \ifx\@empty#3\@empty\else
     \def\MT@SetSub{_{\mskip\MT@SubSkip\relax#3}}%
     \def\MT@SetSubS{^{}_{\mskip\MT@SubSkip\relax#3}}%
  \fi
  \MT@cmd{#1}}
```

7.8 Symbol formatting

`\MT@Symb` This command type `{`⟨*strut*⟩⟨*font-cmd*⟩⟨*symb*⟩`}`, the formatted symbol preceded by the selected strut. It is called from within the main typesetting commands.

```
\newcommand*{\MT@Symb}[1]{\MT@SymbStrt\MT@bold#1}
```

`\MT@SymbC` Puts a widetilde over the symbol for the cross product tensors.

```
\newcommand*{\MT@SymbC}[1]{%
   \MT@SymbStrt\widetilde{%
   \MT@bold#1}}
```

7.9 Main typesetting commands

The commands in this section are the ones pointed to by the `\MT@cmd` and `\MT@@cmd` commands to perform the typesetting of the full tensor symbol.

Declare some save boxes

```
\newsavebox{\MT@Abox}   % for overline/arrow
\newsavebox{\MT@Sbox}   % for symbol
\newsavebox{\MT@Tbox}   % for tempories
\newsavebox{\MT@APbox}  % for overline/arrow phantom
\newsavebox{\MT@SPbox}  % for symbol phantom
```

and some lengths.

```
\newlength{\MT@SPwdth}  % symbol width
\newlength{\MT@BPwdth}  % Bar width
\newlength{\MT@Wwdth}   % leading whitespace width
```

`\MT@OverAandB` This command generates the tensor symbols $\overrightarrow{e}$ and $\overline{e}$. It utilizes the `\mathpalette` macro for the sizing of the final tensor. LaTeX commands `\smash` and `\phantom` with embedded `\mathpalette` calls are avoided to prevent nested `\mathchoice` calls.

```
\newcommand*{\MT@OverAandB}{%
   \mathpalette\MT@@OverAandB}
```

`\MT@@OverAandB` For this command the first parameter `#1` is supplied by `\mathpalette` and consists of math styles `\displaystyle`, `\textstyle`, etc. The second parameter `#2` is the original ⟨*symbol*⟩ from the call `\MT@cmd{`⟨*symbol*⟩`}`.

```
\newcommand*{\MT@@OverAandB}[2]{%
```

Set the symbol inside a box for measurement purposes.

```
    \sbox{\MT@Tbox}{$\m@th#1\MT@Symb{#2}$}%
```

Make phantom symbol box (empty) with size identical to symbol.

```
    \setbox\MT@SPbox\null%
    \ht\MT@SPbox\ht\MT@Tbox%
    \dp\MT@SPbox\dp\MT@Tbox%
    \wd\MT@SPbox\wd\MT@Tbox%
```

Make overline/overarrow over phantom symbol box for measurement.

```
    \sbox{\MT@APbox}{$\m@th\MT@@cmd{#1}{\copy\MT@SPbox}$}%
```

Calculate width difference between symbol and arrow/overline.

```
    \setlength{\MT@Wwdth}{\the\wd\MT@APbox}%
    \addtolength{\MT@Wwdth}{-\the\wd\MT@SPbox}%
```

Make final symbol box with white space in front to center is beneath the arrow/over line and subscript that follows.

```
    \sbox{\MT@Sbox}{$\m@th#1\hskip 0.5\MT@Wwdth\relax\MT@Symb{#2}\MT@SetSubS$}%
```

Add math accent to overline/overarrow and reset box dimensions to original.

```
      \sbox{\MT@Tbox}{$\m@th#1\MT@accent{\xusebox{\MT@APbox}}$}%
      \ht\MT@Tbox\ht\MT@APbox%
      \dp\MT@Tbox\dp\MT@APbox%
```

Final overline/overarrow box including accent and superscript at end.

```
      \sbox{\MT@Abox}{$\m@th#1\xusebox{\MT@Tbox}\MT@SetSup$}%
```

Overtype the symbol and the overline/overarrow boxes. The wider box of the two is typed last to ensure that the spacing after the full tensor symbol is correct.

```
      \ifdim\wd\MT@Abox<\wd\MT@Sbox%
          \leavevmode\rlap{\usebox\MT@Abox}{\usebox\MT@Sbox}%
      \else%
          \leavevmode\rlap{\usebox\MT@Sbox}{\usebox\MT@Abox}%
      \fi}
```

`\MT@UnderAandB` `\MT@@UnderAandB`

This command generates the tensor symbols $\underrightarrow{e}$ and $\underline{e}$. It is identical to the previous command except the sub- and superscripts are swapped and the phantom box is set to the width of the accent.

```
\newcommand*{\MT@UnderAandB}{%
   \mathpalette\MT@@UnderAandB}

\newcommand*{\MT@@UnderAandB}[2]{%
      \sbox{\MT@Tbox}{$\m@th#1\MT@accent{\MT@Symb{#2}}$}%
      \setbox\MT@SPbox\null%
      \wd\MT@SPbox\wd\MT@Tbox%
      \sbox{\MT@Tbox}{$\m@th#1\MT@Symb{#2}$}%
      \ht\MT@SPbox\ht\MT@Tbox%
      \dp\MT@SPbox\dp\MT@Tbox%
      \sbox{\MT@APbox}{$\m@th\MT@@cmd{#1}{\copy\MT@SPbox}$}%
      \setlength{\MT@Wwdth}{\the\wd\MT@APbox}%
      \addtolength{\MT@Wwdth}{-\the\wd\MT@SPbox}%
      \sbox{\MT@Sbox}{$\m@th#1%
         \hskip 0.5\MT@Wwdth\relax\MT@accent{\MT@Symb{#2}}\MT@SetSup$}%
      \sbox{\MT@Abox}{$\m@th#1\xusebox{\MT@APbox}\MT@SetSub$}%
      \ifdim\wd\MT@Abox<\wd\MT@Sbox%
         \leavevmode\rlap{\usebox\MT@Abox}{\usebox\MT@Sbox}%
      \else%
         \leavevmode\rlap{\usebox\MT@Sbox}{\usebox\MT@Abox}%
      \fi}
```

`\MT@@bSb` `\MT@@bSa` `\MT@@aSb` `\MT@@aSa` `\MT@@bCSb` `\MT@@aCSa`

These commands are pointed to by `\MT@@cmd` and called from within `\MT@DoubleAandB`. It has the same calling syntax as the `\MT@Overarrow` and `\MT@Underarrow` commands.

```
\newcommand*{\MT@@bSb}[2]{\MT@Overline{#1}{\MT@Underline{#1}{\MT@Symb{#2}}}}
\newcommand*{\MT@@aSb}[2]{\MT@Overarrow{#1}{\MT@Underline{#1}{\MT@Symb{#2}}}}
\newcommand*{\MT@@bSa}[2]{\MT@Overline{#1}{\MT@Underarrow{#1}{\MT@Symb{#2}}}}
\newcommand*{\MT@@aSa}[2]{\MT@Overarrow{#1}{\MT@Underarrow{#1}{\MT@Symb{#2}}}}
\newcommand*{\MT@@bCSb}[2]{\MT@Overline{#1}{\MT@Underline{#1}{\MT@SymbC{#2}}}}
\newcommand*{\MT@@aCSa}[2]{\MT@Overarrow{#1}{\MT@Underarrow{#1}{\MT@SymbC{#2}}}}
```

\MT@DoubleAandB
\MT@@DoubleAandB

This command is used for the remaining tensor symbols and is not so complex compared to the previous commands.

```
\newcommand*{\MT@DoubleAandB}{%
  \mathpalette\MT@@DoubleAandB}

\newcommand*{\MT@@DoubleAandB}[2]{%
      \sbox{\MT@Abox}{$\m@th#1\MT@@cmd{#1}{#2}$}%
      \sbox{\MT@Tbox}{$\m@th#1\MT@accent{\xusebox{\MT@Abox}}$}%
      \ht\MT@Tbox\ht\MT@Abox%
      \dp\MT@Tbox\dp\MT@Abox%
      \xusebox{\MT@Tbox}\MT@SetSup\MT@SetSub}
```

⟨/package⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

Change History