The mattens package*

Danie Els

2022/03/01

1 Introduction

1.1 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.

The mattens package contains the definitions to typeset vectors and tensors such as $\overrightarrow{\boldsymbol{e}}_i$, $\dot{\overline{\boldsymbol{x}}}^r$, $\underline{\overline{\boldsymbol{E}}}_r^s$, etc., for the representation of common vectors and tensors such as forces, velocities, moments of inertia, etc. It is based on the well defined notation of Hassenpflug[1, 2]. It was developed and over many years of teaching engineering:

> *"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;

*This file has version number v1.3b, last revised 2022/03/01.

- 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.

1.2 Why the mattens package?

The Hassenpflug notation contains symbols such as, $\dot{\overrightarrow{\boldsymbol{e}}}_i$, $\dot{\overline{\boldsymbol{x}}}^r_\alpha$, $\dot{\overline{\underrightarrow{\boldsymbol{E}}}}^s$, 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)
`$\dot{\overline{f}}^a_b$`	$\dot{\overline{f}}^a_b$	
`$\dot{\overline{f}}{}^a_b$`	$\dot{\overline{f}}^a_b$	
`$\dot{\overline{f}^a_b}$`	$\dot{\overline{f}^a_b}$	

2 Usage of mattens package

The mattens package is loaded in the document preamble with:

```
\usepackage[⟨options⟩]{mattens}
```

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.

$\mathcal{AMS}$ recommends the bm package instead of the `\boldsymbol` command for bold italic math symbols[1]. 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.

[1] 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.

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` 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 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*	$\overrightarrow{\underrightarrow{\boldsymbol{E}}}$
Tensor (mixed base)	`\aSb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–Symbol–bar*	$\overrightarrow{\underline{\boldsymbol{E}}}$
Tensor (mixed base)	`\bSa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–Symbol–arrow*	$\overline{\underrightarrow{\boldsymbol{E}}}$
Tensor	`\bSb[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*bar–Symbol–bar*	$\overline{\underline{\boldsymbol{E}}}$
Cross-product tensor†	`\aCSa[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	*arrow–CSymbol–arrow*	$\overrightarrow{\underrightarrow{\boldsymbol{w}}}$
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*⟩`}`	`\`□`S*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`
`\S`□ `[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	`\S`□`*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`
`\`□`S`□ `[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	`\`□`S`□`*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`
`\`□`CS`□ `[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`	`\`□`CS`□`*[`⟨*accent*⟩`]{`⟨*Symbol*⟩`}`

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}`	$\overrightarrow{\underrightarrow{\boldsymbol{z}}}$	$\overrightarrow{\underrightarrow{z}}$

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}
```
$\ddot{\bar{\boldsymbol{x}}}$

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
```
$\dot{\bar{\boldsymbol{x}}}^{s}_{i}$

```
\bS[\Ddot]{x}_i^s
```
$\ddot{\bar{\boldsymbol{x}}}^{s}_{i}$

```
\bS{x}''^s_i
```
$\bar{\boldsymbol{x}}''^{s}_{i}$

```
\bSa{E}^k
```
$\overrightarrow{\bar{\boldsymbol{E}}}^{k}$

```
\aSb[\Dot]{E}_s
```
$\dot{\overrightarrow{\bar{\boldsymbol{E}}}}_{s}$

The commands are also robust and can be used in moving commands such as footnotes[2], 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 $\bar{\boldsymbol{r}}^s(\xi)$, the equation for a line integral

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

gives

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

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*⟩`}`

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

In general a typical setup to include unicode math

```
\usepackage{ifxetex}
\ifxetex
    ... Unicode-math and font selection ...
    \SetSymbFont{\symbfit}
\else
    ... Font selection ...
    \usepackage{bm}
    \SetSymbFont{\bm}
\fi
```

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}`	$\underline{\overrightarrow{\mathsfsl{E}}}_s$

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

`\bS{{}xy} \bS*{xy}`	$\overline{xy}$ $\overline{xy}$
`\bS{xy}`	$\overline{\boldsymbol{x}y}$
`\bS{{xy}}`	$\overline{\boldsymbol{xy}}$

or

`\bCSb*{\bS{x}+\bS{y}}`	$\overline{\overbrace{\boldsymbol{x}+\boldsymbol{y}}}$
`\bSb{E_{313}}^s_r`	$\overline{\boldsymbol{E}_{313}}^{s}_{r}$

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

```
\SetSymbStrut{\mathstrut}
\bSb{E}, \bS{x}, \Sb{y}
```
$\overline{\boldsymbol{E}}$, $\overline{\boldsymbol{x}}$, $\underline{\boldsymbol{y}}$

```
\SetSymbStrut{\vphantom{E}}
\bSb{E}, \bS{x}, \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}}
$\bSb*{\bO_i}$
```
$\overline{\boldsymbol{\Omega}_i}$

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.

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

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

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

6 To do's

- The vertical spacing between the symbols and the lines and arrows differs, $\underline{\overrightarrow{\boldsymbol{E}}}$, $\underline{\overrightarrow{\boldsymbol{E}}}$. 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).
- 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 esvect provides alternative vector symbols that can be used in `\overrightarrow`.

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

```
1 ⟨*package⟩
```

7.1 Identification

```
2 \NeedsTeXFormat{LaTeX2e}
3 \ProvidesPackage{mattens}[2022/03/01
4                           v1.3b
5                           Matrix/Tensors (DNJ Els)]
```

7.2 Options

\MT@SymbStrt
\SetSymbStrut

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

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

\MT@SymbFnt
\SetSymbFont

Initialize the symbol font formatting commands.

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

Process the options

```
14 \DeclareOption*{%
15     \PackageWarning{mattens}{Unknown option: \CurrentOption}}
16 \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.

```
17 \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.

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

\MT@Overline
\MT@Underline

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

```
20 \newcommand{\MT@Overline}[2]{{#1\overline{#2}}}
21 \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.[3] 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}{}
```

[3] 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: $\underrightarrow{\overline{\boldsymbol{E}}}$

```
\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: $\overline{\widetilde{\boldsymbol{\omega}}}$

```
\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: $\overrightarrow{\widetilde{\boldsymbol{\omega}}}$

```
\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: ⟨*symb*⟩`'`$^{\cdot}$`'`$^{\cdot}$`^`⟨*sup*⟩`_`⟨*sub*⟩

```
\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 ⟨*Sym*⟩`^`⟨*sup*⟩ 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.

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

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

```
191       \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.

```
192       \ifdim\wd\MT@Abox<\wd\MT@Sbox%
193           \leavevmode\rlap{\usebox\MT@Abox}{\usebox\MT@Sbox}%
194       \else%
195           \leavevmode\rlap{\usebox\MT@Sbox}{\usebox\MT@Abox}%
196       \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.

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

199 \newcommand*{\MT@@UnderAandB}[2]{%
200       \sbox{\MT@Tbox}{$\m@th#1\MT@accent{\MT@Symb{#2}}$}%
201       \setbox\MT@SPbox\null%
202       \wd\MT@SPbox\wd\MT@Tbox%
203       \sbox{\MT@Tbox}{$\m@th#1\MT@Symb{#2}$}%
204       \ht\MT@SPbox\ht\MT@Tbox%
205       \dp\MT@SPbox\dp\MT@Tbox%
206       \sbox{\MT@APbox}{$\m@th\MT@@cmd{#1}{\copy\MT@SPbox}$}%
207       \setlength{\MT@Wwdth}{\the\wd\MT@APbox}%
208       \addtolength{\MT@Wwdth}{-\the\wd\MT@SPbox}%
209       \sbox{\MT@Sbox}{$\m@th#1%
210          \hskip 0.5\MT@Wwdth\relax\MT@accent{\MT@Symb{#2}}\MT@SetSup$}%
211       \sbox{\MT@Abox}{$\m@th#1\xusebox{\MT@APbox}\MT@SetSub$}%
212       \ifdim\wd\MT@Abox<\wd\MT@Sbox%
213          \leavevmode\rlap{\usebox\MT@Abox}{\usebox\MT@Sbox}%
214       \else%
215          \leavevmode\rlap{\usebox\MT@Sbox}{\usebox\MT@Abox}%
216       \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.

```
217 \newcommand*{\MT@@bSb}[2]{\MT@Overline{#1}{\MT@Underline{#1}{\MT@Symb{#2}}}}
218 \newcommand*{\MT@@aSb}[2]{\MT@Overarrow{#1}{\MT@Underline{#1}{\MT@Symb{#2}}}}
219 \newcommand*{\MT@@bSa}[2]{\MT@Overline{#1}{\MT@Underarrow{#1}{\MT@Symb{#2}}}}
220 \newcommand*{\MT@@aSa}[2]{\MT@Overarrow{#1}{\MT@Underarrow{#1}{\MT@Symb{#2}}}}
221 \newcommand*{\MT@@bCSb}[2]{\MT@Overline{#1}{\MT@Underline{#1}{\MT@SymbC{#2}}}}
222 \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.

Symbols

\@ifl@aded 23

A

\aCSa *3*, 99
\aS *3*, 45
\aSa *3*, 87
\aSb *3*, 75
\AtBeginDocument 22

B

\bCSb *3*, 93
\bS *3*, 51
\bSa *3*, 81
\bSb *3*, 69

M

\mathord 24
\MT@@aCSa 101, 217
\MT@@aSa 89, 217
\MT@@aSb 77, 217
\MT@@bCSb 95, 217
\MT@@bSa 83, 217
\MT@@bSb 71, 217
\MT@@cmd 42, 47, 53, 59, 65, 71, 77, 83, 89, 95, 101, 184, 206, 226
\MT@@DoubleAandB 223
\MT@@OverAandB 177, 178
\MT@@Tensor 105
\MT@@UnderAandB 197
\MT@Abox 168, 191–193, 195, 211–213, 215, 226–229
\MT@accent . 41, 109, 188, 200, 210, 227
\MT@APbox 171, 184, 185, 188–190, 206, 207, 211
\MT@Askip 27, 31, 48, 61, 78, 85, 90, 91, 102, 103
\MT@bold 44, 106, 107, 164, 167
\MT@BPwdth 174
\MT@Bskip 28, 32, 54, 67, 72, 73, 79, 84, 96, 97
\MT@cmd 42, 46, 52, 58, 64, 70, 76, 82, 88, 94, 100, 163
\MT@DoubleAandB 70, 76, 82, 88, 94, 100, 223
\MT@GetPrimedSub 115
\MT@GetPrimedSuper 115
\MT@GetPrimes 113, 115
\MT@GetScripts 110, 111
\MT@GetSub 135, 142
\MT@GetSubSuper 143, 147
\MT@GetSuper 133, 139
\MT@GetSuperSub 140, 145
\MT@OverAandB 46, 52, 176
\MT@Overarrow . . . 18, 47, 218, 220, 222
\MT@Overline 20, 53, 217, 219, 221
\MT@Sbox 169, 187, 192, 193, 195, 209, 212, 213, 215
\MT@SBskip 30, 34, 49, 55
\MT@SetScripts 122, 128, 130, 136, 141, 144, 146, 148, 152
\MT@SetSub . . . 149, 154, 160, 211, 230
\MT@SetSubS 151, 155, 161, 187
\MT@SetSup 149, 153, 157, 191, 210, 230
\MT@SPbox 172, 180–184, 186, 201, 202, 204–206, 208
\MT@SPskip 29, 33, 60, 66
\MT@SPwdth 173
\MT@SubSkip 39, 49, 55, 61, 67, 73, 79, 85, 91, 97, 103, 160, 161
\MT@SupSkip 39, 48, 54, 60, 66, 72, 78, 84, 90, 96, 102, 157
\MT@Symb 164, 179, 187, 200, 203, 210, 217–220
\MT@SymbC 165, 221, 222
\MT@SymbFnt 10, 107
\MT@SymbStrt 6, 164, 166
\MT@Tbox 170, 179, 181–183, 188–191, 200, 202–205, 227–230
\MT@Tensor 50, 56, 62, 68, 74, 80, 86, 92, 98, 104, 105
\MT@UnderAandB 58, 64, 197
\MT@Underarrow . . 18, 59, 219, 220, 222
\MT@Underline . . . 20, 65, 217, 218, 221
\MT@UnprimedScripts 114, 131
\MT@Wwdth . 175, 185–187, 207, 208, 210

O

\overarrow@ 18

P

\prime 113, 117

R

\rightarrowfill@ 18, 19

S

\Sa *3*, 57
\Sb *3*, 63
\SetArrowSkip *6*, 31
\SetBarSkip *6*, 31
\SetSymbFont *4*, 10
\SetSymbStrut *5*, 6
\SetSymSubSkip *6*, 31
\SetSymSupSkip *6*, 31

U

\underarrow@ 19

X

\xusebox . . . 22, 188, 191, 211, 227, 230

Change History

1.0
General: Initial version 1
1.1
\MT@@Tensor: Added to reduce overall code. 11
\MT@GetPrimedSub: Add: Extract primes 12
\MT@GetPrimedSuper: Add: Extract primes 12
\MT@GetPrimes: Add: Extract primes 12
\MT@GetScripts: Add: Code to check for primes 12
\MT@Tensor: Added to reduce overall code. 11
\xusebox: More streamline. 9
General: Code: Remove helper commands 16
Doc: Updated 1
1.2
\MT@@DoubleAandB: Use \xusebox inside accent to account for \dddot. 15
\MT@@OverAandB: Use \xusebox inside accent to account for \dddot. 14
\MT@@UnderAandB: Make provisions for wide accents (\dddot). . . . 15
\xusebox: Test if pdftex.def is loaded. and command is rewritten to use a \mathord to restore script heights 9
1.3
General: Doc: Updated 1
Remove \ensuremath to force math mode. 13
1.3a
General: Remove invalid UTF-8 byte sequences 8
1.3b
General: Doc: Updated 1