mismath
Miscellaneous mathematical macros*

Antoine Missier
antoine.missier@ac-toulouse.fr

December 26, 2022

Contents

1 Introduction

According to the International Standards ISO 31-0:1992 to ISO 31-13:1992, superseded by ISO 80000-2:2009, mathematical constants e, i, π should be typeset in roman (upright shape) and not in italic (sloping shape) like variables (see [1] [2] [3] [4]). This package provides some tools to achieve this (automatically).

Even if it is recommended to typeset vectors names in bold italic style [2] [4], they are often represented with arrows (particularly in school documents or in physics). To draw pretty arrows above vectors, we use the esvect package by Eddie Saudrais [5] and we provide a few more macros related to vectors with arrows, in particular to improve the typesetting of the norm: $\left\|\vec{AB}\right\|$ instead of LaTeX version $\|\vec{AB}\|$ which is not vertically adjusted, or worse $\left\|\vec{AB}\right\|$ (and even ugly with Latin Modern font family).

*This document corresponds to mismath v2.1, dated 2022/12/26. Thanks to François Bastouil for help in English translation.

The package also provides other macros for:

- some standard operator names,
- a few useful aliases,
- improving some spacing in mathematical formulas,
- systems of equations and small matrices,
- displaymath in double columns for long calculation.

To avoid incompatibility, a large majority of our macros will be defined only if there is not another command with the same name in the packages loaded before mismath. If a macro is already defined, compilation will produce a warning message and mismath definition will simply be ignored. To keep mismath command, either load mismath before the other package with which it is in conflict for the name of that command (assuming the other package supports it), or use `\let\`⟨*command*⟩`\relax` before loading mismath.

⟨options⟩ The amsmath package is loaded by mismath without option. For using amsmath with options (see [6]), these options can be added when calling mismath, or amsmath has to be loaded with the required options before mismath.

Another package, mathtools by Morten Høgholm and Lars Madsen [7] is also loaded. It provides many useful macros and improvements of amsmath package.

A recommendation, seldom observed, is to typeset uppercase Greek letters in italic shape like other variables [4]. This is automatically done with the packages fixmath by Walter Schmidt [8], isomath by Günter Milde [9] or pm-isomath by Claudio Beccari [10] and optionally with many others (for instance mathpazo or mathptmx with the option `slantedGreek`), but this feature is not implemented here because this rule is conflicting to the one used in France where all mathematics capitals have to be typeset in upright shape[1]. The choice of loading or not one of these packages remains thus to the user.

2 Usage

2.1 Mathematical constants

`\mathup` `\e` `\i` `\j` As for classic functions identifiers, *predefined* mathematical constants should be typeset in upright shape (generally in roman family), even if this practice is not really common and tedious to respect. To avoid to stuff a document with `\mathrm{e}` or `\mathrm{i}` (or better `\mathup{e}` and `\mathup{i}`[2]), the package provides `\e` command for the base of the natural logarithm and `\i` or `\j` for imaginary numbers.

[1] The frenchmath package [20] takes this rule into account.

[2] `\mathup` is based on `\operatorfont` (from amsopn package, automatically loaded by amsmath). The beamer package uses a default sans serif math font, but `\mathrm` produces a font with serif in beamer. This problem is solved by using `\mathup` instead of `\mathrm`.

Let's notice that \i and \j already exist in LaTeX: using in LR mode, they produce "ı, ȷ" without the point, so you can place accents on them, and in mathematical mode they produce "`LaTeX Warning: Command \i invalid in math mode on input line ⟨line⟩`". Te new definition of \i and \j concerns only mathematical mode[3].

\enumber
\inumber
\jnumber

Nevertheless, it can be tiresome to type a lot of backslashes in a document with many formulas containing e or i. So a way is proposed here to free of it by placing \enumber, \inumber or \jnumber in the preamble: e, i or j will then automatically be set in roman in the whole document, no need to type \e, \i or \j, let's hope that there are not many other *e*, *i* or *j* as variables. However, you can still get italicized *e*, *i* or *j* with LaTeX command \mathit or \mathnormal. Of course, this does not fully comply with LaTeX philosophy: in the document body, objects should be pointed out by their nature rather than their typographical characteristics, defined in the preamble. But these macros are really handy and thanks to them it is possible to bring a document up to the standards afterwards; moreover anyone is free to use them or not.

\pinumber[⟨command⟩]

The mathematical constant π should also be typeset in upright shape (see [1], [2], [4]), which differs from italicized π. This recommendation is even less observed than the one concerning e and i [1]. Several packages allow to typeset mathematical Greek letters in upright shape, let us mention upgreek [11], mathdesign [12] (used here), kpfonts [14], fourier [15], libertinust1math, pxgreeks, txgreeks, libgreek, etc. A special mention to lgrmath of Jean-François Burnol [16] which allow to use, in math mode, any Greek LGR-encoded font. These packages provide commands like \uppi (upgreek), \piup (mathdesign, kpfonts, lgrmath), \otherpi (fourier), etc.[4] To preserve default sloped lowercase Greek letters except for pi, and to avoid typing a lot of \uppi or \piup, we provide the \pinumber macro, which has to be put in the preamble. This command redefines \pi to match the optional command given, for instance `piup`, assuming the appropriate package has been loaded before.

By activating \enumer, \inumber and \pinumber[piup] in the preamble, you can get for instance :

```
$e^{i\pi} = -1$
```
$e^{i\pi} = -1.$

When no argument is given, \pinumber defines \pi with an LGR encoding of Greek letters to produce π. It looks the same as the one supplied with Günter Milde's textalpha package [13]. This π is particularly suitable for use with the default Computer Modern or Latin Modern font family[5].

\itpi

When \pinumber is activated, the original italic π is still available with \itpi.

2.2 Vectors

\vect

By default, the \vect command[6], produces vectors with arrows (thanks to the es-

[3] Due to this \i command redefinition, there is an incompatibility with beamer when using i with accents in beamer titles. A solution is to use the classic \^i command to produce î in beamer titles for example.

[4] They also have options to typeset all the Greek lowercase letters in upright shape by default, but this in not our goal here.

[5] This default π doesn't fit well with many text fonts, more bold than Computer Modern; the upgreek package [11] provides often a better π and it has also a `Symbol` option (using Adobe Symbol font) that fits well with several text fonts, for instance Times.

vect package of Eddie Saudrais[7]) which are more elegant than those produced by LaTeX's `\overrightarrow` command. The esvect package has an optional argument (one letter between `a` and `h`) defining the required type of arrow (see [5]). In mismath, esvect is loaded with the option `b`: `\vect{AB}` gives $\overrightarrow{AB}$. To choose another type of arrow, esvect must be called with the required option *before* mismath, e.g. `\usepackage[d]{esvect}` will give the arrows produced by default in [5].

`\boldvect` The `\vect` macro allow to typeset vector's names using bold italic (according to ISO recommendation [2] [3]) rather than arrows. For this, calling `\boldvect` will modify the behavior of `\vect`, globally or locally, depending on where `\boldvect` is placed:

```
\[ \boldvect \vect{v}
   =\lambda\vect{e}_x+\mu\vect{e}_y. \]
```

$$\boldsymbol{v} = \lambda \boldsymbol{e}_x + \mu \boldsymbol{e}_y.$$

`\boldvectcommand` By default `\boldvect` uses the `\boldsymbol` command[8] from amsbsy package, loaded by amsmath. But other packages producing bold italic can be preferred, e.g. `\bm` from bm package or `\mathbold` from fixmath package or `\mathbfit` from isomath. For that, redefine `\boldvectcommand`, for instance:

```
\renewcommand\boldvectcommand{\mathbold}.
```

By setting `\boldvectcommand` to `\mathbf`, `\vect` produces vectors in bold *upright* shape, which tends to be used instead of bold *italic* (but probably for bad reasons).

`\arrowvect` At any moment, you can get back to the default behavior with the inverse switch `\arrowvect`. These switches can be placed anywhere: inside mathematical mode or inside an environment (with local effect) or outside (with global effect).

`\hvect` When vectors with arrows are typeset side by side, arrows can be set up a bit higher (with a vertical phantom box containing t) to avoid inelegant effects:

- $\overrightarrow{AB} = \overrightarrow{u} + \overrightarrow{AC}$, obtained with `\hvect{u}`, is better than $\overrightarrow{AB} = \overrightarrow{u} + \overrightarrow{AC}$;
- $\overrightarrow{a} \cdot \overrightarrow{b} = 0$, obtained with `\hvect{a}`, is better thant $\overrightarrow{a} \cdot \overrightarrow{b} = 0$.

The `\boldvect` switch has the same effect on `\hvect` than on `\vect`.

`\hvec` In a similar way, `\hvec` raises the little arrow produced by the LaTeX command `\vec` (from height of t letter):

- $\mathscr{P} = \vec{f} \cdot \vec{v}$, obtained with `\hvec{v}`, is better than $\mathscr{P} = \vec{f} \cdot \vec{v}$.
- $\vec{f} = m\vec{a}$, obtained with `\hvec{a}`, is better than $\vec{f} = m\vec{a}$.

`\norm` The norm of a vector is classically produced by the delimiters `\lVert` and `\rVert` (rather than `\|`) or `\left\Vert` and `\right\Vert` for delimiters adapting to the

[6] As for many macros of this package, the definition will take effect only if this macro is not defined before by another package.

[7] esvect provides `\vv` macro used by `\vect`.

[8] `\mathbf` gives upright bold font, even if used in combination with `\mathit`.

content. Unfortunately, these delimiters are always vertically centered, relatively to the middle of the base line, whereas vectors with arrows are asymmetric objects. The code `$\norm{\vec{h}}$` raises a smaller double bar to produce $\|\vec{h}\|$ instead of $\left\|\vec{h}\right\|$. Let's notice that the height of the bars don't adjust to content, but however to context: main text, subscripts or exponents, e.g. $e^{\|\vec{h}\|}$.

2.3 Standard operator names

`\di` The *differential* operator should be typeset in upright shape and not in italic, to make it different from variables (as mentioned in [1] [2] [4] [22]). For this, we provide the `\di` command. See the following examples (notice the thin spaces before the d, as for classic function's names):

```
\[ \iint xy\di x\di y \]
```

$$\iint xy \,\mathrm{d}x \,\mathrm{d}y$$

```
\[ m\frac{\di^2x}{\di t^2}
   + h\frac{\di x}{\di t} + kx = 0 \]
```

$$m\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + h\frac{\mathrm{d}x}{\mathrm{d}t} + kx = 0$$

This command can also stand for *distance* (hence its name):

$$\lambda\, \mathrm{d}(A, \mathscr{F}) + \mu\, \mathrm{d}(B, \mathscr{H}).$$

`\P` `\E` `\PEupright` `\V` To refer to probability[9] and expectation the proper use is to typeset capital letters P, E in roman as for any standard function identifier. This is obtained with `\P` and `\E`. In the same way as for i and j, you can use `\PEupright` in the preamble to avoid typing many `\P` or `\E`. Variance is generally denoted by var or Var (see table below), but some authors prefer to use V, produced by `\V`.

`\Par` The `\P` command already existed to refer to the end of paragraph symbol ¶ and has been redefined, but this symbol can still be obtained with `\Par`.

`\probastyle` Some authors use "blackboard bold" font to represent probability, expectation and variance: $\mathbb{P}, \mathbb{E}, \mathbb{V}$. The `\probastyle` macro sets the appearance of `\P`, `\E` and `\V`: for instance `\renewcommand\probastyle{\mathbb}`[10] brings the previous "open-work" letters. `\mathbb` comes from amsfonts package (loaded by amssymb but also available standalone) which has to be called in the preamble.

The following standard operator names are defined in mismath:

`\adj`	adj	`\erf`	erf	`\Re`	Re
`\Aut`	Aut	`\grad`	$\overrightarrow{\text{grad}}$	`\rot`	$\overrightarrow{\text{rot}}$
`\codim`	codim	`\id`	id	`\sgn`	sgn
`\Conv`	Conv	`\Id`	Id	`\sinc`	sinc
`\cov`	cov	`\im`	im	`\spa`	span
`\Cov`	Cov	`\Im`	Im	`\tr`	tr
`\curl`	$\overrightarrow{\text{curl}}$	`\lb`	lb	`\var`	var
`\divg`	div	`\lcm`	lcm	`\Var`	Var
`\End`	End	`\rank`	rank	`\Zu`	Z

[9] LaTeX provides also `\Pr` which gives Pr.
[10] As for `\boldvect` and `\arrowvect`, effect is local to the container environment.

By default, operators returning vectors, \grad and \curl (or its synonym \rot rather used in Europe), are written with an arrow on the top. When \boldvect is activated, they are typeset in bold style: **grad**, **curl**, **rot**. For the variance, the covariance and the identity function, two notations are proposed, with or without a first capital letter, because they are both very common. On the other hand, "im" stands for the image of a linear transformation (like "ker" for the kernel) whereas "Im" is the imaginary part of a complex number. Notice that \div already exist (÷) and \span is a TEX primitive (used in \multicolumn); they haven't been redefined, therefore the macros \divg (divergence) and \spa (span of a set of vectors) ; \Z is used for the set of integers (see 2.4), therefore we used \Zu, to designate the center of a group: Z(G) (from German Zentrum).

\oldRe
\oldIm

The \Re and \Im macros already existed, to refer to real and imaginary part of a complex number, producing outdated symbols $\mathfrak{R}$ and $\mathfrak{J}$. They have been redefined according to actual use, as mentioned in the above table, but it's still possible to get the old symbols with \oldRe and \oldIm.

Some (inverse) circular or hyperbolic functions, missing in LaTeX, are also provided by mismath:

\arccot	arccot	\arsinh	arsinh	\arcoth	arcoth
\sech	sech	\arcosh	arcosh	\arsech	arsech
\csch	csch	\artanh	artanh	\arcsch	arcsch

\bigO
\bigo
\lito

Asymptotic comparison operators (in Landau notation) are obtained with \bigO or \bigo and \lito commands:

$$n^2 + \mathcal{O}(n \log n) \quad \text{or} \quad n^2 + \mathrm{O}(n \log n) \quad \text{and} \quad \mathrm{e}^x = 1 + x + \mathrm{o}(x^2).$$

2.4 A few useful aliases

In the tradition of Bourbaki and D. Knuth, proper use requires that classic sets of numbers are typeset in bold roman: **R**, **C**, **Z**, **N**, **Q**, whereas "openwork" letters ($\mathbb{R}, \mathbb{Z}, \ldots$) are reserved for writing at blackboard [22]; and likewise to designate a field: **F** or **K** (Körper in German). We get these symbols with the macros:

\R, \C, \Z, \N, \Q, \F, \K.

\mathset

The \mathset command enables to change the behavior of all these macros in a global way: by default, \mathset is an alias for \mathbf, but if you prefer openwork letters, just place \renewcommand\mathset{\mathbb} where you want, for instance in the preamble, after loading amsfonts package (which provides the "blackboard bold" typeface, also loaded by amssymb).

\ds

The \displaystyle command being very common, alias \ds is provided. Not only it eases typing but also it makes source code more readable.

Symbols with limits behave differently for in-line formulas or for displayed equations. In the latter case, "limits" are put under or above whereas for in-line math mode,

they are placed on the right, as subscript or exponent. Compare: $\zeta(s)=\sum_{n=1}^{\infty}\frac{1}{n^s}$ with

$$\zeta(s)=\sum_{n=1}^{\infty}\frac{1}{n^s}.$$

\dlim
\dsum
\dprod
\dcup
\dcap

With in-line math mode, displaymath behavior can be forced with `\displaystyle` or its alias `\ds`, but then, all the rest of the current mathematical environment will be set in displaymath mode too (in the previous example, the fraction will be expanded). Just as the amsmath command `\dfrac` only transforms the required fraction in display style, we can limit the display style effect to the affected symbol, by using the following macros: `\dlim`, `\dsum`, `\dprod`, `\dcup`, `\dcap`. So `$\dlim_{x\to +\infty}\frac{1}{x}$` gives $\dlim_{x\to +\infty}\frac{1}{x}$.

\lbar
\hlbar

Large bars over expressions are obtained with `\overline` or, shorter, its alias `\lbar`, to get for instance $\overline{z_1 z_2}$. Such as for vectors, you can raise the bar (from the height of h) with the `\hlbar` command, in order to correct uneven bars heights.

$\overline{z+z'}=\overline{z}+\overline{z'}$, obtained with `\hlbar{z}`, is better than $\overline{z+z'}=\overline{z}+\overline{z'}$.

\eqdef
\eqdef*

The `\eqdef` macro writes equality symbol topped with "def" or with "$\triangle$" for `\eqdef*` (thanks to the LaTeX command `\stackrel`):

```
$ \e^{\i\theta} \eqdef
  \cos\theta + \i\sin\theta $
```

$$\mathrm{e}^{\mathrm{i}\theta}\stackrel{\text{def}}{=}\cos\theta+\mathrm{i}\sin\theta$$

\unbr

`\unbr` is an alias for `\underbrace`[11], making source code more compact.

```
$ (QAP)^n = \unbr{QAP\mul QAP\mul
  \cdots\mul QAP}_{n\text{ times}} $
```

$$(QAP)^n=\unbr{QAP\times QAP\times\cdots\times QAP}{n\text{ times}}$$

\iif

`\iif` is an alias for "if and only if", to be used in text mode.

2.5 Improved spacing in mathematical formulas

\then
\txt

The `\then` macro produces the symbol $\Longrightarrow$ surrounded by large spaces as the standard macro `\iff` does it with $\Longleftrightarrow$. In a similar way, `\txt` based on the `\text` macro (from the amstext package, loaded by amsmath), leaves em quad spaces (`\quad`) around the text. See the following example:

```
\[ \ln x=a \then x=\e^a \txt{rather than}
    \ln x=a \Longrightarrow x=\e^a \]
```

$$\ln x=a \then x=\mathrm{e}^a \quad \text{rather than} \quad \ln x=a \Longrightarrow x=\mathrm{e}^a$$

\mul

The multiplication symbol obtained with `\times` produces the same spacing than addition or subtraction operators, whereas division obtained with / is closer to its operands. This actually hides the priority of the multiplication on + and −. This is why we provide the `\mul` macro, behaving like / (ordinary symbol) and leaving less space around than `\times`:

[11] The mathtools package by Morten Høgholm and Lars Madsen [7] provides a new improved version of `\underbrace` command (as many other usefull macros); it is loaded by mismath.

$\lambda+\alpha\times b-\beta\times c$, obtained with `\mul`, is better than $\lambda+\alpha\times b-\beta\times c$.

When using `\mul` before a function name or around a `\left...\right` structure, the space may be too large on one side of `\mul`. To get the same amount of space on the two sides of `\mul`, you can use thin negative spaces `\!` or enclose the function or the structure with braces:

$x\times\sin x$, obtained with `x\mul{\sin x}`, is slightly better than $x\times\sin x$.

`$\sin\!{\left( \frac{\pi}{3} \right)} \mul 2$` gives
$\sin\left(\frac{\pi}{3}\right)\times 2$ which is better than $\sin\left(\frac{\pi}{3}\right)\times 2$.

The thin negative space after the function name is not relative to `\mul`, but is due to the fact that spaces around a `\left...\right` structure are bigger than those produced by single parenthesis `(...)`.

`\pow` In the same way, when typesetting an exponent after a closing *big* parenthesis produced by `\right)`, the exponent is little to far from the parenthesis. The command `\pow{`⟨*expr*⟩`}{`⟨*pow*⟩`}` sets ⟨*expr*⟩ between parentheses and puts the exponent ⟨*pow*⟩ slightly closer to the right parenthesis[12]. Compare:

$$\mathrm{e}^a\sim\left(1+\frac{a}{n}\right)^n \quad \text{may be better than} \quad \mathrm{e}^a\sim\left(1+\frac{a}{n}\right)^n.$$

`\abs` Absolute value (or modular for a complex number) should be typeset with `\lvert ... \rvert` rather than | which doesn't respect correct spaces for delimiters; for bars whose height has to adapt to content, we use `\left\vert ... \right\vert` or, more simply, the `\abs{...}` command which is equivalent[13].

`\lfrac` This macro behaves like `\frac` but with thick spaces around the arguments, so the corresponding fraction bar is perceptibly a little bit longer:

```
\[ \lbar{Z} =
   \lfrac{\lbar{z_1-z_2}}{\lbar{z_1+z_2}} \]
```

$$\overline{Z}=\frac{\overline{z_1-z_2}}{\overline{z_1+z_2}}$$

`brackets` Open intervals are usually represented with parenthesis, e.g. $(0,+\infty)$, but sometimes we find also brackets, for example in French mathematics. In that case the space around them is often unsuitable, e.g. $x\in]0,+\infty[$.

We have redefine brackets in the ibrackets package [21] which is loaded by mismath. Simply type `$x\in ]-\pi,0[ \cup ]2\pi,3\pi[$` to get

$$x\in]-\pi,0[\cup]2\pi,3\pi[\quad \text{with ibrackets,}$$
$$\text{instead of} \quad x\in]-\pi,0[\cup]2\pi,3\pi[\quad \text{without ibrackets.}$$

In our code, [and] symbols become "active" and are not defined by default as delimiters. Thereby a line break could occur between the two brackets, but it is always possible to transform them into delimiters with `\left` and `\right`.

[12] This macro gives bad results with normal sized parenthesis.

[13] Another solution is to define `\abs` with the `\DeclarePairedDelimiter` command from the mathtool package [7].

With ibrackets: a bracket become an ordinary character but an open delimiter when it is immediately followed by a + or - character. Thus, when the left boundary contains an operator sign, *you don't have to leave a space between the first bracket and the sign*, otherwise, the spaces surrounding the operator will be too large: e.g. `$x \in ] -\infty, 0]$` yields $x \in]-\infty, 0]$. Contrariwise, when you want to write algebra on intervals then *you must leave a blank space between the second bracket and the +/- operations*, e.g. `$[a, b] + [c, d]$` yields $[a, b] + [c, d]$ but `$[a, b]+ [c, d]$` yields $[a, b]+[c, d]$.

Let us also mention other approaches e.g. `\DeclarePairedDelimiters`, a macro from the mathtools package [7], or the interval package [17] with his `\interval` macro. Nevertheless our solution is lighter.

2.6 Environments for systems of equations and small matrices

`system` The `system` environment produces a system of equations:

```
$\begin{system}
    x=1+2t \\ y=2-t \\ z=-3-t
\end{system}$
```

$$\begin{cases} x = 1+2t \\ y = 2-t \\ z = -3-t \end{cases}$$

`\systemsep` This first example could also have been produced with `cases` environment from amsmath package, although `cases` places mathematical expressions closer to the bracket (which makes sense considering it's use). `\systemsep` enables to set the gap between the bracket and the expressions, set by default to `\medspace`. This gap may be reduced, for instance: `\renewcommand{\systemsep}{\thinspace}`, or enlarged with `\thickspace` (and with `\renewcommand\systemsep}{}` we obtain what `cases` do).

`system[⟨coldef⟩]` By default, a system is written like an `array` environment with only one column, left aligned. The environment has an optional argument to create several columns, specifying their alignment, with the same syntax than the `array` environment of LaTeX: `\begin{system}[cl]` produces a two-column system, the first one being centered, the second being left aligned, such as in the following example:

```
$\begin{system}[cl]
    y & =\dfrac{1}{2}x-2 \\[1ex]
    (x,y) & \neq (0,-2)
\end{system}$
```

$$\begin{cases} y & = \dfrac{1}{2}x-2 \\ (x,y) & \neq (0,-2) \end{cases}$$

`\systemstretch` Default spacing between the lines of a `system` environment has been slightly enlarged compared to the one from `array` environments (from 1.2 factor). This spacing may be changed by typing `\renewcommand{\systemstretch}{⟨stretch⟩}`, inside the current mathematical environment (for a local change) or outside (for a global change). By default, stretch's value is 1.2. In addition we can use the end of line with a spacing option such as it has been done above with `\\[1ex]`.

Another example with `\begin{system}[rl@{\quad}l]`[14]:

$$\left\{\begin{array}{rll} x+3y+5z=0 & R_1 \\ 2x+2y-z=3 & R_2 \\ 3x-y+z=2 & R_3 \end{array}\right. \iff \left\{\begin{array}{rll} x+3y+5z=0 & R_1 \\ 4y+11z=3 & R_2 \leftarrow 2R_1-R_2 \\ 5y+7z=-1 & R_3 \leftarrow \frac{1}{2}(3R_1-R_3) \end{array}\right.$$

Let's mention the systeme package [18] which deals with linear systems with a lighter syntax and automatic alignments on $+$, $-$, $=$, and also the spalign package [19] which moreover produces nice alignments for matrices (with spaces and semicolons as delimiters).

`spmatrix` The amsmath package provides various environments to typeset matrices: for instance `pmatrix` surrounds the matrix with parenthesis or `smallmatrix` typesets a small matrix that can even be inserted in a text line. We provide a combination of the two with `spmatrix`:
`$\vec{u}\begin{spmatrix}-1\\2\end{spmatrix}$` yielding $\vec{u}\begin{spmatrix}-1\\2\end{spmatrix}$.

The mathtools package enhance amsmath matrices environments and provides also a small matrix environment with parenthesis. Furthermore, with starred version `\begin{psmallmatrix*}[`*col*`]`, you can choose the alignment inside the columns (`c`, `l` or `r`). But sadly, the space before the left parenthesis is too narrow regarding to the space inside the parenthesis. Compare previous $\vec{u}\begin{pmatrix}-1\\2\end{pmatrix}$ with $\vec{u}\begin{psmallmatrix}-1\\2\end{psmallmatrix}$.

2.7 Displaymath in double columns

`mathcols` The `mathcols` environment activates mathematical mode and enables to arrange "long"calculation in double columns, separated with a central rule, as shown in the following example. But you have to load the multicol package in the preamble.

$$\begin{aligned} & \frac{1}{2\times\left(\frac{1}{4}\right)^n+1} \geq 0.999 \\ \iff\ & 1 \geq 1.998\left(\frac{1}{4}\right)^n + 0.999 \\ \iff\ & 0.001 \geq \frac{1.998}{4^n} \end{aligned} \qquad \begin{aligned} \iff\ & 4^n \geq 1998 \\ \iff\ & n\ln 4 \geq \ln(1998) \\ \iff\ & n \geq \frac{\ln(1998)}{\ln 4} \approx 5.4 \\ \iff\ & n \geq 6 \end{aligned}$$

`\changecol` The `\changecol` macro causes a change of column; alignment is produced using the classic delimiters `&` and `\\`.

```
\begin{mathcols}
          & \frac{1}{2 \mul {\pow{\frac{1}{4}}{n}} + 1} \geq 0.999 \\
    \iff\ & 1 \geq 1.998  \pow{\frac{1}{4}}{n} + 0.999 \\
    \iff\ & 0.001 \geq \frac{1.998}{4^n} \\
\changecol
    & \iff 4^n \geq 1998 \\
    & \iff n \ln 4 \geq \ln(1998) \\
```

[14] `@{...}` sets inter-column space.

```
    & \iff n \geq \frac{\ln(1998)}{\ln 4} \approx 5.4 \\
    & \iff n \geq 6
\end{mathcols}
```

3 Implementation

```
\DeclareOption*{\PassOptionsToPackage{\CurrentOption}{amsmath}}
\ProcessOptions \relax
\@ifpackageloaded{amsmath}{}{\RequirePackage{amsmath}}
\@ifpackageloaded{esvect}{}{\RequirePackage[b]{esvect}}
\RequirePackage{ifthen}
\RequirePackage{xspace}
\RequirePackage{mathtools}
\RequirePackage{ibrackets}

```

The above conditional packages loading avoids “option clash” errors if the packages have been previously loaded with other options.

`\bslash` The `\bslash` macro comes from Frank Mittelbach’s doc.sty package. It can also be used in other documents instead of `\textbackslash` (which doesn’t work inside warnings).

```
{\catcode`\|=\z@ \catcode`\\=12 |gdef|bslash{\}} % \bslash command

```

`\@mwarning` `\@mmacro` `\@moperator` The three following internal macros are meta commands for a conditional macro definition with a warning message if the macro already exists. They could be useful in other packages.

```
\newcommand\@mwarning[1]{
    \PackageWarningNoLine{mismath}{
        Command \bslash #1 already exist and will not be redefined}
}
\newcommand\@mmacro[2]{
    \@ifundefined{#1}{
        \expandafter\def\csname #1\endcsname{#2}
    }{\@mwarning{#1}}
}
\newcommand\@moperator[3][]{% this macro is ugly, by default #1=#3
    \ifthenelse{\equal{#1}{}}{
        \@ifundefined{#3}{
            \DeclareMathOperator{#2}{#3}
        }{\@mwarning{#3}}
    }{
        \@ifundefined{#1}{
            \DeclareMathOperator{#2}{#3}
        }{\@mwarning{#1}}
    }
}
```

To produce the correct upright shape font even when working with the beamer package, we did not use \mathrm but \mathup (based on \operatorfont from the amsopn package). This command works also fine with other sans serif fonts like cmbright. Moreover for beamer, \enumber must be typeset in the family default font (sans serif), therefore the \AtBeginDocument inside the macro (otherwise it has no effect). The same holds for \inumber and \jnumber.

\AtBeginDocument is also necessary to redefine \i when calling the hyperref package which overwrites the \i definition.

```
\providecommand{\mathup}[1]{{\operatorfont #1}} % also in kpfonts
\@mmacro{e}{\mathup{e}}
\AtBeginDocument{\let\oldi\i \let\oldj\j
    \renewcommand{\i}{\TextOrMath{\oldi}{\mathup{i}}}
    \renewcommand{\j}{\TextOrMath{\oldj}{\mathup{j}}} }

\newcommand{\enumber}{
    \AtBeginDocument{\DeclareMathSymbol{e}\mathalpha{operators}{`e}}}
\newcommand{\inumber}{
    \AtBeginDocument{\DeclareMathSymbol{i}\mathalpha{operators}{`i}}}
\newcommand{\jnumber}{
    \AtBeginDocument{\DeclareMathSymbol{j}\mathalpha{operators}{`j}}}
\newcommand*{\pinumber}[1][defaultpi]{
    \@ifundefined{itpi}{\let\itpi\pi}{\@mwarning{itpi}}
    \ifthenelse{\equal{#1}{defaultpi}}{
        \usepackage[LGR,T1]{fontenc}
        \DeclareSymbolFont{UpGr}{LGR}{lmr}{m}{n}
        \DeclareMathSymbol{\pi}\mathalpha{UpGr}{"70}
    }{
        \@ifundefined{#1}{
            \PackageWarningNoLine{mismath}{
                \bslash pinumber command has changed since v2.0,
                \MessageBreak
                option #1 must be a valid command name \MessageBreak
                (look at the documentation),
                but command \bslash #1 is undefined, \MessageBreak
                I cannot use it for replacement to \bslash pi.
                \MessageBreak
                Perhaps a missing package}
            }{\renewcommand{\pi}{\csname #1\endcsname}}
    }
}

\newboolean{arrowvect}
\setboolean{arrowvect}{true}
\newcommand{\arrowvect}{\setboolean{arrowvect}{true}}
\newcommand{\boldvect}{\setboolean{arrowvect}{false}}
\newcommand{\boldvectcommand}{\boldsymbol} % from amsbsy package
\@mmacro{vect}{\ifthenelse{\boolean{arrowvect}}{
```

```
        \vv}{\boldvectcommand}}
\newcommand*{\hvect}[1]{\vect{\vphantom{t}#1}}
\newcommand*{\hvec}[1]{\vec{\vphantom{t}#1}}

\newcommand*{\@norm}[1]{
    \mbox{\raisebox{1.75pt}{\small$\bigl\Vert$}} #1
    \mbox{\raisebox{1.75pt}{\small$\bigr\Vert$}} }
% works better than with relative length
\newcommand*{\@@norm}[1]{
    \mbox{\footnotesize\raisebox{1pt}{$\Vert$}} #1
    \mbox{\footnotesize\raisebox{1pt}{$\Vert$}} }
\newcommand*{\@@@norm}[1]{
    \mbox{\tiny\raisebox{1pt}{$\Vert$}} #1
    \mbox{\tiny\raisebox{1pt}{$\Vert$}} }
\@ifundefined{norm}{\providecommand*{\norm}[1]{
        \mathchoice{\@norm{#1}}{\@norm{#1}}{\@@norm{#1}}{\@@@norm{#1}}
        }
    }{\@mwarning{norm} } % bad result with libertinust1math

\@mmacro{di}{\mathop{}\!\mathup{d}}
\newcommand\probastyle{}
\let\Par\P % end of paragraph symbol
\renewcommand{\P}{\operatorname{\probastyle{P}}}
\@mmacro{E}{\operatorname{\probastyle{E}}}
\@mmacro{V}{\operatorname{\probastyle{V}}}
\newcommand{\PEupright}{
    \AtBeginDocument{% necessary for working with beamer
        \DeclareMathSymbol{P}\mathalpha{operators}{`P}
        \DeclareMathSymbol{E}\mathalpha{operators}{`E}
    }
}

\@moperator{\adj}{adj}
\@moperator{\Aut}{Aut}
\@moperator{\codim}{codim}
\@moperator{\Conv}{Conv}
\@moperator{\cov}{cov}
\@moperator{\Cov}{Cov}
\@mmacro{curl}{\operatorname{\vect{\mathup{curl}}}}
\@moperator[divg]{\divg}{div}
\@moperator{\End}{End}

\@moperator{\erf}{erf}
\@mmacro{grad}{\operatorname{\vect{\mathup{grad}}}}
\@moperator{\id}{id} % mathop or mathord ?
\@moperator{\Id}{Id}
\@moperator{\im}{im}
\let\oldIm\Im \renewcommand{\Im}{\operatorname{Im}}
\@moperator{\lb}{lb}
\@moperator{\lcm}{lcm}
```

```
122
123 \@moperator{\rank}{rank}
124 \let\oldRe\Re \renewcommand{\Re}{\operatorname{Re}}
125 \@mmacro{rot}{\operatorname{\vect{\mathup{rot}}}}
126 \@moperator{\sgn}{sgn}
127 \@moperator{\sinc}{sinc}
128 \@moperator[spa]{\spa}{span}
129 \@moperator{\tr}{tr}
130 \@moperator{\var}{var}
131 \@moperator{\Var}{Var}
132 \@moperator[Zu]{\Zu}{Z}
133
134 \@moperator{\arccot}{arccot}
135 \@moperator{\sech}{sech}
136 \@moperator{\csch}{csch}
137 \@moperator{\arsinh}{arsinh}
138 \@moperator{\arcosh}{arcosh}
139 \@moperator{\artanh}{artanh}
140 \@moperator{\arcoth}{arcoth}
141 \@moperator{\arsech}{arsech}
142 \@moperator{\arcsch}{arcsch}
143
144 \@moperator[bigO]{\bigO}{\mathcal{O}}
145 \@moperator[bigo]{\bigo}{O}
146 \@moperator[lito]{\lito}{o}
147
148 \@mmacro{mathset}{\mathbf}
149 \@mmacro{R}{\ensuremath{\mathset{R}}\xspace}
150 \@mmacro{C}{\ensuremath{\mathset{C}}\xspace}
151 \@mmacro{N}{\ensuremath{\mathset{N}}\xspace}
152 \@mmacro{Z}{\ensuremath{\mathset{Z}}\xspace}
153 \@mmacro{Q}{\ensuremath{\mathset{Q}}\xspace}
154 \@mmacro{F}{\ensuremath{\mathset{F}}\xspace}
155 \@mmacro{K}{\ensuremath{\mathset{K}}\xspace}
156
157 \@mmacro{ds}{\displaystyle}
158 \@mmacro{dlim}{\lim\limits}
159 \@mmacro{dsum}{\sum\limits}
160 \@mmacro{dprod}{\prod\limits}
161 \@mmacro{dcup}{\bigcup\limits}
162 \@mmacro{dcap}{\bigcap\limits}
163
164 \@mmacro{lbar}{\overline}
165 \@ifundefined{hlbar}{
166     \providecommand*{\hlbar}[1]{\overline{\vphantom{t}#1}}}{
167     \@mwarning{hlbar} }
168 \newcommand\@eqdef{\stackrel{\mathup{def}}{=}}
169 \newcommand\@@eqdef{\stackrel{\Delta}{=}}
170 \@mmacro{eqdef}{\@ifstar{\@@eqdef}{\@eqdef}}
171 \@mmacro{unbr}{\underbrace}
```

```
\@mmacro{iif}{if and only if\xspace}

\@mmacro{then}{\ \Longrightarrow \ \mbox{} }
```

Without \mbox{}, the space produced by \ would be suppressed in tables.

```
\@ifundefined{txt}{
    \providecommand*{\txt}[1]{\quad\text{#1}\quad} }{
    \@mwarning{txt} }
\@mmacro{mul}{\mathord{\times}}
\providecommand\paren{\PackageWarning{mismath}{Command
    \bslash paren is no longer supported}}
\@ifundefined{pow}{
    \providecommand*{\pow}[2]{\left( #1 \right)^{\!#2}} }{
    \@mwarning{pow} }
\@ifundefined{abs}{
    \providecommand*{\abs}[1]{\left\vert#1\right\vert} }{
    \@mwarning{abs} }
\@ifundefined{lfrac}{
    \providecommand*{\lfrac}[2]{\frac{\;#1\;}{\;#2\;}} }{
    \@mwarning{lfrac} }

\newcommand{\systemstretch}{1.2}
\newcommand{\systemsep}{\medspace}
\newenvironment{system}[1][l]{
    \renewcommand{\arraystretch}{\systemstretch}
    \setlength{\arraycolsep}{0.15em}
    \left\{\begin{array}{@{\systemsep}#1@{}} %
}{\end{array}\right.}

\newenvironment{spmatrix}{
    \left(\begin{smallmatrix}
}{\end{smallmatrix}\right)}

\newenvironment{mathcols}{% needs multicol package
    \renewcommand{\columnseprule}{0.1pt}
    \begin{multicols}{2}
        \par\noindent\hfill
        \begin{math}\begin{aligned}\displaystyle
}{%
        \end{aligned}\end{math} \hfill\mbox{}
    \end{multicols}
}
\newcommand{\changecol}{%
    \end{aligned}\end{math} \hfill\mbox{}
    \par\noindent\hfill
    \begin{math}\begin{aligned}\displaystyle
}
```

References

[1] *Typesetting mathematics for science and technology according to ISO 31/XI*, Claudio Beccari, TUGboat Volume 18 (1997), No. 1. http://www.tug.org/TUGboat/tb18-1/tb54becc.pdf.

[2] *Typefaces for Symbols in Scientific Manuscripts*, https://www.physics.nist.gov/cuu/pdf/typefaces.pdf.

[3] *Guide for the Use of the International System of Units (SI)*, NIST (National Institute of Standards and Technology), updated March 4, 2020 https://www.nist.gov/pml/special-publication-811.

[4] *On the Use of Italic and up Fonts for Symbols in Scientific Text*, I.M. Mills and W.V. Metanomski, ICTNS (Interdivisional Committee on Terminology, Nomenclature and Symbols), dec 1999, https://old.iupac.org/standing/idcns/italic-roman_dec99.pdf.

[5] *esvect – Typesetting vectors with beautiful arrow with LaTeX 2ε*, Eddie Saudrais, CTAN, v1.3 2013/07/11.

[6] *amsmath – AMS mathmatical facilities for LaTeX*, Frank Mittelbach, Rainer Schöpf, Michael Downes, Davis M. Jones, David Carlisle, CTAN, v2.17n 2022/04/08.

[7] *The mathtool package*, Morten Høgholm, Lars Madsen, CTAN, v1.29 2022/06/29.

[8] *The fixmath package for LaTeX 2ε*, Walter Schmidt, CTAN, v0.9 2000/04/11.

[9] *isomath – Mathematical style for science and technology*. Günter Milde, CTAN, v0.6.1 2012/09/04.

[10] *PM-ISOmath, The Poor Man ISO math bundle*, the pm-isomath package by Claudio Beccari, CTAN, v1.2.00 2021/08/04.

[11] *The upgreek package for LaTeX 2ε*, Walter Schmidt, CTAN, v2.0 2003/02/12.

[12] *The mathdesign package*, Paul Pichaureau, CTAN, v2.31 2013/08/29.

[13] *The textalpha package* (part of the greek-fontenc bundle), Günter Milde, CTAN, v2.1 14/06/2022.

[14] *Kp-Fonts – The Johannes Kepler project*, Christophe Caignaert, CTAN, v3.34 20/09/2022.

[15] Fourier-GUTenberg, Michel Bovani, CTAN, v1.3 30/01/2005.

[16] *The lgrmath package*, Jean-François B., CTAN, v1.0 2022/11/16.

[17] *The interval package*. Lars Madsen, CTAN, v0.4 2019/03/06.

[18] *L'extension pour TEX et LATEX systeme*, Christian Tellechea, CTAN v0.32 2019/01/13.

[19] *The spalign package*, Joseph Rabinoff, CTAN, 2016/10/05.

[20] *L'extension frenchmath*, Antoine Missier, CTAN, v2.3 2022/12/26.

[21] *Intelligent brackets – The ibrackets package* Antoine Missier, CTAN, v1.0, 2022/12/19.

[22] *The Not So Short Introduction to LATEX2ε*, the lshort package by Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl, CTAN, v6.4 2021/04/09. http://tug.ctan.org/info/lshort/english/lshort.pdf.

[23] *The LATEX Companion*, Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.