

mismath

Miscellaneous mathematical macros*

Antoine Missier
`antoine.missier@ac-toulouse.fr`

May 22, 2019

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According to the International Standards ISO 31-0:1992 to ISO 31-13:1992, superseded by ISO 80000-2:2009, mathematical constants e , i , π should be typeset in upright shape and not in italic (sloping shape) like variables (see [1] [2] [3]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] [3], they are often represented with arrows (particularly in school documents or in physics). To draw pretty arrows above vectors, we use `esvect` package by Eddie Sautrais [8] and we provide a few more macros related to vectors with arrows, in particular to improve the typesetting of the norm: $\|\overrightarrow{AB}\|$ instead of L^AT_EX version $\|\overrightarrow{AB}\|$ which is not adjusted, or worse $\|\overrightarrow{AB}\|$.

The package also provides other macros for:

- some standard operator names,
- a few useful aliases,
- improving some spacings 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

*This document corresponds to `mismath` v1.4, dated 2019/05/22. Thanks to François Bastouil for help in English translation.

for the name of the command (assuming the other package supports it), or use `\let\<command>\relax` before loading `mismath`.

`[<options>]{mismath}`

The `amsmath` package is loaded by `mismath` without option. For using `amsmath` with options (see [5]), these options can be added when calling `mismath`, or `amsmath` has to be loaded with the required options before `mismath`.

Two other usefull packages are loaded: `mathtools` by Morten Høgholm and Lars Madsen [6] and `mathfixs` by Niklas Beiser with `frac` and `root` options [7].

A recommendation, seldom observed, is to typeset uppercase Greek letters in italic shape like other variables [3]. This is automatically done with `fixmath` package by Walter Schmidt [10] or `isomath` by Günter Milde [11] or you can do `\ProvideMathFix{greekcaps}` in the preamble, thanks to `mathfixs` package. But this feature is not implemented in `mismath` because this rule is conflicting to the one used for instance in France where all mathematics capitals have to be typeset in upright shape¹. The choice of loading or not one of these packages remains thus to the user.

2 Usage

2.1 Mathematical constants

`\mathup` 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

`\e` `\mathrm{e}` or `\mathrm{i}` (or better `\mathup{e}` and `\mathup{i}`²), the pack-

`\i` age provides `\e` command for the base of natural logarithm and `\i` or `\j` for

`\j` imaginary numbers. One can notice that `\i` and `\j` already exist in L^AT_EX: using in LR mode, they produce “i” and “j” without the point so one can place accents on them, and in mathematical mode they produce “Latex warning: Command invalid in math mode”. Redefining `\i` and `\j` only concerns mathematical mode.

`\enumber` Nevertheless, it can be tiresome to type a lot of backslashes in a document

`\inumber` with many formulas containing e or i. So a way is proposed here to free of it

`\jnumber` by placing `\enumber`, `\inumber` or `\jnumber` in the preamble: e, i or j will then automatically be set in upright shape 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, one can still get italicized *e*, *i* or *j* with L^AT_EX command `\mathit` or `\mathnormal`. Of course, this does not fully comply with L^AT_EX 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; besides anyone is free to use them or not.

¹The package `frenchmath` [15] 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`.

`\pinumber[⟨font⟩]` Mathematical constant π should also be typeset in upright shape, which differs from italicized π (see [3] and [4]). This recommendation is even less observed than the one concerning e and i [1]. The `upgreek` package by Walter Schmidt [9] makes it possible to typeset greek letters in upright font by using commands such as `\upalph`, `\upbeta`,... To avoid typing a lot of `\uppi`, one can use the `\pinumber` macro, which has to be put in the preamble. This command loads the `upgreek` package with an optional `⟨font⟩` argument: `Symbol` (by default), `Euler` or `Symbolsmallscale` (see [9]). It also redefines the `\pi` command to typeset all `\pi` in the selected upright font.

By activating `\enumber`, `\inumber` and `\pinumber` in the preamble, you can get for instance:

$$e^{i\pi} = -1 \qquad e^{i\pi} = -1$$

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

2.2 Vectors

`\vect` By default, the `\vect` command³, produces vectors with arrows (thanks to `esvect` package by Eddie Soudrais⁴) which are much more elegant than those produced by L^AT_EX `\overrightarrow` command (giving \overrightarrow{AB}). The `esvect` package has an optional argument (one letter between `a` and `h`) defining the required type of arrow (see [8]). 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`, for instance `\usepackage[d]{esvect}` will give the arrows produced by default in [8].

`\boldvect` `\vect` makes also possible to typeset vector's names using bold italic (according to ISO recommendation [4]) rather than arrows. For this, calling `\boldvect` will modify the behavior of `\vect`:

$$\begin{aligned} \text{\texttt{[\boldvect \vect{v} \\}} &= \text{\texttt{\lambda \vect{e}_x + \mu \vect{e}_y. \texttt{]}}} & v = \lambda e_x + \mu e_y. \end{aligned}$$

`\boldvectcommand` By default `\boldvect` uses the `\boldsymbol` command⁵ from `amsbsy` package, loaded by `amsmath`. But other packages producing bold italic can be preferred, e.g. `\bm` from `bm` package or `\mathbf` from `fixmath` package (also implemented in `mathfix`⁶) or `\mathbf` from `isomath`. For that, redefine `\boldvectcommand`: for instance `\renewcommand\boldvectcommand{\mathbf}`.

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

³As for many macros of this package, the definition will take effect only if this macro is not defined before by another package.

⁴`esvect` provides `\vv` macro used by `\vect`.

⁵`\mathbf` gives upright bold font, even if used in combination with `\mathit`.

⁶To activate a feature of `mathfix` like `mathbold`, do `\ProvideMathFix{mathbold}`.

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 h) to avoid inelegants effects:

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

The `\boldvect` switch has no effect on the `\hvect` macro which always typesets arrows on vectors (with the `\vv` command from `esvect` package).

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

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

`\norm` The norm of a vector is classically produced by delimiters `\lVert` and `\rVert` (rather than `\|`) or `\left\Vert` and `\right\Vert` for delimiters adapting to the content. Unfortunately, these delimiters are always vertically centred, relatively to the middle of the base line, whereas vectors with arrows are asymetrics objects, the height above the middle of the base line being superior to the depth under it. The code `$_\norm{\vec{h}}$` raises the double bar to produce $\|\vec{h}\|$. Let's notice that the height of the bars don't adjust to content, but however to context: main text, subscripts or exponents.

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] [3] [16]). To get this, use the `\di` command. In the first following example, notice the thin spaces before the d , as for classic function's names; in the second one, spaces before fractions have been enlarged by the `mathfixs` package:

$$\begin{aligned} & \int \int xy \, \mathrm{d}x \, \mathrm{d}y \\ & m \frac{\mathrm{d}^2 x}{\mathrm{d}t^2} + h \frac{\mathrm{d}x}{\mathrm{d}t} + kx = 0 \end{aligned}$$

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

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

`\P` To refer to probability⁷ and expectation the proper use is to typeset capital letters P , E in upright shape as for any standard function identifier. This is obtained with `\P` and `\E`. Variance is normally denoted by Var (see further), but in some countries we can find 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}`⁸ brings the previous “openwork” letters. `\mathbb` comes from `amsfonts` package (loaded by `amssymb` but also available standalone) which has to be called in the preamble.

The following operator names are also defined in `mismath`:

<code>\adj</code>	adj	<code>\erf</code>	erf	<code>\rank</code>	rank
<code>\Aut</code>	Aut	<code>\grad</code>	$\overrightarrow{\operatorname{grad}}$	<code>\Re</code>	Re
<code>\Conv</code>	Conv	<code>\id</code>	id	<code>\rot</code>	$\overrightarrow{\operatorname{rot}}$
<code>\cov</code>	cov	<code>\Id</code>	Id	<code>\sgn</code>	sgn
<code>\Cov</code>	$\overrightarrow{\operatorname{Cov}}$	<code>\im</code>	im	<code>\spa</code>	span
<code>\curl</code>	$\overrightarrow{\operatorname{curl}}$	<code>\Im</code>	Im	<code>\tr</code>	tr
<code>\divg</code>	div	<code>\lb</code>	lb	<code>\Var</code>	Var
<code>\End</code>	End	<code>\lcm</code>	lcm	<code>\Zu</code>	Z

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 covariance and the identity function, two notations are provided, with or without a first capital letter, because they are both very frequent. On the other hand, “im” stands for the image of a linear transformation (like “ker” for the kernel) but “Im” is the imaginary part of a complex number. Notice that `\div` and `\span` already exist and haven’t been redefined, therefore `\divg` and `\spa` macros; `\Z` is used otherwise (see further), therefore `\Zu`, to designate the center of a group: $Z(G)$ (from German Zentrum).

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

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

<code>\arccot</code>	arccot	<code>\arsinh</code>	arsinh	<code>\arcoth</code>	arcoth
<code>\sech</code>	sech	<code>\arcosh</code>	arcosh	<code>\arsech</code>	arsech
<code>\csch</code>	csch	<code>\artanh</code>	artanh	<code>\arcsch</code>	arcsch

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

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

⁷ \LaTeX provides also `\Pr` which gives Pr.

⁸As for `\boldvect` and `\arrowvect`, effect is local to the container environment.

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**, “openwork” letters ($\mathbb{R}, \mathbb{Z}, \dots$) being restricted to writing at blackboard [16]; and likewise to designate a field: **F** or **K** (Körper in German). We get these symbols with the following macros:

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

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

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

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

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

`\eqdef` The `\eqdef` macro writes equality symbol topped with “def” (thanks to the L^AT_EX command `\stackrel`):

$$\begin{aligned} & \$ \backslash e^{\backslash i \backslash theta} \backslash eqdef & e^{i\theta} \stackrel{\text{def}}{=} \cos \theta + i \sin \theta \\ & \backslash cos \backslash theta + \backslash i \backslash sin \backslash theta \$ \end{aligned}$$

`\unbr` `\unbr` is an alias for `\underbrace`⁹, making source code more compact.

$$\begin{aligned} & \$ (QAP)^n = \backslash unbr \{ QAP \backslash mul \ QAP \backslash mul \\ & \backslash cdots \backslash mul \ QAP \}_n \text{\textit{ times}} \$ \end{aligned} \quad (QAP)^n = \underbrace{QAP \times QAP \times \dots \times QAP}_{n \text{ times}}$$

⁹The `mathtools` package by Morten Høgholm and Lars Madsen [6] provides a new improved version of `\underbrace` command (as many other usefull macros); it is loaded by `mismath`.

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

2.5 Improving some spacings in mathematical formulas

`\mul` The multiplication symbol obtained with `\times` produces the same spacing than addition or subtraction operators, whereas division obtained with `/` is tighten around its arguments. 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`:

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

`\then` The `\then` macro produces the symbol \implies surrounded by large spaces as the standard macro `\iff` does it with \iff . In a similar way, `\txt` based on the `\text` macro (from the `amstext` package, automatically 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`
 $\ln x = a \implies x = e^a$ rather than $\ln x = a \implies x = e^a$

`\pow` When typesetting an exponent after a closing *big* parenthesis produced by `\right)`, the exponent is little to far from the parenthesis. The command `\pow{\langle expr \rangle}{\langle pow \rangle}` sets $\langle expr \rangle$ between parentheses and puts the exponent $\langle pow \rangle$ slightly closer to the right parenthesis¹⁰. Compare:

$$e^a \sim \left(1 + \frac{a}{n}\right)^n \quad \text{and} \quad e^a \sim \left(1 + \frac{a}{n}\right)^n.$$

`\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\lvert ... \right\rvert` or, more simply, the `\abs{...}` command which is equivalent¹¹.

`\lfrac` This macro behaves like `\frac`¹² but with medium spaces around the arguments, so the corresponding fraction bar is perceptibly a little bit longer:

$$\left[\frac{\lvert z \rvert}{\lvert \frac{\lvert z_1 - z_2 \rvert}{\lvert z_1 + z_2 \rvert}} \right]$$

`[]` Brackets symbols `[` and `]` have been redefined for mathematical mode because, in standard L^AT_EX, the space before them can be unsuitable¹³:

$$\left\{ \begin{array}{ll} x \in]0, \pi[\cup]2\pi, 3\pi[& \text{without mismatch} \\ x \in]0, \pi[\cup]2\pi, 3\pi[& \text{with mismatch} \end{array} \right.$$

¹⁰This macro gives bad results with normal sized parenthesis.

¹¹Another solution is to define `\abs` with the `\DeclarePairedDelimiter` command from the `mathtool` package [6].

¹²`\frac` is redefined by `mathfixs` package to produce better surrounding space.

¹³The `interval` package [12] gives another solution, less direct, based on an `\interval` macro.

In our code, [and] symbols are not defined anymore as delimiters. One can regret it because a line break could occur between the two, but in addition to the fact that it works very well like that for spaces (because these symbols are most of the time preceded or followed by relational, binary or punctuation symbols), it is always possible to transform them into delimiters with `\left` and `\right`¹⁴.

2.6 Environments for systems of equations and small matrices

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

$$\begin{array}{l} \text{\$}\backslash\text{begin}\{\text{system}\} \\ \quad x=1+2t \ \backslash\backslash \ y=2-t \ \backslash\backslash \ z=-3-t \\ \text{\$}\backslash\text{end}\{\text{system}\} \end{array} \quad \left\{ \begin{array}{l} x = 1 + 2t \\ y = 2 - t \\ z = -3 - t \end{array} \right.$$

\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 is logical 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 reduce, for instance: **\renewcommand{\systemsep}{\thinspace}**, or enlarged with **\thickspace** (and with **\renewcommand\systemsep{}** we get back to 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 L^AT_EX: **\begin{system}[c1]** produces a two-column system, the first one being centred, the second being left aligned, such as in the following example:

$$\begin{array}{l} \text{\$}\backslash\text{begin}\{\text{system}\}[c1] \\ \quad y \ \& \ = \text{\dfrac{1}{2}}x - 2 \ \backslash\backslash[1ex] \\ \quad (x,y) \ \& \ \neq \ (0,-2) \\ \text{\$}\backslash\text{end}\{\text{system}\} \end{array} \quad \left\{ \begin{array}{l} y = \frac{1}{2}x - 2 \\ (x,y) \neq (0,-2) \end{array} \right.$$

\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 a carriage return with a spacing option such as it has been done above with **\backslash[1ex]**.

Another example with **\begin{system}[r1@{\quad}l]**¹⁵:

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

¹⁴Is L^AT_EX definition of [as **mathopen** really appropriate where this symbol could almost also logically have been defined as **mathclose**?

¹⁵**@{...}** sets inter-column space.

Let's mention the `systeme` package [13] which deals with linear systems with a lighter syntax and automatic alignments on $+$, $-$, $=$, and also the `spalign` package [14] which moreover produces nice alignments for matrices (with spaces and semicolons as delimiters).

`spmatrix` `amsmath` package provides various environments to typeset matrices: for instance `pmatrix` that surrounds the matrix with parenthesis or `smallmatrix` that 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 $\vec{u}\begin{pmatrix}-1\\2\end{pmatrix}$.

`mathtools` package enhance `amsmath` matrices environments and provides also a small matrix environment with parenthesis. Furthermore, with starred version `\begin{psmallmatrix*}[\langle col \rangle]`, 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 $\vec{u}\begin{pmatrix}-1\\2\end{pmatrix}$.

2.7 Displaymath in double columns

`mathcols` `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 `multicol` package in the preamble.

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

`\changeacol` The `\changeacol` macro causes a change of column; alignment is produced using 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} \\
\changeacol
    & \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{mathcols}
```

3 Implementation

```
1 \DeclareOption*{\PassOptionsToPackage{\CurrentOption}{amsmath}}
2 \ProcessOptions \relax
```

```

3 \ifpackage{amsmath}\RequirePackage{amsmath}
4 \ifpackage{esvect}\RequirePackage{esvect}
5 \RequirePackage{ifthen}
6 \RequirePackage{xspace}
7 \RequirePackage{mathtools}
8 \RequirePackage{frac,root}{mathfixs}

```

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

The three following internal commands are meta commands for a conditional macro definition with warning message if the macro already exists. The `\bslash` macro used inside `\@mwarning` comes from `doc.sty` package by Frank Mittelbach. It can also be used in other documents instead of `\textbackslash` (which doesn’t work here).

```

9 {\catcode'\|=z@ \catcode'\=12 \gdef\bslash{\} % the \bslash command
10 \newcommand\@mwarning[1]{
11   \PackageWarning{mismath}{
12     Command \bslash #1 already exist and will not be redefined
13   }
14 }
15 \newcommand\@mmacro[2]{
16   \ifundefined{#1}{
17     \expandafter\def\csname #1\endcsname{#2}
18   }\@mwarning{#1}}
19 }
20 \newcommand\@moperator[3][\% this macro is ugly, TODO: by default #1=#3
21   \ifthenelse{\equal{#1}{}}{
22     \ifundefined{#3}{
23       \DeclareMathOperator{#2}{#3}
24     }\@mwarning{#3}}
25   ){
26     \ifundefined{#1}{
27       \DeclareMathOperator{#2}{#3}
28     }\@mwarning{#1}}
29   }
30 }
31

```

To work correctly with the `beamer` package, we did not use `\mathrm` but `\mathup` (based on `\operatorfont` from `mathopn` package) to produce the correct upright shape font. This command works also fine with other sans serif fonts like `cmbright`. Moreover for `beamer`, `\enumber` must use the family default font defined by `beamer` package (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 `hyperref` package which overwrites the `\i` definition.

```

32 \providecommand{\mathup}[1]{\operatorfont #1}
33 \@mmacro{e}{\mathup{e}}
34 \AtBeginDocument{\let\oldi\i \let\oldj\j}

```

```

35 \renewcommand{\i}{\TextOrMath{\oldi}{\mathup{i}}}
36 \renewcommand{\j}{\TextOrMath{\oldj}{\mathup{j}}}
37 }
38
39 \DeclareSymbolFont{UpSh}{\encodingdefault}{\familydefault}{m}{n}
40 \newcommand{\enumber}{
41   \AtBeginDocument{\DeclareMathSymbol{e}\mathalpha{UpSh}{'e}}
42 }
43 \newcommand{\inumber}{
44   \AtBeginDocument{\DeclareMathSymbol{i}\mathalpha{UpSh}{'i}}
45 }
46 \newcommand{\jnumber}{
47   \AtBeginDocument{\DeclareMathSymbol{j}\mathalpha{UpSh}{'j}}
48 }
49 \newcommand*{\pinumber}[1][Symbol]{
50   \@ifpackageloaded{upgreek}{\usepackage[#1]{upgreek}}
51   \let\itpi\pi
52   \renewcommand{\pi}{\uppi}
53 }
54
55 \newboolean{arrowvect}
56 \setboolean{arrowvect}{true}
57 \newcommand{\arrowvect}{\setboolean{arrowvect}{true}}
58 \newcommand{\boldvect}{\setboolean{arrowvect}{false}}
59 \newcommand{\boldvectcommand}{\boldsymbol} % needs bm package
60 \@mmacro{vect}{\ifthenelse{\boolean{arrowvect}}{\vv}{\boldvectcommand}}
61 \newcommand*{\hvect}[1]{\vv{\vphantom{h}#1}}
62 \newcommand*{\hvec}[1]{\vec{\vphantom{t}#1}}
63
64 \newcommand*{\@norm}[1]{
65   \mbox{\raisebox{1.75pt}{\bigl\Vert$}} #1
66   \mbox{\raisebox{1.75pt}{\bigl\Vert$}} }
67 % works better than with relative length
68 \newcommand*{\@@norm}[1]{
69   \mbox{\footnotesize\raisebox{1pt}{\bigl\Vert$}} #1
70   \mbox{\footnotesize\raisebox{1pt}{\bigl\Vert$}} }
71 \newcommand*{\@@@norm}[1]{
72   \mbox{\tiny\raisebox{1pt}{\bigl\Vert$}} #1
73   \mbox{\tiny\raisebox{1pt}{\bigl\Vert$}} }
74 \providecommand*{\norm}[1]{
75   \mathchoice{\@norm{#1}}{\@norm{#1}}{\@@norm{#1}}{\@@@norm{#1}} }
76
77 \newcommand{\di}{\mathop{}!\mathup{d}}
78 \newcommand{\probastyle}{}
79 \let\Par\P % end of paragraph symbol
80 \renewcommand{P}{\operatorname{\probastyle{P}}}
81 \@mmacro{E}{\operatorname{\probastyle{E}}}
82 \@mmacro{V}{\operatorname{\probastyle{V}}}
83 \newcommand{\PEupright}{
84   \AtBeginDocument{% necessary for working with beamer

```

```

85         \DeclareMathSymbol{P}\mathalpha{UpSh}{‘P}
86         \DeclareMathSymbol{E}\mathalpha{UpSh}{‘E}
87     }
88 }
89
90 \@operator{\adj}{adj}
91 \@operator{\Aut}{Aut}
92 \@operator{\Conv}{Conv}
93 \@operator{\cov}{cov}
94 \@operator{\Cov}{Cov}
95 \@mmacro{curl}{\operatorname{\vect{\mathup{curl}}}}
96 \@operator{divg}{\divg}{div}
97 \@operator{\End}{End}
98
99 \@operator{\erf}{erf}
100 \@mmacro{grad}{\operatorname{\vect{\mathup{grad}}}}
101 \@operator{\id}{id} % mathop or mathord ?
102 \@operator{\Id}{Id}
103 \@operator{\im}{im}
104 \let\oldIm\Im \renewcommand{\Im}{\operatorname{Im}}
105 \@operator{\lb}{lb}
106 \@operator{\lcm}{lcm}
107
108 \@operator{\rank}{rank}
109 \let\oldRe\Re \renewcommand{\Re}{\operatorname{Re}}
110 \@mmacro{rot}{\operatorname{\vect{\mathup{rot}}}}
111 \@operator{\sgn}{sgn}
112 \@operator[spa]{\spa}{span}
113 \@operator{\tr}{tr}
114 \@operator{\Var}{Var}
115 \@operator[Zu]{\Zu}{Z}
116
117 \@operator{\arccot}{arccot}
118 \@operator{\sech}{sech}
119 \@operator{\csch}{csch}
120 \@operator{\arsinh}{arsinh}
121 \@operator{\arcosh}{arcosh}
122 \@operator{\artanh}{artanh}
123 \@operator{\arcoth}{arcoth}
124 \@operator{\arsech}{arsech}
125 \@operator{\arcsch}{arcsch}
126
127 \@operator[big0]{\big0}{\mathcal{0}}
128 \@operator[bigo]{\bigo}{0}
129 \@operator[lito]{\lito}{o}
130
131 \newcommand{\mathset}{\mathbf}
132 \@mmacro{R}{\ensuremath{\mathset{R}}\xspace}
133 \@mmacro{C}{\ensuremath{\mathset{C}}\xspace}
134 \@mmacro{N}{\ensuremath{\mathset{N}}\xspace}

```

```

135 \@macro{Z}{\ensuremath{\mathset{Z}}\xspace}
136 \@macro{Q}{\ensuremath{\mathset{Q}}\xspace}
137 \@macro{F}{\ensuremath{\mathset{F}}\xspace}
138 \@macro{K}{\ensuremath{\mathset{K}}\xspace}
139
140 \@macro{ds}{\displaystyle}
141 \@macro{dlim}{\lim\limits}
142 \@macro{dsum}{\sum\limits}
143 \@macro{dprod}{\prod\limits}
144 \@macro{dcup}{\bigcup\limits}
145 \@macro{dcap}{\bigcap\limits}
146 \@macro{lbar}{\overline}
147 \providecommand*{\hlbar}[1]{\overline{\vphantom{h}#1}}
148 \@macro{eqdef}{\stackrel{\mathup{def}}{=}}
149 \@macro{unbr}{\underbrace}
150 \@macro{iif}{if and only if\xspace}
151
152 \@macro{mul}{\mathord{\times}}
153 \@macro{then}{\ \Longrightarrow \ \mbox{ } }

```

Without `\mbox{}` the space produced by `\` would be suppressed in tabulars.

```

154 \providecommand*{\txt}[1]{\quad\text{#1}\quad}
155 \providecommand*{\pow}[2]{\left( #1 \right)^{\!#2}}
156 \providecommand*{\abs}[1]{\left|\text{#1}\right|}
157 \providecommand*{\lfrac}[2]{\frac{:#1:}{:#2:}}
158 \DeclareMathSymbol{[]\mathord{UpSh}{093} % originally \mathclose
159 \DeclareMathSymbol{[]\mathord{UpSh}{091} % originally \mathopen
160
161 \newcommand{\systemstretch}{1.2}
162 \newcommand{\systemsep}{\medspace}
163 \newenvironment{system}[1][1]{
164     \renewcommand{\arraystretch}{\systemstretch}
165     \setlength{\arraycolsep}{0.15em}
166     \left\{\begin{array}{@{\systemsep}#1@{}} %
167 }\end{array}\right.}
168
169 \newenvironment{spmatrix}{
170     \left(\begin{smallmatrix}
171 }\end{smallmatrix}\right)}
172
173 \newenvironment{mathcols}{% needs multicol package
174     \renewcommand{\columnseprule}{0.1pt}
175     \begin{multicols}{2}
176         \par\noindent\hfill
177         \begin{math}\begin{aligned}\displaystyle
178 }{%
179         \end{aligned}\end{math} \hfill\mbox{}
180     \end{multicols}
181 }
182 \newcommand{\changepcol}{%

```

```

183 \end{aligned}\end{math} \hfill\mbox{}
184 \par\noindent\hfill
185 \begin{math}\begin{aligned}\displaystyle
186 }
```

References

- [1] *Typesetting mathematics for science and technology according to ISO 31/XI*, Claudio Beccari, TUGboat Volume 18 (1997), No. 1.
- [2] *Typefaces for Symbols in Scientific Manuscripts*.
<https://www.physics.nist.gov/cuu/pdf/typefaces.pdf>.
- [3] *On the Use of Italic and up Fonts for Symbols in Scientific Text*, I.M. Mills and W.V. Metanomski, ICTNS (Interdivisional Committee on Nomenclature and Symbols), dec 1999.
- [4] *ISO 80000-2*. https://en.wikipedia.org/wiki/ISO_80000-2
- [5] *The amsmath package*. Frank Mittelbach, Rainer Schöpf, Michael Downes, Davis M. Jones, David Carlisle, CTAN, v2.17b 2018/12/01.
- [6] *The mathtool package*. Morten Høgholm, Lars Madsen, CTAN, v1.21 2018/01/08.
- [7] *The mathfixs Package*. Niklas Beisert, CTAN, v1.01 2018/12/30.
- [8] *Typesetting vectors with beautiful arrow with L^AT_EX 2_ε*. esvect package by Ed-die Saudrais, CTAN, v1.3 2013/07/11.
- [9] *The upgreek package for L^AT_EX 2_ε*, Walter Schmidt, CTAN, v2.0 2003/02/12.
- [10] *The fixmath package for L^AT_EX 2_ε*, Walter Schmidt, CTAN, v0.9 2000/04/11.
- [11] *isomath. Mathematical style for science and technology*. Günter Milde, CTAN, v0.6.1 04/06/2012.
- [12] *The interval package*. Lars Madsen, CTAN, v0.3 2014/08/04.
- [13] *L'extension pour T_EX et L^AT_EX système*. Christian Tellechea, CTAN v0.32 2019/01/13.
- [14] *The spalign package*. Joseph Rabinoff, CTAN, 2016/10/05.
- [15] *L'extension frenchmath*. Antoine Missier, CTAN, v1.2 2019/04/27.
- [16] *The Not So Short Introduction to L^AT_EX 2_ε*. lshort package by Tobias Oetiker, Hubert Partl, Irene Hyna and Elisabeth Schlegl, CTAN, v6.2 2018/02/28.
- [17] *The L^AT_EX Companion*. Frank Mittelbach, Michel Goossens, Johannes Braams, David Carlisle, Chris Rowley, 2nd edition, Pearson Education, 2004.