

The **skmath** package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.4

Abstract The skmath package provides improved and new math commands for superior typesetting with less effort.

1 Introduction

This package intends to provide helpful (re-)definitions of commands related to typesetting mathematics, and specifically typesetting them in a more intuitive, less verbose and more beautiful way. It was originally not intended for use by the public, and as such there may be incompatibilities with other packages of which I am not aware, but I figured it could be useful to other people as well.

2 Usage

2.1 Options

As of version v0.4, the package provides two key-value options.

`commonsets` `true, false` (false)

Optionally define `\N`, `\Z`, `\Q`, `\R` and `\C` as blackboard variants of the respective letters, to represent the common sets of numbers.

`notation` `iso, english, german, legacy` (legacy)

This option controls the style of a few typographic elements that differ between countries and standards (such as the style of integrals, derivatives and greek letters).

*Available on <http://www.ctan.org/pkg/skmath>.

†Development version available on <https://github.com/urdh/skmath>.

2.2 New commands

The package defines a number of new commands that aid in typesetting certain mathematical formulae.

`\N`
`\Z`
`\Q`
`\R`
`\C`

These commands are only available if the `commonsets` option is given. They typeset the set of natural, integer, rational, real and complex numbers respectively.

Example:

$\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}.$

```
\begin{equation*}
  \mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}.
\end{equation*}
```

`\ii`
`\jj`

These commands typeset the imaginary unit (either i as used in mathematics or j as used in electrotechnology). While normal use of the package simply results in italic characters, setting the `notation` option to `iso` will set these upright.

`\norm` $\{\langle expression \rangle\}$
`\abs` $\{\langle expression \rangle\}$

The commands `\norm` and `\abs`, quite expectedly, typeset the norm and absolute value of an expression, respectively. They have one mandatory argument (the expression), and different norms can be achieved by appending a subscript after the argument of `\norm`.

Example:

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}$$

```
\begin{equation*}
  \norm{\vec{x}}_p =
  \left(\sum_{i=1}^n \abs{x_i}^p\right)^{\sfrac{1}{p}}
\end{equation*}
```

\d $\langle \text{variable} \rangle$

There is also a command **\d**, with one mandatory argument, that typesets the differential part of an integral.

Example:

$$\int_{\mathbb{R}} \frac{\sin(x)}{x} dx$$

```
\begin{equation*}
  \int_{\mathbb{R}} \! \! \! \! \frac{\sin{x}}{x} \, \mathrm{d}{x}
\end{equation*}
```

\pd $\star \langle \text{function} \rangle \langle \text{var} \rangle, \langle \text{var} \rangle, \dots$

This macro typesets a partial derivative. The starred variant typesets derivatives as subscripts, i.e. f_{xxy} , while the unstarred variant typesets full fractions:

Example:

$$\frac{\partial^{m+n} f}{\partial x^m \partial y^n}$$

```
\begin{equation*}
  \pd{f}{x^m,y^n}
\end{equation*}
```

As the example shows, the comma-separated list of variables also supports superscripts to denote the number of derivatives, and the sum of the variables is automatically calculated.

\E {*expression*}

The command **\E** typesets the expectation of a random variable.

Example:

$$E[\hat{\mu}] = \mu$$

```
\begin{equation*}
  \E{\hat{\mu}} = \mu
\end{equation*}
```

\P {*expression*}\biven{*expression*}

The **\P** command typesets a probability. The **\biven** command can be used to typeset conditional probabilities, within **\P**.

Example:

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)}$$

```
\begin{equation*}
  \P{A\biven B} =
  \frac{\P{B\biven A}\P{A}}{\P{B}}
\end{equation*}
```

\var {*expression*}

\cov {*expression*}{*expression*}

The commands **\var** and **\cov** typeset the variance and covariance of an expression.

Example:

$$\begin{aligned} \text{Var}(X) &= E[(X - \mu)^2] \\ \text{Cov}(X, Y) &= E[XY] - E[X]E[Y] \end{aligned}$$

```
\begin{gather*}
  \var{X} = \E{(X-\mu)^2} \\
  \cov{X}{Y} = \E{XY} - \E{X}\E{Y}
\end{gather*}
```

2.3 Improved commands

In addition to adding new commands, this package also redefines already existing commands in a mostly backwards-compatible way to improve their usefulness.

\sin [*power*] {*expression*}

\arcsin {*expression*}

\cos [*power*] {*expression*}

\arccos {*expression*}

\tan [*power*] {*expression*}

\arctan {*expression*}

\cot [*power*] {*expression*}

\sinh [*power*] {*expression*}

\cosh [*power*] {*expression*}

\tanh [*power*] {*expression*}

The trigonometric functions have been redefined to typeset more easily. They typeset *expression* as an argument of the expression, and (if applicable) *power* as a superscript between the function and its argument, e.g. $\sin^2(\phi)$. When the argument is empty, no parentheses are emitted: \cos .

\ln {*expression*}

The natural logarithm macro **\ln** has also been redefined to require an argument which is typeset as the argument of the logarithm.

\log [*base*] {*expression*}

The related macro **\log** is redefined in a similar way, but also accepts an optional argument denoting the base of the logarithm: $\log_2(x)$. As with the trigonometric functions, no parentheses are emitted if the mandatory argument is empty: \log .

\exp *{<expression>}

The exponential, **\exp**, is redefined to typeset its argument as a superscript of e in some display styles, and as an argument of `exp` otherwise:

$$e^{\sqrt{2}\exp(x)}$$

Additionally, it is possible to force the `exp` mode by using the starred variant.

\min * [<domain>] {<expression>}
\argmin * [<domain>] {<expression>}
\max * [<domain>] {<expression>}
\argmax * [<domain>] {<expression>}
\sup * [<domain>] {<expression>}
\inf * [<domain>] {<expression>}

The maximum/minimum macros have been redefined in a manner similar to the trigonometric functions. They typeset <expression> inside curly brackets (the starred version omits the brackets), with the optional <domain> typeset in a subscript after the operator (e.g. $\min_{x \in \mathbb{R}_+} f(x)$). Argument variants are also provided, and the <expression> is centered underneath the operator if possible:

$$\arg \min_{x \in \mathbb{R}_+} f(x)$$

2.4 Stylistic changes

Some commands have been redefined in a completely backwards-compatible way to improve the end result of their typesetting.

\frac {<numerator>} {<denominator>}

The **\frac** command has been changed to improve typesetting, allowing `displaystyle` math in some settings.

\bar { $\langle expression \rangle$ }
\vec { $\langle expression \rangle$ }

The **\bar** command has been changed to cover the entire $\langle expression \rangle$ (i.e. \overline{uv}), and **\vec** has been changed to match the `\vectorsym` command provided by `isomath`.

\Re { $\langle expression \rangle$ }
\Im { $\langle expression \rangle$ }

These commands typeset the real and imaginary part of a number. Standard use of the package takes definitions roughly from `amsmath`, while setting the `notation` option to `iso` changes the definitions to match ISO 80000-2.

3 Known issues

A list of current issues is available in the Github repository of this package¹, but as of the release of v0.4, there is one known issue.

#15 The package is incompatible with (at least) `blindtext`, when including math in the blind text. This is due to the redefinition of `\sin` (and friends), which is incompatible with the original `amsmath` definition. This is a feature, not a bug.

If you discover any bugs in this package, please report them to the issue tracker in the `skmath` Github repository.

¹<https://github.com/urdh/skmath/issues>

4 Installation

The easiest way to install this package is using the package manager provided by your \TeX installation if such a program is available. Failing that, provided you have obtained the package source (`skmath.tex` and `Makefile`) from either CTAN or Github, running `make install` inside the source directory works well. This will extract the documentation and code from `skmath.tex`, install all files into the TDS tree at `TEXMFHOME` and run `mktexlsr`.

If you want to extract code and documentation without installing the package, run `make all` instead. If you insist on not using `make`, remember that packages distributed using `skdoc` must be extracted using `pdflatex`, *not* `tex` or `latex`.

5 Changes

vo.1	General: Initial version.	vo.2	General: Use <code>expl3</code> functionality throughout the package.
vo.1c	General: Moved package from <code>docstrip</code> to <code>skdoc</code> .	vo.3	General: Added <code>\min/\max</code> and friends. Added <code>\pd</code> .
vo.1d	General: Fixed fatal documentation and package errors.	vo.3a	General: Added <code>\sinh</code> , <code>\cosh</code> and <code>\tanh</code> .
vo.1e	General: Added statistics commands.	vo.3b	General: Detect empty arguments in trigonometric and logarithmic functions, fix <code>\ln</code> .
vo.1g	General: Documentation fixes.		

General: Added notation
option, macros for complex

6 Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the page where the implementation of the corresponding entry is discussed. Numbers in *roman* refer to other mentions of the entry.

A

\abs *3*
amsmath (package) *8*
\arccos *6*
\arcsin *6*
\arctan *6*
\argmax *7*
\argmin *7*

B

\bar *8*
blindtext (package) *8*

C

\C *2, 3*
commonsets (option) *2, 3*
\cosh *6*
\cos *6*
\cot *6*
\cov *5*

D

\d *4*

E

\E *5*
\exp *7*

F

\frac *7*

G

\given *5*

I

\ii *3*
\Im *8*
\inf *7*
isomath (package) *8*

J

\jj *3*

L

\ln *6*
\log *6*

M

Makefile (file) 9

\max 7

\min 7

N

\N 2, 3

\norm 3

notation (option) 2, 3, 8

P

\pd 4

\P 5

Q

\Q 2, 3

R

\Re 8

\R 2, 3

S

\sinh 6

\sin 6, 8

skdoc (package) 9

skmath.tex (file) 9

\sup 7

T

\tanh 6

\tan 6

V

\var 5

\vec 8

\vectorsym 8

Z

\Z 2, 3