

The skmath package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.3b

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 v0.3b, there is only one option: `commonsets`. By default, it is disabled but if the option is given the package will define `\N`, `\Z`, `\Q`, `\R` and `\C` as blackboard variants of the respective letters, to represent the common sets of numbers.

*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*}
```

`\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 $\langle \textit{variable} \rangle$

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} dx
\end{equation*}
```

\pd $\ast \langle \textit{function} \rangle \langle \textit{var} \rangle, \langle \textit{var} \rangle, \dots$

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 $\langle \textit{expression} \rangle$

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 **\given** 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\given B} =
  \frac{\P{B\given 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.

$\backslash\sin$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\arcsin$ { $\langle expression \rangle$ }
 $\backslash\cos$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\arccos$ { $\langle expression \rangle$ }
 $\backslash\tan$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\arctan$ { $\langle expression \rangle$ }
 $\backslash\cot$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\sinh$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\cosh$ [$\langle power \rangle$] { $\langle expression \rangle$ }
 $\backslash\tanh$ [$\langle power \rangle$] { $\langle expression \rangle$ }

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

$\backslash\ln$ { $\langle expression \rangle$ }

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

$\backslash\log$ [$\langle base \rangle$] { $\langle expression \rangle$ }

The related macro $\backslash\log$ 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 .

$\backslash\exp$ *{ $\langle expression \rangle$ }

The exponential, $\backslash\exp$, 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	* [<i>domain</i>] { <i>expression</i> }
\argmin	* [<i>domain</i>] { <i>expression</i> }
\max	* [<i>domain</i>] { <i>expression</i> }
\argmax	* [<i>domain</i>] { <i>expression</i> }
\sup	* [<i>domain</i>] { <i>expression</i> }
\inf	* [<i>domain</i>] { <i>expression</i> }

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 {*expression*}

\vec {*expression*}

The **\bar** command has been changed to cover the entire *expression* (i.e. \overline{w}), and **\vec** has been changed to match the `\vector sym` command provided by `isomath`.

3 Known issues

A list of current issues is available in the Github repository of this package¹, but as of the release of v0.3b, 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 `pdf \LaTeX` , *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.		

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 *2*
amsmath (**package**) *7*
\arccos *5*
\arcsin *5*
\arctan *5*
\argmax *6*
\argmin *6*

B

\bar *6*
blindtext (**package**) *7*

C

\C *1, 2*
commonsets (**option**) *1, 2*
\cosh *5*
\cos *5*
\cot *5*
\cov *4*

D

\d *3*

E

\E *3*
\exp *5*

F

\frac *6*

G

\given *4*

I

\inf *6*
isomath (**package**) *6*

L

\ln *5*
\log *5*

M

Makefile (**file**) *8*
\max *6*
\min *6*

N

\N *1, 2*
\norm *2*

P

\pd *3*
\P *4*

Q

`\Q` 1,2

R

`\R` 1,2

S

`\sinh` 5

`\sin` 5,7

skdoc (package) 8

`skmath.tex` (file) 8

`\sup` 6

T

`\tanh` 5

`\tan` 5

V

`\var` 4

`\vec` 6

`\vectorsym` 6

Z

`\Z` 1,2