

The **skmath** package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.1h

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 v0.1h, 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/skbundle>.

[†]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:

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

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

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

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*}
```

`\E` $\{\langle expression \rangle\}$

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

Example:

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

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

`\P` $\{\langle expression \rangle \backslash \text{given} \langle expression \rangle\}$

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

`\cov` $\{\langle expression \rangle \{\langle expression \rangle\}$

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

Example:

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

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

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. $\sin^2(\phi)$.

```
\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: $\log_2(x)$.

`\exp` $\{\langle expression \rangle\}$

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:

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

2.4 Stylistic changes

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

`\frac` $\{\langle numerator \rangle\}\{\langle denominator \rangle\}$

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`.

3 Known issues

A list of current issues is available in the Github repository of this package¹, but as of the release of v0.1h, there are no known issues

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

4 Implementation

The package implementation is very simple. First, we do the standard \LaTeX 2_ε preamble thing, then we require some dependencies.

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

```

(package) 1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
2 \ProvidesPackage{skmath}%
3 [2013/04/01 v0.1h skmath improved math commands]
4 \RequirePackage{xparse}
5 \PassOptionsToPackage{intlimits}{amsmath}
6 \RequirePackage{kvoptions,amssymb,mathtools,xfrac,isomath}

```

We begin by declaring an option.

```

(package) 7 \SetupKeyvalOptions{family=skmath,prefix=skmath@}
8 \DeclareBoolOption[false]{commonsets}
9 \ProcessKeyvalOptions*

```

We optionally provide commands to typeset common sets.

```

(package) 10 \ifskmath@commonsets

```

\mathbb{N} (no arguments)

```

(package) 11 \NewDocumentCommand\N{}{\ensuremath{\mathbb{N}}}

```

\mathbb{Z} (no arguments)

```

(package) 12 \NewDocumentCommand\Z{}{\ensuremath{\mathbb{Z}}}

```

\mathbb{Q} (no arguments)

```

(package) 13 \NewDocumentCommand\Q{}{\ensuremath{\mathbb{Q}}}

```

\mathbb{R} (no arguments)

```

(package) 14 \NewDocumentCommand\R{}{\ensuremath{\mathbb{R}}}

```

\mathbb{C} (no arguments)

```

(package) 15 \NewDocumentCommand\C{}{\ensuremath{\mathbb{C}}}

```

```

(package) 16 \fi

```

This is followed by commands to typeset the norm and absolute value.

`\abs`(no arguments)

```
(package) 17 \DeclarePairedDelimiter\abs{\lvert}{\rvert}
```

`\norm`(no arguments)

```
(package) 18 \DeclarePairedDelimiter\norm{\lVert}{\rVert}
```

Next come the statistical commands.

`\E`(no arguments)

Here, we define `\E` after the preamble since it may break otherwise.

```
(package) 19 \AtBeginDocument{
20   \DeclareDocumentCommand\E{m}{%
21     \ensuremath{\mathop{\mathrm{E}}\left[#1\right]}%
22   }
23 }
```

The `\P` command saves any old `\given` command, replacing it locally with the new `\given` command provided by the package.

`\P`(no arguments)

```
(package) 24 \DeclareDocumentCommand\P{m}{%
25   \ensuremath{\mathop{\mathrm{P}}}%
26   \left(%
27   \let\skmath@given\given%
```

`\given` (no arguments)

```
(package) 28   \DeclareDocumentCommand\given{}{\mid}%
29   #1%
30   \let\given\skmath@given%
31   \right)%
```

```

32 }%
33 }

```

\var(no arguments)

```

(package) 34 \DeclareDocumentCommand\var{m}{%
35 \ensuremath{\mathop{\mathrm{Var}}}\left(\#1\right)}%
36 }

```

\cov(no arguments)

```

(package) 37 \DeclareDocumentCommand\cov{mm}{%
38 \ensuremath{\mathop{\mathrm{Cov}}}\left(\#1,\#2\right)}%
39 }

```

We replace all trigonometric functions and some other common functions with alternatives that take an argument (or optionally, several arguments).

```

(package) 40 \let\skmath@sin\sin
41 \let\skmath@cos\cos
42 \let\skmath@tan\tan
43 \let\skmath@cot\cot
44 \let\skmath@arcsin\arcsin
45 \let\skmath@arccos\arccos
46 \let\skmath@arccos\arctan
47 \let\skmath@ln\log
48 \let\skmath@log\log
49 \let\skmath@exp\exp

```

\sin(no arguments)

```

(package) 50 \RenewDocumentCommand\sin{om}{%
51 \IfNoValueTF{\#1}
52 {\ensuremath{\skmath@sin\left(\#2\right)}}
53 {\ensuremath{\skmath@sin^{\#1}\left(\#2\right)}}}%
54 }

```


$\backslash\cos$ (no arguments)

```
(package) 55 \RenewDocumentCommand\cos{om}{%  
56   \IfNoValueTF{#1}  
57     {\ensuremath{\skmath@cos\left(#2\right)}}  
58     {\ensuremath{\skmath@cos^{#1}\left(#2\right)}}}%  
59 }
```

$\backslash\tan$ (no arguments)

```
(package) 60 \RenewDocumentCommand\tan{om}{%  
61   \IfNoValueTF{#1}  
62     {\ensuremath{\skmath@tan\left(#2\right)}}  
63     {\ensuremath{\skmath@tan^{#1}\left(#2\right)}}}%  
64 }
```

$\backslash\cot$ (no arguments)

```
(package) 65 \RenewDocumentCommand\cot{om}{%  
66   \IfNoValueTF{#1}  
67     {\ensuremath{\skmath@cot\left(#2\right)}}  
68     {\ensuremath{\skmath@cot^{#1}\left(#2\right)}}}%  
69 }
```

$\backslash\arcsin$ (no arguments)

```
(package) 70 \RenewDocumentCommand\arcsin{m}{%  
71   \ensuremath{\skmath@arcsin\left(#1\right)}}%  
72 }
```

$\backslash\arccos$ (no arguments)

```
(package) 73 \RenewDocumentCommand\arccos{m}{%  
74   \ensuremath{\skmath@arccos\left(#1\right)}}%  
75 }
```

\backslash arctan(no arguments)

```
(package) 76 \RenewDocumentCommand\arctan{m}{%  
77   \ensuremath{\skmath@arctan\left(#1\right)}}%  
78 }
```

\backslash ln(no arguments)

```
(package) 79 \RenewDocumentCommand\ln{m}{%  
80   \ensuremath{\skmath@ln\left(#1\right)}}%  
81 }
```

\backslash log(no arguments)

```
(package) 82 \RenewDocumentCommand\log{om}{%  
83   \IfNoValueTF{#1}  
84     {\ensuremath{\skmath@log\left(#2\right)}}%  
85     {\ensuremath{\skmath@log_{#1}\left(#2\right)}}%  
86 }
```

\backslash exp(no arguments)

```
(package) 87 \RenewDocumentCommand\exp{m}{\ensuremath{\mathchoice%  
88   {e^{#1}}%  
89   {\skmath@exp\left(#1\right)}}%  
90   {\skmath@exp\left(#1\right)}}%  
91   {\skmath@exp\left(#1\right)}}%  
92 }}
```

The fraction command is modified to improve typesetting.

\backslash frac(no arguments)

```
(package) 93 \RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}%  
94   {\displaystyle #1}{\displaystyle #2}}
```

Definition of \backslash bar and \backslash d is deferred until after all packages are loaded to avoid collisions with other packages.

```
(package) 95 \AtBeginDocument{%
```

The `\bar` command is modified to improve typesetting.

`\bar`(no arguments)

```
(package) 96 \DeclareDocumentCommand\bar{m}{%  
97 \ensuremath{\mkern 1.5mu\overline{\mkern-1.5mu{#1}\mkern-1.5mu}\mkern 1.5mu}}
```

We introduce a command to typeset the differential part of integrals, shamefully stolen from an answer on T_EX.SE.

`\d`(no arguments)

```
(package) 98 \DeclareDocumentCommand\d{m}{\ensuremath{\!,\mathrm{d}}#1%  
99 \@ifnextchar\d{\!}{}}
```

```
(package) 100 }
```

Finally, we define a nicer way to denote vectors.

`\vec`(no arguments)

```
(package) 101 \let\vec\vectorsym
```

```
(package) 102 \endinput
```

5 Changes

v0.1

General: Initial version.

v0.1a

`\d`: Fixed obtuse errors.

v0.1b

General: Load `amsmath` with `intlimits` option.

`\bar`: Added `\bar` replacement.

`\C`: Moved to `xparse` command definition.

`\d`: Moved to `xparse` command definition.

<code>\exp</code> : Moved to xparse command definition.	v0.1e
<code>\frac</code> : Moved to xparse command definition.	General: Added statistics commands.
<code>\N</code> : Moved to xparse command definition.	<code>\cov</code> : Added <code>\cov</code> command.
<code>\Q</code> : Moved to xparse command definition.	<code>\E</code> : Added <code>\E</code> command.
<code>\R</code> : Moved to xparse command definition.	<code>\given</code> : Added <code>\given</code> command.
<code>\Z</code> : Moved to xparse command definition.	<code>\P</code> : Added <code>\P</code> command.
	<code>\var</code> : Added <code>\var</code> command.
	v0.1f
	<code>\E</code> : Fixed ‘Command <code>\E</code> already defined!’ error.
v0.1c	
General: Moved package from docstrip to skdoc.	v0.1g
	General: Documentation fixes.
v0.1d	
General: Fixed fatal documentation and package errors.	v0.1h
	<code>\bar</code> : Wrap in <code>\AtBeginDocument</code> .

6 Index

Numbers written in boldface 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	B
<code>\abs</code> 2 , 7	<code>\bar</code> 5 , 10 , 11
<code>\arccos</code> 4 , 9	
<code>\arcsin</code> 4 , 9	C
<code>\arctan</code> 4 , 9	<code>\C</code> 1 , 2 , 6

commonsets (option)	1, <u>2</u>	\N	1, <u>2</u> , <u>6</u>
\cos	<u>4</u> , <u>8</u>	\norm	<u>2</u> , <u>7</u>
\cot	<u>4</u> , <u>9</u>		
\cov	<u>3</u> , <u>8</u>	P	
		\P	<u>3</u> , <u>7</u>
D			
\d	<u>3</u> , <u>10</u> , <u>11</u>	Q	
		\Q	1, <u>2</u> , <u>6</u>
E			
\E	<u>3</u> , <u>7</u>	R	
\exp	<u>5</u> , <u>10</u>	\R	1, <u>2</u> , <u>6</u>
F		S	
\frac	<u>5</u> , <u>10</u>	\sin	<u>4</u> , <u>8</u>
G		T	
\given	<u>3</u> , <u>7</u>	\tan	<u>4</u> , <u>9</u>
I		V	
isomath (package)	<u>5</u>	\var	<u>3</u> , <u>8</u>
		\vec	<u>5</u> , <u>11</u>
L		\vectorsym	<u>5</u>
\ln	<u>4</u> , <u>10</u>		
\log	<u>4</u> , <u>10</u>	Z	
		\Z	1, <u>2</u> , <u>6</u>
N			