

The **skmath** package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.2

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 v0.2, 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:

,,,,,

```
\begin{equation*}
  \N, \Z, \Q, \R, \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)^{\sfrac{1}{p}}
\end{equation*}
```

`\d` {*<variable>*}

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

Example:

$$\int \frac{\sin(x)}{x} dx$$

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

`\E` {*<expression>*}

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

Example:

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

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

`\P` {*<expression>* \given *<expression>*}

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

Example:

$$\mathrm{P}(A \mid B) = \frac{\mathrm{P}(B \mid A) \mathrm{P}(A)}{\mathrm{P}(B)}$$

```
\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:

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

```
\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` $[\langle power \rangle]\{\langle expression \rangle\}$
`\arcsin` $\{\langle expression \rangle\}$
`\cos` $[\langle power \rangle]\{\langle expression \rangle\}$
`\arccos` $\{\langle expression \rangle\}$
`\tan` $[\langle power \rangle]\{\langle expression \rangle\}$
`\arctan` $\{\langle expression \rangle\}$
`\cot` $[\langle power \rangle]\{\langle expression \rangle\}$

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

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

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` {*⟨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:

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

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{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 v0.2, 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.

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

4 Implementation

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

```
(package) 1 \RequirePackage{expl3,l3keys2e,xparse}
2 \ProvidesExplPackage{skmath}
3 {2013/04/09}{0.1}{improved math commands}
4 \PassOptionsToPackage{intlimits}{amsmath}
5 \RequirePackage{amssymb,mathtools,xfrac,isomath}
```

We optionally provide commands to typeset common sets.

`_define_common_sets:` (no arguments)

```
(package) 6 \cs_new:Nn\__skmath_define_common_sets:{

\N(no arguments)

(package) 7 \NewDocumentCommand\N{}{\ensuremath{\mathbb{N}}}

\Z(no arguments)

(package) 8 \NewDocumentCommand\Z{}{\ensuremath{\mathbb{Z}}}

\Q(no arguments)

(package) 9 \NewDocumentCommand\Q{}{\ensuremath{\mathbb{Q}}}

\R(no arguments)

(package) 10 \NewDocumentCommand\R{}{\ensuremath{\mathbb{R}}}

\C(no arguments)

(package) 11 \NewDocumentCommand\C{}{\ensuremath{\mathbb{C}}}
```

```
(package) 12 }
```

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

`\abs`(no arguments)

```
(package) 13 \DeclarePairedDelimiter\abs{\lvert}{\rvert}
```

`\norm`(no arguments)

```
(package) 14 \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) 15 \AtBeginDocument{
16   \DeclareDocumentCommand\E{m}{%
17     \ensuremath{\operatorname{E}\left[#1\right]}}%
18   }
19 }
```

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

`\P`(no arguments)

```
(package) 20 \DeclareDocumentCommand\P{m}{%
21   \ensuremath{\operatorname{P}\left(
22     \left(
23       \cs_set_eq:NN\__skmath_saved_given:\given%
```

`\given` (no arguments)

```
(package) 24 \DeclareDocumentCommand\given{}{\mid}%
```

```

(package) 25      #1%
26      \cs_set_eq:NN\given\__skmath_saved_given:%
27      \right)%
28    }%
29  }

```

`\var`(no arguments)

```

(package) 30      \DeclareDocumentCommand\var{m}{%
31      \ensuremath{\operatorname{Var}\left(#1\right)}}%
32    }

```

`\cov`(no arguments)

```

(package) 33      \DeclareDocumentCommand\cov{mm}{%
34      \ensuremath{\operatorname{Cov}\left(#1,#2\right)}}%
35    }

```

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

```

(package) 36      \cs_new_eq:NN\__skmath_sin:\sin
37      \cs_new_eq:NN\__skmath_cos:\cos
38      \cs_new_eq:NN\__skmath_tan:\tan
39      \cs_new_eq:NN\__skmath_cot:\cot
40      \cs_new_eq:NN\__skmath_arcsin:\arcsin
41      \cs_new_eq:NN\__skmath_arccos:\arccos
42      \cs_new_eq:NN\__skmath_arctan:\arctan
43      \cs_new_eq:NN\__skmath_ln:\log
44      \cs_new_eq:NN\__skmath_log:\log
45      \cs_new_eq:NN\__skmath_exp:\exp

```

`\sin`(no arguments)

```

(package) 46      \RenewDocumentCommand\sine{om}{%
47      \IfNoValueTF{#1}

```



```

48     {\ensuremath{\_\_skmath\_sin:\left(#2\right)}}
49     {\ensuremath{\_\_skmath\_sin:\c\_math\_superscript\_token{#1}\left(#2\right)}}}%
50   }

```

\cos(no arguments)

```

(package) 51 \RenewDocumentCommand\cos{om}{%
52   \IfNoValueTF{#1}
53   {\ensuremath{\_\_skmath\_cos:\left(#2\right)}}
54   {\ensuremath{\_\_skmath\_cos:\c\_math\_superscript\_token{#1}\left(#2\right)}}}%
55 }

```

\tan(no arguments)

```

(package) 56 \RenewDocumentCommand\tan{om}{%
57   \IfNoValueTF{#1}
58   {\ensuremath{\_\_skmath\_tan:\left(#2\right)}}
59   {\ensuremath{\_\_skmath\_tan:\c\_math\_superscript\_token{#1}\left(#2\right)}}}%
60 }

```

\cot(no arguments)

```

(package) 61 \RenewDocumentCommand\cot{om}{%
62   \IfNoValueTF{#1}
63   {\ensuremath{\_\_skmath\_cot:\left(#2\right)}}
64   {\ensuremath{\_\_skmath\_cot:\c\_math\_superscript\_token{#1}\left(#2\right)}}}%
65 }

```

\arcsin(no arguments)

```

(package) 66 \RenewDocumentCommand\arcsin{m}{%
67   \ensuremath{\_\_skmath\_arcsin:\left(#1\right)}}%
68 }

```

\arccos(no arguments)

```

(package) 69 \RenewDocumentCommand\arccos{m}{%
70   \ensuremath{\_\_skmath_arccos:\left(\#1\right)}}%
71 }

```

\arctan(no arguments)

```

(package) 72 \RenewDocumentCommand\arctan{m}{%
73   \ensuremath{\_\_skmath_arctan:\left(\#1\right)}}%
74 }

```

\ln(no arguments)

```

(package) 75 \RenewDocumentCommand\ln{m}{%
76   \ensuremath{\_\_skmath_ln:\left(\#1\right)}}%
77 }

```

\log(no arguments)

```

(package) 78 \RenewDocumentCommand\log{om}{%
79   \IfNoValueTF{#1}
80     {\ensuremath{\_\_skmath_log:\left(\#2\right)}}
81     {\ensuremath{\_\_skmath_log:\c_math_subscript_token{#1}\left(\#2\right)}}%
82 }

```

\exp(no arguments)

```

(package) 83 \RenewDocumentCommand\exp{m}{\ensuremath{\mathchoice%
84   {e\c_math_superscript_token{#1}}%
85   {\_\_skmath_exp:\left(\#1\right)}}%
86   {\_\_skmath_exp:\left(\#1\right)}}%
87   {\_\_skmath_exp:\left(\#1\right)}}%
88 }}

```

The fraction command is modified to improve typesetting.

\frac(no arguments)

```

(package) 89 \RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}{}%
90           {\displaystyle #1}{\displaystyle #2}}

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

(package) 91 \AtBeginDocument{%

The \bar command is modified to improve typesetting.

\bar(no arguments)

(package) 92 \DeclareDocumentCommand\bar{m}{%
93           \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 TEX.SE.

\d(no arguments)

(package) 94 \DeclareDocumentCommand\d{m}{\ensuremath{\,\,\mathrm{d}}#1%
95           \peek_meaning_ignore_spaces:NT\d{\\!}}

(package) 96 }

Finally, we define a nicer way to denote vectors.

\vec(no arguments)

(package) 97 \cs_set_eq:NN\vec\vectorssym

We end by declaring an option.

(package) 98 \keys_define:nn{skmath}{
99   commonsets .code:n =
100   {\_skmath_define_common_sets:}
101 }
102 \ProcessKeysPackageOptions{skmath}

(package) 103 \endinput

```

5 Installation

The easiest way to install this package is using the package manager provided by your \LaTeX 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`.

6 Changes

v0.1	<code>\frac</code> : Moved to <code>xparse</code> command definition.
General: Initial version.	<code>\N</code> : Moved to <code>xparse</code> command definition.
v0.1a	<code>\Q</code> : Moved to <code>xparse</code> command definition.
<code>\d</code> : Fixed obtuse errors.	<code>\R</code> : Moved to <code>xparse</code> command definition.
v0.1b	<code>\Z</code> : Moved to <code>xparse</code> command definition.
General: Load <code>amsmath</code> with <code>intlimits</code> option.	
<code>\bar</code> : Added <code>\bar</code> replacement.	v0.1c
<code>\C</code> : Moved to <code>xparse</code> command definition.	General: Moved package from <code>docstrip</code> to <code>skdoc</code> .
<code>\d</code> : Moved to <code>xparse</code> command definition.	v0.1d
<code>\exp</code> : Moved to <code>xparse</code> command definition.	General: Fixed fatal documentation and package errors.

<p>v0.1e</p> <p>General: Added statistics commands.</p> <p><code>\cov</code>: Added <code>\cov</code> command.</p> <p><code>\E</code>: Added <code>\E</code> command.</p> <p><code>\given</code>: Added <code>\given</code> command.</p> <p><code>\P</code>: Added <code>\P</code> command.</p> <p><code>\var</code>: Added <code>\var</code> command.</p> <p>v0.1f</p> <p><code>\E</code>: Fixed ‘Command <code>\E</code> already defined!’ error.</p> <p>v0.1g</p> <p>General: Documentation fixes.</p>	<p>v0.1h</p> <p><code>\bar</code>: Wrap in <code>\AtBeginDocument</code>.</p> <p>v0.2</p> <p>General: Use <code>expl3</code> functionality throughout the package.</p> <p><code>\cov</code>: Use <code>\operatorname</code>.</p> <p><code>\d</code>: Use <code>\peek_meaning_ignore_spaces:NT</code> instead of <code>\@ifnextchar</code>.</p> <p><code>\E</code>: Use <code>\operatorname</code>.</p> <p><code>\P</code>: Use <code>\operatorname</code>, use <code>\cs_new_eq:NN</code> instead of <code>\let</code>.</p> <p><code>\var</code>: Use <code>\operatorname</code>.</p> <p><code>\vec</code>: Use <code>\cs_new_eq:NN</code> instead of <code>\let</code>.</p>
--	--

7 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.

Symbols	<code>\arctan</code> <i>4</i> , <u><i>10</i></u>
<code>_skmath_define_common_sets:</code>	
<u><i>6</i></u>	B
	<code>\bar</code> <i>5</i> , <u><i>11</i></u>
A	
<code>\abs</code> <i>2</i> , <u><i>7</i></u>	C
<code>\arccos</code> <i>4</i> , <u><i>9</i></u>	<code>\C</code> <i>1</i> , <i>2</i> , <u><i>6</i></u>
<code>\arcsin</code> <i>4</i> , <u><i>9</i></u>	<code>commonsets</code> (option) <i>1</i> , <i>2</i>

$\backslash\cos$	4, <u>9</u>	N	
$\backslash\cot$	4, <u>9</u>	$\backslash N$	1, 2, <u>6</u>
$\backslash\cos$	4, <u>8</u>	$\backslash\text{norm}$	2, <u>7</u>
D			
$\backslash d$	3, <u>11</u>	P	
		$\backslash P$	3, 7
E			
$\backslash E$	3, <u>7</u>	Q	
$\backslash\exp$	5, <u>10</u>	$\backslash Q$	1, 2, <u>6</u>
F			
$\backslash\text{frac}$	5, <u>10</u>	R	
		$\backslash R$	1, 2, <u>6</u>
G			
$\backslash\text{given}$	3, 7	S	
		$\backslash\sin$	4, <u>8</u>
		skdoc (package)	12
		skmath.tex (file)	12
I			
isomath (package)	5	T	
		$\backslash\tan$	4, <u>9</u>
L			
$\backslash\ln$	4, <u>10</u>	V	
$\backslash\log$	5, <u>10</u>	$\backslash\text{var}$	4, <u>8</u>
		$\backslash\text{vec}$	5, <u>11</u>
		$\backslash\text{vectorsym}$	5
M			
Makefile (file)	12	Z	
		$\backslash Z$	1, 2, <u>6</u>