

The **skmath** package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.1g

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 v0.1g, 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:

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

`\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:

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

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

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

Example:

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

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

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

Example:

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

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

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

Example:

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

`\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:

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

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

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 v0.1g, there is only one known issue:

#4 When using both `fontspec` and `skmath`, sometimes \LaTeX bails out saying that ‘`\bar` is already defined’. This is probably due to fonts defining their own `\bar`, and will happen with other font packages as well, but I haven’t figured out a suitable solution yet.

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.

```
(package) 1 \NeedsTeXFormat{LaTeX2e}[1999/12/01]
```

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

```

2 \ProvidesPackage{skmath}%
3   [2013/02/18 v0.1e 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

```

$\backslash N$ (no arguments)

```

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

```

$\backslash Z$ (no arguments)

```

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

```

$\backslash Q$ (no arguments)

```

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

```

$\backslash R$ (no arguments)

```

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

```

$\backslash C$ (no arguments)

```

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

```

```

(package) 16 \fi

```

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

$\backslash abs$ (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}}}\left(
26     \let\skmath@given\given%
27     \right)%
```

`\given` (no arguments)

```
(package) 28   \DeclareDocumentCommand\given{}{\mid}%
(package) 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

```

$\backslash\sin$ (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.

`\frac`(no arguments)

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

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

`\bar`(no arguments)

```
(package) 95 \RenewDocumentCommand\bar{m}{%
96           \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.S_E. Definition is deferred until after all packages are loaded to avoid collisions with other `\d` commands.

```
(package) 97 \AtBeginDocument{%
```

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

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