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 {⟨expression⟩}
 \abs {⟨expression⟩}
```

The commands `\norm` and `\abs`, quite expectedly, typeset the norm ans 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 {⟨variable⟩}
```

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

Example:

$$\int\limits_{\mathbb{R}} \frac{\sin\left(x\right)}{x} \,\mathrm{d}x$$

`\E` {⟨*expression*⟩}

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

Example:

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

`\P` {⟨*expression*⟩`\given`⟨*expression*⟩}

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

Example:

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

`\var` {⟨*expression*⟩}
`\cov` {⟨*expression*⟩}{⟨*expression*⟩}

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

Example:

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

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 ⟨*expression*⟩ as an argument of the expression, and (if applicable) ⟨*power*⟩ as a superscript between the function and its argument, *e.g.* $\sin^2(\phi)$.

`\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` {⟨*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 package[1], 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$_\varepsilon$ preamble thing, then we require some dependencies.

⟨package⟩

```
\NeedsTeXFormat{LaTeX2e}[1999/12/01]
```

[1] https://github.com/urdh/skmath/issues

```
\ProvidesPackage{skmath}%
    [2013/02/18 v0.1e skmath improved math commands]
\RequirePackage{xparse}
\PassOptionsToPackage{intlimits}{amsmath}
\RequirePackage{kvoptions,amssymb,mathtools,xfrac,isomath}
```

We begin by declaring an option.

```
⟨package⟩ \SetupKeyvalOptions{family=skmath,prefix=skmath@}
          \DeclareBoolOption[false]{commonsets}
          \ProcessKeyvalOptions*
```

We optionally provide commands to typeset common sets.

```
⟨package⟩ \ifskmath@commonsets
```

\N(no arguments)

```
⟨package⟩   \NewDocumentCommand\N{}{\ensuremath{\mathbb{N}}}
```

\Z(no arguments)

```
⟨package⟩   \NewDocumentCommand\Z{}{\ensuremath{\mathbb{Z}}}
```

\Q(no arguments)

```
⟨package⟩   \NewDocumentCommand\Q{}{\ensuremath{\mathbb{Q}}}
```

\R(no arguments)

```
⟨package⟩   \NewDocumentCommand\R{}{\ensuremath{\mathbb{R}}}
```

\C(no arguments)

```
⟨package⟩   \NewDocumentCommand\C{}{\ensuremath{\mathbb{C}}}
```

```
⟨package⟩ \fi
```

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

\abs(no arguments)

```
⟨package⟩ \DeclarePairedDelimiter\abs{\lvert}{\rvert}
```

\norm(no arguments)

⟨package⟩
```
\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⟩
```
\AtBeginDocument{
  \DeclareDocumentCommand\E{m}{%
    \ensuremath{\mathop{\mathrm{E}}\left[#1\right]}%
  }
}
```

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

\P(no arguments)

⟨package⟩
```
\DeclareDocumentCommand\P{m}{%
  \ensuremath{\mathop{\mathrm{P}}%
    \left(%
    \let\skmath@given\given%
```

\given (no arguments)

⟨package⟩
```
    \DeclareDocumentCommand\given{}{\mid}%
```

⟨package⟩
```
    #1%
    \let\given\skmath@given%
    \right)%
  }%
}
```

\var(no arguments)

⟨package⟩
```
  \DeclareDocumentCommand\var{m}{%
    \ensuremath{\mathop{\mathrm{Var}}\left(#1\right)}%
  }
```

\cov(no arguments)

⟨package⟩
```
  \DeclareDocumentCommand\cov{mm}{%
```

```
    \ensuremath{\mathop{\mathrm{Cov}}\left(#1,#2\right)}%
  }
```

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

⟨package⟩

```
\let\skmath@sin\sin
\let\skmath@cos\cos
\let\skmath@tan\tan
\let\skmath@cot\cot
\let\skmath@arcsin\arcsin
\let\skmath@arccos\arccos
\let\skmath@arccos\arctan
\let\skmath@ln\log
\let\skmath@log\log
\let\skmath@exp\exp
```

\sin(no arguments)

⟨package⟩

```
\RenewDocumentCommand\sin{om}{%
  \IfNoValueTF{#1}
    {\ensuremath{\skmath@sin\left(#2\right)}}
    {\ensuremath{\skmath@sin^{#1}\left(#2\right)}}%
}
```

\cos(no arguments)

⟨package⟩

```
\RenewDocumentCommand\cos{om}{%
  \IfNoValueTF{#1}
    {\ensuremath{\skmath@cos\left(#2\right)}}
    {\ensuremath{\skmath@cos^{#1}\left(#2\right)}}%
}
```

\tan(no arguments)

⟨package⟩

```
\RenewDocumentCommand\tan{om}{%
  \IfNoValueTF{#1}
    {\ensuremath{\skmath@tan\left(#2\right)}}
    {\ensuremath{\skmath@tan^{#1}\left(#2\right)}}%
}
```

\cot(no arguments)

⟨package⟩

```
\RenewDocumentCommand\cot{om}{%
  \IfNoValueTF{#1}
```

```
    {\ensuremath{\skmath@cot\left(#2\right)}}
    {\ensuremath{\skmath@cot^{#1}\left(#2\right)}}%
}
```

\arcsin(no arguments)

⟨package⟩

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

\arccos(no arguments)

⟨package⟩

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

\arctan(no arguments)

⟨package⟩

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

\ln(no arguments)

⟨package⟩

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

\log(no arguments)

⟨package⟩

```
\RenewDocumentCommand\log{om}{%
  \IfNoValueTF{#1}
    {\ensuremath{\skmath@log\left(#2\right)}}
    {\ensuremath{\skmath@log_{#1}\left(#2\right)}}%
}
```

\exp(no arguments)

⟨package⟩

```
\RenewDocumentCommand\exp{m}{\ensuremath{\mathchoice%
  {e^{#1}}%
  {\skmath@exp\left(#1\right)}%
  {\skmath@exp\left(#1\right)}%
  {\skmath@exp\left(#1\right)}%
}}
```

The fraction command is modified to improve typesetting.

\frac(no arguments)

⟨package⟩
```
\RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}%
              {\displaystyle #1}{\displaystyle #2}}
```

The \bar command is also modified to impove typesetting.

\bar(no arguments)

⟨package⟩
```
\RenewDocumentCommand\bar{m}{%
    \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. Definition is deferred until after all packages are loaded to avoid collisions with other \d commands.

⟨package⟩
```
\AtBeginDocument{%
```

\d(no arguments)

⟨package⟩
```
\DeclareDocumentCommand\d{m}{\ensuremath{\,\mathrm{d}#1%
                                \@ifnextchar\d{\!}{}}}
```

⟨package⟩
```
}
```

Finally, we define a nicer way to denote vectors.

\vec(no arguments)

⟨package⟩
```
\let\vec\vectorsym
```

⟨package⟩
```
\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.

`\exp`: Moved to xparse command definition.

`\frac`: Moved to xparse command definition.

`\N`: Moved to xparse command definition.

`\Q`: Moved to xparse command definition.

`\R`: Moved to xparse command definition.

`\Z`: Moved to xparse command definition.

v0.1c

General: Moved package from docstrip to skdoc.

v0.1d

General: Fixed fatal documentation and package errors.

v0.1e

General: Added statistics commands.

`\cov`: Added `\cov` command.

`\E`: Added `\E` command.

`\given`: Added `\given` command.

`\P`: Added `\P` command.

`\var`: Added `\var` command.

v0.1f

`\E`: Fixed 'Command `\E` 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 were the implementation of the corresponding entry is discussed. Numbers in roman refer to other mentions of the entry.