The **skmath** package*†

Simon Sigurdhsson ⌈sigurdhsson@gmail.com⌋

Version 0.5

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 v0.5, the package provides two key-value options.

`commonsets` `true,false` (`false`)
Optionally define `\N`, `\Z`, `\Q`, `\R` and `\C` as blackboard variants of the respective letters, to represent the common sets of numbers.

`notation` `iso,english,german,legacy` (`legacy`)
This option controls the style of a few typographic elements that differ between countries and standards (such as the style of integrals, derivatives and greek letters).

[*] 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*}
  \N, \Z, \Q, \R, \C.
\end{equation*}
```

\ii
\jj

These commands typeset the imaginary unit (either i as used in mathematics or j as used in electrotechnology). While normal use of the package simply results in italic characters, setting the `notation` option to `iso` will set these upright.

\ee

This command typesets Euler's number $e = \sum_{n=0}^{\infty} \frac{1}{n!}$. The style is affected by the `notation` option in the same way as **\exp**.

`\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:

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

```
\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:

$$\int_{\mathbb{R}} \frac{\sin(x)}{x} \,\mathrm{d}x$$

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

`\pd` ⋆{⟨*function*⟩}{⟨*var*⟩,⟨*var*⟩,...}

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.

\td {⟨*function*⟩}{⟨*var*⟩}

This macro typesets a total derivative. Unlike **\pd**, this macro does not have a starred variant, and only typesets full fractions:

Example:

$$\frac{d^m f}{dx^m}$$

```
\begin{equation*}
  \td{f}{x^m}
\end{equation*}
```

\E {⟨*expression*⟩}

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

Example:

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

```
\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:

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

```
\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:

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

```
\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 [⟨*power*⟩]{⟨*expression*⟩}
\arcsin {⟨*expression*⟩}
\cos [⟨*power*⟩]{⟨*expression*⟩}
\arccos {⟨*expression*⟩}
\tan [⟨*power*⟩]{⟨*expression*⟩}
\arctan {⟨*expression*⟩}
\cot [⟨*power*⟩]{⟨*expression*⟩}
\sinh [⟨*power*⟩]{⟨*expression*⟩}
\cosh [⟨*power*⟩]{⟨*expression*⟩}
\tanh [⟨*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.* $\sin^2(\phi)$. When the argument is empty, no parentheses are emitted: $\cos$.

\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: $\log_2(x)$. As with the trigonometric functions, no parentheses are emitted if the mandatory argument is empty: $\log$.

\exp $\star${⟨*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:

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

Additionally, it is possible to force the $\exp$ mode by using the starred variant.

```
\min     *[⟨domain⟩]{⟨expression⟩}
\argmin  *[⟨domain⟩]{⟨expression⟩}
\max     *[⟨domain⟩]{⟨expression⟩}
\argmax  *[⟨domain⟩]{⟨expression⟩}
\sup     *[⟨domain⟩]{⟨expression⟩}
\inf     *[⟨domain⟩]{⟨expression⟩}
```

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:

$$\operatorname*{arg\,min}_{x\in\mathbb{R}_+} f(x)$$

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.* $\overline{uv}$), and **\vec** has been changed to match the `\vectorsym` command provided by isomath.

```
\Re      {⟨expression⟩}
\Im      {⟨expression⟩}
```

These commands typeset the real and imaginary part of a number. Standard use of the package takes definitions roughly from amsmath, while setting the `notation` option to `iso` changes the definitions to match ISO 80000-2.

3 Known issues

A list of current issues is available in the Github repository of this package[1], but as of the release of v0.5, 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 redifinition 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 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`.

5 Changes

v0.1

General: Initial version.

v0.1c

General: Moved package from docstrip to skdoc.

v0.1d

General: Fixed fatal documentation and package errors.

v0.1e

General: Added statistics commands.

v0.1g

General: Documentation fixes.

v0.2

General: Use expl3 functionality throughout the package.

v0.3

General: Added `\min`/`\max` and friends. Added `\pd`.

v0.3a

General: Added `\sinh`, `\cosh` and `\tanh`.

v0.3b

General: Detect empty arguments in trigonometric and logarithmic functions, fix `\ln`.

v0.4

General: Added `notation` option, macros for complex numbers.

v0.4a

General: Replaced deprecated/removed expl3 constructs.

v0.4b

General: Track expl3 changes (thanks to Joseph Wright).

v0.5

General: Added `\td`.

6 Index

Numbers written in italic 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.