

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:

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

```
\begin{equation*}
  \mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}.
\end{equation*}
```

`\norm` {*expression*}
`\abs` {*expression*}

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

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}} \sin(x) dx
\end{equation*}
```

`\pd` $\ast\{\langle function \rangle\}\{\langle var \rangle, \langle var \rangle, \dots\}$

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.

$$\backslash E \quad \{ \langle expression \rangle \}$$

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

Example:

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

```
\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:

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

```
\begin{equation*}
  \P{A\given B} =
  \frac{\P{B\given A}\P{A}}{\P{B}}
\end{equation*}
```

`\var {⟨expression⟩}`

`\cov {⟨expression⟩}{⟨expression⟩}`

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` [*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 `\vectorssym` 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.

4 Implementation

The package implementation is very simple. First, we do the standard $\LaTeX 2_{\epsilon}$ preamble thing, then we require some dependencies.

```
(package) 1 \RequirePackage{expl3,l3keys2e,xparse}
2 \ProvidesExplPackage{skmath}
3 {2013/04/09}{0.2}{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}}}
```

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

`\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}}%
22   \mkern1.5mu\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}}

```

We introduce a couple of helper macros for the `\pd` macro backend.

```

(package) 91 \prg_new_conditional:Npnn\_\_skmath_if_numerical:n#1{p,T,F,TF}{
92     \str_case_x:nnn{#1}{
93         {0}{\prg_return_true:}
94         {1}{\prg_return_true:}
95         {2}{\prg_return_true:}
96         {3}{\prg_return_true:}
97         {4}{\prg_return_true:}
98         {5}{\prg_return_true:}
99         {6}{\prg_return_true:}
100        {7}{\prg_return_true:}
101        {8}{\prg_return_true:}
102        {9}{\prg_return_true:}
103    }{\prg_return_false:}
104 }

(package)105 \cs_new_nopar:Npn\_\_skmath_pd_vars_sum:n#1{
106     \tl_clear:N\l_tmpa_tl

```

```

107 \int_zero:N\l_tmpa_int
108 \bool_set_true:N\l_tmpa_bool
109 \clist_set:Nn\l_tmpa_clist{#1}
110 \clist_map_inline:Nn\l_tmpa_clist{
111   \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
112   \int_compare:nTF{\seq_count:N\l_tmpa_seq<\c_two}{
113     \seq_put_right:Nn\l_tmpa_seq{1}
114   }
115   \seq_get_right:NN\l_tmpa_seq\l_tmpb_tl
116   \__skmath_if_numerical:nTF{\tl_use:N\l_tmpb_tl}{
117     \int_add:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
118   }{
119     \bool_set_false:N\l_tmpa_bool
120     \tl_if_empty:NF\l_tmpa_tl{\tl_put_right:Nn\l_tmpa_tl{+}}
121     \tl_put_right:Nx\l_tmpa_tl{\tl_use:N\l_tmpb_tl}
122   }
123 }
124 \int_compare:nTF{\l_tmpa_int>\c_zero}{\int_use:N\l_tmpa_int}
125 \bool_if:NF\l_tmpa_bool{
126   \int_compare:nTF{\l_tmpa_int>\c_zero}{+}
127   \tl_use:N\l_tmpa_tl
128 }
129 }

(package)130 \prg_new_conditional:Npnn\__skmath_pd_if_vars_sum_above_one:n#1{p,T,F,TF}{
131   \clist_set:Nn\l_tmpa_clist{#1}
132   \int_compare:nTF{\clist_count:N\l_tmpa_clist>\c_one}{
133     \prg_return_true:
134   }{
135     \clist_pop:NN\l_tmpa_clist\l_tmpa_tl
136     \seq_set_split:NnV\l_tmpa_seq{^}{\l_tmpa_tl}
137     \int_compare:nTF{\seq_count:N\l_tmpa_seq<\c_two}{
138       \prg_return_false:
139     }{
140       \prg_return_true:
141     }

```

```

142   }
143 }

(package)144 \cs_new_nopar:Npn\__skmath_pd_superscript_vars:n#1{
145   \clist_set:Nn\l_tmpa_clist{#1}
146   \clist_map_inline:Nn\l_tmpa_clist{
147     \partial
148     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
149     \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
150     \tl_use:N\l_tmpb_tl
151     \int_compare:nT{\seq_count:N\l_tmpa_seq>0}{
152       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
153       \c_math_superscript_token
154       \tl_use:N\l_tmpb_tl
155     }
156   }
157 }

(package)158 \cs_new_nopar:Npn\__skmath_pd_subscript_vars:n#1{
159   \clist_set:Nn\l_tmpa_clist{#1}
160   \clist_map_inline:Nn\l_tmpa_clist{
161     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
162     \seq_pop:NN\l_tmpa_seq\l_tmpa_tl
163     \int_set:Nn\l_tmpa_int{\c_one}
164     \int_compare:nT{\seq_count:N\l_tmpa_seq>\c_zero}{
165       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
166       \int_set:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
167     }
168     \prg_replicate:nn{\l_tmpa_int}{\tl_use:N\l_tmpa_tl}
169   }
170 }

(package)171 \cs_new_nopar:Npn\__skmath_pd_fraction:nn#1#2{
172   \frac{
173     \partial
174     \__skmath_pd_if_vars_sum_above_one:nT{#2}{

```

```

175         \c_math_superscript_token{\__skmath_pd_vars_sum:n{#2}}
176     }
177     {#1}
178 }{
179     \__skmath_pd_superscript_vars:n{#2}
180 }
181 }

(package)182 \cs_new_nopar:Npn\__skmath_pd_subscript:nn#1#2{
183     {#1}\c_math_subscript_token{
184         \__skmath_pd_subscript_vars:n{#2}
185     }
186 }

```

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

```
(package)187 \AtBeginDocument{%
```

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

`\bar`(no arguments)

```
(package)188 \DeclareDocumentCommand\bar{m}{%
189     \ensuremath{\mkern 1.5mu\overline{\mkern1.5mu{#1}\mkern1.5mu}\mkern 1.5mu}}
```

This is the partial derivative macro, but most of the functionality was defined as private macros earlier.

`\pd#1`: Boolean distinguishing between starred and unstarred variant

`#2`: Tokens representing a mathematical function

`#3`: Comma-separated list of variables, possibly with superscripts

```
(package)190 \DeclareDocumentCommand\pd{smm}{
191     \ensuremath{
192         \IfBooleanTF{#1}
193         {\__skmath_pd_subscript:nn{#2}{#3}}
194         {\__skmath_pd_fraction:nn{#2}{#3}}
195     }
196 }

```

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

`\d`(no arguments)

```
(package)197 \DeclareDocumentCommand\d{m}{\ensuremath{\,,\mathrm{d}}#1%
198 \peek_meaning_ignore_spaces:NT\d{\!}}
(package)199 }
```

Finally, we define a nicer way to denote vectors.

`\vec`(no arguments)

```
(package)200 \cs_set_eq:NN\vec\vectorssym
We end by declaring an option.
(package)201 \keys_define:nn{skmath}{
202   commonsets .code:n =
203   { \__skmath_define_common_sets: }
204 }
205 \ProcessKeysOptions{skmath}
(package)206 \endinput
```

5 Installation

The easiest way to install this package is using the package manager provided by your L^AT_EX 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 pdf_latex, *not* tex or latex.

6 Changes

vo.1

General: Initial version.

vo.1a

\d: Fixed obtuse errors.

vo.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.

vo.1c

General: Moved package from docstrip to skdoc.

vo.1d

General: Fixed fatal documentation and package errors.

vo.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.

vo.1f

\E: Fixed ‘Command \E already defined!’ error.

vo.1g

General: Documentation fixes.

vo.1h

\bar: Wrap in \AtBeginDocument.

v0.2	\E: Use \operatorname.
General: Use expl3 functionality throughout the package.	\P: Use \operatorname, use \cs_new_eq:NN instead of \let.
\cov: Use \operatorname.	\var: Use \operatorname.
\d: Use \peek_meaning_ignore_spaces instead of \ifnextchar.	\cs_new_eq:NN instead of \let.

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	__skmath_pd_superscript_vars:n
__skmath_define_common_sets:	<u>12</u>
<u>6</u>	__skmath_pd_vars_sum:n <u>11</u>
__skmath_if_numerical:nF <u>11</u>	
__skmath_if_numerical:nT <u>11</u>	A
__skmath_if_numerical:nTF <u>11</u>	\abs <u>2</u> , <u>7</u>
__skmath_if_numerical_p:n <u>11</u>	\arccos <u>5</u> , <u>10</u>
__skmath_pd_fraction:nn <u>13</u>	\arcsin <u>5</u> , <u>10</u>
__skmath_pd_if_vars_sum_above_one:n	\arctan <u>5</u> , <u>10</u>
<u>12</u>	
__skmath_pd_if_vars_sum_above_one:nT	B
<u>12</u>	\bar <u>6</u> , <u>14</u>
__skmath_pd_if_vars_sum_above_one:nTF	
<u>12</u>	C
__skmath_pd_if_vars_sum_above_one:nF	\csc <u>1</u> , <u>2</u> , <u>7</u>
<u>12</u>	commonsets (option) <u>1</u> , <u>2</u>
__skmath_pd_subscript:nn <u>14</u>	\cos <u>5</u> , <u>9</u>
__skmath_pd_subscript_vars:n	\cot <u>5</u> , <u>9</u>
<u>13</u>	\cov <u>4</u> , <u>8</u>

D
 $\backslash d$ [3](#), [14](#)

E
 $\backslash E$ [3](#), [7](#)
 $\backslash exp$ [5](#), [11](#)

F
 $\backslash frac$ [5](#), [11](#)

G
 $\backslash given$ [4](#), [8](#)

I
isomath (package) [6](#)

L
 $\backslash ln$ [5](#), [10](#)
 $\backslash log$ [5](#), [10](#)

M
Makefile (file) [15](#)

N
 $\backslash N$ [1](#), [2](#), [6](#)
 $\backslash norm$ [2](#), [7](#)

P
 $\backslash P$ [4](#), [8](#)
 $\backslash pd$ [3](#), [11](#), [14](#)

Q
 $\backslash Q$ [1](#), [2](#), [7](#)

R
 $\backslash R$ [1](#), [2](#), [7](#)

S
 $\backslash sin$ [5](#), [9](#)
skdoc (package) [15](#)
skmath.tex (file) [15](#)

T
 $\backslash tan$ [5](#), [9](#)

V
 $\backslash var$ [4](#), [8](#)
 $\backslash vec$ [6](#), [15](#)
 $\backslash vectorsym$ [6](#)

Z
 $\backslash Z$ [1](#), [2](#), [6](#)