

# The **skmath** package<sup>\*†</sup>

Simon Sigurdhsson [[sigurdhsson@gmail.com](mailto: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.

---

<sup>\*</sup>Available on <http://www.ctan.org/pkg/skmath>.

<sup>†</sup>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*}
```

$\backslash d$  {*variable*}

There is also a command  $\backslash d$ , 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}} \! \! \! \frac{\sin{x}}{x} \, \mathrm{d}{x}
\end{equation*}
```

$\backslash pd$  \*{*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.

$\backslash E$  {*expression*}

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

**Example:**

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

```
\begin{equation*}
\mathrm{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:

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

`\min`    \* $\langle domain \rangle \{ expression \}$   
`\argmin` \* $\langle domain \rangle \{ expression \}$   
`\max`    \* $\langle domain \rangle \{ expression \}$   
`\argmax` \* $\langle domain \rangle \{ expression \}$   
`\sup`    \* $\langle domain \rangle \{ expression \}$   
`\inf`    \* $\langle domain \rangle \{ expression \}$

The maximum/minimum macros have been redefined in a manner similar to the trigonometric functions. They typeset  $\langle expression \rangle$  inside curly brackets (the starred version omits the brackets), with the optional  $\langle domain \rangle$  typeset in a subscript after the operator (e.g.  $\min_{x \in \mathbb{R}_+} f(x)$ ). Argument variants are also provided, and the  $\langle expression \rangle$  is centered underneath the operator if possible:

$$\arg \min_{x \in \mathbb{R}_+} f(x)$$

## 2.4 Stylistic changes

Some commands have been redefined in a completely backwards-compatible way to improve the end result of their typesetting.

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

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 `\vectorssym` command provided by `isomath`.

### 3 Known issues

A list of current issues is available in the Github repository of this package<sup>1</sup>, 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<sub>ε</sub> 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,sumlimits,namelimits}{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}}}
```

---

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

**\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      \mkern-1.5mu\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
46 \cs_new_eq:NN\__skmath_min:\min
47 \cs_new_eq:NN\__skmath_max:\max
48 \cs_new_eq:NN\__skmath_sup:\sup
49 \cs_new_eq:NN\__skmath_inf:\inf

```

### $\backslash\sin$ (no arguments)

```

(package) 50 \RenewDocumentCommand\sin{om}{%
51   \IfNoValueTF{#1}
52     {\ensuremath{\__skmath_sin:\left(#2\right)}}
53     {\ensuremath{\__skmath_sin:\c_math_superscript_token{#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:\c_math_superscript_token{#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:\c_math_superscript_token{#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:\c_math_superscript_token{#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:\c_math_subscript_token{#1}\left(\#2\right)}}%  
86 }
```

### $\backslash$ exp(no arguments)

```
(package) 87 \RenewDocumentCommand\exp{m}{\ensuremath{\mathchoice%
```

```

88 {e\c_math_superscript_token{#1}}%
89 {\__skmath_exp:\left(#1\right)}%
90 {\__skmath_exp:\left(#1\right)}%
91 {\__skmath_exp:\left(#1\right)}%
92 }}

(package) 93 \cs_new_nopar:Npn\__skmath_minmax_backend:nnnn#1#2#3#4{%
94   \use:c{__skmath_#1:}
95   \IfNoValueF{#3}{
96     \c_math_subscript_token{
97       \mathchoice{\mathclap{#3}}{#3}{#3}{#3}
98     }
99   }
100   \IfBooleanTF{#2}{#4}{\left\{#4\right\}}
101 }

\min(no arguments)

(package) 102 \RenewDocumentCommand\min{som}{%
103   \ensuremath{\__skmath_minmax_backend:nnnn{min}{#1}{#2}{#3}}
104 }

\argmin(no arguments)

(package) 105 \NewDocumentCommand\argmin{som}{%
106   \ensuremath{\arg\__skmath_minmax_backend:nnnn{min}{#1}{#2}{#3}}
107 }

\max(no arguments)

(package) 108 \RenewDocumentCommand\max{som}{%
109   \ensuremath{\__skmath_max_minbackend:nnnn{max}{#1}{#2}{#3}}
110 }

\argmax(no arguments)

(package) 111 \NewDocumentCommand\argmax{som}{%

```

```

112 \ensuremath{\arg\__skmath_minmax_backend:nnnn{max}{#1}{#2}{#3}}
113 }

```

**\sup**(no arguments)

```

(package) 114 \RenewDocumentCommand\sup{som}{%
115 \ensuremath{\__skmath_minmax_backend:nnnn{sup}{#1}{#2}{#3}}
116 }

```

**\inf**(no arguments)

```

(package) 117 \RenewDocumentCommand\inf{som}{%
118 \ensuremath{\__skmath_minmax_backend:nnnn{inf}{#1}{#2}{#3}}
119 }

```

The fraction command is modified to improve typesetting.

**\frac**(no arguments)

```

(package) 120 \RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}{}%
121 \displaystyle #1}{\displaystyle #2}}

```

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

```

(package) 122 \prg_new_conditional:Npnn\__skmath_if_numerical:n#1{p,T,F,TF}{
123 \str_case_x:nnn{#1}{
124 {0}{\prg_return_true:}
125 {1}{\prg_return_true:}
126 {2}{\prg_return_true:}
127 {3}{\prg_return_true:}
128 {4}{\prg_return_true:}
129 {5}{\prg_return_true:}
130 {6}{\prg_return_true:}
131 {7}{\prg_return_true:}
132 {8}{\prg_return_true:}
133 {9}{\prg_return_true:}
134 }{\prg_return_false:}
135 }

```

```

(package) 136 \cs_new_nopar:Npn\__skmath_pd_vars_sum:n#1{
137   \tl_clear:N\l_tmpa_tl
138   \int_zero:N\l_tmpa_int
139   \bool_set_true:N\l_tmpa_bool
140   \clist_set:Nn\l_tmpa_clist{#1}
141   \clist_map_inline:Nn\l_tmpa_clist{
142     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
143     \int_compare:nT{\seq_count:N\l_tmpa_seq<\c_two}{
144       \seq_put_right:Nn\l_tmpa_seq{1}
145     }
146     \seq_get_right:NN\l_tmpa_seq\l_tmpb_tl
147     \__skmath_if_numerical:nTF{\tl_use:N\l_tmpb_tl}{
148       \int_add:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
149     }{
150       \bool_set_false:N\l_tmpa_bool
151       \tl_if_empty:NF\l_tmpa_tl{\tl_put_right:Nn\l_tmpa_tl{+}}
152       \tl_put_right:Nx\l_tmpa_tl{\tl_use:N\l_tmpb_tl}
153     }
154   }
155   \int_compare:nT{\l_tmpa_int>\c_zero}{\int_use:N\l_tmpa_int}
156   \bool_if:NF\l_tmpa_bool{
157     \int_compare:nT{\l_tmpa_int>\c_zero}{+}
158     \tl_use:N\l_tmpa_tl
159   }
160 }

(package) 161 \prg_new_conditional:Npnn\__skmath_pd_if_vars_sum_above_one:n#1{p,T,F,TF}{
162   \clist_set:Nn\l_tmpa_clist{#1}
163   \int_compare:nTF{\clist_count:N\l_tmpa_clist>\c_one}{
164     \prg_return_true:
165   }{
166     \clist_pop:NN\l_tmpa_clist\l_tmpa_tl
167     \seq_set_split:NnV\l_tmpa_seq{^}{\l_tmpa_tl}
168     \int_compare:nTF{\seq_count:N\l_tmpa_seq<\c_two}{
169       \prg_return_false:
170     }{

```

```

171     \prg_return_true:
172   }
173 }
174 }

(package) 175 \cs_new_nopar:Npn\__skmath_pd_superscript_vars:n#1{
176   \clist_set:Nn\l_tmpa_clist{#1}
177   \clist_map_inline:Nn\l_tmpa_clist{
178     \partial
179     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
180     \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
181     \tl_use:N\l_tmpb_tl
182     \int_compare:nT{\seq_count:N\l_tmpa_seq>0}{
183       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
184       \c_math_superscript_token
185       \tl_use:N\l_tmpb_tl
186     }
187   }
188 }

(package) 189 \cs_new_nopar:Npn\__skmath_pd_subscript_vars:n#1{
190   \clist_set:Nn\l_tmpa_clist{#1}
191   \clist_map_inline:Nn\l_tmpa_clist{
192     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
193     \seq_pop:NN\l_tmpa_seq\l_tmpa_tl
194     \int_set:Nn\l_tmpa_int{\c_one}
195     \int_compare:nT{\seq_count:N\l_tmpa_seq>\c_zero}{
196       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
197       \int_set:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
198     }
199     \prg_replicate:nn{\l_tmpa_int}{\tl_use:N\l_tmpa_tl}
200   }
201 }

(package) 202 \cs_new_nopar:Npn\__skmath_pd_fraction:nn#1#2{
203   \frac{

```

```

204     \partial
205     \__skmath_pd_if_vars_sum_above_one:nT{#2}{
206         \c_math_superscript_token{\__skmath_pd_vars_sum:n{#2}}
207     }
208     {#1}
209 }{
210     \__skmath_pd_superscript_vars:n{#2}
211 }
212 }

(package) 213 \cs_new_nopar:Npn\__skmath_pd_subscript:nn#1#2{
214     {#1}\c_math_subscript_token{
215         \__skmath_pd_subscript_vars:n{#2}
216     }
217 }

```

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

```
(package) 218 \AtBeginDocument{%
```

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

**`\bar`**(no arguments)

```
(package) 219 \DeclareDocumentCommand\bar{m}{%
220     \ensuremath{\mkern 1.5mu\overline{\mkern-1.5mu{#1}\mkern-1.5mu}\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) 221 \DeclareDocumentCommand\pd{smm}{
222     \ensuremath{
223         \IfBooleanTF{#1}
224         {\__skmath_pd_subscript:nn{#2}{#3}}
225         {\__skmath_pd_fraction:nn{#2}{#3}}

```



```

226 }
227 }

```

We introduce a command to typeset the differential part of integrals, shamefully stolen from an answer on  $\text{\TeX}$ .SE.

$\backslash d$ (no arguments)

```

(package) 228 \DeclareDocumentCommand\d{m}{\ensuremath{\backslash,\mathrm{d}}#1%
229             \peek_meaning_ignore_spaces:NT\d{\!}}
(package) 230 }

```

Finally, we define a nicer way to denote vectors.

$\backslash vec$ (no arguments)

```

(package) 231 \cs_set_eq:NN\vec\vecsym
          We end by declaring an option.
(package) 232 \keys_define:nn{skmath}{
233     commonsets .code:n =
234     { \__skmath_define_common_sets: }
235 }
236 \ProcessKeysOptions{skmath}
(package) 237 \endinput

```

## 5 Installation

The easiest way to install this package is using the package manager provided by your  $\text{\LaTeX}$  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

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

v0.1g	\E: Use \operatorname.
General: Documentation fixes.	\P: Use \operatorname, use \cs_new_eq:NN instead of \let.
v0.1h	
\bar: Wrap in \AtBeginDocument.	\var: Use \operatorname.
v0.2	\vec: Use \cs_new_eq:NN instead of \let.
General: Use expl3 functionality throughout the package.	
\cov: Use \operatorname.	v0.2a
\d: Use \peek_meaning_ignore_spaces instead of \@ifnextchar.	General: Load amsmath with more limits options.

## 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.

<b>Symbols</b>	\__skmath_pd_if_vars_sum_above_one:nTF
\__skmath_define_common_sets:	<u>14</u>
<u>7</u>	\__skmath_pd_if_vars_sum_above_one_p:n
\__skmath_if_numerical:nF <u>13</u>	<u>14</u>
\__skmath_if_numerical:nT <u>13</u>	\__skmath_pd_subscript:nn <u>16</u>
\__skmath_if_numerical:nTF <u>13</u>	\__skmath_pd_subscript_vars:n
\__skmath_if_numerical_p:n <u>13</u>	<u>15</u>
\__skmath_minmax_backend:nnn	\__skmath_pd_superscript_vars:n
<u>12</u>	<u>15</u>
\__skmath_pd_fraction:nn <u>15</u>	\__skmath_pd_vars_sum:n <u>13</u>
\__skmath_pd_if_vars_sum_above_one:nF	A
<u>14</u>	<del>abs</del> <u>7, 8</u>
\__skmath_pd_if_vars_sum_above_one:nT	\arccos <u>5, 11</u>
<u>14</u>	

$\backslash$ arcsin [5](#), [11](#)

$\backslash$ arctan [5](#), [11](#)

$\backslash$ argmax [6](#), [12](#)

$\backslash$ argmin [6](#), [12](#)

## B

$\backslash$ bar [6](#), [16](#)

## C

$\backslash$ C [1](#), [2](#), [8](#)

commonsets (option) [1](#), [2](#)

$\backslash$ cos [5](#), [10](#)

$\backslash$ cot [5](#), [10](#)

$\backslash$ cov [4](#), [9](#)

## D

$\backslash$ d [3](#), [16](#), [17](#)

## E

$\backslash$ E [3](#), [8](#)

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

## F

$\backslash$ frac [6](#), [13](#)

## G

$\backslash$ given [4](#), [8](#), [9](#)

## I

$\backslash$ inf [6](#), [13](#)

isomath (package) [6](#)

## L

$\backslash$ ln [5](#), [11](#)

$\backslash$ log [5](#), [11](#)

## M

Makefile (file) [17](#)

$\backslash$ max [6](#), [12](#)

$\backslash$ min [6](#), [12](#)

## N

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

$\backslash$ norm [2](#), [8](#)

## P

$\backslash$ P [4](#), [8](#)

$\backslash$ pd [3](#), [13](#), [16](#)

## Q

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

## R

$\backslash$ R [1](#), [2](#), [8](#)

## S

$\backslash$ sin [5](#), [10](#)

skdoc (package) [18](#)

skmath.tex (file) [17](#)

$\backslash$ sup [6](#), [13](#)

## T

$\backslash$ tan [5](#), [10](#)

## V

$\backslash$ var [4](#), [9](#)

$\backslash$ vec [6](#), [17](#)

$\backslash$ vectorsym [6](#)

## Z

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