

The **skmath** package^{*†}

Simon Sigurdhsson [sigurdhsson@gmail.com]

Version 0.3a

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 v0.3a, 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` $\langle expression \rangle$
`\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:

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

```
\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:

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

$$\int_{\mathbb{R}} \sin(x) x \, dx$$
$$\backslash\mathrm{pd} \quad * \{ \langle function \rangle \} \{ \langle var \rangle, \langle var \rangle, \dots \}$$

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.

$\backslash\sin$ [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backslasharcsin { $\langle expression \rangle$ }
 \backslashcos [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backlarccos { $\langle expression \rangle$ }
 \backtan [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backarctan { $\langle expression \rangle$ }
 \backcot [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backsinh [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backcosh [$\langle power \rangle$]{ $\langle expression \rangle$ }
 \backtanh [$\langle power \rangle$]{ $\langle expression \rangle$ }

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. $\sin^2(\phi)$.

\backln { $\langle expression \rangle$ }

The natural logarithm macro \backln has also been redefined to require an argument which is typeset as the argument of the logarithm.

\backlog [$\langle base \rangle$]{ $\langle expression \rangle$ }

The related macro \backlog is redefined in a similar way, but also accepts an optional argument denoting the base of the logarithm: $\log_2(x)$.

\backexp *{ $\langle expression \rangle$ }

The exponential, \backexp , 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` * $\langle domain \rangle \{ expression \}$
`\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 package¹, but as of the release of v0.3a, 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_ϵ preamble thing, then we require some dependencies.

```
(package) 1 \RequirePackage{expl3,l3keys2e,xparse}
2 \ProvidesExplPackage{skmath}
3 {2013/06/24}{0.3a}{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)

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

```
(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\{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_sinh:\sinh
44 \cs_new_eq:NN\__skmath_cosh:\cosh
45 \cs_new_eq:NN\__skmath_tanh:\tanh
46 \cs_new_eq:NN\__skmath_ln:\log
47 \cs_new_eq:NN\__skmath_log:\log
48 \cs_new_eq:NN\__skmath_exp:\exp
49 \cs_new_eq:NN\__skmath_min:\min
50 \cs_new_eq:NN\__skmath_max:\max
51 \cs_new_eq:NN\__skmath_sup:\sup
52 \cs_new_eq:NN\__skmath_inf:\inf

```

\sin(no arguments)

```

(package) 53 \RenewDocumentCommand\sine{om}{%
54     \IfNoValueTF{#1}
55     {\ensuremath{\__skmath_sine:\left(#2\right)}}
56     {\ensuremath{\__skmath_sine:\c_math_superscript_token{#1}\left(#2\right)}}%
57 }

```

\cos(no arguments)

```

(package) 58 \RenewDocumentCommand\cosine{om}{%
59     \IfNoValueTF{#1}
60     {\ensuremath{\__skmath_cos:\left(#2\right)}}
61     {\ensuremath{\__skmath_cos:\c_math_superscript_token{#1}\left(#2\right)}}%
62 }

```

\tan(no arguments)

```

(package) 63 \RenewDocumentCommand\tan{om}{%
64     \IfNoValueTF{#1}
65     {\ensuremath{\__skmath_tan:\left(#2\right)}}
66     {\ensuremath{\__skmath_tan:\c_math_superscript_token{#1}\left(#2\right)}}%
67 }

```

\cot(no arguments)

```

(package) 68 \RenewDocumentCommand\cot{om}{%
69     \IfNoValueTF{#1}
70     {\ensuremath{\_\_skmath_cot:\left(#2\right)}}
71     {\ensuremath{\_\_skmath_cot:\c_math_superscript_token{#1}\left(#2\right)}}}%
72 }

```

\arcsin(no arguments)

```

(package) 73 \RenewDocumentCommand\arcsin{m}{%
74     \ensuremath{\_\_skmath_arcsin:\left(#1\right)}}%
75 }

```

\arccos(no arguments)

```

(package) 76 \RenewDocumentCommand\arccos{m}{%
77     \ensuremath{\_\_skmath_arccos:\left(#1\right)}}%
78 }

```

\arctan(no arguments)

```

(package) 79 \RenewDocumentCommand\arctan{m}{%
80     \ensuremath{\_\_skmath_arctan:\left(#1\right)}}%
81 }

```

\sinh(no arguments)

```

(package) 82 \RenewDocumentCommand\sinh{om}{%
83     \IfNoValueTF{#1}
84     {\ensuremath{\_\_skmath_sinh:\left(#2\right)}}
85     {\ensuremath{\_\_skmath_sinh:\c_math_superscript_token{#1}\left(#2\right)}}}%
86 }

```

\cosh(no arguments)

```

(package) 87 \RenewDocumentCommand\cosh{om}{%
88     \IfNoValueTF{#1}
89     {\ensuremath{\__skmath_cosh:\left(#2\right)}}
90     {\ensuremath{\__skmath_cosh:\c_math_superscript_token{#1}\left(#2\right)}}}%
91 }

```

\tanh(no arguments)

```

(package) 92 \RenewDocumentCommand\tanh{om}{%
93     \IfNoValueTF{#1}
94     {\ensuremath{\__skmath_tanh:\left(#2\right)}}
95     {\ensuremath{\__skmath_tanh:\c_math_superscript_token{#1}\left(#2\right)}}}%
96 }

```

\ln(no arguments)

```

(package) 97 \RenewDocumentCommand\ln{m}{%
98     \ensuremath{\__skmath_ln:\left(#1\right)}}%
99 }

```

\log(no arguments)

```

(package) 100 \RenewDocumentCommand\log{om}{%
101     \IfNoValueTF{#1}
102     {\ensuremath{\__skmath_log:\left(#2\right)}}
103     {\ensuremath{\__skmath_log:\c_math_subscript_token{#1}\left(#2\right)}}}%
104 }

```

\exp(no arguments)

```

(package) 105 \RenewDocumentCommand\exp{sm}{\ensuremath{
106     \IfBooleanTF{#1}{
107         \__skmath_exp:\left(#2\right)
108     }{
109         \mathchoice
110         {e\c_math_superscript_token{#2}}

```

```

111         {\__skmath_exp:\left(#2\right)}
112         {\__skmath_exp:\left(#2\right)}
113         {\__skmath_exp:\left(#2\right)}
114     }
115 }}

(package) 116 \cs_new_nopar:Npn\__skmath_minmax_backend:nnnn#1#2#3#4{%
117     \use:c{__skmath_#1:}
118     \IfNoValueF{#3}{
119         \c_math_subscript_token{
120             \mathchoice{\mathclap{#3}}{#3}{#3}{#3}
121         }
122     }
123     \IfBooleanTF{#2}{#4}{\left\{#4\right\}}
124 }

\min(no arguments)

(package) 125 \RenewDocumentCommand\min{som}{%
126     \ensuremath{\__skmath_minmax_backend:nnnn{min}{#1}{#2}{#3}}
127 }

\argmin(no arguments)

(package) 128 \NewDocumentCommand\argmin{som}{%
129     \ensuremath{\arg\__skmath_minmax_backend:nnnn{min}{#1}{#2}{#3}}
130 }

\max(no arguments)

(package) 131 \RenewDocumentCommand\max{som}{%
132     \ensuremath{\__skmath_max_minbackend:nnnn{max}{#1}{#2}{#3}}
133 }

\argmax(no arguments)

```

```

(package)134 \NewDocumentCommand\argmax{som}{%
135   \ensuremath{\arg\__skmath_minmax_backend:nnnn{max}{#1}{#2}{#3}}
136 }

```

\sup(no arguments)

```

(package)137 \RenewDocumentCommand\sup{som}{%
138   \ensuremath{\__skmath_minmax_backend:nnnn{sup}{#1}{#2}{#3}}
139 }

```

\inf(no arguments)

```

(package)140 \RenewDocumentCommand\inf{som}{%
141   \ensuremath{\__skmath_minmax_backend:nnnn{inf}{#1}{#2}{#3}}
142 }

```

The fraction command is modified to improve typesetting.

\frac(no arguments)

```

(package)143 \RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}{}%
144   {\displaystyle #1}{\displaystyle #2}}

```

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

```

(package)145 \prg_new_conditional:Npnn\__skmath_if_numerical:n#1{p,T,F,TF}{
146   \str_case_x:nnn{#1}{
147     {0}{\prg_return_true:}
148     {1}{\prg_return_true:}
149     {2}{\prg_return_true:}
150     {3}{\prg_return_true:}
151     {4}{\prg_return_true:}
152     {5}{\prg_return_true:}
153     {6}{\prg_return_true:}
154     {7}{\prg_return_true:}
155     {8}{\prg_return_true:}
156     {9}{\prg_return_true:}
157   }{\prg_return_false:}

```

```

158 }

(package)159 \cs_new_nopar:Npn\__skmath_pd_vars_sum:n#1{
160   \tl_clear:N\l_tmpa_tl
161   \int_zero:N\l_tmpa_int
162   \bool_set_true:N\l_tmpa_bool
163   \clist_set:Nn\l_tmpa_clist{#1}
164   \clist_map_inline:Nn\l_tmpa_clist{
165     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
166     \int_compare:nT{\seq_count:N\l_tmpa_seq<\c_two}{
167       \seq_put_right:Nn\l_tmpa_seq{1}
168     }
169     \seq_get_right:NN\l_tmpa_seq\l_tmpb_tl
170     \__skmath_if_numerical:nTF{\tl_use:N\l_tmpb_tl}{
171       \int_add:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
172     }{
173       \bool_set_false:N\l_tmpa_bool
174       \tl_if_empty:NF\l_tmpa_tl{\tl_put_right:Nn\l_tmpa_tl{+}}
175       \tl_put_right:Nx\l_tmpa_tl{\tl_use:N\l_tmpb_tl}
176     }
177   }
178   \int_compare:nT{\l_tmpa_int>\c_zero}{\int_use:N\l_tmpa_int}
179   \bool_if:NF\l_tmpa_bool{
180     \int_compare:nT{\l_tmpa_int>\c_zero}{+}
181     \tl_use:N\l_tmpa_tl
182   }
183 }

(package)184 \prg_new_conditional:Npnn\__skmath_pd_if_vars_sum_above_one:n#1{p,T,F,TF}{
185   \clist_set:Nn\l_tmpa_clist{#1}
186   \int_compare:nTF{\clist_count:N\l_tmpa_clist>\c_one}{
187     \prg_return_true:
188   }{
189     \clist_pop:NN\l_tmpa_clist\l_tmpa_tl
190     \seq_set_split:NnV\l_tmpa_seq{^}{\l_tmpa_tl}
191     \int_compare:nTF{\seq_count:N\l_tmpa_seq<\c_two}{

```

```

192     \prg_return_false:
193   }{
194     \prg_return_true:
195   }
196 }
197 }

(package)198 \cs_new_nopar:Npn\__skmath_pd_superscript_vars:n#1{
199   \clist_set:Nn\l_tmpa_clist{#1}
200   \clist_map_inline:Nn\l_tmpa_clist{
201     \partial
202     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
203     \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
204     \tl_use:N\l_tmpb_tl
205     \int_compare:nT{\seq_count:N\l_tmpa_seq>0}{
206       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
207       \c_math_superscript_token
208       \tl_use:N\l_tmpb_tl
209     }
210   }
211 }

(package)212 \cs_new_nopar:Npn\__skmath_pd_subscript_vars:n#1{
213   \clist_set:Nn\l_tmpa_clist{#1}
214   \clist_map_inline:Nn\l_tmpa_clist{
215     \seq_set_split:Nnn\l_tmpa_seq{^}{##1}
216     \seq_pop:NN\l_tmpa_seq\l_tmpa_tl
217     \int_set:Nn\l_tmpa_int{\c_one}
218     \int_compare:nT{\seq_count:N\l_tmpa_seq>\c_zero}{
219       \seq_pop:NN\l_tmpa_seq\l_tmpb_tl
220       \int_set:Nn\l_tmpa_int{\tl_use:N\l_tmpb_tl}
221     }
222     \prg_replicate:nn{\l_tmpa_int}{\tl_use:N\l_tmpa_tl}
223   }
224 }

```



```

(package)225 \cs_new_nopar:Npn\__skmath_pd_fraction:nn#1#2{
226   \frac{
227     \partial
228     \__skmath_pd_if_vars_sum_above_one:nT{#2}{
229       \c_math_superscript_token{\__skmath_pd_vars_sum:n{#2}}
230     }
231     {#1}
232   }{
233     \__skmath_pd_superscript_vars:n{#2}
234   }
235 }

(package)236 \cs_new_nopar:Npn\__skmath_pd_subscript:nn#1#2{
237   {#1}\c_math_subscript_token{
238     \__skmath_pd_subscript_vars:n{#2}
239   }
240 }

```

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

```
(package)241 \AtBeginDocument{%
```

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

`\bar`(no arguments)

```

(package)242 \DeclareDocumentCommand\bar{m}{%
243   \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)244 \DeclareDocumentCommand\pd{smm}{
245   \ensuremath{
246     \IfBooleanTF{#1}

```

```

247         {\__skmath_pd_subscript:nn{#2}{#3}}
248         {\__skmath_pd_fraction:nn{#2}{#3}}
249     }
250 }

```

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

`\d`(no arguments)

```

⟨package⟩251 \DeclareDocumentCommand\d{m}{\ensuremath{\,,\mathrm{d}}#1%
252                                     \peek_meaning_ignore_spaces:NT\d{\!}}
⟨package⟩253 }

```

Finally, we define a nicer way to denote vectors.

`\vec`(no arguments)

```

⟨package⟩254 \cs_set_eq:NN\vec\vectorssym
    We end by declaring an option.
⟨package⟩255 \keys_define:nn{skmath}{
256     commonsets .code:n =
257         {\__skmath_define_common_sets:}
258     }
259 \ProcessKeysOptions{skmath}
⟨package⟩260 \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 `mktextlsr`.

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

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

v0.1f	\P: Use <code>\operatorname</code> , use <code>\cs_new_eq:NN</code> instead of <code>\let</code> .
\E: Fixed ‘Command <code>\E</code> already defined!’ error.	\var: Use <code>\operatorname</code> .
v0.1g	\vec: Use <code>\cs_new_eq:NN</code> instead of <code>\let</code> .
General: Documentation fixes.	
v0.1h	v0.2a
\bar: Wrap in <code>\AtBeginDocument</code> .	General: Load <code>amsmath</code> with more limits options.
v0.2	v0.3
General: Use <code>expl3</code> functionality throughout the package.	General: Added <code>\min/\max</code> and friends. Added <code>\pd</code> .
\cov: Use <code>\operatorname</code> .	v0.3a
\d: Use <code>\peek_meaning_ignore_spaces:NT</code> instead of <code>\@ifnextchar</code> .	General: Added <code>\sinh</code> , <code>\cosh</code> and <code>\tanh</code> .
\E: Use <code>\operatorname</code> .	\exp: Added starred variant.

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_if_numerical_p:n 14
__skmath_define_common_sets: 7	__skmath_minmax_backend:nnn 13
__skmath_if_numerical:nF 14	__skmath_pd_fraction:nn 16
__skmath_if_numerical:nT 14	__skmath_pd_if_vars_sum_above_one:nF 15
__skmath_if_numerical:nTF 14	

<code>__skmath_pd_if_vars_sum_above_</code>	<code>\ne:nT</code>
<code><u>15</u></code>	<code>\E <u>3</u>,<u>8</u></code>
<code>__skmath_pd_if_vars_sum_above_</code>	<code>\expF<u>12</u></code>
<code><u>15</u></code>	
<code>__skmath_pd_if_vars_sum_above_</code>	<code>\ne:p:n</code>
<code><u>15</u></code>	<code>\frac <u>6</u>,<u>14</u></code>
<code>__skmath_pd_subscript:nn</code>	<code><u>17</u></code>
<code>__skmath_pd_subscript_vars:n</code>	<code>G</code>
<code><u>16</u></code>	<code>\given <u>4</u>,<u>8</u>,<u>9</u></code>
<code>__skmath_pd_superscript_vars:n</code>	
<code><u>16</u></code>	<code>I</code>
<code>__skmath_pd_vars_sum:n</code>	<code><u>15</u></code>
	<code>\inf <u>6</u>,<u>14</u></code>
	<code>isomath (package) <u>6</u></code>
<code>A</code>	
<code>\abs <u>2</u>,<u>8</u></code>	<code>L</code>
<code>\arccos <u>5</u>,<u>11</u></code>	<code>\ln <u>5</u>,<u>12</u></code>
<code>\arcsin <u>5</u>,<u>11</u></code>	<code>\log <u>5</u>,<u>12</u></code>
<code>\arctan <u>5</u>,<u>11</u></code>	
<code>\argmax <u>6</u>,<u>13</u></code>	<code>M</code>
<code>\argmin <u>6</u>,<u>13</u></code>	<code>Makefile (file) <u>18</u></code>
	<code>\max <u>6</u>,<u>13</u></code>
<code>B</code>	<code>\min <u>6</u>,<u>13</u></code>
<code>\bar <u>6</u>,<u>17</u></code>	
	<code>N</code>
<code>C</code>	<code>\N <u>1</u>,<u>2</u>,<u>7</u></code>
<code>\C <u>1</u>,<u>2</u>,<u>8</u></code>	<code>\norm <u>2</u>,<u>8</u></code>
<code>commonsets (option) <u>1</u>,<u>2</u></code>	
<code>\cos <u>5</u>,<u>10</u></code>	<code>P</code>
<code>\cosh <u>5</u>,<u>11</u></code>	<code>\P <u>4</u>,<u>8</u></code>
<code>\cot <u>5</u>,<u>10</u></code>	<code>\pd <u>3</u>,<u>14</u>,<u>17</u></code>
<code>\cov <u>4</u>,<u>9</u></code>	
	<code>Q</code>
<code>D</code>	<code>\Q <u>1</u>,<u>2</u>,<u>7</u></code>
<code>\d <u>3</u>,<u>17</u>,<u>18</u></code>	
	<code>R</code>

$\backslash R$ 1, 2, 8

S

$\backslash sin$ 5, 10

$\backslash sinh$ 5, 11

skdoc (package) 19

skmath.tex (file) 18, 19

$\backslash sup$ 6, 14

T

$\backslash tan$ 5, 10

$\backslash tanh$ 5, 12

V

$\backslash var$ 4, 9

$\backslash vec$ 6, 18

$\backslash vectorsym$ 6

Z

$\backslash Z$ 1, 2, 7