

The skmath package

Simon Sigurdhsson
sigurdhsson@gmail.com

CTAN: <http://www.ctan.org/pkg/skmath>

VC: <https://github.com/urdh/skmath>

Version
805672b
2012-12-15

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 1.1, 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.

2.2 New commands

The package defines a number of new commands that aid in typesetting certain mathematical formulae.

<code>\N</code> <code>\Z</code> <code>\Q</code> <code>\R</code> <code>\C</code>

These commands are only available if the `commonsets` option is given. They typeset the set of natural, integer, rational, real and complex numbers respectively:

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

$\backslash\text{norm}\{\langle expression \rangle\}$
 $\backslash\text{abs}\{\langle expression \rangle\}$

The commands $\backslash\text{norm}$ and $\backslash\text{abs}$, 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 $\backslash\text{norm}$:

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

$\backslash\text{d}\{\langle variable \rangle\}$

There is also a command $\backslash\text{d}$, with one mandatory argument, that typesets the differential part of an integral:

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

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.

$\backslash\text{sin}[\langle power \rangle]\{\langle expression \rangle\}$ $\backslash\text{arcsin}\{\langle expression \rangle\}$
 $\backslash\text{cos}[\langle power \rangle]\{\langle expression \rangle\}$ $\backslash\text{arccos}\{\langle expression \rangle\}$
 $\backslash\text{tan}[\langle power \rangle]\{\langle expression \rangle\}$ $\backslash\text{arctan}\{\langle expression \rangle\}$
 $\backslash\text{cot}[\langle power \rangle]\{\langle expression \rangle\}$

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

$\backslash\text{ln}\{\langle expression \rangle\}$

The natural logarithm macro $\backslash\text{ln}$ has also been redefined to require an argument which is typeset as the argument of the logarithm.

$\backslash\text{log}[\langle base \rangle]\{\langle expression \rangle\}$

The related macro $\backslash\text{log}$ 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 Implementation

The package implementation is very simple. First, we require some dependencies.

```
1 \RequirePackage{xparse}
2 \RequirePackage[intlimits]{amsmath}
3 \RequirePackage{kvoptions,amssymb,mathtools,xfrac,
  isomath}
```

We begin by declaring an option.

```
4 \SetupKeyvalOptions{family=skmath,prefix=skmath@}
5 \DeclareBoolOption[false]{commonsets}
6 \ProcessKeyvalOptions*
```

We optionally provide commands to typeset common sets

```
7 \ifskmath@commonsets
```

`\N`

```
8 \NewDocumentCommand\N{}{\ensuremath{\mathbb{N}}}
```

`\Z`

```
9 \NewDocumentCommand\Z{}{\ensuremath{\mathbb{Z}}}
```

\Q

```
10 \NewDocumentCommand\Q{}{\ensuremath{\mathbb{Q}}}
```

\R

```
11 \NewDocumentCommand\R{}{\ensuremath{\mathbb{R}}}
```

\C

```
12 \NewDocumentCommand\C{}{\ensuremath{\mathbb{C}}}
```

```
13 \fi
```

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

\abs

```
14 \DeclarePairedDelimiter\abs{\lvert}{\rvert}
```

\norm

```
15 \DeclarePairedDelimiter\norm{\lVert}{\rVert}
```

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

```
16 \let\skmath@sin\sin
17 \let\skmath@cos\cos
18 \let\skmath@tan\tan
19 \let\skmath@cot\cot
20 \let\skmath@arcsin\arcsin
21 \let\skmath@arccos\arccos
22 \let\skmath@arccos\arctan
23 \let\skmath@ln\log
24 \let\skmath@log\log
25 \let\skmath@exp\exp
```

\sin

```
26 \RenewDocumentCommand\sin{om}{%
27   \IfNoValueTF{#1}
28     {\ensuremath{\skmath@sin\left(#2\right)}}
29     {\ensuremath{\skmath@sin^{#1}\left(#2\right)}}}%
30 }
```

`\cos`

```
31 \RenewDocumentCommand\cos{om}{%  
32   \IfNoValueTF{#1}  
33     {\ensuremath{\skmath@cos\left(#2\right)}}  
34     {\ensuremath{\skmath@cos^{#1}\left(#2\right)}}%  
35 }
```

`\tan`

```
36 \RenewDocumentCommand\tan{om}{%  
37   \IfNoValueTF{#1}  
38     {\ensuremath{\skmath@tan\left(#2\right)}}  
39     {\ensuremath{\skmath@tan^{#1}\left(#2\right)}}%  
40 }
```

`\cot`

```
41 \RenewDocumentCommand\cot{om}{%  
42   \IfNoValueTF{#1}  
43     {\ensuremath{\skmath@cot\left(#2\right)}}  
44     {\ensuremath{\skmath@cot^{#1}\left(#2\right)}}%  
45 }
```

`\arcsin`

```
46 \RenewDocumentCommand\arcsin{m}{%  
47   \ensuremath{\skmath@arcsin\left(#1\right)}}%  
48 }
```

`\arccos`

```
49 \RenewDocumentCommand\arccos{m}{%  
50   \ensuremath{\skmath@arccos\left(#1\right)}}%  
51 }
```

`\arctan`

```
52 \RenewDocumentCommand\arctan{m}{%  
53   \ensuremath{\skmath@arctan\left(#1\right)}}%  
54 }
```

`\ln`

```
55 \RenewDocumentCommand\ln{m}{%  
56   \ensuremath{\skmath@ln\left(#1\right)}}%  
57 }
```

`\log`

```
58 \RenewDocumentCommand\log{om}{%  
59   \IfNoValueTF{#1}  
60     {\ensuremath{\skmath@log\left(#2\right)}}%  
61     {\ensuremath{\skmath@log_{#1}\left(#2\right)}}%  
62 }
```

`\exp`

```
63 \RenewDocumentCommand\exp{m}{\ensuremath{\mathchoice%  
64   {e^{#1}}%  
65   {\skmath@exp\left(#1\right)}}%  
66   {\skmath@exp\left(#1\right)}}%  
67   {\skmath@exp\left(#1\right)}}%  
68 }}
```

The fraction command is modified to improve typesetting.

`\frac`

```
69 \RenewDocumentCommand\frac{mm}{\genfrac{}{}{}{}{}%  
70   {\displaystyle #1}{\displaystyle #2}}
```

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

`\bar`

```
71 \RenewDocumentCommand\bar{m}{%  
72   \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 T_EX.S.E. Definition is deferred until after all packages are loaded to avoid collisions with other `\d` commands.

```
73 \AtBeginDocument{%
```

`\d`

```
74 \DeclareDocumentCommand\d{m}{\ensuremath{\,\,\mathrm{d}/  
    }#1%  
75                                     \@ifnextchar\d{\!}{}}  
76 }
```

Finally, we define a nicer way to denote vectors.

`\vec`

```
77 \let\vec\vectorsym
```