

The `skmath` package*

Simon Sigurdhsson
`ssimon@student.chalmers.se`

December 13, 2012

1 Introduction

This package provides improved and new math commands for superior typesetting with lower effort.

2 Usage

2.1 Options

As of version v0.1a, 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 Commands

2.2.1 Common sets

`\N` As stated above, these commands (`\N`, `\Z`, `\Q`, `\R` and `\C`) are only available if the
`\Z` `commonsets` option is given. They typeset the set of natural, integer, rational, real
`\Q` and complex numbers respectively:

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

2.2.2 New commands

The following commands are introduced to help improve typesetting.

`\norm` The commands `\norm` and `\abs`, quite expectedly, typeset the norm and abso-
`\abs` lute value of an expression, respectively. They have one mandatory argument (the

*This document corresponds to `skmath` v0.1a, dated 2012/08/30.

expression), and different norms can be achieved by appending a subscript after the argument of `\norm`:

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

`\d` There is also a command `\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

2.3.1 Trigonometric functions

`\sin` Some common trigonometric functions — `\sin`, `\cos`, `\tan`, `\cot`, `\arcsin`,
`\cos` `\arccos` and `\arctan` — have been redefined to typeset more easily. They now
`\tan` take one optional (does not apply to the `\arc*` macros) and one mandatory ar-
`\cot` gument; the mandatory argument is the argument to the trigonometric function
`\arcsin` and the optional argument is typeset in superscript between the function and its
`\arccos` argument, e.g. $\sin^2(\phi)$.
`\arctan`

2.3.2 Logarithmic and exponential functions

`\ln` The natural logarithm macro `\ln` has also been redefined to require an argument
`\log` which is typeset as the argument of the logarithm. The related macro `\log` is
redefined in a similar way, but also accepts an optional argument denoting the
base of the logarithm: $\log_2(x)$.

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:

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

2.3.3 Stylistic changes

`\frac` The `\frac` command has been changed to improve typesetting, and the `\vec`
`\vec` command is defined as a copy of the `\vectorsym` command of the `isomath` package,
as such: \mathbf{x} .

3 Implementation

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

```
1 \RequirePackage[log-declarations=false]{xparse}
2 \RequirePackage{kvoptions,amsmath,amssymb,mathtools,xfrac,isomath}
```

We begin by declaring an option.

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

We optionally provide commands to typeset common sets

```
6 \ifskmath@commonsets
```

`\N`

```
7 \newcommand\N{\ensuremath{\mathbb{N}}}
```

`\Z`

```
8 \newcommand\Z{\ensuremath{\mathbb{Z}}}
```

`\Q`

```
9 \newcommand\Q{\ensuremath{\mathbb{Q}}}
```

`\R`

```
10 \newcommand\R{\ensuremath{\mathbb{R}}}
```

`\C`

```
11 \newcommand\C{\ensuremath{\mathbb{C}}}
```

```
12 \fi
```

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

`\abs`

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

`\norm`

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

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

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

`\sin`

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

```

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

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

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

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

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

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

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

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

```

`\exp`

```
62 \renewcommand\exp[1]{\ensuremath{\mathchoice%
63   {e^{#1}}}%
64   {\skmath@exp\left(#1\right)}}%
65   {\skmath@exp\left(#1\right)}}%
66   {\skmath@exp\left(#1\right)}}%
67 }}
```

The fraction command is modified to improve typesetting.

`\frac`

```
68 \renewcommand{\frac}[2]{\genfrac{}{}{}{}%
69   {\displaystyle #1}{\displaystyle #2}}
```

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.

```
70 \AtBeginDocument{%
```

`\d`

```
71 \renewcommand\d[1]{\ensuremath{\,\,\mathrm{d}}#1%
72   \@ifnextchar\d{\!}{}}
73 }
```

Finally, we define a nicer way to denote vectors.

`\vec`

```
74 \let\vec\vectorsym
```