

# The `jkmath` package

Jonas Kaerts

April 13, 2018

## 1 The package options

The package contains a few options with regards to subsets.

**subsetorder** You are a person that likes your symbols for subsets to resemble the symbols used for ordering numbers. The command `\subset` now displays the symbol  $\subseteq$  while a new command `\stsubset` (for strict subsets) can be used for displaying the symbol  $\subset$ . Similar behavior occurs with `\supset` and `\stsupset`.

**subsetnonorder** You are a person that likes variety. Your symbols for subsets do not resemble the usual ordering symbols. the command `\subset` displays the symbol  $\subset$  while the symbol `\stsubset` displays as  $\subsetneq$ . Same for `\supset` and `\stsupset`.

**subsetnonamb** You like your notation as unambiguous as possible. The command `\subset` displays the symbol  $\subseteq$  while `\stsubset` displays  $\subsetneq$ . Again similar for `\supset` and `\stsupset`.

The advantage of this approach is that you can convert a document from one style of notation to another by simply changing the package option.

There are also two options, `bbsets` and `bfsets` concerning the display of number systems. They provide the following shorthands:

Command	Option <code>bbsets</code>	Option <code>bfsets</code>	Usage
<code>\N</code>	$\mathbb{N}$	$\mathbf{N}$	Natural numbers
<code>\Z</code>	$\mathbb{Z}$	$\mathbf{Z}$	Integers
<code>\Q</code>	$\mathbb{Q}$	$\mathbf{Q}$	Rational numbers
<code>\R</code>	$\mathbb{R}$	$\mathbf{R}$	Real numbers
<code>\C</code>	$\mathbb{C}$	$\mathbf{C}$	Complex numbers
<code>\F</code>	$\mathbb{F}$	$\mathbf{F}$	Fields
<code>\Aff</code>	$\mathbb{A}$	$\mathbf{A}$	Affine Space
<code>\PP</code>	$\mathbb{P}$	$\mathbf{P}$	Projective Space

## 2 Commands with arrays

### 2.1 Systems of equations

This package uses the `array`-package to define some useful math alignment. The first is the `system` environment. There are two new column types (`e` and `o`) to get the spacing around operators right. You can then call the code

```
\begin{system}{rorer}  
4x & + & 3y & 7\\  
2x & - & 5y & 10  
\end{system}
```

to get the result

$$\begin{cases} 4x + 3y = 7 \\ 2x - 5y = 10 \end{cases}.$$

This allows fine control over the alignment of a system of equations while still having the correct spacing. Note that the column type `e` automatically inserts an equality sign.

### 2.2 Augmented matrices

A second class of commands are the augmented matrices. The environment `augmentedmatrix` takes two arguments  $n$  and  $m$  and makes a matrix of  $n+m$  columns with a vertical rule after the  $n$ -th column, allowing the typesetting of systems with (multiple) right hand sides in matrix form. The code

```
\begin{augmentedmatrix}{2}{2}  
1 & 2 & 3 & 4\\  
5 & 6 & 7 & 8  
\end{augmentedmatrix}
```

has the following output:

$$\begin{array}{cc|cc} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{array}$$

At the moment there are two shorthand commands `apmpty` and `ipmpty` which take  $m = 1$  and  $m = n$  respectively and insert parentheses. These are used for solving systems with one right hand side and for calculating inverse matrices. The shorthand name is inspired by the shorthands in the `physics`-package. The code

```

\amqty{2}{1 & 2 & 3 \ 4 & 5 & 6}
\neq
\ipmqty{3}{0 & 1 & 0 & 1 & 0 & 0 \
-1 & 0 & 2 & 0 & 1 & 0 \
0 & 0 & 3 & 0 & 0 & 1}

```

produces the following output:

$$\left(\begin{array}{cc|c} 1 & 2 & 3 \\ 4 & 5 & 6 \end{array}\right) \neq \left(\begin{array}{ccc|ccc} 0 & 1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 3 & 0 & 0 & 1 \end{array}\right)$$

### 3 Intervals

I often use a script to check if my code is consistent in its use of delimiters since L<sup>A</sup>T<sub>E</sub>X allows you to have unmatched parentheses etc. in the text. The commands `\lbrace`, `\rbrace`, `\lbrack` and `\rbrack` are a godsend when I both want my script to give meaningful output and I only need one delimiter (such as in the `system` environment). This package depends on `mathtools` so the commands `\lparens` and `\rparens` for parentheses are also present.

Using these delimiter commands the package also defines four types of intervals: `\oointerval`, `\ccinterval`, `\ocinterval` and `\cointerval`. The `o` and `c` denote whether the left or right endpoint is open or closed.

```

\oointerval{1, 3} → (1,3)  interval open on both ends
\ocinterval{1, 3} → (1,3]  interval open on left side and
                           closed on right
\cointerval{1, 3} → [1,3)  interval closed on left side and
                           open on right
\ccinterval{1, 3} → [1,3]  interval closed on both sides

```

All intervals scale with their argument. You can define your own short-hands for these commands.

### 4 Sets

A general macro for denoting sets is `\set` which automatically places scalable braces around the argument. A scalable version of `\mid`, called `\where`, is also included. This makes sure the (readable) code

```

\set{x\in\mathbb{R}} \where \frac{3}{4}x + 5 = 0}

```

will give the following result:

$$\left\{ x \in \mathbb{R} \mid \frac{3}{4}x + 5 = 0 \right\}.$$

A second macro is `\restr` for denoting restrictions of functions to subsets of their domain. Simple usage is `\restr{f}_U` which displays  $f|_U$ .

## 5 Norms

If the user has not defined a macro `\norm` the package defines such a command. On this macro we build a few others.

<code>\norm{a}</code>	$\rightarrow \ a\ $	Norm of vector (scales with argument)
<code>\lpnorm{f}</code>	$\rightarrow \ f\ _2$	$L_p$ -norm of a function (subscript defaults to 2)
<code>\lpnorm[3]{f}</code>	$\rightarrow \ f\ _3$	Optional argument for other subscript
<code>\supnorm{f}</code>	$\rightarrow \ f\ _\infty$	Supremum norm

Changing the default subscript 2 can be done with the function `\setlpdefault{}` which takes as its argument the new default subscript.

## 6 Combinatorics

Using `\genfrac` from `amsmath` the package defines two commands for Stirling numbers.

<code>\stirlingfirstkind{n}{k}</code>	$\rightarrow \begin{bmatrix} n \\ k \end{bmatrix}$	Stirling number of first kind
<code>\stirlingsecondkind{n}{k}</code>	$\rightarrow \begin{Bmatrix} n \\ k \end{Bmatrix}$	Stirling number of second kind

Shorthands for these two commands have yet to be defined.

## 7 Number Theory

Two commands (with identical results) `\legendre` and `\jacobi` are defined (again using `\genfrac`) to typeset Legendre symbols and Jacobi symbols.

The output is identical but their name differs to make the code more readable.

$$\backslash\mathrm{legendre}\{a\}\{p_1\} \rightarrow \left(\frac{a}{p_1}\right) \quad \text{Legendre symbol}$$

$$\backslash\mathrm{jacobi}\{a\}\{n\} \rightarrow \left(\frac{a}{n}\right) \quad \text{Jacobi symbol}$$

## 8 Names of mathematicians

This section describes three simple commands `\mobius`, `\cech` and `\erdos` so you can mention Möbius, Čech and Erdős without any pain.