The `commath` LaTeXpackage v0.3

Wolfgang Putschögl

July 18, 2006

Contents

Abstract

The `commath`-package provides some commands which help you to format formulas flexibly.

1 Introduction

The purpose of the `commath`-package is to provide some commands which are supposed to produce better formatting and/or provide a more convenient way of input. For example, the `commath`-package provides commands for delimiters for which the size is determined automatically by default or can be controlled by an integer argument. Thus, the size of delimiters can be adapted to the formula by just changing one integer argument.

2 Acknowledgements

I want to thank Simon Dreher and Luca Trevisani for useful feedback and hints.

3 Requirements

The `commath`-package requires the `ifthen`-package and the `amsmath`-package to be installed. If they were not loaded before they are loaded by the `commath`-package.

4 Command overview

d

`\dif` is a command for the differential operator. It is simply an upface d to clarify that it is an operator. E.g.:

`\dif x`	$\mathrm{d}x$

D

`\Dif` is a command for a derivative operator (e.g. Malliavin derivative ...). It is simply an upface D to clarify that it is an operator. E.g.:

`\Dif X`	$\mathrm{D}X$

od

`\od[optional argument]{first argument}{second argument}` is a command for ordinary derivatives. The first argument denotes the function and the second argument denotes the variable with respect to which the derivative is taken. The optional argument denotes the order of differentiation. The style (text style/display style) is determined automatically.
E.g.:

`\od{f}{x}`	$\frac{\mathrm{d}f}{\mathrm{d}x}$
`\od[2]{f}{x}`	$\frac{\mathrm{d}^2f}{\mathrm{d}x^2}$

tnd

The `\tod`-command is essentially the same as the `\od`-command except that the style is set to text style.

dnd

The `\dod`-command is essentially the same as the `\od`-command except that the style is set to display style.

pd

`\pd[optional argument]{first argument}{second argument}` is a command for partial derivatives. The first argument denotes the function and the second argument denotes the variable with respect to which the derivative is taken. The optional argument denotes the order of differentiation. The style (text style/display style) is determined automatically.
E.g.:

`\pd{f}{x}`	$\frac{\partial f}{\partial x}$
`\pd[2]{f}{x}`	$\frac{\partial^2 f}{\partial x^2}$

tpd

The `\tpd`-command is essentially the same as the `\pd`-command except that the style is set to text style.

dpd

The `\dpd`-command is essentially the same as the `\pd`-command except that the style is set to display style.

md

`\md{first argument}...{sixth argument}` is a command for mixed partial derivatives. The first argument denotes the function and the second argument denotes the total order of differentiation. The third and the fifth argument are the variables with respect to which we want to differentiate and the fourth and the sixth argument are the corresponding orders of differentiation. The style (text style/display style) is determined automatically.
E.g.:

`\md{f}{5}{x}{2}{y}{3}`	$\frac{\partial^5 f}{\partial x^2 \partial y^3}$

tmd

The `\tmd`-command is essentially the same as the `\md`-command except that the style is set to text style.

`dmd`

The `\dmd`-command is essentially the same as the `\md`-command except that the style is set to display style.

`del`

`\del[optional argument]{first argument}` is a command for delimiters (normal parathesis). The value for the optional argument ranges from 0 to 4 with higher values resulting in larger delimiters The default value for the optional argument is -1 which results in automatic sizing for the parenthesis.
E.g.:

`\del{x}`	(x)
`\del[0]{x}`	(x)
`\del[1]{x}`	$\bigl(x\bigr)$
`\del[2]{x}`	$\Bigl(x\Bigr)$
`\del[3]{x}`	$\biggl(x\biggr)$
`\del[4]{x}`	$\Biggl(x\Biggr)$

`cbr`

`\cbr[optional argument]{first argument}` is a command for curly braces. The value for the optional argument ranges from 0 to 4 with higher values resulting in larger braces. The default value for the optional argument is -1 which results in automatic sizing for the braces.
E.g.:

`\cbr{x}`	$\{x\}$
`\cbr[0]{x}`	$\{x\}$
`\cbr[1]{x}`	$\bigl\{x\bigr\}$
`\cbr[2]{x}`	$\Bigl\{x\Bigr\}$
`\cbr[3]{x}`	$\biggl\{x\biggr\}$
`\cbr[4]{x}`	$\Biggl\{x\Biggr\}$

`set`

`\set` is the same as `\cbr`

`sbr`

`\sbr[optional argument]{first argument}` is a command for square brackets. The value for the optional argument ranges from 0 to 4 with higher values resulting in larger brackets. The default value for the optional argument is -1 which results in automatic sizing for the brackets.
E.g.:

`\sbr{x}`	$[x]$
`\sbr[0]{x}`	$[x]$
`\sbr[1]{x}`	$\bigl[x\bigr]$
`\sbr[2]{x}`	$\Bigl[x\Bigr]$
`\sbr[3]{x}`	$\biggl[x\biggr]$
`\sbr[4]{x}`	$\Biggl[x\Biggr]$

In the following we present commands for all kinds of intervals. The last two letters will imply whether the left and right boundary are open or closed respectively. The value for the optional argument ranges from 0 to 4 with higher values resulting in larger delimiters. The default value for the optional argument is -1 which results in automatic sizing for the delimiters.

`intoo`

`\intoo[optional argument]{first argument}` produces an interval where the left and right boundaries are open.
E.g.:

`\intoo{x}`	(x)
`\intoo[0]{x}`	(x)
`\intoo[1]{x}`	$\bigl(x\bigr)$
`\intoo[2]{x}`	$\Bigl(x\Bigr)$
`\intoo[3]{x}`	$\biggl(x\biggr)$
`\intoo[4]{x}`	$\Biggl(x\Biggr)$

`intcc`

`\intcc[optional argument]{first argument}` produces an interval where the left and right boundaries are closed.
E.g.:

`\intcc{x}`	$[x]$
`\intcc[0]{x}`	$[x]$
`\intcc[1]{x}`	$\bigl[x\bigr]$
`\intcc[2]{x}`	$\Bigl[x\Bigr]$
`\intcc[3]{x}`	$\biggl[x\biggr]$
`\intcc[4]{x}`	$\Biggl[x\Biggr]$

`intoc`

`\intoc[optional argument]{first argument}` produces an interval where the left boundary is open and right boundary is closed.
E.g.:

`\intoc{x}`	$(x]$
`\intoc[0]{x}`	$(x]$
`\intoc[1]{x}`	$\bigl(x\bigr]$
`\intoc[2]{x}`	$\Bigl(x\Bigr]$
`\intoc[3]{x}`	$\biggl(x\biggr]$
`\intoc[4]{x}`	$\Biggl(x\Biggr]$

intco

`\intco[optional argument]{first argument}` produces an interval where the left boundary is closed and right boundary is open.
E.g.:

`\intco{x}`	$[x)$
`\intco[0]{x}`	$[x)$
`\intco[1]{x}`	$\bigl[x\bigr)$
`\intco[2]{x}`	$\Bigl[x\Bigr)$
`\intco[3]{x}`	$\biggl[x\biggr)$
`\intco[4]{x}`	$\Biggl[x\Biggr)$

eval

`\eval[optional argument]{first argument}` is a command for the notation of an expression denoted by the first argument evaluated at a particular condition. The value for the optional argument ranges from 0 to 4 with higher values resulting in larger delimiters. The default value for the optional argument is -1 which results in automatic sizing for the delimiters.
E.g.:

`\eval{f(\epsilon)}_{\epsilon=0}`	$\left. f(\epsilon) \right\rvert_{\epsilon=0}$
`\eval[0]{f(\epsilon)}_{\epsilon=0}`	$f(\epsilon)\rvert_{\epsilon=0}$
`\eval[1]{f(\epsilon)}_{\epsilon=0}`	$f(\epsilon)\bigr\rvert_{\epsilon=0}$
`\eval[2]{f(\epsilon)}_{\epsilon=0}`	$f(\epsilon)\Bigr\rvert_{\epsilon=0}$
`\eval[3]{f(\epsilon)}_{\epsilon=0}`	$f(\epsilon)\biggr\rvert_{\epsilon=0}$
`\eval[4]{f(\epsilon)}_{\epsilon=0}`	$f(\epsilon)\Biggr\rvert_{\epsilon=0}$

sVert

`\sVert[optional argument]` is essentially the same as `eval`. Use `\sVert` for readability if you don't need automatic sizing of the vert bar.
E.g.:

`f(\epsilon)\sVert[0]_{\epsilon=0}`	$f(\epsilon)\vert_{\epsilon=0}$
`f(\epsilon)\sVert[1]_{\epsilon=0}`	$f(\epsilon)\big\vert_{\epsilon=0}$
`f(\epsilon)\sVert[2]_{\epsilon=0}`	$f(\epsilon)\Big\vert_{\epsilon=0}$
`f(\epsilon)\sVert[3]_{\epsilon=0}`	$f(\epsilon)\bigg\vert_{\epsilon=0}$
`f(\epsilon)\sVert[4]_{\epsilon=0}`	$f(\epsilon)\Bigg\vert_{\epsilon=0}$

There is also one command included in the package which fixes the alignment of := in mathmode.

`envert`

`\envert}[optional argument]{first argument}` is a command is a command which enclose the argument in vert-bar delimiters.
E.g.:

`\envert{x}`	$\lvert x\rvert$
`\envert[0]{x}`	$\lvert x\rvert$
`\envert[1]{x}`	$\bigl\lvert x\bigr\rvert$
`\envert[2]{x}`	$\Bigl\lvert x\Bigr\rvert$
`\envert[3]{x}`	$\biggl\lvert x\biggr\rvert$
`\envert[4]{x}`	$\Biggl\lvert x\Biggr\rvert$

`abs`

The `\abs`-command is the same as the `\envert`-command.

`enVert`

`\enVert}[optional argument]{first argument}` is a command is a command which enclose the argument in double vert-bar delimiters.
E.g.:

`\enVert{x}`	$\|x\|$
`\enVert[0]{x}`	$\|x\|$
`\enVert[1]{x}`	$\big\|x\big\|$
`\enVert[2]{x}`	$\Big\|x\Big\|$
`\enVert[3]{x}`	$\bigg\|x\bigg\|$
`\enVert[4]{x}`	$\Bigg\|x\Bigg\|$

`norm`

The `\norm`-command is the same as the `\enVert`-command.

`fullfunction`

`\fullfunction{first argument}....{fifth argument}` is a command which nicely formats a function. The first argument denotes the function name, the second the domain and the third the image of the function. The forth is the parameter which is mapped to the expression denoted by argument five.
E.g.:
`\fullfunction{f}{\mathbb R}{\mathbb R}{x}{\sqrt{x}}` results in

$$\begin{array}{ccccc} f & : & \mathbb{R} & \longrightarrow & \mathbb{R} \\ & & x & \longmapsto & \sqrt{x} \end{array}$$

`thmref`

`\thmref{first argument}` is a command to reference theorems where the first argument is the label of the corresponding theorem. The result of `\thmref{label}` will be Theorem (number of the label) and analogously for the subsequent commands. The number of the theorem is not separated from the word Theorem by e.g. a linebreak.

`exref`

`\exref{first argument}` is a command to reference examples where the first argument is the label of the corresponding example.

`defnref`

`\defnref{first argument}` is a command to reference definitions where the first argument is the label of the corresponding definition.

`secref`

`\secref{first argument}` is a command to reference sections where the first argument is the label of the corresponding section.

`lemref`

`\lemref{first argument}` is a command to reference lemmas where the first argument is the label of the corresponding lemma.

`propref`

`\propref{first argument}` is a command to reference propositions where the first argument is the label of the corresponding proposition.

`remref`

`\remref{first argument}` is a command to reference remarks where the first argument is the label of the corresponding remark.

`figref`

`\figref{first argument}` is a command to reference figures where the first argument is the label of the corresponding figure.

`colref`

`\colref{first argument}` is a command to reference corollaries where the first argument is the label of the corresponding corollary.

`appref`

`\appref{first argument}` is a command to reference the Appendix where the first argument is the label of the corresponding Appendix.

`assref`

`\assref{first argument}` is a command to reference assumptions where the first argument is the label of the corresponding Assumption.

Index