

The **brodp** package*

Benjamin Tatlock†

November 8, 2012

Abstract

The **brodp** package simplifies the process of writing differential operators and brackets in \LaTeX . The commands facilitate the easy manipulation of equations involving brackets and allow partial differentials to be expressed in alternate forms.

1 Introduction

The **brodp** package introduces a bracket command that automatically selects parentheses, square brackets or braces as appropriate based on the number of brackets. As such, brackets can be freely added and removed without alterations to the rest of an equation. Commands for typing ordinary and partial differentials are also created, with two optional concise forms for partial derivatives. Equations can therefore be written in a standard form and their style altered globally at a later date.

2 Usage

This section outlines the use of the three commands provided by the **brodp** package.

2.1 Brackets

The bracket command replaces `\left(` and `\right)` with the single command, `\br` to produce brackets in a math environment. The input `\br{x}` produces

$$(x).$$

When nested, the type of bracket is determined automatically, for example, `r=\br{\br{x-a}^2+\br{y-b}^2}^{\frac{1}{2}}` results in

$$r = \left[(x - a)^2 + (y - b)^2 \right]^{\frac{1}{2}}.$$

*This documentation corresponds to **brodp** v1.1, dated 2012/11/08.

†Benjamin.Tatlock@mail.com

2.2 Differential operators

2.2.1 Ordinary differential

The `\od` command produces ordinary differentials, such as `\od{y}{x}` which is displayed as

$$\frac{dy}{dx}.$$

The order of the differential can be specified as an optional argument, illustrated here with the command `\od[2]{u}{x}=-\omega^2u` which results in

$$\frac{d^2u}{dx^2} = -\omega^2u.$$

When using `\od` with a sans-serif font, the package should be loaded with the optional argument `sans`.

2.2.2 Partial differential

The command `\pd` follows the same form as `\od`, but instead produces partial differentials. By default, `\pd{u}{t}=6u\pd{u}{x}-\pd[3]{u}{x}` produces

$$\frac{\partial u}{\partial t} = 6u \frac{\partial u}{\partial x} - \frac{\partial^3 u}{\partial x^3}.$$

Loading the package with the option `pdshort` results in an alternative format for partial differentials:

$$u_t = 6uu_x - u_{xxx}$$

If the optional argument is not a positive integer, the default form is reverted to, as with `\pd[n]xy`:

$$\frac{\partial^n x}{\partial y^n}$$

A second alternative form is produced when the option `pdalt` is specified:

$$\partial_t u = 6u \partial_x u - \partial_{xxx} u$$

The syntax for partial differentials is extended to allow higher order terms such as

$$\frac{\partial^3 u}{\partial x^2 \partial t}$$

to be easily typeset with `\pd{u}{x,x,t}`. When applying a partial derivative to a longer term, the first argument can be left blank and the longer string written as an additional argument, as demonstrated by `\pd{}{z}{x+y}` which results in

$$\frac{\partial}{\partial z} (x + y).$$

To prevent brackets from being placed after a partial differential with an empty numerator, use `!` for the first argument, for example, `\pd!{}{x}`:

$$\frac{\partial}{\partial x}$$

2.3 Further examples

Combining the commands provided by this package allows more complex terms to be written concisely. The `\br` allows equations with multiple brackets to be constructed and manipulated quickly and easily. An example of this is `\br{x-y}\br{Q-\frac{1}{2}\br{P+\br{x-1}\br{\br{a+b}\br{u+c}-\od st}}}`, which results in

$$(x-y) \left(Q - \frac{1}{2} \left\{ P + (x-1) \left[(A+B) (u+c) - \frac{ds}{dt} \right] \right\} \right).$$

The form of the partial differentials written with the `\pd` command can be changed at any time. `\pd Et+\sum_{i=1}^3\pd{\br{x_i}}{\br{E+p}u_i}` produces

$$\frac{\partial E}{\partial t} + \sum_{i=1}^3 \frac{\partial}{\partial x_i} [(E+p) u_i]$$

by default. With `pdshort`, the same input instead creates

$$E_t + \sum_{i=1}^3 [(E+p) u_i]_{x_i},$$

while with `pdalt` the output is

$$\partial_t E + \sum_{i=1}^3 \partial_{x_i} [(E+p) u_i].$$

3 Implementation

Announce the name, package version and L^AT_EX 2_ε requirement

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{bropd}
3 [2012/11/08 1.1 automated typing of brackets and differential operators]
```

Initiate conditionals and counters

```
4 \newif\if@bropd@a\newif\if@bropd@b\newif\if@bropd@s\newif\if@bropd@u
5 \newcounter{@bropd@c}\newcounter{@bropd@d}
6 \newcounter{@bropd@e}\newcounter{@bropd@f}
```

Provide optional arguments for alternative form partial differential and sans-serif fonts

```
7 \DeclareOption{pdshort}{\@bropd@atruel\@bropd@ufalse}
8 \DeclareOption{pdalt}{\@bropd@atruel\@bropd@utruel}
9 \DeclareOption{sans}{\@bropd@struel\ProcessOptions}
```

Define character for \pd with empty numerator and no brackets following

```
10 \def\@bropd@t{!}
```

\@bropd@g Internal command for default partial differential form

```
11 \def\@bropd@g#1#2#3#4{\def\@bropd@h{#1}
```

Create numerator

```
12 \ifx\@bropd@o\@bropd@t\let\@bropd@o\empty\fi
```

Reset counters

```
13 \ifx\@bropd@h\empty
14 \setcounter{@bropd@c}0\setcounter{@bropd@d}1
15 \let\@bropd@h\empty
```

Determine order of partial derivative and display as power of partial if greater than one

```
16 \@for\@bropd@i:=#3\do{\stepcounter{@bropd@c}}
17 \frac{#4\ifnum\value{@bropd@c}>1
18 ^{\arabic{@bropd@c}}\fi\@bropd@o}
```

Cycle through each term for the denominator. If a term is the same as the previous, add to counter, otherwise print along with counter value

```
19 {\@for\@bropd@i:=#3\do{
20 \ifx\@bropd@i\@bropd@h
21 \stepcounter{@bropd@d}
22 \else
23 \@bropd@h
24 \ifnum\value{@bropd@d}>1
25 ^{\arabic{@bropd@d}}
26 \fi
27 #4\setcounter{@bropd@d}1
28 \fi
29 \let\@bropd@h\@bropd@i}
30 \@bropd@h
```

```

31 \ifnum\value{@bropd@d}>1
32 ~{\arabic{@bropd@d}}\fi}
33 \else
If optional argument provided, display partial differential to stated order
34 \frac{#4^{#1}\@bropd@o}{#4\noexpand#3^{#1}}
35 \fi}

\@bropd@j Internal command for brackets. Evaluate terms within brackets to determine the
highest bracket level
36 \def\@bropd@j#1{\setcounter{@bropd@f}0
37 {\renewcommand{\br}[1]{\stepcounter{@bropd@e}{##1}
38 \ifnum\value{@bropd@e}>\value{@bropd@f}
39 \setcounter{@bropd@f}{\value{@bropd@e}}
40 \fi
41 \addtocounter{@bropd@e}{-1}}\sbox0{${#1$}}
42 \loop
43 \ifnum\value{@bropd@f}>2
44 \addtocounter{@bropd@f}{-3}
45 \repeat

Determine bracket type for current level
46 \ifnum\value{@bropd@f}>0
47 \ifnum\value{@bropd@f}>1
48 \def\@bropd@k{\left\lbrace#1\right\rbrace}
49 \else
50 \def\@bropd@k{\left[#1\right]}
51 \fi
52 \else
53 \def\@bropd@k{\left(#1\right)}
54 \fi}
55 \if@bropd@a
56 \let\@bropd@l\empty

\br External command for brackets
57 \newcommand\br[1]{\let\@bropd@m\@bropd@l
58 \@bropd@j{\let\@bropd@l\empty#1\let\@bropd@l\@bropd@m}
59 \ifx\@bropd@l\empty
60 \def\@bropd@n{\@bropd@k}
61 \else
62 \let\@bropd@m\@bropd@l
63 \def\@bropd@n{\@bropd@k_{\@bropd@m}}
64 \fi
65 \let\@bropd@l\empty
66 \@bropd@n}
67 \if@bropd@u

\@bropd@q Internal command for first alternative partial differential command
68 \def\@bropd@q#1#2#3{\def\@bropd@h{#1}
69 \ifx\@bropd@h\empty

```

```

70 \@bropd@bfalse
71 \else
Default partial differential if character provided as optional argument
72 \if!\ifnum9<1#1!\else_\fi
73 \@bropd@bfalse
74 \else
75 \@bropd@btrue
76 \fi
77 \fi
78 \ifx\@bropd@o\@bropd@t
79 \@bropd@btrue\let\@bropd@o\empty
80 \fi
81 \if@bropd@b\@bropd@g{#1}\@bropd@o{\noexpand#3}\partial
82 \else
Construct alternate form partial differential
83 \partial_{\ifx\@bropd@h\empty
84 \@for\@bropd@i:=#3\do\@bropd@i\else\setcounter{@bropd@c}0
85 \loop
86 \ifnum\value{@bropd@c}<#1
87 #3\stepcounter{@bropd@c}
88 \repeat
89 \fi}
90 \@bropd@o
91 \fi}
92 \else
Internal command for second alternative partial differential command
93 \def\@bropd@q#1#2#3{\def\@bropd@h{#1}
94 \ifx\@bropd@h\empty
95 \@bropd@bfalse
96 \else
Default partial differential if character provided as optional argument
97 \if!\ifnum9<1#1!\else_\fi
98 \@bropd@bfalse
99 \else
100 \@bropd@btrue
101 \fi
102 \fi
103 \ifx\@bropd@o\@bropd@t
104 \@bropd@btrue\let\@bropd@o\empty
105 \fi
106 \if@bropd@b\@bropd@g{#1}\@bropd@o{\noexpand#3}\partial
107 \else
Construct alternate form partial differential
108 \def\@bropd@p{\ifx\@bropd@h\empty
109 \@for\@bropd@i:=#3\do\@bropd@i\else\setcounter{@bropd@c}0
110 \loop
111 \ifnum\value{@bropd@c}<#1

```

```

112 #3\stepcounter{@brodp@c}
113 \repeat
114 \fi}
115 \ifx\@brodp@o\empty
116 \let\@brodp@l\@brodp@p\else{\@brodp@o}_{\@brodp@p}}\fi\fi}
117 \fi
118 \else
    Internal command for default partial differentials
119 \def\@brodp@q#1#2#3{\@brodp@g{#1}{#2}{\noexpand#3}\partial}

\br  External command for brackets
120 \newcommand\br[1]{\@brodp@j{#1}\@brodp@k}
121 \fi

\pd
122 \newcommand\pd[3][]{\def\@brodp@o{#2}
123 \ifx\@brodp@o\empty
124 \let\@brodp@v\br
125 \else
126 \let\@brodp@v\empty
127 \fi
128 \@brodp@q{#1}{#2}{#3}\@brodp@v}

\@brodp@r  Format ordinary differential based on font type
129 \if@brodp@s
130 \def\@brodp@r{\mathsf d}
131 \else
132 \def\@brodp@r{\mathrm d}
133 \fi

\od  External command for ordinary differentials
134 \newcommand\od[3][]{\def\@brodp@h{#1}
135 \frac{\@brodp@r\ifx\@brodp@h\empty\else^{#1}\fi#2}
136 {\@brodp@r#3\ifx\@brodp@h\empty\else^{#1}\fi}}
137 \endinput

```

Change History

v1.0		
General: Initial version	4
v1.1		
\@bropd@g: Fixed higher order dif-		
ferentials with denominators		
containing commands	5
\@bropd@q: Added second alterna-		
	tive form for partial differen-	
	tials 5
	Allowed partial differential with	
	empty numerator without need	
	for bracket command after 6
	\pd: Brackets automatically added	
	after empty numerator 7