

The **bro**pd package*

Benjamin Tatlock

November 4, 2012

Abstract

The **bro**pd 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 an alternate form.

1 Introduction

The **bro**pd 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 an equation. A command for writing ordinary and partial differentials is also created with an optional concise form for typesetting partial derivatives. This allows equations to be typed 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 **bro**pd 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 **bro**pd v1.0, dated 2012/11/04.

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:

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

Note that when using this form, the `\br` function *must* follow `\pd` when the first argument is left empty, as demonstrated by `\pd{}{z}\br{x+y}` which results in

$$(x+y)_z.$$

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

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

The syntax for partial differentials is extended to allow the second order term such as

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

to be easily typeset with `\pd{u}{x,t}`.

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 $(x-y)\{Q-\frac{1}{2}\{P+(x-1)\{A+B\}(u+c)-t\}\}$, which results in

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

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

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

or, with `pdshort`,

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

3 Implementation

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

```
1 \NeedsTeXFormat{LaTeX2e}
2 \ProvidesPackage{bropd}
3 [2012/11/04 1.0 automated typing of brackets and differential operators]
```

Initiate conditionals and counters

```
4 \newif\if@bropd@a\newif\if@bropd@b\newif\if@bropd@s
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@true}
8 \DeclareOption{sans}{\@bropd@strue}\ProcessOptions
```

Internal command for default partial differential form

```
9 \newcommand\@bropd@g[4]{\def\@bropd@h{#1}
10 \ifx\@bropd@h\empty
11 \setcounter{@bropd@c}0\setcounter{@bropd@d}1
12 \let\@bropd@h\empty
```

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

```
13 \@for\@bropd@i:=#3\do{\stepcounter{@bropd@c}}
14 \frac{#4\ifnum\value{@bropd@c}>1
15 ^{\arabic{@bropd@c}}\fi#2}
```

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

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

If optional argument provided, display partial differential to stated order

```
31 \frac{#4^{#1}#2}{#4#3^{#1}}
32 \fi}
```

Internal command for brackets. Evaluate terms within brackets to determine the highest bracket level

```

33 \newcommand\@brodp@j[1]{\setcounter{@brodp@f}0
34 {\renewcommand{\br}[1]{\stepcounter{@brodp@e}{##1}
35 \ifnum\value{@brodp@e}>\value{@brodp@f}
36 \setcounter{@brodp@f}{\value{@brodp@e}}
37 \fi
38 \addtocounter{@brodp@e}{-1}}\sbox0{${#1$}}
39 \loop
40 \ifnum\value{@brodp@f}>2
41 \addtocounter{@brodp@f}{-3}
42 \repeat

```

Determine bracket type for current level

```

43 \ifnum\value{@brodp@f}>0
44 \ifnum\value{@brodp@f}>1
45 \def\@brodp@k{\left\lbrace#1\right\rbrace}
46 \else
47 \def\@brodp@k{\left[#1\right]}
48 \fi
49 \else
50 \def\@brodp@k{\left(#1\right)}
51 \fi}
52 \if@brodp@a
53 \let\@brodp@l\empty

```

External bracket command

```

54 \newcommand\br[1]{\let\@brodp@m\@brodp@l
55 \@brodp@j{\let\@brodp@l\empty#1\let\@brodp@l\@brodp@m}
56 \ifx\@brodp@l\empty
57 \def\@brodp@n{\@brodp@k}
58 \else
59 \let\@brodp@m\@brodp@l
60 \def\@brodp@n{\@brodp@k_{\@brodp@m}}
61 \fi
62 \let\@brodp@l\empty
63 \@brodp@n}

```

External command for alternative partial differential command

```

64 \newcommand\pd[3][\def\@brodp@h{#1}
65 \ifx\@brodp@h\empty
66 \@brodp@bfalse
67 \else

```

Default partial differential if character provided as optional argument

```

68 \if!\ifnum9<1#1!\else_\fi
69 \@brodp@bfalse
70 \else
71 \@brodp@btrue
72 \fi
73 \fi

```

```

74 \if@bropd@b\@bropd@g{#1}{#2}{\noexpand#3}\partial
75 \else
Construct alternate form partial differential
76 \def\@bropd@o{#2}
77 \def\@bropd@p{\ifx\@bropd@h\empty
78 \@for\@bropd@i:=#3\do\@bropd@i\else\setcounter{@bropd@c}0
79 \loop
80 \ifnum\value{@bropd@c}<#1
81 #3\stepcounter{@bropd@c}
82 \repeat
83 \fi}
84 \ifx\@bropd@o\empty
85 \let\@bropd@l\@bropd@p\else{#2}{_{\@bropd@p}}\fi\fi}
86 \else
External command for default partial differentials
87 \newcommand\pd[3][]{\@bropd@g{#1}{#2}{\noexpand#3}\partial}
External command for brackets
88 \newcommand\br[1]{\@bropd@j{#1}\@bropd@k}
89 \fi
Format ordinary differential based on font type
90 \if@bropd@s
91 \newcommand\@bropd@r{\mathsf{d}}
92 \else
93 \newcommand\@bropd@r{\mathrm{d}}
94 \fi
External command for ordinary differentials
95 \newcommand\od[3][]{\def\@bropd@h{#1}
96 \frac{\@bropd@r\ifx\@bropd@h\empty\else^{#1}\fi#2}
97 {\@bropd@r#3\ifx\@bropd@h\empty\else^{#1}\fi}}
98 \endinput

```