The odesandpdes package

Anakin
anakin@ruc.dk

January 20, 2024

Abstract

This package is the solution no one asked for, to a problem nobody had. Have you ever thought to yourself "wow, I sure do dislike having to remember *multiple* macros for my odes and pdes" and the author of this package has to agree, wholeheartedly. In the modern world of "tik-toking" and "family guy surfing", our brains have rotted beyond salvage for even basic levels of cognitive recall. This package aims to fix this, through two macros that have been set to each have an identical form and function, with an emphasis on intuitive use. Through setting options, the multiple common notational style are easily swapped between, all by a single option. *You're welcome*.

Contents

My funny little ODE/PDE package

Start by first having `odesandpdes.sty` downloaded in an accessible directory, or in the same directory as your overleaf main.tex, using it by inserting;

`\usepackage[⟨options⟩]{odesandpdes}`

into the preamble, Ideally after any font changing packages you use.

1 Usage

If the reader does not wish to be gradually introduced to the package and its features, feel free to skip directly to section 2.

1.1 Options

`notation` `maxprimes` The options included are based off of the three most common notations (according to Wikipedia), Lagrange, Leibniz, and Newton. They can be accessed through the `[`⟨*options*⟩`]` when importing the package;

`\usepackage[notation=⟨option⟩]{odesandpdes}`

In the case of Lagrange or Newton notation, there is the `maxprimes` option for determination of how many physical markings are allowed to be made before the notation switches to a symbolic version;

`\usepackage[maxprimes=⟨integer⟩]{odesandpdes}`

`\setDE` However, if one might wish to change it on a section to section basis, the command `\setDE{`⟨*options*⟩`}` is able to take any package option as an argument and will apply the new option going forward.

Option list	Default Value	Valid Arguments
notation	Leibniz	default, Lagrange, Leibniz, Newton
maxprimes	3	maxprimes = $n, n \in \mathbb{N}_+$

1.2 The Meat and Potatoes

The command(s) are approached with the philosophy of of an intuitive and modular usage. The full extent of its usage can look like;

`\ode*[x]^2 X(x) =\ode T_{\eta} at 0; -\alpha` $\Rightarrow \dfrac{\mathrm{d}^2}{\mathrm{d}x^2}X(x) = \left.\dfrac{\mathrm{d}T_\eta}{\mathrm{d}t}\right|_{t=0} - \alpha$

very quickly, and very easily building complex interactions of differentials. The quick functional break down of each element that comprises the macro;

`\ode`⟨*star*⟩`[`⟨*variable*⟩`]^`⟨*degree*⟩ `{`⟨*function*⟩`}`at␣⟨*position*⟩;

Argument	Usage
`[`⟨*variable*⟩`]`	The variable being derived
⟨*degree*⟩	The order/degree of the derivative
`{`⟨*function*⟩`}`	The function being derived
␣at␣⟨*point*⟩;	Where the function is being derived

All arguments are conditionally optional, only the function is mandatory, but the command can forgo needing a function if a star is placed.

Notation Style

`\LagrODE` `\LeibODE` `\NewtODE` `\LagrPDE` `\LeibPDE` `\NewtPDE`

There are 3 distinct notational styles one can choose between. This choice can be made as a package option in the preamble, in the text with `\setDE{`⟨*options*⟩`}`, or if one only needs to use a notation style once, through its respective macro.

In essense, all the `\ode` or `\pde` commands do are call the respective notational variant aligned with the currently set option. This makes it simple enough to just use one of the notational varients, should one wish to do so:

`\LagrODE[x] c = \LeibODE[x] c = \NewtODE[x] c` $\Rightarrow \quad c' = \frac{\mathrm{d}c}{\mathrm{d}x} = \dot{c}$

`\LagrPDE[x] c = \LeibPDE[x] c = \NewtPDE[x] c` $\Rightarrow \quad c'_x = \frac{\partial c}{\partial x} = \dot{c}$

This also means that all these functions are identical in what arguments they take.

Variable and Function Arguments

`\ode` `\ode*`

The most barebone form can be understood as:

`\ode[`⟨*variable*⟩`]{`⟨*function*⟩`}`
`\ode*[`⟨*variable*⟩`]`

`\pde` `\pde*`

and for the sake of parity, the PDE usage is identical:

`\pde[`⟨*variable*⟩`]{`⟨*function*⟩`}`
`\pde*[`⟨*variable*⟩`]`

Any value you give to the *optional* `[`⟨*variable*⟩`]` argument will be represented as the variable being derived. While the *mandatory* `{`⟨*function*⟩`}` argument will be the function you are deriving. Say you wish to indicate you are deriving $X(t)$, simple as writing `\ode[t]{X}`, however, its worth noting that t is the default variable so writing `\ode{X}` will produce identical results. Hence `\ode[t]{X} = \ode{X}` will produce;

`\ode[t]{X} =\ode{X}` $\Longrightarrow \frac{\mathrm{d}X}{\mathrm{d}t} = \frac{\mathrm{d}X}{\mathrm{d}t}$

While the `{`⟨*function*⟩`}` argument is mandatory using the non-starred command, using the starred varient omits the need for the `{`⟨*function*⟩`}` argument. Therefor, writing the exact same equation, just starred `\ode*[t]{X} = \ode*{X}` will instead produce;

`\ode*[t]{X} =\ode*{X}` $\Longrightarrow \frac{\mathrm{d}}{\mathrm{d}t}X = \frac{\mathrm{d}}{\mathrm{d}t}X$

Effectively one can rewrite the 'bare-bones' display as:

`\ode`⟨*star*⟩`[`⟨*variable*⟩`]{`⟨*function*⟩`}`

Degree of Derivative

The previously shown stated section is something the reader has likely encountered before, made themselves. This is where this package begins to differentiate[1] itself. Consider:

[1] Calculus Pun!

`\ode`⟨*star*⟩`[`⟨*variable*⟩`]`↑⟨*degree*⟩`{`⟨*function*⟩`}`

A feature of this family of commands, is that it can '*easily*' recognize a following exponent should one be placed. There was rational in choosing to check for the exponent immediately after the macro command opposed to checking for the exponent at the end after the function. As, often you would want add a higher degree very quickly as opposed to *after* defining the function.

`\ode^2{f(x)}` as opposed to `\ode{f(x)}^2`

This was one of the main motivations of creating a package to begin with as instead of needing, maybe, two personalized commands, such as "`\ddt{f}` and `\ddxx{f}`", or "`\dd{x}{f}` and `\dd[2]{x}{f}`". One simply needs to treat the `\ode` macro itself as being raised to a higher degree.

$$\texttt{\textbackslash ode* \textbackslash left(\textbackslash ode\{f\} \textbackslash right)=\textbackslash ode\^{}2\{f\}} \Rightarrow \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathrm{d}f}{\mathrm{d}t}\right) = \frac{\mathrm{d}^2 f}{\mathrm{d}t^2}$$

Defining Where the Derivative is

Imagine you, as the reader, are trying to quickly and easily write up the boundry conditions of your problem. One could always make another macro, in what is no doubt an impressive display of differential shortcuts. *Or*:

`\ode`⟨*star*⟩`[`⟨*variable*⟩`]`↑⟨*degree*⟩`{`⟨ *function*⟩`}`␣at␣⟨*postion*⟩;

See, TEX does something very interesting when it uses '*glue*', which is partially replicated by packages such as `TikZ`, where it will happily take 'soft' modifiers written directly in plain english. If one wishes to strictly define paragraph spacing in TEX, they would use '`\parskip=1ex`'. If one would rather give it a range of tolerance the following construct '`\parskip=1ex plus 0.5ex minus 0.5ex`' then allows a spacing of 1 ± 0.5 `ex`.

Glue is of course something special, but that does not mean that the author can not gain inspiration. Say one wishes to define Neumann boundries;

$$\texttt{\textbackslash ode[x]\{c\} at 0;=0\textbackslash land\textbackslash ode[x]\{c\} at L;=1} \Rightarrow \left.\frac{\mathrm{d}c}{\mathrm{d}x}\right|_{x=0} = 0 \wedge \left.\frac{\mathrm{d}c}{\mathrm{d}x}\right|_{x=L} = 1$$

$$\texttt{\textbackslash ode[x]\{c\} at 0 = L;=1} \Rightarrow \left.\frac{\mathrm{d}c}{\mathrm{d}x}\right|_{x=0=L} = 1$$

Literally could not be easier.[2]

Those reading til this point may have recalled that the first example did not contain many braces. This is because with the "proper" spacing, there is little need for the use of the braces, so as to help promote a more fluid, (and readable), workflow without always needing to worry about the f***ing brace. Not that one can not use the brace for personal taste. In the following section, many examples of use will be illustrated to show the range and versitility of the functions.

The most important thing to always remember. *Just because* the author of this package has done as much as they can to '~~*idiot*~~ *user proof*' its functions does not mean the user does not still need to be cautious. This is LATEX we are talking about. There are likely many scenarios that the author did not think of, nor accidentally came across.

[2]My source is that I made it up

2 Examples of use

To show the generality of use. The following examples all take identical form in the TEX/LATEX itself. Additionally, in order to illustrate the functional boundries of the command with respect to each of the notational styles. There is a variety of spacing and bracketing to help highlight these features, and will be shown in the following `verbatim` enviroment;

```
\begin{align*}
\ode A(x)      && \ode[x]{B(x)} && \ode^1 C(x)     && \ode[x]^5 {D(x)} \\
\ode* {E(x)}   && \ode*[x] F(x) && \ode*^2 {G(x)}  && \ode*[x]^6H(x)   \\
\pde[t] I(x)   && \pde[x] {J(x)}&& \pde[t]^3K(x)   && \pde[x]^7 {L(x)} \\
\pde*[t] {M(x)}&& \pde*[x]N(x)  && \pde*[t]^4 O(x) && \pde*[x]^8 P(x)
\end{align*}
```

`\setDE{notation=Lagrange}` *and/or* `\usepackage[notation=Lagrange]{odesandpdes}`

$$\begin{aligned} A'(x) && B(x)' && C'(x) && D(x)^{(5)} \\ f'(t)E(x) && f'(x)F(x) && f''(t)G(x) && f^{(6)}(x)H(x) \\ I_t'(x) && J(x)_x' && K_t'''(x) && L(x)_x^{(7)} \\ f_t'(t)M(x) && f_x'(x)N(x) && f_t^{(4)}(t)O(x) && f_x^{(8)}(x)P(x) \end{aligned}$$

`\setDE{notation=Leibniz}` *and/or* `\usepackage[notation=Leibniz]{odesandpdes}`

$$\begin{aligned} \frac{\mathrm{d}A(x)}{\mathrm{d}t} && \frac{\mathrm{d}B(x)}{\mathrm{d}x} && \frac{\mathrm{d}C(x)}{\mathrm{d}t} && \frac{\mathrm{d}^5D(x)}{\mathrm{d}x^5} \\ \frac{\mathrm{d}}{\mathrm{d}t}E(x) && \frac{\mathrm{d}}{\mathrm{d}x}F(x) && \frac{\mathrm{d}^2}{\mathrm{d}t^2}G(x) && \frac{\mathrm{d}^6}{\mathrm{d}x^6}H(x) \\ \frac{\partial I(x)}{\partial t} && \frac{\partial J(x)}{\partial x} && \frac{\partial^3 K(x)}{\partial t^3} && \frac{\partial^7 L(x)}{\partial x^7} \\ \frac{\partial}{\partial t}M(x) && \frac{\partial}{\partial x}N(x) && \frac{\partial^4}{\partial t^4}O(x) && \frac{\partial^8}{\partial x^8}P(x) \end{aligned}$$

`\setDE{notation=Newton}` *and/or* `\usepackage[notation=Newton]{odesandpdes}`

$$\begin{aligned} \dot{A}(x) && \dot{B(x)} && \dot{C}(x) && \overset{5}{\dot{D(x)}} \\ \dot{t}E(x) && \dot{x}F(x) && \ddot{t}G(x) && \overset{6}{\dot{x}}H(x) \\ \dot{I}(x) && \dot{J(x)} && \dddot{K}(x) && \overset{7}{\dot{L(x)}} \\ \dot{t}M(x) && \dot{x}N(x) && \overset{4}{\dot{t}}O(x) && \overset{8}{\dot{x}}P(x) \end{aligned}$$

`\setDE{maxprimes=7}` *and/or* `\usepackage[maxprimes=7]{odesandpdes}`

$$f' \quad f'' \quad f''' \quad f'''' \quad f''''' \quad f'''''' \quad f''''''' \quad f^{(8)} \quad f^{(9)}$$

$$\dot{f} \quad \ddot{f} \quad \dddot{f} \quad \ddddot{f} \quad \overset{\cdot\cdot\cdot\cdot\cdot}{f} \quad \overset{\cdot\cdot\cdot\cdot\cdot\cdot}{f} \quad \overset{\cdot\cdot\cdot\cdot\cdot\cdot\cdot}{f} \quad \overset{8}{\dot{f}} \quad \overset{9}{\dot{f}}$$

2.2 "at x;" Usage Examples

Now, because the author is not an insane person, and went through the effort of learning how TEX deconstructs text into constitute registries and boxes, the way any sane person might. When using a non-starred version of a command, after the function is defined, you can place an '`at␣`⟨*point*⟩`;`', and the representation will shown according to notational convention.

```
\begin{align*}
   \ode[x]  c at 23\pi;   &= 1 \\
   \ode[x]^3 c   at 69;   &= 2 \\
   \ode[x]^{69} c at L;+t &= 3 \\
   \ode[x]^9  c af 420;   &= 4 \\
   \ode[x]^6  c  a t 13;  &= 5
\end{align*}
```

`\setDE{notation=Lagrange}`	`\setDE{notation=Leibniz}`	`\setDE{notation=Newton}`
$c'(23\pi) = 1$	$\left.\frac{\mathrm{d}c}{\mathrm{d}x}\right\rvert_{x=23\pi} = 1$	$\dot{c}(23\pi) = 1$
$c'''(69) = 2$	$\left.\frac{\mathrm{d}^3c}{\mathrm{d}x^3}\right\rvert_{x=69} = 2$	$\dddot{c}(69) = 2$
$c^{(69)}(L) + t = 3$	$\left.\frac{\mathrm{d}^{69}c}{\mathrm{d}x^{69}}\right\rvert_{x=L} + t = 3$	$\overset{69}{\dot{c}}(L) + t = 3$
$c^{(9)}af420; = 4$	$\frac{\mathrm{d}^9c}{\mathrm{d}x^9}af420; = 4$	$\overset{9}{\dot{c}}af420; = 4$
$c^{(6)}at13; = 5$	$\frac{\mathrm{d}^6c}{\mathrm{d}x^6}at13; = 5$	$\overset{6}{\dot{c}}at13; = 5$

As can be seen in the examples, this '*modifier*' is robust enough that one can write effectively any combination of characters after the function, excluding, *verbatim*, '`at␣`' and it will work as intended.

Important to note, due to a slight difference in how the notational styles are defined, only the Leibniz notation can take arguments for the function that involve subscripts and superscripts without delimiters. Mostly easily illustrated in this following example using the \pde command;

```
\begin{align*}
  \pde[y]   f_1          &= 1 \\
  \pde[y]   f_1    at L; &= 2 \\
  \pde[y]   f      at L; &= 3 \\
  \pde[y] {(f_1)}        &= 4 \\
  \pde[y] {(f_1)} at L;  &= 5
\end{align*}
```

\setDE{notation=Lagrange}	\setDE{notation=Leibniz}	\setDE{notation=Newton}
$f'_{y_1} = 1$	$\frac{\partial f_1}{\partial y} = 1$	$\dot{f}_1 = 1$
$f'_{y_1} atL; = 2$	$\left.\frac{\partial f_1}{\partial y}\right\|_{y=L} = 2$	$\dot{f}_1 atL; = 2$
$f'_y(L) = 3$	$\left.\frac{\partial f}{\partial y}\right\|_{y=L} = 3$	$\dot{f}(L) = 3$
$(f_1)'_y = 4$	$\frac{\partial (f_1)}{\partial y} = 4$	$(\dot{f_1}) = 4$
$(f_1)'_y(L) = 5$	$\left.\frac{\partial (f_1)}{\partial y}\right\|_{y=L} = 5$	$(\dot{f_1})(L) = 5$

2.3 Prime Count Limits

Because the Newton and Lagrange notation is procedural; the only limit is your imagination, and also the fact that TEX can only have something like 127 unplaced tokens at a time.

\setDE{maxprimes=69}