

The odesandpdes package^{*}

Anakin McMullen-Diermayr
you@yournamehere.org

Released 2024/01/04

Contents

^{*}This document corresponds to `odesandpdes` v1.0.0, dated 2024/01/04.

Abstract

Put text here.

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{odesandpdes}
```

into the preamble, Ideally after `amsmath/mathtools`. There are a couple notation options, which can be set document-wide with;

```
\usepackage[notation=<option>]{odesandpdes}|
```

The options included are based off of the three most common notations (according to Wikipedia), Lagrange, Leibniz, and Newton. However, if you wish to change it on a section to section basis, the command `\setDE{notation=<option>}` will change the form of the subsequent uses.

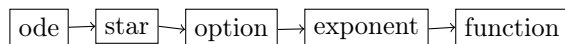
1 Usage

`\ode` The command(s) are approached with the philosophy of of an intuitive and modular usage. The general form can be understood as;

```
\ode[<variable>]<exponent>{<function>}
\ode*[<variable>]<exponent>
```

```
\pde \pde[<variable>]<exponent>{<function>}
\pde* \pde*[<variable>]<exponent>
```

And the starred variants which allow one to omit the function



While this should be sufficient for most use, there are a couple of tricks incorporated into the mechanism that makes this package better than the generic `\newcommand*\ode[2][t]{\{...\}}` Including automatic assignment of degree and starred variants. Example;

```
\begin{align*}
\ode N(t) = N(t) * t \&\& \ode N(t,x) = N(t,x) + \ode[x]^2 N(t,x) \&\& \pde^2 f^2
\end{align*}
```

$$\frac{dN(t)}{dt} = N(t) * t \qquad \frac{dN(t,x)}{dt} = N(t,x) + \frac{d^2}{dx^2} N(t,x) \qquad \frac{\partial^2}{\partial t^2} f^2$$

The `\ode` command will scan for an exponent between `[<variable>]` and `{<function>}`. Should there be indeed an exponent, that exponent gets ‘yoinked’ away and processed in accordance to the style of notation. In the case of Lagrange or Newton notation, there is the `maxprimes=<integer>` option for either the package or the `\setDE{<option>}` command;

```
\usepackage[maxprimes=<integer>]{odesandpdes} and/or
\setDE{maxprimes=<integer>}
```

3 being the default.

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`. As well, with the “proper” spacing, there is little need for the use of the braces, so as to help promote an easier workflow without always needed to worry about the damn brace. Not that one can not use the brace for personal taste. For the sake of parity, the `\pde` command will also take its variable in brackets.

The following examples all take the identical form, shown in the following verbatim enviroment.

```
\begin{align*}
```

```
\ode f(x)      && \ode[x]{f(x)} && \ode^1 f(x)      && \ode[x]^5 {f(x)} \\\
```

```
\ode*          && \ode*[x]      && \ode*^2          && \ode*[x]^6      \\\
```

```
\pde[t] {f(x)} && \pde[x] f(x) && \pde[t]^3{f(x)} && \pde[x]^7 f(x)  \\\
```

```
\pde*[t]       && \pde*[x]      && \pde*[t]^4       && \pde*[x]^8      \\\
```

```
\end{align*}
```

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

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

$f'(x)$	$f(x)'$	$f'(x)$	$f(x)^{(5)}$
$f'(t)$	$f'(x)$	$f''(t)$	$f^{(6)}(x)$
$f(x)_t'$	$f_x'(x)$	$f(x)_t'''$	$f_x^{(7)}(x)$
$f_t'(t)$	$f_x'(x)$	$f_t^{(4)}(t)$	$f_x^{(8)}(x)$

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

$\frac{df(x)}{dt}$	$\frac{df(x)}{dx}$
$\frac{d}{dt}$	$\frac{d}{dx}$
$\frac{\partial f(x)}{\partial t}$	$\frac{\partial f(x)}{\partial x}$
$\frac{\partial}{\partial t}$	$\frac{\partial}{\partial x}$

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

$\dot{f}(x)$	$\dot{f}(x)$
\dot{t}	\dot{x}
$\dot{f}(x)$	$\dot{f}(x)$
\dot{t}	\dot{x}

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

f'	f''	f'''	f''''	f'''''	f''''''
\dot{f}	\ddot{f}	$\dot{\dot{f}}$	$\ddot{\dot{f}}$	$\dot{\dot{\dot{f}}}$	$\ddot{\dot{\dot{f}}}$

Now, because I am not an insane person, and have mostly learned 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 $\langle value \rangle$;’,¹ and the representation will shown according to notational convention.

```

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

\setDE{notation=Leibniz}

\setDE{notation=Newton}

\setDE{notation=Lagrange}

c'(23\pi) = i
c'''(69) = i
c^{(69)}(L) + t = i
c^{(9)}af420; = i
c^{(6)}13 = i

```

Also the Newton and Lagrange notation is procedural;

```

\setDE{maxprimes=69}

f

```

¹heavily inspired by Tikz



Truly beautiful.

In the next semester I expect to try seeing if its possible to, given that you put multiple variable in the options, to procedurally generate partials that address separate variables sequatentially.

$$\frac{\partial^2}{\partial x \partial y}$$

\ode

2 Implementation

\ode Put explanation of \YOURMACRO's implementation here.

```
1 <*package>
=====
2 \RequirePackage{xkeyval}      % Taking in keys for option inputs
=====
3 \newcount\m@rkings%
4 \newcount\expo@de\relax%
5 \countdef\@detempv@1=255%
6 \newtoks\l@wert@ks%
7 \newtoks\upp@rt@ks%
8 \newtoks\@tpost@ks%
9 \newbox\@delowb@x%
10 \newbox\@deuppb@x%
11 \newbox\@deresb@x%
```

2.1 Pacakge options

```
12 % This defines the package option for notational styles
13 % Important to note that defining
14 % the command is not the same as using the command
15 \providecommand\@de@option{Leib}
16 % Creating a key system for options
17 % The pre-defined ones will be defined afterwards
18 \DeclareOptionX{notation}[default]%
19   {\def\@de@option{\csname @denot@tion@#1\endcsname}}%
20 % Pre-defined styles
21 \def\@denot@tion@Lagrange{Lagr}
22 \def\@denot@tion@Leibniz{Leib}
23 \def\@denot@tion@Newton{Newt}
24 \let\@denot@tion@default\@denot@tion@Leibniz
25
26 % This defines the package option for how many primes is okay
27 \DeclareOptionX{maxprimes}[3]%
28   {\m@rkings=#1\advance\m@rkings1}%

```

```

29
30
31 \DeclareOptionX*{\PackageWarning{odesandpdes}{'\CurrentOption' ignored}}
32
33
34 \ExecuteOptionsX{notation,maxprimes}
35 \ProcessOptionsX\relax
36
37 % =====
38 % Package option commands
39 % =====
40
41 % Macro for notation style option to be used by \setDE
42 \define@key[package]{@de}{notation}%
43   {\def\de@option{\csname @denotation@#1\endcsname}}%
44 % Macro for number of primes to be used by \setDE
45 \define@key[package]{@de}{maxprimes}%
46   {\m@rkings=#1\advance\m@rkings1}%
47 % Macro for switching of the style mid-document
48 \newcommand\setDE[1]%
49   {\setkeys[package]{@de}{#1}}
50
51 % =====
52 % To not conflict with amsmath
53 % =====
54
55 \@ifpackageloaded{amsmath}{% Purely because amsmath is a bitch
56   \let\de@ver=\@@over%
57   \let\de@top=\@@atop%
58   \let\de@bove=\@@above%
59   }{% Otherwise just uses the tex pr\de@expony@inkimitives
60   \let\de@ver=\over%
61   \let\de@top=\atop%
62   \let\de@bove=\above%
63   }
64
65 % =====
66 % Foundational macros
67 % =====
68
69 \def\d@@{\mathrm d} % Protected def for the d
70 \let\d@l=\partial % Protected def for the del
71 \def\dem@rkst@red{st@r@d}
72 \def\dem@rkst@r{n@st@r}
73
74 % Macros for ODEs
75 \def\ode{\let\de@perat@r=\d@@% sets the d
76   \de@ifst@r}
77 % Macro for PDEs
78 \def\pde{\let\de@perat@r=\d@l% sets the del
79   \de@ifst@r}
80 % Checks for a star
81 \def\de@ifst@r{\de@ifnextch@r* \@dest@red@rg \@den@st@r@rg}%
82 % Macros for starred and unstarred variants
83 \def\@dest@red@rg*{\@de@isst@r{st@r@d} \@de@ifbr@ck}
84 \def\@den@st@r@rg {\@de@isst@r{n@st@r} \@de@ifbr@ck}
85
86
87
88 % Macro for optional and no optional args
89 \def\de@ifbr@ck{\de@ifnextch@r[\@de@option@l@rg{\@de@ption@l@rg[t]}}

```

```

90 \def\@de@option@l@rg[#1]{\expandafter\l@wert@ks{#1}\relax \@de@ifexpon}%
91 \def\@de@ifexpon{\@deifnextch@r~\@de@exponent@rg{\@de@exponent@rg~1}}
92
93 % Macro for yoinking the exponent
94 \def\@de@exponent@rg~#1{\expo@de#1\relax \@deisitorisntitastar}
95
96 % Yoinks the function variable
97 \def\@dest@r@dfuncy@ink{\expandafter\@definalchosenform}
98 \def\@den@st@rfuncy@ink{\expandafter\@de@funcy@inkf@rm}
99
100 \def\@de@func@Leib#1 {\expandafter\upp@rt@ks{#1}\relax \@deif@tpos}
101 \def\@de@func@ther #1{\expandafter\upp@rt@ks{#1}\relax \@deif@tpos}
102 \let\@de@func@Lagr\@de@func@ther
103 \let\@de@func@Newt\@de@func@ther
104
105 % Yoinks the at args
106 \def\@deif@tpos{\@deifnextch@r{a}\@de@tfinder \@definalchosenform}
107
108 % Macro for authentication of the 'at '
109 \def\@de@tfinder a#1{\ifx#1t\expandafter\@de@tom@at\else
110   \@definalchosenform a#1\fi}%
111
112 % =====
113 % Atoms of the Function
114 % =====
115
116 % Used for choosing which notational form to take
117 \def\@de@tom@def#1{\def\@de@tom@part{\csname @de@tom@#1\endcsname}}
118 \def\@de@tom@optional[#1]{\expandafter\l@wert@ks{#1}\relax}
119 \def\@de@tom@at#1;\{\expandafter\@tpost@ks{#1}\relax \@de@t@posf@rm}
120 \def\@de@tom@function #1{\expandafter\upp@rt@ks{#1}\relax}
121 \def\@de@tom@exponent ~#1{\expo@de=#1\relax}
122
123 % =====
124 % Ancilliary Functions
125 % =====
126
127 \def\@de@isst@r#1{%
128   \def\@deisitorisntitastar{\csname @de#1funcy@ink\endcsname}%
129   \def\@definalchosenform{\csname #1@\@de@option\endcsname}}
130 \def\@de@t@posf@rm{\csname @de@tpl@ce@\@de@option\endcsname}%
131 \def\@de@funcy@inkf@rm{\csname @de@func@\@de@option\endcsname}%
132
133 % Macros for streamlining the process of checking the next character
134 \def\@deifnextch@r#1#2#3{
135   \let\@dedesiredtoken=#1\relax
136   \def\@de@tmpA{#2}%
137   \def\@de@tmpB{#3}
138   \futurelet\@detesttoken\@denext@rg}
139
140 % Need to go through some extra hoops to reject space tokens
141 \def\@denext@rg{%
142   \ifx\@detesttoken\@sptoken%
143     \let\@de@nextact\@degobblesp@ce\else% Space
144     \ifx\@detesttoken\@dedesiredtoken%
145       \let\@de@nextact\@de@tmpA\else% if
146       \let\@de@nextact\@de@tmpB\fi\fi% ifn't
147     \@de@nextact}
148
149
150 % Making sure that the command does not persist past this package

```



```

151 \let\@desavedef\<
152
153 % By defining the function with a non-character token, the space matters
154 \def\<\@degobblesp@ce}
155 \expandafter\def\< {\futurelet\@detesttoken\@denext@rg}
156
157 % =====
158 % Notational shaping time
159 % =====
160
161 % Macro for Lagr+star
162 \def\st@r@d@Lagr{%
163   \setbox\@deresb@x\hbox{%
164     f~{\mkern1mu\m@kem@rk\lagr@prime\lagr@prime\braced@xpon}_\{m@kep@rtLagr}%
165     \mkern-0.5mu\left(\the\l@wert@ks\right)%
166     }%
167   \@derele@se}%
168 % Macro for Lagr
169 \def\nst@r@Lagr{%
170   \setbox\@deresb@x\hbox{%
171     \the\upp@rt@ks~{\mkern1mu\m@kem@rk\lagr@prime\lagr@prime\braced@xpon}_\{m@kep@rtLagr}%
172     }%
173   \@derele@se}%
174 % Macro for Lagr at point
175 \def\@de@tpl@ce@Lagr{%
176   \noexpand\hbox{%
177     \nst@r@Lagr\mkern-0.3mu\left(\the\@tpost@ks\right)%
178     }%
179
180 % Macro for the prime used by the lagrangian notation
181 \def\lagr@prime{\mkern0.35mu\prime\global\advance\expo@de-1}
182 % Macro for making the exponent in parenthesis
183 \def\braced@xpon{\left(\the\expo@de\right)}
184 % Macro for Lagrange partial notations
185 \def\m@kep@rtLagr{\ifx\@de@perat@r@d@l\the\l@wert@ks\else\empty\fi}
186
187
188 % Macro for Leib+star
189 \def\st@r@d@Leib{%\beginnext%
190   \setbox\@deuppb@x\hbox{\$@de@perat@r~{\empty@rexpon}}%
191   \b@se@Leib}%
192 % Macro for Leib
193 \def\nst@r@Leib{%
194   \setbox\@deuppb@x\hbox{\$mkern0.40mu\@de@perat@r~{\empty@rexpon}\the\upp@rt@ks}%
195   \b@se@Leib}%
196
197 % Macro for the base Leibniz form
198 \def\b@se@Leib{%
199   \setbox\@delowb@x\hbox{\$@de@perat@r\mkern0.40mu\the\l@wert@ks~{\empty@rexpon}}%
200   \setbox\@deresb@x\hbox{\kern0.50pt%
201     \$\raise2pt\box\@deuppb@x\@de@ver\lower5pt\box\@delowb@x$%
202     \kern0.50pt}%
203   \@derele@se}%
204
205 % Macro for specification of where the ode is defined
206 \def\@de@tpl@ce@Leib{%
207   \noexpand\hbox{%
208     \left.\nst@r@Leib\mkern0mu\right|_\{\mkern1mu\displaystyle\the\l@wert@ks\mkern2mu%
209     \rlap{\scriptstyle=\mkern2mu\the\@tpost@ks}}}%
210     }%
211   }%

```

```

212
213
214 % Macro for Newt+star
215 \def\st@r@d@Newt{%
216   \setbox\@delowb@x\hbox{$\the\l@wert@ks$}%
217   \b@se@Newt}%
218 % Macro for Newt
219 \def\n@st@r@d@Newt{%
220   \setbox\@delowb@x\hbox{$\displaystyle\the\upp@rt@ks$}%
221   \b@se@Newt}%
222 % Macro for the base Netwon form
223 \def\b@se@Newt{%
224   \setbox\@deuppb@x\hbox{\vbox{\baselineskip=0pt\lineskip=-1pt%
225     \m@kem@rk\@ned@ts\tw@d@ts\newt@end@t}}}%
226   \setbox\@deresb@x\hbox{\vbox{\baselineskip=0pt\lineskip=-0.5pt%
227     \hbox{\raise0.00ex\box\@deuppb@x}%
228     \hbox{\raise0.00ex\box\@delowb@x}}}%
229   \@derele@se}
230
231 % Macro for Newton at point
232 \def\@de@tpl@ce@Newt{%
233   \noexpand\hbox{$%
234     \n@st@r@d@Newt\mkern-02mu\left(\the\@tpost@ks\right)%
235     $}}%
236
237 % Macro for numbering
238 \def\newt@end@t{\hbox{\vbox{%
239   \hbox to 5pt{\hss\raise0.50ex\hbox{$\scriptstyle\empty@rexpon$}\hss}%
240   \hbox to 5pt{\hss\hbox{$\displaystyle\cdot$}\hss}}}%
241 % Macro for Netwon partial notations
242 \def\m@kep@rtNewt{\ifx\@de@perat@r\d@l\empty\fi}
243 % Macro for dots
244 % Tests as "mod2" testing of dot groupings
245 \def\tw@d@ts{\ifnum\expo@de>1%
246   \advance\expo@de-2\hbox to 5pt{\hss$\cdot\cdot$\hss}\fi}%
247 \def\@ned@ts{\@detempv@l=\the\expo@de%
248   \loop\ifnum\@detempv@l>2%
249     \advance\@detempv@l-2\repeat%
250   \ifnum\@detempv@l<2%
251     \advance\expo@de-1\hbox to 5pt{\hss$\cdot$\hss}\fi}%
252
253 % =====
254 % Notational Shaping Tools
255 % =====
256
257 % Macro for determining if the exponent should be empty
258 \def\empty@rexpon{\ifnum2>\expo@de\empty\else\the\expo@de\fi}
259 % Macro for checking if marks should be used or something else
260 \def\m@kem@rk#1#2#3{%
261   \ifnum\expo@de<\m@rkings% If not zero, check if less than max allowed
262     #1\m@rkrepe@ting#2\else% Make primes while below limit
263     #3\fi}
264 % Macro for creating the appropriate number of marks, primes or whatever
265 \def\m@rkrepe@ting#1{\loop\ifnum\expo@de>0#1\repeat}
266
267 % Shorthand for allowing the boxes to rise to the surface
268 \def\@derele@se{\noexpand{\box\@deresb@x}}
269
270
271 \let\<\@desavedef
272 % =====

```

273 \endinput

274 \endpackage