
The derivative package

Written by:
Simon Jensen
sjelatex@gmail.com

Released:
v1.1
2021/06/03

The `derivative` package provides a set of commands which makes writing ordinary and partial derivatives of arbitrary order in a straight forward manner. Additionally, this package provides a set of commands to define variants of the aforementioned derivatives. A set of optional arguments along with lots of package options allow for easy and great flexibility over the derivative's format, such as where the function is positioned, point of evaluation, and switching between fraction styles. Moreover, the mixed order of the partial derivative and variants hereof is automatically computed. This package is written in the `expl3` language and requires therefore the `LATEX3` package bundles `l3kernel` and `l3package`. Additionally, the `mleftright` package is optional and provides the improved automatically scaling `\mleft` and `\mright`.

Contents

1 Derivative	4
1.1 Other derivatives	6
2 Differentials	8
2.1 Other differentials	8
3 Options	10
3.1 Categories	10
3.2 Package options	11
3.3 Derivative options	11
3.4 Differential options	18
3.5 All derivatives and differentials	21
4 Defining variants	22
4.1 Derivative variant	22
4.2 Differential variant	22
5 The mixed order	24
5.1 Sorting algorithms	24
5.1.1 Examples	24
5.2 The numerical term	25
5.3 Reversing the sort algorithm	26
5.4 Mixed order override	26
6 Miscellaneous	27
6.1 Slashfrac	27
7 To do	28
7.1 Future implementation	28
7.2 Future considerations	28
Index	29
Index of Options	29
Index of Commands	30
Change history	31

This package started as a personal package a few years ago that I used in various projects. Firstly, it was written in $\text{T}_{\text{E}}\text{X}$ and $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, which gave rise to various errors as the complexity of the package grew larger and ended up as a spaghetti code. Therefore, the code was rewritten into the $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}3$ language and may now be easily maintained. Originally this package was written because there did not exist any great package for derivatives. It was first much later that I discovered the `diffcoeff` package, which does a good job. However, at this time, I had already written much of the code with no documentation. So I decided to write one and make it into a package for the public.

As a note on terminology, I use the wording *infinitesimal* abbreviated as `inf` for the operator symbols d, ∂, δ etc. that used in derivatives like $\frac{dy}{dx}, \frac{\partial y}{\partial x}, \frac{\delta y}{\delta x}$ and in differentials like $dx, \partial_x, \delta x$. In the description of macros and options, I often write `cs-⟨placeholder⟩` to denote a comma-separated list of `⟨placeholder⟩`. For example, `[⟨cs-numbers⟩]` is used in the option section for math space keys to denote comma-separated list of numbers. It should also be noted that whenever an argument reads `⟨keyvalue list⟩`, it means that it is a comma-separated list of keyvalue pairs.

1 Derivative

`\pdv` * $[\langle keyval list \rangle]{\langle function \rangle}_{\langle variables \rangle}_{\langle point_1 \rangle}^{\langle point_2 \rangle}$

The partial derivative `\pdv` is defined with a set of mandatory and optional arguments that either typeset specific parts or changing the style of the derivative. In this package, the partial derivative is defined with an italic partial differential ∂ as

```
\DeclareDerivative{\pdv}{\partial}[style-var=multiple, sep-inf-ord=1,
delims-eval=(), delims-eval-/=()]
```

- * The first argument of `\pdv` is an optional star that determines where the function is typeset; either in the numerator of the fraction or next to the fraction. Using the option `switch-*=false`, the function is typeset in the numerator when the star is absent, and next to the fraction when the star is present as shown below

$$\begin{aligned} \text{\pdv}\{f\}\{x,y\} &\implies \frac{\partial^2 f}{\partial x \partial y} \\ \text{\pdv*}\{f\}\{x,y\} &\implies \frac{\partial^2}{\partial x \partial y} f \end{aligned}$$

The effect of the star's presence can be switched around using `switch-*=true` i.e. the equations in the above example are exchanged.

$[\langle keyval list \rangle]$

Updated: v1.0

The second argument is optional and is written inside square brackets. This argument is used to set the options of the derivative using a `key=value` semantic. For example the order of differentiation can be set via the `order` option

$$\begin{aligned} \text{\pdv}[order=\{2,3\}]\{f\}\{x,y\} &\implies \frac{\partial^5 f}{\partial x^2 \partial y^3} \\ \text{\pdv}[order=\{\beta,a,n+2a\}]\{f\}\{x,y,z\} &\implies \frac{\partial^{3a+\beta+n} f}{\partial x^\beta \partial y^a \partial z^{n+2a}} \\ \text{\pdv}[order=\{2,n^2,n^2-1\}]\{f\}\{x,y,z\} &\implies \frac{\partial^{2n^2+1} f}{\partial x^2 \partial y^{n^2} \partial z^{n^2-1}} \\ \text{\pdv}[order=\{2,n^2\}]\{f\}\{x,y,z\} &\implies \frac{\partial^{n^2+3} f}{\partial x^2 \partial y^{n^2} \partial z} \end{aligned}$$

The list of options that can be applied to derivatives can be found in [section 3.3](#).

$\{\langle function \rangle\}$

This is the first mandatory argument that typeset the function that is to be differentiated

$$\begin{aligned} \text{\pdv}\{f(x,y,z)\}\{x,y,z\} &\implies \frac{\partial^3 f(x,y,z)}{\partial x \partial y \partial z} \\ \text{\pdv}\{e^x \sin(y) \ln(z)\}\{x,y,z\} &\implies \frac{\partial^3 e^x \sin(y) \ln(z)}{\partial x \partial y \partial z} \end{aligned}$$

The function is simply typeset in the numerator or next to the fraction.

- / The fourth argument is an optional slash written *between the function and the variable* arguments and determines which fraction style the derivative is typeset with, as described in [section 3.3](#). Using the option `switch-/=false` and the package's default fractions, the derivative is typeset with `\frac` when the slash is absent, and `\slashfrac`¹ when the slash is present, as shown below

$$\begin{aligned} \text{\color{red}\code{\pdv{f}{x,y}}} &\implies \frac{\partial^2 f}{\partial x \partial y} \\ \text{\color{red}\code{\pdv{f}/{x,y}}} &\implies \partial^2 f / \partial x \partial y \end{aligned}$$

As for the star argument, the effect of the slash's presence can be switched around using `switch-/=true` i.e. the equations in the above example are exchanged.

- `{(variables)}` This is the second and final mandatory argument, which purpose is to typeset the variable in which the function is differentiated with respect to. The variables are given in a comma-separated list e.g. `{x, y, z}`

$$\begin{aligned} \text{\color{red}\code{\pdv{f}{x}}} &\implies \frac{\partial f}{\partial x} \\ \text{\color{red}\code{\pdv{f}{x,y}}} &\implies \frac{\partial^2 f}{\partial x \partial y} \end{aligned}$$

- `_{\langle point_1 \rangle}^{\langle point_2 \rangle}` This is the last optional argument that specifies the point(s) of evaluation or variables held constant. It is an *e-type* argument in the `xparse` language and is given by `e_{_}^{\wedge}`. This means that the subscript `_` and superscript `^` accepts an argument given in braces. Moreover, the order of `_` and `^` is independent as shown below

$$\begin{aligned} \text{\color{red}\code{\pdv{f}{x,y}_{(x_1,y_1)}}} &\implies \left(\frac{\partial^2 f}{\partial x \partial y} \right)_{(x_1,y_1)} \\ \text{\color{red}\code{\pdv{f}{x,y}^{(x_2,y_2)}}} &\implies \left(\frac{\partial^2 f}{\partial x \partial y} \right)^{(x_2,y_2)} \\ \text{\color{red}\code{\pdv{f}{x,y}_{(x_1,y_1)}^{(x_2,y_2)}}} &\implies \left(\frac{\partial^2 f}{\partial x \partial y} \right)^{(x_2,y_2)}_{(x_1,y_1)} \\ \text{\color{red}\code{\pdv{f}{x,y}^{(x_2,y_2)}_{(x_1,y_1)}}} &\implies \left(\frac{\partial^2 f}{\partial x \partial y} \right)_{(x_1,y_1)}^{(x_2,y_2)} \end{aligned}$$

The subscript argument is commonly used as the point of evaluation or variables held constant. If needed, the superscript argument may be used for the second point of evaluation.

¹which is a macro defined by the package, see [section 6.1](#) for more information.

1.1 Other derivatives

This package offers five other derivatives: Ordinary derivative `\odv`, material derivative `\mdv`, functional derivative `\fdv`, the average rate of change `\adv` and the Jacobian `\jdv`. A unique feature of this package, is that you can define your own derivatives as described in [section 4.1](#).

`\odv`

*`[\keyval list]{\langle function \rangle / \langle variables \rangle}_{\langle point_1 \rangle}^{\langle point_2 \rangle}`

Updated: v1.1

In this package, the ordinary derivative is defined with an upright lowercase d if the package option `upright=true` is used else it will use an italic lowercase d

```
\DeclareDerivative{\odv}{\mathrm{d}}
\DeclareDerivative{\odv}{\mathnormal{d}}
```

`upright=true`
`italic=true`

because it is used by many nowadays books

$$\frac{df}{dx} = \lim_{h \rightarrow 0} \left(\frac{f(x+h) - f(x)}{h} \right)$$

However it can always be changed with the `style-inf` key.

`\mdv`

*`[\keyval list]{\langle function \rangle / \langle variables \rangle}_{\langle point_1 \rangle}^{\langle point_2 \rangle}`

Updated: v1.1

The material derivative is used in some branches of physics e.g. thermodynamic, fluid dynamic etc. It comes with an upright uppercase D when the package option `upright=true` else it will use an italic uppercase D. In this package, the material derivative is defined as

```
\DeclareDerivative{\mdv}{\mathrm{D}}
\DeclareDerivative{\mdv}{\mathnormal{D}}
```

`upright=true`
`italic=true`

In physics, the material derivative is defined by

$$\frac{D\varphi(\mathbf{r}, t)}{Dt} := \frac{\partial\varphi(\mathbf{r}, t)}{\partial t} + \dot{\mathbf{r}} \cdot \nabla\varphi(\mathbf{r}, t)$$

`\fdv`

*`[\keyval list]{\langle function \rangle / \langle variables \rangle}_{\langle point_1 \rangle}^{\langle point_2 \rangle}`

The functional derivative is used in the calculus of variation and uses a lowercase delta. With the package's default settings, it will use an italic delta. In this package, the functional derivative is defined as

```
\DeclareDerivative{\fdv}{\delta}
```

In physics, it is for example used in the Lagrange equation or in the derivation of the Hartree-Fock equation

$$\frac{\delta I}{\delta q_\alpha} = \frac{\partial L}{\partial q_\alpha} - \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_\alpha} = 0, \quad \frac{\delta \mathcal{L}}{\delta \psi_n^*} = \hat{F}|\psi_n\rangle - \epsilon_n|\psi_n\rangle = 0,$$

respectively.

`\adv`

*`[\keyval list]{\langle function \rangle / \langle variables \rangle}_{\langle point_1 \rangle}^{\langle point_2 \rangle}`

The average rate of change is defined to use an upright uppercase delta with the packages default settings. In this package, the average rate of change is defined as

```
\DeclareDerivative{\adv}{\Delta}
```

The average rate of change is used to determine the slope of a straight line

$$\frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$$

`\jdv` **[\keyval list]{\function}/{\variables}_{\point_1}^{\point_2}*

Updated: v1.0

The Jacobian is defined to use an italic partial differential with the package's default settings. Also a pair of parentheses is automatically inserted around the function and variable. In this package, the Jacobian is defined as

```
\DeclareDerivative{\jdv}{\partial}[fun=true, var=1]
```

which gives

$$\frac{\partial(f, g, h)}{\partial(x, y, z)}$$

2 Differentials

`\odif` * $[\langle keyval list \rangle]\{\langle variables \rangle\}$

New: v1.0
Updated: v1.1

The differential `\odif` is defined with a set of mandatory and optional arguments that either typeset specific parts or changing the style of the differential. It is defined with an upright lowercase d when `upright=true` else it will be defined with an italic lowercase d

```
\DeclareDifferential{\odif}{\mathrm{d}}
\DeclareDifferential{\odif}{\mathnormal{d}}
```

`upright=true`
`italic=true`

- * The first argument of `\odif` is an optional star that determines the notation style of the differential. Using the option `switch-*=false`, the variables and orders are typeset in a subscript and superscript respectively when the star is present. When the star is absent, then the macros associated infinitesimal is placed in front of each variable as shown below

$$\begin{aligned} \odif\{x,y,z\} &\implies dx dy dz \\ \odif*\{x,y,z\} &\implies d_{x,y,z} \end{aligned}$$

The effect of the star's presence can be switched around using `switch-*=true` i.e. the equations in the above example are exchanged, as described in [section 3.3](#).

$[\langle keyval list \rangle]$ The second argument is optional and is written inside square brackets. This argument is used to set the options of the differential using a `key=value` semantic.

$$\begin{aligned} \odif[order=\{n,3\}]\{x,y,z\} &\implies d^n x d^3 y dz \\ \odif[sep-var-inf=0]\{x,y,z\} &\implies dx dy dz \\ \odif*[sep-var-var=0]\{x,y,z\} &\implies d_{xyz} \end{aligned}$$

The list of options that can be applied to differential can be is found in [section 3.4](#)

$\{\langle variables \rangle\}$ This is the mandatory argument, which purpose is to typeset the variables, as shown below

$$\begin{aligned} \odif\{x\} &\implies dx \\ \odif\{s_1,s_2,\dots,s_n\} &\implies ds_1 ds_2 \dots ds_n \end{aligned}$$

2.1 Other differentials

This package offers four other differentials: partial differential `\pdif`, uppercase D `\mdif`, delta `\fdif` and Delta `\adif`. A unique feature of this package, is that you can define your own differential as described in [section 4.2](#).

`\pdif` * $[\langle keyval list \rangle]\{\langle variables \rangle\}$

The partial differential `\pdif` is typically used as a shorthand notation for the partial derivative. In this package it is defined as

```
\DeclareDifferential{\pdif}{\partial}[style-notation=single,
style-notation-*=mixed]
```

The non-star and star version gives

$$\partial_{x,y,z}^{i,j,k} := \frac{\partial^{i+j+k}}{\partial x^i \partial y^j \partial z^k}$$

$$\partial_x^i \partial_y^j \partial_z^k := \frac{\partial^{i+j+k}}{\partial x^i \partial y^j \partial z^k}$$

respectively.

```
\mdif *[\langle keyval list \rangle]{\langle variables \rangle}
```

Updated: v1.1

Another commonly used shorthand notation for various derivatives is the differential with an uppercase D. In this package it is defined with a upright D when `upright=true` else it is defined with an italic uppercase D

```
\DeclareDifferential{\mdif}{\mathrm{D}}[style-notation=single,
style-notation-*=mixed]
\DeclareDifferential{\mdif}{\mathnormal{D}}[style-notation=single,
style-notation-*=mixed]
```

`upright=true`

`italic=true`

The non-star and star version gives

$$D_{x,y,z}^{i,j,k}$$

$$D_x^i D_y^j D_z^k$$

respectively.

```
\fdif *[\langle keyval list \rangle]{\langle variables \rangle}
```

When working with functional derivatives another differential is frequently encountered. This differential uses a delta symbol and is defined as

```
\DeclareDifferential{\fdif}{\delta}
```

For example an expression like

$$\delta J = \int_a^b \frac{\partial L}{\partial f} \delta f(x) + \frac{\partial L}{\partial f'} \frac{d}{dx} \delta f(x) dx$$

is frequently encountered.

```
\adif *[\langle keyval list \rangle]{\langle variables \rangle}
```

To wrap it up, a differential for differences is defined as

```
\DeclareDifferential{\adif}{\Delta}
```

For example

$$\Delta E = E_2 - E_1$$

3 Options

This package accepts its options using the well-known *key=value* syntax. The keys are divided into categories, for which each key has its associated category as a prefix.

<code>\derivset</code>	<code>{⟨command⟩}[⟨keyval list⟩]</code>
Updated: v1.0	The <code>\derivset</code> is meant to set the default values for derivatives and differentials options in the preamble. It can be used in the document but this is what the new <code>[⟨keyval list⟩]</code> arguments are for in the derivative and differential commands.
<code>{⟨command⟩}</code>	A mandatory argument that determines which command the <i>key=value</i> pairs are assigned to, where the allowed <code>⟨commands⟩</code> are the derivatives and differentials defined by the package and you, see sections 4.1 and 4.2 for more information. The special value <code>all</code> is also allowed, which gives access to the options that applies to <i>all</i> derivatives and differentials.
<code>[⟨keyval list⟩]</code>	This optional argument accepts its input as a comma-separated list of <i>key=value</i> pairs. Leaving out <code>[⟨keyval list⟩]</code> sets the options to the package's default settings for the chosen <code>{⟨command⟩}</code> . E.g. <code>\derivset{\odv}</code> sets the options for the ordinary derivative to the package's default settings.

3.1 Categories

This section seeks to give a detailed description of each category.

- The `style-⟨...⟩` keys sets either the font style or the fraction style of `⟨...⟩`. The font styles `\mathnormal/\symnormal` and `\mathrm/\symup` are commonly used in literature. The fraction style can be either `\frac`, `\dfrac`, `\tfrac`, `\sfracR`, `\slashfrac1` or any command with two mandatory arguments.
- The `delims-⟨...⟩` keys sets the delimiters used around the `⟨...⟩`. The Rule of Two applies here: 'Always two there are, no more, no less. A left and a right delimiter'. The allowed delimiters are those that can be scaled with `\left`, `\big` etc.
- The `scale-⟨...⟩` keys sets the size of the `⟨...⟩`'s delimiters. The values `big`, `Big`, `bigg`, and `Bigg` are self-explanatory and internally a left and right version are used e.g. `\bigl` and `\bigr`. The value `none` inserts the delimiters (except periods) unscaled. While the value `auto` scales the delimiters automatically using `\left` and `\right`.
- The `sep-⟨...⟩-⟨...⟩` keys inserts their value between `⟨...⟩` and `⟨...⟩`. If the value is a comma-separated list of up to three numbers (e.g. `{x,y,z}`) it will be converted into the syntax `\muskip x mu plus y mu minus z mu` and inserted. All other values are used as they were given.
- The `switch-⟨...⟩` keys serves to change an argument's behaviour by swapping the effect of an optional character's presence.

- The `sort-⟨...⟩` keys deals with the sorting algorithm behind the mixed order, where you can choose the sorting method that suits you best.
- The miscellaneous keys does not fit into any of the above categories and no longer have the prefix `misc`

Note. A value with superscripted **R** means it requires a package. Additionally, some keys have two versions; with and without `-/` at the end. These keys are related to the slash argument. If `switch-/=false` then the keys with and without `-/` are used when the slash argument is present and absent, respectively. Setting `switch-/=true` then these keys are used in the opposite cases of the slash argument's presence.

3.2 Package options

The package options can be used with

$$\backslash\text{usepackage}[\langle\text{keyval list}\rangle]\{\text{derivative}\}$$

<code>italic</code>	<code>true, false</code>	<code>false</code>
<hr/> New: v1.1	Sets the font-style of the infinitesimals d and D used in <code>\odv</code> , <code>\mdv</code> , <code>\odif</code> and <code>\mdif</code> to italic using <code>\mathnormal</code> . The options <code>italic</code> and <code>upright</code> are mutually exclusive.	
<code>upright</code>	<code>true, false</code>	<code>true</code>
<hr/> New: v1.1	Sets the font-style of the infinitesimals d and D used in <code>\odv</code> , <code>\mdv</code> , <code>\odif</code> and <code>\mdif</code> to upright using <code>\mathrm</code> . The options <code>italic</code> and <code>upright</code> are mutually exclusive.	

3.3 Derivative options

The options in this subsection are available for derivatives defined by the package and you with the commands in [section 4.1](#).

Style

<code>style-inf</code>	$\langle\text{math-font-style}\rangle\langle\text{infinitesimal}\rangle$	<code>d</code>
<hr/> Updated: v1.0	Sets the derivative's infinitesimal including font style. This change was necessary to allow multiple font styles, e.g. <code>style-inf=\mathbf{\mathrm{d}}</code> . The default infinitesimal is a plain 'd' with no associated font style.	
<code>style-frac</code>	$\langle\text{fraction}\rangle$	<code>\frac</code>
	The derivative uses the fraction set by this key whenever <code>switch-/=false</code> and the slash argument is absent. The key's default value is the usual fraction <code>\frac</code> .	
<code>style-frac-/</code>	$\langle\text{fraction}\rangle$	<code>\slashfrac</code>
	The derivative uses the fraction set by this key whenever <code>switch-/=false</code> and the slash argument is present. The key's default value is a text-styled fraction <code>\slashfrac</code> ¹ i.e. on the form $\partial^2 f / \partial x \partial y$.	

`style-var` `single, multiple` `single`

New: v1.0 This option determines how the derivative treats its variables and some features. The variable argument is treated as a single variable when `single` is used. Moreover the mixed order is set equal to the first order given in `order=<order>` which by default is 1. The variable argument is treated as a comma separated list of variables with `multiple` similarly to a partial derivative. Moreover the mixed order is automatically calculated from the list of orders given in `order=<order>`.

Scaling

`scale-eval` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used for the point of evaluation. This scaling is used whenever `switch-/=false` and the slash argument is absent. The key's default value is set to scale the delimiters automatically.

`scale-eval-/` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used for the point of evaluation. This scaling is used whenever `switch-/=false` and the slash argument is present. The key's default value is set to scale the delimiters automatically.

`scale-fun` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used around the function. The scaling is used whenever `fun=true` is used. The key's default value is set to scale the delimiters automatically.

`scale-var` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used around the variable. The scaling is used whenever `var=true` is used. The key's default value is set to scale the delimiters automatically.

`scale-frac` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used for around the fraction. This scaling is used whenever `switch-/=false`, the slash argument is absent and `frac=true` is used. The key's default value is set to scale the delimiters automatically.

`scale-frac-/` `auto, none, big, Big, bigg, Bigg` `auto`

This key sets the size of the delimiters used for around the fraction. This scaling is used whenever `switch-/=false`, the slash argument is present and `frac=true` is used. The key's default value is set to scale the delimiters automatically.

Delimiters

`delims-eval` `<left delimiter><right delimiter>` `. \rvert`

This key sets the left and right delimiters used to indicate the point of evaluation. These delimiters are inserted whenever `switch-/=false` and the slash argument is absent. The key's default left and right delimiters are a left and a right parenthesis, respectively.

<code>delims-eval-/</code>	$\langle left\ delimiter \rangle \langle right\ delimiter \rangle$	<code>. \rvert</code>
	This key sets the left and right delimiters used to indicate the point of evaluation. These delimiters are inserted whenever <code>switch-/=false</code> and the slash argument is present. The key's default left and right delimiters are a left and a right parenthesis, respectively.	
<code>delims-fun</code>	$\langle left\ delimiter \rangle \langle right\ delimiter \rangle$	<code>()</code>
	This key sets the left and right delimiters used around the function and these are inserted whenever <code>fun=true</code> is used. The key's default left and right delimiters are a left and a right parenthesis, respectively.	
<code>delims-var</code>	$\langle left\ delimiter \rangle \langle right\ delimiter \rangle$	<code>()</code>
	This key sets the left and right delimiters used around the variable and these are inserted whenever <code>var=true</code> is used. The key's default left and right delimiters are a left and a right parenthesis, respectively.	
<code>delims-frac</code>	$\langle left\ delimiter \rangle \langle right\ delimiter \rangle$	<code>()</code>
	This key sets the left and right delimiters used around the fraction in the derivative e.g. $(\frac{\partial}{\partial x})y$. These delimiters are inserted whenever <code>switch-/=false</code> , the slash argument is absent and <code>frac=true</code> is used. The key's default left and right delimiters are a left and a right parenthesis, respectively.	
<code>delims-frac-/</code>	$\langle left\ delimiter \rangle \langle right\ delimiter \rangle$	<code>()</code>
	This key sets the left and right delimiters used around the fraction in the derivative e.g. $(\partial/\partial x)y$. These delimiters are inserted whenever <code>switch-/=false</code> , the slash argument is present and <code>frac=true</code> is used. The key's default left and right delimiters are a left and a right parenthesis, respectively.	

Math spacing

The options in this subsection inserts extra horizontal math space. The below equation illustrates where the space is inserted for each option

$$\begin{array}{ccc}
 \frac{\partial}{\partial x} \langle \text{frac-fun} \rangle f & \frac{\partial \langle \text{inf-fun} \rangle f}{\partial \langle \text{inf-var} \rangle x} & \frac{\partial^2 f}{\partial x \langle \text{var-inf} \rangle \partial y} \\
 \frac{\partial \langle \text{inf-ord} \rangle^2 f}{\partial x \langle \text{var-ord} \rangle^2} & \frac{\partial^3 \langle \text{ord-fun} \rangle f}{\partial x^2 \langle \text{ord-inf} \rangle \partial y} & \left(\frac{\partial f}{\partial x} \right) \langle \text{eval-sp} \rangle_{x_2} \\
 & & \langle \text{eval-sb} \rangle_{x_1}
 \end{array}$$

where it have been split into six to give a better overview. Here $\langle \dots \rangle$ means the value given to the option `sep- $\langle \dots \rangle$` . Some of the math spaces shown above are only used when `style-var=multiple`. Additionally, when the option `style-var=single` is used then the following math space is used in the denominator

$$\frac{\partial(f, g)}{\partial(x \langle \text{var-var} \rangle y)}$$

where the Jacobian have been used as an example.

<code>sep-inf-ord</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
	Sets the math space that is inserted in the infinitesimal's power left to the mixed order in the numerator. It is only inserted when the mixed order is different from 1. The key's default value is 0 mu.	
<code>sep-inf-fun</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
	This key sets the math space that is inserted between the infinitesimal and the function when the mixed order <i>is equal</i> to 1. The space is only inserted when a non-blank function is printed <i>in the numerator</i> . The key's default value is 0 mu.	
<code>sep-ord-fun</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
	This key sets the math space that is inserted between the infinitesimal's order and the function when the mixed order <i>is different</i> from 1. The space is only inserted when a non-blank function is printed <i>in the numerator</i> . The key's default value is 0 mu.	
<code>sep-frac-fun</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
<u>New: v1.0</u>	This key sets the math space that is inserted between the fractional part of the derivative and the function. The space is only inserted when a non-blank function is printed <i>next to the derivative</i> . The key's default value is 0 mu.	
<code>sep-inf-var</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
	This key sets the math space that is inserted between the infinitesimal and the following variable. The space is only inserted when a non-blank variable is given. The key's default value is 0 mu.	
<code>sep-var-ord</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	0
	This key sets the math space that is inserted in a variable's power left to the order. The space is only inserted when the order is different from 1. The key's default value is 0 mu.	
<code>sep-var-inf</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	$\backslash\mathop{\!}\!$
	They key sets the math space that is inserted between a variable and the following infinitesimal. The space is only inserted when the variable's order <i>is equal</i> to 1. It is only inserted when more than one non-blank variables are given. The key's default value is $\backslash\mathop{\!}\!$.	
<code>sep-ord-inf</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$	$\backslash\mathop{\!}\!$
	They key sets the math space that is inserted between a variable and the following infinitesimal. The space is only inserted when the variable's order <i>is different</i> from 1. It is only inserted when more than one non-blank variables are given. The key's default value is $\backslash\mathop{\!}\!$.	
<code>sep-var-var</code>	$\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$,
<u>New: v1.0</u>	They key sets the math space that is inserted between two variables. The space is only inserted when the <code>style-var=single</code> is used. and when more than one non-blank variables are given. The key's default value is a comma.	

`sep-eval-sb` $\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$ 0

This key sets the math space that is inserted in the evaluation subscript left to the point of evaluation. The space is only inserted when a non-blank subscript is given. The key's default value is 0 mu.

`sep-eval-sp` $\langle cs-number \rangle$, $\{ \langle mspace \rangle \}$, $\{ \langle delimiter \rangle \}$ 0

This key sets the math space that is inserted in the evaluation superscript left to the point of evaluation. The space is only inserted when a non-blank superscript is given. The key's default value is 0 mu.

Switches

`switch-*` `true`, `false` `false`

The effect of the star argument's presence can be switched with the value `true`. That is, the function is typeset next to the fraction when the star is absent and in the numerator when the star is present. As an example, compare below where the option is turned on (`true`) and off (`false`)

$$\begin{aligned} \backslash pdv[switch-*=false]\{y\}\{x\} &\implies \frac{\partial y}{\partial x} \\ \backslash pdv[switch-*=true]\{y\}\{x\} &\implies \frac{\partial}{\partial x} y \end{aligned}$$

respectively. The key's default value is false.

`switch-/` `true`, `false` `false`

The effect of the slash argument's presence can be switched with the value `true`. That is, the derivative is typeset with the fraction set by `style-frac-/` when the slash is absent and with the fraction set by `style-frac` when the slash is present. As an example, compare below where the option is turned on (`true`) and off (`false`)

$$\begin{aligned} \backslash pdv[switch-/=false]\{y\}\{x\} &\implies \frac{\partial y}{\partial x} \\ \backslash pdv[switch-/=true]\{y\}\{x\} &\implies \partial y / \partial x \end{aligned}$$

respectively. The key's default value is false.

Sort

The keys given in this subsection will be briefly described here and a in-depth description will be given in [section 5](#).

`sort-method` `abs`, `number`, `sign`, `symbol` `sign`, `symbol`, `abs`

Updated: v1.0

This key sets the sorting method behind the mixed order using build-in algorithms. The symbolic terms in the mixed order are sorted by their sign, symbolic length, numerical value and absolute value by the `sign`, `symbol`, `number` and `abs` algorithms, respectively. The key takes its input as a comma-separated list of values, accepting up to three values e.g.

$$\begin{aligned} \backslash\text{pdv}[\text{sort-method}=\text{sign}, \text{order}=\{c+kn, -b+2a\}]\{f\}\{x,y\} &\implies \frac{\partial^{c+kn+2a-b} f}{\partial x^{c+kn} \partial y^{-b+2a}} \\ \backslash\text{pdv}[\text{sort-method}=\{\text{sign}, \text{symbol}\}, \text{order}=\{c+kn, -b+2a\}]\{f\}\{x,y\} &\implies \frac{\partial^{kn+c+2a-b} f}{\partial x^{c+kn} \partial y^{-b+2a}} \\ \backslash\text{pdv}[\text{sort-method}=\{\text{sign}, \text{symbol}, \text{abs}\}, \text{order}=\{c+kn, -b+2a\}]\{f\}\{x,y\} &\implies \frac{\partial^{kn+2a+c-b} f}{\partial x^{c+kn} \partial y^{-b+2a}} \end{aligned}$$

are all valid inputs. Note how the terms are ordered differently with different methods. This key is further described in [section 5.1](#). The key's default value uses the three algorithms `sign`, `symbol`, `abs`.

sort-numerical `auto, first, last, symbolic` `auto`

Updated: v1.0

This key determines where the numerical term² is placed in the mixed order. The numerical term will *always* be placed as the first with `first` and as the last term with `last` in the mixed order. While `auto` will automatically determine where to place the numerical term depending on the sign of the first symbolic term. This is shown below

$$\begin{aligned} \backslash\text{pdv}[\text{sort-numerical}=\text{first}, \text{order}=\{n, 2\}]\{f\}\{x,y\} &\implies \frac{\partial^{2+n} f}{\partial x^n \partial y^2} \\ \backslash\text{pdv}[\text{sort-numerical}=\text{last}, \text{order}=\{-n, 2\}]\{f\}\{x,y\} &\implies \frac{\partial^{-n+2} f}{\partial x^{-n} \partial y^2} \end{aligned}$$

where the value `auto` would have placed the numerical term opposite in both equations. The value `symbolic` treats the numerical term as a symbolic term with symbolic length equal zero and is thus sorted alongside with all other terms

$$\backslash\text{pdv}[\text{sort-numerical}=\text{symbolic}, \text{order}=\{2+n, -a\}]\{f\}\{x,y\} \implies \frac{\partial^{n+2-a} f}{\partial x^{2+n} \partial y^{-a}}$$

The key is further described in [section 5.2](#), and its default value is `auto`.

`sort-abs-reverse` `true, false` `false`

The sorting algorithm `abs` sorts the terms by their absolute value in a descending manner. If the value `true` is used, then the terms will be sorted in an ascending manner instead. See [section 5.3](#) for more information.

sort-number-reverse `true, false` `false`

New: v1.0

The sorting algorithm `number` sorts the terms by their numerical value in a descending manner. If the value `true` is used, then the terms will be sorted in an ascending manner instead. See [section 5.3](#) for more information.

`sort-sign-reverse` `true, false` `false`

The sorting algorithm `sign` sorts the terms by their numerical sign where the positive terms are placed before the negative terms. The value `true` will reverse this ordering

²The numerical term is the sum of all the orders that does not contain any symbols but solely consist of numbers.

such that the negative terms are placed before the positive terms. See [section 5.3](#) for more information.

`sort-symbol-reverse` true, false false

The sorting algorithm `symbol` sorts the terms by their symbolic length in a descending manner. If the value `true` is used, then the terms will sorted in an ascending manner instead. See [section 5.3](#) for more information.

Miscellaneous

`fun` true, false false

New: v1.0 Adds or remove delimiters around the function.

$$\begin{aligned} \backslash\text{pdv}[\text{fun}=\text{false}]\{f\}\{x,y,z\} &\implies \frac{\partial^3 f}{\partial x \partial y \partial z} \\ \backslash\text{pdv}[\text{fun}=\text{true}]\{f\}\{x,y,z\} &\implies \frac{\partial^3(f)}{\partial x \partial y \partial z} \\ \backslash\text{pdv}[\text{fun}]\{f\}\{x,y,z\} &\implies \frac{\partial^3(f)}{\partial x \partial y \partial z} \end{aligned}$$

Without setting the option to a value is equivalent to setting it to `true` as seen above.

`frac` true, false false

New: v1.0 Adds or remove delimiters around the fractional part of the derivative.

$$\begin{aligned} \backslash\text{pdv}[\text{frac}=\text{false}]\{f\}\{x,y,z\} &\implies \frac{\partial^3 f}{\partial x \partial y \partial z} \\ \backslash\text{pdv}[\text{frac}=\text{true}]\{f\}\{x,y,z\} &\implies \left(\frac{\partial^3 f}{\partial x \partial y \partial z} \right) \\ \backslash\text{pdv}^*[\text{frac}=\text{true}]\{f\}\{x,y,z\} &\implies \left(\frac{\partial^3}{\partial x \partial y \partial z} \right) f \\ \backslash\text{pdv}[\text{frac}]\{f\}\{x,y,z\} &\implies \left(\frac{\partial^3 f}{\partial x \partial y \partial z} \right) \end{aligned}$$

Without setting the option to a value is equivalent to setting it to `true` as seen above.

`var` none, all, $\langle cs\text{-numbers} \rangle$ none

New: v1.0 Adds or remove delimiters around the variable. The value `all` adds delimiters around all variables and `none` removed all delimiters. However, if only some variables needs delimiters then $\langle cs\text{-numbers} \rangle$ can be used e.g. `var={1,3}` adds delimiters around the first and third variable

$$\begin{aligned} \backslash\text{pdv}[\text{var}=\text{none}]\{f\}\{x,y,z,t\} &\implies \frac{\partial^4 f}{\partial x \partial y \partial z \partial t} \\ \backslash\text{pdv}[\text{var}=\{1,3\}]\{f\}\{x,y,z,t\} &\implies \frac{\partial^4 f}{\partial(x) \partial y \partial(z) \partial t} \\ \backslash\text{pdv}[\text{var}=\text{all}]\{f\}\{x,y,z,t\} &\implies \frac{\partial^4 f}{\partial(x) \partial(y) \partial(z) \partial(t)} \\ \backslash\text{pdv}[\text{var}]\{f\}\{x,y,z,t\} &\implies \frac{\partial^4 f}{\partial(x) \partial(y) \partial(z) \partial(t)} \end{aligned}$$

Without setting the option to a value is equivalent to setting it to `all` as seen above.

`order` $\langle\{cs-orders\}\rangle$ 1

`ord` Sets the order of differentiation for each variable as a comma separated list of values.

New: v1.0

`mixed-order` $\langle\{mixed\ order\}\rangle$ 1

`mixord` The mixed order is automatically calculated from the orders of differentiation set by `order=\langle\{orders\}\rangle`. In cases where it fails or another form is preferred, the calculated mixed order can be overwritten with this option.

New: v1.0

3.4 Differential options

The options in this subsection are available for differentials defined by the package and you with the commands in [section 4.2](#). All options in this section were introduced in v1.0.

Style

`style-inf` $\langle\{math-font-style\}\langle\{infinitesimal}\rangle$ d

Updated: v1.0 Sets the differential's infinitesimal including font style. This change was necessary to allow multiple font styles, e.g. `style-inf=\mathbf{\mathrm{d}}`. The default infinitesimal is a plain 'd' with no associated font style.

`style-notation` `single`, `multiple`, `mixed` `multiple`

Sets the differential's notation when the star argument is absent (given `switch-*=false`). The key's default value is `multiple`. The value `single` creates a single differential with the variables and orders in a subscript and superscript, respectively. The value `multiple` creates a differentials for each variable. The value `mixed` creates multiple differentials with the variables and orders in a subscript and superscript, respectively.

$$\begin{aligned} \backslash\text{odif}[\text{style-notation}=\text{multiple}, \text{order}=\{n,1,2\}]\{x,y,z,t\} &\implies d^n x dy d^2 z dt \\ \backslash\text{odif}[\text{style-notation}=\text{single}, \text{order}=\{n,1,2\}]\{x,y,z,t\} &\implies d_{x,y,z,t}^{n,1,2} \\ \backslash\text{odif}[\text{style-notation}=\text{mixed}, \text{order}=\{n,1,2\}]\{x,y,z,t\} &\implies d_x^n d_y d_z^2 d_t \end{aligned}$$

`style-notation-*` `single, multiple, mixed` `single`
 Sets the differential's notation when the star argument is present (given `switch-*=false`).
 The key's default value is `single`.

Scaling

`scale-var` `auto, none, big, Big, bigg, Bigg` `auto`
 Sets the size of the delimiters used around the variable when the star argument is
 absent (given `switch-*=false`). The scaling is used whenever `var=true` is used. The
 key's default value is set to scale the delimiters automatically.

`scale-var-*` `auto, none, big, Big, bigg, Bigg` `auto`
 Sets the size of the delimiters used around the variable when the star argument is
 present (given `switch-*=false`). The scaling is used whenever `var=true` is used. The
 key's default value is set to scale the delimiters automatically.

Delimiters

`delims-var` `<left delimiter><right delimiter>` `()`
 Sets the left and right delimiters used around the variable when the star argument
 is absent (given `switch-*=false`). The delimiters are inserted when `var=true` is used.
 The key's default delimiters are a left and right parenthesis, respectively.

`delims-var-*` `<left delimiter><right delimiter>` `()`
 Sets the left and right delimiters used around the variable when the star argument
 is present (given `switch-*=false`). The delimiters are inserted when `var=true` is used.
 The key's default delimiters are a left and right parenthesis, respectively.

Math spacing

The options in this subsection inserts extra horizontal math space. Below, illustrates
 where the space is inserted for each option

$$\begin{array}{cc} \partial_{(\text{inf-var})x(\text{var-inf})} \partial y & (\text{begin}) \partial x (\text{end}) \\ \partial^{(\text{inf-ord})2}_{(\text{ord-var})} x & \partial_{x(\text{var-var})y}^{2(\text{ord-ord})2} \end{array}$$

where it have been split into four to give a better overview.

`sep-begin` `<cs-number>, {\mspace}, {\delimiter}` `\mathop{} \!`
 Sets the math space that is inserted to the left of the first differential. The key's default
 value is `\mathop{} \!`.

`sep-inf-ord` `<cs-number>, {\mspace}, {\delimiter}` `0`
 This key sets the math space that is inserted in the infinitesimal's power left to the
 order. It is only inserted when the mixed order is different from 1. The key's default
 value is `0 mu`.

<code>sep-inf-var</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	0
	This key sets the math space that is inserted between the infinitesimal and the following variable. The space is only inserted when the order <i>is equal</i> to 1 when a non-blank variable is given. The key's default value is 0 mu.	
<code>sep-ord-var</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	0
	This key sets the math space that is inserted between the infinitesimal's power and the following variable. The space is only inserted when the order <i>is different</i> from 1 and a non-blank variable is given. The key's default value is 0 mu.	
<code>sep-var-inf</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	<code>\mathop{\}\!</code>
	They key sets the math space that is inserted between a variable and the following infinitesimal. It is only inserted when more than one non-blank variables are given. The key's default value is <code>\mathop{\}\!</code> .	
<code>sep-var-var</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	,
	Sets the math space that is inserted between two variables. The space is only inserted when <code>style-var=single</code> and more than one non-blank variables are given. The key's default value is a comma.	
<code>sep-ord-ord</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	,
	Sets the math space that is inserted between two orders. The space is only inserted when <code>style-var=single</code> and more than one non-blank orders are given. The key's default value is a comma.	
<code>sep-end</code>	<code><cs-number>, {\mspace}, {\delimiter}</code>	0
	Sets the math space that is inserted to the right of the last differential. The key's default value is 0.	

Switches

<code>switch-*</code>	<code>true, false</code>	<code>false</code>
	The effect of the star argument's presence can be switched with the value <code>true</code> . As an example, compare below where the option is turned on (<code>true</code>) and off (<code>false</code>),	

$$\begin{aligned} \text{\color{red}\code{\odif[switch-*=false]\{x,y,z,t\}} } &\implies dx dy dz dt \\ \text{\color{red}\code{\odif[switch-*=true]\{x,y,z,t\}} } &\implies d_{x,y,z,t} \end{aligned}$$

The key's default value is false.

Miscellaneous

<code>var</code>	<code>none, all, <cs-numbers></code>	<code>none</code>
	Adds or remove delimiters around the variable. The value <code>all</code> adds delimiters around all variables and <code>none</code> removed all delimiters. However, if only some variables needs delimiters then <code><cs-numbers></code> can be used e.g. <code>var={1,3}</code> adds delimiters around the first and third variable	

`\odif[var=none]{x,y,z,t}` \Rightarrow $dx dy dz dt$
`\odif[var={1,3}]{x,y,z,t}` \Rightarrow $d(x) dy d(z) dt$
`\odif[var=all]{x,y,z,t}` \Rightarrow $d(x) d(y) d(z) d(t)$
`\odif[var]{x,y,z,t}` \Rightarrow $d(x) d(y) d(z) d(t)$

Without setting the option to a value is equivalent to setting it to `all` as seen above.

`order` `{⟨cs-orders⟩}` 1
`ord` Sets the differentiation order of each variable.

3.5 All derivatives and differentials

The options in this subsection are applied to all derivatives and differential because some options should be consisting regardless of the derivative and differential. The options are accessed using `\derivset{all}[⟨key=value⟩]`.

`scale-auto` `leftright`, `mleftright`^R `leftright`, `mleftright`^R

Updated: v1.1

Sets the dynamic scaling methods of delimiters. The value `leftright` sets the method to use `\left` and `\right`. The value `mleftright` requires the `mleftright` package and sets the method to use `\mleft` and `\mright` instead. The default value is `leftright` unless `mleftright` have been loaded then it is `mleftright`.

4 Defining variants

This section goes into detail with how to define derivative and differential variants based on the package's internal commands, as mentioned earlier. The derivative package provides a $\LaTeX 2_{\epsilon}$ way of defining the derivatives.

4.1 Derivative variant

`\NewDerivative` $\langle control-sequence \rangle \{ \langle infinitesimal \rangle \} [\langle key=value \rangle]$

This family of commands are used to define a derivative variant with the macro name $\langle control-sequence \rangle$. Moreover, the new derivative will use $\{ \langle infinitesimal \rangle \}$ as its infinitesimal and inherit the package's default settings given in [section 3.3](#), but they may be overwritten with $[\langle key=value \rangle]$. The difference between them is as follows:

- `\NewDerivative` is similar to `\NewDocumentCommand` of the `xparse` package and will issue an error if $\langle control-sequence \rangle$ has already been defined.
- `\RenewDerivative` is similar to `\RenewDocumentCommand` of the `xparse` package and will issue an error if $\langle control-sequence \rangle$ has not previously been defined.
- `\ProvideDerivative` is similar to `\ProvideDocumentCommand` of the `xparse` package and will define $\langle control-sequence \rangle$ if it does not have an existing definition. It will not issue any errors.
- `\DeclareDerivative` is similar to `\DeclareDocumentCommand` of the `xparse` package and will always define the $\langle control-sequence \rangle$ with the new definition regardless of whether it already exists.

$\langle control-sequence \rangle$ The first argument is the macro name of the derivative that is to be defined.

$\{ \langle infinitesimal \rangle \}$ The second argument makes the derivative $\langle control-sequence \rangle$ use $\{ \langle infinitesimal \rangle \}$ as its infinitesimal. It is equivalent to setting `style-inf={\langle infinitesimal \rangle}`.

$[\langle key=value \rangle]$ The optional argument accepts its input as a comma-separated list of $key=value$ pairs, which overrides the package's default options for the keys given. The derivative $\langle control-sequence \rangle$ will use the package's default settings if the argument is left out.

4.2 Differential variant

`\NewDifferential` $\langle control-sequence \rangle \{ \langle infinitesimal \rangle \} [\langle key=value \rangle]$

This family of commands are used to define a differential variant with the macro name $\langle control-sequence \rangle$. Moreover, the new derivative will use $\{ \langle infinitesimal \rangle \}$ as its infinitesimal and inherit the package's default options given in [section 3.4](#), but they may be overwritten with $[\langle key=value \rangle]$. The difference between them is as follows:

- `\NewDifferential` is similar to `\NewDocumentCommand` of the `xparse` package and will issue an error if $\langle control-sequence \rangle$ has already been defined.

- `\RenewDifferential` is similar to `\RenewDocumentCommand` of the `xparse` package and will issue an error if $\langle control-sequence \rangle$ has not previously been defined.
- `\ProvideDifferential` is similar to `\ProvideDocumentCommand` of the `xparse` package and will define $\langle control-sequence \rangle$ if it does not have an existing definition. It will not issue any errors.
- `\DeclareDifferential` is similar to `\DeclareDocumentCommand` of the `xparse` package and will always define the $\langle control-sequence \rangle$ with the new definition regardless of whether it already exists.

$\langle control-sequence \rangle$	The first argument is the macro name of the differential that is to be defined.
$\{ \langle infinitesimal \rangle \}$	While the second argument makes the differential $\langle control-sequence \rangle$ use $\{ \langle infinitesimal \rangle \}$ as its infinitesimal. It is equivalent to setting <code>style-inf=$\{ \langle infinitesimal \rangle \}$</code> .
$[(key=value)]$	The optional argument accepts its input as a comma-separated list of <i>key=value</i> pairs, which overrides the package's default options for the keys given. The derivative $\langle control-sequence \rangle$ will use the package's default settings if the argument is left out.

5 The mixed order

5.1 Sorting algorithms

A unique feature of this package is that the sorting method behind the mixed order may be changed using built-in algorithms and there are currently four available: `abs`, `number`, `sign` and `symbol`. The algorithms have been chosen such that it the terms in the mixed order may be arranged in many ways with as few algorithms as possible to suit your liking. A sorting method may consist of up to three algorithms which are applied in layers.

- The sorting algorithm `sign` separates the positive and negative terms, and puts the positive terms before the negative terms in the mixed order. The option `sort-sign-reverse=true` reverses ordering is reversed such that the negative terms appear before the positive terms.
- The sorting algorithm `symbol` separates the terms by their symbolic length, where the terms are sorted in a descending manner according to their length. The option `sort-symbol-reverse=true` sorts the terms in a ascending manner instead.
- The sorting by `abs` separates the terms by their absolute value, where the terms are ordered in a descending manner. The option `sort-abs-reverse=true` sorts the terms in an ascending manner instead.
- The sorting by `number` separates the terms by their numerical value, where the terms are ordered in a descending manner. The option `sort-number-reverse=true` sort the terms in an ascending manner instead.

When using the option `sort-method={...}`, the number of algorithms determines the number of layers e.g. `sort-method={sign,symbol}` is a sorting method with two layers, where the algorithms `sign` and `symbol` are applied in layer 1 and layer 2, respectively. E.i the `sign` algorithm splits the terms into two groups, one with positive terms and one with negative terms. Then the `symbol` algorithm will sorts the terms within each groups according to the terms symbolic length. The mixed order is then formed by combining these two groups.

5.1.1 Examples

The examples below are uses partial derivatives with `order={3a-3hh-2b, 4c+4gg+2ff, -5d-5ee}` and a close up view of the mixed order. Square brackets are used to indicate grouping of terms and the text below a group refers to the algorithm applied. Here the words positive and negative, long and short, and big and low refers to the `sign`, `symbol`, and `abs` algorithms, respectively. A row of square brackets represents a layer.

The packages default sorting method is `sort-method={sign,symbol,abs}` which is used below. The `sign` algorithm separate the positive and negative terms in the first layer. While in the second layer, the terms are separated by symbol length etc.

$$\frac{\partial^{4gg+2ff+4c+3a-5ee-3hh-5d-2b} f}{\partial x^{3a-3hh-2b} \partial y^{4c+4gg+2ff} \partial z^{-5d-5ee}}$$

$$\frac{\underbrace{\underbrace{4gg + 2ff}_{\substack{\text{big} \\ \text{low}}} + \underbrace{4c + 3a}_{\substack{\text{big} \\ \text{low}}}}_{\text{long}} - \underbrace{\underbrace{5ee - 3hh}_{\substack{\text{big} \\ \text{low}}} - \underbrace{5d - 2b}_{\substack{\text{big} \\ \text{low}}}}_{\text{long}}}{\text{positive} \quad \text{negative}}$$

as expected. Interchanging **sign** and **symbol** from the previous example i.e **sort-method={symbol,sign,abs}**, then the **symbol** and **sign** algorithms are now applied in the first and second layer, respectively. The result is shown below

$$\frac{\partial^{4gg+2ff-5ee-3hh+4c+3a-5d-2b} f}{\partial x^{3a-3hh-2b} \partial y^{4c+4gg+2ff} \partial z^{-5d-5ee}}$$

$$\frac{\underbrace{\underbrace{4gg + 2ff}_{\substack{\text{big} \\ \text{low}}} - \underbrace{5ee - 3hh}_{\substack{\text{big} \\ \text{low}}}}_{\text{positive}} + \underbrace{\underbrace{4c + 3a}_{\substack{\text{big} \\ \text{low}}} - \underbrace{5d - 2b}_{\substack{\text{big} \\ \text{low}}}}_{\text{negative}}}{\text{long} \quad \text{short}}$$

where a clear difference is seen between the above two examples.

The above two examples used 3-layer sorting methods but as mentioned earlier, the sorting method can also be constructed with one and two algorithms i.e. 1-layer and 2-layer sorting methods. A sorting method with two layers could be constructed as **sort-method={sign,symbol}** and the terms are ordered by sign and then symbolic length. In this case, there is not a third layer

$$\frac{\partial^{4gg+2ff+3a+4c-3hh-5ee-2b-5d} f}{\partial x^{3a-3hh-2b} \partial y^{4c+4gg+2ff} \partial z^{-5d-5ee}}$$

$$\frac{\underbrace{\underbrace{4gg + 2ff}_{\text{long}} + \underbrace{3a + 4c}_{\text{short}}}_{\text{positive}} - \underbrace{\underbrace{3hh - 5ee}_{\text{long}} - \underbrace{2b - 5d}_{\text{short}}}_{\text{negative}}}{\text{positive} \quad \text{negative}}$$

The last example shows a 1-layer sorting method given as **sort-method=symbol** which gives the result

$$\frac{\partial^{-3hh+4gg+2ff-5ee+3a-2b+4c-5d} f}{\partial x^{3a-3hh-2b} \partial y^{4c+4gg+2ff} \partial z^{-5d-5ee}}$$

$$\underbrace{-3hh + 4gg + 2ff - 5ee}_{\text{long}} + \underbrace{3a - 2b + 4c - 5d}_{\text{short}}$$

as expected.

5.2 The numerical term

With the symbolic part of the mixed order treated, it is time to look at the numerical term² which have a symbolic of zero. For this reason it is treated differently than the symbolic terms.

The numerical term can be placed either at the beginning or at the end of the mixed order with the values **first** and **last**, respectively. The package can also determine where to place the numerical term with the value **auto**. Compare below

$$\text{sort-numerical=auto:} \quad \frac{\partial^{\beta+2} f}{\partial x^{\beta} \partial y^2} \quad \frac{\partial^{\beta-2} f}{\partial x^{\beta} \partial y^{-2}} \quad \frac{\partial^{2-\beta} f}{\partial x^{-\beta} \partial y^2} \quad \frac{\partial^{-2-\beta} f}{\partial x^{-\beta} \partial y^{-2}}$$

<code>sort-numerical=first:</code>	$\frac{\partial^{2+\beta} f}{\partial x^\beta \partial y^2}$	$\frac{\partial^{-2+\beta} f}{\partial x^\beta \partial y^{-2}}$	$\frac{\partial^{2-\beta} f}{\partial x^{-\beta} \partial y^2}$	$\frac{\partial^{-2-\beta} f}{\partial x^{-\beta} \partial y^{-2}}$
<code>sort-numerical=last:</code>	$\frac{\partial^{\beta+2} f}{\partial x^\beta \partial y^2}$	$\frac{\partial^{\beta-2} f}{\partial x^\beta \partial y^{-2}}$	$\frac{\partial^{-\beta+2} f}{\partial x^{-\beta} \partial y^2}$	$\frac{\partial^{-\beta-2} f}{\partial x^{-\beta} \partial y^{-2}}$
<code>sort-numerical=symbolic:</code>	$\frac{\partial^{\beta+2} f}{\partial x^\beta \partial y^2}$	$\frac{\partial^{\beta-2} f}{\partial x^\beta \partial y^{-2}}$	$\frac{\partial^{2-\beta} f}{\partial x^{-\beta} \partial y^2}$	$\frac{\partial^{-\beta-2} f}{\partial x^{-\beta} \partial y^{-2}}$

These three options should satisfy every needs.

5.3 Reversing the sort algorithm

The reverse keys serves to reverse the ordering of the terms by the sorting algorithms. This allows for even greater flexibility over the sorting method. So if the default ordering of an algorithm is not wished it may be reversed with the corresponding reverse key. For example, the `sign` algorithm places the positive terms before the negative terms, but is opposite with `sort-sign-reverse=true`. The reverse keys in action are shown below (`sort-method=(1 algorithm only)` is also used to better show off the reverse functionality)

<code>sort-abs-reverse=false:</code>	$\frac{\partial^{-3b+2a} f}{\partial x^{2a} \partial y^{-3b}}$	<code>sort-abs-reverse=true:</code>	$\frac{\partial^{2a-3b} f}{\partial x^{2a} \partial y^{-3b}}$
<code>sort-number-reverse=false:</code>	$\frac{\partial^{2a-3b} f}{\partial x^{2a} \partial y^{-3b}}$	<code>sort-number-reverse=true:</code>	$\frac{\partial^{-3b+2a} f}{\partial x^{2a} \partial y^{-3b}}$
<code>sort-sign-reverse=false:</code>	$\frac{\partial^{a-b} f}{\partial x^a \partial y^{-b}}$	<code>sort-sign-reverse=true:</code>	$\frac{\partial^{-b+a} f}{\partial x^a \partial y^{-b}}$
<code>sort-symbol-reverse=false:</code>	$\frac{\partial^{ab+c} f}{\partial x^{ab} \partial y^c}$	<code>sort-symbol-reverse=true:</code>	$\frac{\partial^{c+ab} f}{\partial x^{ab} \partial y^c}$

5.4 Mixed order override

In cases where the mixed order fails to be calculated or another form is preferred, then the mixed order override can be used

$$\backslash pdv[\text{order}=\{n+3k, n-k\}]\{f\}\{x,y\} \implies \frac{\partial^{2n+2k} f}{\partial x^{n+3k} \partial y^{n-k}}$$

$$\backslash pdv[\text{order}=\{n+3k, n-k\}, \text{mixed-order}=\{2(n+k)\}]\{f\}\{x,y\} \implies \frac{\partial^{2(n+k)} f}{\partial x^{n+3k} \partial y^{n-k}}$$

6 Miscellaneous

6.1 Slashfrac

`\slashfrac` [*scale*]{*numerator*}{*denominator*}

A text-styled fraction i.e. a/b , is commonly used in text-mode. While `\slashfrac{a}{b}` is certainly longer to write than `a/b`, which only requires three keystrokes, but for the implementation of text-styled derivatives a macro for it is needed.

[*scale*] The optional argument is the scaling parameter for the slash and takes the inputs as shown below. Leaving out [*scale*] sets the scaling parameter to `auto`

`\slashfrac[auto]{y_f}{x}` \Rightarrow y_f/x

`\slashfrac[none]{y_f}{x}` \Rightarrow y_f/x

`\slashfrac[big]{y_f}{x}` \Rightarrow y_f/x

`\slashfrac[Big]{y_f}{x}` \Rightarrow y_f/x

`\slashfrac[bigg]{y_f}{x}` \Rightarrow y_f/x

`\slashfrac[Bigg]{y_f}{x}` \Rightarrow y_f/x

This argument is subject to change, see [consideration 7.2\(ii\)](#) for more information.

{*numerator*} Typesets the fraction's numerator.

{*denominator*} Typesets the fraction's denominator.

7 To do

Entries marked with ✓ will be implemented in a future release specified in the entry. Entries marked with ✗ will not be implemented.

7.1 Future implementation

The list describes what *will be added* to the package in a later release.

- (i) Implement the option to turn off the sorting of the terms in the mixed order. This could be made as a `draft` / `final` option.
✓✗**Conclusion:** *There could be an draft/final option for the package. But what is should turn off/on is harder to say.*

7.2 Future considerations

The list describes what *might be changed or added* to the package in a later release.

- (i) It can become rather cumbersome to read a derivative with a very long function. To ease this up, I am considering to add a switch that exchange the variable and the function arguments. But this might be against the L^AT_EX way of using arguments, which is the reason it have not been done and I would like feedback on it.
- (ii) I am considering making the optional argument of `\slashfrac` into a `[(key=value)]` argument with something like `scale={scaling}`.
✓**Conclusion:** *I want an key-value implementation like this in the future, but it will not be added in the next release.*

Index

Numbers in bold refer to the page where the entry is defined.

Index of Options

delims

delims-eval, **12**
delims-eval-/, **13**
delims-frac, **13**
delims-frac-/, **13**
delims-fun, **13**
delims-var, **13, 19**
delims-var-*, **19**

italic

italic, **6, 8, 9, 11, 11**

misc

frac, **12, 13, 17**
fun, **12, 13, 17**
mixed-order, **18**
mixord, **18**
ord, **18, 21**
order, **4, 12, 18, 18, 21, 24**
var, **12, 13, 17, 17, 19, 20, 20**

scale

scale-auto, **21**
scale-eval, **12**
scale-eval-/, **12**
scale-frac, **12**
scale-frac-/, **12**
scale-fun, **12**
scale-var, **12, 19**
scale-var-*, **19**

sep

sep-begin, **19**
sep-end, **20**
sep-eval-sb, **15**

sep-eval-sp, **15**
sep-frac-fun, **14**
sep-inf-fun, **14**
sep-inf-ord, **14, 19**
sep-inf-var, **14, 20**
sep-ord-fun, **14**
sep-ord-inf, **14**
sep-ord-ord, **20**
sep-ord-var, **20**
sep-var-inf, **14, 20**
sep-var-ord, **14**
sep-var-var, **14, 20**

sort

sort-abs-reverse, **16, 24**
sort-method, **15, 24, 25**
sort-number-reverse, **16, 24**
sort-numerical, **16**
sort-sign-reverse, **16, 24, 26**
sort-symbol-reverse, **17, 24**

style

style-frac, **11, 15**
style-frac-/, **11, 15**
style-inf, **6, 11, 11, 18, 18, 22, 23**
style-notation, **18**
style-notation-*, **19**
style-var, **12, 13, 14, 20**

switch

switch-*, **4, 8, 15, 18, 19, 20**
switch-/, **5, 11-13, 15**

upright

upright, **6, 8, 9, 11, 11**

Index of Commands

A

`\adif`, 8, 9

`\adv`, 6, 6

D

`\DeclareDerivative`, 22, 22

`\DeclareDifferential`, 22, 23

`\derivset`, 10, 10, 21

F

`\fdif`, 8, 9

`\fdv`, 6, 6

J

`\jdv`, 6, 7

M

`\mdif`, 8, 9

`\mdv`, 6, 6

N

`\NewDerivative`, 22, 22

`\NewDifferential`, 22, 22

O

`\odif`, 8, 8

`\odv`, 6, 6, 10

P

`\pdif`, 8, 8

`\pdv`, 4, 4

`\ProvideDerivative`, 22, 22

`\ProvideDifferential`, 22, 23

R

`\RenewDerivative`, 22, 22

`\RenewDifferential`, 22, 23

S

`\slashfrac`, 27, 27

Change history

<hr/> <p>v0.9 2019-07-21 Beta</p> <hr/>	<ul style="list-style-type: none">• First release of the package. The package is currently in a beta version.
<hr/> <p>v0.95 2019-09-18 Beta</p> <hr/>	<ul style="list-style-type: none">• <i>Please ignore this version, since it contained the wrong .sty and .pdf files :(.</i>• Removed the single token restriction of the infinitesimal since it made no sense.• Fixed the documentation errors and typos.• Minor fixes to the code.
<hr/> <p>v0.95b 2019-09-21 Beta</p> <hr/>	<ul style="list-style-type: none">• Contains the correct .sty and .pdf files :).• One minor code fix.
<hr/> <p>v0.96 2019-12-22 Beta</p> <hr/>	<ul style="list-style-type: none">• Fixed the double superscript issue for higher order derivative when the variable contained a superscript.
<hr/> <p>v0.97 2020-02-03 Beta</p> <hr/>	<ul style="list-style-type: none">• Fixed the argument specifier of <code>__deriv_scale_big:nnnn</code> when it was used (it was used with <code>:nnnm</code>).
<hr/> <p>v0.98 2020-07-20 Beta</p> <hr/>	<ul style="list-style-type: none">• Fixed a bug concerning the subscript argument due to recent changes to the <code>xparse</code> package dated 2020-05-14 (the fix works with earlier versions of <code>xparse</code> as well).
<hr/> <p>v1.0 2021-05-25</p> <hr/>	<ul style="list-style-type: none">• Added new options for derivatives• Added new values for existing options• Added new commands to write differentials• Changed usage of <code>style-inf</code>• Changed the order argument to an option argument• Changed default values for some options• Changed math space keys to accept more general inputs• Changed <code>\derivset</code> to define default options• Merged the codes for ordinary and partial derivatives• Removed the mixed order argument• Removed the options <code>misc-add-delims</code> and <code>misc-remove-delims</code>

