`diffcoeff`
a LaTeX package to ease the writing of differential coefficients in all their variety
Version 2

Andrew Parsloe
(ajparsloe@gmail.com)

December 4, 2018

Abstract

`diffcoeff.sty` allows the easy and consistent writing of ordinary, partial and other derivatives of arbitrary (algebraic or numeric) order. For mixed partial derivatives, the total order of differentiation is calculated by the package. Optional arguments allow specification of points of evaluation (ordinary derivatives), or variables held constant (partial derivatives), and the placement of the differentiand (numerator or appended). Version 2 is built on `xtemplate,` allowing the systematic fine-tuning of the display and generation and use of variant forms (like derivatives built from D, Δ or δ). The package requires the LaTeX3 bundles `l3kernel` and `l3packages`.

Contents

1 Introduction

The LaTeX package `diffcoeff.sty` is written in the expl3 language of LaTeX3 and requires the bundles `l3kernel` and `l3packages` (the latter for the `xparse`, `l3keys2e` and `xtemplate` packages). The package is invoked in the usual way by entering

```
\usepackage{diffcoeff}
```

in the preamble of your document. There are two package options. The first is a switch, `ISO`, which turns on formatting conforming to ISO recommendations:

```
\usepackage[ISO]{diffcoeff}
```

The effect of this is discussed in Section 5. The second is a filename for a file containing definitions of variant forms of derivative:

```
\usepackage[def-file=<filename>]{diffcoeff}
```

This is discussed in Subsection 5.6. Of course both options can be used in the same call if desired:

```
\usepackage[ISO,def-file=<filename>]{diffcoeff}
```

For the present document, the call was

```
\usepackage[def-file=diffcoeff-doc]{diffcoeff}
```

with the file `diffcoeff-doc.def` in the same directory as `diffcoeff.tex`.

1.1 Version comparison

The present document discusses *version 2* of the `diffcoeff` package. Unlike version 1, version 2 is built on the the `xtemplate` package (included in the `l3packages` bundle) which makes certain facilities available which it would be silly not to exploit. Hence the coding between the versions is completely different and there are consequences.

1. The `\diffset` command, formerly used to tweak the display of derivatives, has been superseded by the `\diffdef` command. `\diffset` now sends a message warning of its obsolescence to the terminal and LaTeX log but is otherwise functionless. It should not interfere with the compilation of a document but any intended fine-tuning of the display by means of the `\diffset` command will not eventuate. The warning message is: `Obsolete command: \diffset has been superseded by the \diffdef command. See the diffcoeff documentation for further information.` The `\diffdef` command is discussed in Subsection 5.4.

2. The optional trailing argument used to indicate a point of evaluation or variables held constant is now delimited by square brackets, `[` and `]`, as other optional arguments are. For compatibility with version 1, braces can still be used but their use to delimit an *optional* argument is now deprecated in `xparse` on which `diffcoeff` depends. Presumably at some stage this provision will be removed from `xparse`. For future-proofing documents use square brackets.

3. The commands `\Diff`, `\diffd` and `\Diffd` used to construct derivatives from D, δ and Δ in version 1, are still available in version 2, but deprecated. A new optional argument in the `\diff` command offers these and a host of other possibilities and is now the preferred method of forming such variants; see Subsection 5.5.2.

Note on terminology

I refer throughout to the quantity or function being differentiated as the *differentiand* (in line with *integrand*, *operand*, etc.).

2 A Rogues' Gallery of derivatives

Browsing through texts on statistical mechanics, relativity and classical mechanics I find the following choice examples of derivatives 'disporting every which way'.

Multi-character variables of differentiation un-parenthesized:

$$\frac{\partial \frac{\psi}{\Theta}}{\partial \frac{1}{\Theta}}, \quad \frac{\partial E/T}{\partial 1/T}, \quad \frac{\partial \ln f}{\partial \ln x_0}, \quad \frac{\partial^2 \psi}{\partial a_i \, \partial \frac{1}{\Theta}}, \quad \frac{\partial \mathcal{L}}{\partial \eta_{,i}^{(r)}} \tag{1}$$

Multi-character variables of differentiation parenthesized:

$$\frac{\partial H}{\partial \left(\dfrac{\partial S}{\partial q_k} \right)}, \quad \frac{\partial \varepsilon}{\partial (1/\Theta)}, \tag{2}$$

Higher-order derivatives where the parentheses do not or do include the operator:

$$\frac{\partial^2 q}{\partial (\frac{1}{\Theta})^2}, \quad \frac{\partial^2 q}{\partial (1/\Theta)^2}, \quad \frac{\partial^2 \varepsilon}{\partial (a_i)^2}, \quad \frac{d^2 \phi^i(x^i)}{(dx^i)^2}. \tag{3}$$

Should the d or ∂ be included within the parentheses, as in the last of (3), or not, as in the others? Logic says 'yes'; practice suggests (generally) 'no'.

Indicating a point of evaluation is similarly varied:

$$\left.\frac{\partial \phi}{\partial \varepsilon}\right|_{\varepsilon=\varepsilon_0}, \quad \left.\frac{\partial^2 \phi}{\partial \varepsilon^2}\right|_{\varepsilon=\varepsilon_0}, \quad \left[\frac{\partial b^\beta}{\partial a^\alpha}\right]_{b=0}, \quad \left(\frac{du}{dv}\right)_{v=0}. \tag{4}$$

ISO 80000-2 (item 2.11.13) favours the last of these – parentheses – for ordinary derivatives. Presumably, partial derivatives should follow suit, although parentheses are also used to indicate variables held constant:

$$\left(\frac{\partial}{\partial U}\frac{P}{T}\right)_V, \quad \left(\frac{\partial S}{\partial N_2}\right)_{U,V,N_1}, \quad (\partial S/\partial T)_V. \tag{5}$$

Other symbols besides d and ∂ are used to denote derivative-like quantities. From introductory calculus, classical mechanics and thermodynamics come δ and Δ, from fluid mechanics comes D:

$$\frac{\delta y}{\delta x}, \quad \frac{D\rho}{Dt}, \quad \left(\frac{\Delta U}{\Delta T}\right)_V, \quad \Delta U/\Delta T, \quad \frac{\delta \mathcal{L}}{\delta \eta^{(r)}}. \tag{6}$$

There are those, like the International Organization for Standardization (ISO), who stipulate (or prefer) an upright ‘d’ for their derivatives:

$$\frac{\mathrm{d}y}{\mathrm{d}x}. \tag{7}$$

When the differentiand is too big or awkward to sit in the numerator and is appended to the operator, the d or ∂ in the numerator is generally centred – but not always. In texts prior to the age of computerised typesetting one will sometimes find the symbol pushed to the *left*:

$$\frac{\partial}{\partial x^{l^*}}\left(\frac{\partial x^{i^*}}{\partial x^{k^*}}\right), \quad \frac{d}{dt}\left(\frac{m\mathbf{q}_x}{\sqrt{1-q^2}}\right). \tag{8}$$

The observant will note an italic adjustment with the first expression, so that the ∂ in the numerator and the ∂ in the denominator line up in a slanting column, but no such adjustment for the d-s in the second derivative.

And finally, the operator in the numerator may differ from that in the denominator. For instance, in tensor calculus acceleration is sometimes written as

$$\frac{\nabla v^i}{dt} = \frac{dv^i}{dt} + \Gamma_{k\ h}^{\ i} v^h \frac{dy^k}{dt}$$

where ∇v^i is the ‘absolute differential’ of the velocity v^i.

Version 2 of the `diffcoeff` package has the generative power to cope with all these variations – see Section 5 – although it is unlikely an author should need to call on this capacity to anything like the extent required for this Rogues’ Gallery.

3 Ordinary derivatives

Writing `\diff{y}{x}` will produce $\frac{dy}{dx}$ in text style (i.e., placed between `$ $`) or

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

in display style (i.e., placed between `\[ \]`). In fact `\diff yx` (omitting the braces) will produce these results, with a saving on keystrokes. The braces are needed only when an argument – differentiand, variable of differentiation – is more than a single token.

- If you want upright ‘d’s as default, as ISO 80000-2 recommends, rather than the math-italic ‘d’s I am using, this can easily be done; see Section 5 on changing default settings.

For inclusion in a line of text you might prefer to use a slash-fraction form of derivative. That is achieved by inserting a slash, ‘/’, between numerator and denominator arguments: `\diff {\ln x}/x` produces $d\ln x/dx$. (Braces are required for the numerator in this case since it contains more than one token.)

3.1 Order of differentiation

An optional first argument allows the order of differentiation to be specified. The order need not be a number; an algebraic order of differentiation is perfectly acceptable as is a mix of the two:

$$\texttt{\textbackslash diff[2]yx} \implies \frac{d^2 y}{dx^2},$$
$$\texttt{\textbackslash diff[n+1]yx} \implies \frac{d^{n+1} y}{dx^{n+1}}.$$

As mentioned, the braces can be and have been omitted around the x and y; the square brackets around the optional argument, the order of differentiation, are essential. For a first-order derivative, no optional argument is needed and entering `1` as the optional argument has no effect:

$$\texttt{\textbackslash diff[1]yx} \implies \frac{dy}{dx}.$$

In slash style, `\diff[2]y/x` produces d^2y/dx^2, and `\diff[n+1]y/x` produces $d^{n+1}y/dx^{n+1}$.

3.2 Multi-character variables of differentiation

Differentiating a function of a function may involve a multi-character differentiation variable. For instance, to differentiate $\ln\sin x$ in x means forming the product

$$\texttt{\textbackslash diff\{\textbackslash ln\textbackslash sin x\}\{\textbackslash sin x\}\textbackslash diff\{\textbackslash sin x\}x} \implies \frac{d\ln\sin x}{d\sin x}\frac{d\sin x}{dx}.$$

Forming the *second* derivative of $\ln\sin x$ will now involve forming (among other quantities)

$$\texttt{\textbackslash diff[2]\{\textbackslash ln\textbackslash sin x\}\{\textbackslash sin x\}} \implies \frac{d^2\ln\sin x}{d(\sin x)^2}.$$

Parentheses have been inserted automatically by `diffcoeff` around $\sin x$ in the denominator to avoid any visual hint that we are differentiating in the sine of the square of x.

The question is: are the parentheses in the right place? Logically, no. They should include the d: $(d\sin x)^2$ – it is the differential $d\sin x$ that is of the second order. But as the examples in the Rogues' Gallery show – see particularly (3) – the inclination seems to be to do otherwise. This may be because one wants, in any case, to parenthesise the variable. A second, outer pair of parentheses then seems fussy and distracting:

$$\frac{d^2 f(x)}{(d(1/x))^2}.$$

Customary but illogical notations are familiar in mathematics – think of the position of the superscripts in an identity like $\sin^2\theta + \cos^2\theta = 1$. But, like other features of the derivative, the manner of this wrapping in parentheses of long variables for *higher order* derivatives is customisable; see Section 5.

For first order derivatives, parenthesising does not occur. If you want the variable of differentiation to be parenthesised, you need to insert them yourself:

`\diff {f(x)}{1/x}, \quad\diff {f(x)}{(1/x)}` $\Longrightarrow$ $$\frac{df(x)}{d1/x},\quad \frac{df(x)}{d(1/x)}.$$

3.3 Appending the differentiand: `\diff*`

Some differentiands are too big or awkward to be placed neatly in the numerator of a derivative and it is natural to append them to a preceding differential operator. One way to do this is to leave the numerator argument empty in the `\diff` command and follow the command with the differentiand. A better way is to star the `\diff` command. This tells `diffcoeff` to append the differentiand. Thus suppose the differentiand is a polynomial, say $ax^2 + bx + c$. Add a star (an asterisk) to the `\diff` command:

`\diff*{(ax^2+bx+c)}x` $\Longrightarrow$ $$\frac{d}{dx}(ax^2 + bx + c).$$

A virtue of using an asterisk is that if one isn't sure whether a differentiand should be appended or not, it is an easy matter to simply insert or delete the asterisk to compare the results. For example, a second derivative is an iterated derivative – one in which a derivative forms the differentiand of another derivative:

`\diff[2]yx = \diff*{\diff yx}x` $\Longrightarrow$ $$\frac{d^2y}{dx^2} = \frac{d}{dx}\frac{dy}{dx},$$

which is more elegant to my eye than

`\diff[2]yx = \diff{\diff yx}x` $\Longrightarrow$ $$\frac{d^2y}{dx^2} = \frac{d\frac{dy}{dx}}{dx},$$

although whether the *meaning* is clearer is moot. It is easy to switch between the two forms on the right, simply by inserting or removing the asterisk.

In slash style with the star option, the polynomial example becomes

`\diff*{(ax^2+bx+c)}/x` $\Longrightarrow$ $(d/dx)(ax^2 + bx + c)$,

where the parentheses around the differential operator are automatically inserted by `diffcoeff`. Like other elements of automatic formatting, this is user-adjustable; see Section 5.

3.4 Point of evaluation

If you want to specify a point at which the derivative is evaluated, append a final optional argument. Note that there *must be no space* before the left square bracket of the argument:

$$\texttt{\textbackslash diff[2]yx[0]} \Longrightarrow \left.\frac{d^2y}{dx^2}\right|_0$$

If a space does slip in before the final optional argument, it will not cause a LaTeX error. Instead, the argument will be treated as a square-bracketed mathematical expression following the derivative, and typeset as such.

- If you prefer to use subscripted *parentheses* around the derivative to indicate a point of evaluation – as ISO 80000-2 recommends – then this can easily be done; see Section 5 on changing default settings. Or use the `ISO` package option; see the introduction.

Because the slash form spreads the derivative out horizontally, parentheses are preferred here to indicate a point of evaluation:

$$\texttt{\textbackslash diff\{\textbackslash ln sin x\}/\{sin x\}[x=\textbackslash pi/6]} \Longrightarrow (d\ln\sin x/d\sin x)_{x=\pi/6}$$

A vertical rule (or 'pipe') can become too remote from the opening d of the differential coefficient: $d\ln\sin x/d\sin x|_{x=\pi/6}$; parentheses tie the whole cluster of symbols together.

3.4.1 Superscripts

It is easy to add a superscript to a derivative to indicate evaluation at two points and the difference between the values:

$$\texttt{\textbackslash diff \{\textbackslash sin x\}x[0]\^{}\{\textbackslash pi/2\}} \Longrightarrow \left.\frac{d\sin x}{dx}\right|_0^{\pi/2}$$

If you want only the superscript, no subscript, include the final optional argument but leave it empty. Thus, for a particle of mass m moving along a line, distance x at time t, the kinetic energy is:

$$\texttt{\textbackslash tfrac 12 m \textbackslash diff x/t[]\^{}2} \Longrightarrow \tfrac12 m(dx/dt)^2.$$

4 Partial derivatives

Partial derivatives follow the same pattern as for ordinary derivatives, with some extensions. The command this time is `\diffp`. Thus `\diffp{F}{x}`, or, with a saving on keystrokes, `\diffp Fx`, produces $\tfrac{\partial F}{\partial x}$ in text style and

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

in display style. (As for `\diff`, the omission of braces is possible when the differentiand or the differentiation variable are single tokens.) As for `\diff`, there is a slash form, generally preferred for inline use, `\diffp F/x`, displaying as $\partial F/\partial x$. Given that `\partial` takes 8 keystrokes to type, the slash form *does* economise on keystrokes for a partial derivative.

Again an optional argument allows the specification of the order of differentiation and it may be numeric or algebraic or a mix of the two:

`\diffp[3]F/x , \quad \diffp[n]F/x` $\implies \quad \partial^3 F/\partial x^3, \quad \partial^n F/\partial x^n.$

`\diffp[n+1]Fx` $\implies \quad \dfrac{\partial^{n+1} F}{\partial x^{n+1}},$

4.1 Variables held constant

In a subject like thermodynamics, there is a need to indicate which variable or variables are held constant when the differentiation occurs. To show this, append a final square-bracketed optional argument and ensure that it follows *immediately* on the preceding mandatory argument. A space here will detach the argument from the derivative and result in it being treated as a mathematical expression following the derivative. Thus to differentiate the entropy S in temperature T while holding the volume V constant, write

`\diffp ST[V]` $\implies \quad \left(\dfrac{\partial S}{\partial T}\right)_V$

In slash form the same expression looks like

`\diffp S/T[V]` $\implies \quad (\partial S/\partial T)_V.$

This use of a parenthesised, subscripted form to indicate a variable or variables held constant, leaves open the question: how do we represent a point of evaluation? ISO 80000-2 makes no recommendation for *partial* derivatives; presumably we follow the same practice as their recommendation for ordinary derivatives:

`\diffp {F(x,y)}x[(0,0)]` $\implies \quad \left(\dfrac{\partial F(x,y)}{\partial x}\right)_{(0,0)}$

However, you may prefer (as I do) to use a vertical rule for this purpose:

$$\left.\frac{\partial F(x,y)}{\partial x}\right|_{(0,0)}$$

Making this possibility available is discussed in Section 5.

An empty final argument produces a parenthesised derivative with no subscript:

`\diffp yx[]`$\implies \quad \left(\dfrac{\partial y}{\partial x}\right).$

This can be useful sometimes, e.g. for writing Lagrange's equations of motion in analytic mechanics:

```
\diffp L{q_k}-\diff*{\diffp L{\dot{q}_k}[]}t = 0
```

$$\implies \quad \frac{\partial L}{\partial q_k} - \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) = 0.$$

4.1.1 Text-style derivatives

The `diffcoeff` package assumes that derivatives formed as 'numerator over denominator' will be used in display-style expressions, and that the slash form will be used for inline use (text style). This is the familiar practice in the literature. If one *does* use the first form in an inline expression where a variable is held constant, say `\diffp ST[V]` as here $\left(\frac{\partial S}{\partial T}\right)_V$, the result is unsatisfactory, the subscript too tight on the closing parenthesis and too much space between parentheses and derivative. The matter is easily resolved using 'variant forms' – see Subsection 5.5.1 below – giving, for our example, $\bigl(\frac{\partial S}{\partial T}\bigr)_V$.

4.2 Appending the differentiand

For a long or awkward differentiand, it is generally better to *append* it to a preceding differential operator, rather than create a fractional form with the long expression in the numerator. As with ordinary derivatives, this is achieved by adding an asterisk to (i.e. by starring) the `\diffp` command.

$$\texttt{\diffp*[2]\{\phi(x,y,z)\}x} \implies \frac{\partial^2}{\partial x^2}\phi(x,y,z).$$

Alternatively you could leave the first mandatory argument empty and manually append the differentiand, but by deleting or inserting an asterisk, it is easy to compare the two forms, differentiand-in-the-numerator, differentiand-appended, and see which is preferable.

In slash form, parentheses are automatically inserted around the differential operator when the differentiand is appended,

$$\texttt{\diffp*[n]\{f(x)\}/x} \implies (\partial^n/\partial x^n)f(x),$$

although this behaviour can be changed (Section 5 again).

If you wish to both append the differentiand *and* indicate variables held constant, then the starred form is much the easier way to achieve this. Thus, to express a relation in thermodynamics,

```
\diffp*{\frac PT}U[V] = \diffp*{\frac 1T}V[U]
```

$$\implies \left(\frac{\partial}{\partial U}\frac{P}{T}\right)_V = \left(\frac{\partial}{\partial V}\frac{1}{T}\right)_U$$

where the starring automatically takes care of the parentheses and subscripts.

4.3 Mixed partial derivatives

The new thing with partial derivatives, not present with ordinary derivatives, is *mixed* partial derivatives, where there is more than one variable of differentiation. If each variable is differentiated only to the first order, then it is easy to specify the derivative. Suppose F is a function of three variables, x, y and z. Then

$$\texttt{\textbackslash diffp F\{x,y,z\}} \implies \frac{\partial^3 F}{\partial x\,\partial y\,\partial z}.$$

The variables of differentiation are listed in order in a comma list forming the second mandatory argument. The total order of differentiation (3 in this example) is inserted automatically – `diffcoeff` does the calculation. There is also a slash form:

$$\texttt{\textbackslash diffp F/\{x,y,z\}} \implies \partial^3 F/\partial x\partial y\partial z.$$

If we want to differentiate variables to higher order, then their orders need to be specified explicitly. To do so use a comma list for the optional argument:

$$\texttt{\textbackslash diffp[2,3]F\{x,y,z\}} \implies \frac{\partial^6 F}{\partial x^2\,\partial y^3\,\partial z}.$$

Notice that the overall order of the derivative – 6 – is again automatically calculated and inserted as a superscript on the ∂ symbol in the numerator. In this example, the comma list of orders has only two members, even though there are three variables. It is assumed that the orders given in the comma list apply in sequence to the variables, the first order to the first variable, the second to the second variable, and so on, and that any subsequent orders not listed in the optional argument are, by default, 1. Thus we need to specify only 2 and 3 in the example; the order of differentiation of z is 1 by default. But you *cannot* use an order specification like `[,,2]`. Instead write `[1,1,2]`. It is only the *tail* of an order specification which can be omitted.

The automatic calculation of the overall order of differentiation remains true even when some or all of the individual orders are variables rather than numbers. For example, differentiating in three variables to orders `1, m+1, m-1`,

$$\texttt{\textbackslash diffp[1,m+1,m-1]\{F(x,y,z)\}\{x,y,z\}} \implies \frac{\partial^{2m+1} F(x,y,z)}{\partial x\,\partial y^{m+1}\,\partial z^{m-1}}.$$

Should you specify *more* orders in the comma list than there are variables, compilation will fail and an error message will be sent to the terminal and LaTeX log . For example, if on line 53 (say) of my document I specify `[1,m-1,m+1,2]` for the orders of differentiation but list only `{x,y,z}` for the variables, the message will be

```
! Package diffcoeff Error: 4 orders specified for 3
variables; [1,m+1,m-1,2] (on line 53) for variables x,y,z.
```

Perhaps the differentiations are to orders `[2km,m-1,m+1]`:

$$\texttt{\textbackslash diffp[2km,m-1,m+1]\{F(x,y,z)\}\{x,y,z\}} \implies \frac{\partial^{2km+2m} F(x,y,z)}{\partial x^{2km}\,\partial y^{m-1}\,\partial z^{m+1}}.$$

Here the overall order is presented as `2km+2m`. You might prefer this to be presented as `2m(k+1)`. Although `diffcoeff` takes some steps to present the overall order appropriately, it is not a computer algebra system and does not factorise expressions. If you want to present the order in a manner distinct from that presented by `diffcoeff`, use the *order-override option*.

4.3.1 The order-override option

This is a second optional argument immediately following the order specification. For our last example, filling the override option with `2m(k+1)` gives the desired result:

```
\diffp[2km,m-1,m+1][2m(k+1)]{F(x,y,z)}{x,y,z}
```

$$\implies \quad \frac{\partial^{2m(k+1)}F(x,y,z)}{\partial x^{2km}\,\partial y^{m-1}\,\partial z^{m+1}}.$$

As another example, left to its own devices, `diffcoeff` produces

`\diffp[m/2+n/2,m/2,n/2]F{x,y,z}` $\implies \quad \dfrac{\partial^{2m/2+2n/2}F}{\partial x^{m/2+n/2}\,\partial y^{m/2}\,\partial z^{n/2}},$

whereas we would like the total order to be presented as $m + n$. Using the override option,

`\diffp[m/2+n/2,m/2,n/2][m+n]F{x,y,z}` $\implies \quad \dfrac{\partial^{m+n}F}{\partial x^{m/2+n/2}\,\partial y^{m/2}\,\partial z^{n/2}}.$

The order-override option does exactly that: overrides the presentation of the calculated order with the manually given one. In fact the calculation algorithm does not get called at all when the override option is used so that one can in this way present the total order in whatever manner one wishes or, indeed, add essentially arbitrary material as a superscript to the ∂ symbol in the numerator.

4.3.2 Comma list of variables of differentiation

In tensor calculus the differentiations are almost always in terms of super- or subscripted coordinates. In many other contexts this is the case too – the reciprocal of the temperature in thermodynamics or generalized cooredinates in analytical mechanics. This is why a comma list is used in `diffcoeff` for specifying variables of differentiation for mixed partial derivatives. Although it would be nice to write the minimal `{xy}` rather than `{x,y}` when two variables x and y are involved, the extra writing is trivial and the comma list allows a simpler handling of multi-character variables. For instance in tensor calculus we get expressions like

`\diffp{A_i}{ x^j,x^k }` $\implies \quad \dfrac{\partial^2 A_i}{\partial x^j\,\partial x^k}.$

It is easier to write `{x^j,x^k}` here than, say, `{{x^j}{x^k}}` to distinguish the variables. It does mean that should the variable of differentiation include a comma then that comma needs to be enclosed in braces. There are plenty of instances of this out in the world (see, e.g., the last equation of (1)) but it is overall a rare occurrence.

4.3.3 Overkill territory

Two previous examples illustrate limitations of the algorithm that calculates the overall order of differentiation: `2m/2+2n/2` is not simplified to `m+n` and `2km+2m` is not factorised to `2m(k+1)`. But there is much that the algorithm *can* handle – for instance, the simple use of parentheses:

```
\diffp[2m-(k+1),2(k+1)-m]{F(x,y,z)}{x,y,z}
```

$$\Longrightarrow \quad \frac{\partial^{m+k+2} F(x,y,z)}{\partial x^{2m-(k+1)}\,\partial y^{2(k+1)-m}\,\partial z}.$$

Dynamic use of parentheses For parenthetic expressions to be evaluated as in this example – the *dynamic* use of parentheses – the left parenthesis must be preceded at most by a sign or a number; the right parenthesis must be followed at most by a sign.

If a right parenthesis is followed by a *variable*, say by `m` as in the order spec. `[(2n+1)m,(2n-1)m]`, it will throw an error and halt compilation. A message will be sent to the terminal and the LaTeX log like the following (which assumes the order spec. was on line 53 of the document):

```
! Package diffcoeff Error: Is this intended: ) followed
by m in the order specification [(2n+1)m,(2n-1)m] on
line 53? Diffcoeff cannot calculate the overall order of
differentiation from the specification in this case. Use
the order-override option to enter the overall order.
```

This is a limitation on the dynamic use of parentheses – but they *can* be nested.

Static use of parentheses If a left parenthesis is preceded by a *variable* (i.e., not a sign or a number) this signals to `diffcoeff` the *static* use of parentheses. No attempt is made to evaluate what is between them and they are treated simply as an extension of the variable. For example,

`\diffp[f(k+1)+1,f(k-1)-1]{F(x,y)}{x,y}` $\Longrightarrow \quad \dfrac{\partial^{f(k+1)+f(k-1)} F(x,y)}{\partial x^{f(k+1)+1}\,\partial y^{f(k-1)-1}}.$

In the static case you *can* follow the right parenthesis by a variable without generating an error. You can nest them, and you can include static parentheses within a dynamic pair; for example,

```
\diffp[2(3+f(k))+1,1-3(f(k)-2)]{F(x,y)}{x,y}
```

$$\Longrightarrow \quad \frac{\partial^{14-f(k)} F(x,y)}{\partial x^{2(3+f(k))+1}\,\partial y^{1-3(f(k)-2)}}.$$

However, the reverse is not possible: you can't have dynamic parentheses within a static pair.

Other refinements Exponents and subscripts on a *variable* are fine in an order specification, so long as the exponent or subscript consists of a *single* token:

```
\diffp[m^2+2(k-1),m^2-(k+1)]F{x,y,z,w}
```

$$\Longrightarrow \quad \frac{\partial^{2m^2+k-1}F}{\partial x^{m^2+2(k-1)}\,\partial y^{m^2-(k+1)}\,\partial z\,\partial w}.$$

Braced arguments containing *multiple* tokens as exponents or subscripts to variables will generally not halt compilation but will usually give nonsensical results, as will *signs* treated as superscripts or subscripts. Neither circumstance is checked for by `diffcoeff`.

Numbers raised to a power will cause a LaTeX error. (The `l3int` module of the expl3 language is used to evaluate expressions, and this does not cater for raising to a power.)

Override There are limitations on what order specifications the `diffcoeff` package can 'digest'; equally, it can digest a wide variety of such constructs, but it is *not* a computer algebra system. In all those cases where it fails to calculate or present a correct total order, the order-override option is available. Yes, this is not as convenient as having the overall order calculated automatically but, let's remind ourselves, we are deep in overkill territory. Mixed partial derivatives are used far less often than the pure derivatives, and when they *are* used it is nearly always to orders 1 or 2 in the variables. Mixed partial derivatives to exotic orders of differentiation are rarely used, so that the limitations of the calculational algorithm are of little real moment – and the override option is always available for such cases.

4.4 Parentheses around multi-character variables

In thermodynamics and statistical mechanics one may want to differentiate in the reciprocal of the temperature, $1/T$ (or $1/\Theta$):

`\diffp[2]q{\frac 1\Theta}` $\Longrightarrow \quad \dfrac{\partial^2 q}{\partial(\frac{1}{\Theta})^2}.$

In this case and for other *higher order* derivatives of multi-character variables of differentiation, the parentheses are inserted automatically by `diffcoeff`. Precisely where parentheses should be placed is moot. The placement in this example is not strictly logical, although it feels intuitive, but the placement can be customised (Section 5).

Parentheses are automatically inserted like this only for higher order derivatives. When the differentiation is to first order, parenthesising is up to the user:

`\diffp q{(\frac 1\Theta),V}` $\Longrightarrow \quad \dfrac{\partial^2 q}{\partial(\frac{1}{\Theta})\,\partial V}.$

4.5 Jacobians

`diffcoeff` provides a command `\jacob` for constructing Jacobians. For example

$$\texttt{\textbackslash jacob\{u,v,w\}\{x,y,z\}} \implies \frac{\partial(u,v,w)}{\partial(x,y,z)}.$$

The comma lists can contain any number of variables. `\jacob` does *not* check that the two arguments contain the same number of variables, so it is perfectly possible to form an object like `\jacob{u,v,w}{x,y}` which as far as I know has no meaning.

5 Changing defaults; variant forms

To write the range of different examples displayed in the Rogues' Gallery (Section 2) I have had to make extensive use of forms of derivative other than the default. Version 2 of `diffcoeff` (as distinct from version 1) is built around the facilities offered by the `xtemplate` package (included in the LaTeX3 bundle `l3packages`). These facilities are harnessed by means of a command, `\diffdef`, and a further optional argument of the `\diff` command.

How a derivative is displayed in a document is determined by specifying values in a 'key = value' list. This is done with the `\diffdef` command, which also allows a name to be associated with the list. By using that name as an argument in the `\diff` command, a derivative is formed shaped by those values. Examples will make the process clear.

Table 1: Defaults (ordinary derivatives)

key	default
op-symbol	`d`
op-symbol-alt	`= op-symbol`
op-order-sep	`1 mu`
*-op-left	`false`
*-italic-nudge	`3 mu`
*/-op-wrap	`true`
long-var-wrap	`d(v)`
denom-term-sep	`2 mu`
/-denom-term-sep	`1 mu`
left-delim	`\left .`
right-delim	`\right |`
elbowroom	`0 mu`
subscr-nudge	`0 mu`
/-left-delim	`(`
/-right-delim	`)`
/-elbowroom	`0 mu`
/-subscr-nudge	`0 mu`

5.1 Default values: ordinary derivatives

Table 1 lists the keys available for forming derivatives and the default values[1] they have been assigned. These default values have been chosen to coincide with those relevant for *ordinary* derivatives (apart from the keys `denom-term-sep` and `/-denom-term-sep` which are ignored for ordinary derivatives but apply

[1]Note that a mu is a 'math unit', 1/18 of an em in the math font used.

to the case of mixed partial derivatives when there is more than one variable of differentiation.) Keys with an opening slash, /, apply only to the slash form of the derivative; keys with an opening asterisk, *, apply only when the differentiand is appended.

Note that these settings are, in general, font dependent. The values given are (in the author's opinion) appropriate for the default LaTeX math fonts, or latin modern fonts. There are also likely to be variations required for whether a derivative is used in a display-style or text-style or script-style expression. That matter is discussed below in Subsection 5.5.1.

op-symbol the operator symbol; for ordinary derivatives, generally one of `d` or `\mathrm{d}`, `D` or `\mathrm{D}`, `\delta` or `\Delta`; for partial derivatives `\partial`

op-symbol-alt if different from **op-symbol** then used in the denominator while **op-symbol** is used in the numerator; otherwise (and usually) defaults to **op-symbol**; e.g. for $\frac{\nabla v^i}{dt}$, `op-symbol = \nabla` and `op-symbol-alt = d`

op-order-sep extra horizontal space added between the op-symbol and the superscripted order of differentiation in higher order derivatives; compare d^2 with $d^{\,2}$, ∂^2 with $\partial^{\,2}$ where the first symbol in each case has no extra space and the second has an extra 1 mu

***-op-left** a choice of `true` or `false` indicating whether the op-symbol is left-aligned or not when the differentiand is appended; generally it is centred; does not apply to slash forms of the derivative

***-italic-nudge** if ***-op-left** is `true`, makes an italic adjustment in the numerator, so that the op-symbols in numerator and denominator align in the same slanting column; for an upright `d` this would be set to `0 mu`

***/-op-wrap** a choice of `true` or `false` for slash forms of the derivative when the differentiand is appended, dictating whether the differential coefficient is wrapped in parentheses or not; the default is `true`, as here: $(\partial/\partial x)F(x,y)$

long-var-wrap to avoid ambiguity in higher order derivatives it may be advisable to wrap multi-token variables of differentiation in parentheses; the choices are

- `dv` no wrapping, e.g. dx_i^2 or $d\frac{1}{\Theta}^2$, ∂x_i^2 or $\partial\frac{1}{\Theta}^2$,
- `d(v)` wrap the variable only, e.g. $d(x_i)^2$ or $d(\frac{1}{\Theta})^2$, $\partial(x_i)^2$ or $\partial(\frac{1}{\Theta})^2$,
- `(dv)` wrap the op-symbol and variable, e.g. $(dx_i)^2$ or $(d\frac{1}{\Theta})^2$, $(\partial x_i)^2$ or $(\partial\frac{1}{\Theta})^2$

denom-term-sep (mixed partial derivatives only) extra horizontal spacing inserted between the differentials in the denominator of a mixed partial derivative

/-denom-term-sep (mixed partial derivatives only) extra horizontal spacing inserted between the differentials in the denominator of a slash-form mixed partial derivative

left-delim the left member of a delimiter pair wrapping the derivative, the right member of which is subscripted to indicate a point of evaluation or variables held constant

right-delim the right member of a delimiter pair wrapping the derivative and subscripted to indicate a point of evaluation or variables held constant

elbowroom adjustment to the whitespace between the left and right delimiters and the enclosed derivative; negative values reduce the space

subscr-nudge horizontal adjustment of the subscript's placing relative to the **right-delim**iter, e.g., a negative value compensates for the curving inwards of a large right parenthesis; may be font dependent

/-left-delim for the slash form of derivative, the left member of a delimiter pair wrapping the derivative and subscripted to indicate a point of evaluation or variables held constant

/-right-delim for the slash form of derivative, the right member of a delimiter pair wrapping the derivative, the right member of which is subscripted to indicate a point of evaluation or variables held constant

/-elbowroom adjustment to the whitespace between the left and right delimiters and the enclosed slash-form derivative

/-subscr-nudge for the slash form of derivative, horizontal adjustment of the subscript's placing relative to the /-**right-delim**iter; may be font dependent

5.2 ISO defaults

You may not like the default settings that come with `diffcoeff`. The package does not follow ISO 80000-2 – it does not use upright 'd's nor does it wrap an ordinary differential coefficient in subscripted parentheses to indicate a point of evaluation. Both 'defects' can be remedied by calling the package with the option `ISO`:[2]

Table 2: ISO default changes

key	default
op-symbol	`\mathrm{d}`
op-order-sep	`0 mu`
left-delim	`\left (`
right-delim	`\right )`
subscr-nudge	`-6 mu`

```
\usepackage[ISO]{diffcoeff}
```

[2]One can also use `ISO=true` to turn the option on and `ISO=false` to turn the option off.

The uppercase is essential – an option `iso` is not recognised. The `ISO` option results in changes to the default settings of Table 1 as listed in Table 2. Any settings not mentioned in Table 2 retain the values presented in Table 1. Note that the subscript nudge figure specified here is *not* part of the standard, which makes no recommendation about the subscript's positioning. But: the `-6 mu` figure with a default or latin modern font gives a better representation of what is displayed in the standard than a zero figure.

Because the 'd' is upright with the `ISO` option, no extra space is required between the symbol and the superscript in a higher order derivative. Hence the zero value for the `op-order-sep`. ISO recommends subscripted parentheses to indicate a point of evaluation. Hence the other entries in the table. Because a large right parenthesis (penultimate setting) bends inwards, to the left, a negative value for the last entry ensures the subscript does not become detached from the derivative, looking lost in a sea of whitespace.

Note that the `ISO` option will also produce upright 'D's in derivatives formed from 'D'; see Subsection 5.5.2 below.

5.3 Partial derivatives

The default values given in Table 1, when they are relevant, apply to *ordinary* derivatives. For partial derivatives, the defaults are those of Table 3. All other keys take ther default values listed in Table 1. The last three entries here are not an attempt at ISO compatibility but reflect the use of subscripted parentheses with partial derivatives to indicate variables held constant, for instance in the Maxwell relations of thermodynamics, one of which is

Table 3: Default changes: partial derivatives

key	default
op-symbol	`\partial`
left-delim	`\left (`
right-delim	`\right )`
subscr-nudge	`-6 mu`

$$\left(\frac{\partial S}{\partial V}\right)_T = \left(\frac{\partial P}{\partial T}\right)_V .$$

5.4 Setting your own defaults: `\diffdef`

Version 2 of the `diffcoeff` package provides a command, `\diffdef`, that enables users to set their own defaults. For example, if you wish to use upright 'd's but not follow the ISO's use of subscripted parentheses to indicate a point of evaluation, enter in the preamble of your document the command

```
\diffdef {}
    {
        op-symbol    = \mathrm{d},
        op-order-sep = 0 mu
    }
```

Since a list of settings, like this one, is a comma-*separated* list, no comma is required for the last entry. That entry is a consequence of the first: upright symbols do not require any extra separation between the 'd' and the superscript in a higher order derivative.

The other point to note is the empty pair of braces after the `\diffdef` command. *They matter*. Their emptiness is what determines that it is the *default* values that are changed. If they contain some content, then that content provides a *name* for the particular set of values in the following list. The `diffcoeff` package exploits this facility to cope with the wide variety of forms displayed in the Rogues' Gallery of Section 2.

5.5 Variant forms

For this package I needed to have a number of variant forms available to illustrate the wide variety of ways in which derivatives are displayed. The `\diffdef` command in which the first argument is *filled* provides one half of the means of doing this. I've given the single-letter name `p` to the following settings:

```
\diffdef { p }
  {
    op-symbol    = \partial ,
    left-delim   = \left (  ,
    right-delim  = \right ) ,
    subscr-nudge = -6 mu
  }
```

The second half of providing variant forms is to insert this name, `p`, between dots (periods, full stops) as the *first* argument of the `\diff` command. Thus, repeating an example at the end of Subsection 4.2,

```
\diff.p.*{\frac PT}U[V] = \diff.p.*{\frac 1T}V[U]
```

$$\Longrightarrow \quad \left(\frac{\partial}{\partial U}\frac{P}{T}\right)_{V} = \left(\frac{\partial}{\partial V}\frac{1}{T}\right)_{U}$$

The effect is exactly the same as previously, when the `\diffp` command was used. Indeed, `diffcoeff` identifies `\diffp` with `\diff.p.`, saving a few keystrokes and maintaining compatibility with version 1 of the package. In LaTeX 2_ε synatx,

```
\newcommand { \diffp } { \diff.p. }
```

Note that this identification of `\diffp` with `\diff.p.` means there is no equivalent dot-delimited argument available for `\diffp`. The dot-delimited argument applies only to `\diff`.

For example, to illustrate the upright-d form of derivative, without changing the default math-italic form (which I prefer), one might enter in the preamble

```
\diffdef { up }
  {
    op-symbol    = \mathrm{d},
    op-order-sep = 0 mu
  }
```

Apart from the key = value settings, the critical feature here is the name, `up` (which is at your discretion and could equally be `upright` or `roman` or even `Fred` if you so fancied). This ensures that the changed settings apply only to this particular variant and do not 'infect' the overall defaults. To use this variant, all that is needed is to add the name, between dots, to the `\diff` command:

$$\texttt{\textbackslash diff.up.yx} \implies \frac{\mathrm{d}y}{\mathrm{d}x}.$$

Each variant derivative inherits all the default values that it does not explicitly countermand. Thus a point of evaluation is indicated by a vertical rule which is the `diffcoeff` default[3]:

$$\texttt{\textbackslash diff.up.*\{\textbackslash frac\{F(x)\}\{G(x)\}\}x[x=1]} \implies \left.\frac{\mathrm{d}}{\mathrm{d}x}\frac{F(x)}{G(x)}\right|_{x=1}$$

Dot-delimited arguments must always be the *first* argument of the `\diff` command, even preceding an asterisk (star) as in this example.

As another example, suppose for the subscripted indication of variables held constant in a partial derivative that you want to see what things look like if the subscript is *not* nudged in towards the right parenthesis. In that case define a variant form

```
\diffdef { padrift } { subscr-nudge = 0 mu }
```

I have attached a name, `padrift`, to this setting,

$$\texttt{\textbackslash diff.padrift.Fx[y]} \implies \left(\frac{\partial F}{\partial x}\right)_{\;y}$$

since, to my eye, the subscript seems detached from the expression it qualifies and 'adrift in a sea of whitespace'. Is it perhaps a typo? This is to be compared with the default

$$\texttt{\textbackslash diffpFx[y]} \implies \left(\frac{\partial F}{\partial x}\right)_{\!y}$$

where the subscript is tucked in close to the right parenthesis and is clearly connected to it and the expression it delimits.

Some might want to distinguish notationally a point of evaluation for a partial derivative from variables held constant, perhaps using a vertical rule for the former and (the default) parentheses for the latter. It would suffice then to add to the preamble

[3]But not the ISO recommendation.

```
\diffdef { pvrule } { op-symbol = \partial }
```

(or some other name of your choosing). This gives

`\diff.pvrule.{F(x,y)}x[x=1]` $\Longrightarrow \left.\dfrac{\partial F(x,y)}{\partial x}\right|_{x=1}$

5.5.1 Text-style and script-style derivatives

As noted earlier, the `diffcoeff` package assumes that derivatives of fraction-like form will be used in display-style expressions and that the slash form will be used for inline use (text style). This is the usual practice. But if one does want to use the fraction form in an inline expression, say `\diffp ST[V]` displaying as $\left(\frac{\partial S}{\partial T}\right)_V$, then some tweaking of settings is necessary: the subscript is obviously too close to the right parenthesis and, to my eye, there is too much 'elbowroom' between the derivative and the enclosing parentheses:

```
    \diffdef { ptxt }
      {
        op-symbol      = \partial,
        denom-term-sep = 1 mu    ,
        left-delim     = \left ( ,
        right-delim    = \right ),
        elbowroom      = -2 mu   ,
        subscr-nudge   = -3 mu
      }
```

This gives, for the same example, `\diff.ptxt.ST[V]` displaying as $\left(\frac{\partial S}{\partial T}\right)_V$, where the subscript is better positioned and there is a better fit between parentheses and derivative. For a mixed partial derivative, `\diff.ptxt.{F(x,y,z)}{x,y}[z]` displays now as $\left(\frac{\partial^2 F(x,y,z)}{\partial x \partial y}\right)_z$.

5.5.2 Derivatives from D, \delta, \Delta

In addition to `\diff.p.`, `diffcoeff` has three further *built-in* variant forms that are commonly used: `\diff.D.`, `\diff.delta.`, and `\diff.Delta.`, corresponding to derivatives formed from D, δ and Δ respectively.

In fluid dynamics the *material* or *substantive* derivative uses an uppercase D in place of d. For example, the continuity equation is,

`\diff.D.{\rho}t=\diffp\rho t + \mathbf{u\cdot}\nabla\rho`

$$\Longrightarrow \frac{D\rho}{Dt} = \frac{\partial\rho}{\partial t} + \mathbf{u}\cdot\nabla\rho$$

where `\diff.D.` produces the D-derivative. If you want upright 'D's, then the `ISO` package option will produce that effect. Alternatively,

```
\diffdef { Up }
  {
    op-symbol    = \mathrm{D},
    op-order-sep = 0 mu
  }
```

provides a variant with upright 'D's.

In introductory calculus texts the simple δ-derivative is used. This is achieved with the `\diff.delta.` command

$$\texttt{\textbackslash diff.delta.yx} \implies \frac{\delta y}{\delta x}.$$

This form also features in analytical mechanics (in the Rogues' Gallery, the final example at (6)).

Similarly, `\diff.Delta.` forms a derivative from Δ:

$$\texttt{\textbackslash diff.Delta.y/x} \implies \Delta y/\Delta x,$$

where the slash form of the derivative is shown in this instance.

Higher order forms of these derivatives, points of evaluation, appending the differentiand with a star argument, all follow exactly as for the 'pure' `\diff` command.

5.5.2.1 The commands \Diff, \diffd, \Diffd

For compatibility with version 1 of `diffcoeff`, the commands `\Diff`, `\diffd` and `\Diffd` are available and also produce the D, δ and Δ derivatives. Just as `\diffp` is identified with `\diff.p.` for partial derivatives, these commands are identified with `\diff.D.`, `\diff.delta.`, and `\diff.Delta` through commands equivalent to[4]

```
\newcommand{\Diff}{\diff.D.}
\newcommand{\diffd}{\diff.delta.}
\newcommand{\Diffd}{\diff.Delta.}
```

Unless one is using such variant forms frequently, it seems simpler to remember that they are available as dot-delimited arguments to the `\diff` command, using the obvious name in each case, rather than having to remember the precise camel-case form of name of the `\Diff`, `\diffd` and `\Diffd` commands.

5.6 The .def file

This mechanism of variant formation is how I have been able to illustrate in the Rogues' Gallery, Section 2, the wide variety of different usages culled from the literature. But the thought arises: if a variant is to be used only once or twice, isn't this a lot of bother? Why not just construct the variant derivative

[4] In fact the actual commands in `diffcoeff.sty` use the syntax of the `xparse` package, e.g. `\NewDocumentCommand { \Diff } { } {` \diff.D. }, and similarly for the others.

'by hand' out of `\frac` and `\mkern` for example? The reason for making such definitions is that they can be transferred from document to document. For instance, definitions placed in the preamble can be copied to the preamble of another document.

But that is hardly optimal. Instead, `diffcoeff` allows such definitions to be placed in a text file with the the extension `.def` and a name of your choosing. For the present document the file is called `diffcoeff-doc.def` and has been placed in the same directory as `diffcoeff.tex`. To use these definitions, the `diffcoeff` package is called with the command

```
\usepackage[def-file=diffcoeff-doc]{diffcoeff}
```

But even this process still means copying a definition file from directory to directory as one works on different documents. The solution is to make a definition file available for *all* documents and the way to do that is by placing it in the texmf tree, preferably not the one created by your TEX distribution (perhaps MiKTeX or TexLive), but your own *personal* texmf tree.

Personal texmf tree?
This is a directory for 'waifs and strays' of the TEX system that are not included in standard distributions like MiKTEX or TEXLive. For instance, it is the place for personal packages designed for your own particular circumstances or preferences, and is structured like the standard MiKTEX or TEXLive hierarchy but placed in another location so that there is no chance of its being overwritten when MiKTEX or TEXLive are updated. However, those distributions need to be alerted to its existence. For MiKTEX, open the MiKTEX console, click on Settings and then the Directories tab. Click the + button and navigate to your personal texmf tree to add it to the MiKTEX search path. Having added it, you will then need to refresh the filename database by clicking on the Tasks menu and selecting the obvious entry. I am not familiar with TEXLive but presume a similar process will apply there.

Provided your LATEX distribution knows about your personal texmf tree, then a `.def` file placed within it, will be accessible to all documents.

5.6.1 Structure of the `.def` file

The best way to see what a `.def` file looks like is to view `diffcoeff-doc.def` in a text editor.[5]

If you want your variant definitions to use defaults different from those supplied with the `diffcoeff` package, then the first definition in the `.def` file should be the one setting the new defaults, with an *empty* first argument to the `\diffdef` command:

[5]This file should be in the same directory as `diffcoeff.pdf` and `diffcoeff.tex` in your LATEX distribution.

```
\diffdef {}
    {
        key-1 = value-1,
        key-2 = value-2,
        ...
        key-n = value-n
    }
```

The key-value list is a comma-separated list; hence the last entry doesn't need to end with a comma. Nudge and separation values need to include the unit, `mu`; a numerical value alone will result in error. Because a `.def` file is a LaTeX file, comments need to start with a `%` character.

5.6.2 `diffcoeff.def`

Note that if the `diffcoeff` package is invoked without an explicit `def-file=<filename>` option statement, as here,

```
\usepackage{diffcoeff}
```

then it will search in the texmf tree (the LaTeX distribution's and your personal one) and the document directory for a file `diffcoeff.def` and if found will load that. This file should contain definitions of those variants you are likely to use in multiple documents. In my personal texmf tree (which I've put at `D:\texmf\` on a Windows machine) the file `diffcoeff.def` is located in the directory `D:\texmf\tex\latex\diffcoeff\`. (The backslashes are replaced by forward slashes on linux machines.) Variants likely to be of value only to a specific document should be added to the preamble of that document, or they can be added to `diffcoeff.def` and that file saved to the document directory under a *different* name – e.g. I've saved the variants required for the present document under the name `diffcoeff-doc.def`. Many of these variants were created solely to illustrate points in the present document and I have no intention of using them in my own work. Consequently my `diffcoeff.def` file is smaller, containing only a selection from `diffcoeff-doc.def`.

6 Rationale

Version 1 of the `diffcoeff` package arose from a need to simplify the parsing of differential coefficients for another program I was working on which was struggling to 'read' all the possible permutations of `\frac` or `\tfrac` or `\dfrac` or slash forms of the derivative, of `d` or `\mathrm{d}` or `\partial` or `D` or `\mathrm{D}` or `\nabla`, and of points of evaluation delimited by vertical rules or parentheses. Although regular expressions coped with most of these cases, it was *messy.*

There are other packages which have commands for the derivative (e.g., `bropd`, `commath`, `esdiff`, `physymb`) but none quite gave what I wanted – although they probably cope with most users' needs. `esdiff` came closest to what I was seeking but failed when it came to combining algebraic and numeric orders of differentation in a mixed partial derivative (and made heavier use of braces than I would like in that case too).

6.1 `diffcoeff.sty`

I have tried to make using `diffcoeff` intuitive. Looking at the other packages mentioned, writing something like `\diff[n]{f}{x}` (which can be trimmed to `\diff[n]fx` for single-token arguments) seems 'natural' – only `physymb` deviates from the pattern.

- It seems consistent with this pattern to use a comma list for specifying the orders of differentiation of the variables in a higher order mixed partial derivative (and its suppression when all are of order 1)
- Having specified the orders, surely the program itself should calculate the overall order? `esdiff` does this for numerical orders; `diffcoeff` does this for both numeric and algebraic orders,

```
\diffp[m-(k+1),m+(k-1)]{F(x,y,z)}{x,y,z}
```

$$\Longrightarrow \frac{\partial^{2m-1}F(x,y,z)}{\partial x^{m-(k+1)}\,\partial y^{m+(k-1)}\,\partial z},$$

- and where it fails, either to calculate at all or to present the result in a preferred form, offers the order-override option:

```
\diffp[m+(k+1),m+(k-1)][2(m+k+1)]{F(x,y,z,w)}{x,y,z,w}
```

$$\Longrightarrow \frac{\partial^{2(m+k+1)}F(x,y,z,w)}{\partial x^{m+(k+1)}\,\partial y^{m+(k-1)}\,\partial z\,\partial w}.$$

- I wished to avoid the unnecessary writing of superscripts, subscripts and brace pairs. In the examples just given, no superscript tokens `^` are written by the user despite the higher-order differentiation in x and y, and only the two inescapable brace pairs are required.
- The use of a comma list for the second mandatory argument in a partial derivative – the list of variables – makes differentiations in super- or subscripted symbols (as occurs prolifically in tensor calculus) easier to both write and read by avoiding unnecessary brace pairs.

`\diffp{A_i}{ x^j,x^k }` $\Longrightarrow \dfrac{\partial^2 A_i}{\partial x^j\,\partial x^k}.$

- Should a point of evaluation or variables held constant be considered part of the derivative? Thermodynamic usage was decisive here. The partial derivative alone is ambiguous – the parentheses and subscript are essential to understand what is being stated:

$$\left(\frac{\partial S}{\partial T}\right)_V$$

Hence provision for these extra elements was included in the derivative commands.

- Given the position of the subscripted symbol in the displayed derivative, it's positioning as the *final* argument in the derivative commands feels inevitable.

- Version 1 of `diffcoeff` used braces for this argument to avoid any possible confusion with a following mathematical expression. That use of braces is now deprecated in `xparse`. Consequently version 2 of `diffcoeff` uses square brackets, conforming with familiar LaTeX practice. The only special remembering needed is avoidance of a space before the argument – and if it does slip in, it won't cause a LaTeX error. It will be treated as part of a following mathematical expression and displayed as such.

- The star option also prompted the question: is it needed? After all, one can always leave the first mandatory argument empty and append the differentiand 'by hand'. But once the provision for points of evaluation or variables held constant was incorporated into the derivative commands, the star option became the simplest way of handling appended differentiands since the parentheses for a variable held constant must wrap around the differential operator *and* differentiand. Once available, it provides a simple way of switching between (and comparing) the appearance of differentiand-in-the-numerator and differentiand-appended.

- The slash option was added to the derivative commands after seeing how widely such forms are used in texts at all levels. The placement of the slash, between the two mandatory arguments, seems more-or-less self-evident.

- The final option added to `\diff` (and not present in version 1) was the dot-delimited name option. Once `xtemplate` was used as the basis of the package this seemed the most straightforward way of making available, ready to hand, the wealth of variants that `xtemplate` makes possible. (It's just a pity that the second dot is needed, and a single-dot naming scheme can't be used, but `xparse` forces my hand here.)

- Having added the dot-delimited name option, the use of a `.def` file to store variants or preferred defaults is more-or-less forced, otherwise one is faced with making these definitions anew for each new document (or locating a previous document and copying from that to the new one).

- To handle possible differences between display-style and text-style (and script-style) derivatives (see Subsection 5.5.1) I considered using TEX's `\mathchoice` command. This command takes four arguments, corresponding to display-, text-, script- and scriptscript-styles and would require forming four derivatives each time a derivative is used, 'just in case'. In fact fraction-form derivatives are used overwhelmingly in display-style expressions, the slash form being used for inline use. Given the ease of defining a fraction-form variant for text-style use, and the rareness of such use, employing variants seemed the way to go. It is the one adopted and avoids the computational burden associated with the use of `\mathchoice`.