Covariant differentiation

One way to compute the covariant derivative $v_{a;b}$ is to expand the right hand side of

$$v_{a;b} A^a D^b = \frac{d\left(v_a A^a\right)}{ds}$$

while using

$$\begin{aligned}
\frac{dv_a}{ds} &= v_{a,b} D^b \\
0 &= \nabla_D A^a = \frac{dA^a}{ds} + \Gamma^a_{bc} A^b D^c \\
0 &= \nabla_D D^a = \frac{dD^a}{ds} + \Gamma^a_{bc} D^b D^c
\end{aligned}$$

and then to compare corresponding coefficients of the $A^a D^b$ terms. This leads directly to the standard result

$$v_{a;b} = v_{a,b} - \Gamma^c_{ab} v_c$$

```
# --- some basic declarations ---------------------------------------------

::PostDefaultRules( @@collect_terms!(%) ).

{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Indices.

\nabla{#}::PartialDerivative.
\partial{#}::PartialDerivative.

A^{a}::Depends(\partial).
D^{a}::Depends(\partial).
v_{a}::Depends(\partial).
\Gamma^{a}_{b c}::Depends(\partial).

# --- compute the covariant derivative  -----------------------------------

rhs00:=v_{a} A^{a}:

rhs01:=D^{c}\partial_{c}{@(rhs00)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@substitute!(%)(D^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}D^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):

# --- tidy up and display the results  ------------------------------------

@factor_out!!(rhs00){A^{a}}:
@factor_out!!(rhs00){D^{a}};

@factor_out!!(rhs01){A^{a}}:
@factor_out!!(rhs01){D^{a}};

# --- Cadabra output is printed only when a %! command is given -----------
```

$$rhs00 := v_a A^a$$

$$rhs01 := A^a D^b \left(\partial_b v_a - \Gamma^c{}_{ab} v_c\right)$$

Symmetrised covariant derivatives

Repeated application of d/ds will produce symmetrised covariant derivatives,

$$v_{a;i_1 i_2 i_3 \cdots i_n} A^a D^{i_1} D^{i_2} D^{i_3} \cdots D^{i_n} = \frac{d^n \left(v_a A^a\right)}{ds^n}$$

The procedure is similar to the previous example. We expand the right hand side, one derivative at a time, while using the above equations to eliminate the ordinary derivatives in A^a and D^a.

Now we shall train Cadabra to compute $v_{a;(bcd)} A^a D^b D^c D^d$.

```
# --- some basic declarations ----------------------------------------------

\partial{#}::PartialDerivative.

A^{a}::Depends(\partial).
D^{a}::Depends(\partial).
v_{a}::Depends(\partial).
\Gamma^{a}_{b c}::Depends(\partial).

# --- compute the first 3 symmetrised covariant derivatives  ---------------

rhs00:=v_{a} A^{a}:

rhs01:=D^{c}\partial_{c}{@(rhs00)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@substitute!(%)(D^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}D^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):

rhs02:=D^{c}\partial_{c}{@(rhs01)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@substitute!(%)(D^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}D^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):

rhs03:=D^{c}\partial_{c}{@(rhs02)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@substitute!(%)(D^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}D^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):

# --- tidy up and display the results  -------------------------------------

@factor_out!!(rhs00){A^{a}}:
@factor_out!!(rhs00){D^{a}};

@factor_out!!(rhs01){A^{a}}:
@factor_out!!(rhs01){D^{a}};
```

```
@factor_out!!(rhs02){A^{a}}:
@factor_out!!(rhs02){D^{a}};

@factor_out!!(rhs03){A^{a}}:
@factor_out!!(rhs03){D^{a}};
```

$$rhs00 := v_a A^a$$

$$rhs01 := A^a D^b \left(\partial_b v_a - \Gamma^c{}_{ab} v_c\right)$$

$$rhs02 := A^a D^b D^c \left(-2\,\Gamma^d{}_{ab}\partial_c v_d - \Gamma^d{}_{bc}\partial_d v_a + \partial_{bc} v_a + \Gamma^d{}_{ab}\Gamma^e{}_{dc} v_e + \Gamma^d{}_{ae}\Gamma^e{}_{bc} v_d - \partial_b \Gamma^d{}_{ac} v_d\right)$$

$$\begin{aligned} rhs03 := A^a D^b D^c D^d \big(&3\,\Gamma^e{}_{ab}\Gamma^f{}_{ec}\partial_d v_f + 3\,\Gamma^e{}_{af}\Gamma^f{}_{bc}\partial_d v_e + 3\,\Gamma^e{}_{ab}\Gamma^f{}_{cd}\partial_f v_e - 3\,\partial_b\Gamma^e{}_{ac}\partial_d v_e - 3\,\Gamma^e{}_{ab}\partial_{cd} v_e \\ &+ \Gamma^e{}_{bc}\Gamma^f{}_{ed}\partial_f v_a + \Gamma^e{}_{bc}\Gamma^f{}_{de}\partial_f v_a - \partial_b\Gamma^e{}_{cd}\partial_e v_a - 3\,\Gamma^e{}_{bc}\partial_{de} v_a + \partial_{bcd} v_a - \Gamma^e{}_{ab}\Gamma^f{}_{ec}\Gamma^g{}_{fd} v_g \\ &- \Gamma^e{}_{af}\Gamma^f{}_{bc}\Gamma^g{}_{ed} v_g - 2\,\Gamma^e{}_{ab}\Gamma^f{}_{cd}\Gamma^g{}_{ef} v_g + \Gamma^e{}_{fb}\partial_c\Gamma^f{}_{ad} v_e + 2\,\Gamma^e{}_{ab}\partial_c\Gamma^f{}_{ed} v_f - \Gamma^e{}_{af}\Gamma^f{}_{gb}\Gamma^g{}_{cd} v_e \\ &- \Gamma^e{}_{af}\Gamma^f{}_{bg}\Gamma^g{}_{cd} v_e + 2\,\Gamma^e{}_{bc}\partial_d\Gamma^f{}_{ae} v_f + \Gamma^e{}_{af}\partial_b\Gamma^f{}_{cd} v_e + \Gamma^e{}_{bc}\partial_e\Gamma^f{}_{ad} v_f - \partial_{bc}\Gamma^e{}_{ad} v_e\big) \end{aligned}$$

Covariant derivatives and the Riemann tensor

```
# --- some basic declarations ---------------------------------------------

\partial{#}::PartialDerivative.

A^{a}::Depends(\partial).
D^{a}::Depends(\partial).
E^{a}::Depends(\partial).
v_{a}::Depends(\partial).
\Gamma^{a}_{b c}::Depends(\partial).
\Gamma^{a}_{b c}::TableauSymmetry(shape={2}, indices={1,2}).

# --- construct the scalar v_{a} A^{a}  -----------------------------------

scalar:=v_{a} A^{a}:

# --- compute the covariant derivative in the direction of D^a  -----------

derivD:=D^{c}\partial_{c}{@(scalar)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@substitute!(%)(D^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}D^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):

# --- compute the covariant derivative in the direction of E^a  -----------

derivDE:=E^{c}\partial_{c}{@(derivD)}:
@distribute!(%):
@prodrule!(%):
@distribute!(%):
@substitute!(%)(E^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}E^{c}):
@substitute!(%)(E^{a}\partial_{a}{D^{b}} -> -\Gamma^{b}_{a c}D^{a}E^{c}):
@substitute!(%)(E^{a}\partial_{a}{E^{b}} -> -\Gamma^{b}_{a c}E^{a}E^{c}):
@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%);
```

```
# --- copy to derivED then swap the order of the derivatives ------------------

derivED:=@(derivDE):

@substitute!(%)(E^{a} -> F^{a}):
@substitute!(%)(D^{a} -> E^{a}):
@substitute!(%)(F^{a} -> D^{a});

# --- subtract and simplify ---------------------------------------------------

diff:=@(derivDE) - @(derivED):

{A^{a},D^{a},E^{a},v_{a},\Gamma^{a}_{b c}}::SortOrder.

@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):
@collect_terms!(%);

# --- tidy up and display the results  ----------------------------------------

@factor_out!(%){A^{a}}:
@factor_out!(%){D^{a}}:
@factor_out!(%){E^{a}}:
@factor_out!(%){v_{a}};

# --- Cadabra output is printed only when a %! command is given ---------------

@print["A^a D^b E^c (v_{a;b;c}-v_{a;c;b})="~@(diff)];
```

$$\begin{aligned} derivDE :=& -A^{a} D^{b} E^{c} \Gamma^{d}{}_{ac} \partial_{b} v_{d} - A^{a} D^{b} E^{c} \Gamma^{d}{}_{bc} \partial_{d} v_{a} + A^{a} D^{b} E^{c} \partial_{bc} v_{a} + A^{a} D^{b} E^{c} \Gamma^{d}{}_{ac} \Gamma^{e}{}_{bd} v_{e} \\ & + A^{a} D^{b} E^{c} \Gamma^{d}{}_{ae} \Gamma^{e}{}_{bc} v_{d} - A^{a} D^{b} E^{c} \partial_{c} \Gamma^{d}{}_{ab} v_{d} - A^{a} D^{b} E^{c} \Gamma^{d}{}_{ab} \partial_{c} v_{d} \end{aligned}$$

$$\begin{aligned} derivED :=& -A^{a} E^{b} D^{c} \Gamma^{d}{}_{ac} \partial_{b} v_{d} - A^{a} E^{b} D^{c} \Gamma^{d}{}_{bc} \partial_{d} v_{a} + A^{a} E^{b} D^{c} \partial_{bc} v_{a} + A^{a} E^{b} D^{c} \Gamma^{d}{}_{ac} \Gamma^{e}{}_{bd} v_{e} \\ & + A^{a} E^{b} D^{c} \Gamma^{d}{}_{ae} \Gamma^{e}{}_{bc} v_{d} - A^{a} E^{b} D^{c} \partial_{c} \Gamma^{d}{}_{ab} v_{d} - A^{a} E^{b} D^{c} \Gamma^{d}{}_{ab} \partial_{c} v_{d} \end{aligned}$$

$$diff := A^{a} D^{b} E^{c} v_{d} \Gamma^{d}{}_{be} \Gamma^{e}{}_{ac} - A^{a} D^{b} E^{c} v_{d} \partial_{c} \Gamma^{d}{}_{ab} - A^{a} D^{b} E^{c} v_{d} \Gamma^{d}{}_{ce} \Gamma^{e}{}_{ab} + A^{a} D^{b} E^{c} v_{d} \partial_{b} \Gamma^{d}{}_{ac}$$

$$diff := A^{a} D^{b} E^{c} v_{d} \left(\Gamma^{d}{}_{be} \Gamma^{e}{}_{ac} - \partial_{c} \Gamma^{d}{}_{ab} - \Gamma^{d}{}_{ce} \Gamma^{e}{}_{ab} + \partial_{b} \Gamma^{d}{}_{ac} \right)$$

$$A^{a} D^{b} E^{c} \left(v_{a;b;c} - v_{a;c;b} \right) = A^{a} D^{b} E^{c} v_{d} \left(\Gamma^{d}{}_{be} \Gamma^{e}{}_{ac} - \partial_{c} \Gamma^{d}{}_{ab} - \Gamma^{d}{}_{ce} \Gamma^{e}{}_{ab} + \partial_{b} \Gamma^{d}{}_{ac} \right)$$

Printing real equations

That's all very nice, but how do I get real equations printed? (i.e. with both the left and right hand sides looking like normal mathematics). Simply use Cadabra's @print algorithm. Here is how we could could display some of the results from the previous computation.

```
@print["v_{a;b}A^aD^b="~@(rhs01)];
@print["v_{a;bc}A^aD^bD^c="~@(rhs02)];
```

$$v_{a;b} A^{a} D^{b} = A^{a} D^{b} \left(\partial_{b} v_{a} - \Gamma^{c}{}_{ab} v_{c} \right)$$

$$v_{a;bc} A^{a} D^{b} D^{c} = A^{a} D^{b} D^{c} \left(-2 \Gamma^{d}{}_{ab} \partial_{c} v_{d} - \Gamma^{d}{}_{bc} \partial_{d} v_{a} + \partial_{bc} v_{a} + \Gamma^{d}{}_{ab} \Gamma^{e}{}_{dc} v_{e} + \Gamma^{d}{}_{ae} \Gamma^{e}{}_{bc} v_{d} - \partial_{b} \Gamma^{d}{}_{ac} v_{d} \right)$$

Libraries

Cadabra has a feature where the output of a single expression can be sent to a file. Here is an example.

```
rhs:=@(rhs02); "rhs.txt"
```

The above will save the output in the file `rhs.txt`. Can we save many expressions in one file? No, the above syntax (as far as I can tell) limits you to one expression per file.

I've written a shell script and a support program that allow us to create libraries of useful cadabra expressions. These libraries can be used by other cadabra notebooks. Here is an example where a new library is created and two expressions are exported.

```
# --- create two new expressions and save their output

result01:=@(rhs01); "result01.txt"
result02:=@(rhs02); "result02.txt"

# --- create a library, this deletes any previous libray named sample.lib

com:="open sample.lib":
@run(com){"/Users/leo/local/sh/cdbfile"}:

# --- append result01.txt and result02.txt to the library and delete the .txt files

com:="export sample.lib result01.txt":
@run(com){"/Users/leo/local/sh/cdbfile"}:

com:="export sample.lib result02.txt":
@run(com){"/Users/leo/local/sh/cdbfile"}:
```

Note that when the export command finishes it will also delete the file it read from (for example `result01.txt` and `result02.txt`). The symbol `com` must be created before the `@run` command as it forms the link between Cadabra and the unix shell. It first contains the arguments to be passed to the shell script and later it receives the shell script's output.

You can also import expressions from other libraries

```
# --- import metric from metric.lib

metric:="import metric.lib metric":
@run(metric){"/Users/leo/local/sh/cdbfile"}:

# --- import the inverse metric from metric-inv.lib

inverse:="import metric-inv.lib invMetric":
@run(inverse){"/Users/leo/local/sh/cdbfile"};
```

Notice the change of name in the above example. The inverse metric was known as `invMetric` inside the library `metric-inv.lib` but in this notebook we will use the name `inverse`. Just to prove that we have sucessfully imported the expression, here it is

$$inverse := g^{ab} + \frac{1}{3}\, x^c x^d R^a{}_c{}^b{}_d + \frac{1}{6}\, x^c x^d x^e \nabla_c R^a{}_d{}^b{}_e$$

Configuration

Each TeX file created by `cdbmerge` will begin with

```
\documentclass{cadabra}
```

You are welcome to cutsomise the `cadabra.cls` class to whatever suits your fancy. But such changes will effect every TeX file (well just those that use the `cdbmerge` program). If you want to make changes that apply to

one file, say `fred.tex`, then you can create a file `fred.cfg` and place in it any set of LaTeX commands. The cadabra class will instruct LaTeX to read this file (if it exists) as the last step in processing the `\documentclass` command.

One useful macro that is defined in `cadabra.cls` is the `\papersize` macro. It takes six arguments being the height, width, left right, top and bottom margin sizes. For exampe,

```
\papersize{210mm}{297mm}{2cm}{2cm}{2cm}{2cm}}
```

will produce an A4 paper in landsacpe form with 2cm margins along all four edges.

One limitation of the `cdbmerge` program is that it does not respect the cadabra attribute LaTeXForm. That is, if you use

```
\pd::LaTeXForm{\partial}.
```

in your cadabra code you will find that LaTeX will complain about the undefined symbol `\pd`. This problem is easily fixed by including

```
\def\pd{\partial}
```

in your configuration file (e.g. `fred.cfg`).