

This differs from source-alt.cdbp in that here we use TableauSymmetry to set the symmetry on the connection.

```
# =====
# The connection
# =====

# --- The metric connection -----

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

g_{a b}::Metric.

\partial_{#}::PartialDerivative.

cderiv:=\partial_{c}{g_{a b}} - g_{a d}\Gamma^d_{b c}
        - g_{d b}\Gamma^d_{a c};

Gamma:=\Gamma^a_{b c} -> (1/2) g^a d ( \partial_{b}{g_{d c}}
        + \partial_{c}{g_{b d}}
        - \partial_{d}{g_{b c}} );

@substitute!(cderiv)(@(Gamma));

@distribute! (%);
@eliminate_metric! (%);
@canonicalise! (%);
@collect_terms! (%);
```

$$cderiv := \partial_c g_{ab} - g_{ad} \Gamma^d_{bc} - g_{db} \Gamma^d_{ac}$$

$$Gamma := \Gamma^a_{bc} \rightarrow \frac{1}{2} g^{ad} (\partial_b g_{dc} + \partial_c g_{bd} - \partial_d g_{bc})$$

$$cderiv := \partial_c g_{ab} - \frac{1}{2} g_{ad} g^{de} (\partial_b g_{ec} + \partial_c g_{be} - \partial_e g_{bc}) - \frac{1}{2} g_{db} g^{de} (\partial_a g_{ec} + \partial_c g_{ae} - \partial_e g_{ac})$$

$$cderiv := \partial_c g_{ab} - \frac{1}{2} g_{ad} g^{de} \partial_b g_{ec} - \frac{1}{2} g_{ad} g^{de} \partial_c g_{be} + \frac{1}{2} g_{ad} g^{de} \partial_e g_{bc} - \frac{1}{2} g_{db} g^{de} \partial_a g_{ec} - \frac{1}{2} g_{db} g^{de} \partial_c g_{ae} + \frac{1}{2} g_{db} g^{de} \partial_e g_{ac}$$

$$cderiv := \partial_c g_{ab} - \frac{1}{2} \partial_b g_{ac} - \frac{1}{2} \partial_c g_{ba} + \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_b g_{ac}$$

$$cderiv := \partial_c g_{ab} - \frac{1}{2} \partial_b g_{ac} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_a g_{bc} - \frac{1}{2} \partial_c g_{ab} + \frac{1}{2} \partial_b g_{ac}$$

$$cderiv := 0$$

```

# =====
# One covariant derivative
# =====

@reset.

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

\partial_{#}::PartialDerivative.

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

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

# --- tidy up and display the results -----
@factor_out!(scalar){A^{a}}:
@factor_out!(scalar){D^{a}};

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

```

$$scalar := v_a A^a$$

$$derivD := A^a D^b (\partial_b v_a - \Gamma^b_{ac} v_b)$$

```

# =====
# Commutation of covariant derivatives and the Riemann tensor
# =====

@reset.

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

\partial_{#}::PartialDerivative.

# --- force Gamma to be symmetric in its lower two indices -----
\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)}:
@distributed! (%):
@prodrule! (%):
@distributed! (%):
@substitute! (%) (D^{a}\partial_{a}{A^{b}} -> -\Gamma^{b}_{a c}A^{a}D^{c}):
@prodsort! (%):
@rename_dummies! (%):
@canonicalise! (%):

# --- compute the covariant derivative in the direction of E^a -----
derivDE:=E^{c}\partial_{c}{@(derivD)}:
@distributed! (%):
@prodrule! (%):
@distributed! (%):
@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}):
@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}):

# --- compute difference in mixed covariant derivatives -----
diff:=@(derivDE) - @(derivED):

# --- tidy up and display the results -----
{A^{a},D^{a},E^{a},v_{a},\Gamma^{a}_{b c}}::SortOrder.

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

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

```

```
@factor_out!({v_{a}});
```

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

$$\begin{aligned} derivDE := & -A^a D^b E^c \Gamma^d_{ac} \partial_b v_d - A^a D^b E^c \Gamma^d_{bc} \partial_d v_a + A^a D^b E^c \partial_{bc} v_a + A^a D^b E^c \Gamma^d_{ac} \Gamma^e_{bd} v_e \\ & + A^a D^b E^c \Gamma^d_{ae} \Gamma^e_{bc} v_d - A^a D^b E^c \partial_c \Gamma^d_{ab} v_d - A^a D^b E^c \Gamma^d_{ab} \partial_c v_d \end{aligned}$$

$$\begin{aligned} derivED := & -A^a E^b D^c \Gamma^d_{ac} \partial_b v_d - A^a E^b D^c \Gamma^d_{bc} \partial_d v_a + A^a E^b D^c \partial_{bc} v_a + A^a E^b D^c \Gamma^d_{ac} \Gamma^e_{bd} v_e \\ & + A^a E^b D^c \Gamma^d_{ae} \Gamma^e_{bc} v_d - A^a E^b D^c \partial_c \Gamma^d_{ab} v_d - A^a E^b D^c \Gamma^d_{ab} \partial_c v_d \end{aligned}$$

$$diff := A^a D^b E^c v_d (\Gamma^d_{be} \Gamma^e_{ac} - \partial_c \Gamma^d_{ab} - \Gamma^d_{ce} \Gamma^e_{ab} + \partial_b \Gamma^d_{ac})$$

$$A^a D^b E^c (v_{a;b;c} - v_{a;c;b}) = A^a D^b E^c v_d (\Gamma^d_{be} \Gamma^e_{ac} - \partial_c \Gamma^d_{ab} - \Gamma^d_{ce} \Gamma^e_{ab} + \partial_b \Gamma^d_{ac})$$

```

# =====
# Gauss equation for 3d/4d curvatures
# =====

@reset.

# --- The Gauss equation -----

::PostDefaultRules( @@collect_terms!(%) ).
{a,b,c,d,e,f,g,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Indices(position=fixed).

\nabla_{#}::Derivative.

K_{a b}::Symmetric.
g^{a}_{b}::KroneckerDelta.

# --- define the projection operator -----
hab:=h^{a}_{b} -> g^{a}_{b} - n^{a} n_{b}:

# --- 3-covariant derivative obtained by projection on 4-covariant derivative -
vpq:=v_{p q} -> h^{a}_{p}h^{b}_{q}\nabla_{b}\{v_{a}\}:

# --- compute 3-curvature by commutation of covariant derivatives -----
vpqr:=h^{a}_{p}h^{b}_{q}h^{c}_{r} ( \nabla_{c}\{v_{a b}\} - \nabla_{b}\{v_{a c}\} ):

@substitute!(vpq)(@(hab)):
@substitute!(vpqr)(@(vpq)):

@distribute!(%):
@prodrule!(%):
@distribute!(%):
@eliminate_kr!(%):

# --- standard substitutions -----
@substitute!(%)(h^{a}_{b} n^{b} -> 0):
@substitute!(%)(h^{a}_{b} n_{a} -> 0):
@substitute!(%)(\nabla_{a}\{g^{b}_{c}\} -> 0):
@substitute!(%)(n^{a}\nabla_{b}\{v_{a}\} -> -v_{a}\nabla_{b}\{n^{a}\}):
@substitute!(%)(v_{a}\nabla_{b}\{n^{a}\} -> v_{p}h^{p}_{a}\nabla_{b}\{n^{a}\}):
@substitute!(%)(h^{p}_{a}h^{q}_{b}\nabla_{p}\{n_{q}\} -> K_{a b}):
@substitute!(%)(h^{p}_{a}h^{q}_{b}\nabla_{p}\{n^{b}\} -> K_{a}^{q}):

# --- tidy up and display the results -----
{h^{a}_{b},\nabla_{a}\{v_{b}\}}::SortOrder.

@prodsort!(%):
@rename_dummies!(%):
@canonicalise!(%):
@factor_out!!(%) {h^{a}_{b}};

```

$$vpqr := h^a_p h^b_q h^c_r (\nabla_c \nabla_b v_a - \nabla_b \nabla_c v_a) + K_{pr} K_q^a v_a - K_{pq} K_r^a v_a$$