```
::KeepHistory(false).
::PostDefaultRules( @@collect_terms!(%), @@sumflatten!(%) ).

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

\nabla_{#}::PartialDerivative.

g_{a b}::Metric.
R_{a b c d}::RiemannTensor.
R^{a}_{b c d}::RiemannTensor.
R^{a}_{b c}^{d}::RiemannTensor.
R^{a}_{b}^{c d}::RiemannTensor.
# --- imported from geodesic-bvp.lib -----------------------------------------------------------
# --- the Q are shorthand for the genGamma, saves typing ---------------------------------------
Q^{a}_{b c}::TableauSymmetry(shape={2}, indices={1,2}).
Q^{a}_{b c d}::TableauSymmetry(shape={3}, indices={1,2,3}).
Q^{a}_{b c d e}::TableauSymmetry(shape={4}, indices={1,2,3,4}).
Q^{a}_{b c d e f}::TableauSymmetry(shape={5}, indices={1,2,3,4,5}).

yGamma:="import geodesic-bvp.lib y05Gamma":
@run(yGamma){"/Users/leo/local/sh/cdbfile"}:
# --- imported from metric.lib -----------------------------------------------------------------
gabcd:="import metric.lib gabcd":
@run(gabcd){"/Users/leo/local/sh/cdbfile"}:

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

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

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

g_{a b c d}::TableauSymmetry(shape={2}, indices={2,3}).
```

```
g_{a b c d e}::TableauSymmetry(shape={3}, indices={2,3,4}).
g_{a b c d e f}::TableauSymmetry(shape={4}, indices={2,3,4,5}).
g_{a b c d e f g}::TableauSymmetry(shape={5}, indices={2,3,4,5,6}).

metric:=g_{a b} + g_{a b c d} x^c x^d
                + g_{a b c d e} x^c x^d x^e
                + g_{a b c d e f} x^c x^d x^e x^f
                + g_{a b c d e f g} x^c x^d x^e x^f x^g:

# --- compute lsq ------------------------------------------------------------------------

{g_{a b},g_{a b c d},g_{a b c d e},g_{a b c d e f},g_{a b c d e f g},
 Q^{a}_{b c},Q^{a}_{b c d},Q^{a}_{b c d e},Q^{a}_{b c d e f},x^{a},Dx^{a}}::SortOrder.

lsq:=g_{a b} y^{a} y^{b}:

@substitute!(lsq)(y^{a} -> @(yGamma)):
@substitute!(lsq)(g_{a b} -> @(metric)):
@distribute!(lsq):

# ************************************************************************************************
# LCB : I use "s s s" for s**3 etc. otherwise @keep_weight returns zero (see bug13.cdbp)
# ************************************************************************************************

# --- truncate lsq to include all terms up to O(\eps^5) ----------------------------------

poly:=@(lsq):

# LCB : I'm too lazy to write \eps so I'll use "s"

@substitute!(poly)(Q^{a}_{b c} -> s s Q^{a}_{b c},
                   Q^{a}_{b c d} -> s s s Q^{a}_{b c d},
                   Q^{a}_{b c d e} -> s s s s Q^{a}_{b c d e},
                   Q^{a}_{b c d e f} -> s s s s s Q^{a}_{b c d e f}):

@substitute!(poly)(g_{a b c d} -> s s g_{a b c d},
                   g_{a b c d e} -> s s s g_{a b c d e},
                   g_{a b c d e f} -> s s s s g_{a b c d e f},
                   g_{a b c d e f g} -> s s s s s g_{a b c d e f g}):

@distribute!(poly):
```

```
s::Weight(label=epsterms,value=1).

term00:=@(poly): @keep_weight!(term00){epsterms}{0}:
term01:=@(poly): @keep_weight!(term01){epsterms}{1}:
term02:=@(poly): @keep_weight!(term02){epsterms}{2}:
term03:=@(poly): @keep_weight!(term03){epsterms}{3}:
term04:=@(poly): @keep_weight!(term04){epsterms}{4}:
term05:=@(poly): @keep_weight!(term05){epsterms}{5}:

# --- done with "s", set it equal to 1

@substitute!(term00)(s -> 1):
@substitute!(term01)(s -> 1):
@substitute!(term02)(s -> 1):
@substitute!(term03)(s -> 1):
@substitute!(term04)(s -> 1):
@substitute!(term05)(s -> 1):

lsq:=@(term00) + @(term01) + @(term02) + @(term03) + @(term04) + @(term05):

@distribute!(lsq):
@prodsort!(lsq):
@rename_dummies!(lsq):
@canonicalise!(lsq):

# ============================================================================================
# By inspection of lsq we see that there eleven quadratic terms, five of the form Q.Q
# and six of the form g.Q  Following the method used in gedodesic-bvp we will process each
# quadratic term separately.
#
# Note : these Q.Q terms maye be the same as those used in geodesic-bvp but rather than importing
# those Q.Qs that are the same I will simply re-compute all of them here. Hang the cost!
# ============================================================================================

QQ1:=Q^{a}_{c d} Q^{c}_{e f}:
QQ2:=Q^{a}_{c d} Q^{b}_{e f}:
QQ3:=Q^{a}_{c d} Q^{c}_{e f g}:
QQ4:=Q^{c}_{d e} Q^{a}_{c f g}:
QQ5:=Q^{a}_{c d} Q^{b}_{e f g}:
```

```
gQ1:=g_{a b c d} Q^{b}_{e f}:
gQ2:=g_{a b c d} Q^{a}_{e f}:
gQ3:=g_{a b c d} Q^{b}_{e f g}:
gQ4:=g_{a b c d} Q^{a}_{e f g}:
gQ5:=g_{a b c d e} Q^{b}_{f g}:
gQ6:=g_{a b c d e} Q^{a}_{f g}:

# replace quadratic terms with "simple" terms, drop repeated dummy indices

@substitute!(lsq)(Q^{a}_{c d} Q^{c}_{e f}   -> QQ1^{a}_{d e f}  ,
                  Q^{a}_{c d} Q^{b}_{e f}   -> QQ2^{a}_{c d}^{b}_{e f}  ,
                  Q^{a}_{c d} Q^{c}_{e f g} -> QQ3^{a}_{d e f g},
                  Q^{c}_{d e} Q^{a}_{c f g} -> QQ4_{d e}^{a}_{f g},
                  Q^{a}_{c d} Q^{b}_{e f g} -> QQ5^{a}_{c d}^{b}_{e f g}):

@substitute!(lsq)(g_{a b c d} Q^{b}_{e f}   -> gQ1_{a c d e f}  ,
                  g_{a b c d} Q^{a}_{e f}   -> gQ2_{b c d e f}  ,
                  g_{a b c d} Q^{b}_{e f g} -> gQ3_{a c d e f g},
                  g_{a b c d} Q^{a}_{e f g} -> gQ4_{b c d e f g},
                  g_{a b c d e} Q^{b}_{f g} -> gQ5_{a c d e f g},
                  g_{a b c d e} Q^{a}_{f g} -> gQ6_{b c d e f g}):

# --- imported from geodesic-ivp.lib -----------------------------------------------------------

genGamma02:="import geodesic-ivp.lib genGamma02":
@run(genGamma02){"/Users/leo/local/sh/cdbfile"}:

genGamma03:="import geodesic-ivp.lib genGamma03":
@run(genGamma03){"/Users/leo/local/sh/cdbfile"}:

genGamma04:="import geodesic-ivp.lib genGamma04":
@run(genGamma04){"/Users/leo/local/sh/cdbfile"}:

genGamma05:="import geodesic-ivp.lib genGamma05":
@run(genGamma05){"/Users/leo/local/sh/cdbfile"}:

genGamma06:="import geodesic-ivp.lib genGamma06":
@run(genGamma06){"/Users/leo/local/sh/cdbfile"}:
```

```
# clear the factor B^{a} in the genGamm0*

@substitute!(genGamma02)(B^{a} -> 1):
@substitute!(genGamma03)(B^{a} -> 1):
@substitute!(genGamma04)(B^{a} -> 1):
@substitute!(genGamma05)(B^{a} -> 1):

# the genGamma0* imported from geodesic-ivp.cdbp are not symmetrised, do so now

@sym(genGamma02){_{b}, _{c}}:
@sym(genGamma03){_{b}, _{c}, _{d}}:

@substitute!(QQ1)(Q^{a}_{b c} -> @(genGamma02)):
@substitute!(QQ2)(Q^{a}_{b c} -> @(genGamma02)):
@substitute!(QQ3)(Q^{a}_{b c} -> @(genGamma02), Q^{a}_{b c d} -> @(genGamma03)):
@substitute!(QQ4)(Q^{a}_{b c} -> @(genGamma02), Q^{a}_{b c d} -> @(genGamma03)):
@substitute!(QQ5)(Q^{a}_{b c} -> @(genGamma02), Q^{a}_{b c d} -> @(genGamma03)):

@substitute!(gQ1)(Q^{a}_{b c}   -> @(genGamma02), g_{a b c d} -> @(gabcd)):
@substitute!(gQ2)(Q^{a}_{b c}   -> @(genGamma02), g_{a b c d} -> @(gabcd)):
@substitute!(gQ3)(Q^{a}_{b c d} -> @(genGamma03), g_{a b c d} -> @(gabcd)):
@substitute!(gQ4)(Q^{a}_{b c d} -> @(genGamma03), g_{a b c d} -> @(gabcd)):
@substitute!(gQ5)(Q^{a}_{b c}   -> @(genGamma02), g_{a b c d e} -> @(gabcde)):
@substitute!(gQ6)(Q^{a}_{b c}   -> @(genGamma02), g_{a b c d e} -> @(gabcde)):

# --- to retain 5-th order curvature terms, we can truncate QQ1,QQ2 at O(x^3) and QQ3,QQ4,QQ5 at O(x^2) ---

# --- truncate QQ1 to include O(x^3)

poly:=@(QQ1):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:
term03:=@(poly): @keep_weight!(term03){xterms}{3}:

QQ1:=@(term00) + @(term01) + @(term02) + @(term03):
```

```
# --- truncate QQ2 to include O(x^3)

poly:=@(QQ2):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:
term03:=@(poly): @keep_weight!(term03){xterms}{3}:

QQ2:=@(term00) + @(term01) + @(term02) + @(term03):

# --- truncate QQ3 to include O(x^2)

poly:=@(QQ3):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

QQ3:=@(term00) + @(term01) + @(term02):

# --- truncate QQ4 to include O(x^2)

poly:=@(QQ4):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

QQ4:=@(term00) + @(term01) + @(term02):

# --- truncate QQ5 to include O(x^2)
```

```
poly:=@(QQ5):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

QQ5:=@(term00) + @(term01) + @(term02):

# --- to retain 5-th order curvature terms, we can truncate qQ1,2,3,4 at O(x^2) and gQ5,6 at O(x^1) ---

# --- truncate qQ1 to include O(x^2)

poly:=@(gQ1):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

gQ1:=@(term00) + @(term01) + @(term02):

# --- truncate qQ2 to include O(x^2)

poly:=@(gQ2):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

gQ2:=@(term00) + @(term01) + @(term02):

# --- truncate qQ3 to include O(x^2)
```

```
poly:=@(gQ3):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

gQ3:=@(term00) + @(term01) + @(term02):

# --- truncate qQ4 to include O(x^2)

poly:=@(gQ4):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:
term02:=@(poly): @keep_weight!(term02){xterms}{2}:

gQ4:=@(term00) + @(term01) + @(term02):

# --- truncate qQ5 to include O(x^1)

poly:=@(gQ5):
@distribute!(%):

x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:

gQ5:=@(term00) + @(term01):

# --- truncate qQ6 to include O(x^1)

poly:=@(gQ6):
@distribute!(%):
```

```
x^{a}::Weight(label=xterms,value=1).

term00:=@(poly): @keep_weight!(term00){xterms}{0}:
term01:=@(poly): @keep_weight!(term01){xterms}{1}:

gQ6:=@(term00) + @(term01):

# --- return to lsq, substitute and simplify ----------------------------------------------------

{R_{a b c d},\nabla_{e}{R_{a b c d}},
             \nabla_{e f}{R_{a b c d}},
             \nabla_{e f g}{R_{a b c d}},x^{a},Dx^{a}}::SortOrder.

@substitute!(lsq)(Q^{a}_{b c} -> @(genGamma02),
                  Q^{a}_{b c d} -> @(genGamma03),
                  Q^{a}_{b c d e} -> @(genGamma04),
                  Q^{a}_{b c d e f} -> @(genGamma05),
                  g_{a b c d} -> @(gabcd),
                  g_{a b c d e} -> @(gabcde),
                  g_{a b c d e f} -> @(gabcdef),
                  g_{a b c d e f g} -> @(gabcdefg),
                  QQ1^{a}_{d e f}           -> @(QQ1),
                  QQ2^{a}_{c d}^{b}_{e f}   -> @(QQ2),
                  QQ3^{a}_{d e f g}         -> @(QQ3),
                  QQ4_{d e}^{a}_{f g}       -> @(QQ4),
                  QQ5^{a}_{c d}^{b}_{e f g} -> @(QQ5),
                  gQ1_{a c d e f}           -> @(gQ1),
                  gQ2_{b c d e f}           -> @(gQ2),
                  gQ3_{a c d e f g}         -> @(gQ3),
                  gQ4_{b c d e f g}         -> @(gQ4),
                  gQ5_{a c d e f g}         -> @(gQ5),
                  gQ6_{b c d e f g}         -> @(gQ6)):

@distribute!(%): @eliminate_metric!(%): @prodsort!(%): @rename_dummies!(%): @canonicalise!(%):

# --- reorder lsq into successively higher orders in the curvatures -----------------------------

poly:=@(lsq):

@substitute!(poly)(R^{a}_{b c d} -> R_{a b c d}):
```

```
x^{a}::Weight(label=xDxterms,value=1).
Dx^{a}::Weight(label=xDxterms,value=1).

term00:=@(poly): @keep_weight!(term00){xDxterms}{0}:
term01:=@(poly): @keep_weight!(term01){xDxterms}{1}:
term02:=@(poly): @keep_weight!(term02){xDxterms}{2}:
term03:=@(poly): @keep_weight!(term03){xDxterms}{3}:
term04:=@(poly): @keep_weight!(term04){xDxterms}{4}:
term05:=@(poly): @keep_weight!(term05){xDxterms}{5}:
term06:=@(poly): @keep_weight!(term06){xDxterms}{6}:
term07:=@(poly): @keep_weight!(term07){xDxterms}{7}:

@prodsort!(term00): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term01): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term02): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term03): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term04): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term05): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term06): @rename_dummies!(%): @canonicalise!(%):
@prodsort!(term07): @rename_dummies!(%): @canonicalise!(%):

lsq:=@(term00) + @(term01) + @(term02) + @(term03) + @(term04) + @(term05) + @(term06) + @(term07):

@substitute!(lsq)(Dx^{a} -> \Delta{x}^{a}):
@substitute!(lsq)(Dx_{a} -> g_{b a} \Delta{x}^{b}):
@rename_dummies!(lsq); "lsq.trn"

@print["\Btag{01}\intPQ="~@(lsq)~"+\BigO{\eps^6}\Etag{01}"];
```

```
# ============================================================================================
#   The squared geodesic length
# ============================================================================================
```

The squared geodesic length between the points P and Q with coordinates x^a and $x^a + \Delta x^a$ is

$$\left(\int_P^Q ds\right)^2 = \Bigg(g_{ab}\Delta x^{\,a}\Delta x^{\,b} - \frac{1}{3}\,R_{abcd}x^a x^c \Delta x^{\,b}\Delta x^{\,d} - \frac{1}{12}\,\nabla_a R_{bcde}x^b x^d \Delta x^{\,a}\Delta x^{\,c}\Delta x^{\,e} - \frac{1}{6}\,\nabla_a R_{bcde}x^a x^b x^d \Delta x^{\,c}\Delta x^{\,e} + \frac{2}{45}\,R_{abcd}R_{aefg}x^b x^c x^f \Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,g}$$
$$- \frac{1}{20}\,\nabla_{ab}R_{cdef}x^a x^c x^e \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,f} - \frac{11}{45}\,R_{abcd}R_{aefg}x^c x^f \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,g} - \frac{1}{60}\,\nabla_{ab}R_{cdef}x^c x^e \Delta x^{\,a}\Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,f} + \frac{2}{45}\,R_{abcd}R_{aefg}x^b x^c x^e x^f \Delta x^{\,d}\Delta x^{\,g}$$
$$- \frac{1}{20}\,\nabla_{ab}R_{cdef}x^a x^b x^c x^e \Delta x^{\,d}\Delta x^{\,f} + \frac{1}{45}\,R_{abcd}\nabla_e R_{afgh}x^c x^e x^f x^g \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,h} + \frac{1}{45}\,R_{abcd}\nabla_e R_{afgh}x^b x^c x^f x^g \Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,h}$$
$$+ \frac{1}{45}\,R_{abcd}\nabla_e R_{afgh}x^b x^c x^e x^g \Delta x^{\,d}\Delta x^{\,f}\Delta x^{\,h} - \frac{1}{60}\,\nabla_{abc}R_{defg}x^a x^b x^d x^f \Delta x^{\,c}\Delta x^{\,e}\Delta x^{\,g} - \frac{2}{27}\,R_{abcd}\nabla_e R_{afgh}x^c x^f x^g \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,h}$$
$$- \frac{1}{12}\,R_{abcd}\nabla_a R_{efgh}x^c x^e x^g \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,f}\Delta x^{\,h} - \frac{5}{27}\,R_{abcd}\nabla_e R_{afgh}x^c x^e x^g \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,f}\Delta x^{\,h} + \frac{1}{54}\,R_{abcd}\nabla_e R_{afgh}x^c x^g \Delta x^{\,b}\Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,f}\Delta x^{\,h}$$
$$+ \frac{1}{18}\,R_{abcd}\nabla_e R_{afgh}x^b x^c x^g \Delta x^{\,d}\Delta x^{\,e}\Delta x^{\,f}\Delta x^{\,h} + \frac{2}{45}\,R_{abcd}\nabla_e R_{afgh}x^b x^c x^e x^f x^g \Delta x^{\,d}\Delta x^{\,h} - \frac{1}{90}\,\nabla_{abc}R_{defg}x^a x^b x^c x^d x^f \Delta x^{\,e}\Delta x^{\,g}\Bigg) + \mathcal{O}\left(\epsilon^6\right)$$

```
# ============================================================================================
#   export lsq for use by truncate.cdbp
# ============================================================================================

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

com:="export geodesic-lsq.lib lsq.trn":
@run(com){"/Users/leo/local/sh/cdbfile"}:
```