

Geodesic arc-length

Given a pair of points P and Q the geodesic arc-length can be computed using

$$L_{PQ} = \int_P^Q \left(g_{ab}(x) \frac{dx^a}{ds} \frac{dx^b}{ds} \right)^{1/2} ds \quad (1)$$

Since the path is a geodesic the integrand is constant and thus

$$L_{PQ}^2 = g_{ab}(x) \frac{dx^a}{ds} \frac{dx^b}{ds} \Big|_P \quad (2)$$

where s is a re-scaled parameter (0 at P and 1 at Q). The point P has RNC coordinates x^a while the point Q has coordinates $x^a + Dx^a$.

The vector dx^a/ds at P is given by the solution of the geodesic boundary value problem. This was found in the previous code (`geodesic-bvp`). That is

$$\frac{dx^b}{ds} \Big|_P = y^a \quad (3)$$

and thus

$$L_{PQ}^2 = g_{ab}(x) y^a y^b \quad (4)$$

It is possible to directly evaluate the right hand side of (4) using the results from the `geodesic-bvp` and `metric` codes. The result would need to be truncated (to an order consistent with the results from those codes). But doing so would be computationally expensive as at least half of the terms will be thrown away. A better approach is compute just the terms that will survive the truncation. This is done by expanding $g_{ab}(x)$ and y^a as a truncated series in the curvatures and its derivatives.

The $g_{ab}(x)$ and y^a are written in a (truncated) formal power series in the curvature and its derivatives

$$y^a = \overset{0}{y}^a + \overset{2}{y}^a + \overset{3}{y}^a + \overset{4}{y}^a + \overset{5}{y}^a + \mathcal{O}(\epsilon^6) \quad (5)$$

$$g_{ab}(x) = \overset{0}{g}_{ab} + \overset{2}{g}_{ab} + \overset{3}{g}_{ab} + \overset{4}{g}_{ab} + \overset{5}{g}_{ab} + \mathcal{O}(\epsilon^6) \quad (6)$$

Note that this use of $\overset{i}{y}$ differs from that used in `geodesic-bvp`. Here the index above y^a denotes a particular term in the curvature expansion while in `geodesic-bvp` the index denoted the iteration number (in the fixed point scheme used to solve the BVP for y^a).

Stage 1

The formal curvature expansions are substituted into equation (4), expanded and truncated to retain terms of order $\mathcal{O}(\epsilon^5)$ or less. The expansion to 4th order terms is as follows.

$$L_{PQ}^2 = {}^0g_{ab}{}^0y^a{}^0y^b + 2{}^0g_{ab}{}^0y^a{}^2y^b + {}^2g_{ab}{}^0y^a{}^0y^b + 2{}^0g_{ab}{}^0y^a{}^3y^b + {}^3g_{ab}{}^0y^a{}^0y^b + 2{}^0g_{ab}{}^0y^a{}^4y^b + {}^0g_{ab}{}^2y^a{}^2y^b + 2{}^2g_{ab}{}^0y^a{}^2y^b + {}^4g_{ab}{}^0y^a{}^0y^b$$

From `geodesic-bvp` (actually from `rnc2rnc` which reformatted the results nicely) we have

$${}^0y^a = Dx^a$$

$${}^2y^a = -\frac{1}{3}x^b Dx^c Dx^d g^{ae} R_{bcde}$$

$${}^3y^a = x^b x^c Dx^d Dx^e \left(-\frac{1}{12} g^{af} \nabla_d R_{becf} - \frac{1}{6} g^{af} \nabla_b R_{cdef} + \frac{1}{24} g^{af} \nabla_f R_{bdce} \right) - \frac{1}{12} x^b Dx^c Dx^d Dx^e g^{af} \nabla_c R_{bdef}$$

$$\begin{aligned} {}^4y^a = & x^b x^c Dx^d Dx^e Dx^f \left(-\frac{2}{45} g^{ag} g^{hi} R_{bdeh} R_{cfdgi} + \frac{1}{45} g^{ag} g^{hi} R_{bdeh} R_{cifgi} - \frac{1}{45} g^{ag} g^{hi} R_{bdeh} R_{cgfhi} + \frac{1}{45} g^{ag} g^{hi} R_{bdch} R_{egfhi} - \frac{1}{60} g^{ag} \nabla_{de} R_{bfch} \right. \\ & \left. - \frac{1}{40} g^{ag} \nabla_{db} R_{cefg} - \frac{1}{40} g^{ag} \nabla_{bd} R_{cefg} + \frac{1}{240} g^{ag} \nabla_{gd} R_{becf} + \frac{1}{240} g^{ag} \nabla_{dg} R_{becf} \right) \\ & + x^b x^c x^d Dx^e Dx^f \left(-\frac{4}{45} g^{ag} g^{hi} R_{befh} R_{cgdi} + \frac{2}{45} g^{ag} g^{hi} R_{bech} R_{difg} + \frac{1}{45} g^{ag} g^{hi} R_{bech} R_{dghfi} - \frac{1}{40} g^{ag} \nabla_{eb} R_{cfdg} - \frac{1}{40} g^{ag} \nabla_{be} R_{cfdg} - \frac{1}{20} g^{ag} \nabla_{bc} R_{defg} \right. \\ & \left. - \frac{1}{45} g^{ag} g^{hi} R_{bech} R_{dfgi} + \frac{1}{80} g^{ag} \nabla_{gb} R_{cedf} + \frac{1}{80} g^{ag} \nabla_{bg} R_{cedf} \right) + x^b Dx^c Dx^d Dx^e Dx^f \left(-\frac{1}{45} g^{ag} g^{hi} R_{bcdh} R_{egfhi} - \frac{1}{60} g^{ag} \nabla_{cd} R_{befg} \right) \end{aligned}$$

and from `metric` we have

$${}^0g_{ab} = g_{ab}$$

$$3{}^2g_{ab} = -x^c x^d R_{acbd}$$

$$6{}^3g_{ab} = -x^c x^d x^e \nabla_c R_{adbe}$$

$$180{}^4g_{ab} = x^c x^d x^e x^f (8g^{gh} R_{acdgh} R_{befh} - 9\nabla_{cd} R_{aebf})$$

Stage 2

The results from the `geodesic-bvp` and `metric` codes are read to provide values for the \bar{y}^a and \bar{g}_{ab} . These are substituted into the result from Stage 1, et volia, the final answer. To 4th-order terms the result is given by

$$\begin{aligned}
L_{PQ}^2 = & g_{ab} D x^a D x^b - \frac{1}{3} x^a x^b D x^c D x^d R_{acbd} - \frac{1}{12} x^a x^b D x^c D x^d D x^e \nabla_c R_{adbe} - \frac{1}{6} x^a x^b x^c D x^d D x^e \nabla_a R_{bdce} \\
& + \frac{1}{360} x^a x^b D x^c D x^d D x^e D x^f (-8g^{gh} R_{acdg} R_{befh} - 6\nabla_{cd} R_{aebf}) + \frac{1}{360} x^a x^b x^c D x^d D x^e D x^f (16g^{gh} R_{adbg} R_{cefh} - 9\nabla_{da} R_{becf} - 9\nabla_{ad} R_{becf}) \\
& + \frac{1}{360} x^a x^b x^c x^d D x^e D x^f (16g^{gh} R_{aebg} R_{cfdh} - 18\nabla_{ab} R_{cedf}) + \frac{1}{1080} x^a x^b x^c D x^d D x^e D x^f D x^g (-4g^{hi} R_{adeh} \nabla_f R_{bgci} - 24g^{hi} R_{adeh} \nabla_b R_{cfgi} \\
& \quad + 10g^{hi} R_{adeh} \nabla_i R_{bfcg} + 16g^{hi} R_{adbh} \nabla_e R_{cfgi} - 4\nabla_{dea} R_{bfcg} - 4\nabla_{dae} R_{bfcg} - 4\nabla_{ade} R_{bfcg}) \\
& + \frac{1}{1080} x^a x^b D x^c D x^d D x^e D x^f D x^g (-18g^{hi} R_{acdh} \nabla_e R_{bfgi} - 3\nabla_{cde} R_{afbg}) \\
& + \frac{1}{1080} x^a x^b x^c x^d D x^e D x^f D x^g (24g^{hi} R_{aefh} \nabla_b R_{cgdi} + 24g^{hi} R_{aebh} \nabla_f R_{cgdi} + 24g^{hi} R_{aebh} \nabla_c R_{dfgi} - 6\nabla_{eab} R_{cfdg} - 6\nabla_{aeb} R_{cfdg} - 6\nabla_{abe} R_{cfdg}) \\
& + \frac{1}{1080} x^a x^b x^c x^d x^e D x^f D x^g (48g^{hi} R_{afbh} \nabla_c R_{dgei} - 12\nabla_{abc} R_{dfeg}) + \mathcal{O}(\epsilon^5)
\end{aligned}$$

Shared properties

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

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

g_{a b}::Metric.
g^{a b}::InverseMetric.
g_{a}^{b}::KroneckerDelta.
g^{a}_{b}::KroneckerDelta.
\delta^{a}_{b}::KroneckerDelta.
\delta_{a}^{b}::KroneckerDelta.

R_{a b c d}::RiemannTensor.
R^{a}_{b c d}::RiemannTensor.
R_{a b c}^{d}::RiemannTensor.

\Gamma^{a}_{b c}::TableauSymmetry(shape={2}, indices={1,2}).
\Gamma^{a}_{b c d}::TableauSymmetry(shape={3}, indices={1,2,3}).
\Gamma^{a}_{b c d e}::TableauSymmetry(shape={4}, indices={1,2,3,4}).
\Gamma^{a}_{b c d e f}::TableauSymmetry(shape={5}, indices={1,2,3,4,5}).

x^{a}::Depends(D{#}).

g_{a b}::Depends(\partial{#}).
R_{a b c d}::Depends(\partial{#}).
R^{a}_{b c d}::Depends(\partial{#}).
\Gamma^{a}_{b c}::Depends(\partial{#}).

R_{a b c d}::Depends(\nabla{#}).
R^{a}_{b c d}::Depends(\nabla{#}).

g0{#}::LaTeXForm ("\\ngab{0}").
g2{#}::LaTeXForm ("\\ngab{2}").
g3{#}::LaTeXForm ("\\ngab{3}").
g4{#}::LaTeXForm ("\\ngab{4}").
```

```
g5{#}::LaTeXForm ("\\ngab{5}").
```

```
y0{#}::LaTeXForm ("\\ny{0}").
```

```
y2{#}::LaTeXForm ("\\ny{2}").
```

```
y3{#}::LaTeXForm ("\\ny{3}").
```

```
y4{#}::LaTeXForm ("\\ny{4}").
```

```
y5{#}::LaTeXForm ("\\ny{5}").
```

Stage 1: The formal expansion

```
g0_{a b}::Symmetric.
g2_{a b}::Symmetric.
g3_{a b}::Symmetric.
g4_{a b}::Symmetric.
g5_{a b}::Symmetric.

g0_{a b}::Weight(label=num,value=0).
g2_{a b}::Weight(label=num,value=2).
g3_{a b}::Weight(label=num,value=3).
g4_{a b}::Weight(label=num,value=4).
g5_{a b}::Weight(label=num,value=5).

y0^{a}::Weight(label=num,value=0).
y2^{a}::Weight(label=num,value=2).
y3^{a}::Weight(label=num,value=3).
y4^{a}::Weight(label=num,value=4).
y5^{a}::Weight(label=num,value=5).

# note: keeping numbering as is (out of order) to ensure R appears before \nabla R etc.
def product_sort (obj):
    substitute (obj,$ A^{a}          -> A001^{a}          $)
    substitute (obj,$ x^{a}          -> A002^{a}          $)
    substitute (obj,$ Dx^{a}         -> A003^{a}          $)
    substitute (obj,$ g_{a b}        -> A004_{a b}       $)
    substitute (obj,$ g^{a b}        -> A005^{a b}       $)
    substitute (obj,$ \nabla_{e f g h}{R_{a b c d}} -> A010_{a b c d e f g h} $)
    substitute (obj,$ \nabla_{e f g}{R_{a b c d}}   -> A009_{a b c d e f g}  $)
    substitute (obj,$ \nabla_{e f}{R_{a b c d}}     -> A008_{a b c d e f}   $)
    substitute (obj,$ \nabla_{e}{R_{a b c d}}       -> A007_{a b c d e}    $)
    substitute (obj,$ R_{a b c d}         -> A006_{a b c d}    $)
    sort_product (obj)
    rename_dummies (obj)
    substitute (obj,$ A001^{a}        -> A^{a}          $)
    substitute (obj,$ A002^{a}        -> x^{a}          $)
    substitute (obj,$ A003^{a}        -> Dx^{a}         $)
    substitute (obj,$ A004_{a b}      -> g_{a b}        $)
```

```

substitute (obj,$ A005^{a b}          -> g^{a b}          $)
substitute (obj,$ A006_{a b c d}      -> R_{a b c d}      $)
substitute (obj,$ A007_{a b c d e}    -> \nabla_{e}\{R_{a b c d}\}  $)
substitute (obj,$ A008_{a b c d e f}   -> \nabla_{e f}\{R_{a b c d}\} $)
substitute (obj,$ A009_{a b c d e f g} -> \nabla_{e f g}\{R_{a b c d}\} $)
substitute (obj,$ A010_{a b c d e f g h} -> \nabla_{e f g h}\{R_{a b c d}\} $)

return obj

def truncate (obj,n):
    ans = Ex(0)

    for i in range (0,n+1):
        foo := @(obj).
        bah = Ex("num = " + str(i))
        keep_weight (foo, bah)
        ans = ans + foo

    return ans

# expansions wrt the curvature

defgab := g_{a b} -> g0_{a b} + g2_{a b} + g3_{a b} + g4_{a b} + g5_{a b}.
defy   := y^{a}   -> y0^{a} + y2^{a} + y3^{a} + y4^{a} + y5^{a}.

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

substitute (lsq,defgab)
substitute (lsq,defy)
distribute (lsq)

def tidy (obj):
    foo := @(obj).
    sort_product      (foo)
    rename_dummies    (foo)
    canonicalise      (foo)
    return foo

```

```

lsq0 = tidy ( truncate (lsq,0) ) # cdb (lsq0.002,lsq0)
lsq2 = tidy ( truncate (lsq,2) ) # cdb (lsq2.002,lsq2)
lsq3 = tidy ( truncate (lsq,3) ) # cdb (lsq3.002,lsq3)
lsq4 = tidy ( truncate (lsq,4) ) # cdb (lsq4.002,lsq4)
lsq5 = tidy ( truncate (lsq,5) ) # cdb (lsq5.002,lsq5)

d20 := @(lsq2) - @(lsq0).      # cdb (d20.001,d20)  # check, should contain only O(2) terms
d32 := @(lsq3) - @(lsq2).      # cdb (d32.001,d32)  # check, should contain only O(3) terms
d43 := @(lsq4) - @(lsq3).      # cdb (d43.001,d43)  # check, should contain only O(4) terms
d54 := @(lsq5) - @(lsq4).      # cdb (d54.001,d54)  # check, should contain only O(5) terms

d5 := @(lsq5) - @(lsq).        # cdb (d5.001,d5)
d5 = tidy (d5)                 # cdb (d5.002,d5)  # all higher order terms, should see no O(5) terms

```


$$\text{lsq0.002} := g_{ab}^0 y^a y^b$$

$$\text{lsq2.002} := \overset{0}{g}_{ab} \overset{0}{y}^a \overset{0}{y}^b + 2 \overset{0}{g}_{ab} \overset{0}{y}^a \overset{2}{y}^b + \overset{2}{g}_{ab} \overset{0}{y}^a \overset{0}{y}^b$$

$$\text{lsq3.002} := g_{ab}^0 y^a y^b + 2g_{ab}^0 y^a y^b + g_{ab}^2 y^a y^b + 2g_{ab}^0 y^a y^b + g_{ab}^3 y^a y^b$$

$$\text{lsq4.002} := \overset{0}{g}_{ab}\overset{0}{y}^a\overset{0}{y}^b + 2\overset{0}{g}_{ab}\overset{0}{y}^a\overset{2}{y}^b + \overset{2}{g}_{ab}\overset{0}{y}^a\overset{0}{y}^b + 2\overset{0}{g}_{ab}\overset{0}{y}^a\overset{3}{y}^b + \overset{3}{g}_{ab}\overset{0}{y}^a\overset{0}{y}^b + 2\overset{0}{g}_{ab}\overset{0}{y}^a\overset{4}{y}^b + \overset{0}{g}_{ab}\overset{2}{y}^a\overset{2}{y}^b + 2\overset{2}{g}_{ab}\overset{0}{y}^a\overset{2}{y}^b + \overset{4}{g}_{ab}\overset{0}{y}^a\overset{0}{y}^b$$

$$\begin{aligned} \text{lsq5.002} := & g_{ab}^0 y^a y^b + 2g_{ab}^0 y^a y^2 b + g_{ab}^2 y^a y^b + 2g_{ab}^0 y^a y^3 b + g_{ab}^3 y^a y^b + 2g_{ab}^0 y^a y^4 b + g_{ab}^0 y^2 a y^2 b \\ & + 2g_{ab}^2 y^a y^2 b + g_{ab}^4 y^a y^b + 2g_{ab}^0 y^a y^5 b + 2g_{ab}^0 y^2 a y^3 b + 2g_{ab}^2 y^a y^3 b + 2g_{ab}^3 y^a y^2 b + g_{ab}^5 y^a y^b \end{aligned}$$

$$\text{d20.001} := 2g_{ab}^0 y^a y^b + g_{ab}^2 y^a y^b$$

$$\text{d32.001} := 2g_{ab}^0 y^a y^b + g_{ab}^3 y^a y^b$$

$$\text{d43.001} := 2g_{ab}^0 y^a y^b + g_{ab}^0 y^a y^b + 2g_{ab}^2 y^a y^b + g_{ab}^4 y^a y^b$$

$$\text{d54.001} := 2g_{ab}^0 y^a y^b + 2g_{ab}^0 y^a y^b + 2g_{ab}^2 y^a y^b + 2g_{ab}^3 y^a y^b + g_{ab}^5 y^a y^b$$

$$\begin{aligned} \text{d5.002} := & -2g_{ab}^0y^2a^4b - 2g_{ab}^0y^2a^5b - g_{ab}^0y^3a^3b - 2g_{ab}^0y^3a^4b - 2g_{ab}^0y^3a^5b - g_{ab}^0y^4a^4b - 2g_{ab}^0y^4a^5b - g_{ab}^0y^5a^5b - 2g_{ab}^2y^0a^4b - 2g_{ab}^2y^0a^5b - g_{ab}^2y^2a^2b - 2g_{ab}^2y^2a^3b - 2g_{ab}^2y^2a^4b \\ & - 2g_{ab}^2y^2a^5b - g_{ab}^2y^3a^3b - 2g_{ab}^2y^3a^4b - 2g_{ab}^2y^3a^5b - g_{ab}^2y^4a^4b - 2g_{ab}^2y^4a^5b - g_{ab}^2y^5a^5b - 2g_{ab}^3y^0a^3b - 2g_{ab}^3y^0a^4b - 2g_{ab}^3y^0a^5b - g_{ab}^3y^2a^2b - 2g_{ab}^3y^2a^3b \\ & - 2g_{ab}^3y^2a^4b - 2g_{ab}^3y^2a^5b - g_{ab}^3y^3a^3b - 2g_{ab}^3y^3a^4b - 2g_{ab}^3y^3a^5b - g_{ab}^3y^4a^4b - 2g_{ab}^3y^4a^5b - g_{ab}^3y^5a^5b - 2g_{ab}^4y^0a^2b - 2g_{ab}^4y^0a^3b - 2g_{ab}^4y^0a^4b - 2g_{ab}^4y^0a^5b \\ & - g_{ab}^4y^2a^2b - 2g_{ab}^4y^2a^3b - 2g_{ab}^4y^2a^4b - 2g_{ab}^4y^2a^5b - g_{ab}^4y^3a^3b - 2g_{ab}^4y^3a^4b - 2g_{ab}^4y^3a^5b - g_{ab}^4y^4a^4b - 2g_{ab}^4y^4a^5b - g_{ab}^4y^5a^5b - 2g_{ab}^5y^0a^2b - 2g_{ab}^5y^0a^3b \\ & - 2g_{ab}^5y^0a^4b - 2g_{ab}^5y^0a^5b - g_{ab}^5y^2a^2b - 2g_{ab}^5y^2a^3b - 2g_{ab}^5y^2a^4b - 2g_{ab}^5y^2a^5b - g_{ab}^5y^3a^3b - 2g_{ab}^5y^3a^4b - 2g_{ab}^5y^3a^5b - g_{ab}^5y^4a^4b - 2g_{ab}^5y^4a^5b - g_{ab}^5y^5a^5b \end{aligned}$$

Stage 2: Substution of y^a and g_{ab}

```
import cdblib

g0ab = cdblib.get('g_ab_0', 'metric.json')
g2ab = cdblib.get('g_ab_2', 'metric.json')
g3ab = cdblib.get('g_ab_3', 'metric.json')
g4ab = cdblib.get('g_ab_4', 'metric.json')
g5ab = cdblib.get('g_ab_5', 'metric.json')

defg0ab := g0_{a b} -> @(g0ab).
defg2ab := g2_{a b} -> @(g2ab).
defg3ab := g3_{a b} -> @(g3ab).
defg4ab := g4_{a b} -> @(g4ab).
defg5ab := g5_{a b} -> @(g5ab).

y0a = cdblib.get('y50', 'geodesic-bvp.json')
y2a = cdblib.get('y52', 'geodesic-bvp.json')
y3a = cdblib.get('y53', 'geodesic-bvp.json')
y4a = cdblib.get('y54', 'geodesic-bvp.json')
y5a = cdblib.get('y55', 'geodesic-bvp.json')

defy0a := y0^{a} -> @(y0a).
defy2a := y2^{a} -> @(y2a).
defy3a := y3^{a} -> @(y3a).
defy4a := y4^{a} -> @(y4a).
defy5a := y5^{a} -> @(y5a).

def substitute_gab_ya (obj):

    foo := @(obj).

    substitute (foo, defg0ab)
    substitute (foo, defg2ab)
    substitute (foo, defg3ab)
    substitute (foo, defg4ab)
    substitute (foo, defg5ab)
```

```

substitute (foo,defy0a)
substitute (foo,defy2a)
substitute (foo,defy3a)
substitute (foo,defy4a)
substitute (foo,defy5a)

distribute      (foo)
sort_product    (foo)
rename_dummies  (foo)
canonicalise    (foo)

substitute      (foo,$g_{a b} g^{c b} -> \delta^{c}_{a}$)
eliminate_kronecker (foo)
foo = product_sort (foo)
rename_dummies   (foo)
canonicalise     (foo)

return foo

def get_Rterm (obj,n):

# I would like to assign different weights to \nabla_{a}, \nabla_{a b}, \nabla_{a b c} etc. but no matter
# what I do it appears that Cadabra assigns the same weight to all of these regardless of the number of subscripts.
# It seems that the weight is assigned to the symbol \nabla alone. So I'm forced to use the following substitution trick.

Q_{a b c d}::Weight(label=numR,value=2).
Q_{a b c d e}::Weight(label=numR,value=3).
Q_{a b c d e f}::Weight(label=numR,value=4).
Q_{a b c d e f g}::Weight(label=numR,value=5).

tmp := @(obj).

distribute (tmp)

substitute (tmp, $\nabla_{e f g}\{R_{a b c d}\} -> Q_{a b c d e f g}$)
substitute (tmp, $\nabla_{e f}\{R_{a b c d}\} -> Q_{a b c d e f}$)
substitute (tmp, $\nabla_{e}\{R_{a b c d}\} -> Q_{a b c d e}$)

```

```

substitute (tmp, $R_{a b c d} -> Q_{a b c d}$)

foo := @(tmp).
bah = Ex("numR = " + str(n))
keep_weight (foo, bah)

substitute (foo, $Q_{a b c d e f g} -> \nabla_{e f g}\{R_{a b c d}\}$)
substitute (foo, $Q_{a b c d e f} -> \nabla_{e f}\{R_{a b c d}\}$)
substitute (foo, $Q_{a b c d e} -> \nabla_e\{R_{a b c d}\}$)
substitute (foo, $Q_{a b c d} -> R_{a b c d}$)

return foo

lsq2 = substitute_gab_ya (lsq2) # cdb (lsq2.101,lsq2)
lsq3 = substitute_gab_ya (lsq3) # cdb (lsq3.101,lsq3)
lsq4 = substitute_gab_ya (lsq4) # cdb (lsq4.101,lsq4)
lsq5 = substitute_gab_ya (lsq5) # cdb (lsq5.101,lsq5)

lsq50 = get_Rterm (lsq5,0)
lsq52 = get_Rterm (lsq5,2)
lsq53 = get_Rterm (lsq5,3)
lsq54 = get_Rterm (lsq5,4)
lsq55 = get_Rterm (lsq5,5)

cdblib.create ('geodesic-lsq.json')

cdblib.put ('lsq2',lsq2,'geodesic-lsq.json')
cdblib.put ('lsq3',lsq3,'geodesic-lsq.json')
cdblib.put ('lsq4',lsq4,'geodesic-lsq.json')
cdblib.put ('lsq5',lsq5,'geodesic-lsq.json')

cdblib.put ('lsq50',lsq50,'geodesic-lsq.json')
cdblib.put ('lsq52',lsq52,'geodesic-lsq.json')
cdblib.put ('lsq53',lsq53,'geodesic-lsq.json')
cdblib.put ('lsq54',lsq54,'geodesic-lsq.json')
cdblib.put ('lsq55',lsq55,'geodesic-lsq.json')

```

$$\text{lsq2.101} := Dx^a Dx^b g_{ab} - \frac{1}{3} x^a x^b Dx^c Dx^d R_{acbd}$$

$$\text{lsq3.101} := Dx^a Dx^b g_{ab} - \frac{1}{3} x^a x^b Dx^c Dx^d R_{acbd} - \frac{1}{12} x^a x^b Dx^c Dx^d Dx^e \nabla_c R_{adbe} - \frac{1}{6} x^a x^b x^c Dx^d Dx^e \nabla_a R_{bdce}$$

$$\begin{aligned} \text{lsq4.101} := & Dx^a Dx^b g_{ab} - \frac{1}{3} x^a x^b Dx^c Dx^d R_{acbd} - \frac{1}{12} x^a x^b Dx^c Dx^d Dx^e \nabla_c R_{adbe} - \frac{1}{6} x^a x^b x^c Dx^d Dx^e \nabla_a R_{bdce} - \frac{1}{45} x^a x^b Dx^c Dx^d Dx^e Dx^f g^{gh} R_{acd g} R_{bef h} \\ & + \frac{2}{45} x^a x^b x^c Dx^d Dx^e Dx^f g^{gh} R_{adbg} R_{cef h} - \frac{1}{40} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{da} R_{becf} - \frac{1}{40} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{ad} R_{becf} \\ & - \frac{1}{60} x^a x^b Dx^c Dx^d Dx^e Dx^f \nabla_{cd} R_{aebf} + \frac{2}{45} x^a x^b x^c Dx^d Dx^e Dx^f g^{gh} R_{aebg} R_{cf dh} - \frac{1}{20} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{ab} R_{cedf} \end{aligned}$$

$$\begin{aligned} \text{lsq5.101} := & Dx^a Dx^b g_{ab} - \frac{1}{3} x^a x^b Dx^c Dx^d R_{acbd} - \frac{1}{12} x^a x^b Dx^c Dx^d Dx^e \nabla_c R_{adbe} - \frac{1}{6} x^a x^b x^c Dx^d Dx^e \nabla_a R_{bdce} \\ & - \frac{1}{45} x^a x^b Dx^c Dx^d Dx^e Dx^f g^{gh} R_{acd g} R_{bef h} + \frac{2}{45} x^a x^b x^c Dx^d Dx^e Dx^f g^{gh} R_{adbg} R_{cef h} \\ & - \frac{1}{40} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{da} R_{becf} - \frac{1}{40} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{ad} R_{becf} - \frac{1}{60} x^a x^b Dx^c Dx^d Dx^e Dx^f \nabla_{cd} R_{aebf} \\ & + \frac{2}{45} x^a x^b x^c Dx^d Dx^e Dx^f g^{gh} R_{aebg} R_{cf dh} - \frac{1}{20} x^a x^b x^c Dx^d Dx^e Dx^f \nabla_{ab} R_{cedf} - \frac{1}{270} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{adeh} \nabla_f R_{bgci} \\ & - \frac{1}{45} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{adeh} \nabla_b R_{cf gi} + \frac{1}{108} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{adeh} \nabla_i R_{bf cg} \\ & - \frac{1}{60} x^a x^b Dx^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{acd h} \nabla_e R_{bf gi} + \frac{1}{45} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{aef h} \nabla_b R_{cg di} \\ & + \frac{1}{45} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{aebh} \nabla_f R_{cg di} + \frac{1}{45} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{aebh} \nabla_c R_{df gi} - \frac{1}{180} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{eab} R_{cf dg} \\ & - \frac{1}{180} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{aeb} R_{cf dg} - \frac{1}{180} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{abe} R_{cf dg} + \frac{2}{135} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{adbh} \nabla_e R_{cf gi} \\ & - \frac{1}{270} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{dea} R_{bf cg} - \frac{1}{270} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{dae} R_{bf cg} - \frac{1}{270} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{ade} R_{bf cg} \\ & - \frac{1}{360} x^a x^b Dx^c Dx^d Dx^e Dx^f Dx^g \nabla_{cde} R_{af bg} + \frac{2}{45} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g g^{hi} R_{af bh} \nabla_c R_{dgei} - \frac{1}{90} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g \nabla_{abc} R_{df eg} \end{aligned}$$

Stage 3: Reformatting

```
def reformat (obj,scale):
    foo = Ex(str(scale))
    bah := @(foo) @(obj).
    distribute (bah)
    bah = product_sort (bah)
    rename_dummies (bah)
    canonicalise (bah)
    substitute (bah,$Dx^{b}->zzz^{b}$)
    factor_out (bah,$x^{a?},zzz^{b?}$)
    substitute (bah,$zzz^{b}->Dx^{b}$)
    ans := @(bah) / @(foo).
    return ans

def rescale (obj,scale):
    foo = Ex(str(scale))
    bah := @(foo) @(obj).
    distribute (bah)
    substitute (bah,$Dx^{b}->zzz^{b}$)
    factor_out (bah,$x^{a?},zzz^{b?}$)
    substitute (bah,$zzz^{b}->Dx^{b}$)
    return bah

Rterm0 := @(lsq50).
Rterm2 := @(lsq52).
Rterm3 := @(lsq53).
Rterm4 := @(lsq54).
Rterm5 := @(lsq55).

Rterm0 = reformat (Rterm0, 1)      # cdb(Rterm0.301,Rterm0) # LCB: returns Dx before g, not what I want
Rterm2 = reformat (Rterm2, 3)      # cdb(Rterm2.301,Rterm2)
Rterm3 = reformat (Rterm3, 12)     # cdb(Rterm3.301,Rterm3)
Rterm4 = reformat (Rterm4, 360)    # cdb(Rterm4.301,Rterm4)
Rterm5 = reformat (Rterm5,1080)    # cdb(Rterm5.301,Rterm5)

Rterm0 := g_{a b} Dx^{a} Dx^{b}.  # LCB: fixes the order of terms, g before Dx,
```

```

lsq3 := @(Rterm0) + @(Rterm2). # cdb (lsq4.301,lsq3)
lsq4 := @(Rterm0) + @(Rterm2) + @(Rterm3). # cdb (lsq4.301,lsq4)
lsq5 := @(Rterm0) + @(Rterm2) + @(Rterm3) + @(Rterm4). # cdb (lsq5.301,lsq5)
lsq6 := @(Rterm0) + @(Rterm2) + @(Rterm3) + @(Rterm4) + @(Rterm5). # cdb (lsq5.301,lsq6)

lsq := @(lsq6). # cdb (lsq.301,lsq)

scaled0 = rescale (Rterm0, 1) # cdb (scaled0.301,scaled0) # LCB: returns Dx before g, not what I want
scaled2 = rescale (Rterm2, 3) # cdb (scaled2.301,scaled2)
scaled3 = rescale (Rterm3, 12) # cdb (scaled3.301,scaled3)
scaled4 = rescale (Rterm4, 360) # cdb (scaled4.301,scaled4)
scaled5 = rescale (Rterm5, 1080) # cdb (scaled5.301,scaled5)

scaled0 := g_{a b} Dx^{a} Dx^{b}. # cdb (scaled0.301,scaled0) # LCB: fixes the order of terms, g before Dx, good

```

Geodesic arc-length

$$\begin{aligned}
(\Delta s)^2 = & g_{ab} Dx^a Dx^b - \frac{1}{3} x^a x^b Dx^c Dx^d R_{acbd} - \frac{1}{12} x^a x^b Dx^c Dx^d Dx^e \nabla_c R_{adbe} - \frac{1}{6} x^a x^b x^c Dx^d Dx^e \nabla_a R_{bdce} \\
& + \frac{1}{360} x^a x^b Dx^c Dx^d Dx^e Dx^f (-8g^{gh} R_{acd g} R_{b e f h} - 6\nabla_{cd} R_{a e b f}) + \frac{1}{360} x^a x^b x^c Dx^d Dx^e Dx^f (16g^{gh} R_{adb g} R_{c e f h} - 9\nabla_{da} R_{b e c f} - 9\nabla_{ad} R_{b e c f}) \\
& + \frac{1}{360} x^a x^b x^c x^d Dx^e Dx^f (16g^{gh} R_{a e b g} R_{c f d h} - 18\nabla_{ab} R_{c e d f}) + \frac{1}{1080} x^a x^b x^c Dx^d Dx^e Dx^f Dx^g (-4g^{hi} R_{ade h} \nabla_f R_{b g c i} - 24g^{hi} R_{ade h} \nabla_b R_{c f g i} \\
& \quad + 10g^{hi} R_{ade h} \nabla_i R_{b f c g} + 16g^{hi} R_{adb h} \nabla_e R_{c f g i} - 4\nabla_{dea} R_{b f c g} - 4\nabla_{dae} R_{b f c g} - 4\nabla_{ade} R_{b f c g}) \\
& + \frac{1}{1080} x^a x^b Dx^c Dx^d Dx^e Dx^f Dx^g (-18g^{hi} R_{acd h} \nabla_e R_{b f g i} - 3\nabla_{cde} R_{a f b g}) \\
& + \frac{1}{1080} x^a x^b x^c x^d Dx^e Dx^f Dx^g (24g^{hi} R_{a e f h} \nabla_b R_{c g d i} + 24g^{hi} R_{a e b h} \nabla_f R_{c g d i} + 24g^{hi} R_{a e b h} \nabla_c R_{d f g i} - 6\nabla_{eab} R_{c f d g} - 6\nabla_{aeb} R_{c f d g} - 6\nabla_{abe} R_{c f d g}) \\
& + \frac{1}{1080} x^a x^b x^c x^d x^e Dx^f Dx^g (48g^{hi} R_{a f b h} \nabla_c R_{d g e i} - 12\nabla_{abc} R_{d f e g}) + \mathcal{O}(\epsilon^6)
\end{aligned}$$

Geodesic arc-length curvature expansion

$$(\Delta s)^2 = \overset{0}{\Delta} + \overset{2}{\Delta} + \overset{3}{\Delta} + \overset{4}{\Delta} + \overset{5}{\Delta} + \mathcal{O}(\epsilon^6)$$

$$\overset{0}{\Delta} = g_{ab} D x^a D x^b$$

$$3\overset{2}{\Delta} = -x^a x^b D x^c D x^d R_{acbd}$$

$$12\overset{3}{\Delta} = -x^a x^b D x^c D x^d D x^e \nabla_c R_{adbe} - 2x^a x^b x^c D x^d D x^e \nabla_a R_{bdce}$$

$$360\overset{4}{\Delta} = x^a x^b D x^c D x^d D x^e D x^f (-8g^{gh} R_{acd g} R_{b e f h} - 6\nabla_{cd} R_{a e b f}) + x^a x^b x^c D x^d D x^e D x^f (16g^{gh} R_{ad b g} R_{c e f h} - 9\nabla_{da} R_{b e c f} - 9\nabla_{ad} R_{b e c f}) \\ + x^a x^b x^c x^d D x^e D x^f (16g^{gh} R_{a e b g} R_{c f d h} - 18\nabla_{ab} R_{c e d f})$$

$$1080\overset{5}{\Delta} = x^a x^b x^c D x^d D x^e D x^f D x^g (-4g^{hi} R_{a d e h} \nabla_f R_{b g c i} - 24g^{hi} R_{a d e h} \nabla_b R_{c f g i} + 10g^{hi} R_{a d e h} \nabla_i R_{b f c g} + 16g^{hi} R_{a d b h} \nabla_e R_{c f g i} - 4\nabla_{d e a} R_{b f c g} - 4\nabla_{d a e} R_{b f c g} \\ - 4\nabla_{a d e} R_{b f c g}) + x^a x^b D x^c D x^d D x^e D x^f D x^g (-18g^{hi} R_{a c d h} \nabla_e R_{b f g i} - 3\nabla_{c d e} R_{a f b g}) \\ + x^a x^b x^c x^d D x^e D x^f D x^g (24g^{hi} R_{a e f h} \nabla_b R_{c g d i} + 24g^{hi} R_{a e b h} \nabla_f R_{c g d i} + 24g^{hi} R_{a e b h} \nabla_c R_{d f g i} - 6\nabla_{e a b} R_{c f d g} - 6\nabla_{a e b} R_{c f d g} - 6\nabla_{a b e} R_{c f d g}) \\ + x^a x^b x^c x^d x^e D x^f D x^g (48g^{hi} R_{a f b h} \nabla_c R_{d g e i} - 12\nabla_{a b c} R_{d f e g})$$

```
cdblib.create ('geodesic-lsq.export')

# 3rd to 6th order lsq
cdblib.put ('lsq3',lsq3,'geodesic-lsq.export')
cdblib.put ('lsq4',lsq4,'geodesic-lsq.export')
cdblib.put ('lsq5',lsq5,'geodesic-lsq.export')
cdblib.put ('lsq6',lsq6,'geodesic-lsq.export')

# 6th order lsq terms, scaled
cdblib.put ('lsq60',scaled0,'geodesic-lsq.export')
cdblib.put ('lsq62',scaled2,'geodesic-lsq.export')
cdblib.put ('lsq63',scaled3,'geodesic-lsq.export')
cdblib.put ('lsq64',scaled4,'geodesic-lsq.export')
cdblib.put ('lsq65',scaled5,'geodesic-lsq.export')

checkpoint.append (lsq4)

checkpoint.append (scaled0)
checkpoint.append (scaled2)
checkpoint.append (scaled3)
checkpoint.append (scaled4)
checkpoint.append (scaled5)
```