Example 10 The determinant of the metric

Our game here is to compute (the leading terms) in $\det g$ of the metric in RNC form

$$g_{ab}(x) = g_{ab} - \frac{1}{3}x^c x^d R_{acbd} - \frac{1}{6}x^c x^d x^e \nabla_c R_{adbe} + \frac{1}{180}x^c x^d x^e x^f \left(8g^{gh}R_{acdg}R_{befh} - 9\nabla_{cd}R_{aebf}\right) + \cdots$$

For the sake of simplicity let's assume that we are working in 3-dimensions. The following analysis is easily generalsied to other dimensions (and the final answers for $\det g$ and friends are unchanged).

Define ϵ^{abc}_{ijk} by

$$\epsilon^{abc}_{ijk} = \delta^a_i\delta^b_j\delta^c_k - \delta^b_i\delta^a_j\delta^c_k + \delta^c_i\delta^a_j\delta^b_k - \delta^c_i\delta^b_j\delta^a_k + \delta^b_i\delta^c_j\delta^a_k - \delta^a_i\delta^c_j\delta^b_k \tag{1}$$

It is easy to see that ϵ^{abc}_{ijk} is anti-symmetric in both its upper and lower indices. A trivial computation shows that for any 3×3 square matrix M_{ab},

$$\epsilon^{abc}_{123}M_{1a}M_{2b}M_{3c} = \left(\delta^a_1\delta^b_2\delta^c_3 - \delta^b_1\delta^a_2\delta^c_3 + \delta^c_1\delta^a_2\delta^b_3 - \delta^c_1\delta^b_2\delta^a_3 + \delta^b_1\delta^c_2\delta^a_3 - \delta^a_1\delta^c_2\delta^b_3\right) M_{1a}M_{2b}M_{3c} = \det M \tag{2}$$

This can be easily generalised to

$$\epsilon^{abc}_{ijk}M_{pa}M_{qb}M_{rc} = \begin{cases} \pm\det M & \text{when } (ijk) \text{ and } (pqr) \text{ are permutations of } (123) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

The $\pm$ sign in the above depends on the particular permutations of (ijk) and (pqr). If both permutations are even or both odd then the sign is $+1$ otherwise the sign is -1. The same arguments can also be applied to a matrix inverse N^{-1} leading to

$$\epsilon^{ijk}_{uvw}N^{pu}N^{qv}M^{rw} = \begin{cases} \pm\det N^{-1} & \text{when } (ijk) \text{ and } (pqr) \text{ are permutations of } (123) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Note that the $\pm$ in this case will match exactly that for the case of $\det M$. Thus, multiplying both expressions and summing over all choices for (ijk) and (pqr) leads to

$$\sum_{\substack{(ijk)\\(pqr)}} \left(\det N^{-1}\right)\det M = \epsilon^{ijk}_{uvw}N^{pu}N^{qv}M^{rw}\epsilon^{abc}_{ijk}M_{pa}M_{qb}M_{rc} \tag{5}$$

where the sum on the left hand side includes just those (ijk) and (prq) that are permutations of (123). There are 3! choices for (ijk) and 3! choices for (pqr) and thus the left hand side is easily reduced to $(3!\,)^2 \det M/\det N$ where $\det N = 1/\det(N^{-1})$. For the right hand side notice that

$$\epsilon^{ijk}_{uvw}\epsilon^{abc}_{ijk} = 3!\,\epsilon^{abc}_{uvw} \tag{6}$$

which leads to

$$\det M = \frac{1}{3!}\det N \epsilon^{abc}_{uvw} M_{pa} M_{qb} M_{rc} N^{pu} N^{qv} N^{rw} \tag{7}$$

For our RNC metric we will set $N^{ab} = g^{ab}$ and $M_{ij} = g_{ij}(x)$. Since g^{ab} is of the form $\mathrm{diag}(-1, 1, 1, 1)$ we have $\det g = -1$ and thus

$$\det g(x) = -\frac{1}{3!}\epsilon^{abc}_{ijk}\, g_{pa}(x)\, g_{qb}(x)\, g_{rc}(x)\, g^{ip} g^{jq} g^{kr} \tag{8}$$

The ϵ^{abc}_{ijk} can be constructed in Cadabra by applying the `asym` algorithm to the upper indices of $\delta^a_i\delta^b_j\delta^c_k$. Note that `asym` will include the 1/3! coeffcient as part of its output.

The following code computes $-\det g$ rather than $\det g$.

Note that Calzetta etal. use an opposite sign for R_{abcd} so when comparing the following results against Calzetta do take note of this flipped sign in R_{abcd}.

The determinant of the metric

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

{a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Integer(1..3).

\nabla{#}::Derivative.

d{#}::KroneckerDelta.

g^{a b}::Symmetric.
g_{a b}::Symmetric.

R_{a b c d}::RiemannTensor.

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

def truncate (obj,n):

    ans = Ex("0")  # create a Cadabra object with value zero

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

    return ans

gab := g_{a b}
       - (1/3)   x^{c} x^{d} R_{a c b d}
       - (1/6)   x^{c} x^{d} x^{e} \nabla_{c}{R_{a d b e}}
       + (1/180) x^{c} x^{d} x^{e} x^{f} ( 8 g^{g h} R_{a c d g} R_{b e f h}
                                          -9 \nabla_{c d}{R_{a e b f}} ).          # cdb (ex-10.gab.000,gab)

iab := g^{a b}
       + (1/3)  x^{c} x^{d} g^{a e} g^{b f} R_{c e d f}
```

```
       + (1/6)  x^{c} x^{d} x^{e} g^{a f} g^{b g} \nabla_{c}{R_{d f e g}}
       + (1/60) x^{c} x^{d} x^{e} x^{f} g^{a g} g^{b h}
                              ( 4 g^{i j} R_{c g d i} R_{e h f j}
                               +3 \nabla_{c d}{R_{e g f h}} ).                         # cdb(ex-10.iab.000,iab)

distribute (gab)
distribute (iab)

gxab := gx_{a b} -> @(gab).

eps := d^{a}_{i} d^{b}_{j} d^{c}_{k}.   # cdb (ex-10.eps.001,eps) # includes a factor of 1/3!
asym (eps,$^{a},^{b},^{c}$)             # cdb (ex-10.eps.002,eps)

# compute negative detg rather than det g, note 1/3! included in eps
Ndetg := @(eps) gx_{p a} gx_{q b} gx_{r c} g^{i p} g^{j q} g^{k r}.   # cdb (ex-10.Ndetg.001,Ndetg)

substitute       (Ndetg,gxab)                                         # cdb (ex-10.Ndetg.002,Ndetg)
distribute       (Ndetg)                                              # cdb (ex-10.Ndetg.003,Ndetg)
Ndetg = truncate (Ndetg,4)                                            # cdb (ex-10.Ndetg.004,Ndetg)
substitute       (Ndetg,$g^{a b} g_{b c} -> d^{a}_{c}$,repeat=True)   # cdb (ex-10.Ndetg.005,Ndetg)
eliminate_kronecker (Ndetg)                                           # cdb (ex-10.Ndetg.006,Ndetg)
sort_product     (Ndetg)                                              # cdb (ex-10.Ndetg.007,Ndetg)
rename_dummies   (Ndetg)                                              # cdb (ex-10.Ndetg.008,Ndetg)
canonicalise     (Ndetg)                                              # cdb (ex-10.Ndetg.009,Ndetg)

# introduce the Ricci tensor

substitute (Ndetg,$R_{a b c d} g^{a c}               -> R_{b d}$,repeat=True)               # cdb (ex-10.Ndetg.101,Ndetg)
substitute (Ndetg,$\nabla_{a}{R_{b c d e}} g^{b d}   -> \nabla_{a}{R_{c e}}$,repeat=True)   # cdb (ex-10.Ndetg.102,Ndetg)
substitute (Ndetg,$\nabla_{a b}{R_{c d e f}} g^{c e} -> \nabla_{a b}{R_{d f}}$,repeat=True) # cdb (ex-10.Ndetg.103,Ndetg)

# the following was based on sqrt-Ndetg.tex

sqrtNdetg := 1/2 + (1/2) @(Ndetg)
           - (1/8) (1/9) R_{a b} R_{c d} x^{a} x^{b} x^{c} x^{d}
           - (1/4) (1/18) R_{a b} \nabla_{c}{R_{d e}} x^{a} x^{b} x^{c} x^{d} x^{e}.
                                                                      # cdb (ex-10.sqrtNdetg.001,sqrtNdetg)
```

```
sort_product   (sqrtNdetg)                                          # cdb (ex-10.sqrtNdetg.002,sqrtNdetg)
rename_dummies (sqrtNdetg)                                          # cdb (ex-10.sqrtNdetg.003,sqrtNdetg)
canonicalise   (sqrtNdetg)                                          # cdb (ex-10.sqrtNdetg.004,sqrtNdetg)

logNdetg := -1 + @(Ndetg)
           - (1/2) (1/9) R_{a b} R_{c d} x^{a} x^{b} x^{c} x^{d}
           - (1/18) R_{a b} \nabla_{c}{R_{d e}} x^{a} x^{b} x^{c} x^{d} x^{e}.
                                                                    # cdb (ex-10.logNdetg.001,logNdetg)

sort_product   (logNdetg)                                           # cdb (ex-10.logNdetg.002,logNdetg)
rename_dummies (logNdetg)                                           # cdb (ex-10.logNdetg.003,logNdetg)
canonicalise   (logNdetg)                                           # cdb (ex-10.logNdetg.004,logNdetg)

# =================================================================================
# the remaining code is just for pretty printing

def product_sort (obj):
    substitute (obj,$ x^{a}                          -> A000^{a}              $)
    substitute (obj,$ g^{a b}                        -> A001^{a b}            $)
    substitute (obj,$ \nabla_{c}{R_{a b}}            -> A004_{a b c}          $)
    substitute (obj,$ \nabla_{c d}{R_{a b}}          -> A005_{a b c d}        $)
    substitute (obj,$ \nabla_{c d e}{R_{a b}}        -> A006_{a b c d e}      $)
    substitute (obj,$ \nabla_{c d e f}{R_{a b}}      -> A007_{a b c d e f}    $)
    substitute (obj,$ \nabla_{e}{R_{a b c d}}        -> A008_{a b c d e}      $)
    substitute (obj,$ \nabla_{e f}{R_{a b c d}}      -> A009_{a b c d e f}    $)
    substitute (obj,$ \nabla_{e f g}{R_{a b c d}}    -> A010_{a b c d e f g}  $)
    substitute (obj,$ \nabla_{e f g h}{R_{a b c d}}  -> A011_{a b c d e f g h} $)
    substitute (obj,$ R_{a b}                        -> A002_{a b}            $)
    substitute (obj,$ R_{a b c d}                    -> A003_{a b c d}        $)
    sort_product   (obj)
    rename_dummies (obj)
    substitute (obj,$ A000^{a}               -> x^{a}                         $)
    substitute (obj,$ A001^{a b}             -> g^{a b}                       $)
    substitute (obj,$ A002_{a b}             -> R_{a b}                       $)
    substitute (obj,$ A003_{a b c d}         -> R_{a b c d}                   $)
    substitute (obj,$ A004_{a b c}           -> \nabla_{c}{R_{a b}}           $)
    substitute (obj,$ A005_{a b c d}         -> \nabla_{c d}{R_{a b}}         $)
    substitute (obj,$ A006_{a b c d e}       -> \nabla_{c d e}{R_{a b}}       $)
```

```
    substitute (obj,$ A007_{a b c d e f}       -> \nabla_{c d e f}{R_{a b}}      $)
    substitute (obj,$ A008_{a b c d e}         -> \nabla_{e}{R_{a b c d}}        $)
    substitute (obj,$ A009_{a b c d e f}       -> \nabla_{e f}{R_{a b c d}}      $)
    substitute (obj,$ A010_{a b c d e f g}     -> \nabla_{e f g}{R_{a b c d}}    $)
    substitute (obj,$ A011_{a b c d e f g h}   -> \nabla_{e f g h}{R_{a b c d}}  $)

def get_term (obj,n):

    x^{a}::Weight(label=xnum).

    foo := @(obj).
    bah  = Ex("xnum = " + str(n))
    keep_weight (foo,bah)

    return foo

def reformat (obj,scale):
    foo  = Ex(str(scale))
    bah := @(foo) @(obj).
    distribute     (bah)
    product_sort   (bah)
    rename_dummies (bah)
    canonicalise   (bah)
    sort_sum       (bah)
    factor_out     (bah,$x^{a?}$)
    ans := @(bah) / @(foo).
    return ans

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

# ---------------------------------------------------------------
# reformat Ndetg

```

```
Rterm0 = get_term (Ndetg,0)       # cdb (ex-10.Rterm0.701,Rterm0)
Rterm1 = get_term (Ndetg,1)       # cdb (ex-10.Rterm1.701,Rterm1)
Rterm2 = get_term (Ndetg,2)       # cdb (ex-10.Rterm2.701,Rterm2)
Rterm3 = get_term (Ndetg,3)       # cdb (ex-10.Rterm3.701,Rterm3)
Rterm4 = get_term (Ndetg,4)       # cdb (ex-10.Rterm4.701,Rterm4)

Rterm0 = reformat (Rterm0,  1)    # cdb (ex-10.Rterm0.702,Rterm0)
Rterm1 = reformat (Rterm1,  1)    # cdb (ex-10.Rterm1.702,Rterm1)
Rterm2 = reformat (Rterm2,  3)    # cdb (ex-10.Rterm2.702,Rterm2)
Rterm3 = reformat (Rterm3,  6)    # cdb (ex-10.Rterm3.702,Rterm3)
Rterm4 = reformat (Rterm4,180)    # cdb (ex-10.Rterm4.702,Rterm4)

Ndetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.Ndetg.701,Ndetg)

# -----------------------------------------------------------------
# reformat sqrtNdetg

Rterm0 = get_term (sqrtNdetg,0)   # cdb (ex-10.Rterm0.801,Rterm0)
Rterm1 = get_term (sqrtNdetg,1)   # cdb (ex-10.Rterm1.801,Rterm1)
Rterm2 = get_term (sqrtNdetg,2)   # cdb (ex-10.Rterm2.801,Rterm2)
Rterm3 = get_term (sqrtNdetg,3)   # cdb (ex-10.Rterm3.801,Rterm3)
Rterm4 = get_term (sqrtNdetg,4)   # cdb (ex-10.Rterm4.801,Rterm4)

Rterm0 = reformat (Rterm0,  1)    # cdb (ex-10.Rterm0.802,Rterm0)
Rterm1 = reformat (Rterm1,  1)    # cdb (ex-10.Rterm1.802,Rterm1)
Rterm2 = reformat (Rterm2,  6)    # cdb (ex-10.Rterm2.802,Rterm2)
Rterm3 = reformat (Rterm3, 12)    # cdb (ex-10.Rterm3.802,Rterm3)
Rterm4 = reformat (Rterm4,360)    # cdb (ex-10.Rterm4.802,Rterm4)

sqrtNdetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.sqrtNdetg.801,sqrtNdetg)

# -----------------------------------------------------------------
# reformat logNdetg

Rterm0 = get_term (logNdetg,0)    # cdb (ex-10.Rterm0.801,Rterm0)
Rterm1 = get_term (logNdetg,1)    # cdb (ex-10.Rterm1.801,Rterm1)
Rterm2 = get_term (logNdetg,2)    # cdb (ex-10.Rterm2.801,Rterm2)
Rterm3 = get_term (logNdetg,3)    # cdb (ex-10.Rterm3.801,Rterm3)
```

```
Rterm4 = get_term (logNdetg,4)     # cdb (ex-10.Rterm4.801,Rterm4)

Rterm0 = reformat (Rterm0,  1)     # cdb (ex-10.Rterm0.802,Rterm0)
Rterm1 = reformat (Rterm1,  1)     # cdb (ex-10.Rterm1.802,Rterm1)
Rterm2 = reformat (Rterm2,  3)     # cdb (ex-10.Rterm2.802,Rterm2)
Rterm3 = reformat (Rterm3,  6)     # cdb (ex-10.Rterm3.802,Rterm3)
Rterm4 = reformat (Rterm4,180)     # cdb (ex-10.Rterm4.802,Rterm4)

logNdetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.logNdetg.901,logNdetg)

checkpoint.append (Ndetg)
checkpoint.append (sqrtNdetg)
checkpoint.append (logNdetg)
```

The metric determinant in Riemann normal coordinates

$$-\det g(x) = 1 - \frac{1}{3}x^a x^b R_{ab} - \frac{1}{6}x^a x^b x^c \nabla_a R_{bc} + \frac{1}{180}x^a x^b x^c x^d \left(-9\nabla_{ab}R_{cd} + 10R_{ab}R_{cd} - 2g^{ef}g^{gh}R_{aebg}R_{cfdh}\right) + \cdots$$

The volume element in RNC

If $-\det g(x)$ is non-negative then we also have

$$\sqrt{-\det g(x)} = 1 - \frac{1}{6}x^a x^b R_{ab} - \frac{1}{12}x^a x^b x^c \nabla_a R_{bc} + \frac{1}{360}x^a x^b x^c x^d \left(-9\nabla_{ab}R_{cd} + 5R_{ab}R_{cd} - 2g^{ef}g^{gh}R_{aebg}R_{cfdh}\right) + \cdots$$

The log of -detg in RNC

$$\log\left(-\det g(x)\right) = -\frac{1}{3}x^a x^b R_{ab} - \frac{1}{6}x^a x^b x^c \nabla_a R_{bc} + \frac{1}{180}x^a x^b x^c x^d \left(-9\nabla_{ab}R_{cd} - 2g^{ef}g^{gh}R_{aebg}R_{cfdh}\right) + \cdots$$

Apart from the signs, this matches exactly the expression given by Calzetta etal. (eq. A14)