

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

$$g_{ab}(x) = g_{ab} - \frac{1}{3}x^c x^d R_{acbd} - \frac{1}{6}x^c x^d x^e \nabla_c R_{adbe} + \frac{1}{180}x^c x^d x^e x^f (8g^{gh} R_{acd g} R_{b e f h} - 9\nabla_{cd} R_{a e b f}) + \dots$$

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

Define ϵ_{ijk}^{abc} by

$$\epsilon_{ijk}^{abc} = \delta_i^a \delta_j^b \delta_k^c - \delta_i^b \delta_j^a \delta_k^c + \delta_i^c \delta_j^a \delta_k^b - \delta_i^c \delta_j^b \delta_k^a + \delta_i^b \delta_j^c \delta_k^a - \delta_i^a \delta_j^c \delta_k^b \quad (1)$$

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

$$\epsilon_{123}^{abc} M_{1a} M_{2b} M_{3c} = (\delta_1^a \delta_2^b \delta_3^c - \delta_1^b \delta_2^a \delta_3^c + \delta_1^c \delta_2^a \delta_3^b - \delta_1^c \delta_2^b \delta_3^a + \delta_1^b \delta_2^c \delta_3^a - \delta_1^a \delta_2^c \delta_3^b) M_{1a} M_{2b} M_{3c} = \det M \quad (2)$$

This can be easily generalised to

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

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

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

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

$$\sum_{\substack{(ijk) \\ (pqr)}} (\det N^{-1}) \det M = \epsilon_{uvw}^{ijk} N^{pu} N^{qv} M^{rw} \epsilon_{ijk}^{abc} M_{pa} M_{qb} M_{rc} \quad (5)$$

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 (prq) 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

$$\epsilon_{uvw}^{ijk} \epsilon_{ijk}^{abc} = 3! \epsilon_{uvw}^{abc} \quad (6)$$

which leads to

$$\det M = \frac{1}{3!} \det N \epsilon_{uvw}^{abc} M_{pa} M_{qb} M_{rc} N^{pu} N^{qv} N^{rw} \quad (7)$$

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

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

The ϵ_{ijk}^{abc} can be constructed in Cadabra by applying the `asym` algorithm to the upper indices of $\delta_i^a \delta_j^b \delta_k^c$. Note that `asym` will include the $1/3!$ coefficient as part of its output.

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

Note that Calzetta et al. 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

```
1 {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Indices(position=independent).
2
3 {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Integer(1..3).
4
5 \nabla{#}::Derivative.
6
7 d{#}::KroneckerDelta.
8
9 g^{a b}::Symmetric.
10 g_{a b}::Symmetric.
11
12 R_{a b c d}::RiemannTensor.
13
14 x^{a}::Weight(label=num,value=1).
15
16 def truncate (obj,n):
17
18     ans = Ex("0") # create a Cadabra object with value zero
19
20     for i in range (0,n+1):
21         foo := @(obj).
22         bah = Ex("num = " + str(i))
23         distribute (foo)
24         keep_weight (foo, bah)
25         ans = ans + foo
26
27     return ans
28
29 gab := g_{a b}
30         - (1/3) x^{c} x^{d} R_{a c b d}
31         - (1/6) x^{c} x^{d} x^{e} \nabla_{c}{R_{a d b e}}
32         + (1/180) x^{c} x^{d} x^{e} x^{f} ( 8 g^{g h} R_{a c d g} R_{b e f h}
33                                         -9 \nabla_{c d}{R_{a e b f}} ). # cdb (ex-10.gab.000,gab)
34
35 iab := g^{a b}
36         + (1/3) x^{c} x^{d} g^{a e} g^{b f} R_{c e d f}
```

```

37      + (1/6) x^{c} x^{d} x^{e} g^{a f} g^{b g} \nabla_{c}\{R_{d f e} g\}
38      + (1/60) x^{c} x^{d} x^{e} x^{f} g^{a g} g^{b h}
39          ( 4 g^{i j} R_{c g d i} R_{e h f j}
40          + 3 \nabla_{c d}\{R_{e g f h}\} ).
41
42      distribute (gab)
43      distribute (iab)
44
45      gxab := gx_{a b} -> @(gab).
46
47      eps := d^{a}_{i} d^{b}_{j} d^{c}_{k}.      # cdb (ex-10.eps.001,eps) # includes a factor of 1/3!
48      asym (eps,$^{a},^{b},^{c}$)             # cdb (ex-10.eps.002,eps)
49
50      # compute negative detg rather than det g, note 1/3! included in eps
51      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)
52
53      substitute      (Ndetg,gxab)                    # cdb (ex-10.Ndetg.002,Ndetg)
54      distribute      (Ndetg)                          # cdb (ex-10.Ndetg.003,Ndetg)
55      Ndetg = truncate (Ndetg,4)                       # cdb (ex-10.Ndetg.004,Ndetg)
56      substitute      (Ndetg,$g^{a b} g_{b c} -> d^{a}_{c}$,repeat=True)      # cdb (ex-10.Ndetg.005,Ndetg)
57      eliminate_kronecker (Ndetg)                     # cdb (ex-10.Ndetg.006,Ndetg)
58      sort_product     (Ndetg)                         # cdb (ex-10.Ndetg.007,Ndetg)
59      rename_dummies   (Ndetg)                         # cdb (ex-10.Ndetg.008,Ndetg)
60      canonicalise     (Ndetg)                         # cdb (ex-10.Ndetg.009,Ndetg)
61
62      # introduce the Ricci tensor
63
64      substitute (Ndetg,$R_{a b c d} g^{a c} -> R_{b d}$,repeat=True)      # cdb (ex-10.Ndetg.101,Ndetg)
65      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)
66      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)
67
68      # the following was based on sqrt-Ndetg.tex
69
70      sqrtNdetg := 1/2 + (1/2) @(Ndetg)
71          - (1/8) (1/9) R_{a b} R_{c d} x^{a} x^{b} x^{c} x^{d}
72          - (1/4) (1/18) R_{a b} \nabla_{c}\{R_{d e}\} x^{a} x^{b} x^{c} x^{d} x^{e}.
73
74          # cdb (ex-10.sqrtNdetg.001,sqrtNdetg)

```

```

75 sort_product (sqrtNdetg) # cdb (ex-10.sqrtNdetg.002,sqrtNdetg)
76 rename_dummies (sqrtNdetg) # cdb (ex-10.sqrtNdetg.003,sqrtNdetg)
77 canonicalise (sqrtNdetg) # cdb (ex-10.sqrtNdetg.004,sqrtNdetg)
78
79 logNdetg := -1 + @(Ndetg)
80 - (1/2) (1/9) R_{a b} R_{c d} x^{a} x^{b} x^{c} x^{d}
81 - (1/18) R_{a b} \nabla_{c}{R_{d e}} x^{a} x^{b} x^{c} x^{d} x^{e}.
82 # cdb (ex-10.logNdetg.001,logNdetg)
83
84 sort_product (logNdetg) # cdb (ex-10.logNdetg.002,logNdetg)
85 rename_dummies (logNdetg) # cdb (ex-10.logNdetg.003,logNdetg)
86 canonicalise (logNdetg) # cdb (ex-10.logNdetg.004,logNdetg)
87
88 # =====
89 # the remaining code is just for pretty printing
90
91 def product_sort (obj):
92     substitute (obj,$ x^{a} -> A000^{a} $)
93     substitute (obj,$ g^{a b} -> A001^{a b} $)
94     substitute (obj,$ \nabla_{c}{R_{a b}} -> A004_{a b c} $)
95     substitute (obj,$ \nabla_{c d}{R_{a b}} -> A005_{a b c d} $)
96     substitute (obj,$ \nabla_{c d e}{R_{a b}} -> A006_{a b c d e} $)
97     substitute (obj,$ \nabla_{c d e f}{R_{a b}} -> A007_{a b c d e f} $)
98     substitute (obj,$ \nabla_{e}{R_{a b c d}} -> A008_{a b c d e} $)
99     substitute (obj,$ \nabla_{e f}{R_{a b c d}} -> A009_{a b c d e f} $)
100    substitute (obj,$ \nabla_{e f g}{R_{a b c d}} -> A010_{a b c d e f g} $)
101    substitute (obj,$ \nabla_{e f g h}{R_{a b c d}} -> A011_{a b c d e f g h} $)
102    substitute (obj,$ R_{a b} -> A002_{a b} $)
103    substitute (obj,$ R_{a b c d} -> A003_{a b c d} $)
104    sort_product (obj)
105    rename_dummies (obj)
106    substitute (obj,$ A000^{a} -> x^{a} $)
107    substitute (obj,$ A001^{a b} -> g^{a b} $)
108    substitute (obj,$ A002_{a b} -> R_{a b} $)
109    substitute (obj,$ A003_{a b c d} -> R_{a b c d} $)
110    substitute (obj,$ A004_{a b c} -> \nabla_{c}{R_{a b}} $)
111    substitute (obj,$ A005_{a b c d} -> \nabla_{c d}{R_{a b}} $)
112    substitute (obj,$ A006_{a b c d e} -> \nabla_{c d e}{R_{a b}} $)

```

```

113 substitute (obj,$ A007_{a b c d e f}      -> \nabla_{c d e f}{R_{a b}}      $)
114 substitute (obj,$ A008_{a b c d e}        -> \nabla_{e}{R_{a b c d}}      $)
115 substitute (obj,$ A009_{a b c d e f}      -> \nabla_{e f}{R_{a b c d}}      $)
116 substitute (obj,$ A010_{a b c d e f g}    -> \nabla_{e f g}{R_{a b c d}}    $)
117 substitute (obj,$ A011_{a b c d e f g h}  -> \nabla_{e f g h}{R_{a b c d}}  $)
118
119 def get_term (obj,n):
120
121     x^{a}::Weight(label=xnum).
122
123     foo := @(obj).
124     bah = Ex("xnum = " + str(n))
125     keep_weight (foo,bah)
126
127     return foo
128
129 def reformat (obj,scale):
130     foo = Ex(str(scale))
131     bah := @(foo) @(obj).
132     distribute      (bah)
133     product_sort    (bah)
134     rename_dummies  (bah)
135     canonicalise    (bah)
136     sort_sum        (bah)
137     factor_out      (bah,$x^{a?}$)
138     ans := @(bah) / @(foo).
139     return ans
140
141 def rescale (obj,scale):
142     foo = Ex(str(scale))
143     bah := @(foo) @(obj).
144     distribute      (bah)
145     factor_out      (bah,$x^{a?}$)
146     return bah
147
148 # -----
149 # reformat Ndetg
150

```

```

151 Rterm0 = get_term (Ndetg,0)      # cdb (ex-10.Rterm0.701,Rterm0)
152 Rterm1 = get_term (Ndetg,1)      # cdb (ex-10.Rterm1.701,Rterm1)
153 Rterm2 = get_term (Ndetg,2)      # cdb (ex-10.Rterm2.701,Rterm2)
154 Rterm3 = get_term (Ndetg,3)      # cdb (ex-10.Rterm3.701,Rterm3)
155 Rterm4 = get_term (Ndetg,4)      # cdb (ex-10.Rterm4.701,Rterm4)
156
157 Rterm0 = reformat (Rterm0, 1)     # cdb (ex-10.Rterm0.702,Rterm0)
158 Rterm1 = reformat (Rterm1, 1)     # cdb (ex-10.Rterm1.702,Rterm1)
159 Rterm2 = reformat (Rterm2, 3)     # cdb (ex-10.Rterm2.702,Rterm2)
160 Rterm3 = reformat (Rterm3, 6)     # cdb (ex-10.Rterm3.702,Rterm3)
161 Rterm4 = reformat (Rterm4,180)    # cdb (ex-10.Rterm4.702,Rterm4)
162
163 Ndetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.Ndetg.701,Ndetg)
164
165 # -----
166 # reformat sqrtNdetg
167
168 Rterm0 = get_term (sqrtNdetg,0)   # cdb (ex-10.Rterm0.801,Rterm0)
169 Rterm1 = get_term (sqrtNdetg,1)   # cdb (ex-10.Rterm1.801,Rterm1)
170 Rterm2 = get_term (sqrtNdetg,2)   # cdb (ex-10.Rterm2.801,Rterm2)
171 Rterm3 = get_term (sqrtNdetg,3)   # cdb (ex-10.Rterm3.801,Rterm3)
172 Rterm4 = get_term (sqrtNdetg,4)   # cdb (ex-10.Rterm4.801,Rterm4)
173
174 Rterm0 = reformat (Rterm0, 1)     # cdb (ex-10.Rterm0.802,Rterm0)
175 Rterm1 = reformat (Rterm1, 1)     # cdb (ex-10.Rterm1.802,Rterm1)
176 Rterm2 = reformat (Rterm2, 6)     # cdb (ex-10.Rterm2.802,Rterm2)
177 Rterm3 = reformat (Rterm3, 12)    # cdb (ex-10.Rterm3.802,Rterm3)
178 Rterm4 = reformat (Rterm4,360)    # cdb (ex-10.Rterm4.802,Rterm4)
179
180 sqrtNdetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.sqrtNdetg.801,sqrtNdetg)
181
182 # -----
183 # reformat logNdetg
184
185 Rterm0 = get_term (logNdetg,0)    # cdb (ex-10.Rterm0.801,Rterm0)
186 Rterm1 = get_term (logNdetg,1)    # cdb (ex-10.Rterm1.801,Rterm1)
187 Rterm2 = get_term (logNdetg,2)    # cdb (ex-10.Rterm2.801,Rterm2)
188 Rterm3 = get_term (logNdetg,3)    # cdb (ex-10.Rterm3.801,Rterm3)

```

```

189 Rterm4 = get_term (logNdetg,4)      # cdb (ex-10.Rterm4.801,Rterm4)
190
191 Rterm0 = reformat (Rterm0, 1)      # cdb (ex-10.Rterm0.802,Rterm0)
192 Rterm1 = reformat (Rterm1, 1)      # cdb (ex-10.Rterm1.802,Rterm1)
193 Rterm2 = reformat (Rterm2, 3)      # cdb (ex-10.Rterm2.802,Rterm2)
194 Rterm3 = reformat (Rterm3, 6)      # cdb (ex-10.Rterm3.802,Rterm3)
195 Rterm4 = reformat (Rterm4,180)     # cdb (ex-10.Rterm4.802,Rterm4)
196
197 logNdetg := @(Rterm0) + @(Rterm1) + @(Rterm2) + @(Rterm3) + @(Rterm4).  # cdb (ex-10.logNdetg.901,logNdetg)
198
199 checkpoint.append (Ndetg)
200 checkpoint.append (sqrtNdetg)
201 checkpoint.append (logNdetg)

```


The metric determinant in Riemann normal coordinates

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

The volume element in RNC

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

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

The log of -detg in RNC

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

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