

## Exercise 7.1 C-code for a $R_{ab}$ for a generic metric

```
1 {x,y,z}::Coordinate.
2 {a,b,c,d,e,f,g,h,i,j,k,l,m,n,o,p,q,r,s,t,u#}::Indices(values={x,y,z},position=independent).
3
4 \partial{#}::PartialDerivative;
5
6 g_{a b}::Metric;
7 g^{a b}::InverseMetric;
8
9 import cdblib
10
11 FourRab = cdblib.get ('FourRab','ex-0309.json')
12
13 Rab := 1/4 @(FourRab).
14
15 substitute (Rab, $ \partial_{a b}{g_{c d}} -> dg_{c d a b} $)
16 substitute (Rab, $ \partial_a{g_{b c}} -> dg_{b c a} $)
17
18 # -----
19 # build rules to export Cadabra expressions to Python
20 # use known symmetries for g_{a b}, dg_{ab,c,d} etc.
21 # note: replacements must not contain underscores (reserved for subscripts),
22 #      so g_{x x}-> g_xx is not allowed
23
24 gabRule := {g_{x x} -> gxx, g_{x y} -> gxy, g_{x z} -> gxz,
25             g_{y x} -> gxy, g_{y y} -> gyy, g_{y z} -> gyz,
26             g_{z x} -> gxz, g_{z y} -> gyz, g_{z z} -> gzz}.
27
28 iabRule := {g^{x x} -> ixx, g^{x y} -> ixy, g^{x z} -> ixz,
29             g^{y x} -> ixy, g^{y y} -> iyy, g^{y z} -> iyz,
30             g^{z x} -> ixz, g^{z y} -> iyz, g^{z z} -> izz}.
31
32 d1gabRule := {dg_{x x x} -> dgxxx, dg_{x y x} -> dgxyx, dg_{x z x} -> dgxzx,
33              dg_{y x x} -> dgxyx, dg_{y y x} -> dgyyx, dg_{y z x} -> dgyzx,
34              dg_{z x x} -> dgxzx, dg_{z y x} -> dgyzx, dg_{z z x} -> dgzzx,
35
36              dg_{x x y} -> dgxxy, dg_{x y y} -> dgxyy, dg_{x z y} -> dgxzy,
```

```

37         dg_{y x y} -> dgxyy, dg_{y y y} -> dgyyy, dg_{y z y} -> dgyzy,
38         dg_{z x y} -> dgxzy, dg_{z y y} -> dgyzy, dg_{z z y} -> dgzzy,
39
40         dg_{x x z} -> dgxxz, dg_{x y z} -> dgxyz, dg_{x z z} -> dgxzz,
41         dg_{y x z} -> dgxyz, dg_{y y z} -> dgyyz, dg_{y z z} -> dgyzz,
42         dg_{z x z} -> dgxzz, dg_{z y z} -> dgyzz, dg_{z z z} -> dgzzz}.
43
44     d2gabRule := {dg_{x x x x} -> dgxxxx, dg_{x y x x} -> dgxyxx, dg_{x z x x} -> dgxzzx,
45         dg_{y x x x} -> dgxyxx, dg_{y y x x} -> dgyyxx, dg_{y z x x} -> dgyzxx,
46         dg_{z x x x} -> dgxzzx, dg_{z y x x} -> dgyzxx, dg_{z z x x} -> dgzzxx,
47         dg_{x x y x} -> dgxxyx, dg_{x y y x} -> dgxyyx, dg_{x z y x} -> dgxzyx,
48         dg_{y x y x} -> dgxyyx, dg_{y y y x} -> dgyyyx, dg_{y z y x} -> dgyzyx,
49         dg_{z x y x} -> dgxzyx, dg_{z y y x} -> dgyzyx, dg_{z z y x} -> dgzzyx,
50         dg_{x x z x} -> dgxxzx, dg_{x y z x} -> dgxyzx, dg_{x z z x} -> dgxzzx,
51         dg_{y x z x} -> dgxyzx, dg_{y y z x} -> dgyyzx, dg_{y z z x} -> dgyzzx,
52         dg_{z x z x} -> dgxzzx, dg_{z y z x} -> dgyzzx, dg_{z z z x} -> dgzzzx,
53
54         dg_{x x x y} -> dgxxxxy, dg_{x y x y} -> dgxyxy, dg_{x z x y} -> dgxzxy,
55         dg_{y x x y} -> dgxyxy, dg_{y y x y} -> dgyyxy, dg_{y z x y} -> dgyzxy,
56         dg_{z x x y} -> dgxzxy, dg_{z y x y} -> dgyzxy, dg_{z z x y} -> dgzzxy,
57         dg_{x x y y} -> dgxxyy, dg_{x y y y} -> dgxyyy, dg_{x z y y} -> dgxzyy,
58         dg_{y x y y} -> dgxyyy, dg_{y y y y} -> dgyyyy, dg_{y z y y} -> dgyzyy,
59         dg_{z x y y} -> dgxzyy, dg_{z y y y} -> dgyzyy, dg_{z z y y} -> dgzzyy,
60         dg_{x x z y} -> dgxxzy, dg_{x y z y} -> dgxyzy, dg_{x z z y} -> dgxzzy,
61         dg_{y x z y} -> dgxyzy, dg_{y y z y} -> dgyyzy, dg_{y z z y} -> dgyzzy,
62         dg_{z x z y} -> dgxzzy, dg_{z y z y} -> dgyzzy, dg_{z z z y} -> dgzzzy,
63
64         dg_{x x x z} -> dgxxxz, dg_{x y x z} -> dgxyxz, dg_{x z x z} -> dgxzzz,
65         dg_{y x x z} -> dgxyxz, dg_{y y x z} -> dgyyxz, dg_{y z x z} -> dgyzzz,
66         dg_{z x x z} -> dgxzzz, dg_{z y x z} -> dgyzzz, dg_{z z x z} -> dgzzzz,
67         dg_{x x y z} -> dgxxyz, dg_{x y y z} -> dgxyyz, dg_{x z y z} -> dgxzyz,
68         dg_{y x y z} -> dgxyyz, dg_{y y y z} -> dgyyyz, dg_{y z y z} -> dgyzyz,
69         dg_{z x y z} -> dgxzyz, dg_{z y y z} -> dgyzyz, dg_{z z y z} -> dgzzyz,
70         dg_{x x z z} -> dgxxzz, dg_{x y z z} -> dgxyzz, dg_{x z z z} -> dgxzzz,
71         dg_{y x z z} -> dgxyzz, dg_{y y z z} -> dgyyzz, dg_{y z z z} -> dgyzzz,
72         dg_{z x z z} -> dgxzzz, dg_{z y z z} -> dgyzzz, dg_{z z z z} -> dgzzzz}.
73
74     def write_code (obj,name,filename,rank):

```

```

75
76 import os
77
78 from sympy.printing.ccode import C99CodePrinter as printer
79 from sympy.printing.codeprinter import Assignment
80
81 idx=[] # indices in the form [{x, x}, {x, y} ...]
82 lst=[] # corresponding terms [termxx, termxy, ...]
83
84 for i in range( len(obj[rank]) ): # rank = number of free indices
85     idx.append( str(obj[rank][i][0]._sympy_()) ) # indices for this term
86     lst.append( str(obj[rank][i][1]._sympy_()) ) # the matching term
87
88 mat = sympy.Matrix([lst]) # row vector of terms
89 sub_exprs, simplified_rhs = sympy.cse(mat) # optimise code
90
91 with open(os.getcwd() + '/' + filename, 'w') as out:
92
93     for lhs, rhs in sub_exprs:
94         out.write(printer().doprint(Assignment(lhs, rhs))+'\n')
95
96     for index, rhs in enumerate (simplified_rhs[0]):
97         lhs = sympy.Symbol(name+' '+(idx[index]).replace(' ', ',')['])
98         out.write(printer().doprint(Assignment(lhs, rhs))+'\n')
99
100 evaluate (Rab, gabRule+d1gabRule+d2gabRule+iabRule, simplify=False)
101
102 write_code (Rab, 'Rab', 'ex-0701-rab.c',2)

```

The code for  $R_{ab}$  can be found in the file `ex-0701-rab.c`. It is long and it would require more work to turn it into something useful in a numerical code. For example, functions would be needed to compute the first and second partial derivatives of the metric. But that is not a Cadabra issue.