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

```
{x,y,z}::Coordinate.
              \{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).
              \partial{#}::PartialDerivative.
              g_{a b}::Metric.
              g^{a b}::InverseMetric.
              import cdblib
10
              FourRab = cdblib.get ('FourRab', 'ex-0309.json')
11
12
              Rab := 1/4 @(FourRab).
14
              substitute (Rab, $ \partial_{a b}{g_{c d}} -> dg_{c d a b} $)
15
              substitute (Rab, \ \partial_{a}{g_{b c}} -> dg_{b c a} $)
16
17
              # build rules to export Cadabra expressions to Python
              # use known symmetries for g_{a b}, dg_{ab,c,d} etc.
              # note: replacements must not contain underscores (reserved for subscripts),
21
                                   so g_{x} = x - g_{x} is not allowed
22
23
              gabRule := \{g_{x x} -> gxx, g_{x y} -> gxy, g_{x z} -> gxz,
                                              g_{y} = \{y \mid x\} -> gxy, g_{y} -> gyy, g_{y} -> gyz,
                                              g_{z} = \{z \mid z\} -> gxz, g_{z} = \{z \mid z\} -> gzz\}.
27
              iabRule := \{g^{x} = x\} \rightarrow ixx, g^{x} \rightarrow ixy, g^{x} \rightarrow ixy, g^{x} = x\} \rightarrow ixz,
28
                                              g^{y} = x^{y} - ixy, g^{y} - iyy, g^{y} - iyz,
29
                                              g^{z} = x^{-1} = x^
30
31
              d1gabRule := \{dg_{x x x} -> dgxxx, dg_{x y x} -> dgxyx, dg_{x z x} -> dgxzx,
32
                                                    dg_{y x x} \rightarrow dgxyx, dg_{y y x} \rightarrow dgyyx, dg_{y z x} \rightarrow dgyzx,
33
                                                    dg_{z x x} \rightarrow dgxzx, dg_{z y x} \rightarrow dgyzx, dg_{z z x} \rightarrow dgzzx,
34
35
                                                    dg_{x y} - dg_{xy}, dg_{x y} - dg_{xy}, dg_{x z} - dg_{xy}
```

```
dg_{y x y} \rightarrow dgxyy, dg_{y y y} \rightarrow dgyyy, dg_{y z y} \rightarrow dgyzy,
37
                        dg_{z x y} \rightarrow dgxzy, dg_{z y y} \rightarrow dgyzy, dg_{z z y} \rightarrow dgzzy,
38
                        dg_{x z} -> dgxxz, dg_{x z} -> dgxyz, dg_{x z} -> dgxzz,
                        dg_{y z} \rightarrow dg_{y z}, dg_{y z} \rightarrow dg_{y z}, dg_{y z} \rightarrow dg_{y z},
41
                        dg_{z x z} \rightarrow dgxzz, dg_{z y z} \rightarrow dgyzz, dg_{z z} \rightarrow dgzzz.
42
43
      d2gabRule := \{dg_{x x x x}\} \rightarrow dgxxxx, dg_{x y x x} \rightarrow dgxyxx, dg_{x z x x} \rightarrow dgxzxx,
44
                        dg_{y x x} -> dgxyxx, dg_{y x x} -> dgyyxx, dg_{y z x} -> dgyzxx,
45
                        dg_{z \times x} - dg_{z \times x}, dg_{z \times x} - dg_{z \times x}, dg_{z \times x} - dg_{z \times x},
46
                        dg_{x y y} \rightarrow dgxxy, dg_{x y y x} \rightarrow dgxyy, dg_{x y y} \rightarrow dgxyy,
47
                        dg_{y x y x} \rightarrow dgxyxy, dg_{y y y x} \rightarrow dgyyxy, dg_{y z y x} \rightarrow dgyzxy,
                        dg_{z} = x y x -> dgxzxy, dg_{z} = x y x -> dgyzxy, dg_{z} = x y x -> dgzzxy,
49
                        dg_{x x z x} \rightarrow dgxxxz, dg_{x y z x} \rightarrow dgxyxz, dg_{x z z x} \rightarrow dgxzxz,
                        dg_{y x z x} \rightarrow dgxyxz, dg_{y y z x} \rightarrow dgyyxz, dg_{y z z x} \rightarrow dgyzxz,
                        dg_{z} = x z + - dgxzxz, dg_{z} = x + - dgyzxz, dg_{z} = x + - dgzzxz,
53
                        dg_{x x x y} \rightarrow dgxxxy, dg_{x y x y} \rightarrow dgxyxy, dg_{x z x y} \rightarrow dgxzxy,
54
                        dg_{y x x y} \rightarrow dgxyxy, dg_{y x y} \rightarrow dgyyxy, dg_{y z x y} \rightarrow dgyzxy,
55
                        dg_{z \times y} \rightarrow dgxzy, dg_{z \times y} \rightarrow dgyzy, dg_{z \times y} \rightarrow dgzzy,
                        dg_{x y y} \rightarrow dgxyy, dg_{x y y} \rightarrow dgxyy, dg_{x z y y} \rightarrow dgxyy,
                        dg_{y x y y} -> dgxyyy, dg_{y y y y} -> dgyyyy, dg_{y z y y} -> dgyzyy,
58
                        dg_{z} = x y  y -> dgxzyy, dg_{z} = x y  y -> dgyzyy, dg_{z} = x y  y -> dgzzyy,
59
                        dg_{x z y} -> dgxyz, dg_{x z y} -> dgxyyz, dg_{x z z y} -> dgxzyz,
                        dg_{y x z y} \rightarrow dg_{y y z}, dg_{y y z y} \rightarrow dg_{y y z}, dg_{y z z y} \rightarrow dg_{y z y z},
61
                        dg_{z x z y} \rightarrow dgxzyz, dg_{z y z y} \rightarrow dgyzyz, dg_{z z z y} \rightarrow dgzzyz,
                        dg_{x x x z} \rightarrow dgxxz, dg_{x y x z} \rightarrow dgxyz, dg_{x z x z} \rightarrow dgxzzz,
64
                        dg_{y x x z} \rightarrow dgxyxz, dg_{y y x z} \rightarrow dgyyxz, dg_{y z x z} \rightarrow dgyzxz,
65
                        dg_{z \times z} \rightarrow dgxzxz, dg_{z \times z} \rightarrow dgyzxz, dg_{z \times z} \rightarrow dgzzxz,
66
                        dg_{x y z} \rightarrow dgxyz, dg_{x y z} \rightarrow dgxyyz, dg_{x z y z} \rightarrow dgxzyz,
67
                        dg_{y x y z} \rightarrow dgxyyz, dg_{y y y z} \rightarrow dgyyyz, dg_{y z y z} \rightarrow dgyzyz,
                        dg_{z} = x y z -> dgxzyz, dg_{z} = x y z -> dgyzyz, dg_{z} = x y z -> dgzzyz,
                        dg_{x z z} \rightarrow dgxzz, dg_{x z z} \rightarrow dgxzz, dg_{x z z} \rightarrow dgxzzz,
70
                        dg_{y z z} \rightarrow dgxyzz, dg_{y z z} \rightarrow dgyyzz, dg_{y z z} \rightarrow dgyzzz,
71
                        dg_{z} = x z  -> dgxzzz, dg_{z} = x z -> dgyzzz, dg_{z} = x z -> dgzzzz.
72
73
      def write_code (obj,name,filename,rank):
```

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

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.