Preprint: arXiv:0903.2087
Journal: *Class.Quantum.Grav.* **26** 175017 (2009)

Riemann Normal Coordinate expansions using Cadabra

Leo Brewin

School of Mathematical Sciences
Monash University, 3800
Australia

12-Aug-2009

Abstract

We present the results of using the computer algebra program Cadabra to develop Riemann normal coordinate expansions of the metric and other geometrical quantities, in particular the geodesic arc-length. All of the results are given to sixth-order in the curvature tensor.

1 Introduction

In a previous paper [1] we demonstrated, through a series of simple examples, how a new computer algebra program, Cadabra ([2], [3], [4], [5]) could be employed to do the kinds of tensor computations often encountered in General Relativity. The examples were deliberately chosen to be simple. So it is reasonable to wonder how Cadabra might handle much more challenging computations. To explore this question we put Cadabra to the task of computing the Riemann normal coordinate expansions of the metric and other geometrical quantities, in particular the geodesic arc length. Such computations are known to be, at higher orders, very demanding and prohibitively difficult to compute by hand. Here we will show that Cadabra handles these computations with ease.

There is also a direct practical purpose to these calculations. We are developing ([6], [7]) an approach to numerical relativity that uses a lattice, similar to that used in the Regge calculus ([8], [9], [10]), as a way to record the

metric (through the table of leg-lengths) and topology (through the connections between vertices) of the spacetime. Central to that approach is the use of Riemann normal coordinates in the computation of the Riemann curvatures given just the leg-lengths and (some) angles. For this to work we need an equation that links the curvatures to the geodesic arc-length (the leg-lengths). The result is equation (11.21) in section (11).

The basic idea behind Riemann normal coordinates is to use the geodesics through a given point to define the coordinates for nearby points. Let the given point be O (this will be the origin of the Riemann normal frame) and consider some nearby point P. If P is sufficiently close to O then there exists a unique geodesic joining O to P. Let v^a be the components of the unit tangent vector to this geodesic at O and let s be the geodesic arc length measured from O to P. Then the Riemann normal coordinates of P relative to O are defined to be $x^a = sv^a$. These coordinates are well defined provided the geodesics do not cross (which we can always ensure by choosing the neighbourhood of O to be sufficiently small).

One trivial consequence of this definition is that all geodesics through O are of the form $x^a(s) = sv^a$ and that the v^a are constant along each geodesic. This implies, by direct substitution into the geodesic equation, that $\Gamma^c{}_{ab} = 0$ at O which in turn implies that $g_{ab,c} = 0$ at O. Suppose now that we were to expand the metric as a Taylor series in x^a about O. In that series there would only be the zero, second and higher derivatives of the g_{ab}. Thus the leading terms of the metric could be expressed as a sum of a constant part plus a curvature part. If the curvature is weak this can be interpreted as an expansion of the metric in powers (and derivatives) of the curvature. Likewise one can imagine similar expansions of other geometrical quantities (e.g. geodesics, arc length) in terms of a flat space part plus a curvature contribution.

Higher order expansions are extremely tedious to compute by hand. Some hardy souls ([11], [12], [13]) have endured the journey (Müller et al ([14]) venture as far as 11-th order expansions for the metric (i.e. to terms involving the 8-th derivative of the curvatures, the notion of order will be defined in the following section). However most people settle for just the first two non-trivial curvature terms (i.e. R and ∇R).

As we will see, the easiest expansion to compute is that for the metric in powers of the curvatures and its derivatives. Much more challenging is the expression that allows the Riemann normal coordinates to be constructed from a given set of coordinates. This later calculation involves the solution of a two point boundary value problem – not a job for the faint-hearted.

The expansion of the metric in Riemann normal form can be found in many articles. For those with a mathematical bent see ([15], [16], [17]) and in particular the elegant exposition by Gray ([18], [19]). For applications in physics see ([20], [21], [22]).

2 Conformal coordinates

Each algorithm given later in this paper yields polynomial approximations to particular geometric quantities (e.g. the metric). Higher order approximations are obtained by recursive application of the algorithms.

In this section we will define what we mean when we say that the polynomial S_ϵ is an expansion of S up to and including terms of order $\mathcal{O}(\epsilon^n)$. We will do so by introducing a conformal transformation of the original metric.

Consider some neighbourhood of O and let ϵ be a typical length scale for O (for example, ϵ might be the length of the largest geodesic that passes through O and confined by the neighbourhood). Construct any regular set of coordinates x^a (i.e. such that the metric components are non-singular) in the neighbourhood of O and let the coordinates of O be $x^a_\star$. We will use the word *patch* to denote the neighbourhood of O in which these coordinates are defined. Now define a new set of coordinates y^a by

$$x^a = x^a_\star + \epsilon y^a$$

and thus

$$ds^2 = g_{ab}(x)dx^a dx^b = \epsilon^2 g_{ab}(x_\star + \epsilon y)dy^a dy^b$$

Now define the conformal metric $d\tilde{s}$ by $d\tilde{s} = ds/\epsilon^2$. This leads to

$$d\tilde{s}^2 = g_{ab}(x_\star + \epsilon y)dy^a dy^b = \tilde{g}_{ab}(y,\epsilon)dy^a dy^b$$

and

$$\tilde{g}_{ab} = g_{ab}\,, \qquad \tilde{g}_{ab,c} = \epsilon g_{ab,c}\,, \qquad \tilde{g}_{ab,cd} = \epsilon^2 g_{ab,cd} \qquad \text{at } O$$

where the partial derivatives on the left are with respect to y and those on the right are with respect to x. Since $g_{ab}(x_\star)$ does not depend on ϵ we have the general result that

$$\tilde{g}_{ab,i_1 i_2 i_3 \cdots i_n} = \mathcal{O}(\epsilon^n) \qquad \text{at } O$$

From this it follows, by simple inspection of the standard equations, that

$$\tilde{\Gamma}^a{}_{bc,i_1i_2i_3\cdots i_n} = \mathcal{O}\left(\epsilon^{n+1}\right) \qquad \text{at } O$$
$$\tilde{R}^a{}_{bcd,i_1i_2i_3\cdots i_n} = \mathcal{O}\left(\epsilon^{n+2}\right) \qquad \text{at } O$$

There are now two ways to look at the patch. We can view it as patch of length scale ϵ with a curvature independent of ϵ. Or we can view it as patch of fixed size but with a curvature that depends on ϵ (and where the limit $\epsilon \to 0$ corresponds to flat space). This later view is useful since in using it we can be sure that the series expansions around flat space are convergent (for a sufficiently small ϵ).

We will use these conformal coordinates for the remainder of this paper. As there is no longer any reason to distinguish between x^a and y^a we replace y^a with x^a. The x^a will now be treated as generic coordinates (but keep in mind that we are working in a conformal gauge).

Finally, when we say that S_ϵ is an expansion of S up to and including terms of order $\mathcal{O}\left(\epsilon^n\right)$ we mean that

$$0 < \lim_{\epsilon \to 0} \frac{|S - S_\epsilon|}{\epsilon^{n+1}} < M$$

for some finite positive M i.e. if S were expanded as a Taylor series in ϵ around $\epsilon = 0$ then S and S_ϵ would differ by terms proportional to ϵ^{n+1}.

3 Riemann Normal Coordinates

Suppose, as is almost always the case, that our coordinates x^a are *not* in Riemann normal form. How might we transform to a local set of Riemann normal coordinates? If we were to appeal to the simple definition $y^a = sv^a$ we would soon encounter a hurdle. The quantities v^a are rarely known explicitly but must instead be computed by solving a two-point boundary value problem. This is non-trivial but it can be dealt with in stages. First construct a Taylor series expansion for an arbitrary geodesic passing through O. This solution to the initial value problem will depend on two integration constants, x^a and $\dot{x}^a$, being the respective values of $x^a(s)$ and dx^a/ds at $s = 0$. Next use a fixed-point iterative scheme to produce successive approximations to $\dot{x}^a$ so that the geodesic passes through both points O and P. Then finally compute v^a as dx^a/ds at $s = 0$. This is exactly the plan which we will follow.

It is well known that Riemann normal coordinates can always be constructed *locally* around any non singular point (see [19]). Thus we will not concern ourselves here with issues such as existence and convergence but rather we will focus our attention on developing algorithms for expressing various quantities in Riemann normal form.

3.1 The initial value problem

Our aim here is to obtain a Taylor series, about the point O, for solution of the geodesic equation

$$0 = \frac{d^2x^a}{ds^2} + \Gamma^a{}_{bc}(x)\frac{dx^b}{ds}\frac{dx^c}{ds}$$

subject to the initial conditions $x^a(s) = x^a$ and $dx^a/ds = \dot{x}^a$ at $s = 0$.

We choose $s = 0$ at O and we write the Taylor series for $x^a(s)$ as

$$x^a(s) = x^a|_{s=0} + s\left.\frac{dx^a}{ds}\right|_{s=0} + \sum_{n=2}^{\infty}\left.\frac{s^n}{n!}\frac{d^nx^a}{ds^n}\right|_{s=0}$$

The second and higher derivatives can be obtained by successive differentiation of the geodesic equation leading to

$$x^a(s) = x^a + s\dot{x}^a - \sum_{n=2}^{\infty}\frac{s^n}{n!}\Gamma^a{}_{i_1i_2i_3\cdots i_n}\dot{x}^{i_1}\dot{x}^{i_2}\dot{x}^{i_3}\cdots\dot{x}^{i_n} \tag{3.1}$$

where the $\Gamma^a{}_{i_1i_2i_3\cdots i_n}$, known as *generalised connections*, are defined recursively by

$$\Gamma^a{}_{i_1i_2i_3\cdots i_n} = \Gamma^a{}_{(i_1i_2i_3\cdots i_{n-1},i_n)} - n\Gamma^a{}_{p(i_2i_3\cdots i_{n-1}}\Gamma^p{}_{i_1i_n)} \tag{3.2}$$

Note that the use of round brackets (...) denotes total symmetrisation over the included indices (see Appendix A for more details).

A convenient shorthand for equation (3.2) in terms of covariant derivatives can be obtained if you ignore (in this context alone) the single upper index. This leads to the compact notation

$$\Gamma^a{}_{i_1i_2i_3\cdots i_n} = \Gamma^a{}_{(i_1i_2;i_3i_4i_5\cdots i_n)} \tag{3.3}$$

3.2 The boundary value problem

Here we seek to juggle $\dot{x}^a$ so that the geodesic passes through not only O but also P. Suppose that the coordinates of P are $x^a + \Delta x^a$ where x^a are the coordinates of O. The solution (3.1) already passes through O so we have only now to force it to pass through P. Let s_P be the geodesic distance from O to P. Then our challenge is to solve

$$\Delta x^a = s_P \dot{x}^a - \sum_{n=2}^{\infty} \frac{s_P^n}{n!} \Gamma^a{}_{i_1 i_2 i_3 \cdots i_n} \dot{x}^{i_1} \dot{x}^{i_2} \dot{x}^{i_3} \cdots \dot{x}^{i_n}$$

for $\dot{x}^a$ in terms of Δx^a and the generalised connections.

Put $y^a = s_P \dot{x}^a$ (this introduces the Riemann normal coordinates) and rearrange the equation into the form

$$y^a = \Delta x^a + \sum_{n=2}^{\infty} \frac{1}{n!} \Gamma^a{}_{i_1 i_2 i_3 \cdots i_n} y^{i_1} y^{i_2} y^{i_3} \cdots y^{i_n} \tag{3.4}$$

We now plan to solve this equation for y^a by constructing a sequence of approximations y^a_m to y^a. In principle we could imagine that y^a has been found and that we have expanded it as a infinite series in powers of ϵ (i.e. as a power series in the curvatures). We will choose y^a_m to be the Taylor polynomial of y^a to order ϵ^m. That is, y^a_m is a polynomial in the curvatures (and its derivatives) up to and including terms of order $\mathcal{O}(\epsilon^m)$. We can compute y^a_m by truncating both side of (3.4) to terms no higher than $\mathcal{O}(\epsilon^m)$. But note that the $\Gamma^a{}_{i_1 i_2 i_3 \cdots i_n}$ are of order $\mathcal{O}(\epsilon^{n-1})$. The upshot is that the infinite series may be truncated at $n = m$ while still retaining all terms up to and including ϵ^m. Thus we have

$$y^a_m = \Delta x^a + T^m_\epsilon \left(\sum_{n=2}^{n=m} \frac{1}{n!} \Gamma^a{}_{i_1 i_2 i_3 \cdots i_n} y^{i_1}_m y^{i_2}_m y^{i_3}_m \cdots y^{i_n}_m \right)$$

Where T^m_ϵ is a simple truncation operator (it deletes all terms of order $\mathcal{O}(\epsilon^{m+1})$ or higher). This is a marginal improvement on (3.4) (at least we have a finite series) but it is still a non-linear equation for y^a_m. But fortunately we can do better. Notice, once again, that $\Gamma^a{}_{i_1 i_2 i_3 \cdots i_n} = \mathcal{O}(\epsilon^{n-1})$ and this allows us to use lower order estimates for y^a in the product terms on the right hand side. This leads to

$$y^a_m = \Delta x^a + T^m_\epsilon \left(\sum_{n=2}^{n=m} \frac{1}{n!} \Gamma^a{}_{i_1 i_2 i_3 \cdots i_n} y^{i_1}_{m-n+1} y^{i_2}_{m-n+1} y^{i_3}_{m-n+1} \cdots y^{i_n}_{m-n+1} \right) \tag{3.5}$$

Now we see that y^a_m appears only on the left hand side and thus we can use this equation to recursively compute y^a_p for $p = 2, 3, 4, \cdots$. Here are the first few y^a_m. We start with the lowest order approximation,

$$y^a_0 = \Delta x^a$$

and as there are no ϵ^1 terms in (3.4) we have

$$y^a_1 = y^a_0 = \Delta x^a$$

Now set $m = 2$ in equation (3.5) to obtain

$$\begin{aligned} y^a_2 &= \Delta x^a + T^2_\epsilon \left(\frac{1}{2!} \Gamma^a{}_{i_1 i_2} y^{i_1}_1 y^{i_2}_1 \right) \\ &= \Delta x^a + \frac{1}{2} \Gamma^a{}_{i_1 i_2} \Delta x^{i_1}_1 \Delta x^{i_2}_1 \end{aligned}$$

and once more, with $m = 3$, with the result

$$\begin{aligned} y^a_3 &= \Delta x^a + T^3_\epsilon \left(\frac{1}{2!} \Gamma^a{}_{i_1 i_2} y^{i_1}_2 y^{i_2}_2 + \frac{1}{3!} \Gamma^a{}_{i_1 i_2 i_3} y^{i_1}_1 y^{i_2}_1 y^{i_3}_1 \right) \\ &= \Delta x^a + \frac{1}{2} \Gamma^a{}_{i_1 i_2} \Delta x^{i_1} \Delta x^{i_2} + \frac{1}{6} \left(\Gamma^a{}_{b i_1} \Gamma^b{}_{i_2 i_3} + \Gamma^a{}_{i_1 i_2, i_3} \right) \Delta x^{i_1} \Delta x^{i_2} \Delta x^{i_3} \end{aligned}$$

This process may seem simple but looks can be deceiving – the higher order y^a_m contain a profusion of terms that, when computed by hand, are largely unmanageable beyond $m \approx 7$. We will return to this point later when we discuss the use of Cadabra to perform these computations.

This completes our first objective – to find a way to transform from any non-singular set of coordinates to a local set of Riemann normal coordinates. The question we now pose is – what form does the metric take in these coordinates? This is the subject of the second next section. But first we shall take a short moment to introduce some new notation.

4 Notation

The calculations we are undertaking are flooded with expression such as

$$\Gamma^a{}_{i_1 i_2 i_3 \cdots i_{n+1}} = \Gamma^a{}_{(i_1 i_2 i_3 \cdots i_n, i_{n+1})} - (n+1) \Gamma^a{}_{p(i_2 i_3 \cdots i_n} \Gamma^p{}_{i_1 i_{n+1})}$$

in which long stretches of indices like $i_1 i_2 i_3 \cdots i_n$ abound. This is tedious to write and, in the authors opinion, rather untidy. Here we propose a variation

on the notation that is both easy to read and write while not detracting from the meaning in the expression. The proposal is that a sequence of indices such as $i_1 i_2 i_3 \cdots i_n$ be replaced with a single index of the form $\underline{i}$. In this notation the previous equation could be written as

$$\Gamma^a{}_{b\underline{c}d} = \Gamma^a{}_{(b\underline{c},d)} - (n+1)\Gamma^a{}_{p(\underline{c}}\Gamma^p{}_{bd)}$$

where $\underline{c}$ contains $n > 0$ indices. In cases where the number of hidden indices needs to be made clear we will either say so in words (as we did in the previous example) or we will include the number as a subscript, for example

$$\Gamma^a{}_{b\underline{c}_n d} = \Gamma^a{}_{(b\underline{c}_n,d)} - (n+1)\Gamma^a{}_{p(\underline{c}_n}\Gamma^p{}_{bd)}$$

This version is however, not as clean as the previous version.

For equations like

$$0 = \Gamma^a{}_{bc,i_1 i_2 i_3 \cdots i_n} A^{i_1} A^{i_2} A^{i_3} \cdots A^{i_n}$$

we will write

$$0 = \Gamma^a{}_{bc,\underline{d}} A^{\cdot\underline{d}}$$

We chose this notation of single dot in $A^{\cdot\underline{d}}$ because of its suggestive form (of multiplication of as many copies of A as required to exhaust the indices within $\underline{d}$). Here is another common construction

$$\Big(\cdots \Big(\big(\Gamma^a{}_{bc,i_1} A^{i_1} \big)_{,i_2} A^{i_2} \Big)_{,i_3} A^{i_3} \cdots \Big)_{,i_n} A^{i_n}$$

How might we tidy this up? By including a dot before the derivative index $\underline{d}$, like this

$$\Gamma^a{}_{bc,\cdot\underline{d}} A^{\cdot\underline{d}}$$

These simple changes bring some degree of normalcy to the printed form but those gains rapidly pale into insignificance when we ask Cadabra (in section (11)) to display the results for the sixth order Riemann normal expansions. These results contain expressions with symmetries in some of the indices which when printed can stretch over many A4-pages. The reason is in part that the expressions are inherently long but also because Cadabra does not use the round-bracket notation to denote symmetrisation over a set of indices. Instead it uses the fully expanded form which, on paper, can lead to an $n!$ explosion in otherwise similar looking terms. We will try to minimise this problem as follows. Suppose we have an object A_{abcde} which we know to be symmetric on the indices (cde). We create an arbitrary B^a and instruct

Cadabra to simplify the expression $A_{abcde}B^cB^dB^e$ as much as possible. Then we set B^a equal to one just prior to printing the expression. But how do we convey to the reader that this operation has been performed? The expression that we are trying to print will always be the right hand side of some equation such as $D_{abcde} = A_{ab(cde)}$ and thus D_{abcde} is also symmetric on (cde). So when we print this equation we will include the round brackets on the left hand side. Thus we would display the results as $D_{ab(cde)} = A_{abcde}$ and we would understand that the right hand side should be symmetrised over (cde). Clearly this notational device should only ever be used at the end of the calculations.

Our final notational device concerns cases where we want to exclude an index from symmetrisation. The normal practise is to exclude an index by enclosing it in a pair of vertical lines. In our variation we will place a dot above the excluded index. Thus, where other authors might write $(ab|c|d|e|fg)$ to signify symmetrisation over only a, b, d, f and g, we will write $(ab\dot{c}d\dot{e}fg)$.

Though these variations might appear to make only marginal improvements in the printed equations they have made it considerably easier for the author in creating the LaTeX code for this document.

5 The metric in Riemann normal form

In the preceding section we chose to distinguish between generic and Riemann normal coordinates by using the symbols x^a and y^a respectively. We will now, for notational convenience and to accord with convention, revert to using x^a for the Riemann normal coordinates while stripping y^a of any special meaning.

Our aim in this section is to express the metric in Riemann normal form. This will take the form of an infinite series in powers of the curvature tensor and its derivatives. We start by writing out the Taylor series for the metric around $x^a = 0$

$$g_{ab}(x) = g_{ab} + \sum_{n=1}^{\infty} \frac{1}{n!}\, g_{ab,\underline{c}}\, x^{\cdot\underline{c}}$$

where $\underline{c}$ contains n indices and g_{ab} are constants (e.g. $g_{ab} = \mathrm{diag}(1, 1, 1, \cdots)$).

Our present task is to express the partial derivatives of the metric in terms of the Riemann tensor. From the standard definition of a metric compatible connection we have

$$g_{ab,c\underline{d}} = \left(g_{ae}\Gamma^e{}_{bc} + g_{eb}\Gamma^e{}_{ac}\right)_{,\underline{d}}$$

and since $g_{ab,c\underline{d}}$ is totally symmetric in $c\underline{d}$ we also have

$$g_{ab,c\underline{d}} = \left(g_{ae}\Gamma^{e}{}_{b(c}\right)_{,\underline{d})} + \left(g_{eb}\Gamma^{e}{}_{a(c}\right)_{,\underline{d})}$$

Two points should be noted, first, the connection appears only in the form $\Gamma^{a}{}_{b(c,\underline{d})}$, second, the left hand side contains derivatives one order higher than in the corresponding terms on the right hand side. The upshot is that we can use this equation to recursively compute all of the metric derivatives solely in terms of the $\Gamma^{a}{}_{b(c,\underline{d})}$ and the constants g_{ab}. In this way we could express the above Taylor series for the metric in terms of the connection. But we can go one stage further – the derivatives of the connection must surely tie in with the curvatures. Thus we are led to review the standard definition for the curvature, which after a series of derivatives, can be written in the form

$$R^{a}{}_{(bc\dot{d},\underline{e})} = \Gamma^{a}{}_{d(bc,\underline{e})} - \Gamma^{a}{}_{(bc,\underline{e})d} + \left(\Gamma^{a}{}_{i(c}\Gamma^{i}{}_{b\dot{d}}\right)_{,\underline{e})} - \left(\Gamma^{a}{}_{id}\Gamma^{i}{}_{(bc}\right)_{,\underline{e})}$$

(Please note that the $\Gamma^{a}{}_{dbc}$ in $\Gamma^{a}{}_{d(bc,\underline{e})}$ is the first of the generalised connections discussed earlier in section (3.1)). Can we use this to eliminate the connection and its derivatives from the metric? Yes, but only after we specialise to the Riemann normal coordinates.

Recall that, in Riemann normal coordinates, all geodesics through O are of the form

$$x^{a}(s) = sv^{a}$$

which upon substitution into the geodesic equations leads to

$$0 = \Gamma^{a}{}_{(b\underline{c})} \qquad \text{at } O$$

It follows, by recursion on equation (3.3), that

$$0 = \Gamma^{a}{}_{(bc,\underline{d})} \qquad \text{at } O \tag{5.1}$$

We also know that $\Gamma^{a}{}_{bc,\underline{e}d}$ is separately symmetric in its first pair of indices and in the remaining $(n+1)$ lower indices (assuming $\underline{e}$ contains n indices). Thus using equation (A.1) we have

$$0 = (n+3)\Gamma^{a}{}_{(bc,\underline{e}d)} = 2\Gamma^{a}{}_{d(b,c\underline{e})} + (n+1)\Gamma^{a}{}_{(bc,\underline{e})d}$$

We can use this to eliminate the $\Gamma^{a}{}_{(bc,\underline{e})d}$ term in the previous equation for the curvature. The result, after a minor shuffling of terms is

$$(n+3)\Gamma^{a}{}_{d(b,c\underline{e})} = (n+1)\left(R^{a}{}_{(bc\dot{d},\underline{e})} - \left(\Gamma^{a}{}_{i(c}\Gamma^{i}{}_{b\dot{d}}\right)_{,\underline{e})} + \left(\Gamma^{a}{}_{id}\Gamma^{i}{}_{(bc}\right)_{,\underline{e})}\right)$$

(the reason for rearranging the terms will become clear in a moment). Note also that the last term in the previous equation can be eliminated by equation (5.1) and the product rule.

In summary, the equations of interest are

$$g_{ab}(x) = g_{ab} + \sum_{n=1}^{\infty} \frac{1}{n!}\, g_{ab,\underline{c}}\, x^{\cdot\underline{c}} \tag{5.2}$$

$$g_{ab,c\underline{d}} = \left(g_{ae}\Gamma^{e}{}_{b(c}\right)_{,\underline{d})} + \left(g_{eb}\Gamma^{e}{}_{a(c}\right)_{,\underline{d})} \tag{5.3}$$

$$(n+3)\Gamma^{a}{}_{d(b,c\underline{e})} = (n+1)\left(R^{a}{}_{(bc\dot{d},\underline{e})} - \left(\Gamma^{a}{}_{i(c}\Gamma^{i}{}_{bd}\right)_{,\underline{e})}\right) \tag{5.4}$$

We use these equations as follows. First, equation (5.4) is used to recursively compute the $\Gamma^{a}{}_{b(c,\underline{e}d)}$ in terms of the Riemann tensor and its partial derivatives (this was the reason behind the shuffling of terms noted above). Note that $\underline{e}$ in equation (5.4) contains n hidden indices. The $\Gamma^{a}{}_{b(c,\underline{d})}$ are then substituted into (5.3) which in turn is used to recursively express all of the $g_{ab,\underline{c}}$ in terms of the Riemann tensor and its partial derivatives. When the dust settles we will have a finite series expansion for the metric in terms of the Riemann tensor and its partial derivatives. The result, accurate to $\mathcal{O}\left(\epsilon^4\right)$, is

$$g_{ab}\left(x\right) = g_{ab} - \frac{1}{3}\, x^c x^d R_{acbd} - \frac{1}{6}\, x^c x^d x^e \partial_c R_{adbe} + \mathcal{O}\left(\epsilon^4\right)$$

We conclude this section by introducing one final variation to the algorithm just given – an option to re-express the metric in terms of the covariant rather than partial derivatives of the curvatures.

It is not hard to see that after a series of covariant derivatives one would obtain an equation of the form, in any coordinate frame,

$$R^{a}{}_{(bc\dot{d};\underline{e})} = R^{a}{}_{(bc\dot{d},\underline{e})} + Q^{a}{}_{(bc\dot{d}\underline{e})}$$

where $Q^{a}{}_{(bc\dot{d}\underline{e})}$ is a function of the $\Gamma^{a}{}_{bc}$, the $R^{a}{}_{bcd}$ and their partial derivatives. If this is going to sit nicely with our algorithm given above then we will need to show, in the Riemann normal frame, that this equation only contains connection terms of the form $\Gamma^{p}{}_{q(r,\underline{s})}$. Fortunately this is rather easy to do.

Each term of the form $\Gamma^{p}{}_{qr,\underline{s}}$ in Q arose during one round of covariant differentiation. Thus *at least one* of the indices q, r and *all* of the indices in $\underline{s}$ must be drawn from the index list $\underline{e}$. If both q and r are contained in $\underline{e}$ then the term is of the form $\Gamma^{p}{}_{(qr,\underline{s})}$ and thus will vanish when we specialise

to the Riemann normal frame. This completes the proof. If we re-arrange the above equation into the following form

$$R^a{}_{(bc\dot{d},\underline{e})} = R^a{}_{(bc\dot{d};\underline{e})} - Q^a{}_{(bc\dot{d}\underline{e})} \tag{5.5}$$

we can use it to recursively compute all of the partial derivatives of the curvatures in terms of their covariant derivatives. The $Q^a{}_{(bc\dot{d}\underline{e})}$ will contain lower order derivatives of the curvatures and partial derivatives of the connection all of which can be eliminated (in favour of covariant derivatives) using previously computed results. For the first two derivatives we find that the partial and covariant derivatives are equal but differences do appear in higher order derivatives. More details will be given in a later section.

These calculations, as simple as they may appear, are exceedingly tedious to do except for the first few terms. The recursive nature of the calculations requires frequent substitution of one result into another which causes an explosion in the number of terms that must be handled. Not only is this tedious but its is also extremely prone to human error. Calculations of this kind are clearly best left to a computer. We shall return to this point later on.

6 The inverse metric in Riemann normal form

Most of the hard work is now behind us and we can now develop algorithms for Riemann normal expansions for other interesting quantities, in this instance the inverse metric $g^{ab}(x)$. In the previous section we used $0 = g_{ab;\underline{c}}$ as our starting point. On this occasion we start with $0 = g^{ab}{}_{;\underline{c}}$. Then, following a path similar to that used in the previous section, we arrive at the following equations

$$g^{ab}(x) = g^{ab} + \sum_{n=1}^{\infty} \frac{1}{n!}\, g^{ab}{}_{,\underline{c}}\, x^{\cdot\underline{c}} \tag{6.1}$$

$$g^{ab}{}_{,c\underline{d}} = -\left(g^{ae}\Gamma^b{}_{e(c}\right)_{,\underline{d})} - \left(g^{eb}\Gamma^a{}_{e(c}\right)_{,\underline{d})} \tag{6.2}$$

$$(n+3)\Gamma^a{}_{d(b,c\underline{e})} = (n+1)\left(R^a{}_{(bc\dot{d},\underline{e})} - \left(\Gamma^a{}_{i(c}\Gamma^i{}_{b\dot{d}}\right)_{,\underline{e})}\right) \tag{5.4}$$

These equations can be used to construct the series expansion for $g^{ab}(x)$, which to $\mathcal{O}\left(\epsilon^4\right)$ is

$$g^{ab}\left(x\right) = g^{ab} + \frac{1}{3}\, x^c x^d R^a{}_c{}^b{}_d + \frac{1}{6}\, x^c x^d x^e \partial_c R^a{}_d{}^b{}_e + \mathcal{O}\left(\epsilon^4\right)$$

7 Generalised connections

In section (3.1) we saw that the generalised connections $\Gamma^a{}_{bc\underline{d}}$ arose from successive differentiation of the geodesic equation and that they can be computed recursively using

$$\Gamma^a{}_{b\underline{c}d} = \Gamma^a{}_{(b\underline{c},d)} - (n+1)\Gamma^a{}_{p(\underline{c}}\Gamma^p{}_{bd)} \tag{3.2}$$

where the index list $\underline{c}$ contains $n > 0$ indices, or directly using

$$\Gamma^a{}_{bc\underline{d}} = \Gamma^a{}_{(bc;\underline{d})} \tag{3.3}$$

Here are the first three generalised connections

$$\Gamma^a_{(bc)}(x) = \Gamma^a{}_{bc}$$

$$\Gamma^a_{(bcd)}(x) = \partial_b\Gamma^a{}_{cd} - 2\,\Gamma^a{}_{be}\Gamma^e{}_{cd}$$

$$\begin{aligned}\Gamma^a_{(bcde)}(x) = &-\Gamma^f{}_{bc}\partial_f\Gamma^a{}_{de} - 4\,\Gamma^f{}_{bc}\partial_d\Gamma^a{}_{ef} + \partial_{bc}\Gamma^a{}_{de} \\ &+ 2\,\Gamma^a{}_{fg}\Gamma^f{}_{bc}\Gamma^g{}_{de} + 4\,\Gamma^a{}_{bf}\Gamma^f{}_{cg}\Gamma^g{}_{de} - 2\,\Gamma^a{}_{bf}\partial_c\Gamma^f{}_{de}\end{aligned}$$

and when we specialise to Riemann normal coordinates we obtain

$$\Gamma^a_{(bc)}(x) = \frac{2}{3}\,x^d R^a{}_{bdc} + \frac{1}{12}\,x^d x^e\,(2\,\nabla_b R^a{}_{dec} + 4\,\nabla_d R^a{}_{bec} + \nabla^a R_{dbec}) + \mathcal{O}\left(\epsilon^4\right)$$

$$\Gamma^a_{(bcd)}(x) = \frac{1}{2}\,x^e\nabla_b R^a{}_{ced} + \mathcal{O}\left(\epsilon^4\right)$$

$$\Gamma^a_{(bcde)}(x) = \mathcal{O}\left(\epsilon^4\right)$$

8 Geodesics

When first we spoke of Riemann normal coordinates we restricted our attention to the geodesics that passed through the point O. Now we wish to be somewhat less restrictive. We would like to know how to construct any geodesic in the neighbourhood of O. Here we will once again be building solutions of the geodesic equations and as before we will consider two separate cases, first, the geodesic initial value problem and second, the geodesic boundary value problems.

Most of the machinery that we need to tackle these questions has already been developed. Here we apply the formalism developed in sections (3.1) and (3.2) to the metric in Riemann normal form as obtained in section (5).

8.1 Geodesic initial value problem

Consider a point P distinct from O. At P we can assume that the generalised connections do not vanish (which is generally true, the exception being flat space). Thus the coordinates x^a in the neighbourhood of P do not constitute a Riemann normal frame relative to P. But as P lies in the patch for O we know that the metric is non-singular at P and thus we should be able to construct a new set of Riemann normal coordinates, y^a, with P as the origin.

We have seen this problem once before, in section (3.2). Using equation (3.1) and the generalised connections from section (7) we find

$$x^a(s) = x^a + s\dot{x}^a - \frac{1}{24}s^2\dot{x}^b\dot{x}^c\left(8\,x^d R^a{}_{bdc} + 2\,x^d x^e \nabla_b R^a{}_{dec} + 4\,x^d x^e \nabla_d R^a{}_{bec} + x^d x^e \nabla^a R_{dbec}\right) - \frac{1}{12}s^3\dot{x}^b\dot{x}^c\dot{x}^d x^e \nabla_b R^a{}_{ced} + \mathcal{O}\left(\epsilon^4\right)$$

8.2 Geodesic boundary value problem

Consider now the case where we have three distinct points O, P and Q. In this section we seek to compute the geodesic that passes through P and Q. We use the equation (3.5) of the generalised connections from section (7) to obtain

$$x^a(\lambda) = x^a + \lambda x_1^a + \lambda^2 x_2^a + \lambda^3 x_3^a + \mathcal{O}\left(\epsilon^4\right)$$

$$\begin{aligned} x_1^a = {} & \Delta x^a + \frac{1}{3}x^b \Delta x^c \Delta x^d R^a{}_{cbd} + \frac{1}{12}x^b x^c \Delta x^d \Delta x^e \nabla_d R^a{}_{bce} \\ & + \frac{1}{6}x^b x^c \Delta x^d \Delta x^e \nabla_b R^a{}_{dce} + \frac{1}{24}x^b x^c \Delta x^d \Delta x^e \nabla^a R_{bdce} \\ & + \frac{1}{12}x^b \Delta x^c \Delta x^d \Delta x^e \nabla_c R^a{}_{dbe} \end{aligned}$$

$$\begin{aligned} x_2^a = {} & -\frac{1}{3}x^b \Delta x^c \Delta x^d R^a{}_{cbd} - \frac{1}{12}x^b x^c \Delta x^d \Delta x^e \nabla_d R^a{}_{bce} \\ & - \frac{1}{6}x^b x^c \Delta x^d \Delta x^e \nabla_b R^a{}_{dce} - \frac{1}{24}x^b x^c \Delta x^d \Delta x^e \nabla^a R_{bdce} \end{aligned}$$

$$x_3^a = \left(\frac{-1}{12}\right) x^b \Delta x^c \Delta x^d \Delta x^e \nabla_c R^a{}_{dbe}$$

where $\lambda = s/L_{PQ}$ is the scaled geodesic distance from $P(\lambda = 0)$ to $Q(\lambda = 1)$.

9 Geodesic arc-length

Since we now have explicit expressions for the metric and the geodesic that joins the points P and Q we can compute the length of that geodesic by way of the integral

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

We have, to this point, taken s to be the proper distance along the geodesic. However, after careful inspection of the geodesic path (3.1) we see that any uniform scaling of s is allowed. Thus we can re-scale s so that $s = 0$ at P and $s = 1$ at Q (of course, the parameter s no longer measures proper distance). Furthermore we know that the integrand is constant along the geodesic and can thus be evaluated at any point which we shall chose to be P. Thus the integral is trivial and we have

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

Using our previous results we obtain

$$\begin{aligned} L_{PQ}^2 = g_{ab} \Delta x^{\,a} \Delta x^{\,b} - \frac{1}{3} x^a x^b \Delta x^{\,c} \Delta x^{\,d} R_{acbd} - \frac{1}{12} x^a x^b \Delta x^{\,c} \Delta x^{\,d} \Delta x^{\,e} \nabla_c R_{adbe} \\ - \frac{1}{6} x^a x^b x^c \Delta x^{\,d} \Delta x^{\,e} \nabla_a R_{bdce} + \mathcal{O}\left(\epsilon^4\right) \end{aligned}$$

10 Cadabra

No matter how determined one might be, the recursive nature of the preceding equations will wreak havoc with one's sanity should one dare to venture beyond the first few terms. For the good of all concerned it is far better to leave such calculations to computer programs such as Cadabra. In this section we discuss some particular issues we encountered when writing our Cadabra programs.

The examples given in the previous paper [1] actually arose from our earlier investigations of Cadabra as a tool to compute Riemann normal expansions. Thus it is no surprise that the techniques given in that paper are well suited to (most of) our current needs but with two exceptions. First, we need to develop Cadabra code for the truncation operator T_ϵ^m introduced in section (3.2). Second, we need to extend the ideas given in [1] to allow Cadabra to

compute the symmetrised derivatives of arbitrary tensors, such as $R^a{}_{b(cd;\underline{e})}$. We shall deal with these issues in the following sections after which we will present the final Cadabra generated expressions (to $\mathcal{O}(\epsilon^6)$ and in all their gory detail).

10.1 Truncation of polynomials

In section (3.2) we introduced a truncation operator T^m_ϵ. Here we discuss how we constructed that operator in our Cadabra code. It happens to be an extremely useful piece of code and is used extensively throughout our Cadabra code.

Suppose you are asked to extract the leading terms from an expression such as

$$P^a(x) = c^a + c^a_b x^b + c^a_{bc} x^b x^c + c^a_{bcd} x^b x^c x^d$$

A standard approach would be to compute the derivatives of $P^a(x)$ at $x = 0$. This approach would be rather simple to code in Cadabra. However a minor issue does pop up. Unless otherwise told, Cadabra will assume that all objects have non-zero derivatives. Thus if Cadabra were instructed to compute d/ds of the above expression it would dutifully do so but it would treat the x's and the coefficients $c^a_{\underline{b}}$ as possibly depending on s. Cadabra can be coaxed to restrict the derivative operators to act only on specific objects by using the `::Depends` property and the `@unwrap` algorithm.

A better solution, one that does not involve derivatives, is to use Cadabra's `::Weight` property and the `@keep_weight` algorithm. The idea is to assign weights to nominated objects (through the `::Weight` property) and then extract terms matching a chosen weight (using the `@keep_weight` algorithm). Here is a small piece of Cadabra code that does the job.

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

poly:=  c^{a}
      + c^{a}_{b} x^{b}
      + c^{a}_{b c} x^{b} x^{c}
      + c^{a}_{b c d} x^{b} x^{c} x^{d};

term00:=@(poly): @keep_weight!(term00){xterms}{0};
term01:=@(poly): @keep_weight!(term01){xterms}{1};
term02:=@(poly): @keep_weight!(term02){xterms}{2};
```

The first line identifies the x^a terms as our target (which we name as `xterms` so that they can be distinguished from other targets decalared by other instances of `::Weight`). The next three lines then extract the 0^{th}, 1^{st} and 2^{nd} terms in the expression `poly`. The result would be exactly as if we had written

```
term00:=c^{a};
term01:=c^{a}_{b} x^{b};
term02:=c^{a}_{b c} x^{b} x^{c};
```

The truncated polynomial (in this case a quadratic) could then be computed simply as `@(term00)+@(term01)+@(term02)`.

10.2 Symmetrised covariant derivatives

Let v_a be a tensor field and suppose we wish to compute $v_{a;b}$ at the origin of the Riemann normal frame, O. Construct any geodesic through P and an auxiliary field A^a throughout the patch containing O. Let the geodesic be parametrised by the proper distance s and described by $x^a = x^a(s)$ with unit tangent vector $D^a = dx^a/ds$. We choose A^a so that it is auto-parallel along the geodesic. Thus we have

$$\frac{dv_a}{ds} = v_{a,b}D^b$$

$$0 = \nabla_D\, D^a = \frac{dD^a}{ds} + \Gamma^a{}_{bc}D^bD^c$$

$$0 = \nabla_D\, A^a = \frac{dA^a}{ds} + \Gamma^a{}_{bc}A^bD^c$$

from which it follows that

$$v_{a;\underline{b}}A^aD^{\underline{b}} = \frac{d^n(v_aA^a)}{ds^n} \qquad \text{at } O$$

where $\underline{b}$ contains n indices. Thus any higher order covariant derivative can be obtained simply by expanding the right hand side, one derivative at a time, while using the parallel transport conditions listed above to eliminate derivatives in A^a and D^a. The Cadabra code for this is much the same. Each successive covariant derivative is obtained by applying d/ds to the previous result then using substitutions to eliminate the newly introduced derivatives of A^a and D^a. Comparing coefficients of $A^aD^{\underline{b}}$ across the equals sign will then reveal an expression for $v_{a(;\underline{b})}$, the fully symetrised covariant derivative of v_a at O.

Note that $0 = \Gamma^a{}_{(bc,\underline{d})}$ at O in Riemann normal coordinates and thus $0 = d^n D^a/ds^n$. This can be used to considerably simplify the above computations.

This idea can be easily extended to other cases. For example, suppose we require $v_{ab(c;de)}$ at O. In this case we would introduce two auxiliary fields, A^a and B^a, each constrained to be auto-parallel along the chosen geodesic. Then following the above procedure we obtain

$$v_{ab(c;de)}A^aB^bD^cD^dD^e = \frac{d^2(v_{abc}A^aB^bD^c)}{ds^2} \qquad \text{at } O$$

10.3 Symmetrised derivatives of the Riemann tensor

By inspection of equation (5.4) we see that the only derivatives of the Riemann tensor that enter our calculations are of the form $R^a{}_{(b\dot{c}d,\underline{e})}$ and $R^a{}_{(b\dot{c}d;\underline{e})}$. According to the prescription given above we can compute the covariant derivatives in terms of the partial derivatives by the following procedure.

$$0 = \nabla_D\, D^a = \frac{dD^a}{ds}$$

$$0 = \nabla_D\, B^a{}_b = \frac{dB^a{}_b}{ds} + \Gamma^a{}_{dc}B^d{}_bD^c - \Gamma^d{}_{bc}B^a{}_dD^c$$

$$R^a{}_{(b\dot{c}d;\underline{e})}B^c{}_aD^bD^dD^{\underline{e}} = \frac{d^n}{ds^n}\left(R^a{}_{bcd}B^c{}_aD^bD^d\right)$$

This is not exactly what we want – it yields the *covariant* derivatives in terms of the *partial* derivatives. We need instead, the *partial* derivatives expressed in terms of the *covariant* derivatives. In section (5) we provided one solution to this problem. There we argued that the equations could be re-written in the form

$$R^a{}_{(bc\dot{d},\underline{e})} = R^a{}_{(bc\dot{d};\underline{e})} - Q^a{}_{(bc\dot{d}\underline{e})} \tag{5.5}$$

where $Q^a{}_{(bc\dot{d}\underline{e})}$ contains all the lower order symmetrised partial derivatives of $R^a{}_{bcd}$. This algorithm certainly works but it is computationally expensive. If for the moment we label the equation for the n-th derivative as E_n then our algorithm entails a whole hierarchy of substitutions of E_j into E_k for $j < k$. For example, to compute E_4 we need to substitute E_1 into E_2, then E_1 and E_2 into E_3, and finally, E_1, E_2 and E_3 into E_4. Our code took about 4 seconds to compute the first four derivatives but with one extra derivative this time grew to 11 minutes. We did not bother to compute the sixth derivative (in fact for the results given here we only need the first three derivatives).

It seems reasonable to ask : is there a better way? Indeed there is and the changes required are very simple.

The auxiliary field $B^a{}_b$ can be freely chosen so there is nothing stopping us from setting $B^a{}_b$ to be constant throughout the neighbourhood of O. Thus every partial derivative of $B^a{}_b$ is zero at O. What use is this to us? Consider the equation

$$\left(R^a{}_{bcd}B^c{}_aD^bD^d\right)_{;\underline{e}}D^{\underline{e}} = \frac{d^n}{ds^n}\left(R^a{}_{bcd}B^c{}_aD^bD^d\right)$$

This is clearly true for any $B^a{}_b$ and $D^a = dx^a/ds$ (and in any coordinate frame). In our Riemann normal frame we have $0 = d^nD^a/ds^n$ and we have chosen $0 = B^a{}_{c,\underline{e}}$. Thus the right hand side can be reduced to $R^a{}_{bcd,\underline{e}}B^c{}_aD^bD^dD^{\underline{e}}$. This leads to the following modified scheme (after swapping the left and right hand sides)

$$0 = D^a{}_{;b}D^b = \frac{dD^a}{ds}$$

$$\nabla_D\, B^a{}_b = B^a{}_{b;c}D^c = \Gamma^a{}_{dc}B^d{}_bD^c - \Gamma^d{}_{bc}B^a{}_dD^c$$

$$R^a{}_{bcd,\underline{e}}B^c{}_aD^bD^dD^{\underline{e}} = \left(R^a{}_{bcd}B^c{}_a\right)_{;\underline{e}}D^bD^dD^{\underline{e}}$$

This algorithm computes all the partial derivatives directly, without requiring any substitutions from previous results. It took less than 10 seconds to compute the first five derivatives which is a dramatic improvement over our previous algorithm (11 minutes).

In our Cadabra code we compute all of the symmetrised partial derivatives (using the above algorithm) before we compute the metric expansion (equations (5.2, 5.3, 5.4)). In this way we obtain a series expansions for the metric in terms of the covariant derivatives of the curvatures.

11 Expansions to sixth order

All of our $\mathcal{O}(\epsilon^6)$ Cadabra programs were not overly demanding on computational resources, taking less than 2 minutes to run (on a MacOSX with an Intel cpu) and requiring less than 13 Mbyte of memory.

The Cadabra codes and several support scripts are available from the author's web site `http://users.monash.edu.au/~leo`.

Some of the following expansions can be compared directly with results obtained by traditional methods. In particular our expansions for the metric and its inverse, equations (11.1) and (11.2) agree exactly with those given in equations (17) and (18) respectively of [11]. Also, our equation (11.3) agrees with that given in equation (A12) of [20].

The metric

$$\begin{aligned}180 g_{ab}\left(x\right) = 180\, g_{ab} &- 60\, x^c x^d R_{acbd} - 30\, x^c x^d x^e \nabla_c R_{adbe} + 8\, x^c x^d x^e x^f R_{acgd} R_{begf} \\ &- 9\, x^c x^d x^e x^f \nabla_{cd} R_{aebf} + 4\, x^c x^d x^e x^f x^g R_{achd} \nabla_e R_{bfhg} \\ &+ 4\, x^c x^d x^e x^f x^g R_{bchd} \nabla_e R_{afhg} - 2\, x^c x^d x^e x^f x^g \nabla_{cde} R_{afbg} + \mathcal{O}\left(\epsilon^6\right)\end{aligned} \tag{11.1}$$

The inverse metric

$$\begin{aligned}180 g^{ab}\left(x\right) = 180\, g^{ab} &+ 60\, x^c x^d R^a{}_c{}^b{}_d + 30\, x^c x^d x^e \nabla_c R^a{}_d{}^b{}_e \\ &+ 12\, x^c x^d x^e x^f R^a{}_{cdg} R^b{}_{efg} + 9\, x^c x^d x^e x^f \nabla_{cd} R^a{}_e{}^b{}_f \\ &+ 6\, x^c x^d x^e x^f x^g R^a{}_{cdh} \nabla_e R^b{}_{fgh} + 6\, x^c x^d x^e x^f x^g R^b{}_{cdh} \nabla_e R^a{}_{fgh} \\ &+ 2\, x^c x^d x^e x^f x^g \nabla_{cde} R^a{}_f{}^b{}_g + \mathcal{O}\left(\epsilon^6\right)\end{aligned} \tag{11.2}$$

Generalised connections

$$\begin{aligned}180 \Gamma^a_{(bc)}(x) = {} & 120\, x^d R^a{}_{bdc} + 15\, x^d x^e \left(2\, \nabla_b R^a{}_{dec} + 4\, \nabla_d R^a{}_{bec} + \nabla^a R_{dbec}\right) \\ & + x^d x^e x^f \left(32\, R^a{}_{deg} R_{fbcg} - 16\, R^a{}_{bdg} R_{ecfg} - 8\, R^a{}_{dbg} R_{ecfg}\right. \\ & \quad \left. + 18\, \nabla_{db} R^a{}_{efc} + 18\, \nabla_{de} R^a{}_{bfc} - 8\, R^a{}_{gdb} R_{ecfg} + 9\, \nabla^a{}_d R_{ebfc}\right) \\ & + x^d x^e x^f x^g \left(16\, R_{dbch} \nabla_e R^a{}_{fgh} + 6\, R^a{}_{deh} \nabla_b R_{fcgh}\right. \\ & \quad + 16\, R^a{}_{deh} \nabla_f R_{gbch} - 8\, R^a{}_{bdh} \nabla_e R_{fcgh} - 4\, R^a{}_{dbh} \nabla_e R_{fcgh} \\ & \quad - 4\, R_{dbeh} \nabla_c R^a{}_{fgh} - 8\, R_{dbeh} \nabla_f R^a{}_{cgh} - 4\, R_{dbeh} \nabla_f R^a{}_{gch} \\ & \quad + 6\, \nabla_{deb} R^a{}_{fgc} + 4\, \nabla_{def} R^a{}_{bgc} - 5\, R^a{}_{deh} \nabla_h R_{fbgc} \\ & \quad - 4\, R^a{}_{hdb} \nabla_e R_{fcgh} - 4\, R_{dbeh} \nabla^a R_{fcgh} - 4\, R_{dbeh} \nabla_f R^a{}_{hgc} \\ & \quad \left. + 3\, \nabla^a{}_{de} R_{fbgc}\right) + \mathcal{O}\left(\epsilon^6\right)\end{aligned} \tag{11.3}$$

$$
\begin{aligned}
180\Gamma^{a}_{(bcd)}(x) = {} & 90\, x^{e}\nabla_{b}R^{a}{}_{ced} \\
& + 3\, x^{e}x^{f}\,(8\, R^{a}{}_{ebg}R_{fcdg} + 32\, R^{a}{}_{beg}R_{fcdg} - 8\, R^{a}{}_{bcg}R_{edfg} \\
& \quad + 18\,\nabla_{eb}R^{a}{}_{cfd} + 6\,\nabla_{bc}R^{a}{}_{efd} + 24\, R^{a}{}_{geb}R_{fcdg} + 3\,\nabla^{a}{}_{b}R_{ecfd}) \\
& + 10\, x^{e}x^{f}x^{g}\,(2\, R_{ebch}\nabla_{d}R^{a}{}_{fgh} + 2\, R_{ebch}\nabla_{h}R^{a}{}_{fgd} \\
& \qquad + 4\, R_{ebch}\nabla_{f}R^{a}{}_{dgh} + 4\, R_{ebch}\nabla_{f}R^{a}{}_{hgd} + 2\, R_{ebch}\nabla^{a}R_{fdgh} \\
& \qquad + 2\, R^{a}{}_{beh}\nabla_{c}R_{fdgh} + 4\, R^{a}{}_{beh}\nabla_{f}R_{gcdh} - R^{a}{}_{beh}\nabla_{h}R_{fcgd} \\
& + 2\, R^{a}{}_{heb}\nabla_{c}R_{fdgh} + 4\, R^{a}{}_{heb}\nabla_{f}R_{gcdh} - R^{a}{}_{heb}\nabla_{h}R_{fcgd}) + \mathcal{O}\left(\epsilon^{6}\right)
\end{aligned}
\tag{11.4}
$$

$$
\begin{aligned}
18\Gamma^{a}_{(bcde)}(x) = {} & 8\, x^{f}R^{a}{}_{bcg}R_{fdeg} \\
& + x^{f}x^{g}\,(2\, R^{a}{}_{bch}\nabla_{d}R_{fegh} + 4\, R^{a}{}_{bch}\nabla_{f}R_{gdeh} - R^{a}{}_{bch}\nabla_{h}R_{fdge} \\
& \quad + 2\, R_{fbch}\nabla_{d}R^{a}{}_{geh} + 10\, R_{fbch}\nabla_{d}R^{a}{}_{hge} + 4\, R_{fbch}\nabla_{g}R^{a}{}_{deh} \\
& \quad + 8\, R_{fbch}\nabla_{h}R^{a}{}_{dge} + 2\, R_{fbch}\nabla^{a}R_{gdeh} + 12\, R_{fbch}\nabla_{d}R^{a}{}_{egh} \\
& \quad + 6\, R^{a}{}_{bfh}\nabla_{c}R_{gdeh} + 6\, R^{a}{}_{hfb}\nabla_{c}R_{gdeh}) + \mathcal{O}\left(\epsilon^{6}\right)
\end{aligned}
\tag{11.5}
$$

$$
3\Gamma^{a}_{(bcdef)}(x) = x^{g}\,(2\, R^{a}{}_{bch}\nabla_{d}R_{gefh} + 3\, R_{gbch}\nabla_{d}R^{a}{}_{efh}) + \mathcal{O}\left(\epsilon^{6}\right) \tag{11.6}
$$

$$
\Gamma^{a}_{(bcdefg)}(x) = \mathcal{O}\left(\epsilon^{6}\right) \tag{11.7}
$$

Partial derivatives of the Riemann curvature tensor.

$$
R^{u}{}_{(bc\dot{v},a)} = \nabla_{a}R^{u}{}_{bcv} \tag{11.8}
$$

$$
R^{u}{}_{(cd\dot{v},ab)} = \nabla_{ab}R^{u}{}_{cdv} \tag{11.9}
$$

$$
2R^{u}{}_{(de\dot{v},abc)} = 2\,\nabla_{abc}R^{u}{}_{dev} - R_{vabf}\nabla_{c}R^{u}{}_{def} + R^{u}{}_{abf}\nabla_{c}R_{vdef} \tag{11.10}
$$

$$
\begin{aligned}
5R^{u}{}_{(ef\dot{v},abcd)} = {} & 5\,\nabla_{abcd}R^{u}{}_{efv} - 7\, R_{vabg}\nabla_{cd}R^{u}{}_{efg} + 7\, R^{u}{}_{abg}\nabla_{cd}R_{vefg}
\end{aligned}
\tag{11.11}
$$

Riemann normal coordinates

$$y^a = \Delta x^a + y^a_{bc}\Delta x^b\Delta x^c + y^a_{bcd}\Delta x^b\Delta x^c\Delta x^d + y^a_{bcde}\Delta x^b\Delta x^c\Delta x^d\Delta x^e \qquad (11.12)$$
$$+ y^a_{bcdef}\Delta x^b\Delta x^c\Delta x^d\Delta x^e\Delta x^f + \mathcal{O}\left(\epsilon^6\right)$$

$$2y^a_{bc} = \Gamma^a{}_{bc} \qquad (11.13)$$

$$6y^a_{bcd} = \Gamma^a{}_{be}\Gamma^e{}_{cd} + \partial_b\Gamma^a{}_{cd} \qquad (11.14)$$

$$24y^a_{bcde} = 2\,\Gamma^a{}_{bf}\partial_c\Gamma^f{}_{de} + \Gamma^a{}_{fg}\Gamma^f{}_{bc}\Gamma^g{}_{de} + \Gamma^f{}_{bc}\partial_f\Gamma^a{}_{de} + \partial_{bc}\Gamma^a{}_{de} \qquad (11.15)$$

$$\begin{aligned}360y^a_{bcdef} = {}& -4\,\Gamma^a{}_{bg}\Gamma^g{}_{ch}\Gamma^h{}_{di}\Gamma^i{}_{ef} + 2\,\Gamma^a{}_{bg}\Gamma^g{}_{ch}\partial_d\Gamma^h{}_{ef} + 3\,\Gamma^a{}_{bg}\Gamma^g{}_{hi}\Gamma^h{}_{cd}\Gamma^i{}_{ef}\\
&+ 6\,\Gamma^a{}_{bg}\Gamma^h{}_{cd}\partial_h\Gamma^g{}_{ef} - 6\,\Gamma^a{}_{bg}\Gamma^h{}_{cd}\partial_e\Gamma^g{}_{fh} + 9\,\Gamma^a{}_{bg}\partial_{cd}\Gamma^g{}_{ef}\\
&+ 4\,\Gamma^a{}_{gh}\Gamma^g{}_{bc}\Gamma^h{}_{di}\Gamma^i{}_{ef} + 13\,\Gamma^a{}_{gh}\Gamma^g{}_{bc}\partial_d\Gamma^h{}_{ef}\\
&+ \Gamma^g{}_{bc}\Gamma^h{}_{dg}\partial_h\Gamma^a{}_{ef} - 4\,\Gamma^g{}_{bc}\Gamma^h{}_{dg}\partial_e\Gamma^a{}_{fh} + 7\,\partial_g\Gamma^a{}_{bc}\partial_d\Gamma^g{}_{ef}\\
&+ 2\,\partial_b\Gamma^a{}_{cg}\partial_d\Gamma^g{}_{ef} + 3\,\Gamma^g{}_{bc}\Gamma^h{}_{de}\partial_g\Gamma^a{}_{fh} + 3\,\Gamma^g{}_{bc}\Gamma^h{}_{de}\partial_f\Gamma^a{}_{gh}\\
&+ 6\,\Gamma^g{}_{bc}\partial_{dg}\Gamma^a{}_{ef} - 3\,\Gamma^g{}_{bc}\partial_{de}\Gamma^a{}_{fg} + 3\,\partial_{bcd}\Gamma^a{}_{ef}\end{aligned} \qquad (11.16)$$

$$\begin{aligned}360y^a_{bc} = {}& 120\,x^dR^a{}_{bdc} + 15\,x^dx^e\left(2\,\nabla_bR^a{}_{dec} + 4\,\nabla_dR^a{}_{bec} + \nabla^aR_{dbec}\right)\\
&+ x^dx^ex^f\left(32\,R^a{}_{dge}R_{gbfc} - 16\,R^a{}_{bgd}R_{gefc} - 8\,R^a{}_{dgb}R_{gefc}\right.\\
&\left.+ 18\,\nabla_{db}R^a{}_{efc} + 18\,\nabla_{de}R^a{}_{bfc} + 8\,R^a{}_{gdb}R_{gefc} + 9\,\nabla^a{}_dR_{ebfc}\right)\\
&+ x^dx^ex^fx^g\left(16\,R_{hbdc}\nabla_eR^a{}_{fhg} + 6\,R^a{}_{dhe}\nabla_bR_{hfgc} + 16\,R^a{}_{dhe}\nabla_fR_{hbgc}\right.\\
&- 8\,R^a{}_{bhd}\nabla_eR_{hfgc} - 4\,R^a{}_{dhb}\nabla_eR_{hfgc} - 4\,R_{hdeb}\nabla_cR^a{}_{fhg}\\
&- 8\,R_{hdeb}\nabla_fR^a{}_{chg} - 4\,R_{hdeb}\nabla_fR^a{}_{ghc} + 6\,\nabla_{deb}R^a{}_{fgc} + 4\,\nabla_{def}R^a{}_{bgc}\\
&+ 5\,R^a{}_{dhe}\nabla_hR_{fbgc} + 4\,R^a{}_{hdb}\nabla_eR_{hfgc} - 4\,R_{hdeb}\nabla^aR_{hfgc}\\
&\left.+ 4\,R_{hdeb}\nabla_fR^a{}_{hgc} + 3\,\nabla^a{}_{de}R_{fbgc}\right)\end{aligned} \qquad (11.17)$$

$$\begin{aligned}1080y^a_{bcd} = {}& 90\,x^e\nabla_bR^a{}_{ced} + 3\,x^ex^f\left(8\,R^a{}_{egb}R_{gcfd} + 32\,R^a{}_{bge}R_{gcfd} - 8\,R^a{}_{bgc}R_{gefd}\right.\\
&\left.+ 18\,\nabla_{eb}R^a{}_{cfd} + 6\,\nabla_{bc}R^a{}_{efd} + 56\,R^a{}_{geb}R_{gcfd} + 3\,\nabla^a{}_bR_{ecfd}\right)\\
&+ 10\,x^ex^fx^g\left(2\,R_{hbec}\nabla_dR^a{}_{fhg} + 4\,R_{hbec}\nabla_hR^a{}_{fgd} + 4\,R_{hbec}\nabla_fR^a{}_{dhg}\right.\\
&+ 8\,R_{hbec}\nabla_fR^a{}_{hgd} - R_{hbec}\nabla^aR_{hfgd} + 2\,R^a{}_{bhe}\nabla_cR_{hfgd}\\
&+ 4\,R^a{}_{bhe}\nabla_fR_{hcgd} + R^a{}_{bhe}\nabla_hR_{fcgd} + 4\,R^a{}_{heb}\nabla_cR_{hfgd}\\
&\left.+ 8\,R^a{}_{heb}\nabla_fR_{hcgd} + 2\,R^a{}_{heb}\nabla_hR_{fcgd}\right)\end{aligned} \qquad (11.18)$$

(11.18)

$$432y^a_{bcde} = 8\,x^f R^a{}_{bgc}R_{gdfe} + x^f x^g\,(2\,R^a{}_{bhc}\nabla_d R_{hfge} + 4\,R^a{}_{bhc}\nabla_f R_{hdge} + R^a{}_{bhc}\nabla_h R_{fdge} + 2\,R_{hbfc}\nabla_d R^a{}_{ghe} - 10\,R_{hbfc}\nabla_d R^a{}_{hge} + 4\,R_{hbfc}\nabla_g R^a{}_{dhe} + 28\,R_{hbfc}\nabla_h R^a{}_{dge} + 2\,R_{hbfc}\nabla^a R_{hdge} + 12\,R_{hbfc}\nabla_d R^a{}_{ehg} + 6\,R^a{}_{bhf}\nabla_c R_{hdge} + 18\,R^a{}_{hfb}\nabla_c R_{hdge})$$

(11.19)

$$360y^a_{bcdef} = x^g\,(2\,R^a{}_{bhc}\nabla_d R_{hegf} + 3\,R_{hbgc}\nabla_d R^a{}_{ehf}) \tag{11.20}$$

Geodesic arc-length

$$L^2_{PQ} = f_{ab}\Delta x^a\Delta x^b + f_{abc}\Delta x^a\Delta x^b\Delta x^c + f_{abcd}\Delta x^a\Delta x^b\Delta x^c\Delta x^d + f_{abcde}\Delta x^a\Delta x^b\Delta x^c\Delta x^d\Delta x^e + \mathcal{O}\left(\epsilon^6\right) \tag{11.21}$$

$$180f_{ab} = 180\,g_{ab} - 60\,x^c x^d R_{cadb} - 30\,x^c x^d x^e\nabla_c R_{daeb} + x^c x^d x^e x^f\,(8\,R_{gcda}R_{gefb} - 9\,\nabla_{cd}R_{eafb}) + 2\,x^c x^d x^e x^f x^g\,(4\,R_{hcda}\nabla_e R_{hfgb} - \nabla_{cde}R_{fagb})$$

(11.22)

$$180f_{abc} = -15\,x^d x^e\nabla_a R_{dbec} + x^d x^e x^f\,(8\,R_{gdea}R_{gbfc} - 9\,\nabla_{da}R_{ebfc}) + x^d x^e x^f x^g\,(4\,R_{hadb}\nabla_e R_{hfgc} + 4\,R_{hdea}\nabla_b R_{hfgc} + 4\,R_{hdea}\nabla_f R_{hbgc} - 3\,\nabla_{dea}R_{fbgc})$$

(11.23)

$$540f_{abcd} = -3\,x^e x^f\,(44\,R_{gaeb}R_{gcfd} + 3\,\nabla_{ab}R_{ecfd}) - 5\,x^e x^f x^g\,(8\,R_{haeb}\nabla_c R_{hfgd} + 9\,R_{haeb}\nabla_h R_{fcgd} + 20\,R_{haeb}\nabla_f R_{hcgd} - 6\,R_{hefa}\nabla_b R_{hcgd})$$

(11.24)

$$54f_{abcde} = x^f x^g R_{hafb}\nabla_c R_{hdge} \tag{11.25}$$

12 Discussion

The value of any new computational tool comes not just in being able to do routine computations, computations that we could do by hand, but rather in giving us the option to perform computations we would not otherwise undertake. New tools should open new opportunities for research. Cadabra seems to be such a tool.

13 Source

A .tar.gz archive of the Cadabra files used in preparing this paper can be found at this URL http://users.monash.edu.au/~leo/research/papers/files/lcb09-03.html

14 Acknowledgements

I am very grateful to Kasper Peeters for his many helpful suggestions. Any errors, omissions or inaccuracies in regard to Cadabra are entirely my fault.

Appendix A. Symmetrisation of tensors

The totally symmetric part of a tensor $A_{i_1i_2i_3\cdots i_n}$ is commonly defined by

$$A_{(i_1i_2i_3\cdots i_n)} = \frac{1}{n!}\left(A_{i_1i_2i_3\cdots i_n} + A_{i_1i_2i_3\cdots i_n} + A_{i_1i_2i_3\cdots i_n} + \cdots\right)$$

where the sum on the right hand side includes every permutation of the indices of $i_1i_2i_3\cdots i_n$. If the tensor $A_{i_1i_2i_3\cdots i_n}$ happens to be symmetric in every pair of indices then we observe

$$A_{(i_1i_2i_3\cdots i_n)} = A_{i_1i_2i_3\cdots i_n}$$

From the above definition it is very easy to establish the following theorems

$$A_{(i_1i_2i_3\cdots(j_1j_2j_3\cdots j_m)\cdots i_n)} = A_{(i_1i_2i_3\cdots j_1j_2j_3\cdots j_m\cdots i_n)}$$

$$A_{(i_1i_2i_3\cdots i_n}B_{j_1j_2j_3\cdots j_m)} = A_{((i_1i_2i_3\cdots i_n)B_{(j_1j_2j_3\cdots j_m))}}$$

$$nA_{(i_1i_2i_3i_4\cdots i_n)} = A_{i_1(i_2i_3i_4\cdots i_n)} + A_{i_2(i_1i_3i_4\cdots i_n)} + A_{i_3(i_1i_2i_4\cdots i_n)} + \cdots + A_{i_n(i_1i_2i_3\cdots i_{n-1})}$$

$$nA_{(i_1i_2i_3i_4\cdots i_n)} = A_{(i_2i_3i_4\cdots i_n)i_1} + A_{(i_1i_3i_4\cdots i_n)i_2} + A_{(i_1i_2i_4\cdots i_n)i_3} + \cdots + A_{(i_1i_2i_3\cdots i_{n-1})i_n}$$

Suppose now that we have $A_{i_1i_2i_3\cdots i_n} = A_{(i_1i_2i_3\cdots i_n)}$, that is, $A_{i_1i_2i_3\cdots i_n}$ is totally symmetric. Then for any B_j we have

$$(n+1)A_{(i_1i_2i_3\cdots i_n}B_{j)} = A_{ji_2i_3\cdots i_n}B_{i_1} + A_{i_1ji_3\cdots i_n}B_{i_2} + A_{i_1i_2j\cdots i_n}B_{i_3} + \cdots + A_{i_1i_2i_3\cdots i_{n-1}j}B_{i_n}$$

and

$$(n+1)A_{(i_1i_2i_3\cdots i_n,j)} = A_{ji_2i_3\cdots i_n,i_1} + A_{i_1ji_3\cdots i_n,i_2} + A_{i_1i_2j\cdots i_n,i_3} + \cdots + A_{i_1i_2i_3\cdots i_{n-1}j,i_n} + A_{i_1i_2i_3\cdots i_n,j}$$

All of the above are very easy to prove but one result which requires just a little more thought is the following.

Suppose $A_{i_1i_2j_3j_4j_5\cdots j_n}$ is symmetric in the pair i_1i_2 and symmetric in all the indices $j_3j_4j_5\cdots j_n$. That is, it is symmetric under the interchange of any pair of i's and any pair of j's but it is *not* necessarily symmetric when any i is swapped with any j. What can we say about $A_{(i_1i_2j_3j_4j_5\cdots j_n)}$? Here is the result

$$nA_{(i_1i_2i_3\cdots i_n)} = 2A_{i_n(i_1i_2i_3\cdots i_{n-1})} + (n-2)A_{(i_1i_2i_3\cdots i_{n-1})i_n} \tag{A.1}$$

The proof is very easy. Begin by writing out $n!\,A_{(i_1i_2i_3\cdots i_n)}$ in full. Then partition the terms into two disjoint sets, one set in which i_n appears in one of the first two index slots, the other set in which i_n appears in any of the remaining $n-2$ slots. The terms in the first set are exactly those that define $A_{i_n(i_1i_2i_3\cdots i_{n-1})}$ while those in the second set define $A_{(i_1i_2i_3\cdots i_{n-1})i_n}$. The above equation follows by simply counting the number of terms in each set

$(2(n-1)!$ and $(n-2)(n-1)!$ respectively) and the simple observation that $n!\,A_{(i_1i_2i_3\cdots i_n)}$ equals the sum of the terms from both sets.

Finally we note that if $Q = A_{i_1i_2i_3\cdots i_n}x^{i_1}x^{i_2}x^{i_3}\cdots x^{i_n}$ then we have

$$Q_{,i_1i_2i_3\cdots i_n} = n!\,A_{(i_1i_2i_3\cdots i_n)}$$

$$Q = A_{(i_1i_2i_3\cdots i_n)}x^{i_1}x^{i_2}x^{i_3}\cdots x^{i_n}$$

References

[1] L. Brewin, A brief introduction to Cadabra: a tool for tensor computations in General Relativity, arXiv:0903.2085. http://users.monash.edu.au/~leo/research/papers/files/lcb09-02.html.

[2] K. Peeters, Introducing Cadabra: a symbolic computer algebra system for field theory problems, arXiv:hep-th/0701238.

[3] K. Peeters, A field-theory motivated approach to symbolic computer algebra, arXiv:cs/0608005v2.

[4] K. Peeters, Cadabra: a field-theory motivated symbolic computer algebra system, *Computer Physics Communications* **176** (2007) no. 8, 550–558.

[5] K. Peeters. The Cadabra home page. http://www.aei.mpg.de/~peekas/cadabra/.

[6] L. Brewin, Long term stable integration of a maximally sliced Schwarzschild black hole using a smooth lattice method, *Classical and Quantum Gravity* **19** (2002) 429–455.

[7] L. Brewin and J. Kajtar, A Smooth Lattice construction of the Oppenheimer-Snyder spacetime, arXiv:0903.5367. http://users.monash.edu.au/~leo/research/papers/files/lcb09-05.html.

[8] A. P. Gentle, Regge calculus: a unique tool for numerical relativity, *Gen. Rel. Grav.* **34** (2002) 1701–1718, gr-qc/0408006.

[9] T. Regge, General Relativity without coordinates, *Il Nuovo Cimento* **XIX** (1961) no. 3, 558–571.

[10] R. Willimas and P. Tuckey, Regge calculus : A bibliography and a brief review, *Classical and Qunatum Gravity* **9** (1992) 1409–1422.

[11] A. Hatzinikitas, A note on Riemann normal coordinates, arXiv:hep-th/0001078v1.

[12] K. Higashijima, E. Itou, and M. Nitta, Normal Coordinates in Kahler Manifolds and the Background Field Method, *Prog.Theor.Phys.* **108** (2002) 185–202, arXiv:hep-th/0203081v3.

[13] Y. Yamashita, Computer calculation of tensors in riemann normal coordinates, *General Relativity and Gravitation* **16** (1984) no. 2, 99–110.

[14] U. Müller, C. Schubert, and A. E. van de Ven, A Closed Formula for the Riemann Normal Coordinate Expansion, *Gen.Rel.Grav* **31** (1999) 1759–1768, arXiv:gr-qc/9712092v2.

[15] S. Chern, W. Chen, and K. Lam, *Lectures on Differential Geometry.* World Scientific, Singapore, 2000.

[16] I. Chavel, *Riemannian Geometry. A modern introduction, 2nd ed.* Cambridge University Press, Cambridge., 2006.

[17] L. P. Eisenhart, *Riemannian Geometry.* Princeton University Press, Princeton, 1926.

[18] A. Gray, The volume of a small geodesic ball of a Riemannian manifold, *Michigan.Math.J.* **20** (1973) 329–344.

[19] T. Willmore, *Riemannian Geometry.* Oxford University Press, Oxford, 1996.

[20] E. Calzetta, S. Habib, and B. Hu, Quantum kinetic field theory in curved spacetime: Covariant Wigner function and Liouville-Vlasov equations, *Phys.Rev.D* **37** (1988) no. 10, 2901–2919.

[21] A. Petrov, *Einstein Spaces.* Pergamon Press, Oxford, 1969.

[22] A. I. Nesterov, Riemann normal coordinates, Fermi reference system and the geodesic deviation equation, *Classical and Quantum Gravity* **16** (1999) 465–477, arXiv:gr-qc/0010096v1.