Class.Q.Grav. Vol.15(1998) pp.2427–2449. 28-Feb-1998

An ADM 3+1 formulation for Smooth Lattice General Relativity

Leo Brewin

Department of Mathematics
Monash University
Clayton, Vic. 3168
Australia

Abstract

A new hybrid algorithm for numerical relativity will be presented. We will employ a lattice in which the metric is recorded as pure scalar quantities such as the geodesic leg lengths. The dynamics of the lattice will be governed by the standard ADM 3+1 equations which will be given as a set of second order ordinary differential equations for the scalar data of the lattice. The only approximation introduced in this algorithm is in the approximation of a smooth metric by a lattice. No approximations are introduced into the field equations nor in their application to the lattice. An example for the Kasner T^3 cosmology will be presented and the results will be shown to be in excellent agreement with the continuum solution.

1. Introduction

It is fair to say that some of the most impressive calculations in numerical relativity (such as Choptuik's discovery of critical phenomena in gravitational collapse) have arisen from the use of just one discretisation procedure, namely, finite differences. Like any numerical problem, one can not expect that finite differences will be the ideal choice for every class of problem in numerical relativity. It is thus wise to explore alternative procedures (some of which are discussed by Laguna in [1]).

In a recent paper [2] a new numerical algorithm was developed and successfully applied in the construction of time symmetric initial data for a Schwarzschild black hole. The method was based on three key ideas, i) the approximation of a smooth spacetime by a piecewise smooth metric on a lattice, ii) the use of short geodesic segments as the legs of the lattice and iii) the use of Riemann normal coordinates local to each vertex. This lattice approach was motivated by the following observations.

- ◇ General relativity allows for a wide variety of topologies in spacetime. Thus it is desirable to have a method whereby non-trivial topologies can be easily accommodated. Such is the case with a lattice, this being a simple combinatoric assembly of vertices and legs. This also has the feature of allowing for dynamical changes in the topology of the Cauchy surface (eg. during the merger of two black holes).

◇ The only physically interesting quantities in a spacetime are the scalars derived from the spacetime. Thus it seems sensible to work with such quantities at the outset rather than extracting them from the coordinate based data. In the lattice the metric consists of the geodesic leg lengths and the angles between these geodesic segments.

◇ As the Einstein equations are expressed in some coordinate form one needs a link between the scalar data (the leg lengths and angles) and the coordinate data (the Riemann and extrinsic curvatures). For this purpose, local Riemann normal coordinates are used. These coordinates are never singular (provided the computational cell is small relative to the curvature length scales). In the 4-dimensional setting, the Riemann normal coordinates constitute a freely falling frame of reference. These in turn are intimately related to the Equivalence Principle and thus their use has a nice physical appeal.

Our primary aim in this paper is to demonstrate a hybrid algorithm in which a 3-dimensional spacelike lattice can be evolved using the standard ADM vacuum 3+1 equations. It should be note that this is not an alternative to the ADM approach – it **is the ADM approach applied directly to the 3-dimensional lattice**.

The key challenges are i) to find a reliable method for computing the six Riemann and extrinsic curvatures and ii) establishing how one can impose the ADM evolution equations on the lattice. We will begin by reviewing and improving the smooth lattice algorithm given in [2] for the computation of the Riemann curvatures. In section (2) we will show how the extrinsic curvatures can be computed while in section (3) we will demonstrate how the ADM evolution equations can be imposed on the lattice. In this paper we will not present any workable algorithm for solving the ADM constraint equations on the lattice. In the final section (4) we shall apply our algorithm to the full evolution of a Kasner T^3 cosmology.

2. Smooth lattice kinematics

In this section our focus will be on the computation of the Riemann and extrinsic curvature tensors.

2.1. The Riemann curvature tensor

The algorithm for computing the Riemann curvatures on a lattice, as given in [2], can be summarised as follows. The lattice is constructed as a network of vertices and legs and it is fully specified by the connectivity matrix and the leg lengths. To each vertex there is an associated computational cell defined by some local sub-set of the lattice (eg. the legs and vertices of the tetrahedra attached to this vertex, though larger subsets could be considered). Riemann normal coordinates are employed in each computational cell leading to a metric of the form

$$g_{\mu\nu}(x) = g_{\mu\nu} - \frac{1}{3} R_{\mu\alpha\nu\beta}\, x^{\alpha} x^{\beta} + \mathcal{O}(\epsilon^3) \tag{2.1.1}$$

where $g_{\mu\nu}$ may be chosen to be $\mathrm{diag}(1,1,1)$ and ϵ is a typical length scale for the computational cell. This form of the metric is nothing other than a Taylor series expansion around $x^\mu = 0$ adapted to Riemann normal coordinates (in which $g_{\mu\nu,\alpha} = 0$ at $x^\mu = 0$).

The crucial assertion that the legs are geodesic segments of this metric leads directly to the following equation for the leg length L_{ij} between vertices (i) and (j)

$$L^2_{ij} = g_{\mu\nu} \Delta x^{\mu}_{ij} \Delta x^{\nu}_{ij} - \frac{1}{3} R_{\mu\alpha\nu\beta}\, x^{\mu}_{i} x^{\nu}_{i} x^{\alpha}_{j} x^{\beta}_{j} + \mathcal{O}(\epsilon^5) \tag{2.1.2}$$

where $\Delta x^{\mu}_{ij} = x^{\mu}_{j} - x^{\mu}_{i}$. In three dimensions there are just six algebraicly independent components of the Riemann tensor. This leads to the following equivalent equation for L^2_{ij}

$$L^2_{ij} = g_{\mu\nu} \Delta x^{\mu}_{ij} \Delta x^{\nu}_{ij} - \frac{1}{3} F_{\mu\nu} w^{\mu}_{ij} w^{\nu}_{ij} + \mathcal{O}(\epsilon^5) \tag{2.1.3}$$

where $F_{\mu\nu} = F_{\nu\mu}$ and

$$\begin{array}{ccc}
F_{11} = R_{xyxy} & F_{22} = R_{xzxz} & F_{33} = R_{yzyz} \\
F_{12} = R_{xyxz} & F_{13} = R_{xyyz} & F_{23} = R_{xzyz} \\
w^{1}_{ij} = x^{x}_{i} x^{y}_{j} - x^{y}_{i} x^{x}_{j} & w^{2}_{ij} = x^{x}_{i} x^{z}_{j} - x^{z}_{i} x^{x}_{j} & w^{3}_{ij} = x^{y}_{i} x^{z}_{j} - x^{z}_{i} x^{y}_{j}
\end{array}$$

For each computational cell there would be a small set of such equations which could, in principle, be solved for the Riemann tensor and any coordinates not fixed by gauge freedoms. In [2] these where referred to as the smooth lattice equations. The computations for any one cell are fully decoupled from those of any other cell. Thus this is a local problem and should be amenable to Newton-Raphson methods. It is at this point that the design of the lattice becomes crucial. It must be such that the equations in each cell yield unambiguous estimates for the Riemann tensor. If the system is singular then there will exist continuous families of solutions for the curvatures. This indicates that the lattice lacks sufficient information to tie down a unique estimate (or at worst a finite discrete set of estimates) for the curvatures. The solution then would be to add extra information to the lattice such as extra leg lengths.

Indeed, subsequent experiments have shown that many seemingly suitable lattices (such as a dodecahedron with one interior vertex, the origin of the RNC) are singular. Following these experiments a slight variation to the above algorithm has proved to be very successful. It entails the addition of extra information in the form of the scalar products of pairs of tangent vectors to the legs attached to the central vertex. This not only supplies the missing information but it also significantly simplifies the set of smooth lattice equations – it allows all of the x^μ_i to be computed ahead of the curvatures.

The modified algorithm works as follows. Consider one particular computational cell and let us denote its central vertex by the label 0 (we will continue with this notation through this paper). We will denote by m_{ij} the scalar product associated with the pair of legs (oi) and (oj). If the Riemann normal coordinates of the vertices (i) and (j) are known then the m_{ij} may be computed from

$$m_{ij} = g_{\mu\nu}\Delta x^\mu_{oi}\Delta x^\nu_{oj} \tag{2.1.4}$$

Notice that this equation does not contain any curvature terms. In our modified scheme we use these equations to compute the Riemann normal coordinates x^μ_i from the given m_{ij}. This may appear to be a non-trivial task, however, as shown in the Appendix, there is a very simple algorithm which is no more demanding than the evaluation of one square root per vertex. In this algorithm each vertex of the cell is visited in turn with each vertex requiring just two m_{ij}'s and one L_{oi} for the calculations. It should be noted that we do not employ an m_{ij} for every pair of legs, rather we employ the minimal number required to compute the Riemann normal coordinates (for n radial legs this requires $(2n-3)$ m_{ij}'s).

Note that m_{ij} is related to the angle α_{ij} subtended at the origin between the pair legs (oi) and (oj) via the equation $m_{ij} = L_{oi}L_{oj}\cos(\alpha_{ij})$.

Once all the x^μ_i have been computed then the six curvatures can be computed by solving the leg length equations (2.1.3) applied to the legs on the surface of the computational cell (the radial legs provide no information on the curvatures). Here we encounter a minor problem – the set of equations (2.1.3) for the curvatures are almost certainly overdetermined, there being more than six legs on the surface of the cell. This problem of overdetermined systems will occur frequently in the course of this paper. Though there are a number of ways to directly solve such systems (eg. least squares) we chose instead to select a reduced set of equations by taking suitable linear combinations. For example, for an appropriately chosen set of λ^k_{ij}, the linear combinations

$$0 = \sum_{ij} \lambda^k_{ij} \left(L^2_{ij} - g_{\mu\nu}\Delta x^\mu_{ij}\Delta x^\nu_{ij} + \frac{1}{3}F_{\mu\nu}w^\mu_{ij}w^\nu_{ij} \right) \tag{2.1.5}$$

will form six properly determined equations for the six curvature components. The choice of the λ^k_{ij} will depend only on the connectivity of the smooth lattice (an example will be given later in section (4)).

In this algorithm the basic data given on the lattice are the L_{ij} and m_{ij} from which we compute, by the above equations, the x^μ_i and the $R_{\mu\nu\alpha\beta}$. Similarly, given the time derivatives of the L_{ij} and m_{ij} one can compute the time derivatives of the x^μ_i and the $R_{\mu\nu\alpha\beta}$.

In the construction of our Riemann normal coordinates we have taken advantage of the coordinate freedoms to tie the origin of the RNC to the central vertex ie. $x^\mu_o = 0$. We are also free to choose the three coordinate axes to be orthogonal at the origin, $g_{\mu\nu} = \mathrm{diag}(1, 1, 1)$. All that remains is the freedom for rigid body rotation which we can account for by specifying three of the Riemann normal coordinates, such as $x^x_1 = x^y_1 = x^y_2 = 0$.

2.2. Lapse, shift and drift

In the familiar 3+1 formulation of general relativity a typical spacetime can be represented as a sequence of distinct 3-dimensional hypersurfaces of the spacetime. In this picture, each hypersurface records one instant in the evolution of the 3-metric. The information that one must specify to fully define the 4-metric on the spacetime are the 3-metric, the lapse function and the shift vectors on each hypersurface. In this section we shall adapt this picture to one in which each hypersurface is a smooth 3-dimensional lattice.

The general idea will be to allow the basic data for each lattice, namely the L_{ij} and m_{ij}, to be functions of time. On each hypersurface we will continue to employ Riemann normal coordinates and the above equations for computing the curvature components. Since the lattice data are now presumed to be functions of time, so too must the coordinates x^μ_i and the curvature components $R_{\mu\nu\alpha\beta}$. We will also impose our previously stated gauge conditions on each hypersurface, in particular that $g_{\mu\nu} = \mathrm{diag}(1, 1, 1)$ and that the origin of the RNC remains forever tied to the central vertex of the cell (ie. $x^\mu_o = 0$ for all time). One could choose a dynamical set of gauge conditions but to do so at this stage would be an unnecessary complication.

It is customary to use Latin indices to denote spatial components (eg. g_{ij} for the 3-metric) and Greek indices for spacetime quantities (eg. $g_{\mu\nu}$ for the 4-metric). As we would like to use Latin indices to denote vertices of the lattice we shall choose to use Greek indices for spatial components. Thus in the following $g_{\mu\nu}, R_{\mu\nu\alpha\beta}$ etc. will denote 3-tensors that live in each 3-dimensional hypersurface.

We will be making frequent reference to the current and future hypersurfaces and the various structures that define and link these hypersurfaces. We will use Σ_0 to represent the current hypersurface and $\Sigma_{\delta t}$ to represent the hypersurface just slightly to the future of Σ_0. The time coordinate t is constant on each hypersurface with $t = t_0$ on Σ_0 and $t = t_0 + \delta t$ on $\Sigma_{\delta t}$. The lapse function will be denoted by N and the shift vector by N^μ. Both the lapse function and shift vectors will be defined on the vertices of the lattice. Note that even though we are using Riemann normal coordinates on each hypersurface the 4-dimensional coordinates (t, x^μ) will not, in general, be in Riemann normal form.

Consider any 3-dimensional smooth lattice and focus attention on one vertex. Through this vertex one can construct three well defined curves – the timelike worldline of the vertex, the timelike geodesic tangent to the normal vector at the vertex and the timelike worldline of the observer with constant spatial coordinates. These three curves are distinct and though they share a common origin in Σ_0 they will intersect $\Sigma_{\delta t}$ at three distinct points (as depicted in Figure (1)). In the usual 3+1 picture only two of these curves, the normal geodesic and the observers worldline, enter into the discussion. In the smooth lattice we must also consider the freedom for vertices to drift relative to these two curves. For this reason we will introduce a new vector, the drift vector γ^μ. This vector measures the displacement of the vertex's worldline from that of the normal.

From Figure (1) we can clearly see that the shift and drift vectors at vertex (i) are related by

$$\frac{dx_i^\mu}{dt} = -\gamma_i^\mu + N_i^\mu \tag{2.2.1}$$

where $x_i^\mu(t)$ are the time dependent coordinates of vertex (i). The dx_i^μ/dt can be easily computed from (2.1.4) (see the Appendix for full details). Note that in our chosen gauge $x_i^\mu = dx_i^\mu/dt = 0$ at the origin of the RNC and thus the shift and drift vectors coincide at the central vertex.

One now has a choice as to which of the drift or shift vectors should be freely specified on the lattice. Since the drift vector has a strong geometric appeal we shall choose to freely specify it and to use the above equation (2.2.1) to compute the shift vector. A simple choice is to set the drift vector to be zero at the origin (ie. the vertex at the origin of the RNC in Σ_0 is evolved along the timelike normal). Though, of course, there may be occasions where such a choice is not appropriate (eg. in the case of gravitational collapse).

We will now introduce a second lattice which will help us later on in making the transition to the standard ADM 3+1 equations. Consider one hypersurface and its associated lattice. Consider now a second lattice coincident with the original lattice. We will refer to these as the primary and shadow lattices respectively. The shadow lattice will be chosen so such that its vertices are evolved along the normals to the hypersurface (ie. as if the drift vectors were set to zero everywhere). We do this because the ADM 3+1 equations take on a particularly simple form when expressed as time derivatives along the normals. The shadow lattice will only be employed for one time step. Upon completion of the time step the current shadow lattice will be discarded and a new shadow lattice created coinciding with the updated primary lattice. Each of these shadow lattices are introduced only as an aid in the exposition of our algorithm – they need never be created in any computer program.

Our current task is to see how we should specify the data on the shadow lattice so that it evolves along the normals.

On the primary lattice we have various quantities such as L_{ij}, m_{ij} and x^μ_i all of which we assume to be functions of time t. Their counterparts on the shadow lattice will be denoted by the addition of a dash (ie. L'_{ij}, x'^μ_i etc.). The dash on the x'^μ_i should not be confused with coordinate transformation – the x'^μ_i are just the x^μ coordinates of vertex (i'). On the initial hypersurface Σ_0 all of the corresponding dashed and un-dashed quantities are equal (ie. $L'_{ij} = L_{ij}, x'^\mu_i = x^\mu_i$) though they may drift apart between successive hypersurfaces.

By inspection of Figure (1) one can easily verify that, on Σ_0,

$$\begin{aligned} x'^\mu_i &= x^\mu_i \\ \dot{x}'^\mu_i &= \dot{x}^\mu_i - \gamma^\mu_i \\ \ddot{x}'^\mu_i &= \ddot{x}^\mu_i - \dot{\gamma}^\mu_i \end{aligned} \tag{2.2.2}$$

where each dot denotes d/dt.

Consider now equation (2.1.3) applied to both lattices,

$$\begin{aligned} L^2_{ij}(t) &= g_{\mu\nu}\Delta x^\mu_{ij}\Delta x^\nu_{ij} - \frac{1}{3}F_{\mu\nu}w^\mu_{ij}w^\nu_{ij} \\ L'^2_{ij}(t) &= g_{\mu\nu}\Delta x'^\mu_{ij}\Delta x'^\nu_{ij} - \frac{1}{3}F_{\mu\nu}w'^\mu_{ij}w'^\nu_{ij} \end{aligned}$$

where

$$w'^1_{ij} = x'^x_i x'^y_j - x'^y_i x'^x_j \qquad w'^2_{ij} = x'^x_i x'^z_j - x'^z_i x'^x_j \qquad w'^3_{ij} = x'^y_i x'^z_j - x'^z_i x'^y_j$$

All of the terms in the above equations should be considered to be functions of time (except the $g_{\mu\nu}$ which we choose to be diag$(1,1,1)$). Differentiating with respect to t and using the above equation for dx'^μ_i/dt we obtain on Σ_0

$$\frac{dL'^2_{ij}}{dt} = \frac{dL^2_{ij}}{dt} - g_{\mu\nu}\Delta\gamma^\mu_{ij}\Delta x^\nu_{ij} - \frac{2}{3}F_{\mu\nu}\, p^\mu_{ij}w^\nu_{ij} \tag{2.2.3}$$

$$\begin{aligned} \frac{d^2L'^2_{ij}}{dt^2} = \frac{d^2L^2_{ij}}{dt^2} &- 2g_{\mu\nu}\left(\Delta\dot{\gamma}^\mu_{ij}\Delta x^\nu_{ij} + 2\Delta\gamma^\mu_{ij}\Delta\dot{x}^\nu_{ij} - \Delta\gamma^\mu_{ij}\Delta\gamma^\nu_{ij}\right) \\ &- \frac{4}{3}\dot{F}_{\mu\nu}p^\mu_{ij}w^\nu_{ij} - \frac{2}{3}F_{\mu\nu}\left(q^\mu_{ij}w^\nu_{ij} + 2p^\mu_{ij}\dot{w}^\nu_{ij} + p^\mu_{ij}p^\nu_{ij}\right) \end{aligned} \tag{2.2.4}$$

where $\Delta\gamma^\mu_{ij} = \gamma^\mu_j - \gamma^\mu_i$ and

$$\begin{aligned}
p^1_{ij} =&\ -\gamma^x_i x^y_j + \gamma^y_i x^x_j - x^x_i \gamma^y_j + x^y_i \gamma^x_j \\
p^2_{ij} =&\ -\gamma^x_i x^z_j + \gamma^z_i x^x_j - x^x_i \gamma^z_j + x^z_i \gamma^x_j \\
p^3_{ij} =&\ -\gamma^y_i x^z_j + \gamma^z_i x^y_j - x^y_i \gamma^z_j + x^z_i \gamma^y_j \\
q^1_{ij} =&\ -\dot\gamma^x_i x^y_j + \dot\gamma^y_i x^x_j - x^x_i \dot\gamma^y_j + x^y_i \dot\gamma^x_j - 2(\gamma^x_i \dot x^y_j - \gamma^y_i \dot x^x_j + \dot x^x_i \gamma^y_j - \dot x^y_i \gamma^x_j + \gamma^x_i \gamma^y_j - \gamma^y_i \gamma^x_j) \\
q^2_{ij} =&\ -\dot\gamma^x_i x^z_j + \dot\gamma^z_i x^x_j - x^x_i \dot\gamma^z_j + x^z_i \dot\gamma^x_j - 2(\gamma^x_i \dot x^z_j - \gamma^z_i \dot x^x_j + \dot x^x_i \gamma^z_j - \dot x^z_i \gamma^x_j + \gamma^x_i \gamma^z_j - \gamma^z_i \gamma^x_j) \\
q^3_{ij} =&\ -\dot\gamma^y_i x^z_j + \dot\gamma^z_i x^y_j - x^y_i \dot\gamma^z_j + x^z_i \dot\gamma^y_j - 2(\gamma^y_i \dot x^z_j - \gamma^z_i \dot x^y_j + \dot x^y_i \gamma^z_j - \dot x^z_i \gamma^y_j + \gamma^y_i \gamma^z_j - \gamma^z_i \gamma^y_j)
\end{aligned}$$

The quantities p^μ_{ij} and q^μ_{ij} arise from an expansion of the w'^μ_{ij} on Σ_0, namely,

$$\begin{aligned}
w'^\mu_{ij} &= w^\mu_{ij} \\
\dot w'^\mu_{ij} &= \dot w^\mu_{ij} + p^\mu_{ij} \\
\ddot w'^\mu_{ij} &= \ddot w^\mu_{ij} + q^\mu_{ij}
\end{aligned}$$

The $\dot x^\mu_i$ and $\dot F_{\mu\nu}$ can be obtained by differentiation of (2.1.4) and (2.1.5) respectively.

A similar argument can be applied to the m_{ij}. Starting with

$$\begin{aligned}
m_{ij} &= g_{\mu\nu}\Delta x^\mu_{oi}\Delta x^\nu_{oj} \\
m'_{ij} &= g_{\mu\nu}\Delta x'^\mu_{oi}\Delta x'^\nu_{oj}
\end{aligned}$$

we obtain, by direct differentiation, that on Σ_0

$$\frac{dm'_{ij}}{dt} = \frac{dm_{ij}}{dt} - g_{\mu\nu}\left(\Delta\gamma^\mu_{oi}\Delta x^\nu_{oj} + \Delta\gamma^\mu_{oj}\Delta x^\nu_{oi}\right) \tag{2.2.5}$$

$$\frac{d^2m'_{ij}}{dt^2} = \frac{d^2m_{ij}}{dt^2} - g_{\mu\nu}\Big(\Delta\dot\gamma^\mu_{oi}\Delta x^\nu_{oj} + \Delta\dot\gamma^\mu_{oj}\Delta x^\nu_{oi} + 2\Delta\gamma^\mu_{oi}\Delta\dot x^\nu_{oj} + 2\Delta\gamma^\mu_{oj}\Delta\dot x^\nu_{oi} - 2\Delta\gamma^\mu_{oi}\Delta\gamma^\nu_{oj}\Big) \tag{2.2.6}$$

2.3. Extrinsic curvature

The dL'^2_{ij}/dt and the dm'_{ij}/dt measure rates of change along the normals to Σ_0. They should therefore be expressible in terms of the extrinsic curvatures $K_{\mu\nu}$.

Consider the shadow lattice and the coordinates x^μ. Let us introduce a second coordinate system x''^μ such that the x''^μ are constant along each worldline of the vertices of the shadow lattice. This clearly establishes a time dependent coordinate transformation between x^μ and

x''^μ. Note that on Σ_0 both coordinate systems are identical. Note also that on subsequent hypersurfaces the x''^μ need not be in Riemann normal coordinate form.

Since the worldlines of the shadow lattice are tangent to the normals to Σ_0 we must have

$$\frac{dg''_{\mu\nu}}{dt} = -2NK''_{\mu\nu}$$

In the x''^μ coordinates the $L'_{ij}(t)$ and $m'_{ij}(t)$ may be estimated from

$$L'^2_{ij}(t) = g''_{\mu\nu}(t)\Delta x''^\mu_{ij}\Delta x''^\nu_{ij} + \mathcal{O}(\epsilon^4) \tag{2.3.1}$$

$$m'_{ij}(t) = g''_{\mu\nu}(t)\Delta x''^\mu_{oi}\Delta x''^\nu_{oj} + \mathcal{O}(\epsilon^4) \tag{2.3.2}$$

where $g''_{\mu\nu}$ is evaluated on the geodesic tangent to the normal to Σ_0 at the origin of the RNC. Since the x''^μ need not be in Riemann normal coordinate form we can expect these estimates to be less accurate than their Riemann normal coordinate counterparts (eg. equations (2.1.3) and (2.1.4)). This is the price we pay for not using Riemann normal coordinates. Increased accuracy could be obtained for the $L'^2_{ij}(t)$ by evaluating the right hand side at the centre of the legs. This then forces us to explicitly account for variations of $K_{\mu\nu}$ throughout the cell. It is not clear how one might make a similar 'centred' approximation for the $m'_{ij}(t)$. Rather than trying to resolve these issues in this paper we shall settle on the simple estimates given above.

Differentiating the previous two equations with respect to time and using the above equation for $K''_{\mu\nu}$ we obtain, on Σ_0,

$$\frac{dL'^2_{ij}}{dt} = -2(NK_{\mu\nu})_o\Delta x^\mu_{ij}\Delta x^\nu_{ij} \tag{2.3.3}$$

$$\frac{dm'_{ij}}{dt} = -2(NK_{\mu\nu})_o\Delta x^\mu_{oi}\Delta x^\nu_{oj} \tag{2.3.4}$$

This overdetermined system of equations can be solved for the six $K_{\mu\nu}$ at the origin of the RNC either directly (eg. least squares) or by first forming suitable linear combinations, such as

$$0 = \sum_{ij}\eta^k_{ij}\left(\frac{dL'^2_{ij}}{dt} + 2(NK_{\mu\nu})_o\Delta x^\mu_{ij}\Delta x^\nu_{ij}\right) \tag{2.3.5}$$

$$0 = \sum_{ij}\rho^k_{ij}\left(\frac{dm'_{ij}}{dt} + 2(NK_{\mu\nu})_o\Delta x^\mu_{oi}\Delta x^\nu_{oj}\right) \tag{2.3.6}$$

for an appropriate set of weights η^k_{ij} and ρ^k_{ij}.

In a similar fashion we can obtain

$$\frac{d^2 L'^2_{ij}}{dt^2} = -2\left(\frac{d}{dt}(NK_{\mu\nu})\right)_o \Delta x^\mu_{ij}\Delta x^\nu_{ij} \tag{2.3.7}$$

$$\frac{d^2 m'_{ij}}{dt^2} = -2\left(\frac{d}{dt}(NK_{\mu\nu})\right)_o \Delta x^\mu_{oi}\Delta x^\nu_{oj} \tag{2.3.8}$$

These equations will be used later when imposing the Einstein equations on the lattice.

At this point we are in a position to calculate both the extrinsic and Riemann curvatures tensors, at the origin of the RNC, given just the basic lattice data, namely the L_{ij}, m_{ij}, their time derivatives and for any choice of lapse function and drift vector. However we have not discussed how $K_{\mu\nu|\alpha}$ and $N_{|\mu\nu}$ may be calculated. In essence this is a standard interpolation problem – given data on a set of nodes, estimate the 1st and 2nd partial derivatives of that data. This is a trivial problem whenthe data points are arranged on a uniformly spaced grid. Unfortunately, this is not likely to be the case in a generic lattice. We will present one possible method based on least squares (though other methods could be considered).

Consider first the calculation of $N_{|\mu\nu}$. Since N is a scalar function we have $N_{|\mu} = \partial N/\partial x^\mu$ which we can approximate at the centre of the leg (ij) by

$$\left(N_{|\mu}\right)_{ij} = \frac{N_i - N_j}{L^2_{ij}} g_{\mu\nu}\Delta x^\nu_{ij} \tag{2.3.9}$$

Such an expression can be formed at the centre of each leg of the computational cell. From this data we can form a linear approximation $\widetilde{N}_{|\mu}(x^\alpha)$ to $N_{|\mu}(x^\alpha)$

$$\widetilde{N}_{|\mu}(x^\alpha) = \widetilde{N}_{|\mu} + \widetilde{N}_{|\mu\nu}x^\nu \tag{2.3.10}$$

where the constants $\widetilde{N}_{|\mu}$ and $\widetilde{N}_{|\mu\nu} = \widetilde{N}_{|\nu\mu}$ are obtained by a least squares fit. We can take the $\widetilde{N}_{|\mu\nu}$ as estimates of $N_{|\mu\nu}$ at the origin of the RNC. Similarly, the $\widetilde{N}_{|\mu}$ could be used as an approximation to $N_{|\mu}$ at the origin of the RNC. The appropriate least squares sum is chosen to be

$$S(\widetilde{N}_{|\mu}, \widetilde{N}_{|\mu\nu}) = \sum_\mu \sum_{ij} \left(\left(N_{|\mu}\right)_{ij} - \widetilde{N}_{|\mu} - \frac{1}{2}\widetilde{N}_{|\mu\nu}(x^\nu_i + x^\nu_j)\right)^2$$

Note that the symmetry condition $\widetilde{N}_{|\mu\nu} = \widetilde{N}_{|\nu\mu}$ must be explicitly imposed in this sum.

Do we have enough data to compute the nine numbers $\widetilde{N}_{|\mu}$ and $\widetilde{N}_{|\mu\nu}$? The answer is yes – each leg gives us three samples for $N_{|\mu}$ and each computational cell is certain to have more than three legs.

The computation of the $K_{\mu\nu|\alpha}$ can be performed in a similar manner. The first thing to do is to compute the $K_{\mu\nu}$ at the origin of each RNC for each computational cell. Now consider one specific computational cell. The vertices on the surface of the cell will themselves be the origins of neighbouring cells. Each of these cells carries their own estimates of the $K_{\mu\nu}$ in their own coordinates. As there is a well defined coordinate transformation between these neighbouring frames it is possible to obtain estimates of the $K_{\mu\nu}$ on each of the vertices in our chosen cell and in the coordinates of that chosen cell (as indicated in Figure (2)). We can then form a linear approximation to each of the six $K_{\mu\nu}$ in the chosen cell. Thus we put

$$\widetilde{K}_{\mu\nu}(x^\alpha) = \widetilde{K}_{\mu\nu} + \widetilde{K}_{\mu\nu|\alpha} x^\alpha$$

with the constants $\widetilde{K}_{\mu\nu}$ and $\widetilde{K}_{\mu\nu|\alpha}$ obtained by another application of the least squares method. Finally we take the $\widetilde{K}_{\mu\nu|\alpha}$ as our estimate of $K_{\mu\nu|\alpha}$ at the origin of our chosen cell. Note that in this case we would solve six separate least squares problems, one for each of the six $\widetilde{K}_{\mu\nu}(x^\alpha)$. In each case we need to compute just four numbers, one $\widetilde{K}_{\mu\nu}$ and three $\widetilde{K}_{\mu\nu|\alpha}$. Once again we see that the least squares method is appropriate.

It must be emphasized that this method has not been tested. In the example presented in section (4) the $K_{\mu\nu|\alpha}$ and $N_{|\alpha\beta}$ are both identically zero.

3. Smooth lattice dynamics

In the standard ADM 3+1 formulation of vacuum General Relativity there are four constraint equations

$$0 = R + K^2 - K_{\mu\nu} K^{\mu\nu} \tag{3.1}$$

$$0 = K_{|\mu} - K^{\nu}_{\mu|\nu} \tag{3.2}$$

and six evolution equations

$$\frac{\partial g_{\mu\nu}}{\partial t} = -2NK_{\mu\nu} + N_{\mu|\nu} + N_{\nu|\mu}$$

$$\frac{\partial K_{\mu\nu}}{\partial t} = -N_{|\mu\nu} + N\left(R_{\mu\nu} + KK_{\mu\nu} - 2K_{\mu\alpha}K^{\alpha}_{\nu}\right) + N^\alpha K_{\mu\nu|\alpha} + K_{\mu\alpha}N^{\alpha}_{|\nu} + K_{\nu\alpha}N^{\alpha}_{|\mu}$$

where $K = K^\alpha_\alpha, R_{\mu\nu} = g^{\alpha\beta} R_{\alpha\mu\beta\nu}$ and $R = g^{\mu\nu} R_{\mu\nu}$.

Let us concentrate, for the moment, on the evolution equations. When cast as evolution equations along the timelike normals these equations take on a particularly simple form, namely,

$$\frac{dg_{\mu\nu}}{dt} = -2NK_{\mu\nu} \tag{3.3}$$

$$\frac{dK_{\mu\nu}}{dt} = -N_{|\mu\nu} + N\left(R_{\mu\nu} + KK_{\mu\nu} - 2K_{\mu\alpha}K^{\alpha}_{\nu}\right) \tag{3.4}$$

The easiest road to these equations is to simply put the shift vector to zero in the original equations. However, these equations can also be easily derived by direct calculation.

We can now use the ADM equations to obtain evolution equations for the smooth lattice. Upon eliminating the reduced data L'_{ij} and m'_{ij}, by combining equations (2.2.4, 2.2.6) and (2.3.7, 2.3.8), we obtain

$$\begin{aligned}\frac{d^2L^2_{ij}}{dt^2} = &-2\left(\frac{d}{dt}(NK_{\mu\nu})\right)_o \Delta x^\mu_{ij}\Delta x^\nu_{ij} \\ &+ 2g_{\mu\nu}\left(\Delta\dot\gamma^\mu_{ij}\Delta x^\nu_{ij} + 2\Delta\gamma^\mu_{ij}\Delta\dot x^\nu_{ij} - \Delta\gamma^\mu_{ij}\Delta\gamma^\nu_{ij}\right) \\ &+ \frac{4}{3}\dot F_{\mu\nu}p^\mu_{ij}w^\nu_{ij} + \frac{2}{3}F_{\mu\nu}\left(q^\mu_{ij}w^\nu_{ij} + 2p^\mu_{ij}\dot w^\nu_{ij} + p^\mu_{ij}p^\nu_{ij}\right)\end{aligned} \tag{3.5}$$

$$\begin{aligned}\frac{d^2m_{ij}}{dt^2} = &-2\left(\frac{d}{dt}(NK_{\mu\nu})\right)_o \Delta x^\mu_{oi}\Delta x^\nu_{oj} \\ &+ g_{\mu\nu}\left(\Delta\dot\gamma^\mu_{oi}\Delta x^\nu_{oj} + \Delta\dot\gamma^\mu_{oj}\Delta x^\nu_{oi} + 2\Delta\gamma^\mu_{oi}\Delta\dot x^\nu_{oj} + 2\Delta\gamma^\mu_{oj}\Delta\dot x^\nu_{oi} - 2\Delta\gamma^\mu_{oi}\Delta\gamma^\nu_{oj}\right)\end{aligned} \tag{3.6}$$

We will take as initial data on Σ_0 the $L_{ij}, m_{ij}, dL_{ij}/dt$ and dm_{ij}/dt and we will assume that this data is a valid solution of the constraint equations (though in this paper we will not propose any algorithm for obtaining such data). We will also assume that well defined rules are given for choosing the N and γ^μ on each vertex of the lattice for all time.

The set of equations (2.1.3–2.1.4, 2.2.3, 2.2.5, 2.3.3, 2.3.4, 3.4–3.6) can then be used (via a suitable numerical algorithm such as predictor-corrector, leapfrog or Runge-Kutta methods) to evolve this data.

The are a number of points that need to be made. As each leg will appear in one or more computational cells there is an ambiguity in how L_{ij} and should be updated. For example, in the cubic lattice used in the example in section (4), each leg is attached to the central vertices of exactly two different computational cells. In this case the L_{ij} could be updated as the average from both cells. Other options are possible and should be explored by direct numerical experimentation. Notice that there is no such ambiguity in updating the m_{ij} since these are defined at the origin of each computational cell.

The drift vectors are prescribed only at the central vertex of each RNC cell. Their values on the vertices on the surface of the cell must be imported from neighbouring cells. As each such vertex is the central vertex of a neighbouring cell this procedure will be well defined. One way to do this is as follows. Let the γ'^{μ} be the drift vector defined at the central vertex o' and suppose we seek its values γ^{μ} in the RNC frame of a neighbouring cell with central vertex o. At o' we can form scalar products of γ'^{μ} with three of the legs shared by the cells (there must be at least three legs since the pair of cells have a non-trivial overlap). These scalar products can then be inverted to yield the γ^{μ} in the coordinates of o. This procedure would then be applied for each of the remaining surface vertices.

Another important issue is to what extent does the use of least squares approximations effect the accuracy and stability of the evolution? The least squares approximations are bound to introduce some degree of smoothing in the estimates of $N_{|\mu\nu}, K_{\mu\nu|\alpha}$ etc. Does this wash out unimportant numerical noise or important short scale variations in the lattice data? Most likely both important and unimportant information is lost. The question is : does this effect the long term stability of the evolution? Again, this can be investigated by way of a number of numerical experiments on a series of successively refined lattices.

The discussion so far has dealt entirely with the evolution of the lattice. Some words are in order regarding the construction of initial data on the lattice. We have already seen how all of the quantities needed for the constraint equations can be extracted from the lattice. Thus we are able to evaluate the right hand side of the constraint equations. Should this be non-zero then clearly we do not have valid initial data and some corrections must be made. How this might be achieved is far from clear. Can an approach similar to York's conformal method be used? Part of that method is to propose a 3-metric of the form $g_{\mu\nu} = \phi^4 \widetilde{g}_{\mu\nu}$ where $\widetilde{g}_{\mu\nu}$ is a given 3-metric (commonly chosen to be flat) and ϕ is a conformal factor. With two metrics one has a choice as to which should be expressed in Riemann normal form. If we choose $\widetilde{g}$ then we can imagine also having a conformal lattice with leg lengths $\widetilde{L}_{ij}$. The L_{ij} of the physical lattice could then be estimated as $L_{ij} = \phi_i \phi_j \widetilde{L}_{ij}$ (as suggested by Piran and Williams [3] in the context of the Regge calculus). It is unclear at present how York's transverse traceless tensor can be represented on the conformal lattice. On the other hand if one chooses to express the physical metric $g_{\mu\nu}$ in Riemann normal form then what form does a flat $\widetilde{g}_{\mu\nu}$ take in these coordinates? We conclude that it may not be a simple task to apply York's method to a smooth lattice while simultaneously employing local Riemann normal coordinates.

In the example in section (4) we will take a totally different approach. There we shall explicitly choose the Riemann and extrinsic curvatures and then construct the lattice data consistent with this choice of curvatures.

Another important question concerns the order of the discretisation errors. Previous calculations (see [2]) showed that the smooth lattice method yielded $\mathcal{O}(\epsilon^2)$ accurate estimates for both the metric and the Riemann tensors. However, this is unlikely to be the case for generic lattices. The formal truncation error in the Taylor series expansion of the metric, equation (2.1.1), is $\mathcal{O}(\epsilon^3)$ and thus we can expect the discretisation error in the curvatures to be of $\mathcal{O}(\epsilon^1)$. This will couple to the smooth lattice equations leading to an expected error in the metric of $\mathcal{O}(\epsilon^1)$. This is less than optimal – we would prefer an $\mathcal{O}(\epsilon^2)$ error for generic lattices. The challenge then is to find ways in which second order accurate estimates for the curvatures on the lattice can be obtained. This also applies to other terms such as $N_{|\mu\nu}$ and $K_{\mu\nu|\alpha}$. We hope to report on this in a later paper.

3.1. Variations

There is one particular variation to our algorithm that deserves special mention.

It was noted in [2] that the Bianchi identities are a necessary condition for the Riemann curvatures to be derivable from a metric. It may seem from equations (2.1.1) that such conditions must be automatically met by any smooth lattice metric. It is true that within any single RNC the Bianchi identities will be satisfied. However, the question of whether or not the set of curvatures from across the whole lattice can be derived from a global metric is not so obvious. One could test the Bianchi identities by importing the Riemann tensors from neighbouring cells and using them to form finite difference estimates of the derivatives of the curvatures. Demanding that the Bianchi identities be satisfied imposes constraints on the leg lengths of local groups of cells. In our study of the Schwarzschild initial data [2] we found that imposing the Bianchi identities was essential in obtaining the correct initial data. We did so by imposing an integral form of the Bianchi identities alongside the smooth lattice equations.

An alternative approach is to include derivatives of the curvatures in the Riemann normal coordinate expansion of the metric, namely,

$$g_{\mu\nu}(x) = g_{\mu\nu} - \frac{1}{3}R_{\mu\alpha\nu\beta}x^\alpha x^\beta - \frac{1}{6}R_{\mu\alpha\nu\beta,\lambda}x^\alpha x^\beta x^\lambda + \mathcal{O}(\epsilon^4)$$

If we fit this form of the metric to the lattice, while observing the Bianchi identities, we will have produced estimates for the curvatures which are derivable from a global metric. In the process we will also obtain a second order accurate estimate for the curvatures for any smooth lattice. It is straightforward, using the same techniques as used in [4], to establish that the basic smooth lattice equations (2.1.2) must be replaced by

$$L^2_{ij} = g_{\mu\nu}(\overline{x}_{ij})\Delta x^\mu_{ij}\Delta x^\nu_{ij} + \mathcal{O}(\epsilon^6)$$

where $\overline{x}^\mu_{ij} = (x^\mu_i + x^\mu_j)/2$. From these one can proceed to develop equivalent versions of the remaining smooth lattice equations (2.1.3, 2.2.3, 2.2.4, 3.5, 3.6). We shall defer such a development to a later paper.

This variation was not tested in our numerical investigations.

4. An example : The Kasner cosmology

The Kasner metric

$$ds^2 = -(dt)^2 + f^2_x(t)(dx')^2 + f^2_y(t)(dy')^2 + f^2_z(t)(dz')^2$$

describes a homogeneous cosmology with a T^3 topology. This metric, for the particular choice $f_x(t) = t^{p_x}, f_y(t) = t^{p_y}, f_z(t) = t^{p_z}$, is a solution of the vacuum Einstein equations when

$$0 = 1 - (p_x + p_y + p_z) \tag{4.1}$$
$$0 = 1 - (p^2_x + p^2_y + p^2_z) \tag{4.2}$$

The second equation is equivalent to the Hamiltonian constraint while the first equation arises from both the dynamical equations and the Hamiltonian constraint.

A trivial transformation of the spatial coordinates

$$\begin{aligned} x &= f_x(t)(x' - x'_o) \\ y &= f_y(t)(y' - y'_o) \\ z &= f_z(t)(z' - z'_o) \end{aligned}$$

where $(x, y, z)'_o$ is any nominated point, leads to a 3-metric which, at the chosen point, is in Riemann normal form with $g_{ij} = \mathrm{diag}(1, 1, 1)$. This fact and the clear simplicity of the metric suggests that the 3+1 smooth lattice approach should yield accurate approximations to the Kasner metric.

We will try to avoid importing knowledge from the continuum solution in our attempts to construct the smooth lattice spacetime. We will proceed in the usual manner of making various anzats for the metric (and relaxing those anzats should a contradiction arise when solving the field equations).

We will make the following anzats

- ◇ The smooth lattice has a T^3 topology.
- ◇ On the initial hypersurface, the smooth lattice is homogeneous.

◇ On the initial hypersurface, all spatial derivatives vanish.

◇ On the initial hypersurface, the extrinsic curvature is diagonal.

Some of these anzats can be built directly into the lattice. For example, the T^3 topology can be imposed by employing a cubic lattice with opposite faces identified. Homogeneity can be achieved by demanding that each vertex is equivalent to every other vertex. For the cubic lattice this can be achieved by requiring each cell to look identical to any other cell. This forces the pattern by which the legs are connected as well as the L_{ij} and m_{ij} to be the same from cell to cell. This implementation of homogeneity (there are others) will be imposed only on the initial data. We know (from the continuum) that the lattice should remain homogeneous. Thus by observing the future free evolution we can test how well our smooth lattice algorithm is performing.

One important consequence of the third anzats is that $0 = R_{\mu\alpha\nu\beta}$ and $0 = K_{\mu\nu|\alpha}$ and thus the four constraint equations reduce to the one equation

$$0 = K_{xx}K_{yy} + K_{xx}K_{zz} + K_{yy}K_{zz} \tag{4.3}$$

We will return to this equation latter when we describe how the initial data were chosen. However, before we do so, some further details of the lattice must be presented.

The basic structure of a typical RNC cell is shown in Figures (3) and (4). This cell consists of the central vertex and the eight cubes attached to that central vertex minus the 8 corner vertices and their associated legs. The complete lattice looks like a cubic lattice with the addition of two diagonal legs on each 2-dimensional face (note that there are no body diagonals, though they could be included).

In passing, we note that this cell is well suited to generic spaces and that it can be easily used in an adaptive refinement scheme.

4.1. Initial data

The initial data for each cell consists of the 18 $L^2_{ij}, dL^2_{ij}/dt$ for each leg attached to the central vertex and all of the 33 $m_{ij}, dm_{ij}/dt$'s in the cell. The remaining $L^2_{ij}, dL^2_{ij}/dt$ are assigned their values from the radial legs of the neighbouring cells. The initial data for one cell were assigned as follows. First, a consistent solution of the one remaining constraint equation (4.3) was obtained by freely choosing K_{xx}, K_{yy} and then solving for K_{zz}. Next, coordinates were assigned to each vertex of the cell in the form

$$x^\mu_i = (\alpha\Delta x, \beta\Delta y, \gamma\Delta z)^\mu$$

where α, β and γ are chosen from the set $-1, 0, +1$ (and at least one of the α, β, γ is zero). This places the 18 vertices either directly on the coordinate axes or on the coordinate planes. The initial data for this cell were then computed from

$$\begin{aligned} L^2_{ij} &= g_{\mu\nu}\Delta x^\mu_{ij}\Delta x^\nu_{ij} \qquad & \frac{dL^2_{ij}}{dt} &= -2(NK_{\mu\nu})_o\Delta x^\mu_{ij}\Delta x^\nu_{ij} \\ m_{ij} &= g_{\mu\nu}\Delta x^\mu_{oi}\Delta x^\nu_{oj} \qquad & \frac{dm_{ij}}{dt} &= -2(NK_{\mu\nu})_o\Delta x^\mu_{oi}\Delta x^\nu_{oj} \end{aligned}$$

This data was then replicated throughout the remainder of the lattice.

The above process amounts to choosing the curvatures and then searching for a lattice which supports these curvatures. Fortunately, for this spacetime, this is rather easy to do. It is not clear whether or not this is a reasonable approach for other spacetimes.

4.2. Evolution

The various m_{ij}'s and the sequence in which the vertices were visited when computing the coordinates x^μ_i is given in Table 1. The 6 extrinsic and 6 Riemann curvatures were computed from two sets of equations, 6 in each, formed as linear combinations of the smooth lattice equations (2.1.5, 2.3.5, 2.3.6). The weights for these equations are listed in Table 2.

The following gauge conditions were imposed throughout the evolution,

- ◇ A unit lapse function, $N_i = 1$.
- ◇ A zero shift vector at the origin of each RNC, $0 = N^\mu_o$.
- ◇ A zero drift vector everywhere, $0 = \gamma^\mu_i$.
- ◇ Coordinate axes aligned such that $0 = m_{12} = m_{13} = m_{23}$.

The homogeneity anzats was not explicitly enforced during the evolution of the lattice as we allowed each leg to evolve independently of every other leg. We can thus check the stability and/or accuracy of our smooth lattice evolution by monitoring any deviations from homogeneity.

4.3. Results

In each time step we performed two sweeps of the lattice. In the first sweep we computed the 2nd derivatives for each radial leg of each cell. In the second sweep, each leg was updated by employing 2nd derivatives computed as the average of the values obtained in the first sweep. The m_{ij} were updated in the first sweep since they are not shared across cells. In the first sweep the 2nd derivatives were calculated only for the radial legs since that was sufficient to cover every leg of the lattice.

In all of our calculations we updated the data via a naive Euler integration $x \leftarrow x + \dot{x}\Delta t$. The time step was chosen to be $\Delta t = 0.01$ and we set $1 = \Delta x = \Delta y = \Delta z$ when constructing the initial data.

The simplest test of our method is to compare the metric of the lattice against that of the continuum. From the continuum metric we can easily establish that, in local Riemann normal coordinates,

$$
\begin{gathered}
R_{\mu\alpha\nu\beta} = 0 \\
K_{xy} = K_{xz} = K_{yz} = 0 \\
\frac{dx_i}{dt} = \frac{p_x x_i}{t} \qquad \frac{dy_i}{dt} = \frac{p_y y_i}{t} \qquad \frac{dz_i}{dt} = \frac{p_z z_i}{t} \\
K_{xx} = \frac{-p_x}{t} \qquad K_{yy} = \frac{-p_y}{t} \qquad K_{zz} = \frac{-p_z}{t} \\
L^2_{xx} = t^{2p_x}(\Delta x)^2 \qquad L^2_{yy} = t^{2p_y}(\Delta y)^2 \qquad L^2_{zz} = t^{2p_z}(\Delta z)^2 \\
\frac{dL^2_{xx}}{dt} = 2p_x\frac{L^2_{xx}}{t} \qquad \frac{dL^2_{yy}}{dt} = 2p_y\frac{L^2_{yy}}{t} \qquad \frac{dL^2_{zz}}{dt} = 2p_z\frac{L^2_{zz}}{t}
\end{gathered}
$$

If the parameters, p_x, p_y, p_z and t where known then the above quantities could be compared with their numerical counterparts obtained from the smooth lattice. However, on the smooth lattice we do not have immediate knowledge of these four parameters – they must be extracted from the lattice data (ie. the L^2_{ij}'s, m_{ij}'s and their derivatives). This may be done as follows. By assigning to L^2_{xx} and dL^2_{xx}/dt their corresponding values from the lattice one can use the above equations as a pair of simultaneous equations for p_x and t which can be easily solved via a Newton-Raphson method. In this manner we can obtain estimates for the p_x, p_y and p_z as well as three estimates for the time (which we reduced to a single estimate by forming an average).

In all of our results (except those in the Figure (11)) we formed an initial estimate of the Kasner parameters which we retained throughout the evolution while updating t by simply incrementing it by the known time step.

The results are shown in Figures (6–12). The errors in the Riemann curvatures Figure (8) and the off diagonal extrinsic curvatures Figure (10) are clearly zero to within machine precision while the errors in the remaining lattice data can be attributed solely to the use of an in-exact numerical time integration method. To demonstrate this, we have plotted in Figure (12) the errors in the lattice data at $t = 11$ obtained for various values of Δt. The curves are all very close to being straight lines (as expected of Euler) and, most importantly, each curve would appear, upon extrapolation to $\Delta t = 0$, to pass through the origin. Thus in the limit as $\Delta t \to 0$ there will be no error in the leg lengths nor in the curvatures. We conclude that there is no spatial discretisation error in this lattice method (for this particular spacetime and for this particular lattice).

In Figure (11) we have plotted the residuals of the Kasner equations (4.1–4.2) along with the residual of the Hamiltonian constraint (4.3). Though they are not close to zero, at this time step $\Delta t = 0.01$, they are stable and their values do vanish, throughout the evolution, as $\Delta t \rightarrow 0$ (as demonstrated in Figure (12)).

Another useful test of the method is to examine how far the lattice drifted from its initially homogeneous state. This we did by computing the fractional difference between pairs of similar quantities (ie. quantities which in the continuum are exactly equal, for example, the values of K_{xx} in two neighbouring RNC cells). In all cases we found the differences to be zero to within machine precision.

We also ran our code on a sequence of lattices, from a coarse lattice with only 3 distinct cells along each of the orthogonal axes (for a total of 27 vertices) through to a fine lattice with 10 cells along each axis (and a total of 1000 vertices). The T^3 cosmology was imposed simply by assigning the same vertex labels to vertices on opposite faces of the lattice. We found no significant variation between the evolutions of the fine and coarse lattices. In fact the data from a typical cell in the coarse lattice (which cell does not really matter as the lattice is homogeneous) agreed to within machine precision with the data of a typical cell of the fine lattice.

Both of these observations must in part be attributed to the symmetric way in which the lattice was evolved. Recall that in each time step every cell is processed in identical fashion thus it is not a surprise that the lattice remained homogeneous throughout its evolution. This may not have been the case had we chosen a different integration scheme. For example, we could have employed just one sweep of the lattice with immediate updating of the legs via $x \leftarrow x + \dot{x}\Delta t/2$. After one sweep each leg would have been visited twice, each with a time step of $\Delta t/2$. This is not a symmetric scheme and it is possible that the evolution may develop instabilities. Fortunately this is not a scheme which we would ever use so it is a moot point whether or not it would lead to instabilities. However, it does demonstrate the important role the updating scheme can play in the stability of the evolution. Other (reasonable) schemes should be tried though as yet we have not done so.

Though we have demonstrated that the smooth lattice gives results which are in excellent agreement with the continuum it must be noted that this is a simple example and that it does not touch upon some of the more delicate aspects of the method. For example, in section (2.3), we noted that the computation of the $K_{\mu\nu}$'s could be improved by evaluating the equations at the centre of the legs. However, in this example, the continuum metric is homogeneous and so we can expect the same $K_{\mu\nu}$ regardless of where we evaluate the equations on the leg. Furthermore, we were fortunate in that the lapse could be set equal to 1 everywhere leading to the exact estimate $N_{|\mu\nu} = 0$. In other cases, such as for a Schwarzschild spacetime with maximal slicing, there would be some (presumably small) errors in estimating $N_{|\mu\nu}$. What effect these errors might have on the evolution remains to be seen. A similar problem will arise when trying to compute the $K_{\mu\nu|\alpha}$ (though this will only be required when solving the initial data problem, by an as yet unspecified algorithm).

This same spacetime, vacuum T^3 Kasner, has also been studied by Gentle and Miller [5]. Their calculations differ significantly from ours in that they chose a cubic lattice with one diagonal per face and one body diagonal per cube, and they evolved the lattice using a form of the Regge calculus due to Sorkin [6]. In their paper they solved both the initial and evolution problems. There are two principle differences between their results and those presented here. Firstly, their lattice carried a quadratic discretisation error in contrast to the zero discretisation error in our lattice. Secondly, and more importantly, their solutions for the L_{ij}'s demonstrated some high frequency wiggles. For legs of length order L, the amplitude and wavelengths of the wiggles were observed to vary as $\mathcal{O}(L^2)$ and $\mathcal{O}(L)$ respectively. Neither the amplitude nor the wavelength varied with the the time step. The source of these wiggles is unknown. It may be an artifact of the Regge calculus or it may be tied to the Sorkin evolution scheme (this is a single sweep asymmetric integration method). Another possibility is that the seeds of the wiggles are introduced during the solution of the initial data problem. Joint work is currently under way to investigate this behaviour.

5. Discussion

There have been other attempts to adapt the ADM 3+1 equations to a lattice. In particular there are the works of Piran and Williams [3] and Friedmann and Jack [7]. In both cases the equations were presented in the context of the Regge calculus. This is another lattice method in which the metric is assumed to be flat inside each pair of adjacent tetrahedra. The principle difference between their approach and ours is in the way in which the ADM equations were imposed on the lattice. In Piran+Williams and Friedmann+Jack the equations of motion for the lattice were obtained from the ADM 3+1 action principle. They were unable to evaluate the action directly because in the Regge calculus one does not have direct access to quantities such as $R_{\mu\nu}, N_{|\mu\nu}$ etc. Instead one has pure scalar quantities such as the defect angle, the areas and volumes of simplices. Thus in developing their equations both pairs of authors needed to make many (reasonable) assumptions about how various terms in the standard ADM action could be translated into a Regge form. This is a somewhat ad hoc method and thus casts some doubt on the validity of the final equations. Indeed the equations given by Friedmann and Jack differ from those given by Piran and Williams (though they do agree for appropriate choices of lapse and shift).

In contrast our equations have been derived in a systematic manner. The use of Riemann normal coordinates has allowed us to extract in a very natural way all of the relevant coordinate data from the lattice data. This in turn has lead to a very natural adaptation of the ADM 3+1 equations to the lattice. Despite this there are still many variations that need to be explored (eg. are the estimates of the curvatures sensitive to choice of weights?, should the L_{ij} be updated by an arithmetic or geometric mean?, how well does the method perform with different integrators?). There is also the major question of how one can solve the initial value constraint equations. Despite these concerns, we believe that the basics of our algorithm are well founded and should survive in any subsequent lattice methods for numerical relativity.

However, the proof of the pudding is always in the eating. The method has been shown to give excellent results for the Kasner cosmology. This is encouraging but it must be noted that this example avoided many of the potential problems associated with this method. So it is imperative that the proposed method be put to further tests. We have already successfully constructed the initial data for a time symmetric initial data slice in the Schwarzschild spacetime [2]. We are currently applying our proposed method to evolve that data. The results will be reported in a later paper.

6. Appendix

The purpose of this appendix is to provide full details of how the Riemann normal coordinates of each vertex in a computational cell can be computed from the L^2_{ij} and m_{ij}.

Consider three radial legs of the RNC cell and suppose the x^μ_i are known for two of the vertices (eg. vertices 1 and 2 of Figure 5). Our aim is to compute the coordinates x^μ_3 of the third vertex by solving the following equations

$$\begin{aligned} m_{13} &= g_{\mu\nu} x^\mu_1 x^\nu_3 \\ m_{23} &= g_{\mu\nu} x^\mu_2 x^\nu_3 \\ L^2_{03} &= g_{\mu\nu} x^\mu_3 x^\nu_3 \end{aligned}$$

Fortunately this system is rather easy to solve. One can easily verify that the solution is given by

$$x^\mu_3 = P x^\mu_1 + Q x^\mu_2 + R x^\mu_\star$$

where

$$x^\mu_\star = g^{\mu\nu} e^{123}_{\nu\alpha\beta} x^\alpha_1 x^\beta_2$$

$$P = \frac{m_{13} L^2_{02} - m_{23} m_{12}}{L^2_\star} \qquad Q = \frac{m_{23} L^2_{01} - m_{13} m_{12}}{L^2_\star} \qquad L^2_\star = L^2_{01} L^2_{02} - m^2_{12}$$

$$R = \pm \frac{\left(L_{03}^2 - P^2 L_{01}^2 - Q^2 L_{02}^2 - 2PQm_{12}\right)^{1/2}}{L_\star}$$

The two solutions, one for each choice of the ± sign, correspond to the two possible locations of the third vertex, one on each side of the plane containing the three vertices 0,1 and 2. Which choice is taken will depend on the design of the lattice. A systematic choice can be made by noting that that the vectors x_1^μ, x_2^μ and $x_\star^\mu$ form a right handed system. With $R > 0$ the vector x_3^μ lives on the same side of the plane as $x_\star^\mu$.

The dx_3^μ/dt can be computed by direct differentiation of the above equations provided $R \neq 0$. This later case arises when the third vertex lies in the plane containing the three vertices $0, 1$ and 2 (ie. the tetrahedron formed by the 4 vertices has zero volume). This is an exceptional case which can always be avoided by a careful choice of the vertices 1 and 2.

All that remains is show how the coordinates of the first two vertices may be computed. This is rather straightforward as we can make strong use of our coordinate gauge freedoms. The central vertex can always be assigned the coordinates $x^\mu = 0$. For vertex 1 we can align the RNC axes so that $x^\mu = (0, 0, L_{01})^\mu$. Finally, by appropriate rotations, we can force vertex 2 to lie in the $x - z$ plane, ie. $x_2^\mu = (A, 0, B)^\mu$ for some pair of numbers $A > 0$ and B. The numbers A, B are found by solving

$$\begin{aligned} m_{12} &= g_{\mu\nu} x_1^\mu x_2^\nu \\ L_{02}^2 &= g_{\mu\nu} x_2^\mu x_2^\nu \end{aligned}$$

leading to $B = m_{12}/L_{01},\ A = (L_{02}^2 - B^2)^{1/2}$

7. Acknowledgements

An earlier version of this paper appeared as the preprint `gr-qc/9708039` on the Los Alamos preprint server.

8. References

[1] Laguna, P.
Numerical relativity without finite differences.
To appear in *Living Reviews in relativity.*

[2] Brewin, L.C.
Riemann normal coordinates, smooth lattices and numerical relativity.
Applied Mathematics Preprint. 97/3 also available at `gr-qc/9701057` and
`http://newton.maths.monash.edu.au:8000/preprints/slgr.ps.gz`
Submitted to *Class.Q.Grav.*

[4] Brewin, L.C.
Is the Regge Calculus a consistent approximation to General Relativity?
Applied Mathematics Preprint. 95/3 also available at `gr-qc/9502043` and
`http://newton.maths.monash.edu.au:8000/preprints/truncation.ps.gz`
Submitted to *Class.Q.Grav.*

[3] Piran, T. and Williams, R.M.
Three-plus-one formulation of Regge calculus.
Phys.Rev.D. Vol.33(1986) pp.1622-1633.

[7] Friedmann, J.L. and Jack, I.
3+1 Regge calculus with conserved momentum and Hamiltonian constraints.
J.Math.Phys. Vol.27(1986) pp.2973-2986.

[5] Gentle, A.P. and Miller, W.A.
A fully (3+1)-dimensiuonal Regge calculus model of the Kasner cosmology.
Class.Q.Grav. Vol.15(1998) pp.389-405.

[6] Barret, J.W., Galassi, M., Miller, W.A., Sorkin, R.D., Tuckey, P.A. and Williams, R.M.
A parallelizable implicit evolution scheme for Regge Calculus.
Int.J.Theor.Phys. Vol.36(1997) pp.815-839.

$0 : (\star,\star)$	$1 : (\star,\star)$	$2 : (1,\star)$	$3 : (1,2)$	$4 : (1,2)$
$5 : (1,4)$	$6 : (2,5)$	$7 : (1,2)$	$8 : (1,5)$	$9 : (3,1)$
$10 : (3,2)$	$11 : (1,3)$	$12 : (1,4)$	$13 : (3,4)$	$14 : (3,6)$
$15 : (2,3)$	$16 : (2,6)$	$17 : (2,1)$	$18 : (2,5)$	

Table 1. This table records the pairs of vertices employed in the computation of the x_i^μ for each vertex. An entry such as $i\ :\ (j,k)$ means that the coordinates for vertex i are computed from the data m_{ij}, m_{ik} and L_{oi}. The order of j and k in (j,k) is crucial – it has been chosen so that the vectors (oi) and $(oj) \times (ok)$ lie on the same side of the plane spanned by (oj) and (ok) (ie. the + root is always taken for R, see the appendix). The $\star$'s denote data that are set by guage conditions.

Equation	$\sum$ Weights(equation)
λ^1	$+(1,2)+(2,6)+(6,4)+(4,1)$
λ^2	$+(1,5)+(5,6)+(6,3)+(3,1)$
λ^3	$+(2,3)+(3,4)+(4,5)+(5,2)$
λ^4	$+(9,8)-(8,10)+(10,7)-(7,9)+(13,12)-(12,14)+(14,11)-(11,13)$
λ^5	$+(17,8)-(8,18)+(18,12)-(12,17)+(15,7)-(7,16)+(16,11)-(11,15)$
λ^6	$+(17,9)-(9,15)+(15,13)-(13,17)+(18,10)-(10,16)+(16,14)-(14,18)$
η^1	$+(0,2)+(0,4)$
η^2	$+(0,3)+(0,5)$
η^3	$+(0,1)+(0,6)$
ρ^4	$+(1,2)-(2,6)+(6,4)-(4,1)$
ρ^5	$+(1,5)-(5,6)+(6,3)-(3,1)$
ρ^6	$+(2,3)-(3,4)+(4,5)-(5,2)$

Table 2. This table records the weights and terms used to form the reduced smooth lattice equations (2.1.5, 2.3.5, 2.3.6). An entry such as $\tau(i,j)$ denotes one term in the sum. This term is based on the equation for leg (i,j) with weight $\tau = \pm 1$. The first six equations are used to compute the Riemann curvatures, while the second six are used for the extrinsic curvatures. The η entries correspond to equations (2.3.5) while the ρ entries correspond to equations (2.3.6). These equations, on this lattice, can be used to calculate the curvatures for any space, not just the Kasner space considered in the text.

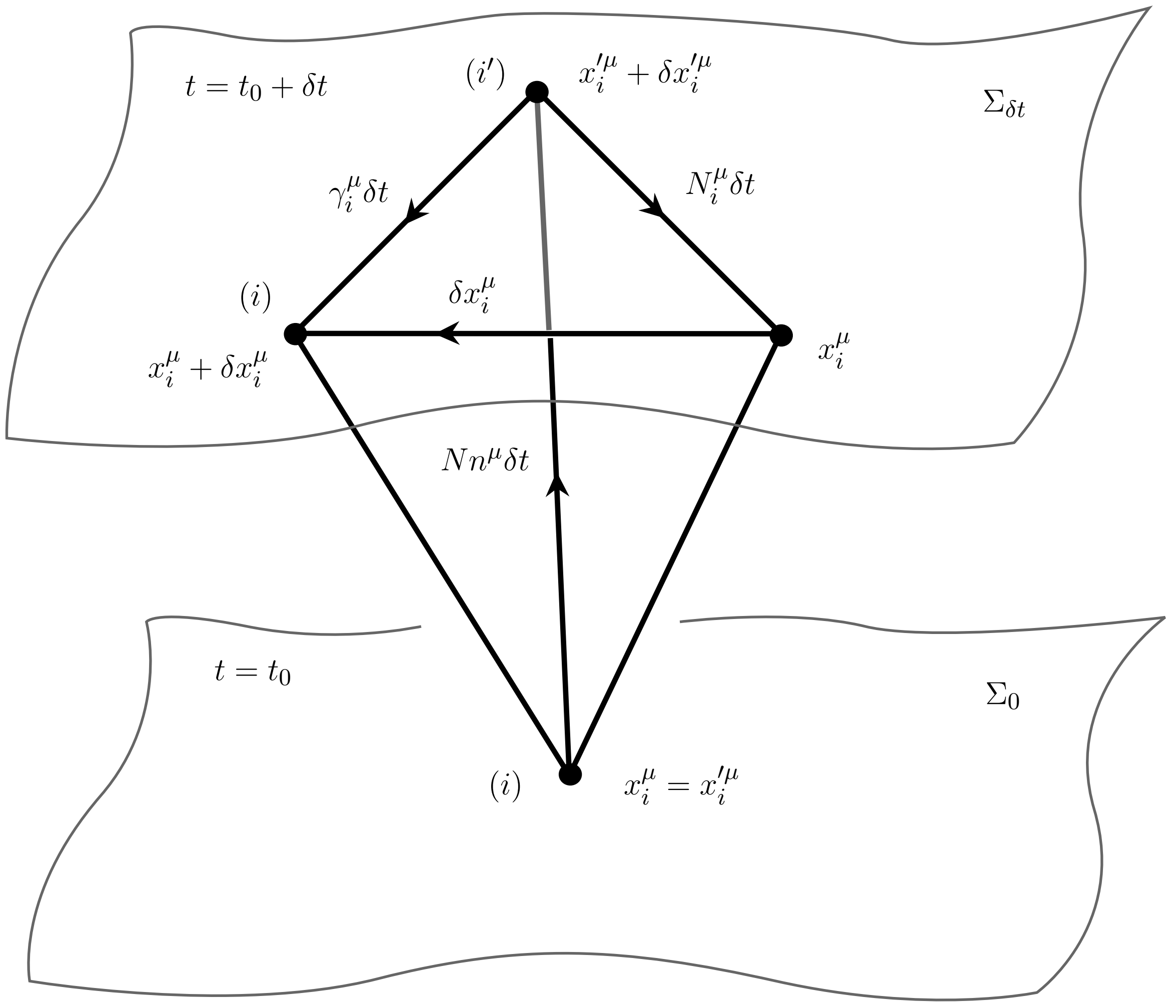

Figure 1. This figures displays the drift vector γ_i^μ and the relationships between the coordinates on the primary and shadow lattices. Clearly $\delta x_i^\mu = (-\gamma^\mu + N_i^\mu)\delta t$ leading directly to equation (2.2.1). We also have, by construction, that $x_i^\mu = x_i'^\mu$ on Σ_0. Thus $\delta x_i^\mu = \delta x_i'^\mu + \gamma_i^\mu \delta t$ which in turn leads to equation (2.2.2).

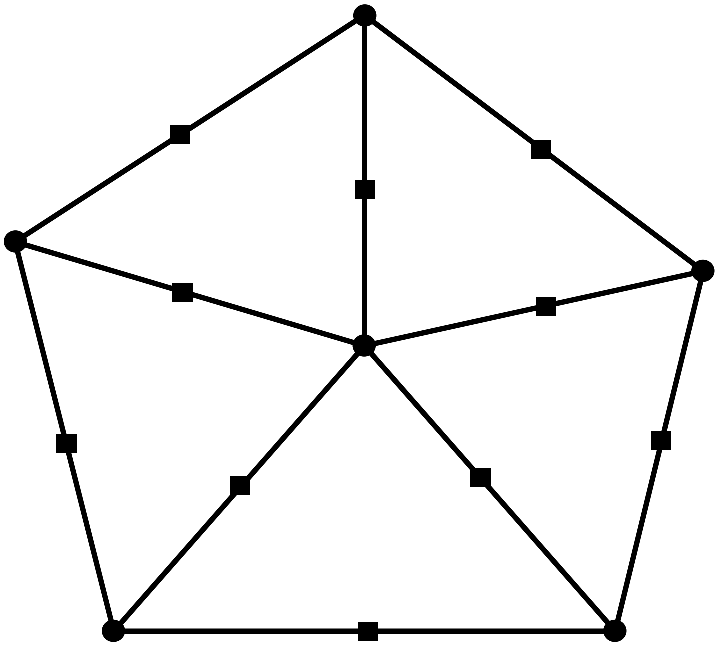

Figure 2. In the first pass over the lattice, $K_{\mu\nu}$ is computed at each vertex (●). In a second pass, $K_{\mu\nu|\alpha}$ at the central vertex is estimated by fitting the linear approximation $\widetilde{K}_{\mu\nu}(x) = \widetilde{K}_{\mu\nu} + \widetilde{K}_{\mu\nu|\alpha}x^\alpha$ to the data on each vertex. This will require a coordinate transformation from neighbouring cells to get data on the boundary of the cell. The $N_{|\mu\nu}$ at the central vertex can be estimated by first forming estimates for $N_{|\mu}$ at the centre of each leg (■). That data is then approximated by a linear function $\widetilde{N}_{|\mu}(x) = \widetilde{N}_{|\mu} + \widetilde{N}_{|\mu\nu}x^\nu$. We can then use $\widetilde{N}_{|\mu\nu}$ as an estimate of $N_{|\mu\nu}$ at the central vertex.

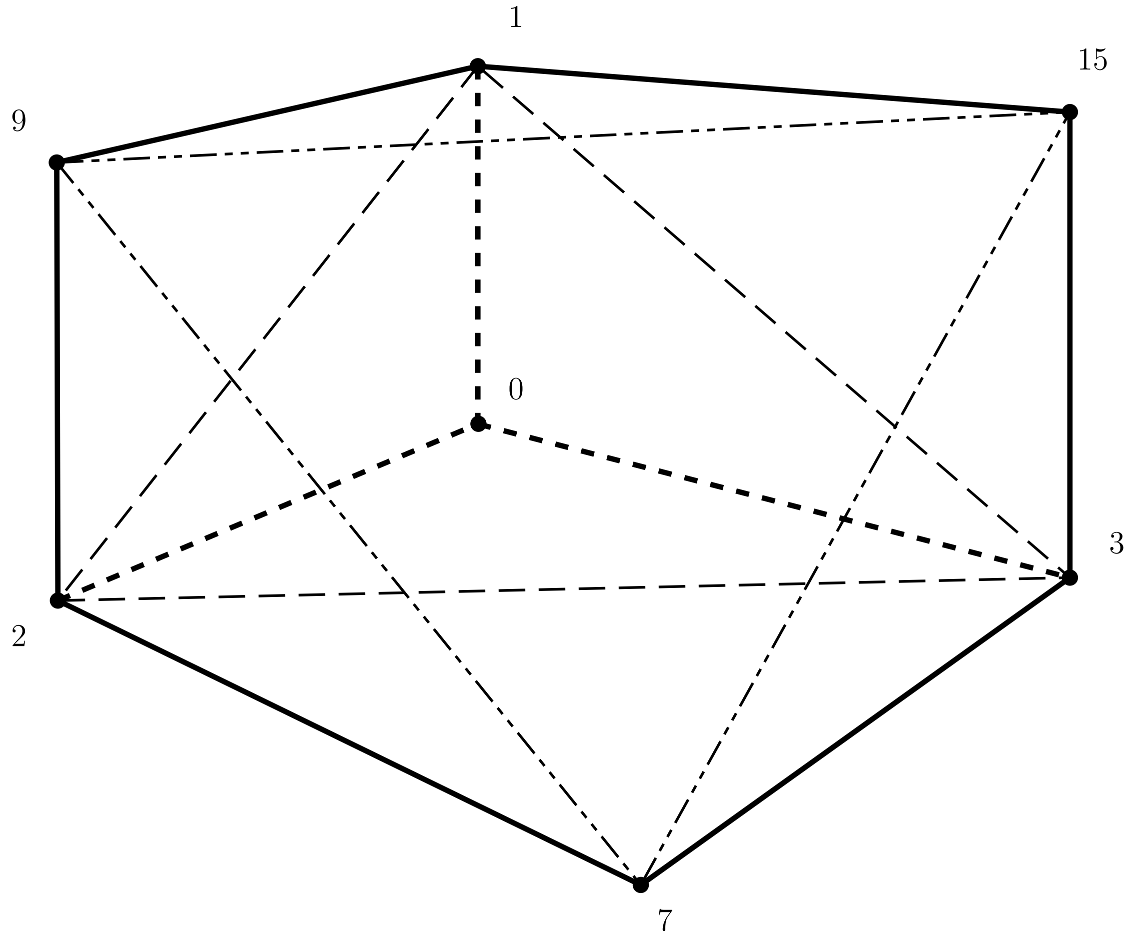

Figure 3. A typical octant of one computational cell of the cubic lattice. Vertex 0 is the central vertex. Each face should have two diagonals, however, for clarity we have suppressed one of the diagonals. The diagonals shown are those used in the computation of the Riemann curvatures. The vertex labels correspond exactly to those in Figure (4).

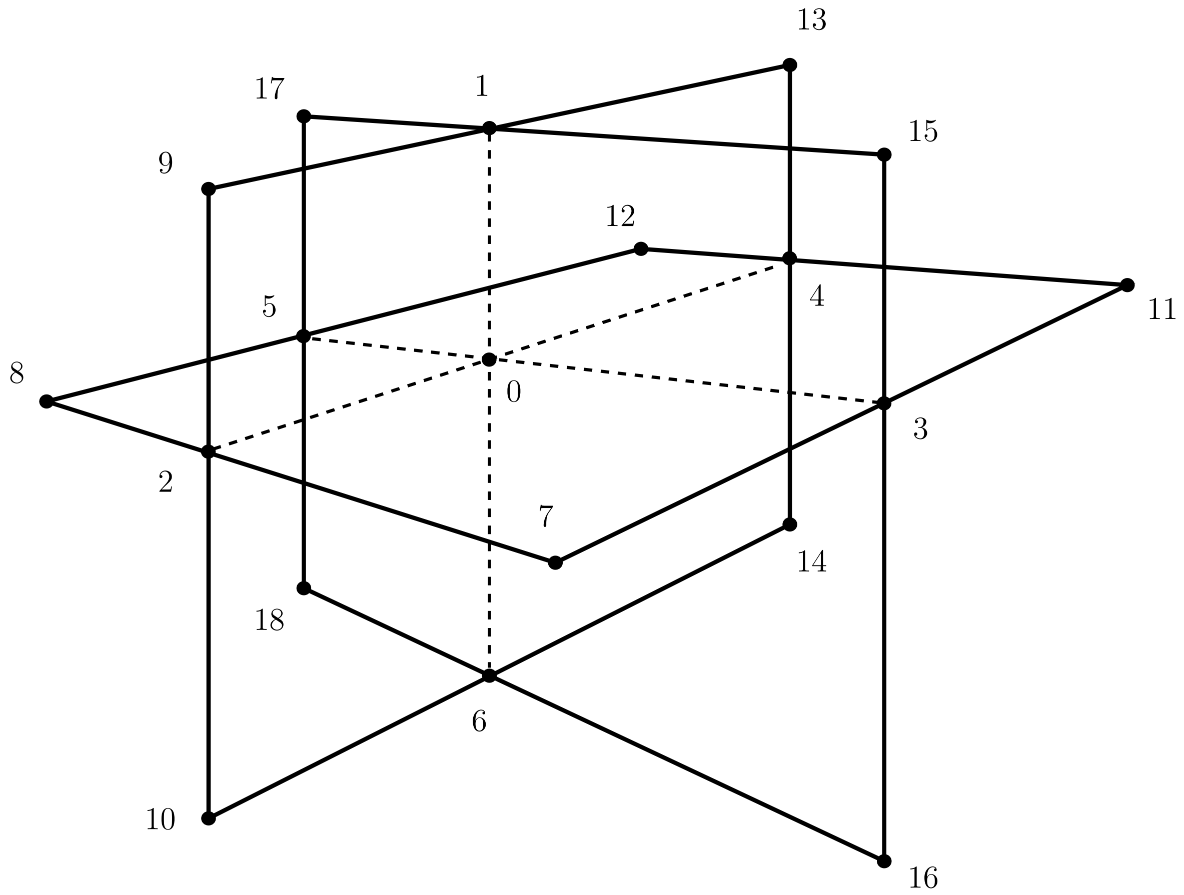

Figure 4. This figure displays the vertices and their labels for a typical RNC cell. The complete lattice can be generated by replicating this structure, and the associated diagonals, such as those in Figure (3), along the three coordinate axes. Note that this leads to RNC cells which overlap. Note also that each rectangle, such as (0192), should contain two diagonals but for clarity they have been excluded from this figure.

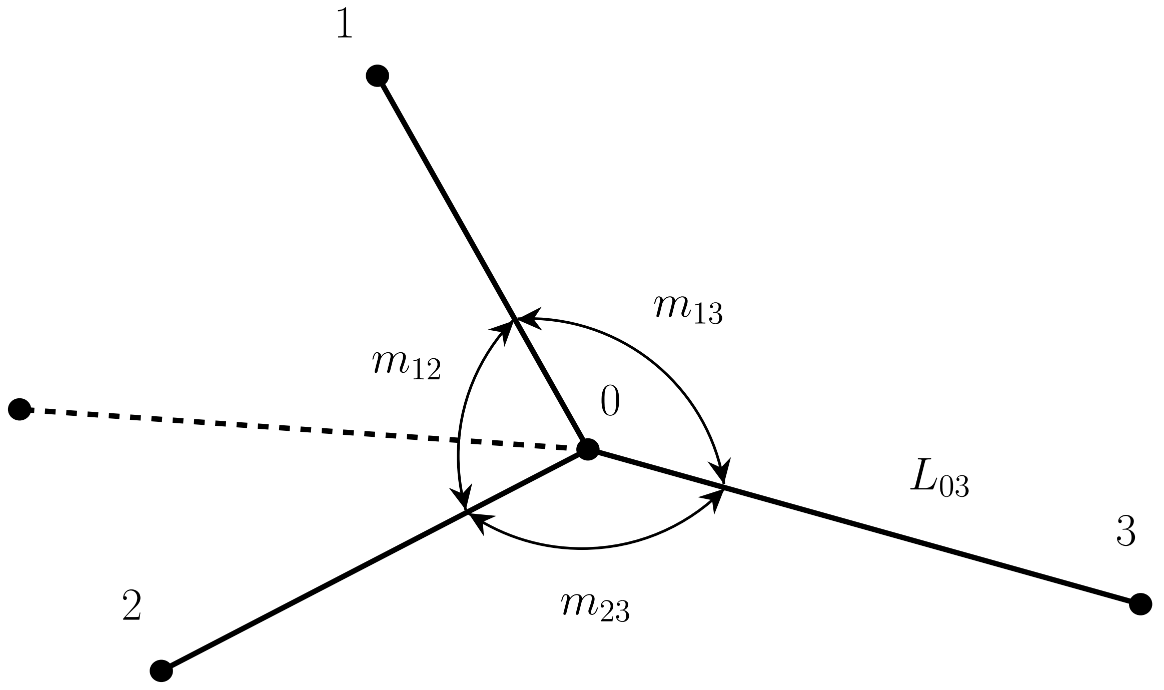

Figure 5. This figure displays the various data and vertices used in the computation of x_3^μ. The dashed line denotes the second solution arising from the negative root for R. The full details are given in the Appendix.

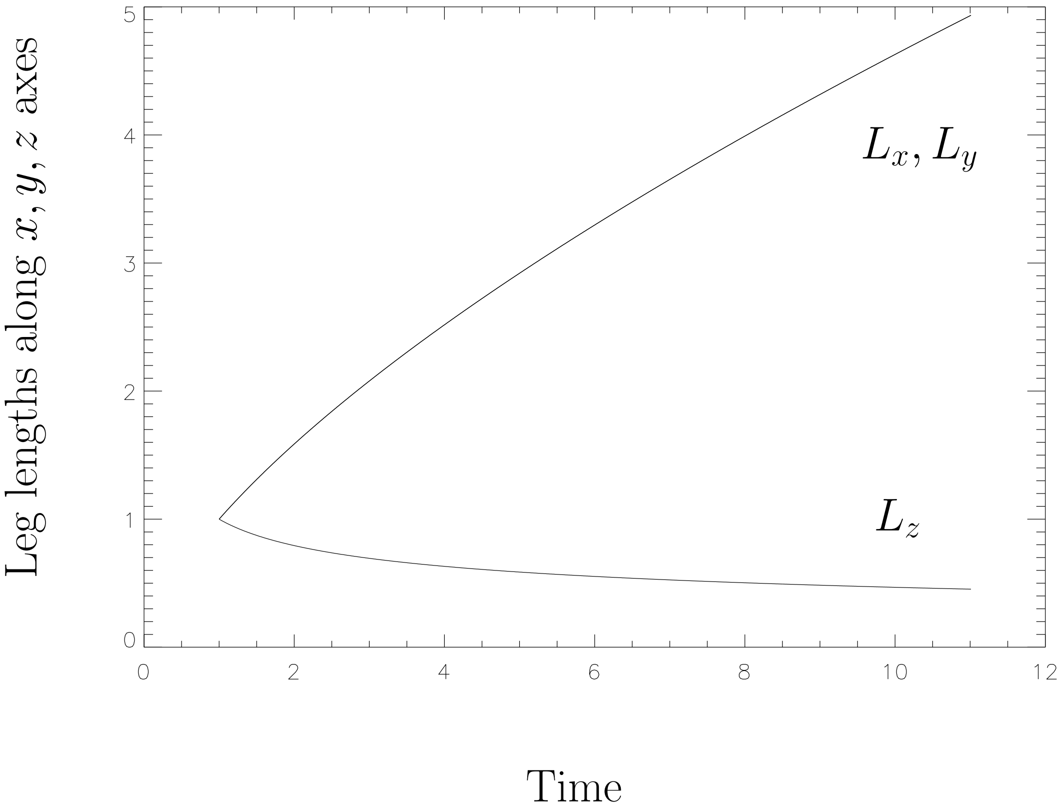

Figure 6. This figure displays the evolution of three leg lengths, one each along the three coordinate axes. In this and the following plots, $\Delta t = 0.01, p_x = p_y = 2/3, p_z = -1/3$ and the initial data was constructed with $\Delta x = \Delta y = \Delta z = 1$. The curves for L_x and L_y are coincident.

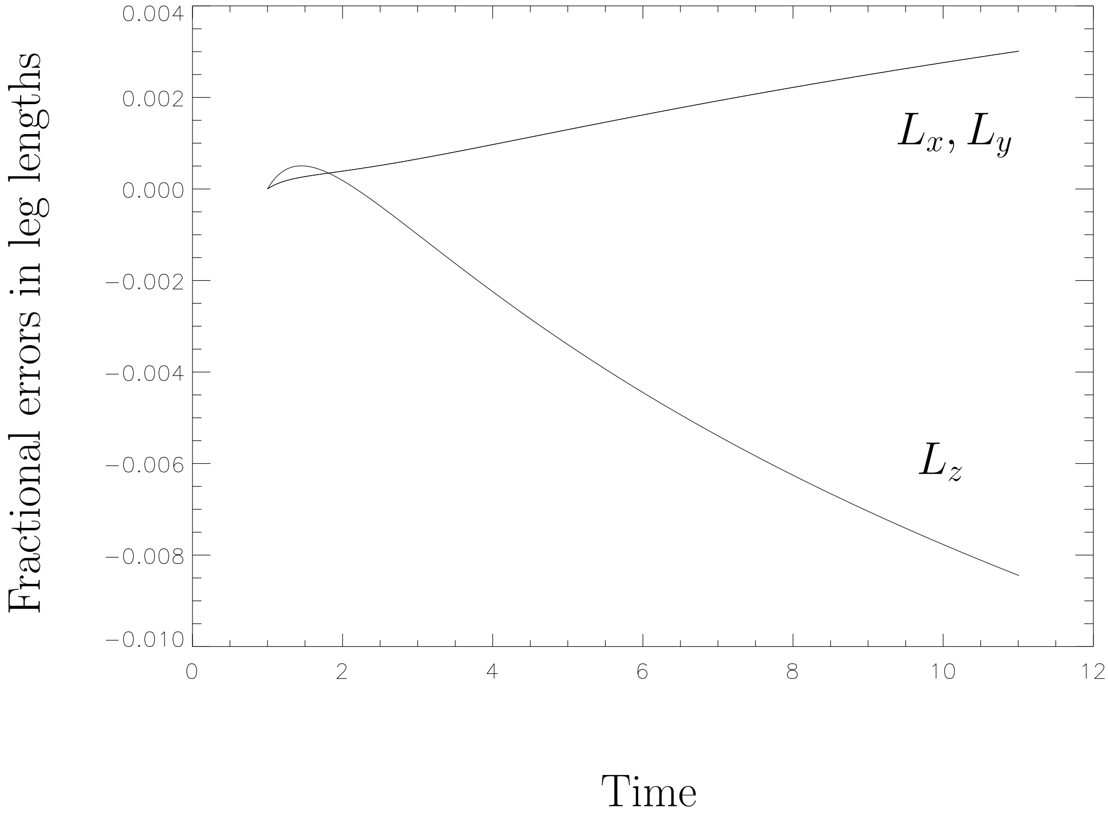

Figure 7. This figure displays the fractional errors associated with the three leg lengths. Once again there is no difference between the curves for L_x and L_y. These errors arise solely from the numerical time integration (as demonstrated in Figure (12)).

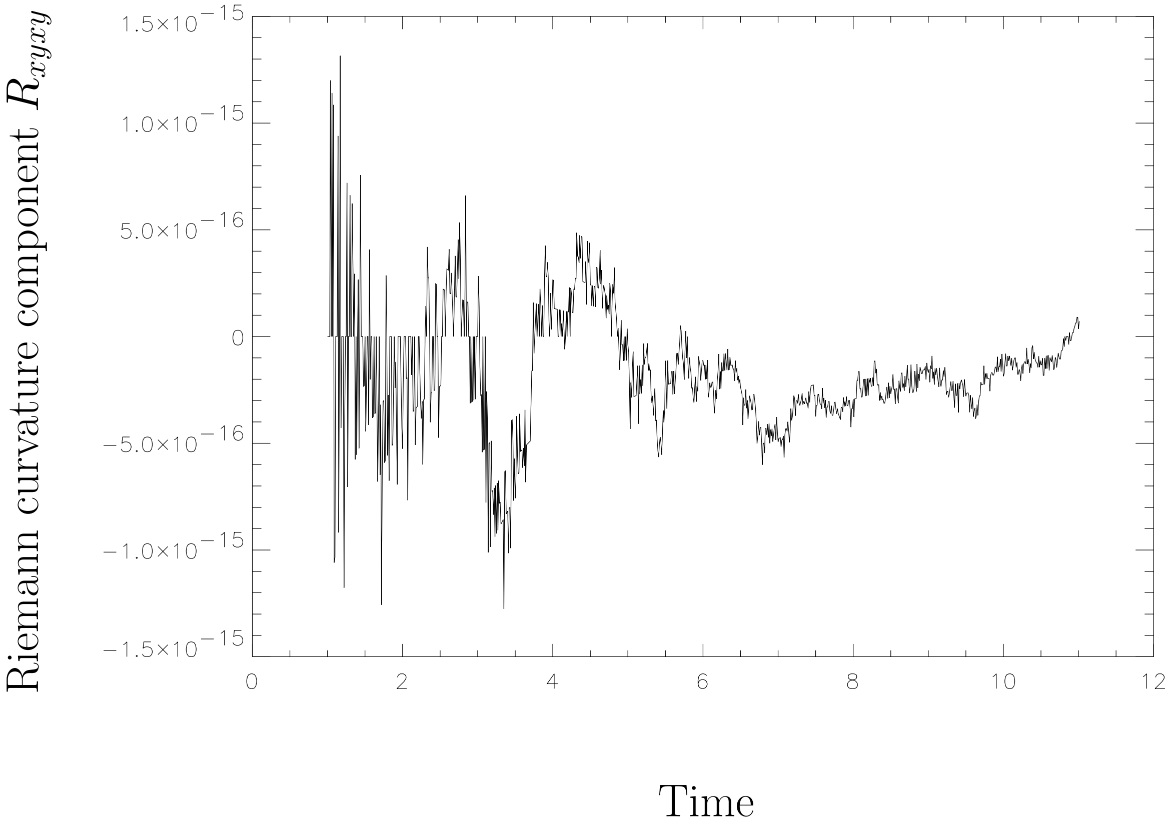

Figure 8. This figure displays the R_{xyxy} component of the Riemann tensor as a function of time. R_{xzxz} and R_{yzyz} behave in a similar fashion while the remaining three curvatures remain at approximately 10^{-30} throughout the evolution. The typical leg lengths are of order 1 and thus from the smooth lattice equations (2.1.2) we can see that the Riemann curvatures are zero to within machine precision (with 64-bit arithmetic this is of order 15 decimal digits).

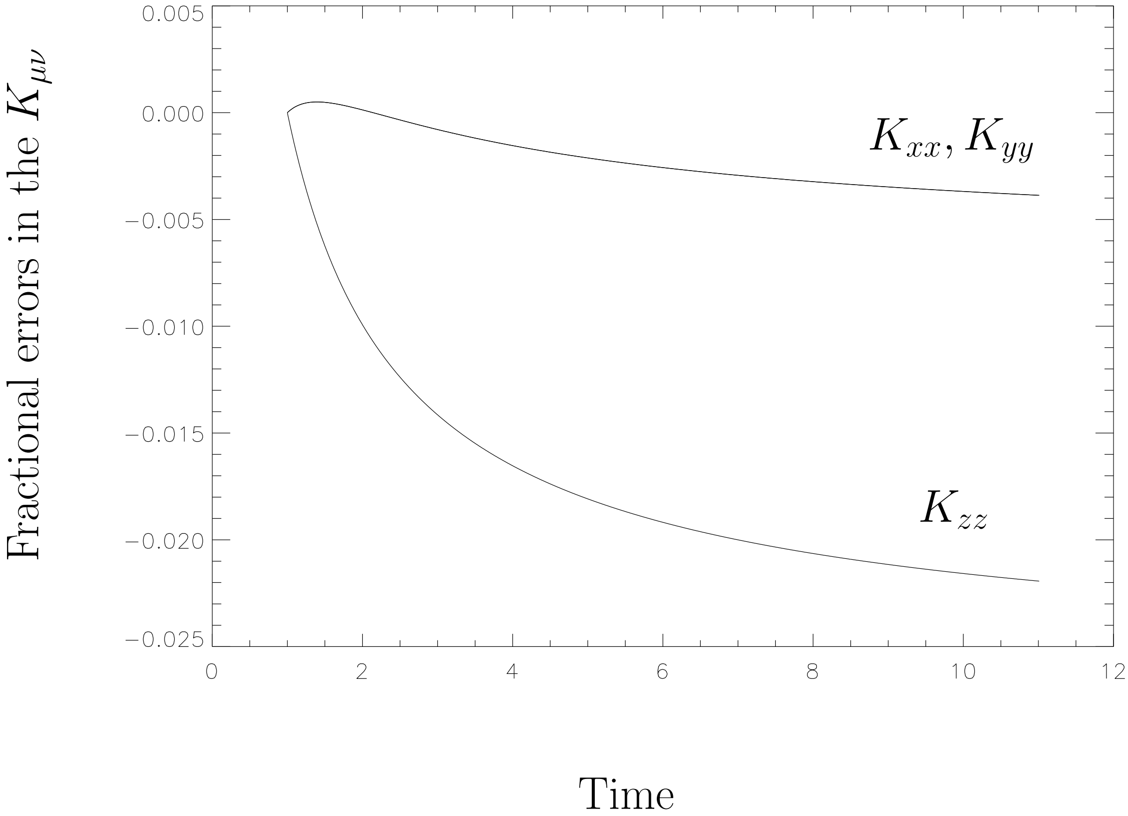

Figure 9. This figure displays the evolution of the fractional errors in the diagonal components of the extrinsic curvatures. The off diagonal components are zero to within machine precision (see Figure (10)). The fractional errors for K_{xx} and K_{yy} are equal (to within machine precision).

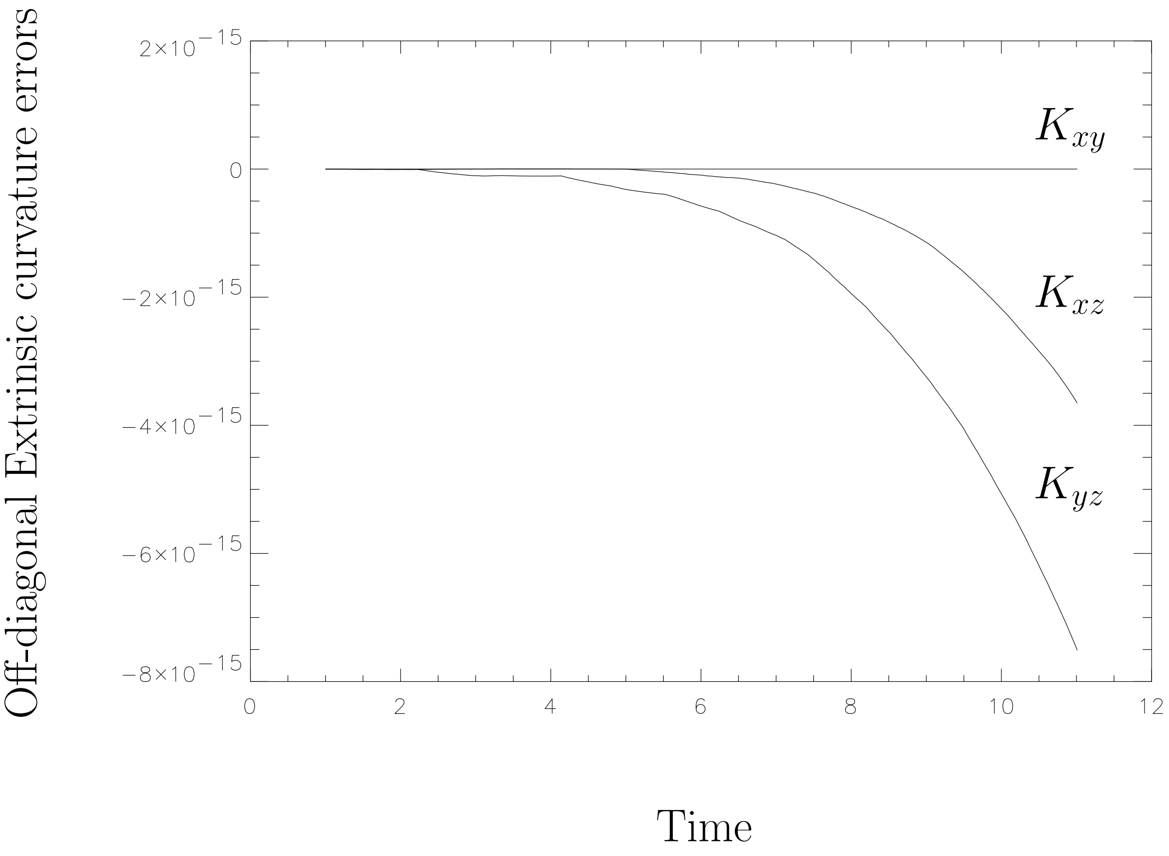

Figure 10. As per Figure (8) but for the off-diagonal extrinsic curvature components. Note that in this case there is an asymmetry between the x and y axes though at this level the differences are insignificant (they are equal to within machine precision when compared to the diagonal terms).

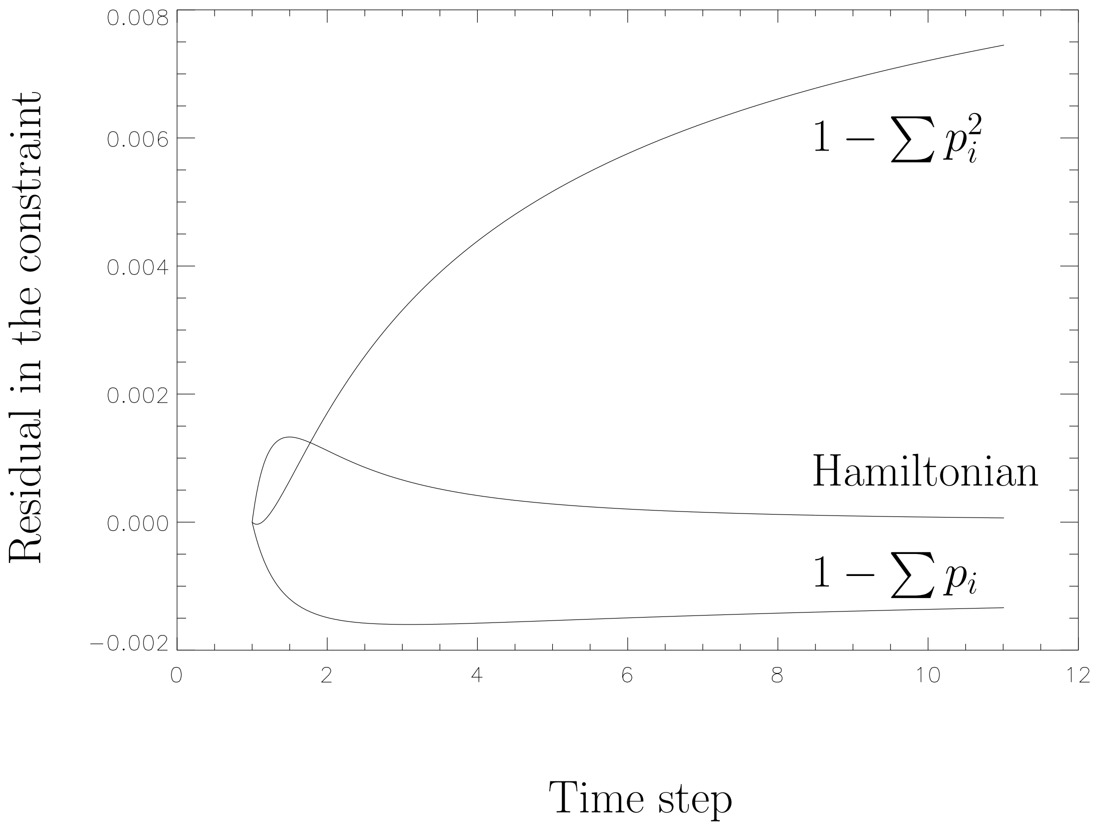

Figure 11. The figure displays the residuals in the two Kasner constraints, equations (4.1–4.2), and the Hamiltonian constraint, equation (4.3).

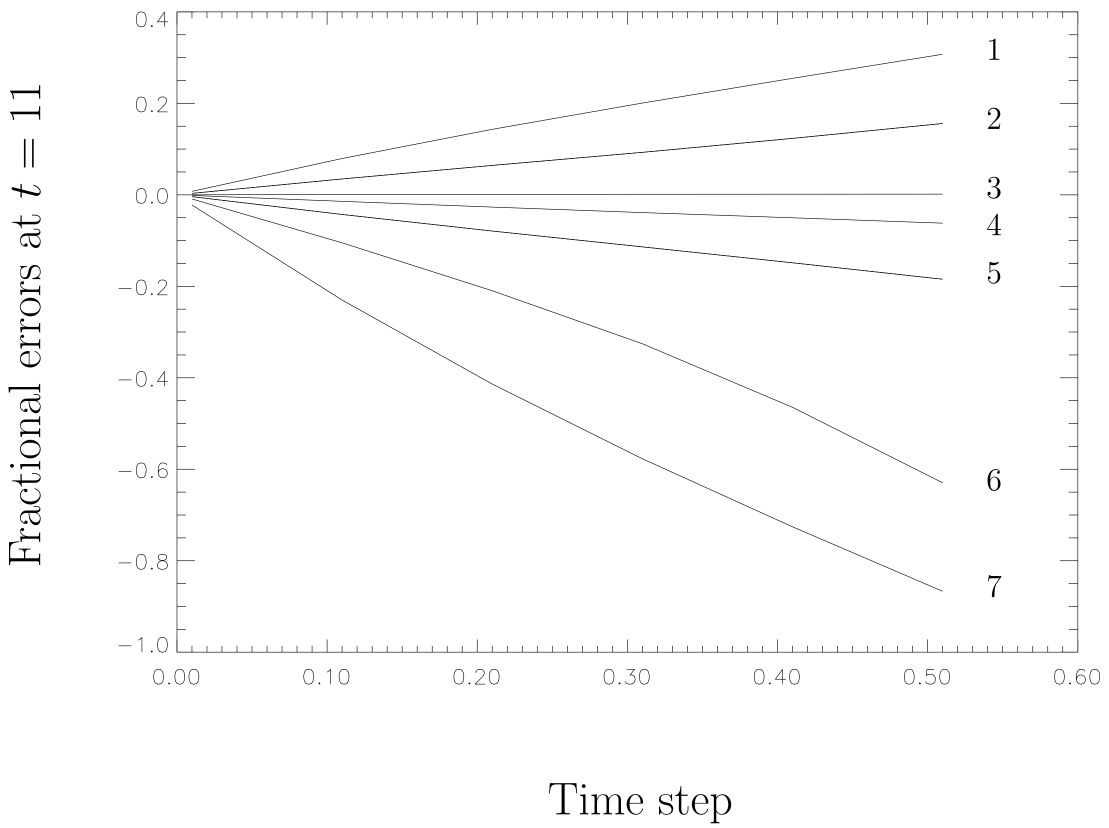

Figure 12. This figure displays the various errors at $t = 11$ as a function of the time step. Each curve is straight, as expected for an Euler integration, and each curve can be extrapolated through the origin proving that the sole source of error in the evolution of the lattice is due solely to the use of an in-exact numerical time integration. The curves are 1) $1 - \sum p_i^2$, 2) the fractional errors (f.e.'s) in L_x, L_y, 3) the Hamiltonian, 4) $1 - \sum p_i$, 5) the f.e.'s in K_{xx}, K_{yy}, 6) the f.e.'s in L_{zz} and 7) the f.e.'s in K_{zz}. Six evenly spaced time steps between 0.01 and 0.51 were used. The data points were not plotted so as to make it easier to see the convergence to zero.