Smooth lattice construction of the Oppenheimer-Snyder spacetime

Leo Brewin[*] and Jules Kajtar[†]

School of Mathematical Sciences Monash University, 3800 Australia
(Received 23 May 2009; published 5 November 2009)

We present test results for the smooth lattice method using an Oppenheimer-Snyder spacetime. The results are in excellent agreement with theory and numerical results from other authors.

I. INTRODUCTION

In recent times many numerical relativists have good reason to celebrate—the long battle to secure the holy grail [1] is over (though some might prefer to redraw the battle lines). The works of Pretorius [2,3] and others [4,5] have opened a new era for computational general relativity. This has spawned many new projects that directly address the needs of the gravitational wave community. Many groups are now running detailed simulations of binary systems in full general relativity as a matter of course. Does this mean that the development of computational methods for general relativity is now over? The experience in other fields would suggest otherwise, look, for example, at computational fluid dynamics where a multitude of techniques are commonly used, including spectral methods, finite element methods, smooth particle hydrodynamics, high resolution shock capture methods and the list goes on. The important point to note is that one method does not solve all the problems and thus in numerical relativity it is wise, even in the face of the current successes, to seek other methods to solve the Einstein equations. It is in that spirit that we have been developing what we call the smooth lattice method [6–8]. This is a fundamentally discrete approach to general relativity based on a large collection of short geodesic segments connected to form a lattice representation of spacetime. The Einstein equations are cast as evolution equations for the leg-lengths with the Riemann and energy-momentum tensors acting as sources. Of course the Riemann tensor must be computed from the leg-lengths and this can be done in a number of related ways, such as by fitting a local Riemann normal coordinate expansion to a local cluster of legs or to use the geodesic deviation equation, or, and with more generality, to use the second variation of arc-length. Past applications of the method have included a full $3 + 1$ simulation of the vacuum Kasner cosmology [8] and a $1 + 1$ maximally sliced Schwarzschild spacetime [6]. In both cases the simulations were stable and showed excellent agreement with the known solutions while showing no signs of instabilities (the maximally sliced Schwarzschild solution ran for $t > 1000m$ and was stopped only because there was no point in running the code any longer).

In this paper we report on our recent work using the Oppenheimer-Snyder [9] spacetime as a benchmark for our smooth lattice method [6–8]. We chose this spacetime for many reasons, it has been cited by many authors [10–16] as a standard benchmark for numerical codes (and thus comparative results are available), the analytic solution is known (in a number of time slicings), the equations are simple and there are many simple diagnostics that can be used to check the accuracy of the results (as described in Secs. XI and XII).

In an impressive series of papers, Shapiro and Teukolsky ([10,11,14]) used the Oppenheimer-Snyder spacetime as the first in a series of test cases. They were motivated by certain problems in relativistic stellar dynamics (such as the formation of neutron stars and black holes from supernova) and they developed a set of codes based on the standard ADM equations, adapted to spherical symmetry, in both maximal and polar slicing and using an N-body particle simulation for the hydrodynamics. They made limited use of the exact Schwarzschild solution to develop an outer boundary condition for the lapse function while using both the Schwarzschild and FRW solutions to set the initial data. Though their discussion on the size of their errors is brief (for the Oppenheimer-Snyder test case), they did note that the errors were of the order of a percent or so (for a system with 240 grid points and 1180 dust particles). In a later work, Baumgarte *et al.* [15] extended their work by expressing the metric and the equations in terms of an outgoing null coordinate. This leads to a slicing that covers all of the spacetime outside (and arbitrarily close to) the event horizon. In this version of their code Baumgarte *et al.* [15] chose to solve only the equations for the dustball by using the Schwarzschild solution as an outer boundary condition.

This idea, to replace the exterior equations with the known Schwarzschild solution, has been used by Gourgoulhon [13], Schinder *et al.* [12] and Romero *et al.* [16]. Gourgoulhon [13] used a radial gauge and polar slicing while solving the equations using a spectral method and reported errors in the metric variables between 10^{-7} to 10^{-5}. However, with the onset of the Gibbs phenomena, the code could only be run until the central lapse collapsed to around 2×10^{-3}. Schinder *et al.* [12] used the same equations as Gourgoulhon [13] but with a discretization based on a standard finite difference scheme. They re-

―――――――
*Leo.Brewin@sci.monash.edu.au
†Jules.Kajtar@sci.monash.edu.au

ported errors of order 1% for evolution times similar to those of Gourgoulhon [13]. The work of Romero *et al.* [16] differs from that of Schinder *et al.* [12] in that they used high-resolution shock capture methods for the hydrodynamics. They report evolutions down to a central lapse of 1.3×10^{-10}.

Our results compare very well against those given above with our errors being of the order fractions of a percent for 1200 grid points. Our code runs, without any signs of instabilities, for maximal slicing out to $t = 500m$ where the central lapse has collapsed to 10^{-110}. We make no use of the known solutions other than the conservation of local rest energy (we use a particle like method to compute the rest density). We also provide extensive comparisons of our results with the exact solution (see Sec. XII).

In the following sections we will describe all aspects of our code, including the design of the lattice (Sec. II), the curvature and evolution equations (Secs. III, IV, and VI), computing the density (Sec. VII), the junction conditions (Sec. IX), setting the initial data (Sec. X E) and finally the results (Sec. XII).

We will make frequent reference to two papers, our earlier paper on the Schwarzschild spacetime [8] and a companion paper showing how the Einstein equation can be applied to the lattice [17]. We will refer to these as Paper 1 and Paper 2, respectively.

II. THE OPPENHEIMER-SNYDER LATTICE

What design should we choose for the lattice? We will take a minimalist approach—build the simplest lattice that captures the required symmetries while being sufficiently general to allow the full dynamics to be expressed through the evolution of the lattice data. Here is a construction of such a lattice. Take a single spacelike radial geodesic, in one Cauchy surface, extending from the center of the dustball out to the distant asymptotically flat regions and subdivide it into a series of short legs with lengths denoted by L_{zz}. We will refer to the end points of each leg as the lattice nodes. Note that we are free to choose the L_{zz} as we see fit (in the same way that we are free to choose the lapse function in an ADM evolution). Now construct a clone of this geodesic by rotating it through any small angle (while remaining in the Cauchy surface). Finally connect the corresponding nodes of the pair of geodesics by a second set of geodesic legs, with lengths denoted this time by L_{xx} (see Fig. 1). We now have a spacelike three-dimensional lattice contained within one Cauchy surface. From here on in we allow this lattice to vary smoothly with time.

Note that each leg of this lattice is a geodesic segment of the 3-metric of the Cauchy surface. We could also connect the nodes of the lattice with geodesic segments of the full four-dimensional spacetime (much like constructing chords to arcs of a circle). This gives us two representations of the lattice, both sharing the same node points with the
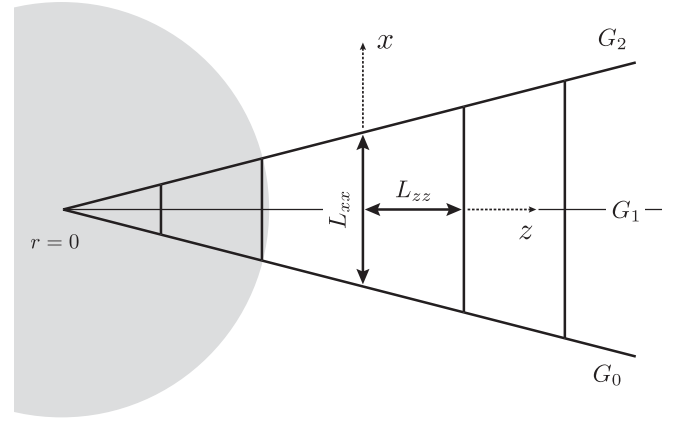

FIG. 1. In this figure we show how the lattice is constructed from two radial geodesics G_0 and G_2 and the series of interconnecting legs L_{xx}. The third geodesic G_1 lies midway between G_0 and G_2 is used to define the radial legs L_{zz}. The grey patch to the left represents (part of) the dustball. Note that the L_{xx} legs have been drawn as straight lines as a reminder that these legs are geodesic segments.

first composed of short 3-geodesics and the second composed of short 4-geodesics. Suppose that typical leg-lengths in the two representations are ${}^3L_{ij}$ and ${}^4L_{ij}$ respectively. Then it is not hard to see that ${}^4L_{ij} = {}^3L_{ij} + \mathcal{O}({}^3L_{ij}^3)$. The upshot is that in all of our equation in this paper we are free to use either representation (the differences being at least as small as the truncation errors).

The L_{zz} and L_{xx} are all that we need to describe the geometry of each Cauchy surface but we also need some way to represent the dustball on the lattice. Again, we shall take a minimalist approach—we know that the dust can be described as a set of particles travelling on timelike geodesics with conserved rest mass. Thus we add a series of dust particles on the radial geodesic with each particle carrying a conserved rest mass.

As noted above, we are free to distribute the lattice nodes as we see fit. How should we do this? We know that the dustball will collapse so it makes sense to tie the lattice nodes to the dust particles, i.e. the lattice nodes follow infalling timelike geodesics. But what of the nodes outside the dustball? Again, by appeal to simplicity, we demand that *every* lattice node, interior and exterior, follow the infalling timelike geodesics. In this scheme the lattice nodes do *not* follow the trajectories normal to the Cauchy surface (in contrast to the scheme in Paper 1). This introduces a drift vector γ^μ (see Fig. 2) (which is similar to but distinct from the shift vector, see [8]).

The lattice just described differs from the Schwarzschild lattice of Paper 1 in a number of important ways—it contains an internal boundary (the edge of the dustball), the lattice nodes are not at rest in each Cauchy surface, the lattice carries a set of dust particles and at the inner boundary $L_{xx} = 0$. Thus we will need to develop new
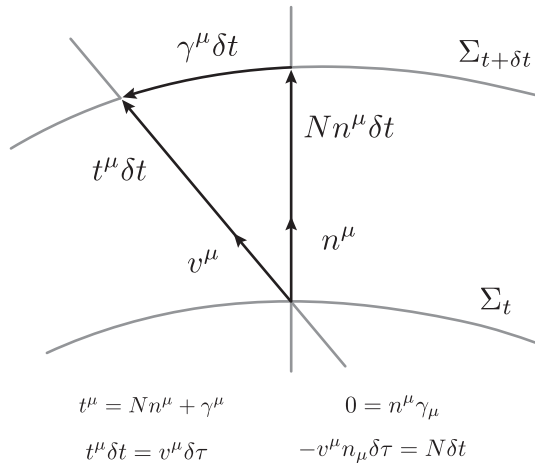
FIG. 2. In this figure we introduce most of the kinematical quantities on the lattice. The dust particle's unit 4-velocity is v^μ, while γ^μ is the drift vector, n^μ the unit normal to the Cauchy surface Σ_t, N is the lapse function and $\delta\tau$ is the proper time measured along the dust particle's trajectory.

boundary conditions (Sec. IX), new evolution equations for the nodes (i.e. adapt the geodesic equations to the lattice, Sec. VI) and an algorithm to compute the rest energy density from the rest masses carried by the dust particles (Sec. VII).

In Paper 1 we employed Riemann normal coordinates as a stepping stone to develop the purely scalar equations for the leg-lengths, time derivatives, constraints etc. We went on to speculate whether or not these coordinates imbued the numerical scheme with any favorable properties (we argued that they did not). One way to avoid this coordinate issue is simply to derive the equations without reference to a coordinate system. In this paper we will represent tensors, such as the Riemann and extrinsic curvatures, by their frame components. We will use an orthonormal frame built as follows. We choose the first two basis vectors, m_x^μ and m_z^μ, to be the unit tangent vectors to L_{xx} and L_{zz} respectively at the midpoint of L_{xx} (see Fig. 1). The remaining two basis vectors (m_y^μ and m_n^μ) can be chosen freely (subject to the orthornormal condition, e.g. m_n^μ could be chosen as the unit normal to the Cauchy surface). With this choice of basis a typical frame component for the extrinsic curvature could be written as $K_{\mu\nu}m_a^\mu m_b^\mu$. Such notation quickly becomes tiresome so we will introduce the abbreviation $\mathcal{K}_{ab}$ to represent $K_{\mu\nu}m_a^\mu m_b^\mu$ with an obvious generalization to other tensors.

We will allow a slight variation to this notation. On occasions we will find it useful to refer to a leg by its end points, such as i and j. That leg will have its own unit tangent vector which we denote by m^μ. We will then take $\mathcal{K}_{ij}$ to be $K_{\mu\nu}m^\mu m^\nu$. This small change will only ever be used for the extrinsic curvature.

The dust particles follow future pointing timelike geodesics. We will use v^μ to denote the velocity 4-vector of

the dust particles and we will record the frame components as $v_n = -v_\mu n^\mu$ and $v_z = v_\mu m_z^\mu$ where n^μ is the unit normal to the Cauchy surface.

The notation just introduced sits quite nicely with the notation used in Paper 1. In that paper we wrote K_{xx}, for example, to denote the x-x coordinate components of $K_{\mu\nu}$ in the local Riemann normal frame. In that frame we chose the three metric $g_{\mu\nu}(x)$ at the origin to equal diag(1,1,1) and the basis vectors m_a^μ to have values δ_a^μ. Thus $K_{xx} = K_{\mu\nu}m_x^\mu m_x^\nu$. The upshot is that coordinate components of Paper 1 have the same numerical values as the frame components used in this paper. Thus we would reasonably expect that the equations used in Paper 1 should carry over to this paper with only minor changes to accommodate the introduction of the dust. This indeed proves to be the case (which is reassuring). The details will be presented in Sec. IV where we will use a formalism developed in Paper 2 to derive, from scratch, the evolution equations for the lattice.

III. THE RIEMANN CURVATURES

The question here is: How do we compute the Riemann curvatures, $\mathcal{R}_{xyxy}$ and $\mathcal{R}_{xzxz}$, from the L_{xx} and L_{zz}? In our previous paper, Paper 1, we computed two Riemann curvatures, $\mathcal{R}_{xyxy}$ and $\mathcal{R}_{xzxz}$, using

$$0 = \frac{d^2 L_{xx}}{dz^2} + \mathcal{R}_{xzxz}L_{xx} \qquad \text{geodesic deviation,} \qquad (1)$$

$$0 = \frac{d(L_{xx}^2 \mathcal{R}_{xyzy})}{dz} - \mathcal{R}_{xzxz}\frac{dL_{xx}^2}{dz} \qquad \text{Bianchi identity.} \qquad (2)$$

These equations will be used as follows. First we use the geodesic deviation equation to compute the $\mathcal{R}_{xzxz}$ for each node across the lattice. This then allows us to integrate the Bianchi identity for $\mathcal{R}_{xyxy}$ from the center to the outer boundary.

This scheme sounds simple but there a number of (obvious) complications. Firstly, the equations are singular at the center (where $L_{xx} = 0$) and secondly, $\mathcal{R}_{xzxz}$ will not be continuous across the junction. These complications are new to this investigation but we also inherit one further complication from Paper 1: what boundary condition should we use at $z = 0$ when integrating the Bianchi identity? This last problem is rather easy to deal with. At the center of the dustball we know that the metric must be isotropic and thus we can be certain that $\mathcal{R}_{xyxy} = \mathcal{R}_{xzxz}$ at $z = 0$.

How do we handle the singularity at $z = 0$? Again, by symmetry arguments we can assert that $\mathcal{R}_{xzxz,\mu} = 0$ at $z = 0$. Thus in a small neighborhood of $z = 0$ we must have $\mathcal{R}_{xzxz} = A + Bz^2$ where A and B are independent of position and z is the radial proper distance measured from $z = 0$. Thus it is not unreasonable to use a quadratic *interpo-*

lation of $\mathcal{R}_{xzxz}$ to estimate $\mathcal{R}_{xzxz}$ at $z = 0$. Our experience shows that this works very well but it does require some care (see Sec. X A for the full details). Before dealing with the junction issue we should emphasize that this process is an interpolation rather than an extrapolation of the data to $z = 0$. To see this just imagine extending the radial geodesics of the lattice through $z = 0$ so that we can use $\mathcal{R}_{xzxz} = A + Bz^2$ with z in a range $-z_0 < z < z_0$ for some small z_0.

The frame components $\mathcal{R}_{xx} = R_{\mu\nu}m_x^\mu m_x^\nu$ and $\mathcal{R}_{zz} = R_{\mu\nu}m_z^\mu m_z^\nu$ of the Ricci tensor are rather easy to construct from $\mathcal{R}_{xyxy}$ and $\mathcal{R}_{xzxz}$. Using the orthonormal frame m_x^μ, m_y^μ and m_z^μ we can easily deduce that

$$\mathcal{R}_{xx} = \mathcal{R}_{xyxy} + \mathcal{R}_{xzxz}, \tag{3}$$

$$\mathcal{R}_{yy} = \mathcal{R}_{xyxy} + \mathcal{R}_{xzxz}, \tag{4}$$

$$\mathcal{R}_{zz} = 2\mathcal{R}_{xzxz}, \tag{5}$$

with all other $\mathcal{R}_{ab} = 0$. From these equations it is easy to verify that the scalar curvature is given by

$$R = 2\mathcal{R}_{xyxy} + 4\mathcal{R}_{xzxz}. \tag{6}$$

The one remaining complication is the discontinuity in $\mathcal{R}_{xzxz}$ at the junction. This will be discussed in detail in Sec. IX.

IV. THE EVOLUTION EQUATIONS

The equations of Paper 1 have served us well so far but now we must chart a new path. The reason is that, unlike our approach in Paper 1, here we allow the lattice nodes to drift across the Cauchy surfaces and this will introduce extra terms in the evolution equations. There is also the issue of introducing the energy-momentum sources but, as we shall see later, this is really very easy to do (it amounts to little more than adding a term of the form $8\pi k T_{\mu\nu}m^\mu m^\nu$ to the vacuum equations). So how do we develop evolution equations for a nonzero drift vector? In Paper 2 we showed how the standard $3 + 1$ ADM equations with a zero shift vector can be recovered from the equations for the second variation of arc-length. And as arc-lengths of geodesics are central to our smooth lattice approach this new formalism is well suited to our current task.

We begin by recalling from Paper 2 the equations for the first and second variation of the geodesic segment that connects nodes i and j

$$\frac{dL_{ij}}{dt} = [m_\mu t^\mu]_i^j = \int_i^j m_\mu m^\nu t^\mu{}_{;\nu} ds, \tag{7}$$

$$\frac{d^2 L_{ij}}{dt^2} = [t^\alpha{}_{;\mu}t^\mu m_\alpha]_i^j - \int_i^j {}^4 R_{\mu\alpha\nu\beta}m^\mu m^\nu t^\alpha t^\beta ds$$
$$+ \int_i^j (t_{\mu;\alpha}t^\mu{}_{;\nu}m^\alpha m^\nu - (m_\mu m^\nu t^\mu{}_{;\nu})^2) ds. \tag{8}$$

It is tempting to jump in by setting $t^\mu = N n^\mu + \gamma^\mu$ and to let the equations take us where they will. Indeed this works well for the first variation. We start be making the said substitution and massage the result as follows

$$\frac{dL_{ij}}{dt} = [m^\mu t_\mu]_i^j = [m^\mu (N n_\mu)]_i^j + [m^\mu \gamma_\mu]_i^j$$

$$= \int_i^j m^\mu m^\nu (N n_\mu)_{;\nu} ds + [m^\mu \gamma_\mu]_i^j$$

$$= \int_i^j m^\mu m^\nu (-N K_{\mu\nu}) ds + [m^\mu \gamma_\mu]_i^j$$

$$= -N \mathcal{K}_{ij} L_{ij} + [m^\mu \gamma_\mu]_i^j.$$

In the second last line we have used $N n_{\mu;\nu} = -(\perp N_{,\mu})n_\nu - N K_{\mu\nu}$ and $m^\mu n_\mu = 0$ while in writing the last line we have assumed that the leg-length is sufficiently short that the integrand can estimated by a simple quadrature, in this case the midpoint rule (later in Sec. X C we will have reason to change this to a Trapezoidal rule). A similar equation can be found in Paper 2 (differing only in the absence of the γ terms). For the pair of legs L_{xx} and L_{zz} we thus obtain

$$\frac{dL_{xx}}{dt} = -N \mathcal{K}_{xx} L_{xx} + [m_x^\mu \gamma_\mu]$$

$$\frac{dL_{zz}}{dt} = -N \mathcal{K}_{zz} L_{zz} + [m_z^\mu \gamma_\mu]$$

and to keep the notation a little less cluttered we have not written the end points on the $[\cdots]$ terms.

What can we say about the $[m_\mu \gamma^\mu]$ terms? Let $\tilde{\gamma}^\mu$ be the unit vector parallel to γ^μ. Then we can immediately use the first variation equation once again (see Fig. 3) to deduce that $[m_{x\mu}\tilde{\gamma}^\mu]$ equals dL_{xx}/dz where, as usual, z is the radial proper distance measured along G_1. However, $\tilde{\gamma}^\mu = \gamma^\mu/\gamma^z$ and by spherical symmetry we know that γ^z does not change from one radial geodesic to the next. Thus we deduce that $[m_{x\mu}\gamma^\mu] = \gamma^z dL_{xx}/dz$. We now turn to the other leg, L_{zz}. In this case m_z^μ and γ^μ are parallel and thus we can not invoke the first variation equation. But that is of
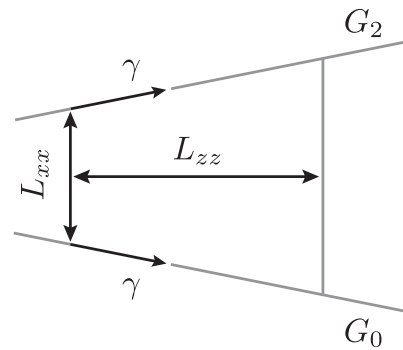

FIG. 3. In this diagram it is easy to see that $[m_{x\mu}\tilde{\gamma}^\mu] = dL_{xx}/dz$ where $\tilde{\gamma}$ is the unit vector parallel to γ. This result is used in Sec. IV when deriving Eq. (9).

no concern simply because $m_{z\mu}\gamma^\mu = \gamma^z = Nv_z/v_n$. Thus we have $[m_{z\mu}\gamma^z] = [Nv_z/v_n]$. The equations for the first time derivatives of L_{xx} and L_{zz} can now be written as

$$\frac{dL_{xx}}{dt} = -N\mathcal{K}_{xx}L_{xx} + \left(\frac{Nv_z}{v_n}\right)\frac{dL_{xx}}{dz}, \qquad (9)$$

$$\frac{dL_{zz}}{dt} = -N\mathcal{K}_{zz}L_{zz} + \left[\frac{Nv_z}{v_n}\right]. \qquad (10)$$

We turn our attention now to adapting the equations for the second variation to our simple lattice.

Our job would be greatly simplified if it happened that $\gamma^\mu = 0$ but on the current lattice that is not the case. So we now introduce a second lattice on which we set $\gamma^\mu = 0$. The nodes of the first lattice will follow the dust particles while those of the second lattice will follow trajectories normal to their Cauchy surfaces (which will differ from those of the first lattice). Note that the second lattice has been introduced solely to aid the exposition—the second lattice will never be needed nor used in our actual computer programs. To keep the bookkeeping clear we will identify data on the second lattice by the addition of a dash. The second lattice is created at some generic time, say $t = t_0$, and we choose to assign identical initial data to both lattices, i.e. $L'_{ij} = L_{ij}$, $\mathcal{K}'_{ij} = \mathcal{K}_{ij}$ etc. on $\Sigma(t_0)$. We have no reason to use distinct Cauchy surfaces for each lattice (we only want to set $\gamma^\mu = 0$) so we are free to set $N' = N$ and $dN'/dt = dN/dt$. It follows that we also have $^4R'_{\mu\nu\alpha\beta} = {}^4R_{\mu\nu\alpha\beta}$ across $\Sigma(t_0)$. Our task now is to adapt the equations for the first and second variations to the second lattice. This has already been done in Paper 2 where we have shown that

$$\frac{dL'_{ij}}{dt} = -NK'_{\mu\nu}m^\mu m^\nu L'_{ij}$$

$$\frac{d^2L'_{ij}}{dt^2} = \frac{1}{N}\frac{dN}{dt}\frac{dL'_{ij}}{dt} - \frac{1}{L'_{ij}}\left(\frac{dL'_{ij}}{dt}\right)^2$$
$$+ N^2K'_{\mu\alpha}K'^\mu{}_\beta m^\alpha m^\beta L'_{ij} + NN_{;\alpha\beta}m^\alpha m^\beta L'_{ij}$$
$$- N^2({}^4R_{\mu\alpha\nu\beta})m^\mu m^\nu n^\alpha n^\beta L'_{ij}.$$

Where we head next depends upon what type of equations we wish to work with. We can develop either a second-order set of equations involving both dL'_{ij}/dt and $d^2L'_{ij}/dt^2$ or a first order system involving dL'_{ij}/dt and $d\mathcal{K}'_{ij}/dt$. We will take the second approach for two reasons, it mimics the standard ADM approach and, more importantly, it eliminates the dN/dt term (which would add undue complexity when using maximal slicing). Between this pair of equations we can easily eliminate $d^2L'_{ij}/dt^2$ with the following result

$$\frac{d\mathcal{K}'_{ij}}{dt} = -N_{;\alpha\beta}m^\alpha m^\beta + N({}^4R_{\mu\alpha\nu\beta})m^\mu m^\nu n^\alpha n^\beta$$
$$+ 2N(\mathcal{K}'_{ij})^2 - NK'_{\mu\alpha}K'^\mu{}_\beta m^\alpha m^\beta,$$

where $\mathcal{K}'_{ij} := K'_{\mu\nu}m^\mu m^\nu$. This last equation controls the evolution of $\mathcal{K}'_{ij}$ for the second lattice. Can we use this information to deduce the evolution of $\mathcal{K}_{ij}$ on the first lattice? Yes, by simply splitting the evolution into a part parallel to the normal plus a part parallel to the drift vector. Since $\mathcal{K}_{ij}$ is a scalar function we can use a standard chain rule to write

$$\frac{d\mathcal{K}_{ij}}{dt} = \frac{d\mathcal{K}'_{ij}}{dt} + \mathcal{K}'_{ij,\mu}\gamma^\mu$$

and as $\mathcal{K}_{ij} = \mathcal{K}'_{ij}$ on $\Sigma(t_0)$ we arrive at

$$\frac{d\mathcal{K}_{ij}}{dt} = -N_{;\alpha\beta}m^\alpha m^\beta + N({}^4R_{\mu\alpha\nu\beta})m^\mu m^\nu n^\alpha n^\beta$$
$$+ 2N(\mathcal{K}_{ij})^2 - NK_{\mu\alpha}K^\mu{}_\beta m^\alpha m^\beta + \mathcal{K}_{ij,\mu}\gamma^\mu.$$

In a moment we will apply this equation to $\mathcal{K}_{xx} = K_{\mu\nu}m^\mu_x m^\nu_x$ and $\mathcal{K}_{zz} = K_{\mu\nu}m^\mu_z m^\nu_z$ but first we recall that both γ^μ and m^μ are tangent to Σ, $T^{\mu\nu} = \rho v^\mu v^\nu$ and for our spherically symmetric lattice, $K_{\mu\nu}$ is diagonal and $\gamma^z = Nv_z/v_n$. We will also need the contracted Gauss equation, namely,

$${}^4R_{\mu\alpha\nu\beta}m^\mu m^\nu n^\alpha n^\beta = (-\bot^4R_{\mu\nu} + R_{\mu\nu} + KK_{\mu\nu}$$
$$- K_{\alpha\mu}K^\alpha{}_\nu)m^\mu m^\nu.$$

We then find that the above equation for $d\mathcal{K}_{ij}/dt$ when applied to $\mathcal{K}_{xx}$ and $\mathcal{K}_{zz}$ leads to

$$\frac{d\mathcal{K}_{xx}}{dt} = -N_{,xx} + N(\mathcal{R}_{xx} + K\mathcal{K}_{xx} - 4\pi k\rho)$$
$$+ \left(\frac{Nv_z}{v_n}\right)\mathcal{K}_{xx,z} \qquad (11)$$

$$\frac{d\mathcal{K}_{zz}}{dt} = -N_{,zz} + N(\mathcal{R}_{zz} + K\mathcal{K}_{zz} + (4 - 8v_n^2)\pi k\rho)$$
$$+ \left(\frac{Nv_z}{v_n}\right)\mathcal{K}_{zz,z}. \qquad (12)$$

This pair of equations coupled with (9) and (10) are the evolution equations for the lattice.

V. THE CONSTRAINTS

The general form of the Hamiltonian and momentum constraints are

$$R + K^2 - K_{\mu\nu}K^{\mu\nu} = 16\pi kT_{\mu\nu}n^\mu n^\nu$$

$$\bot(K_{|\nu} - K^\mu{}_{\nu|\mu}) = 8\pi k\bot(T_{\mu\nu}n^\mu),$$

where $K = K^\mu{}_\mu$. It is a simple matter to apply these

equations to the Schwarzschild spacetime, see Paper 1 for details. For the present case we need to account for the nonzero $T_{\mu\nu}$ in the interior of the dustball. We can easily adapt the equations of Paper 1 by simply adding on the terms $8\pi kT_{\mu\nu}n^\mu n^\nu$ for the Hamiltonian and $8\pi kT_{\mu\nu}n^\mu m_z^\nu$ for the momentum constraints (projections in the other two directions m_x^μ and m_y^μ yield the trivial equation $0 = 0$). This leads to

$$0 = \mathcal{R}_{xyxy} + 2\mathcal{R}_{xzxz} + \mathcal{K}_{xx}^2 + 2\mathcal{K}_{xx}\mathcal{K}_{zz} - 8\pi k\rho v_n^2, \tag{13}$$

$$0 = \frac{1}{L_{xx}}\left(\mathcal{K}_{zz}\frac{dL_{xx}}{dz} - \frac{d(L_{xx}\mathcal{K}_{xx})}{dz}\right) - 4\pi k\rho v_n v_z \tag{14}$$

(for simplicity we have cleared a common factor of 2 from both equations).

VI. THE PARTICLE EQUATIONS

Here we will derive the equations governing the evolution of the particle 4-velocities.

We will use the geodesic equation $0 = v^\mu{}_{;\nu}v^\nu$ to obtain evolution equations for the v_n and v_z components of the particle's 4-velocity.

The computation are simple but do entail a few steps. We begin by writing dv_n/dt as a directional derivative along $t^\mu = \lambda v^\mu$. The Leibniz rule is then applied which in turn allows the geodesic equations $0 = v^\mu{}_{;\nu}v^\nu$ to be imposed. Finally, we use

$$Nn_{\mu;\nu} = -\bot N_{,\mu}n_\nu - NK_{\mu\nu} \tag{15}$$

to rewrite $n_{\mu;\nu}$ in terms of the lapse and extrinsic curvatures. The details are as follows.

$$\frac{dv_n}{dt} = v_{n;\nu}t^\nu = -(v^\mu n_\mu)_{;\nu}(\lambda v^\nu) = -\lambda v^\mu v^\nu n_{\mu;\nu}$$

$$= \lambda v^\mu v^\nu\left(\frac{1}{N}(\bot N_{,\mu})n_\nu + K_{\mu\nu}\right)$$

$$= -v_z v_n\frac{\lambda}{N}N_{,z} + \frac{1}{\lambda}\gamma^\mu\gamma^\nu K_{\mu\nu}.$$

But we also know that $\lambda = N/v_n$ and $0 = \gamma^x = \gamma^y$ while $\gamma^z = Nv_z/v_n$ so this last equation may be further reduced to just

$$\frac{dv_n}{dt} = -v_z N_{,z} + N\frac{v_z^2}{v_n}\mathcal{K}_{zz}. \tag{16}$$

The computations for dv_z/dt are much the same,

$$\frac{dv_z}{dt} = v_{z;\nu}t^\nu = (v^\mu m_{z\mu})_{;\nu}(\lambda t^\nu) = \lambda v^\mu v^\nu m_{z\mu;\nu}$$

$$= \lambda(v_n n^\mu + v_z m_z^\mu)v^\nu m_{z\mu;\nu} = \lambda v_n n^\mu v^\nu m_{z\mu;\nu}.$$

The term $n^\mu v^\nu m_{z\mu;\nu}$ can be computed by expanding $0 = (m_{z\mu}n^\mu)_{;\nu}v^\nu$ and then using (15) to obtain

$$0 = m_{z\mu;\nu}n^\mu v^\nu + \frac{v_n}{N}N_{,z} - K_{\mu\nu}m_z^\mu m_z^\nu$$

which, when substituted into the previous equation for dv_z/dt, leads to

$$\frac{dv_z}{dt} = -v_n N_{,z} + Nv_z\mathcal{K}_{zz}. \tag{17}$$

One simple check we can immediately apply to our equations is to ask: do they preserve the unit normalization of v^μ? Since we have chosen n^μ and m_z^μ to be unit vectors the question reduces to asking if $d(-v_n^2 + v_z^2)/dt$ vanishes for all t. From the above equations this is easily seen to be so.

VII. THE DENSITY

There are at least two ways to compute the density, either by solving the Hamiltonian constraint or by integrating the equations of motion for the dust, namely, $0 = (\rho v^\mu v^\nu)_{;\nu}$.

Recall that the Hamiltonian constraint is given by

$$\mathcal{R}_{xyxy} + 2\mathcal{R}_{xzxz} + \mathcal{K}_{xx}^2 + 2\mathcal{K}_{xx}\mathcal{K}_{zz} = 8\pi k\rho v_n^2. \tag{18}$$

This equation is trivial to solve for ρ since on each Cauchy surface all of the other quantities are known. Notice that $v_n^2 = 1 + v_z^2$ and thus $v_n > 0$.

Using the Hamiltonian constraint is one of many tricks used in numerical relativity to coerce better stability properties from the evolution equations. The merits of doing so have been debated over the years and is not something we will delve into here. However as we are trying to establish the limitations of the smooth lattice method it makes sense to explore other methods to compute the density. So for our second method we turn to the energy-momentum equations. From $0 = (\rho v^\mu v^\nu)_{;\nu}$ we learn two things (i) the dust particles follow time like geodesics, $0 = v^\mu{}_{;\nu}v^\nu$ and (ii) the rest mass is conserved along the worldtube generated by the dust particles $0 = d/dt\int\rho dV'$ where d/dt is the time derivative following the dust and dV' is the proper volume in the dust's rest frame. We will need both equations to compute the density.

Recall that we have chosen to tie the dust particles to the nodes of the lattice. As the nodes drift relative to the Cauchy surface there will be a nonzero boost between the rest frame of the dust and that of the Cauchy surface. Thus, in terms of the volume element dV on the Cauchy surface we have

$$\int_{C_0}\rho v_n dV = \int_{C_1}\rho v_n dV,$$

where C_0, C_1 denote the intersections of a dust worldtube with a pair of Cauchy surfaces, one at time t_0 and another at a later time t_1.

The question which arises now is: how do we construct the three-dimensional cross-sections C_0 from the two-

dimensional lattice? The solution is depicted in Fig. 6 where we have simply taken the original lattice and rotated it by $\pi/2$ about the central geodesic G_1. This creates C_0 and C_1 as truncated pyramids with a square cross section. In each of these we take the density to be constant. The volume of C_0 and C_1 can be computed by elementary Euclidean geometry (the dust is minimally coupled to the geometry and thus curvature corrections can be ignored). This leads to

$$(V)_i = \frac{1}{3}(L_{zz})_i((L_{xx}^2)_i + (L_{xx})_i(L_{xx})_{i+1}(L_{xx}^2)_{i+1}),$$

where $(L_{xx})_i$ and $(L_{xx})_{i+1}$ are the values of L_{xx} at nodes i and $i+1$ respectively. The previous conservation equation can now be rewritten as

$$3m_i = (\rho v_n)_i(L_{zz})_i((L_{xx}^2)_i + (L_{xx})_i(L_{xx})_{i+1} + (L_{xx}^2)_{i+1}),$$
(19)

where m_i is the conserved rest mass along the worldtube (m_i is set as part of the initial conditions). The v_n are estimated at the center of each cell by quadratic interpolation from the neighboring nodes (which will draw in nodes beyond this basic cell). This equation can then be solved for ρ. We assign that ρ to the center of the cell and then use quadratic interpolation to estimate ρ at the lattice nodes.

VIII. MAXIMAL SLICING

A maximally sliced spacetime is defined to be a spacetime for which $K = 0$ everywhere. Such spacetimes are often constructed by first setting $K = 0$ on an initial Cauchy surface (e.g. on a time-symmetric initial slice) and then demanding that $dK/dt = 0$ throughout the evolution. For our lattice we have $K = 2\mathcal{K}_{xx} + \mathcal{K}_{zz}$ and thus from the Eqs. (11) and (12) we see that $dK/dt = 0$ provided

$$0 = 2N_{,xx} + N_{,zz} - N(R - 4\pi k(1 + 2v_n^2)\rho).$$

But in Paper 1 we showed that under spherical symmetry

$$N_{,xx} = \frac{1}{L_{xx}}\frac{dL_{xx}}{dz}N_{,z},$$

which allows us to rewrite the previous equation as

$$0 = N_{,zz} + \frac{2}{L_{xx}}\frac{dL_{xx}}{dz}N_{,z} - N(R - 4\pi k(1 + 2v_n^2)\rho).$$
(20)

We treat this as an ordinary differential equation for N. The boundary conditions are simple, at $z = 0$ we require $dN/dz = 0$ while at the outer boundary we require $1 = \lim_{z\to\infty}N$. Note also that the differential equation is singular at $z = 0$ (due to the $1/L_{xx}$ term). We deal with this by appealing to the spherical symmetry of the solution at $z = 0$ to deduce that $N_{,xx} = N_{,zz}$ and thus our original differential equation for N can be rewritten as

$$0 = 3N_{,zz} - N(R - 4\pi k(1 + 2v_n^2)\rho) \quad \text{at } z = 0, \quad (21)$$

which is clearly nonsingular. The same result can also be obtained by applying l'Hôpital's rule to $(1/L_{xx})(dN/dz)$ as $z \to 0$. At the junction we know that ρ and R suffer a jump discontinuity. Thus we expect a corresponding jump discontinuity in d^2N/dz^2 which in turn forces both N and dN/dz to be continuous across the junction. This adds extra constraints to the numerical solution of the above equation. We will cover this in more detail in Sec. X B but for the moment we note that our method computes two separate solutions, one for either side of the junction, which are then matched at the junction.

IX. THE JUNCTION CONDITIONS

Darmois [18] and later Israel [19] developed a very elegant approach to handle discontinuities in a metric in general relativity. However, their method requires some work to push through so we defer the details to Appendix A preferring instead to present here a direct approach.

By integrating the geodesic deviation Eq. (1) over a short interval $z \in (-\epsilon, +\epsilon)$ we obtain

$$0 = \left[\frac{dL_{xx}}{dz}\right]_{-\epsilon}^{+\epsilon} + \int_{-\epsilon}^{+\epsilon} \mathcal{R}_{xzxz}L_{xx}dz.$$

If we require $\mathcal{R}_{xzxz}$ to be bounded on each Cauchy surface then we must have

$$0 = \lim_{\epsilon\to 0}\left[\frac{dL_{xx}}{dz}\right]_{-\epsilon}^{+\epsilon}$$

and thus dL_{xx}/dz is continuous everywhere on the lattice and, most importantly, across the junction. We also know that L_{xx} and dL_{xx}/dt must be continuous and thus from the evolution Eq. (9) we see that $0 = \lim_{\epsilon\to 0}[\mathcal{K}_{xx}]_{-\epsilon}^{+\epsilon}$. From here on we shall dispense with the limits on the square brackets and take $[\ldots]$ to mean $\lim_{\epsilon\to 0}[\cdots]_{-\epsilon}^{+\epsilon}$.

Applying a similar integration to the Bianchi identity leads to

$$0 = [L_{xx}^2\mathcal{R}_{xyxy}] - \lim_{\epsilon\to 0}\int_{-\epsilon}^{+\epsilon} \mathcal{R}_{xzxz}\frac{dL_{xx}^2}{dz}dz$$

and thus

$$0 = [\mathcal{R}_{xyxy}]. \quad (22)$$

since L_{xx} must be continuous every where on the lattice. Thus we conclude that $\mathcal{R}_{xyxy}$ is continuous on the lattice. However, by inspection of the Hamiltonian constraint (13), we see that the same can not be said for $\mathcal{R}_{xzxz}$. Since we know that $0 = [L_{xx}]$ and $0 = [\mathcal{K}_{xx}]$ we see that continuity of the Hamiltonian requires

$$[R_{xzxz}] = [4\pi k\rho v_n^2 - \mathcal{K}_{xx}\mathcal{K}_{zz}]. \quad (23)$$

We also need suitable junction conditions for the lapse function when using maximal slicing. First we demand

that the clocks of a pair of observers travelling close to but on opposing sides of the junction should remain synchronized throughout their journey. Thus we find that the lapse is continuous across the junction, $0 = [N]$. For the first derivative we follow the method outlined above. Integrating the maximal slicing Eq. (20) over the short interval $z \in (-\epsilon, +\epsilon)$ leads to

$$0 = [N_{,z}] + \lim_{\epsilon \to 0} \int_{-\epsilon}^{+\epsilon} \left(\frac{2}{L_{xx}} \frac{dL_{xx}}{dz} N_{,z} - N(R - 4\pi k(1 + 2v_n^2)\rho) \right) dz$$

and as we except all terms in the integral to be bounded (at worst) and $L_{xx} > 0$ we see that this requires

$$0 = [N_{,z}]. \tag{24}$$

Eqs. (22)–(24) constitute the full set of junction conditions for our lattice. Other conditions such as $0 = [L_{xx}]$ and $0 = [N]$ are trivially implemented in the numerical code (they require no special care). However we have no freedom in our data to guarantee $0 = [dL_{xx}/dz]$. The reason is that all of the L_{xx} leg lengths are subject to the evolution equations and we have to live with what they dictate. Of course we expect the jump in dL_{xx}/dz to be small and to vanish as the lattice is progressively refined.

X. NUMERICAL METHODS

To obtain numerical solutions of our equations we turn once again to the techniques developed in Paper 1. We use second-order accurate finite differences (on a nonuniform grid) for all of the spatial derivatives, such as dL_{xx}/dz and d^2N/dz^2 (though with a two exceptions, as noted below in Sec. X D, for the three nodes centered on the junction). The time integration employs a standard 4th-order Runge-Kutta method and the time step is chosen so that the Courant factor for the smallest L_{zz} on the lattice is $1/2$ (the leg on which this occurs lies on the surface of the dustball).

The lattice and its attendant equations in this paper differ most notably from those of Paper 1 by the presence of the dustball. This not only introduces new terms in the equations but it also forces many of the variables, or their derivatives, to be discontinuous at the junction. Dealing with these discontinuities requires some care. For the geodesic deviation Eq. (1), the Bianchi identity (2) and the maximal lapse Eq. (20), the general approach is to solve those equations twice, once on either side of the junction, and then use the junction conditions to match the solutions. The details are as follows.

A. The Riemann curvatures

The discretized forms of the geodesic deviation Eq. (1) and the Bianchi identity (2) were given in Paper 1 and, apart form some minor notational changes, are equivalent to the following pair of equations

$$(L_{xx})_i (\mathcal{R}_{xzxz})_i = -\left(\frac{d^2 L_{xx}}{dz^2} \right)_i \tag{25}$$

$$2(L_{xx}^2)_i (\mathcal{R}_{xyxy})_i = (L_{xx}^2)_i ((\mathcal{R}_{xzxz})_i + (\mathcal{R}_{xzxz})_{i-1}) + (L_{xx}^2)_{i-1} (2(\mathcal{R}_{xyxy})_{i-1} - (\mathcal{R}_{xzxz})_i - (\mathcal{R}_{xzxz})_{i-1}). \tag{26}$$

where the second derivatives of L_{xx} are computed using the second-order nonuniform finite differences (as described in Sec. X D).

Our plan is to use this pair of equations to calculate the Riemann curvatures on the lattice but we immediately encounter two problems, the equations are singular at $z = 0$ and, as previously noted, the second derivatives of L_{xx} are not continuous across the junction. The first problem is rather easy to deal with. We draw upon the required spherical symmetry at $z = 0$ to deduce that $d\mathcal{R}_{xzxz}/dz = 0$ at $z = 0$ and thus $\mathcal{R}_{xzxz}(z) = A + Bz^2 + z^3$ near $z = 0$. The coefficients A and B are obtained by fitting $\mathcal{R}_{xzxz}(z) = A + Bz^2$ to two samples for $(\mathcal{R}_{xzxz})_i$ (typically $(\mathcal{R}_{xzxz})_4$ and $(\mathcal{R}_{xzxz})_8$ for $n_J = 120$) and then setting $(\mathcal{R}_{xzxz})_i = A + Bz_i^2$ for each node near $z = 0$ (i.e. at $z = 0$, z_1, z_2 and z_3). For $\mathcal{R}_{xzxz}$ we again call on the spherical symmetry to assert that $(\mathcal{R}_{xyxy})_0 = (\mathcal{R}_{xzxz})_0$. Our numerical experiments show that we have no need to use the quadratic interpolation scheme for $(\mathcal{R}_{xyxy})$ near $z = 0$.

We turn now to the issue of the junction. As with the lapse function, we compute both Riemann curvatures separately on each side of the junction. We first use the above equations to compute the curvatures for all of the interior lattice nodes excluding the node at the junction. At the junction we apply a series of interpolations in conjunction with the boundary conditions to set the curvatures on the junction and one node point outside it. The details are as follows.

First we use cubic extrapolation to compute the one-sided limits $\lim_{z \uparrow z_J} \mathcal{R}_{xyxy}$ and $\lim_{z \uparrow z_J} \mathcal{R}_{xzxz}$, which we abbreviate as $(\mathcal{R}_{xyxy})^-$ and $(\mathcal{R}_{xzxz})^-$. We then use the junction condition (22) and the Hamiltonian constraint (13), which we rewrite as

$$(\mathcal{R}_{xyxy})^+ = (\mathcal{R}_{xyxy})^- \tag{27}$$

$$(\mathcal{R}_{xzxz})^+ = -\frac{1}{2}(\mathcal{R}_{xyxy} + \mathcal{K}_{xx}^2 + \mathcal{K}_{xx}\mathcal{K}_{zz})^+. \tag{28}$$

to step across the junction (with the $+$ superscript denoting the right hand one-sided limit). We then return to the above discrete Eqs. (25) and (26) to compute the curvatures in the exterior region. This too requires some explanation. We first compute $(\mathcal{R}_{xzxz})_i$ from $i = n_J + 2$ to $i = n_\infty - 1$ (i.e. we skip the first exterior node and stop one node in from the outer boundary). We then return to the node we skipped over (i.e. $i = n_J + 1$) and use cubic interpolation (using the nodes n_J, $n_J + 2$, $n_J + 3$ and $n_J + 4$) to estimate

$(\mathcal{R}_{xzxz})$ at that node. The Bianchi identity can then be applied to all the exterior nodes (except the node on the outer boundary). Finally, we use cubic extrapolation to compute the curvatures on the boundary nodes. This completes the computation of the curvatures.

B. Maximal slicing

The discrete form of the maximal lapse Eq. (20) is of the form

$$0 = a_i(N)_{i+1} + b_i(N)_i + c_i(N)_{i-1} \tag{29}$$

for some set of coefficients a_i, b_i and c_i (see Appendix B for the details). We wish to solve this set of equations subject to the following conditions

$$0 = N_{,z} \quad \text{at } z = 0, \tag{30}$$

$$0 = [N] \quad \text{at } z = z_J, \tag{31}$$

$$0 = [N_{,z}] \quad \text{at } z = z_J, \tag{32}$$

$$1 = \lim_{z \to \infty} N. \tag{33}$$

By reflection symmetry at $z = 0$ we can easily extend the lattice to $z < 0$. Thus a discrete version of (30) would be $N_{-1} = N_{+1}$. Continuity at $z = z_J$ allows us to use one value of N at z_J, which we denote by N_J. However, the continuity of $N_{,z}$ is not something we can prescribe but must be obtained by an iterative process (to be described below). We denote the left and right hand limits for $N_{,z}$ at $z = z_J$ by $(N_{,z})^-$ and $(N_{,z})^+$ respectively. We compute these one-sided limits, for a given set of N_i, by a cubic extrapolation of $(N_{,z})_i$. This too requires some explanation. We start with the four nodes nearest to but excluding the junction. We then use cubic extrapolation of the $(N)_i$ on these nodes to extend the $(N)_i$ to the junction and two nodes beyond (we store this generated data in a separate array so as not to overwrite the data already defined on those nodes). Finally we use the standard nonuniform second-order finite differences to estimate the first derivative at the junction. This computation is done twice, once for each one-sided limit. For the outer boundary we simply set $1 = N_\infty$.

The discrete equations for $(N)_i$ are solved in three iterations with each iteration involving separate solutions for $(N)_i$ on each side of the junction. The algorithm requires two guesses for N, one for N at $z = 0$ and one for N at $z = z_J$ which we denote by $(GN)_0$ and $(GN)_J$ respectively. With given values for these guesses we use a Thomas algorithm to solve the tridiagonal system (29) for $0 \leq z \leq z_J$ and again for $z_J \leq z \leq z_\infty$. Our guesses are unlikely to be correct (at first) so we record the errors in the boundary conditions by $E_0 = (N)_{+1} - (N)_{-1}$ and $E_J = (N_{,z})^+ - (N_{,z})^-$. Our aim is to choose the two guesses so that $0 = E_0$ and $0 = E_J$. We chose three pairs of guesses (0,0), (0, 1/2) and (1,1) for $((GN)_0, (GN)_J)$ and

we recorded the corresponding errors as $E_0^{(j)}$ and $E_J^{(j)}$ for $j = 1, 2, 3$. Since the discrete equations are linear and homogeneous in $(N)_i$ we can form a linear combination such as

$$N_i = \alpha_1 N_i^{(1)} + \alpha_2 N_i^{(2)} + \alpha_3 N_i^{(3)} \tag{34}$$

to satisfy the boundary and junction by an appropriate choice of constants α_1, α_2 and α_3. The result is a 3 by 3 system of equations

$$1 = N_\infty = \alpha_1 + \alpha_2 + \alpha_3$$

$$0 = E_0 = \alpha_1 E_0^{(1)} + \alpha_2 E_0^{(2)} + \alpha_3 E_0^{(3)}$$

$$0 = E_J = \alpha_1 E_J^{(1)} + \alpha_2 E_J^{(2)} + \alpha_3 E_J^{(3)}$$

which is easily solved for the three weights α_i which in turn allows the final (correct) solution for the maximal lapse to be computed from (34).

C. The time derivatives

Spatial derivatives are calculated at each node using data from the surrounding nodes, and in cases where this might draw in data from across the junction, we first use cubic extrapolation to extend the data across the junction (which we store separately so as not to overwrite exiting data).

With the exception of the junction node there is no ambiguity in applying the evolution equations to the nodes of the lattice. However, the discontinuities at the junction demand, once again, that we tread carefully near and at the junction. Consider $\mathcal{K}_{zz}$ which in geodesic slicing will be multiple valued at the junction. How do we handle this situation? We have already exhausted our supply of junction conditions in forming the two jump conditions (27) and (28) for the Riemann curvatures. So in the absence of any further information about $\mathcal{K}_{xx}$ we have no choice but to consider its left and right hand limits as independent of each other (despite the loose coupling afforded by the evolution equations). Each term could be evolved by evaluating time derivatives built from one-sided limits of the source terms. There is however an easier approach which we found to work quite well. The idea is to reinterpret the junction node not as node on which to apply the evolution equations but rather as a convenient staging post to impose the junction conditions. In this view we do *not* evolve the data on the junction node. Rather we treat that data as kinematical which we compute by one-sided extrapolations of the surrounding data (which are evolved via the normal evolution equations).

So in our code we use (9), (11), (12), (16), and (17) (subject to a minor change noted below) to evolve L_{xx}, $\mathcal{K}_{xx}$, $\mathcal{K}_{zz}$, v_n and v_z on the nodes $i = 0, 1, 2, \cdots n_J - 1$ and $i = n_J + 1, n_J + 2, n_J + 3, \cdots n_\infty - 1$. We use (10) (again, see below) to evolve the L_{zz} for all legs not connected to the junction. For the two legs attached to junction we use one-sided cubic extrapolation of dL_{zz}/dt to com-

pute their time derivatives. At the outer boundary we impose static boundary conditions for all of the data.

There is one exception to this simple algorithm. We use a one-sided extrapolation to set dL_{zz}/dt at the node $n_J - 1$. This proved to be essential for long-term stability with maximal slicing (but made no difference in geodesic slicing). We can offer no reasonable explanation as to why this works other than the following admittedly vague rationalization. By extrapolating the time derivatives outwards from the interior of the dustball to the node $n_J - 1$ we might be halting or minimizing the inward propagation of any errors that arise at the junction. Delving deeper into this mystery is best left for another time.

There is one remaining subtlety that we must address. The careful reader may have noticed that in the present context we are treating the $\mathcal{K}_{xx}$ and $\mathcal{K}_{zz}$ as being defined on the nodes whereas the extrinsic curvatures arose in Sec. IV by approximating the integrals by a midpoint rule. Thus if we wish to use node-based values for $\mathcal{K}_{xx}$ and $\mathcal{K}_{zz}$ we should use a Trapezoidal rule to estimate the integrals. This is a minor change and leads to the following node-based equations

$$\frac{dL_{xx}}{dt} = -\langle N\mathcal{K}_{xx}\rangle L_{xx} + \left\langle\frac{Nv_z}{v_n}\right\rangle\frac{dL_{xx}}{dz}, \qquad (35)$$

$$\frac{dL_{zz}}{dt} = -\langle N\mathcal{K}_{zz}\rangle L_{zz} + \left[\frac{Nv_z}{v_n}\right], \qquad (36)$$

where the angle-brackets denotes an average of that quantity over the leg while the square-brackets continues to denote the change across a leg. In fact for the L_{xx} equation the angle-brackets are redundant (the end points carry identical values) but were retained simply for emphasis. Since the Riemann curvatures are already node-based we see that no such averaging is required for the extrinsic curvature Eqs. (25) and (26). Note also that the spatial derivatives are also node based (by suitable choice of the finite difference operators).

D. The spatial derivatives

The evolution Eqs. (9)–(12) and the momentum constraint (14) require spatial derivatives of the L_{xx}, L_{zz}, $\mathcal{K}_{xx}$ and $\mathcal{K}_{zz}$. For all but the two nodes either side of the junction (i.e. at nodes $n_J - 1$ and $n_J + 1$), and the junction itself, we employ second-order nonuniform spatial derivatives as described in Paper 1. On the two nodes either side of the junction we use one-sided quadratic extrapolation. This is the only point in the code where we used quadratic approximations and we do so because both linear and (interestingly) cubic interpolation lead to instabilities forming at the junction (at around $t \approx 13$ for cubic extrapolation and only for one of our models with $n_J = 240$ and $n_\infty = 1200$). The derivatives at the junction are computed last using one-sided cubic extrapolation.

The only other spatial derivatives that need to be computed are the first and second derivatives of the lapse function [for use in the maximal slicing Eq. (20) and in the particle Eqs. (16) and (17)]. Once again we use the second-order nonuniform spatial derivatives from Paper 1 for all of the nodes with the exception of the five nodes centered on the junction. For nodes $n_J \pm 2$ and $n_J \pm 1$ we use cubic extrapolation to build an extended set data. This introduces some temporary and artificial nodes which we chose to be symmetric to the real nodes (e.g. when extending the data for node $n_J - 1$ we create new nodes, $n_J, n_J + 1$ that are the mirror images (in $n_J - 1$) of $n_J - 2$ and $n_J - 3$). The derivatives on nodes $n_J \pm 2$ and $n_J \pm 1$ are then computed on this extended data set using the standard nonuniform centered differences while the derivatives on the junction are computed using one-sided cubic extrapolation.

Once the maximal slicing equation has been solved we do have the option of using that equation as an alternative way to calculate the second derivatives of the lapse. We chose not to do so because we did not want to give the smooth lattice method a helping hand—we want to test the method under conditions closer (albeit in $1 + 1$ form) to what we would expect for other spacetimes (i.e. for a true $3 + 1$ evolution).

E. The initial data

We require two things of our initial data, first they must satisfy the constraints (13) and (14), and second they must describe a time-symmetric initial slice. This last condition is readily satisfied upon setting $\mathcal{K}_{xx} = 0$, $\mathcal{K}_{zz} = 0$ and $v_n = 1$, $v_z = 0$ which in turn ensures that the momentum constraint is also satisfied. What we are left with is the Hamiltonian constraint, the leg lengths, L_{xx}, L_{zz} and the density ρ, i.e. we have one constraint for three (sets) of data. Clearly there are a range of options here, so what should we do? We turn once again to the scheme developed in Paper 1. There we chose to set the L_{zz} and then use the Hamiltonian constraint to set the L_{xx}. But here we also need the density.

Keep in mind that our aim is neither to discover nor explore the Oppenheimer-Snyder solution but rather to use it as a test of the smooth lattice method. Thus it is not unreasonable to borrow some information from the exact solution to set some of the data on the lattice, in particular, the density. We recall here some basic equations from the exact solutions for the Oppenheimer-Snyder spacetime (see [9,20–22]).

There are two free parameters in the solution, the ADM mass m and the Schwarzschild areal radius R_0 of the dustball. From these we can compute the proper radius of the dustball z_J, the FRW parameters a_m and $\chi_0 > 0$ and the density ρ using

$$\sin^2\chi_0 = \frac{2m}{R_0}, \qquad (37)$$

$$a_m = \frac{2m}{\sin^3\chi_0}, \qquad (38)$$

$$z_J = a_m \chi_0, \tag{39}$$

$$8\pi k \rho = \frac{3}{a_m^2}. \tag{40}$$

Clearly, we also have $\rho = 0$ in the Schwarzschild exterior.

We used these equations to set z_J and ρ, for a given m and R_0. To this we added choices for the total length of the lattice z_∞, the number of interior nodes n_J and the total number of nodes n_∞ on the lattice.

Note that we still have the freedom to distribute the nodes along the z-axis (this amounts to setting the $(L_{zz})_i$). We know that some of the spatial gradients are zero at $z = 0$, that they rise to a maximum near the junction and then settle down in the distant asymptotically flat regions of the lattice. Thus it makes sense to concentrate the nodes around the junction. With this in mind we chose to start at the junction and use a geometric progression to set the $(L_{zz})_i$ in both the interior and exterior regions. We chose the same geometric ratio in both regions while also requiring $(L_{zz})_{n_{J-1}} = (L_{zz})_{n_J}$. From here it is simple matter to compute all of the $(L_{zz})_i$ across the lattice.

We now turn to the problem of setting L_{xx} and the Riemann curvatures. By reworking the Hamiltonian constraint, geodesic deviation and Bianchi identity we find that across the lattice

$$(L_{xx})_i = (L_{xx})_{i-1} + \frac{(L_{zz})_{i-1}}{(L_{zz})_{i-2}}((L_{xx})_{i-1} - (L_{xx})_{i-2})$$
$$- \frac{1}{2}(L_{zz})_{i-1}((L_{zz})_{i-1} + (L_{zz})_{i-2})(L_{xx}R_{xzxz})_{i-1} \tag{41}$$

while the curvatures in the dustball are uniform and are given by

$$(\mathcal{R}_{xzxz})_i = (\mathcal{R}_{xyxy})_i = \frac{8\pi k \rho}{3}, \tag{42}$$

and finally, in the Schwarzschild region, we find

$$(\mathcal{R}_{xzxz})_i = (\mathcal{R}_{xzxz})_{i-1} \left(\frac{5(L_{xx}^2)_{i-1} - (L_{xx}^2)_i}{5(L_{xx}^2)_i - (L_{xx}^2)_{i-1}} \right) \tag{43}$$

$$(\mathcal{R}_{xyxy})_i = -2(\mathcal{R}_{xzxz})_i. \tag{44}$$

Note that the initial data is required (by assumption) to be homogeneous and isotropic inside the dustball. Thus it follows that the Riemann curvatures and the density are uniformly distributed throughout the dustball.

These equations can be used to set the L_{xx}, $\mathcal{R}_{xyxy}$ and $\mathcal{R}_{xzxz}$ across the lattice (a process that will require the junction conditions for the curvatures). But to start the ball rolling we must make some choice for $(L_{xx})_0$, and $(L_{xx})_1$. Clearly $(L_{xx})_0 = 0$ but for $(L_{xx})_1$ we are free to make any choice we like (we chose $(L_{xx})_1 = 0.001(L_{zz})_0$ so that $dL_{zz}/dz = 0.001$ at $z = 0$, as discussed below in Sec. XII).

F. Density

In Sec. VII we noted that the density can be computed using either the Hamiltonian constraint, in the form (18), or by the conservation Eq. (19). We find that, for long-term stability when using the second method, we are forced to use the Hamiltonian constraint at exactly the two nodes just inside the junction (i.e. at nodes $n_J - 2$ and $n_J - 1$). This was found by pure numerical experimentation. Why this should be so is unclear to us but it is probably tied to the same mechanism noted above (with regard to halting the inward propagation of errors from the junction by imposing "correct" values near the junction).

XI. DIAGNOSTICS

From the known solution for the Oppenheimer-Snyder spacetime a number of useful diagnostics can be drawn. Here we will discuss those diagnostics which, in the following section, we will apply to our numerical results.

For geodesic slicing it is rather easy to show [21] that the proper radius of the dustball z_J varies with proper time t according to

$$z_J(t) = \frac{a_m \chi_0}{2}(1 + \cos\eta(t)), \tag{45}$$

where $a_m = 2m/\sin^3 \chi_0$ and $\eta(t)$ is the solution of $0 = -2t + a_m(\eta + \sin\eta)$ with $\eta > 0$ (notice that Petrich *et al.* use η where we use $\eta - \pi$).

Another simple diagnostics arises from the central density which is given by

$$\rho(t) = 24a_m^2(1 + \cos\eta(t))^{-3}. \tag{46}$$

This is singular when $\eta = \pi$ at which point the proper radius is zero and the dustball has collapsed onto the singularity. This will occur after a proper time of

$$t_g^s = \pi m \left(\frac{R_0}{2m} \right)^{3/2} \tag{47}$$

and at this moment, or a short time before, we expect our code to crash.

As the dustball collapses an outer apparent horizon will form and this too provides useful checks on our numerics. It is known that when the outer most apparent horizon forms it does so at the surface of the dustball. In our numerical code we locate the horizon by noting where on the radial axis the quantity $dL_{xx}/dz - \mathcal{K}_{xx}L_{xx}$ vanishes. The root of this equation is the location of the apparent horizon (this follows from the condition that $0 = \partial A/\partial u$ where A is the area of a 2-sphere and $\partial/\partial u$ is the outward pointing null vector to the 2-sphere, see Paper 1 for more details). The time at which the horizon forms is also well known and this affords yet another check on our numerical results. For geodesic slicing it can be shown that the time, t_g^H, and location z_g^H, of the apparent horizon are given by

$$t_g^H = \frac{m}{\sin^3\chi_0}(\pi - 2\chi_0 + \sin(2\chi_0)) \qquad (48)$$

$$z_g^H = \frac{m}{\sin^3\chi_0}(1 - \cos(2\chi_0)). \qquad (49)$$

Note that in geodesic slicing the nodes are at rest relative to the Cauchy surfaces and thus this time t_g^H equals the proper time measured by the observer following that junction as it falls inwards and eventually meets the outward expanding event horizon. The quantity z_g^H measures the proper distance out from the center of the dustball to the junction.

Hawking's area theorem can also be used as a diagnostic. The theorem requires that the area of the event horizon should be constant once all of the dust has fallen within the event horizon. For our lattice this would require that the L_{xx} on the event horizon should be constant for the remainder of the evolution. This is easily checked (by interpolating the values of L_{xx} from the nodes onto the event horizon).

Equations for the time and location of the horizon, as well as the density and radius diagnostics, are also available for maximal slicing but with one drawback—the equations as given by Petrich *et al.* require a numerical integration of some elliptic integrals. This introduces its own set of numerical issues and we found that our implementation of the Petrich equations could only be reliably used for $t \lesssim 32$ (for $m = 1$ and $R_0 = 5$). Even so, this was sufficient time to allow for a useful comparison to be made.

We also have one extra diagnostic for the case of maximal slicing. There it is known that the lapse function will, after an initial period, settle into an exponential decay. Petrich *et al.* show that $N(t, 0) \sim A \exp(\beta t)$ where A is a constant and $\beta = -(2/3)^{(3/2)} \approx -0.5443311$. We can use this to test our code by measuring the slope of the $\log N$ versus t.

There are of course two other diagnostics—the Hamiltonian and momentum constraints.

In summary we have the following set of diagnostics.
 (i) The constraints.
 (ii) The history of the junction.
 (iii) The history of the central density.
 (iv) The crash time for geodesic slicing.
 (v) The Petrich solution for maximal slicing.
 (vi) The exponential collapse of the central lapse.
 (vii) The time and location of the first apparent horizon.
(viii) The constancy of the area of the event horizon in the vacuum region.

Clearly we have a raft of diagnostics and it is now time to turn to the actual results.

XII. RESULTS

Our aim was to write a code that used as few assumptions as needed to obtain reliable results. In the end we have split the computation of the lapse from the rest of the code. The evolution of the code takes as input (at each time step) the values of the lapse across the lattice. We do not use the Hamiltonian or momentum constraints apart from the two exceptions noted in Secs. X A and X F. We employ no artificial smoothing such as artificial viscosity nor do we add on any constraint preserving terms. Our time integrations are conducted using a 4th-order Runge-Kutta routine and our time step was updated after every time step by setting it equal to $1/2$ the shortest L_{zz} on the grid (which usually is the leg on or just inside the junction). This choice sets the Courant factor to $1/2$ for legs near the junction (with smaller values for legs away from the junction).

We set our initial data using $8\pi k = 1$, $m = 1$, $R_0 = 5$, $z_\infty = 400$ and $dL_{xx}/dz = 0.001$ at $z = 0$. We ran the code for three separate models, with $(n_J, n_\infty) = (60, 300)$, $(120, 600)$ and $(240, 1200)$ for both the geodesic and maximal slicing and one further model with $(n_J, n_\infty) = (240, 2400)$ for maximal slicing. The results for a selection of quantities are displayed in Figs. (4 and 7–16). The first point to note is that the results are well behaved with no apparent instabilities even through to very late in the evolution. The junction remains sharp without any noticeable smoothing
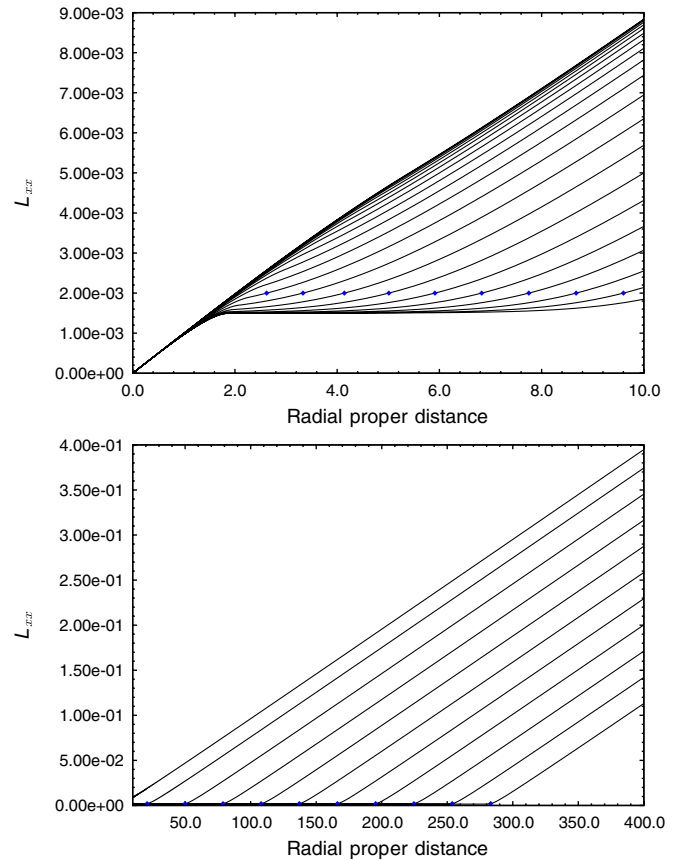

FIG. 4 (color online). The L_{xx} leg lengths for $t = 0$ to $t = 32$ in 20 steps (top, with $0 < z < 10$) and $t = 0$ to $t = 500$ in 10 steps (bottom, with $10 < z < 400$). This time the motion of the apparent horizon is much more noticeable than for the case of geodesic slicing.
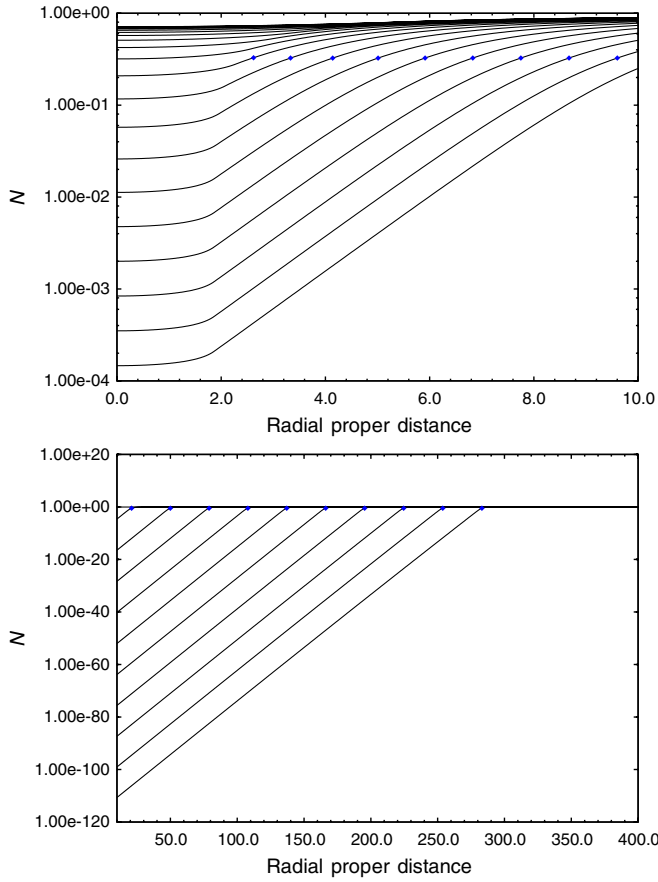
FIG. 5 (color online). A logarithmic plot of the lapse. The even gaps between the curves shows clearly that the collapse is exponential in time. Note the extreme value of the lapse at the origin for late times, of order 10^{-110}.

and the constraints, though not zero, do not show the exponential growth often associated with unstable evolutions.

We ran the geodesic code until it crashed at time $t^S = 12.41793$ which compares well with the exact time $t^S_g = 12.41824$ (note that the time step at the crash was 8.19×10^{-6} which is considerably smaller than the initial time step of 5.38×10^{-3}).
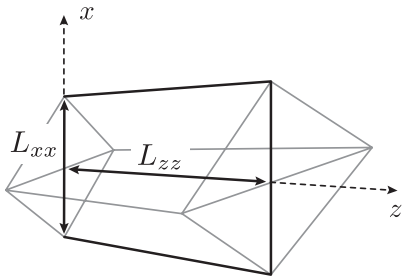
FIG. 6. Here we display the three-dimensional cell which we use to compute the energy density from the conservation Eq. (19).
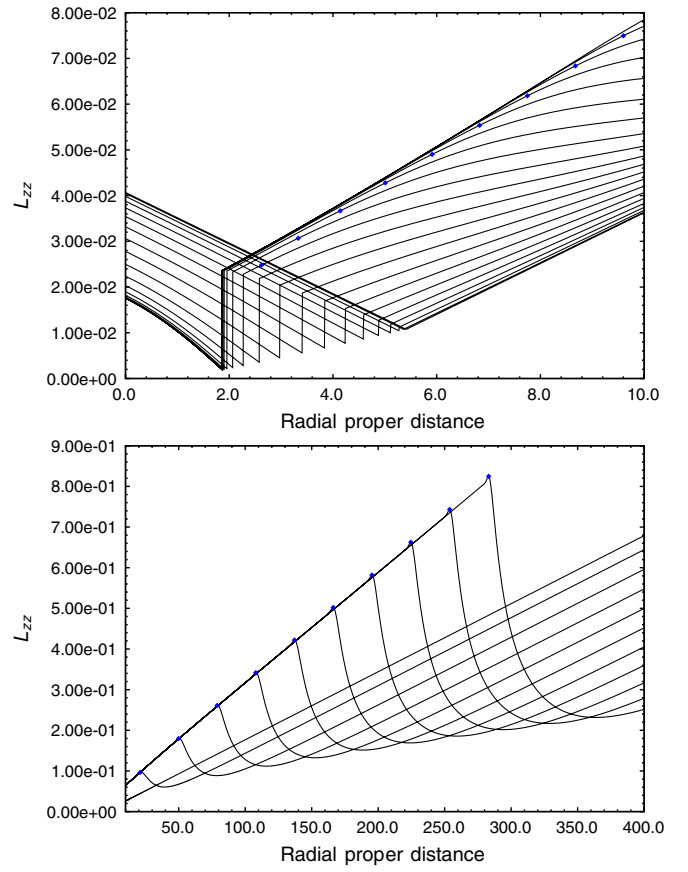
FIG. 7 (color online). The L_{zz} leg lengths for $t = 0$ to $t = 32$ in 20 steps (top, with $0 < z < 10$) and $t = 0$ to $t = 500$ in 10 steps (bottom, with $10 < z < 400$). Notice the extreme change in L_{zz} at the junction. The curves bunch together late in the evolution due to the exponential collapse of the lapse (see Fig. 5).

For geodesic slicing we found the apparent horizon formed at $t^H = 10.87837$ and $z^H = 2.16534$ while the exact values are $t^H_g = 10.87804$ and $z^H_g = 2.16527$. While for maximal slicing the numerical values were $t^H = 16.98238$, $z^H = 2.38015$ compared with the exact values $t^H_m = 17.02246$, $z^H_m = 2.37971$.

For maximal slicing and the collapse of the lapse diagnostic we estimated the slope over the interval $25.0 \leq t \leq 35.0$ and obtained $\beta = -0.54424$ compared with the exact value of -0.54433.

In Fig. 14 we have plotted the fractional errors in the radius and the central density, for maximal slicing, for the first three models (as described above). The errors do decrease with increasing number of nodes however it would appear that the errors are not converging to zero. The simple explanation is that we set $N = 1$ on a finite outer boundary and this clearly incurs an error. To test this we reran our code with different choices for the location of the outer boundary (while retaining the same number of nodes). This showed that the peaks in Fig. 14 varied
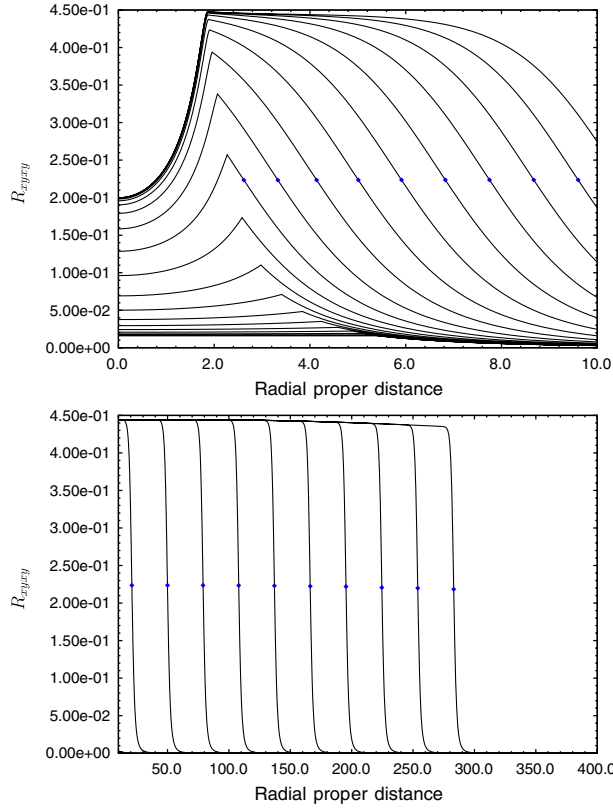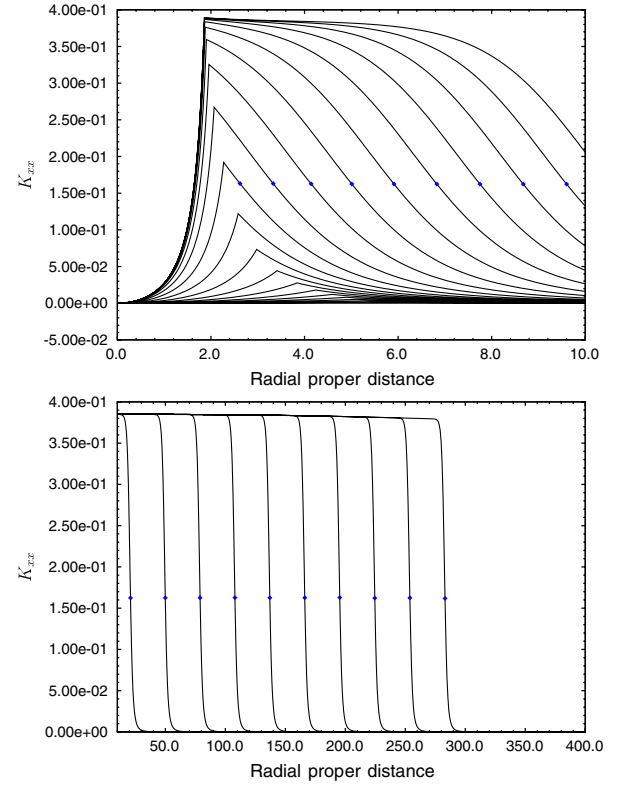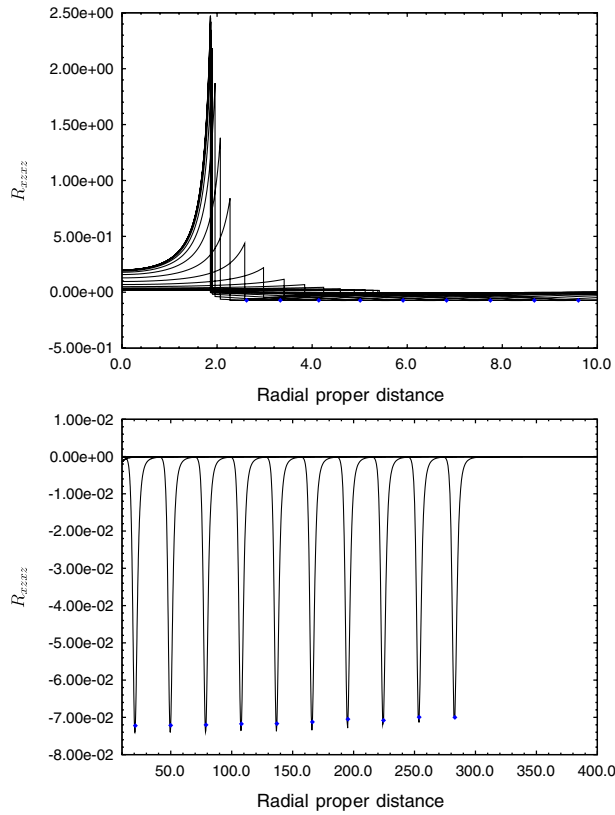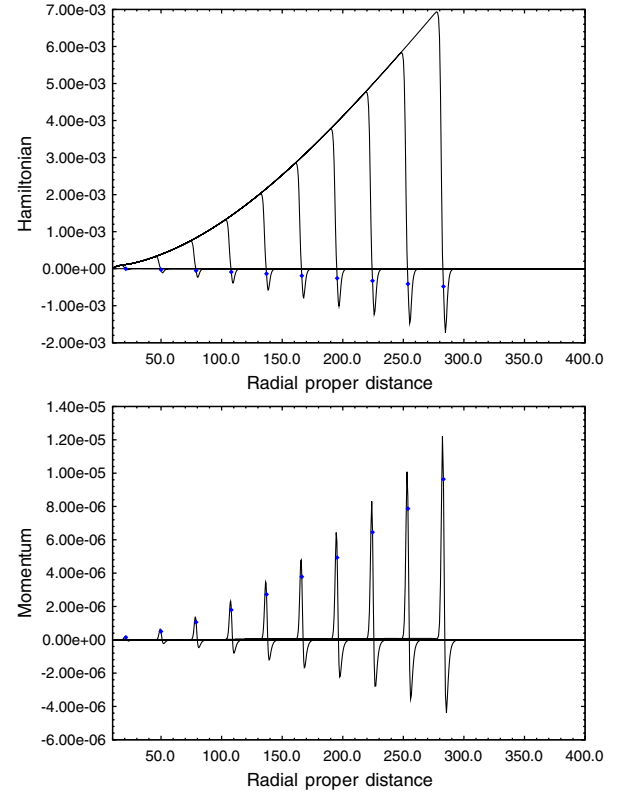
FIG. 8 (color online).   The Riemann curvature $\mathcal{R}_{xyxy}$.

FIG. 9 (color online).   The Riemann curvature $\mathcal{R}_{xzxz}$.

FIG. 10 (color online).   The extrinsic curvature $\mathcal{K}_{xx}$.

FIG. 11 (color online).   The Hamiltonian and momentum constraints. This shows a slowly growing error for later times.
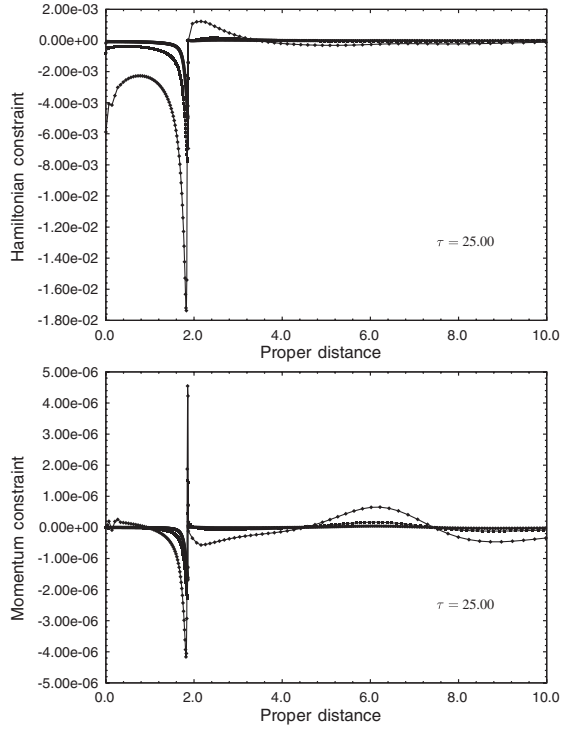
FIG. 12. The Hamiltonian and momentum constraints across the lattice at a fixed time with maximal slicing. The peak occurs near the junction and appears to vary as $1/n_\infty$.
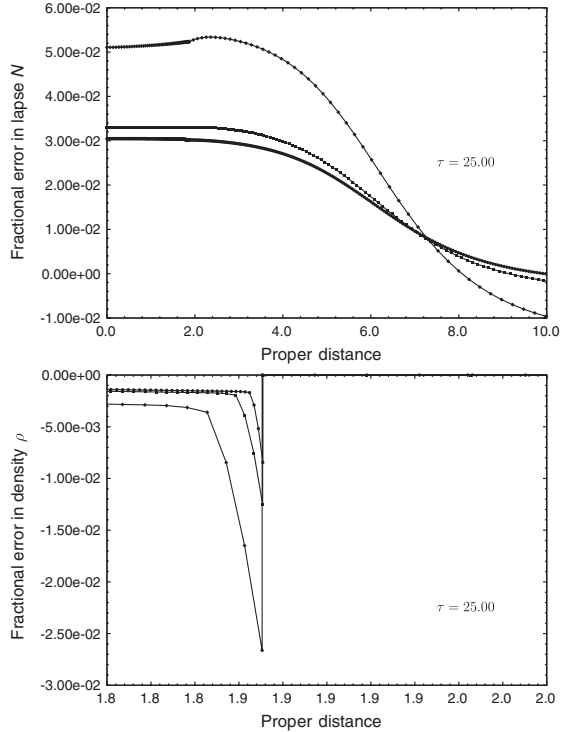

FIG. 13. The fractional errors for the lapse and the rest density across the lattice with maximal slicing. The large error in the coarsest lattice is probably due to having too few nodes. The finer lattice show much better errors but note that the lapse appears not to converge at the origin. This is due to the use of a finite outer boundary for the lapse.
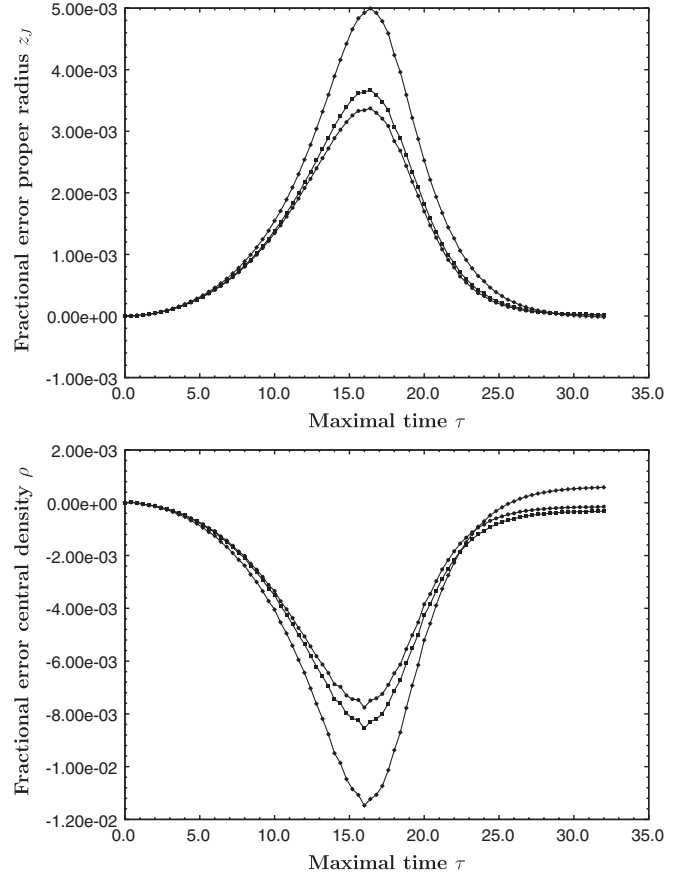

FIG. 14. The fractional errors for the radius and the central density for maximal slicing. The peak occurs around the time when the apparent horizon forms. The height of the peak for the finest resolution is limited by the location of the outer boundary. Doubling z_∞ halves the height of the peak.

inversely with the distance to the outer boundary z_∞. Incidentally, the broad peaks in those figures correspond to the formation of the apparent horizon.

For maximal slicing we have taken a snapshot of the numerical data at a fixed time, Fig. 13, to compare the density and the lapse with their exact values (from the Petrich code) across the lattice. Once again we see an initial convergence from coarse to fine resolutions but then the convergence appears to falter. This is also due to the use a of finite outer boundary, the peaks in the errors being proportional to $1/z_\infty$. Similar considerations apply to the snapshots of the Hamiltonian and momentum constraints, see Fig. 12.

The fractional changes in the horizon L_{xx} are shown in Fig. 16. This shows that for $t < 32$ the horizon area varied by no more than 5×10^{-2} percent for the coarsest model improving to less than 1×10^{-3} percent for the finest model. By $t \approx 500$ the error had grown to less than 2% for the finest model.

We also ran our code using the Hamiltonian constraint to set the density and found results very similar to those just given.
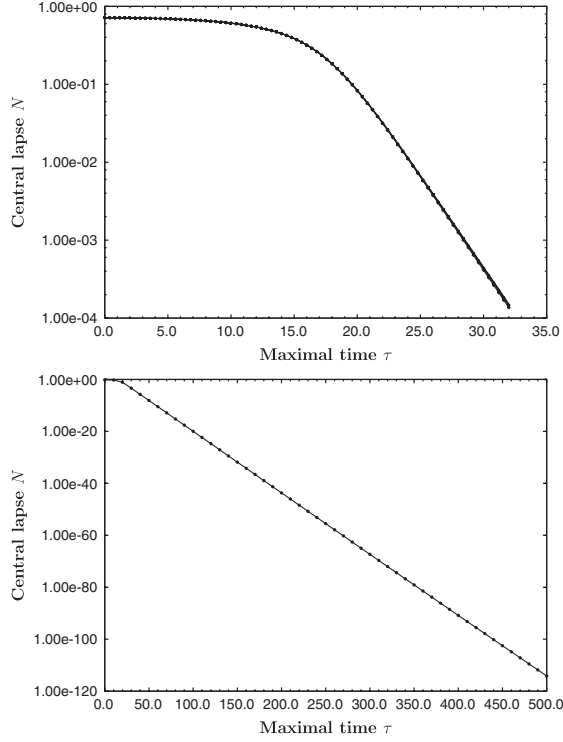
FIG. 15. The lapse at the origin for the three models superimposed on the exact data of Petrich *et al.* (top) and for the single long-term model (bottom). This shows clearly that the lapse collapses exponentially.
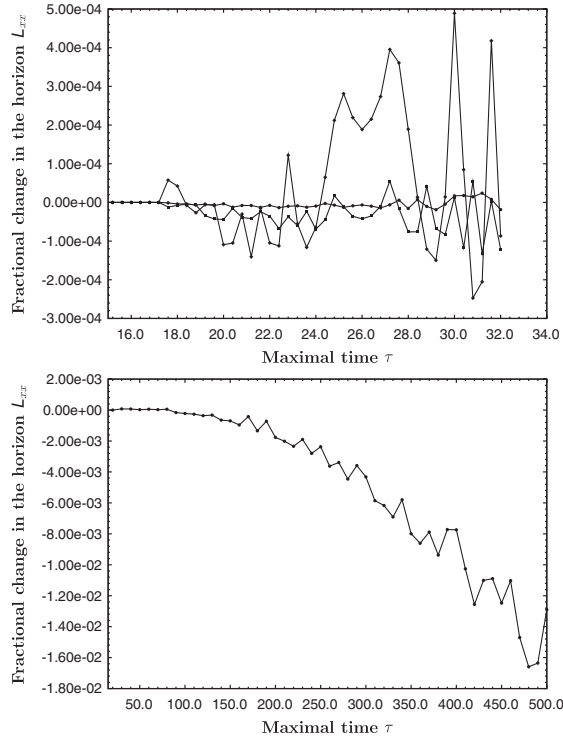

FIG. 16. The fractional change in the L_{xx} on the horizon for the three models (top) and for the long-term integration (bottom) in maximal slicing). These errors should be zero by Hawking area theorem.

XIII. DISCUSSION

The results just presented are very encouraging. They are consistent with our previous investigations of the smooth lattice method [6–8] yielding excellent results with only minor demands on computational resources. This gives us confidence that the method is viable but further tests are certainly required, in particular, an example in full $3 + 1$ dimensions, without symmetries, is imperative. This is a work in progress and we hope to report on this soon.

One striking feature of the results for maximal slicing which we have so far ignored is the wavelike behavior displayed in many of the plots (and similar behavior was also noted in Paper 1). This is certainly not a gravitational wave (the spacetime is spherically symmetric). Can this behavior be understood from the evolution equations? Without delving too far into the analysis we note that the first order Eqs. (9) and (11) can be recast as a single second-order equation for L_{xx}. This will involve $d^2 L_{xx}/dt^2$ and the Riemann curvatures. But in these late times, where the waves are apparent, we see that $|\mathcal{R}_{xzxz}| \ll |\mathcal{R}_{xyxy}|$ and thus the curvatures are dominated by $\mathcal{R}_{xyxy}$ which, through the geodesic deviation equation, (1), introduces $d^2 L_{xx}/dz^2$ into the second-order evolution equation for L_{xx}. Thus we have in the one equation the two key elements of the one-dimensional wave equation for L_{xx} and so wavelike behavior is not surprising. Of course this is a very loose argument and there are many more terms to contend with before it can be said that the wavelike behavior can be understood in standard terms. We will pursue this matter in a later paper.

APPENDIX A: THE DARMOIS-ISRAEL JUNCTION CONDITIONS

Consider a spacetime $(g, \mathcal{M})$ and let $\mathcal{S}$ be some three-dimensional time like surface in $\mathcal{M}$. This surface will divide $\mathcal{M}$ into two parts; one part, $\mathcal{M}^{\mathcal{L}}$, to the left of $\mathcal{S}$ and another part, $\mathcal{M}^{\mathcal{R}}$, to the right. In the absence of surface layers (e.g. infinitesimally thin shells of dust with nonzero energy) the Darmois-Israel junction conditions [18,19] ensure that g is a solution of Einstein's equations everywhere in $\mathcal{M}$ provided it is a solution in $\mathcal{M}/\mathcal{S}$, and most importantly, that the first and second fundamental forms on $\mathcal{S}$ are continuous across $\mathcal{S}$.

Suppose we denote the first and second fundamental forms on $\mathcal{S}$ by $\tilde{h}$ and $\tilde{K}$ respectively. Then each of these quantities can be calculated from the embedding of $\mathcal{S}$ in either $\mathcal{M}^{\mathcal{L}}$ or in $\mathcal{M}^{\mathcal{R}}$. The junction conditions requires that both computations yield identical results, that is $0 = [\tilde{h}]$ and $0 = [\tilde{K}]$.

In our case we take $(g, \mathcal{M})$ to be the Oppenheimer-Snyder spacetime and $\mathcal{S}$ to be the surface generated by the evolution of the surface of the dust. We will use a $\tilde{}$ symbol to denote quantities that live on $\mathcal{S}$, for example, $\tilde{h}$

and $\tilde{K}$ will represent the 3-metric and extrinsic curvatures, respectively, on $\mathcal{S}$. We extend this notation slightly to allow $\tilde{n}$ to be unit (space like) normal to $\mathcal{S}$ in $\mathcal{M}$.

Our first task will be to express the junction conditions in terms of data on Σ.

We know that L_{xx} lies in $\mathcal{S}$ and thus the junction condition $0 = [\tilde{h}]$ requires both $0 = [L_{xx}]$ and $0 = [dL_{xx}/dt]$ while $0 = [\tilde{K}]$ requires $0 = [dL_{xx}/dz]$ (note that d/dz is not normal to $\mathcal{S}$ but it can be resolved into pieces parallel and normal to $\mathcal{S}$ and the result follows). Looking back at the evolution Eq. (9) we see that this series of observations leads to the simple condition that $0 = [\mathcal{K}_{xx}]$. We will make use of this result in the following discussions on the Riemann curvatures. Consider the Gauss equation for $\mathcal{S}$, namely,

$$\tilde{\perp}(^4R_{\mu\alpha\nu\beta}) = \tilde{R}_{\mu\alpha\nu\beta} - \tilde{K}_{\mu\nu}\tilde{K}_{\alpha\beta} + \tilde{K}_{\mu\beta}\tilde{K}_{\alpha\nu},$$

where $\tilde{\perp}$ is the projection operator for $\mathcal{S}$ i.e. $\tilde{\perp}^{\mu}{}_{\nu} = \delta^{\mu}{}_{\nu} - \tilde{n}^{\mu}\tilde{n}_{\nu}$. Since the vectors m_x^{μ}, m_y^{μ} are both tangent to $\mathcal{S}$ and since $0 = [\tilde{K}_{\mu\nu}]$ we have

$$0 = [^4R_{\mu\alpha\nu\beta}m_x^{\mu}m_x^{\nu}m_y^{\alpha}m_y^{\beta}].$$

We can apply the Gauss equation once again, but this time for Σ rather than $\mathcal{S}$, that is

$$\perp(^4R_{\mu\alpha\nu\beta}) = R_{\mu\alpha\nu\beta} + K_{\mu\nu}K_{\alpha\beta} - K_{\mu\beta}K_{\alpha\nu}.$$

This leads to the simple equation

$$0 = [\mathcal{R}_{xyxy}], \tag{A1}$$

where we have used $0 = [\mathcal{K}_{xx}]$ and the fact that $K_{\mu\nu}$ is diagonal. This is one of our two junction conditions for the Riemann curvature. The second condition will apply to $\mathcal{R}_{xzxz}$ and as we shall soon see amounts to no more than requiring continuity of the Hamiltonian constraint across the junction (as we would expect).

We repeat the above procedure this time using the vectors m_x^{μ} and t^{μ} and after the first Gauss equation we find

$$0 = [^4R_{\mu\alpha\nu\beta}m_x^{\mu}m_y^{\nu}t^{\alpha}t^{\beta}].$$

Now t^{μ} is spanned by n^{μ} and m_z^{μ}, that is $t^{\mu} = v_n n^{\mu} + v_z m_z^{\mu}$, and thus we have

$$0 = [v_n^2 \perp(R_{\mu\alpha\nu\beta}n^{\mu}n^{\nu})m_z^{\alpha}m_z^{\beta}$$
$$+ 2v_n v_z \perp(R_{\mu\nu\alpha\beta}n^{\mu})m_z^{\nu}m_x^{\alpha}m_x^{\beta}$$
$$+ v_z^2 \perp(R_{\mu\nu\alpha\beta})m_z^{\mu}m_z^{\nu}m_x^{\alpha}m_x^{\beta}],$$

where we have also included the projection operator $\perp$ for Σ (since m_x^{μ} and m_z^{μ} are both tangent to Σ) in preparation for the second application of the Gauss equation. This time we will need the Gauss equation and its contractions with n^{μ}, that is

$$\perp(^4R_{\mu\alpha\nu\beta}) = R_{\mu\alpha\nu\beta} + K_{\mu\nu}K_{\alpha\beta} - K_{\mu\beta}K_{\alpha\nu}$$

$$\perp(^4R_{\mu\alpha\nu\beta}n^{\mu}) = K_{\alpha\beta|\nu} - K_{\alpha\nu|\beta}$$

$$\perp(^4R_{\mu\alpha\nu\beta}n^{\mu}n^{\nu}) = -\perp(^4R_{\alpha\beta}) + R_{\alpha\beta} + KK_{\alpha\beta} - K_{\alpha\mu}K^{\mu}{}_{\beta}.$$

Using the Einstein equations, $^4R_{\alpha\beta} = 8\pi k(T_{\alpha\beta} - (1/2) \times g_{\alpha\beta}T)$, the constraint equation $K_{|\mu} - K_{\mu}{}^{\nu}{}_{|\nu} = 8\pi k\perp \times (T_{\mu\nu}n^{\nu})$ and the diagonal character of $K_{\mu\nu}$ we find that

$$\perp(R_{\mu\nu\alpha\beta})m_z^{\mu}m_z^{\nu}m_x^{\alpha}m_x^{\beta} = \mathcal{R}_{xzxz} + \mathcal{K}_{xx}\mathcal{K}_{zz}$$

$$\perp(R_{\mu\nu\alpha\beta}n^{\mu})m_z^{\nu}m_x^{\alpha}m_x^{\beta} = -4\pi k\rho v_n v_m$$

$$\perp(R_{\mu\alpha\nu\beta}n^{\mu}n^{\nu})m_z^{\alpha}m_z^{\beta} = -4\pi k\rho + \mathcal{R}_{xyxy} + \mathcal{R}_{xzxz}$$
$$+ \mathcal{K}_{xx}\mathcal{K}_{zz} + \mathcal{K}_{xx}^2$$

and thus our junction condition can be reduced to

$$0 = \left[\frac{1}{2}\rho v_n^2 - \mathcal{R}_{xzxz} - \mathcal{K}_{xx}\mathcal{K}_{zz}\right], \tag{A2}$$

where we have used $v_z^2 = v_n^2 - 1$ to eliminate v_z. Looking back at our constraint Eqs. (13) we see that this last equation, along with $0 = [\mathcal{R}_{xyxy}]$ and $0 = [\mathcal{K}_{xx}]$, shows that the Hamiltonian constraint must be conserved across the junction (as expected).

APPENDIX B: THE MAXIMAL LAPSE EQUATION

Let $(N)_i$ be the node values of the lapse function across the lattice. Then using second order accurate finite differences (on a nonuniform grid) we obtain the following discrete equations

$$0 = a_i(N)_{i+1} + b_i(N)_i + c_i(N)_{i-1} \tag{B1}$$

with

$$a_i = \frac{2}{(L_{zz})_i}\left(\frac{(L_{zz})_{i-1}}{(L_{xx})_i}\left(\frac{dL_{xx}}{dz}\right)_i + 1\right) \tag{B2}$$

$$c_i = \frac{-2}{(L_{zz})_{i-1}}\left(\frac{(L_{zz})_i}{(L_{xx})_i}\left(\frac{dL_{xx}}{dz}\right)_i - 1\right) \tag{B3}$$

$$b_i = \frac{4(\bar{L}_{zz})_i}{(L_{zz})_{i-1}(L_{zz})_i}\left(\frac{(\Delta L_{zz})_i}{(L_{xx})_i}\left(\frac{dL_{xx}}{dz}\right)_i - 1\right)$$
$$- 2(\bar{L}_{zz})_i(R - \pi k\rho(8v_n^2 + 4))_i \tag{B4}$$

for $z > 0$ and

$$a_i = \frac{6}{(L_{zz})_i} \qquad c_i = \frac{6}{(L_{zz})_{i-1}} \tag{B5}$$

$$b_i = \frac{-4(\bar{L}_{zz})_i}{(L_{zz})_{i-1}(L_{zz})_i} - 2(\bar{L}_{zz})_i(R - \pi k\rho(8v_n^2 + 4))_i \tag{B6}$$

for $z = 0$.

In the above equations we have introduced $2(\bar{L}_{zz})_i = (L_{zz})_i + (L_{zz})_{i-1}$ and $(\Delta L_{zz})_i = (L_{zz})_i - (L_{zz})_{i-1}$.

[1] S. L. Shapiro and S. A. Teukolsky, in *Dynamical Spacetimes and Numerical Relativity*, edited by J. Centrella (CUP, Cambridge, 1986), pp. 74–100.

[2] F. Pretorius, Phys. Rev. Lett. **95**, 121101 (2005).

[3] F. Pretorius, arXiv:0710.1338v1.

[4] M. Campanelli, C. O. Lousto, P. Marronetti, and Y. Zlochower, Phys. Rev. Lett. **96**, 111101 (2006).

[5] J. G. Baker, J. Centrella, D.-I. Choi, M. Koppitz, and J. van Meter, Phys. Rev. Lett. **96**, 111102 (2006).

[6] L. Brewin, (Paper 1), Classical Quantum Gravity **19**, 429 (2002).

[7] L. Brewin, Classical Quantum Gravity **15**, 3085 (1998).

[8] L. Brewin, Classical Quantum Gravity **15**, 2427 (1998).

[9] J. Oppenheimer and H. Snyder, Phys. Rev. **56**, 455 (1939).

[10] S. L. Shapiro and S. A. Teukolsky, Astrophys. J. **298**, 34 (1985).

[11] S. L. Shapiro and S. A. Teukolsky, Astrophys. J. **307**, 575 (1986).

[12] P. J. Schinder, S. A. Bludman, and T. Piran, Phys. Rev. D **37**, 2722 (1988).

[13] E. Gourgoulhon, Astron. Astrophys. **252**, 651 (1991).

[14] S. L. Shapiro and S. A. Teukolsky, Phil. Trans. R. Soc. A **340**, 365 (1992).

[15] T. W. Baumgarte, S. L. Shapiro, and S. A. Teukolsky, Astrophys. J. **443**, 717 (1995).

[16] J. Romero, J. Ibanez, J. Marti, and J. Miralles, Astrophys. J. **462**, 839 (1996).

[17] L. Brewin, (Paper 2), Phys. Rev. D **80**, 084030 (2009).

[18] G. Darmois, in *Memorial des Sciences Mathematiques, Fascicule XXV ch V* (Gauthier-Villars, Paris, 1927).

[19] W. Israel, Il Nuovo Cimento **44B**, 1 (1966).

[20] L. I. Petrich, S. L. Shapiro, and S. A. Teukolsky, Phys. Rev. D **31**, 2459 (1985).

[21] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation* (W. H. Freeman and Company, San Francisco, 1973).

[22] P. Hajicek, in *Lecture Notes in Phys.* (Springer-Verlag, Berlin Heidelberg, 2008), Vol. 750, Chap. 6, pp. 209–235.