IOP Publishing Classical and Quantum Gravity
Class. Quantum Grav. **37** (2020) 045015 (11pp) https://doi.org/10.1088/1361-6382/ab5e99

A no-boundary method for numerical relativity

Lydia Bieri[1], **David Garfinkle**[2,3] **and Shing-Tung Yau**[4]

[1] Department of Mathematics, University of Michigan, Ann Arbor, MI 48109-1120, United States of America
[2] Department of Physics, Oakland University, Rochester, MI 48309, United States of America
[3] Leinweber Center for Theoretical Physics, Randall Laboratory of Physics, University of Michigan, Ann Arbor, MI 48109-1120, United States of America
[4] Department of Mathematics, Harvard University, Cambridge, MA 02138 United States of America

E-mail: lbieri@umich.edu, garfinkl@oakland.edu and yau@math.harvard.edu

Received 24 May 2019, revised 12 November 2019
Accepted for publication 4 December 2019
Published 27 January 2020

Abstract

We propose a method for numerical relativity in which the spatial grid is finite and no outer boundary condition is needed. As a 'proof of concept' we implement this method for the case of a self-gravitating, spherically symmetric scalar field.

Keywords: numerical relativity, boundary conditions, black hole formation

1. Introduction

Asymptotically flat spacetimes are infinite in spatial extent, but computational grids are finite. What then should a numerical relativity method do when simulating an asymptotically flat spacetime on a finite grid? In the asymptotic region, the metric behaves like a propagating wave on a flat background. Thus the simplest thing to do is to place the outer boundary somewhat far out in the asymptotic region and to impose the sort of outgoing wave boundary condition that works for the wave equation. However, such simple outer boundary conditions are not consistent with the Einstein field equation, and it is not clear whether the errors made by this inconsistency would be small. The boundary, even if it is far out in the asymptotic region, encloses a finite region of space. Therefore, from the mathematical point of view one should treat this situation by writing the Einstein field equation not as an initial value problem, but as an initial-boundary value problem, using only boundary conditions consistent with this sort of formulation. Such an initial-boundary formulation was first produced by Friedrich and Nagy in [1]. However, the formulation of [1] is somewhat complicated and the variables it uses are not the sort usually used in numerical relativity (though see [2] for a numerical

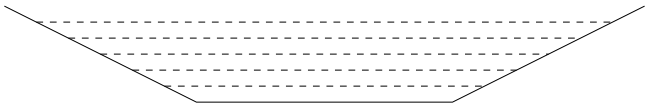

Figure 1. Slices for the No-Boundary method.

implementation). Furthermore, it is not clear how within the allowed boundary conditions of [1] to pick one that physically corresponds to outgoing gravitational waves. Subsequently, a different initial-boundary formulation, compatible with the harmonic coordinate method was developed [3, 4].

Another method is compactification at spatial infinity [5, 6]. Here one chooses spatial coordinates that make the outer boundary of the computational grid correspond to spatial infinity. At that outer boundary, one simply sets the spatial metric to the Euclidean metric and the extrinsic curvature to zero. This is consistent with the Einstein field equation. However eventually waves approach sufficiently close to the outer boundary that there are not enough grid points to resolve them and the simulation loses accuracy.

To maintain resolution of the waves, one can instead compactify at null infinity. One way to do this is through Cauchy-characteristic matching [7, 8]. Here each time slice consists of two pieces: a spacelike piece where variables are evolved using a standard Cauchy method, and a null piece where variables are evolved using a characteristic method. The two sets of variables are matched at the place where the two parts of the surface join, and coordinates are chosen on the null piece so that the outer boundary of the grid is at null infinity. Alternatively, the time slices can be 'hyperboloidal,' that is spacelike slices that go out to null infinity [11]. As yet another alternative, since the formalism of asymptotic flatness involves an unphysical spacetime conformally related to the physical spacetime, one can simply evolve the variables of the unphysical spacetime [9, 10]. Since in the unphysical spacetime, the extent of the physical spacetime is finite, it is natural to have the boundary of the computational grid correspond to the boundary of the physical spacetime within the unphysical spacetime.

In this paper, we propose a simple alternative to all these methods. Our method involves only solving the Cauchy problem and does not require any specialized coordinates or any compactification methods. The main idea of the method is illustrated in figure 1. Here the solid lines represent the initial data and the dashed lines represent the constant time surfaces produced by the Cauchy evolution. Note that the initial data consists of two pieces: a horizontal line representing the $t = 0$ surface, and tilted lines that represent the rest of the initial data. All the dashed lines are within the domain of dependence of the initial data. In numerically evolving from one $t =$ constant surface to the next, most of the evolution is done using a standard Cauchy method, while the information for what to do at the boundary is provided by that part of the initial data on the tilted slices. Since each dashed line is larger than the previous one, each numerical step involves adding points to the computational grid.

As a proof of concept for this method, we will implement it for the case of a spherically symmetric, self-gravitating scalar field. The equations of motion for this system are described in section 2, results of the simulations are presented in section 3, and a discussion is given in section 4.

2. Equations of motion

We will choose the time slices to be maximal (trace of the extrinsic curvature vanishes) and the shift to be zero. These conditions imply that the determinant of the spatial metric does not

change with time, and we will choose coordinates on the initial slice so that square root of the determinant is $r^2 \sin\theta$. With these conditions, the spacetime metric takes the form

$$\mathrm{d}s^2 = -\alpha^2 \mathrm{d}t^2 + e^{-2A}\mathrm{d}r^2 + e^{A} r^2(\mathrm{d}\theta^2 + \sin^2\theta \mathrm{d}\varphi^2) \quad . \tag{1}$$

There is a scalar field Φ that satisfies the curved spacetime wave equation $\nabla^a \nabla_a \Phi = 0$, and from the Einstein field equation we have

$$R_{ab} = \nabla_a \Phi \nabla_b \Phi \tag{2}$$

where we are using units where $8\pi G = 1$. We will put the wave equation in first order form by defining the quantities $P \equiv n^a \nabla_a \Phi$ and $\psi = u^a \nabla_a \Phi$ where n^a is the unit normal vector to the $t =$ const. hypersurfaces and u^a is the unit radial vector in the hypersurface. From equation (1) it then follows that

$$P = \alpha^{-1} \frac{\partial \Phi}{\partial t} \tag{3}$$

$$\psi = e^{A} \frac{\partial \Phi}{\partial r}. \tag{4}$$

From equation (3) it follows that

$$\frac{\partial \Phi}{\partial t} = \alpha P \quad . \tag{5}$$

Taking the time derivative of equation (4) we obtain

$$\frac{\partial \psi}{\partial t} = e^{A}\left(\alpha \frac{\partial P}{\partial r} + P \frac{\partial \alpha}{\partial r}\right) + \alpha K^{r}{}_{r} \psi \quad . \tag{6}$$

Here $K^{r}{}_{r}$ is the eigenvalue of the extrinsic curvature in the radial direction, and we have used the fact that

$$\partial_t A = \alpha K^{r}{}_{r}. \tag{7}$$

From the wave equation and equations (3)–(4) we obtain

$$\frac{\partial P}{\partial t} = e^{A}\psi \frac{\partial \alpha}{\partial r} + \frac{\alpha}{r^2}\frac{\partial}{\partial r}\left(r^2 e^{A} \psi\right) \quad . \tag{8}$$

Equations (5)–(8) constitute the equations of motion of our system. However, in order to implement these equations we need to find the quantities α, A and $K^{r}{}_{r}$. The time components of the Einstein field equations yield a momentum constraint and a Hamiltonian constraint (Gauss–Codazzi equations). From the momentum constraint we obtain

$$\frac{\partial K^{r}{}_{r}}{\partial r} = -K^{r}{}_{r}\left(\frac{3}{2}\frac{\partial A}{\partial r} + \frac{3}{r}\right) - e^{-A} P \psi \quad , \tag{9}$$

while the Hamiltonian constraint yields

$$\frac{\partial^2 A}{\partial r^2} = \frac{1}{r^2}\left(e^{-3A} - 1\right) - \frac{\partial A}{\partial r}\left(\frac{5}{r} + \frac{7}{4}\frac{\partial A}{\partial r}\right) - e^{-2A}\left[\frac{3}{4}(K^{r}{}_{r})^2 + \frac{1}{2}(P^2 + \psi^2)\right] \quad . \tag{10}$$

Finally, the maximal slicing condition yields

$$\frac{\partial^2 \alpha}{\partial r^2} + \left(2\frac{\partial A}{\partial r} + \frac{2}{r}\right)\frac{\partial \alpha}{\partial r} = \alpha e^{-2A}\left[\frac{3}{2}(K^{r}{}_{r})^2 + P^2\right] \quad . \tag{11}$$

The computer program works as follows: on a given time step, we know P and ψ, and so our goal is to find P and ψ on the next time step. We do this using equations (8) and (6), but in order to implement those equations, we need to find A, $K^r{}_r$ and α. We integrate equations (9) and (10) outward from $r = 0$ using the boundary condition that A and $K^r{}_r$ must be zero there. Then we solve equation (11) for α using the fact that $\partial\alpha/\partial r$ must vanish at $r = 0$ and choosing α at the outer boundary in accordance with the spacetime coordinates assigned to the boundary (see appendix). Thus in summary, equations (9)–(11) are integrated in the whole spatial domain (which changes at every time step), while equations (5), (6) and (8) are used to evolve forward in time.

Consistency of the equations requires that $\partial_t A - \alpha K^r{}_r = 0$ so we can use this as a check to see whether the code is working.

However, we need one more piece of information to implement the evolution: the values of P and ψ at the outermost gridpoint of each time slice. This information is given by initial data on the tilted part of the initial data slice, which we will refer to as the boundary surface. Let $\tilde{n}^a$ be the unit normal to the boundary surface and let $\tilde{u}^a$ be the unit radial vector in the boundary surface. In analogy with P and ψ define $\tilde{P} \equiv \tilde{n}^a \nabla_a \Phi$ and $\tilde{\psi} \equiv \tilde{u}^a \nabla_a \Phi$. Just as P and ψ can be freely specified on the horizontal part of the initial data surface, so $\tilde{P}$ and $\tilde{\psi}$ can be freely specified on the boundary surface, subject only to the condition that they match smoothly where the two parts of the initial data surface join. Since both (n^a, u^a) and $(\tilde{n}^a, \tilde{u}^a)$ are orthonormal bases, there must be an angle β such that

$$\tilde{u}^a = u^a \cosh\beta + n^a \sinh\beta \quad , \tag{12}$$

$$\tilde{n}^a = n^a \cosh\beta + u^a \sinh\beta \quad . \tag{13}$$

Inverting this relation, we find

$$u^a = \tilde{u}^a \cosh\beta - \tilde{n}^a \sinh\beta \quad , \tag{14}$$

$$n^a = \tilde{n}^a \cosh\beta - \tilde{u}^a \sinh\beta \quad . \tag{15}$$

The boundary data that we need are the values of P and ψ. However, contracting equations (14) and (15) with $\nabla_a \Phi$ we obtain

$$\psi = \tilde{\psi} \cosh\beta - \tilde{P} \sinh\beta \quad , \tag{16}$$

$$P = \tilde{P} \cosh\beta - \tilde{\psi} \sinh\beta \quad . \tag{17}$$

Thus we can determine the boundary values of P and ψ from the boundary data $(\tilde{P}, \tilde{\psi})$ provided that we know the angle β. We will choose the boundary surface to be generated by outgoing radial spacelike geodesics. This condition along with the maximal slicing condition yields the following evolution equation for β.

$$\frac{\mathrm{d}\beta}{\mathrm{d}t} = \frac{\alpha K^r{}_r}{\tanh\beta} - e^A \frac{\partial\alpha}{\partial r} \quad . \tag{18}$$

See appendix for a derivation of equation (18). Thus at each time step, we evolve β using equation (18) and we use equations (17) and (16) to find P and ψ at the last gridpoint.

One can think of equations (16)–(18) as playing the role of 'boundary condition' in our no-boundary method. However, it is important to remember that our 'boundary' is really just part of the initial data, and that equations (16)–(18) simply implement the way in which that part of the initial data is used to determine the solution. It would not be difficult to compare the solutions obtained using our no-boundary method to those obtained with some standard boundary conditions. However, we have not made such a comparison.

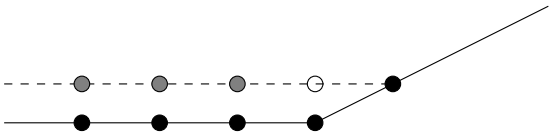

Figure 2. Stencil for the No-Boundary method.

The stencil for the evolution of ψ and P is shown in figure 2. Here the circles represent grid points, the solid horizontal line represents time step n, the dashed horizontal line represents time step $n+1$, and the solid tilted line represents the boundary. We suppose that we have evolved up to time step n and now need to perform the task of evolving from time step n to time step $n+1$. The black circles represent points where the fields are known, either because they are are at time step n, or because they are on the boundary and thus, with our no-boundary method are part of the initial data. The gray circles represent points where the evolution can be done using the standard method, while the white circle represents a special point where something different must be done. The evolution is done using the iterative Crank–Nicholson method. This method calculates the time derivatives of ψ and P using equations (6) and (8) at time step n and iteratively at time step $n+1$ where centered spatial differences are used to calculate the spatial derivatives that occur on the right hand side of equations (6) and (8). For each gray circle, there are points at the previous time step on either side in space. Thus there are enough points to calculate centered spatial differences at that previous time step. For the white circle, there is no point at the previous time step and one further space step. Thus, centered spatial differences at the previous time step cannot be calculated. For this point, we calculate the time derivative using only the (iteratively found) spatial differences at time step $n+1$.

3. Results

We choose the flat part of the initial data surface to be $t = 0$, $0 \leqslant r \leqslant r_0$ and the tilted part of the initial data surface to be $t = (r - r_0)/2$, $r_0 \leqslant r \leqslant r_{\max}$ where r_0 and $r_{\max}$ are constants. The fact that $\mathrm{d}t/\mathrm{d}r = 1/2$ on the boundary requires us to choose $\alpha = 2e^{-A}\tanh\beta$ on the boundary. A simple way to get initial data that is smooth where the two parts of the surface join is simply to take a smooth function on spacetime and pull it back to the initial data surface. We choose this spacetime function to be a solution of the flat spacetime wave equation. This is also a physically reasonable choice since far from the center a solution of the curved spacetime wave equation is well approximated by a solution of the flat spacetime wave equation. In flat spacetime a smooth solution of the spherically symmetric wave equation takes the form

$$\Phi(t,r) = \frac{1}{r}(f(t+r) - f(t-r)) \tag{19}$$

where $f(s)$ is any smooth function. We choose

$$f(s) = \frac{a}{1+ks^2}. \tag{20}$$

Where a and k are constants. This function represents a scalar field with amplitude a, with its energy concentrated around the center in a region of radius approximately $1/\sqrt{k}$. Note that the initial value of Φ is zero, but its initial time derivative is nonzero. We can check to see when

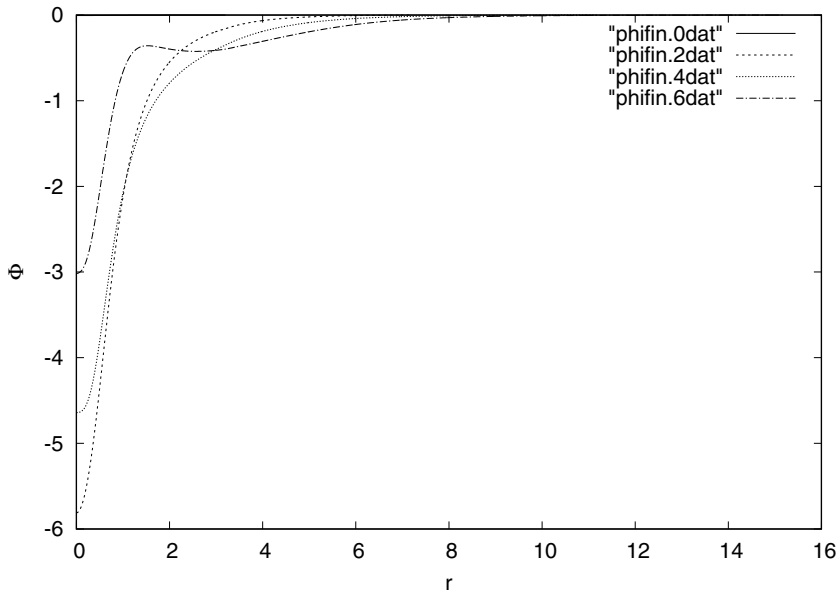

Figure 3. $\Phi(r)$ at times 0, 2, 4 and 6. Here the amplitude is 1.4 and no black hole forms.

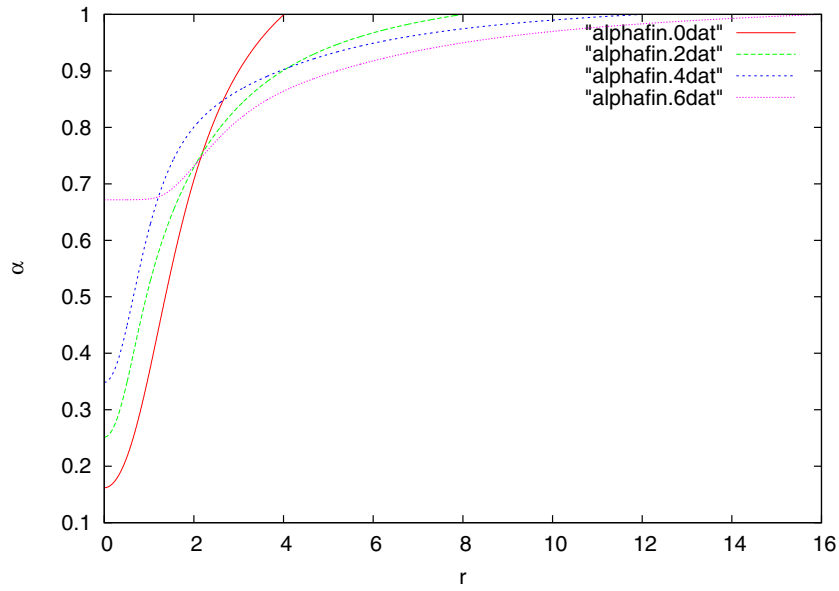

Figure 4. $\alpha(r)$ at times 0, 2, 4 and 6. Here the amplitude is 1.4 and no black hole forms.

a black hole forms by looking for a marginally outer trapped surface (MOTS). The condition for a MOTS is

$$1 + \frac{r}{2}\left(\frac{\partial A}{\partial r} + e^{-A}K^{r}{}_{r}\right) = 0. \tag{21}$$

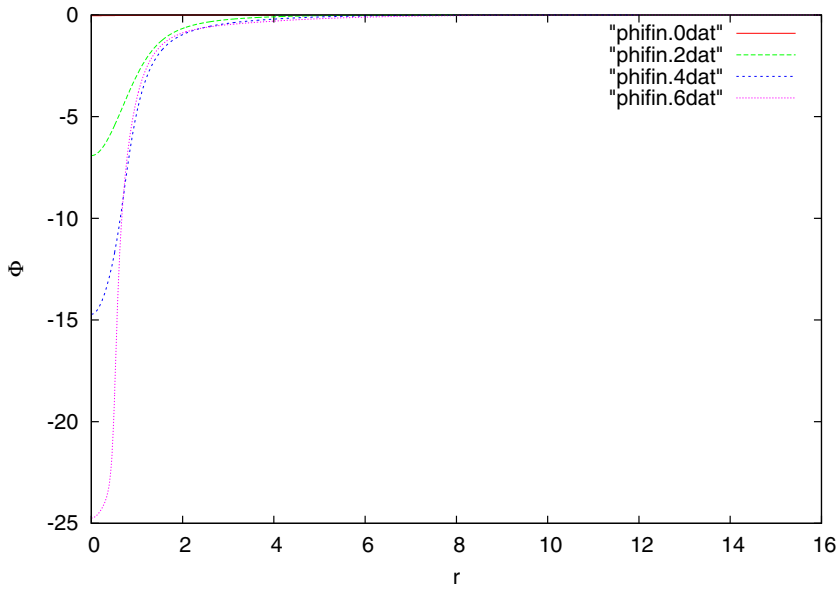

Figure 5. $\Phi(r)$ at times 0, 2, 4 and 6. Here the amplitude is 1.7 and a black hole forms.

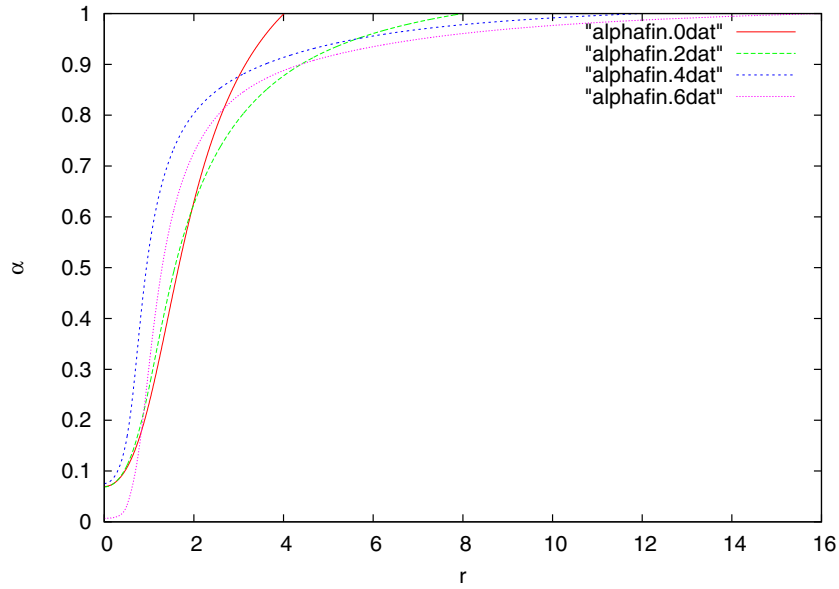

Figure 6. $\alpha(r)$ at times 0, 2, 4 and 6. Here the amplitude is 1.7 and a black hole forms.

In figures 3 and 4 we show the results of a simulation where a black hole does not form. In this case the scalar field eventually disperses and the lapse evolves towards its flat space value of 1.

In figures 5 and 6 we show the results of a simulation where a black hole forms. Here a portion of the scalar field remains trapped in the central region. Also in this region there is the standard 'collapse of the lapse' in which α takes on values close to zero.

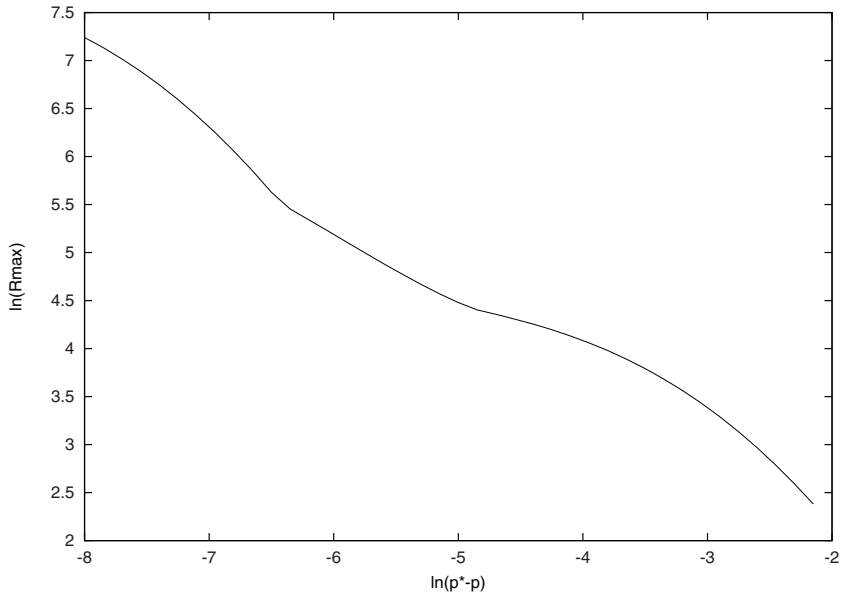

Figure 7. Curvature scaling for critical collapse.

With our method, the code can only run for as long as there is data on the tilted part of the initial data surface. But that is long enough, provided that all the interesting physics happens by that time. To illustrate this point, we performed numerical simulations of critical gravitational collapse [12]. In critical collapse, one examines a family of initial data depending on a parameter p where the threshold of black hole formation occurs at $p = p*$. For p slightly greater than $p*$, there is a scaling relation for black hole mass

$$M \propto (p - p*)^{\gamma}. \tag{22}$$

While for p slightly less than $p*$ there is a scaling relation for the maximum value of the spacetime curvature [13]

$$R_{\max} \propto (p* - p)^{-2\gamma} \tag{23}$$

where the constant γ in equation (23) is the same as in equation (22). More precisely: equation (22) implies that a graph of $\ln M$ versus $\ln(p - p*)$ is a straight line with slope γ, while the actual result is a line with average slope γ but with a small periodic wiggle [14]. Correspondingly, in the subcritical case, a graph of $\ln(R_{\max})$ versus $\ln(p* - p)$ is a line with average slope -2γ but with a periodic wiggle. In our case, we use the amplitude a as our parameter p. Using a binary search, we find the critical value $p*$. We then run a sequence of simulations slightly below the threshold of black hole formation. The results are plotted in figure 7. Note that the maximum curvature has the appropriate scaling.

Our critical collapse simulations were run with a maximum radius of 30 and with a fixed mesh size of $\mathrm{d}r = 0.0006$ and with $\mathrm{d}t$ given by 1/2 of the maximum allowed by the Courant condition. We thus have a fairly low resolution treatment of critical collapse, as compared to the higher resolution that can be obtained with mesh refinement. Nonetheless, it is of interest to treat critical collapse with the maximal slicing condition that we use, because such slicing, in contrast to the slicing condition used in [12], can follow the evolution even after the formation of a black hole. This issue will be addressed elsewhere [15] using mesh refinement for high resolution.

4. Discussion

We now consider the possibility of generalizing our method, first to the case of other types of matter in spherical symmetry and then to the case of the Einstein field equations with no symmetry. One simple generalization would be to use a fluid as the matter instead of a scalar field. Here the main complication is that fluids can have shocks, and therefore require a numerical shock capturing scheme. However, since the entire initial data surface is spacelike, including the boundary, this means that if no shocks are present initially, then no shocks will propagate to the boundary. Thus, the part of the numerical evolution that involves use of the boundary data would not need any special shock capturing feature, and would simply involve the use of equations (12) and (13) to find the analog for fluid data of the transformation given in equations (16) and (17).

For the case of the vacuum Einstein equation with no symmetry, we note that the main features of our method should apply to any system of hyperbolic equations, and should therefore be suitable for Einstein's equations in the generalized harmonic formulation [6, 16, 17] or the BSSN formulation [18, 19]. In the spherically symmetric case, the initial data for the scalar field could be freely specified, while in the general case, the initial data for the Einstein field equations would have to satisfy constraint equations. However, this is no different from the usual Cauchy problem for the Einstein equation: the only difference is that the constraint equations would have to be solved on both the flat and tilted parts of the initial data surface, with care taken to impose a condition of smoothness at the place where the two parts of the initial data surface join.

Finally, we want to consider the possible expense, in terms of computer memory and time, of our method. Since the method adds extra spatial points at each time step, it is possible that the method could become unweildy when run long enough to extract the relevant physics. This could certainly be the case when using a Cartesian coordinate system, since in that case extra points need to be added in all three directions. However, for a method using spherical coordinates [20] or where the outermost coordinate patch uses spherical coordinates [21] one only needs to add radial points: no additional angular resolution is needed. Furthermore, using our method it may well be that one could get away with making the initial outer boundary radius r_0 smaller than that of the fixed outer boundary used in the usual Cauchy codes. Thus our method could be *less* expensive at early times and only become more expensive towards the later parts of the simulation. More specifically, consider a possible application of our method to the SpEC code of [21]. The outer boundary of the SpEC code is typically placed at a distance of $600M$ to $1200M$ [22] (where M is the sum of the Christodoulou masses of the black holes) with a nested set of spherical domains covering the radiation zone. These outer domains need a sufficient number of radial colocation points to resolve features whose radial length scale is of the order of the gravitational radiation wavelength. Suppose instead that we place the outer boundary of the horizontal part of our initial data surface at say $50M$ and use the post-Newtonian approximation for the radiation fields coming from the black hole inspiral to determine the boundary data (i.e. the data on the tilted part of our initial data surface). If the post-Newtonian approximation is also used to help in determining the data on the horizontal part of the initial data surface, this could significantly cut down on the amount of 'junk radiation' present in the initial data. Thus the computational domain would start off much smaller than that of the usual simulations, but would grow with time. As the evolution proceeded, one would need to add radial colocation points to cover the additional amount of space, but only in domains where the radial size of the features to be resolved are of order the wavelength of the gravitational waves. For sufficiently long evolutions, the need to add so many radial points

might become onerous; but at that point one could switch to the ingoing boundary method of Ripley [23] to reduce the computational burden.

We therefore expect that our no-boundary method will be a useful addition to the tools used in numerical relativity, as well as being a simple way of making numerical simulations consistent with the mathematics of the Einstein field equation.

Acknowledgments

David Garfinkle thanks the Harvard Black Hole Initiative for hospitality, and acknowleges support from NSF Grants PHY-15105565 and PHY-1806219. Lydia Bieri acknowleges support from NSF Grants DMS-1253149 and DMS-1811819 and Simons Fellowship in Mathematics 555809. Shing-Tung Yau acknowledges support from NSF Grant DMS-1607871. We thank Luis Lehner, Carsten Gundlach, Justin Ripley, and Mark Scheel for helpful discussions.

Appendix. Evolution of β

We now derive the equation of motion for β (equation (18). From equation (1) it follows that

$$n^a = \alpha^{-1}\left(\frac{\partial}{\partial t}\right)^a \tag{A.1}$$

$$u^a = e^A\left(\frac{\partial}{\partial r}\right)^a. \tag{A.2}$$

Then using equations (A.1) and (A.2) in equation (12) we obtain

$$\tilde{u}^a = \frac{\sinh\beta}{\alpha}\left[\left(\frac{\partial}{\partial t}\right)^a + \alpha e^A \coth\beta\left(\frac{\partial}{\partial r}\right)^a\right]. \tag{A.3}$$

Let $d/\mathrm{d}t$ denote derivative with respect to t in the direction along the boundary. That is, for any quantity F we have $\mathrm{d}F/\mathrm{d}t = (\tilde{u}^a\nabla_a F)/(\tilde{u}^a\nabla_a t)$. In particular, it then follows from equation (A.3) that

$$\frac{\mathrm{d}r}{\mathrm{d}t} = \alpha e^A \coth\beta. \tag{A.4}$$

Since in our simulations we pick the boundary to have $\mathrm{d}r/\mathrm{d}t = 2$, it follows that on the boundary we must impose the condition

$$\alpha = 2e^{-A}\tanh\beta. \tag{A.5}$$

From equations (12) and (A.3) it follows that

$$\tilde{u}^a\nabla_a(\tilde{u}^b n_b) = -\alpha^{-1}\sinh\beta\cosh\beta\frac{\mathrm{d}\beta}{\mathrm{d}t}. \tag{A.6}$$

However, since $\tilde{u}^a$ is geodetic, we have $\tilde{u}^a\nabla_a(\tilde{u}^b n_b) = \tilde{u}^a\tilde{u}^b\nabla_a n_b$, and then it follows using equation (12) that

$$\tilde{u}^a\nabla_a(\tilde{u}^b n_b) = \cosh^2\beta u^a u^b\nabla_a n_b + \sinh^2\beta n^a n^b\nabla_a n_b + \cosh\beta\sinh\beta(u^a n^b + n^a u^b)\nabla_a n_b. \tag{A.7}$$

But n^a is a unit vector, and therefore $n^b\nabla_a n_b = 0$, so we have

$$\tilde{u}^a \nabla_a (\tilde{u}^b n_b) = \cosh^2\beta u^a u^b \nabla_a n_b + \cosh\beta \sinh\beta n^a u^b \nabla_a n_b. \tag{A.8}$$

Since u^a is the unit vector in the r direction, it follows that $u^a u^b \nabla_a n_b = -K^r{}_r$. However using equations (A.1) and (A.2) we obtain

$$n^a u^b \nabla_a n_b = 2 n^a u^b \nabla_{[a} n_{b]} = -2 n^a u^b \partial_{[a}(\alpha \partial_{b]} t) = \alpha^{-1} e^A \frac{\partial \alpha}{\partial r}. \tag{A.9}$$

Collecting terms in equation (A.8) we then find

$$\tilde{u}^a \nabla_a (\tilde{u}^b n_b) = -\cosh^2\beta K^r{}_r + \cosh\beta \sinh\beta \alpha^{-1} e^A \frac{\partial \alpha}{\partial r}. \tag{A.10}$$

Finally, equating the right hand sides of equations (A.6) and (A.10) we obtain equation (18).

ORCID iDs

Lydia Bieri https://orcid.org/0000-0002-2469-3409
David Garfinkle https://orcid.org/0000-0003-4415-774X

References

[1] Friedrich H and Nagy G 1999 *Commun. Math. Phys.* **201** 619
[2] Frauendiener J and Stevens C 2014 *Phys. Rev.* D **89** 104026
[3] Kreiss H O and Winicour J 2006 *Class. Quantum Grav.* **23** S405
[4] Kreiss H O, Reula O, Sarbach O and Winicour J 2007 *Class. Quantum Grav.* **24** 5973
[5] Garfinkle D and Duncan G C 2001 *Phys. Rev.* D **63** 044011
[6] Pretorius F 2006 *Class. Quantum Grav.* **23** S529
[7] Winicour J 2012 *Living Rev. Relativ.* **15** 2
[8] Bishop N T, Gomez R, Lehner L, Maharaj M and Winicour J 1997 *Phys. Rev.* D **56** 6298
[9] Friedrich H 1983 *Commun. Math. Phys.* **91** 445
[10] Frauendiener J 1998 *Phys. Rev.* D **58** 064002
[11] Moncrief V and Rinne O 2009 *Class. Quantum Grav.* **26** 125010
[12] Choptuik M W 1983 *Phys. Rev. Lett.* **70** 9
[13] Garfinkle D and Duncan G C 1998 *Phys. Rev.* D **58** 064024
[14] Gundlach C 1997 *Phys. Rev.* D **55** 695
[15] Bieri L, Garfinkle D and Sanchez B in preparation
[16] Friedrich H 1985 *Commun. Math. Phys.* **100** 525
[17] Garfinkle D 2002 *Phys. Rev.* D **65** 044029
[18] Shibata M and Nakamura T 1995 *Phys. Rev.* D **52** 5428
[19] Baumgarte T W and Shapiro S L 1999 *Phys. Rev.* D **59** 024007
[20] Baumgarte T W, Montero P J, Cordero-Carrion I and Muller E 2013 *Phys. Rev.* D **87** 044026
[21] Mroue et at A H 2013 *Phys. Rev. Lett.* **111** 241104
[22] Scheel M 2019 private communication
[23] Ripley J 2019 (arXiv:1908.04234v2)