

# Black Hole Excision with Multiple Grid Patches

Jonathan Thornburg\*

*Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, Am Mühlenberg 1,  
D-14476 Golm, Germany*

(Dated: Id: multipatch.tex,v 1.264 2004/04/14 15:13:07 jonathan Exp )

When using black hole excision to numerically evolve a black hole spacetime with no continuous symmetries, most  $3 + 1$  finite differencing codes use a Cartesian grid. It's difficult to do excision on such a grid because the natural  $r = \text{constant}$  excision surface must be approximated either by a very different shape such as a contained cube, or by an irregular and non-smooth “LEGO<sup>†</sup> sphere” which may introduce numerical instabilities into the evolution. In this paper I describe an alternate scheme, which uses multiple  $\{r \times (\text{angular coordinates})\}$  grid patches, each patch using a different (nonsingular) choice of angular coordinates. This allows excision on a smooth  $r = \text{constant}$  2-sphere.

I discuss the key design choices in such a multiple-patch scheme, including the choice of ghost-zone versus internal-boundary treatment of the interpatch boundaries (I use a ghost-zone scheme), the number and shape of the patches (I use a 6-patch “inflated-cube” scheme), the details of how the ghost zones are “synchronized” by interpolation from neighboring patches, the tensor basis for the Einstein equations in each patch, and the handling of non-tensor field variables such as the BSSN  $\tilde{\Gamma}^i$  (I use a scheme which requires ghost zones which are twice as wide for the BSSN conformal factor  $\phi$  as for  $\tilde{\Gamma}^i$  and the other BSSN field variables).

I present sample numerical results from a prototype implementation of this scheme. This code simulates the time evolution of the (asymptotically flat) spacetime around a single (excised) black hole, using 4th order finite differencing in space and time. Using Kerr initial data with  $J/m^2 = 0.6$ , I present evolutions to  $t \gtrsim 1500m$ . The lifetime of these evolutions appears to be limited only by outer boundary instabilities, not by any excision instabilities or by any problems inherent to the multiple-patch scheme.

PACS numbers: 04.25.Dm, 02.70.-c, 02.70.Bf, 02.60.Lj

Keywords: numerical relativity, black hole, excision, multiple grid patches

## I. INTRODUCTION

When time-evolving a black hole spacetime in  $3 + 1$  (Cauchy) numerical relativity, the numerical computation must somehow be kept from encountering the singularity(ies) within the black hole(s). There are three common means of doing this: freezing slicings (Lichnerow-

---

\*Electronic address: [jthorn@aei.mpg.de](mailto:jthorn@aei.mpg.de); URL: <http://www.aei.mpg.de/~jthorn>

<sup>†</sup> LEGO is a registered trademark of the LEGO Company, <http://www.lego.com>.

icz (1944); Smarr and York (1978)), the Brandt-Brügmann “puncture” method (Brandt and Brügmann (1997)), and excision (Anninos *et al.* (1995); Seidel and Suen (1992); Thornburg (1985, 1987, 1993); Unruh (1984)). In this paper I focus on excision in the context of finite-difference evolutions, in particular the case where the spacetime has no continuous symmetries and hence a fully 3-spatial-dimensional numerical grid must be used. For convenience of exposition I primarily consider the case where there is precisely one (excised) black hole in each slice; I briefly discuss possible extensions to multiple black holes in the conclusions.

Polar spherical coordinates and grids centered on a black hole provide a natural way to do excision, with an  $r = \text{constant}$  excision surface being a (smooth) grid plane. However, problems with  $z$  axis coordinate singularities (and the difficulty of generalizing to multiple black holes) have led most researchers to switch to Cartesian grids over the past decade.

Unfortunately, it’s difficult to do excision on a Cartesian grid:

1. If the excision boundary is a cube, there are severe causality problems (Calabrese and Neilsen (2004b); Lehner (2003); Lehner *et al.* (2004a); Scheel and Kidder (2000)). For Schwarzschild spacetime in Eddington-Finkelstein (spacelike) coordinates, the excised region must be *very* small to keep the evolution causal. For Kerr spacetime in Kerr-Schild coordinates this problem is even worse, and for most spin parameters there is *no* cubical excision region which keeps the evolution causal. For a dynamic spacetime this problem is even more severe.
2. If the excision boundary is some other regular polyhedron which better approximates a sphere (such as the 14-sided shape obtained by cutting a pyramid off each of a cube’s corners), the continuum causality problem is less severe, but the additional edges and corners considerably complicate the construction of a stable finite differencing scheme.
3. If the excision boundary is an irregular “LEGO sphere”, then at the continuum level (approximating the boundary as a sphere centered on the origin), any non-extremal Kerr spacetime admits a causal evolution (Thornburg (1993, figure 3.8)).<sup>1,2</sup> However, at the discrete level, the irregularity of the boundary makes it very difficult to construct stable finite differencing schemes. (Notably, it’s not clear whether the powerful numerical schemes of Calabrese *et al.* (2003a,b) can be extended to this case.) The causality problem of case 1 may also still apply to the individual “LEGO sphere” grid points on the excision boundary.

In this paper I discuss an alternative technique for doing excision, which is similar to the use of a polar spherical grid, but with the  $(\theta, \phi)$  angular grid replaced by *multiple angular grid patches* which collectively cover  $S^2$  in a non-singular manner. The corresponding  $\{r \times (\text{multiple angular patches})\}$  coordinate system can be constructed to have no coordinate singularities, and the multiple-patch grid still has  $r = \text{constant}$  as a (smooth) excision surface.

---

<sup>1</sup> For the Kerr slicing of Kerr spacetime, an  $r = \text{constant}$  excision boundary gives a causal evolution whenever the boundary lies between the inner and outer horizons. This gives a relatively broad range of allowable boundary positions for small spin parameters, narrowing as the spin increases.

<sup>2</sup> Since the causality depends only *algebraically* on the 4-metric, the same property also holds for all “nearby” spacetimes in some functional neighborhood of Kerr spacetime.

Like a polar spherical grid, such a multiple-patch grid provides a smooth  $r = \text{constant}$  outer boundary for the imposition of outgoing radiation conditions, and makes it easy to place the  $r = \text{constant}$  shells of grid points nonuniformly in radius so as to give high resolution close to the black hole while still having the outer boundary relatively far away.<sup>3</sup>

Multiple-patch finite difference schemes with smooth excision surfaces have long been used in computational fluid dynamics (see, for example, the discussions of Rubbert and Lee (1982), Thompson *et al.* (1985, section II.4), Chesshire and Henshaw (1990, 1994), Gustafsson *et al.* (1995, section 13.4), Brown *et al.* (1997), and Petersson (1999)). However, such schemes have seen relatively little use in numerical relativity, and that mainly for elliptic problems, either initial data construction (Thornburg (1985, 1987))<sup>4</sup> or apparent-horizon finding (Thornburg (2004)).

Gómez *et al.* (1997) have described a multiple-patch “eth formalism” for representing tensor fields on  $S^2$ , and this has proven useful for characteristic (null-cone) evolutions of black hole spacetimes (see, for example, Bishop *et al.* (1997); Gómez *et al.* (1998)). The eth formalism uses a pair of overlapping stereographic-coordinate patches to cover  $S^2$ , with field variables in each patch’s ghost zone interpolated (in 2-D) from the other patch.

Calabrese and Neilsen (2004a,b) have described a multiple-patch scheme for the axisymmetric evolution of a scalar field on a Schwarzschild background. They use two overlapping patches: a polar spherical inner patch (with a smooth  $r = \text{constant}$  excision surface) and a cylindrical outer patch. Ghost-zone values of all the dynamical fields in each patch’s ghost zone are set by (2-D) bilinear interpolation from the other patch. Their discretization (Calabrese *et al.* (2003a,b)) is based on finite difference operators which preserve discrete energy norms within each patch, maximally dissipative boundary conditions, and added artificial dissipation to ensure stability. Their scheme gives long-term stable evolutions with generally 2nd order overall accuracy;<sup>5</sup> a variant discretization gives 3rd order overall accuracy.

Lehner *et al.* (2004b); Reula (2003); Tiglio (2003, 2004) have described a different multiple-patch scheme for studying scalar field tails in Schwarzschild or Kerr spacetime. They use an “internal boundaries” scheme, where adjacent patches just touch (rather than overlapping). By using a symmetric hyperbolic form of the PDEs, the dynamical fields can be unambiguously decomposed into ingoing and outgoing modes crossing each interpatch boundary. By adding penalty terms to the field equations, the difference between the value of each ingoing mode at a patch boundary and its value as an outgoing mode in the adjacent patch, can be driven to zero.

In this paper I describe a multiple-patch excision scheme for the full nonlinear vacuum  $3 + 1$  Einstein equations. In terms of past multiple-patch schemes in numerical relativity, this scheme most resembles those of Gómez *et al.* (1997) and Calabrese and Neilsen (2004b),

---

<sup>3</sup> Berger-Oliger mesh refinement (Berger (1982, 1986); Berger and Colella (1989); Berger and Oliger (1984)) provides a much more general and efficient solution here, but it’s quite difficult to implement for the Einstein equations (Choptuik (1986, 1989); Schnetter *et al.* (2004)).

<sup>4</sup> Kidder and Finn (2000) and Pfeiffer *et al.* (2003) have described multiple-patch spectral methods for elliptic PDEs, and used them for initial data construction.

<sup>5</sup> Although bilinear interpolation has an  $O((\Delta x)^2)$  truncation error, this error is non-smooth at the interpolation points (Thornburg (1999a, appendix F)), so spatial derivatives of the field variables are generically only 1st order accurate at grid points where interpolated values enter into the derivative molecules. However, in practice this doesn’t seem to seriously impair the overall convergence of the scheme (Calabrese and Neilsen (2004a)). I discuss the analogous point for my multiple-patch scheme in section III.

in that I couple the patches by ghost-zone interpolation of all the dynamical fields. However, my scheme uses patches which may either overlap or just touch, and there are many other differences in detail.

I first implemented the “ $g$ - $K$ ” ADM form of the  $3 + 1$  Einstein equations (York (1979)), but it’s now well known that these are ill-posed even in the absence of boundary conditions (see, for example, the numerical tests of Alcubierre *et al.* (2004)), and I saw rapid error growth near the interpatch boundaries, with evolutions lasting  $\lesssim 100m$  (Thornburg (2003)). I now use the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) form of the  $3 + 1$  Einstein equations first introduced by Nakamura and Oohara (1989, 1998); Nakamura *et al.* (1987); Shibata (1999); Shibata and Nakamura (1995), and popularized by Baumgarte and Shapiro (1999). In particular, I use the “actively forced  $A$ ” variant described by Alcubierre *et al.* (2000).

In this paper I also present initial numerical results for the BSSN system, evolving Kerr spacetime with high accuracy for  $\gtrsim 1000m$ , and lasting for  $t \gtrsim 1500m$  before crashing. These evolutions are unstable at the outer boundary (where I have only partially implemented Sommerfeld outgoing-radiation boundary conditions), but they show no signs of any instabilities caused by the multiple-patch scheme. There are also no signs of any instabilities at the inner (excision) boundary.

## A. Notation

I generally follow the sign and notation conventions of Wald (1984). I use the Penrose abstract-index notation, with indices  $a$ – $z$  running over both Cartesian coordinates  $\bar{x}^i \equiv (x, y, z)$  in a spacelike  $3 + 1$  slice, and the generic patch coordinates  $x^i \equiv (r, \rho, \sigma)$  defined in section II.B.

$g_{ij}$  is the 3-metric in the slice,  $g$  is its determinant, and  $\nabla_i$  is the associated covariant derivative operator.  $K_{ij}$  is the extrinsic curvature of the slice (I use the sign convention of York (1979), not that of Wald (1984)) and  $K \equiv K_i^i$  is its trace. “ $N$ - $D$ ” abbreviates “ $N$ -dimensional”, and specifically refers only to spatial dimensions (i.e. the temporal dimension is never included).

I use **sans-serif** font for the names of patches, ghost zones, and (in appendix A) ghost zone widths and patch interpolation regions. In section II.E I use indices  $abc$  and  $ijk$  for the  $(r, \rho, \sigma)$  coordinates of a pair of neighboring patches  $\mathbf{p}$  and  $\mathbf{q}$  respectively, and I use the notation  $f(\mathbf{p})$  for the grid function  $f$  in the local coordinate basis of patch  $\mathbf{p}$ .

In the context of a particular angular patch boundary, I use angular coordinates  $(\perp, \parallel)$  which are (perpendicular, parallel) to the boundary. In appendix A I also use the corresponding integer grid-point coordinates and grid-function array indices (`iperp`, `ipar`), and I use a notation inspired by the C++ programming language, where `(p.iperp, p.ipar)` refers to the (`iperp`, `ipar`) coordinates of patch  $\mathbf{p}$ .

In appendix C I use overbars (as in  $\bar{\phi}$ ) to refer to field variables in (strictly speaking, their coordinate components with respect to) the Cartesian coordinates.

## II. DESIGN OF THE MULTIPLE-PATCH SYSTEM

In this section I discuss a variety of design issues which arise when trying to design and implement such an  $\{r \times (\text{multiple angular patches})\}$  numerical scheme for the  $3 + 1$  vacuum Einstein equations.

## A. Treatment of Interpatch Boundaries

A fundamental design question for any multiple-patch scheme is how the interpatch boundaries should be treated in the numerical scheme, or equivalently, how the patches should be coupled together.

In terms of finite differencing, the usual ghost-zone scheme is conceptually simple: each patch’s nominal grid is surrounded by a molecule-radius “ghost zone” of extra grid points on each side (face) of the patch. Values of all the dynamical fields in the ghost zones are computed (“synchronized”) by interpolation from the adjacent patches. Finite differencing can then be done at all the nominal grid points without further concern for the interpatch boundaries. This scheme can be used with essentially any formulation of the Einstein equations, and also generalizes to elliptic PDEs in a reasonably straightforward and efficient manner (Thornburg (1985, 1987, 2004)). The main problem with the ghost-zone scheme is that there’s relatively little analytical theory regarding its stability, although Olsson and Petersson (1996) have obtained some promising results for 1-D model problems.

In contrast, the “internal-boundaries” scheme used by Lehner *et al.* (2004b); Reula (2003); Tiglio (2003, 2004) requires a formulation of the Einstein equations in which the modes crossing each internal boundary can be cleanly separated. This must be done everywhere on each interpatch boundary, including (in general) in strong-field regions. Although this scheme does generalize to elliptic PDEs, this generalization is quite inefficient in multiple spatial dimensions.<sup>6</sup> The main advantage of the internal-boundaries scheme is its amenability to mathematical analysis at both the continuum and the finite differencing levels. Also, potentially-destabilizing feedback loops from one patch to another and back again only occur due to nonlinear effects coupling the oppositely-directed modes, whereas in the ghost-zone scheme all the field variables participate directly in such feedback loops.

Because of its simplicity, and for historical reasons, I have chosen the ghost-zone scheme for this work.

## B. Choice of Angular Patches and Coordinates

There are a number of ways to cover  $S^2$  with nonsingular coordinate patches, of which two seem particularly interesting:

One possibility (Gómez *et al.* (1997)) is to use a pair of stereographic-coordinate patches, one covering the northern hemisphere and one the southern. This has the advantage of relative simplicity (there is only a single interpatch boundary), and the eth formalism provides an elegant way to represent and manipulate tensor fields on  $S^2$ .

However, these coordinates have relatively large coordinate distortion near the equator, which makes finite differencing somewhat less accurate there. Another potential problem is the coordinate choice on each patch: If Cartesian stereographic coordinates are used on

---

<sup>6</sup> Briefly, in this scheme nonlinear elliptic PDEs are first linearized in the Newton-Kantorovich sense (Boyd (2000, appendix C)). Linear elliptic PDEs are then solved by introducing homogeneous solutions where all the interpatch field values vanish, and one particular solution for each interpatch boundary point. Thus in a single spatial dimension, the number of particular solutions is just twice the number of patches. But in multiple spatial dimensions, it’s twice the number of interpatch-boundary *grid points*, which is generally quite large.

each patch, then the grids overlap irregularly near the equator (requiring 2-D interpatch interpolation there), while if polar stereographic coordinates are used on each patch, then there are coordinate singularities at the north and south poles.

Another possibility is to use 6 patches, covering neighborhoods of the  $\pm x$ ,  $\pm y$ , and  $\pm z$  axes respectively. This has the disadvantage of relative complexity: there are many interpatch boundaries, and there are also corners where 3 patches meet. However, this system has the advantage of relatively low coordinate distortion, yielding accurate finite differencing. Also, if the coordinates are suitably chosen (I describe this in detail below), it's possible to have adjacent patches always share a common angular coordinate, so only 1-D interpatch interpolation is needed. Hopefully this will yield an evolution scheme less prone to instabilities than one using multidimensional interpolation.

Because of the lower coordinate distortion, and the potential better stability of time evolutions using 1-D interpatch interpolation, I use a 6-patch system here. In more detail, I use the “inflated-cube” angular 6-patch system described in Thornburg (2004): Given Cartesian coordinates  $(x, y, z)$ , with the excised (black hole) region at the origin, I define 3 angular coordinates on  $S^2$  based on rotation angles about the  $xyz$  coordinate axes:

$$\begin{aligned}\mu &\equiv \text{rotation angle about the } x \text{ axis} = \arctan(y/z) \\ \nu &\equiv \text{rotation angle about the } y \text{ axis} = \arctan(x/z) \\ \varphi &\equiv \text{rotation angle about the } z \text{ axis} = \arctan(y/x)\end{aligned}\tag{1}$$

where all the arctangents are 4-quadrant based on the signs of  $x$ ,  $y$ , and  $z$ . I then define 6 coordinate patches covering neighborhoods of the  $\pm z$ ,  $\pm x$ , and  $\pm y$  axes, using the generic (angular) patch coordinates  $(\rho, \sigma)$  defined by

$$\begin{aligned}\pm z \text{ patch has } (\rho, \sigma) &= (\mu, \nu) \\ \pm x \text{ patch has } (\rho, \sigma) &= (\nu, \varphi) \\ \pm y \text{ patch has } (\rho, \sigma) &= (\mu, \varphi)\end{aligned}\tag{2}$$

Notice that each patch's  $(\rho, \sigma)$  coordinates are nonsingular throughout a neighborhood of the patch, and that adjacent patches always share the (common) angular coordinate perpendicular to their mutual boundary. The name “inflated-cube” comes from another way to visualize these patches and coordinates: Imagine an  $xyz$  cube with  $xyz$  grid lines painted on its face. Now imagine the cube to be flexible, and inflate it like a balloon, so it becomes spherical in shape. The resulting coordinate lines will closely resemble those for  $(\mu, \nu, \varphi)$  coordinates.

This set of 6 patches covers  $S^2$  without coordinate singularities. Alternatively, if the spacetime has  $z \leftrightarrow -z$  reflection symmetry about the coordinate origin, then the 5 patches  $+z$ ,  $\pm x$ , and  $\pm y$  cover the  $+z$  hemisphere of  $S^2$ . Similarly, suitable sets of 4 or 3 patches may be used to cover quadrants or octants. Figure 1 shows an example of a 3-patch system covering the  $(+, +, +)$  octant of  $S^2$ .

Defining the usual radial coordinate  $r \equiv (x^2 + y^2 + z^2)^{1/2}$ , the corresponding 6-patch (or 5-, 4-, or 3-patch)  $(r, \rho, \sigma)$  coordinates cover all of  $\mathfrak{R}^3$  (or the corresponding subsets) with no singularities except at the origin (which is excised).

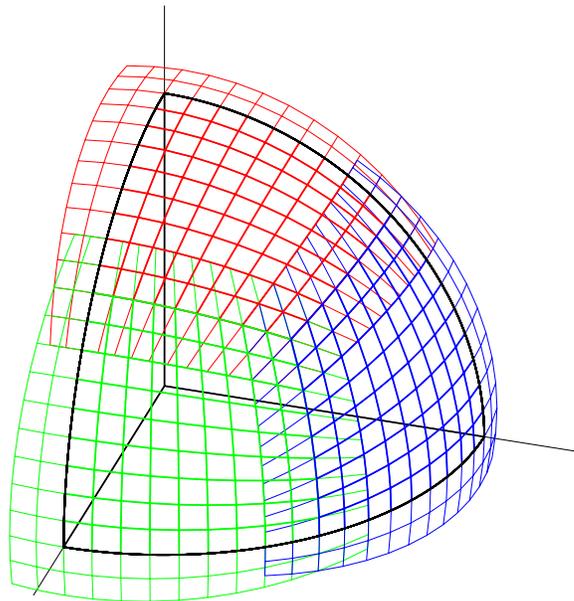


FIG. 1 This figure shows a multiple-grid-patch system covering the  $(+, +, +)$  octant of  $S^2$  with 3 patches, at an angular resolution of  $\Delta\rho\sigma = 5^\circ$ . The  $+z$ ,  $+x$ , and  $+y$  patches are shown in red, green, and blue respectively. The patch's nominal grids (shown in thick lines) just touch; their ghost zones (which are 2 points wide, and are shown in thin lines) overlap.

### C. Choice of Tensor Basis

For numerical solution I decompose the  $3 + 1$  Einstein equations into coordinate components with respect to a tensor basis. This raises the question of what basis should be used in each patch. There are two natural approaches:

#### Use the same (typically Cartesian) basis in each patch

This approach avoids the need for any change of basis in the interpatch interpolation. By using a Cartesian basis, it may also make it easier to interface with other numerical relativity software which uses this type of tensor basis.

However, this approach has the disadvantage that a nontrivial (frame) transformation is required at each grid point to convert grid finite differences into approximate coordinate partial derivatives. This may both slow down the code, and possibly introduce time evolution instabilities, or at least complicate stability analysis. It also makes it difficult to reuse existing software modules for the evolution.

Using a Cartesian basis also makes it harder to treat the radial coordinate specially for such things as special finite differencing near the excision boundary, outer boundary conditions, etc.

#### Use each patch's own $(r, \rho, \sigma)$ coordinates to define its tensor basis

Because this approach uses a *different* tensor basis in each patch, it requires a change of basis in the interpatch interpolation. (This is discussed in detail in section II.E.) However, this extra computation is only needed at the patch boundaries – in the

interior of each patch there is no extra overhead, and existing unigrid software can potentially be reused with little or no change.

This approach also makes it easy to treat the radial coordinate specially in finite differencing and/or in the continuum formulation of the equations.

In this work I use the second approach: in each patch I use that patch’s own  $(r, \rho, \sigma)$  coordinates to define the tensor basis.

#### D. Synchronizing Ghost Zones

As described in section II.A, I use the usual “ghost zone” technique for handling finite differencing near the patch boundaries. I refer to the non-ghost-zone part of a patch’s grid as its “nominal” grid, and to the process of computing values for a set of grid functions in all ghost zones of all patches as “synchronizing” these grid functions. This must be done for all grid functions to which finite difference molecules will be applied, i.e. in practice, for all grid functions representing dynamical field variables.

Each patch is a rectangular solid (“cuboid”) in its own  $(r, \rho, \sigma)$  coordinates, so it has 6 ghost zones: 2 radial (inner and outer) and 4 angular ( $\rho_{\min}$ ,  $\rho_{\max}$ ,  $\sigma_{\min}$ , and  $\sigma_{\max}$ ). There are three types of ghost zones, each with corresponding synchronization techniques:

**A radial ghost zone** is synchronized by extrapolation from the nominal grid. I describe this in detail in appendix A.

**A “symmetry” (angular) ghost zone** is one where the patch boundary is a discrete symmetry plane of the spacetime, such as the  $x = 0$ ,  $y = 0$ , or  $z = 0$  plane in the octant-symmetry example of figure 1. In this case the symmetry operation defines a mapping from the ghost zone into the nominal grid of some (possibly other) patch, so the ghost-zone grid function values can be computed by copying from the symmetry-image grid points.

**An “interpatch” (angular) ghost zone** is one where the ghost zone lies within the nominal grid of some other patch.<sup>7</sup> In this case the ghost-zone grid function values can be computed by interpolating from the neighboring patch. Because all patches share a common radial coordinate, and (by construction) adjacent patches share the angular coordinate perpendicular to their mutual boundary, the (1-D) interpatch interpolation is done (independently in each line of constant perpendicular coordinate and  $r$ ) only in the direction parallel to the boundary.

Because off-centered interpolations are relatively inaccurate, and have large phase errors which may lead to finite differencing instabilities in the time evolution, I try to keep the interpatch interpolations as centered as possible.

A major complication in ghost zone synchronization is that ghost zones have *corners*, i.e. there are ghost-zone points which are outside the nominal grid in more than one dimension. It’s quite tricky to ensure that the corner ghost-zone points are properly updated while still keeping the interpatch interpolations as centered as possible near the patch corners. I use the

---

<sup>7</sup> To ensure that this is always so, adjacent patches’ nominal grids must either overlap, just touch, or be separated by a gap of at most one grid spacing.

3-phase algorithm of Thornburg (2004, appendix A), generalized to also handle radial ghost zones and non-scalar grid functions, to synchronize ghost zones. I describe the resulting algorithm in detail in appendix A. It turns out that if the interpatch-interpolation molecule is 4-point or smaller, then all the interpatch interpolations can in fact be kept centered. If the interpatch-interpolation molecule is 5-point or larger, then all the interpatch interpolations can still be kept centered except in the immediate neighborhood of a radial line where 3 patches meet.

## E. Change of Tensor Basis

In numerical relativity we need to deal with tensors and other non-scalar grid functions, so interpatch symmetry operations and interpolations entail a change of basis (step 4 in the synchronization algorithm of figure 9).

Without loss of generality I assume that all the fields are known at a point (event) in some patch  $\mathbf{p}$  in some other patch  $\mathbf{q}$ 's  $(r, \rho, \sigma)$  basis  $\{x^i(\mathbf{q})\}$ , and we wish to transform them to patch  $\mathbf{p}$ 's  $(r, \rho, \sigma)$  basis  $\{x^a(\mathbf{p})\}$ . I define the transformation matrices

$$X^a{}_i = \frac{\partial x(\mathbf{p})^a}{\partial x(\mathbf{q})^i} \quad (3a)$$

$$Y^i{}_a = \frac{\partial x(\mathbf{q})^i}{\partial x(\mathbf{p})^a} \quad (3b)$$

$$Y^i{}_{ab} = \frac{\partial^2 x(\mathbf{q})^i}{\partial x(\mathbf{p})^a \partial x(\mathbf{p})^b} \quad (3c)$$

These are straightforward to compute analytically from the coordinate definitions (1).

Of the BSSN field variables,  $\alpha$  and  $K$  are scalars,  $\beta^i$  is a tensor and transforms as

$$\beta^a(\mathbf{p}) = X^a{}_k \beta^k(\mathbf{q}) \quad (4)$$

and  $\phi$  is (proportional to) the logarithm of a tensor density and transforms as

$$\phi(\mathbf{p}) = \phi(\mathbf{q}) + \frac{1}{6} \log |Y^{\parallel}| \quad (5)$$

where  $\parallel$  refers to each patch's angular coordinate parallel to  $\mathbf{p}$  and  $\mathbf{q}$ 's mutual interpatch boundary.  $\tilde{g}_{ij}$  and  $\tilde{A}_{ij}$  are tensor densities, and transform as

$$\tilde{g}_{ab}(\mathbf{p}) = |Y^{\parallel}|^{-2/3} Y^i{}_a Y^j{}_b \tilde{g}_{ij}(\mathbf{q}) \quad (6a)$$

$$\tilde{A}_{ab}(\mathbf{p}) = |Y^{\parallel}|^{-2/3} Y^i{}_a Y^j{}_b \tilde{A}_{ij}(\mathbf{q}) \quad (6b)$$

$\tilde{\Gamma}^i$  isn't a tensor or tensor density, so it transforms in a more complicated manner (I outline the derivation of this in appendix B),

$$\begin{aligned} \tilde{\Gamma}^a(\mathbf{p}) = & |Y^{\parallel}|^{2/3} X^a{}_k \tilde{\Gamma}^k(\mathbf{q}) + X^a{}_k Y^k{}_{bc} \tilde{g}^{bc}(\mathbf{p}) \\ & - 2 |Y^{\parallel}|^{2/3} X^a{}_k \tilde{g}^{k\ell}(\mathbf{q}) \partial_\ell \phi(\mathbf{q}) + 2 \tilde{g}^{ab}(\mathbf{p}) \partial_b \phi(\mathbf{p}) \end{aligned} \quad (7)$$

The first two terms here are straightforward to implement, but the 3rd and 4th terms are problematic in several ways:

- The 3rd term of (7) involves the inverse conformal metric  $\tilde{g}^{kl}(\mathbf{q})$ . Since this is a patch- $\mathbf{q}$  quantity, it's presumably only known on the patch- $\mathbf{q}$  grid, and so must be interpolated to the (incommensurate) patch- $\mathbf{p}$  grid. This requires either keeping  $\tilde{g}^{ij}$  as a grid function so that it can be interpolated (instead of the more economical-of-memory alternative of only storing its value at the current grid point), or interpolating  $\tilde{g}_{ij}$  and then computing  $\tilde{g}^{ij}$  from that. In my code I do the latter.
- The 3rd term of (7) also involves  $\partial_\ell\phi(\mathbf{q})$ . Since  $\phi$  is a dynamic field variable, this partial derivative must be computed by finite differencing. This is difficult because this is a patch- $\mathbf{q}$  derivative, but it's needed at a patch- $\mathbf{p}$  grid point. The computation can be done by either using a differentiating interpolator<sup>8</sup>, or (less efficiently) by storing the  $\partial_\ell\phi(\mathbf{q})$  as grid functions and interpolating them directly. (This computation is being done at a point which lies within patch  $\mathbf{q}$ 's nominal grid, so there's no problem in computing  $\partial_\ell\phi(\mathbf{q})$ .) For historical reasons, I use the latter technique in my current code.
- The 4th term of (7) involves  $\partial_b\phi(\mathbf{p})$ , which is particularly difficult to compute, because (as in the 3rd term) this derivative must be computed by finite differencing, and this must be done at a point which lies in patch  $\mathbf{p}$ 's ghost zone. There are two ways to compute this term:
  - $\partial_b\phi(\mathbf{p})$  can be computed by straightforward finite differencing of  $\phi(\mathbf{p})$  if  $\phi$  has a ghost zone twice as wide as that of  $\tilde{\Gamma}^i$  and the other BSSN field variables, so that the computation point (in  $\tilde{\Gamma}^i$ 's ghost zone) is still guaranteed to be surrounded by at least a molecule-sized neighborhood of  $\phi(\mathbf{p})$  grid points.
  - Alternatively, using the analytically-known  $\phi(\mathbf{p}) \leftrightarrow \phi(\mathbf{q})$  transformation (5),  $\partial_b\phi(\mathbf{p})$  can be rewritten in terms of the  $\partial_\ell\phi(\mathbf{q})$ , which can be computed using the methods described above for the 3rd term of (7).<sup>9</sup> This avoids having to introduce double-width ghost zones for  $\phi$ .

Because it minimizes the amount of interpolation to be done (and thus – I hope – the chances of finite differencing instabilities in the time evolution), I use the double-width ghost zone approach for all the numerical results presented here. In particular (given the 5 point finite difference molecules described in section III), I use angular ghost zones which are 2 grid points wide for most grid functions, and 4 grid points wide for  $\phi$ .

Although this approach is conceptually simple, it turns out to be somewhat awkward to retrofit to an existing code, as large parts of the code generally implicitly assume

---

<sup>8</sup> An interpolator generally works by (conceptually) locally fitting a fitting function (usually a low-degree polynomial) to the data points in a neighborhood of the interpolation point, then evaluating the fitting function at the interpolation point. By evaluating the *derivative* of the fitting function, the  $\partial_\ell\phi(\mathbf{q})$  values can be interpolated very cheaply, using only the input  $\phi(\mathbf{q})$  values which are used anyway for interpolating  $\phi(\mathbf{q})$  to the patch- $\mathbf{p}$  grid.

<sup>9</sup> Actually this is only needed for the perpendicular derivative  $\partial_\perp\phi(\mathbf{p})$ , since the radial and parallel derivatives  $\partial_r\phi(\mathbf{p})$  and  $\partial_{\parallel(\mathbf{p})}\phi(\mathbf{p})$  are taken in directions parallel to the ghost zone, and so can be computed by standard finite differencing techniques without using any  $\phi(\mathbf{p})$  data from outside the ghost zone.

that the ghost-zone width is an inherent property of a grid, rather than (potentially) varying from one grid function to another.

### III. FINITE DIFFERENCING

In this section I describe the finite differencing scheme used to obtain the numerical results presented in section IV. This is a straightforward generalization to 3-D of the 1-D finite differencing scheme I described in Thornburg (1999b).

To allow higher resolution near the black hole while allowing the outer boundary to be placed relatively far away, I place the  $r = \text{constant}$  shells of grid points non-uniformly in  $r$ : they are uniformly spaced in a new (dimensionless) radial coordinate  $w = w(r)$ . The implementation of this nonuniform gridding is identical to that of Thornburg (1999b), with the parameters  $r_0 = 1.5m$ ,  $a = \infty$ ,  $b = 5m$ , and  $c = 100m$ . For these parameters  $w$  qualitatively resembles a logarithmic radial coordinate in the inner part of the grid, and a uniform radial coordinate in the outer part of the grid.

I use the method of lines, with a low-storage variant of the classical 4th order Runge-Kutta time integrator (Blum (1962); Williamson (1980)). For my nonuniform-grid parameters and coordinate conditions (section IV.B), the empirical CFL limit of my code is a Courant number of  $\Delta t/\Delta w = 0.63 \pm 0.01$ ; I use  $\Delta t/\Delta w = 0.5$  for all the numerical results presented here.

For the spatial finite differencing, I first transform all  $(r, \rho, \sigma)$  coordinate partial derivatives into  $(w, \rho, \sigma)$  coordinate partial derivatives (for example,  $\partial_r f = (\partial w/\partial r)\partial_w f$  for any field variable  $f$ ). I then approximate the  $(w, \rho, \sigma)$  derivatives by the usual centered 5-point 4th order 1st or 2nd derivative molecules as appropriate, except that for the shift vector advection derivatives I use offcentered 5-point molecules, upwinded by 1 grid point in the radial direction based on the sign of  $\beta^r$  (which is always positive for the numerical results presented here).

At the inner and outer grid boundaries I use 4th order Lagrange polynomial extrapolation of the nominal-grid field variables to fill in values in the radial ghost zones (steps 1 and 6 in the ghost-zone synchronization algorithm of figure 9). Because of the upwind derivatives in the radial direction, the ghost zones are 2(3) grid points wide at the inner(outer) boundary for most grid functions, or 4(6) grid points wide for  $\phi$ .

I use 5th order (6-point) Lagrange polynomial interpolation for all the interpatch interpolations (step 3 in the ghost-zone synchronization algorithm of figure 9). This gives an  $O((\Delta x)^6)$  truncation error for all the interpolated field variables. However, these errors – and thus the interpolated field variables themselves – are *non-smooth* at the interpolation points (Thornburg (1999a, appendix F)). Because of this non-smoothness, the consistency argument of Choptuik (1991) does *not* apply here: taking numerical derivatives of the interpolated field variables *does* lower the order of accuracy.

For the ADM equations, taking numerical 2nd derivatives of the interpolated field variables gives 4th order accuracy, and my numerical tests (omitted here in the interests of brevity) confirm that the scheme attains this. However, for the BSSN equations the  $\tilde{\Gamma}^i$  interpatch change-of-basis transformation (7) involves numerical 1st derivatives of the BSSN conformal factor  $\phi$ , so the results of this transformation have non-smooth 5th order errors (as well as the usual smooth 4th order errors). Taking numerical 2nd derivatives of  $\tilde{\Gamma}^i$  in the BSSN evolution and constraint equations thus lowers the overall accuracy of my scheme to 3rd order near interpatch boundaries. This could be raised to 4th order by using a higher-

order or smoother (Hermite or spline) interpatch interpolation, but I have only made limited trials of this so far.

Because the inner (excision) grid boundary (placed at  $r = 1.5m$  for all the numerical results presented here) is spacelike, no further boundary conditions are needed there, and I use the usual interior evolution equations to determine the time evolution of the field variables. Because I only use 4th order radial extrapolation, not 5th order, the finite differencing is only 3rd order accurate for 2nd derivatives at the inner and outer grid boundaries. This doesn't degrade the overall 4th order evolution accuracy at the inner boundary because the grid points there are an "outflow" boundary with respect to the evolution causality (Gary (1975); Gustafsson (1971, 1975, 1982); Gustafsson and Kreiss (1979)).<sup>10</sup>

At the outer boundary I use the outer boundary conditions described in appendix C to determine the time derivatives of the BSSN field variables.<sup>11</sup> Unfortunately, there's no reason to think these boundary conditions are either well-posed or constraint-preserving, and outer boundary instabilities seem to be the limiting factor for the evolutions I present in section IV.

Olsson and Petersson (1996) found that some artificial dissipation was necessary to obtain stable evolutions with a two-patch scheme for a simple (linear) 1-D model problem. I haven't included any artificial dissipation in my numerical scheme, though a small amount of dissipation is present inherently due to the time integrator and the upwind finite differencing of the shift vector advection terms. I don't know whether adding artificial dissipation would improve my scheme's stability.

## IV. NUMERICAL RESULTS

In this section I present various sample results from a prototype implementation of the numerical scheme described in this paper. This code is a standalone uniprocessor code, not using any larger relativity toolkit such as Cactus (Goodale *et al.* (2003)). The multiple-patch infrastructure (roughly 10K lines of C++) was quite difficult to design and debug, though I suspect a reimplemention would be somewhat simpler. I found no problems in combining the multiple-patch scheme with "relativity" code machine-generated from a higher-level tensor form.

### A. Initial Data

For the numerical results presented here, I use initial data for the lapse, shift, and BSSN field variables which is a Kerr-coordinate slice of Kerr spacetime, with spin  $J/m^2 = 0.6$ . I use this same slice as the background slice for my outer boundary conditions (appendix C). In Kerr coordinates the horizon is at  $r = (1 + \sqrt{1 - a^2})m = 1.8m$  for this spin; as mentioned above, I place the inner (excision) grid boundary at  $r = 1.5m$ .

---

<sup>10</sup> If it were desirable, there would be no particular problem with using 5th order radial extrapolation to avoid the lower order accuracy at the radial boundaries (Thornburg (1999b)); I have not done this in the present work only for historical reasons.

<sup>11</sup> The errors at the outer boundary are dominated by those from the continuum outer boundary conditions (appendix C), so the 3rd order finite differencing there is unimportant.

## B. Coordinate Conditions

I use a time-independent shift vector for all the numerical results presented here, with  $\beta^i$  set to a suitable (position-dependent) value on the initial slice and not updated thereafter.

I use generalizations of the Bona-Masso slicings (Bona *et al.* (1995)). In particular, I use a slicing condition slightly adapted from the “ $K$ -driver” condition of Alcubierre *et al.* (2003a,b),

$$\partial_t \alpha = -\alpha f(\alpha)(\alpha K - \nabla_i \beta^i) \quad (8)$$

with  $f(\alpha) = A\alpha^n$ . As discussed by Alcubierre *et al.* (2003a, section III.A), these slicings have a gauge propagation speed of

$$c_{\text{lapse}} = \alpha \sqrt{f(\alpha)} \quad (9)$$

I made some evolutions with  $A = 2$ ,  $n = -1$  (the “1 + log” slicing recommended by Alcubierre *et al.* (2003a)), but these suffered from severe gauge instabilities. I use  $A = 2$ ,  $n = 0$  (a variant of harmonic slicing) for all the results reported here.

## C. Numerical Grid Parameters

I have evolved this initial data with a number of different finite differencing grids, as shown in table I. All the results presented here are for data with octant symmetry.

## D. Diagnostics

I use two main diagnostics to assess the code’s accuracy: the energy constraint  $C$ , and the error of the BSSN state vector with respect to its analytical (Kerr) value for my initial data,

$$\delta S \equiv (\delta\phi)^2 + \sum_{i \leq j} (\delta\tilde{g}_{ij})^2 + (\delta K)^2 + \sum_{i \leq j} (\delta\tilde{A}_{ij})^2 + \sum_i (\delta\tilde{\Gamma}^i)^2 \quad (10)$$

where for each BSSN field variable  $f \in \{\phi, \tilde{g}_{ij}, K, \tilde{A}_{ij}, \tilde{\Gamma}^i\}$ ,  $\delta f \equiv f - f_{\text{Kerr}}$ .

I consider a run to “crash” if any BSSN field variable exceeds  $10^{10}$  in magnitude at any grid point. Table I lists the times at which this occurs for each run.

## E. Convergence

As discussed in section III, for infinite resolution and in the absence of outer-boundary effects, my numerical scheme should give 4th order convergence for the energy constraint  $C$  in patch interiors, 3rd order for  $C$  near interpatch boundaries, and 4th order convergence everywhere for the state vector error  $\delta S$ . The numerical results are quite close to this for a considerable period of time. For example, figures 2 and 3 show typical convergence results for the energy constraint  $C$  at  $t = 1000m$  in the 33k-wrmax4, 50k-wrmax4, and 66k-wrmax4 evolutions. The best-fitting convergence exponents are about 3.9 for the patch interiors and 2.9 for the interpatch boundaries.

Figures 4 and 5 show typical convergence results for the state vector error  $\delta S$  at the same time in these same evolutions. The convergence isn’t as good as for  $C$ , but it’s still between 3rd and 4th order everywhere in the grid; the overall best-fitting convergence exponent is about 3.8.

Model	$\Delta w$	Radial Resolution $\Delta r$			Outer Boundary Position		Angular Grid		Outcome
		Inner	$w = 0.12$	Outer	$w_{\max}$	$r_{\max}$	Resolution $\Delta\rho\sigma$	Patch Overlap	
33k-wrmax4	0.03	0.143m	0.204m	2.7m	4	248m	4.5°	$\pm 0$ (1)	crash at $t = 1296m$
50k-wrmax4	0.02	0.095m	0.071m	1.8m	4	248m	3.0°	$\pm 0$ (1)	crash at $t = 1719m$
66k-wrmax4	0.015	0.071m	0.102m	1.3m	4	248m	2.25°	$\pm 0$ (1)	crash at $t = 1747m$
33k	0.03	0.143m	0.204m	2.2m	2	82m	4.5°	$\pm 0$ (1)	crash at $t = 1030m$
50k	0.02	0.095m	0.071m	1.5m	2	82m	3.0°	$\pm 0$ (1)	crash at $t = 991m$
66k	0.015	0.071m	0.102m	1.1m	2	82m	2.25°	$\pm 0$ (1)	crash at $t = 897m$
75k	0.01333	0.063m	0.091m	1.0m	2	82m	2.0°	$\pm \frac{1}{2}$ (2)	crash at $t = 1448m$
33k-overlap3	0.03	0.143m	0.204m	2.2m	2	82m	4.5°	$\pm 3$ (7)	crash at $t = 302m$
50k-overlap3	0.02	0.095m	0.071m	1.5m	2	82m	3.0°	$\pm 3$ (7)	crash at $t = 789m$
66k-overlap3	0.015	0.071m	0.102m	1.1m	2	82m	2.25°	$\pm 3$ (7)	crash at $t = 1307m$
33k-9overlap	0.03	0.143m	0.204m	2.2m	2	82m	4.5°	$\pm 9^\circ$ (5)	crash at $t = 492m$
50k-9overlap	0.02	0.095m	0.071m	1.5m	2	82m	3.0°	$\pm 9^\circ$ (7)	crash at $t = 789m$
66k-9overlap	0.015	0.071m	0.102m	1.1m	2	82m	2.25°	$\pm 9^\circ$ (9)	still running at $t = 540m$
33k-wrmax10	0.03	0.143m	0.204m	2.9m	10	813m	4.5°	$\pm 0$ (1)	crash at $t = 859m$
50k-wrmax10	0.02	0.095m	0.071m	1.9m	10	813m	3.0°	$\pm 0$ (1)	crash at $t = 1511m$
66k-wrmax10	0.015	0.071m	0.102m	1.4m	10	813m	2.25°	$\pm 0$ (1)	still running at $t = 856m$

TABLE I This table shows the grid parameters and run outcomes for the various evolutions. The radial resolution is given first as the radial grid spacing in  $w$ , then as the radial grid resolution  $\Delta r$  at the inner boundary, the  $w = 0.12$  ( $r = 2.19m$ ) position used for the angular convergence plots, and the outer boundary. The “patch overlap” is specified in one of two ways:  $\pm m$  (n) or  $\pm x^\circ$  (n).  $\pm m$  or  $\pm x^\circ$  gives the distance of the patches’ nominal grid boundaries from  $45^\circ$  (either  $m$  angular grid spacings, or  $x$  degrees), while  $n$  gives the number of perpendicular coordinate values common to a pair of overlapping patches. For example, for a model with angular grid resolution  $\Delta\rho\sigma = 3^\circ$ , “ $\pm 3$  (7)” and “ $9^\circ$  (7)” both mean that the patches’ nominal grid boundaries are at  $45^\circ \pm 3\Delta\rho\sigma = 45^\circ \pm 9^\circ$ , so adjacent patches have 7 grid points in common in their perpendicular-to-the-boundary direction. (Thus the 50k-overlap3 and 50k-9overlap models are actually identical.)

## F. Time Dependence

Figure 6 shows the convergence of the energy constraint  $C$  and the state vector error  $\delta S$ , at the fixed radial position  $w = 0.12$  ( $r = 2.19m$ ), as a function of time. After an initial transient lasting about  $250m$ , the evolution appears to settle down to an almost-stationary state (apart from the error waves described below) for some time. Both  $C$  and  $\delta S$  generally grow with time, but remain quite small in magnitude ( $\ll 1$ ) throughout the first  $t = 1000m$  of the evolution. ( $\delta S$  seems to grow at a faster rate than  $C$ ; this seems to be a gauge instability (Alcubierre and Massó (1998); Alcubierre and Schutz (1996).))

Notice that although the energy constraint  $C$  is substantially larger at the interpatch boundaries than in the patch interiors (this is visible in figure 2), this difference doesn’t grow with time, in fact it even decreases a bit. This suggests that the the interpatch interpolation is not introducing any instability into the evolution.

A major exception to the “roughly stationary state” in figure 6 is the sharp spikes in  $C$  at roughly  $t = 440m, 680m, 880m$ , and succeeding times. Figure 7 shows that each “spike” is actually the passage of an error wave which originates at the outer boundary and propagates inward at approximately the speed of light. (The error waves are spaced roughly  $250m$  apart, matching the speed-of-light propagation time from the outer boundary in to the black hole.) At late times there are also visible error waves propagating outward from

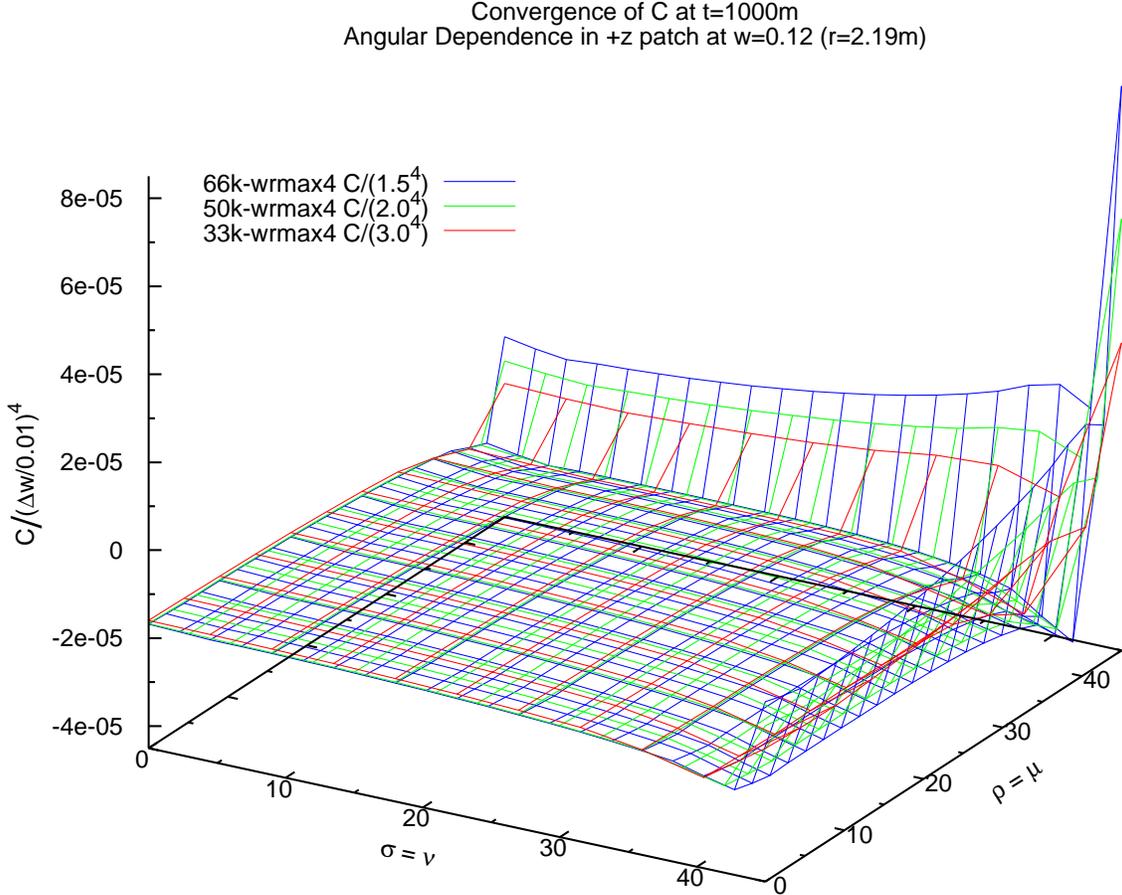


FIG. 2 This figure shows the convergence of the energy constraint  $C$  for the  $+z$  patch of the  $w = 0.12$  ( $r = 2.19m$ ) shell of grid points at  $t = 1000m$ . The horizontal axes show the angular coordinates in degrees (so  $(0, 0)$  is the  $z$  axis and  $(45, 45)$  is the “triple point” where the 3 patches meet in figure 1). The  $z$  axis shows the energy constraint (scaled by the 4th power of the resolution) for the 33k-wrmax4, 50k-wrmax4, and 66k-wrmax4 models.

the strong-field region, and eventually high-frequency oscillations in  $C$  appear in the outer part of the grid.

### G. Overlapping versus Just-Touching Patches

As discussed in section II.D, in my numerical scheme adjacent patches’ nominal grids must either overlap, just touch, or be separated by a gap of one grid spacing.

Figure 8 shows a comparison of results from models with adjacent patches just touching, versus models with adjacent patches overlapping by  $\pm 3$  grid points about their mean nominal-grid-boundary position (i.e. models with adjacent patches having 7 grid points in common in their perpendicular-to-the-boundary direction).<sup>12</sup> The just-touching models are

<sup>12</sup> The results for the models with adjacent patches overlapped by a fixed angular distance of  $\pm 9^\circ$  are generally similar to those for the models with adjacent patches overlapped by  $\pm 3$  grid points.

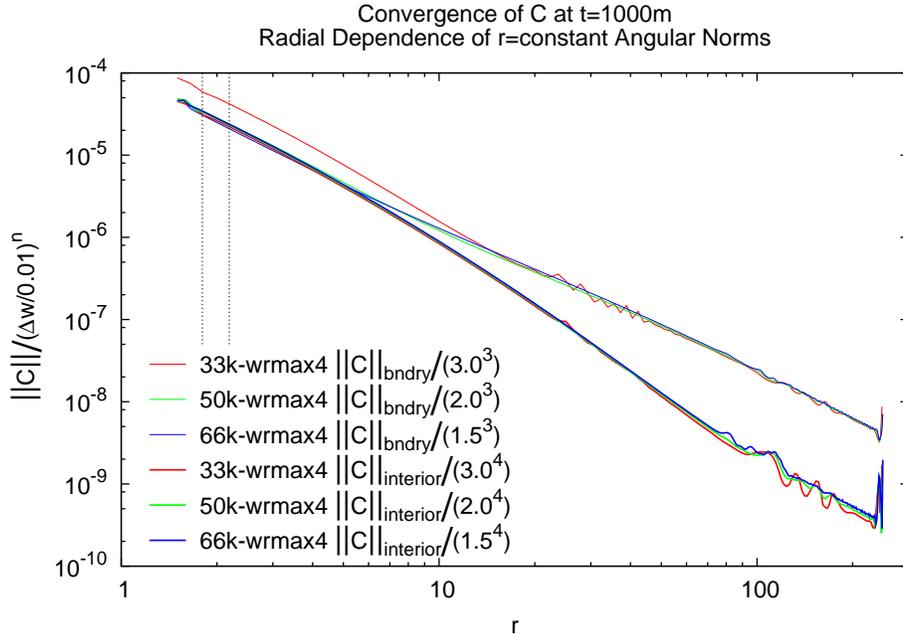


FIG. 3 This figure shows the convergence of various  $r = \text{constant}$  angular RMS-norms of the energy constraint  $C$  at  $t = 1000m$ . The lower [upper] curves show angular RMS-norms of the energy constraint (scaled by the 4th [3rd] power of the resolution) over the patch-interior [interpatch-boundary] grid points in each  $r = \text{constant}$  shell; in all cases  $\infty$ -norms are within an order of magnitude of the RMS-norms shown. The two vertical dotted lines show the horizon position and the  $w = 0.12$  ( $r = 2.19m$ ) position used for the angular convergence plots.

considerably more stable (strictly speaking, less unstable) than the overlapping-patch models. In particular, notice that (apart from error waves similar to those discussed in the previous section)<sup>13</sup> the just-touching models display roughly constant  $C$  for roughly  $500m$ , while the overlapping-patch models display steadily rising  $C$  in the patch interiors over the same time span.

Another interesting pattern visible in table I and figure 8, is that among the overlapping-patch models, the higher-resolution models are more stable (less unstable). It's hard to draw any conclusions about a continuum limit, though, because while the just-touching models show good 4th (3rd) order convergence in the patch interiors (interpatch boundaries), even the highest-resolution overlapping-patch models aren't yet in the asymptotic convergence regime.<sup>14</sup>

<sup>13</sup> The error wave spacing again roughly matches the speed-of-light propagation time from the outer boundary in to the black hole.

<sup>14</sup> The formal convergence exponents for the highest-resolution pair of overlapping-patch models are about 5(4.5) in the patch interiors (interpatch boundaries). Both of these exponents are greater than 4 (the order of the finite differencing scheme), so the models can't be in the asymptotic convergence regime.

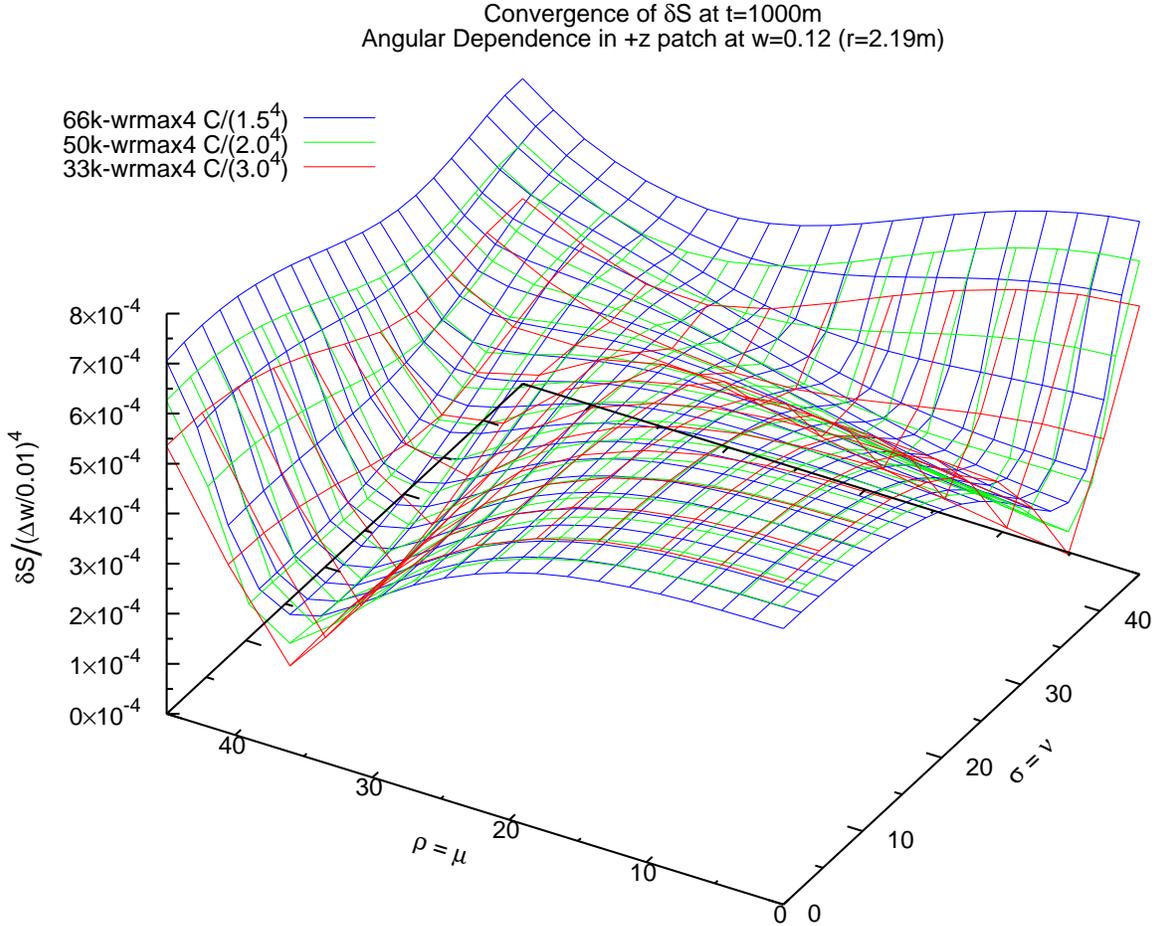


FIG. 4 This figure shows the convergence of the state vector error  $\delta S$  for the  $+z$  patch of the  $w = 0.12$  ( $r = 2.19m$ ) shell of grid points at  $t = 1000m$ . The horizontal axes show the angular coordinates in degrees (so  $(0, 0)$  is the  $z$  axis and  $(45, 45)$  is the “triple point” where the 3 patches meet in figure 1). The  $z$  axis shows the state vector error (scaled by the 4th power of the resolution) for the 33k-wrmax4, 50k-wrmax4, and 66k-wrmax4 models.

## V. DISCUSSION AND CONCLUSIONS

Because excision in Cartesian grids is so difficult, and polar spherical grids suffer from  $z$  axis coordinate singularities, there is growing interest in multiple-patch black hole excision schemes which allow a smooth  $r = \text{constant}$  excision surface. As well as greatly simplifying excision, such schemes also provide a smooth outer boundary, and (by placing the  $r = \text{constant}$  shells of grid points nonuniformly in radius) they can give high resolution close to the black hole while still having the outer boundary relatively far away.

In this paper I present a detailed description of what I believe is the first multiple-patch excision scheme for the full nonlinear Einstein equations in  $3 + 1$  dimensions, together with sample numerical results from a prototype implementation of this scheme. I use the BSSN form of the  $3 + 1$  equations.

I use a “ghost zone” technique to handle the interpatch boundaries, with all the dynamical fields interpolated into each patch’s ghost zones from adjacent patches. I use an

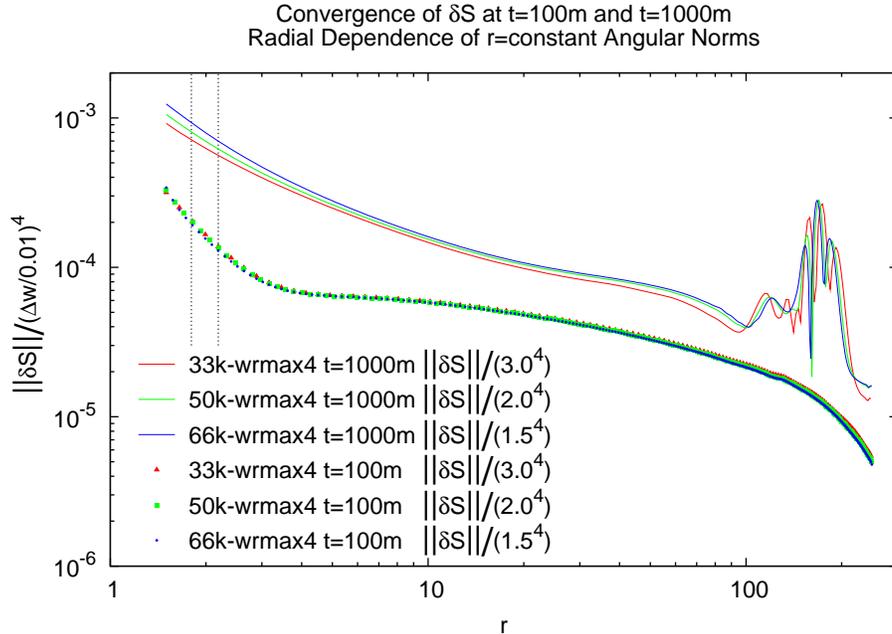


FIG. 5 This figure shows the convergence of various  $r = \text{constant}$  angular RMS-norms of the state vector error  $\delta S$  at  $t = 100m$  and  $t = 1000m$  for the 33k-wrmax4, 50k-wrmax4, and 66k-wrmax4 models. The curves show angular RMS-norms of the state vector error (scaled by the 4th power of the resolution) over all grid points in each  $r = \text{constant}$  shell; in all cases  $\infty$ -norms are within a factor of 3 of the RMS-norms shown. The two vertical dotted lines show the horizon position and the  $w = 0.12$  ( $r = 2.19m$ ) position used for the angular convergence plots.

“inflated-cube” 6-patch system of the form  $\{r \times (6 \text{ angular patches covering } S^2)\}$ . By suitably choosing the angular coordinates, this allows adjacent patches to always share the angular coordinate perpendicular to their mutual boundary, so the interpatch interpolation need only be done in the (angular) dimension parallel to the boundary.

I use each patch’s local coordinates to define its tensor basis, so this basis is different from one patch to another, and hence the interpatch interpolation must include a change of tensor basis. This is straightforward for all the BSSN variables except for the  $\tilde{\Gamma}^i$ . The  $\tilde{\Gamma}^i$  aren’t tensors or tensor densities, so their change-of-basis transformation law includes terms containing spatial derivatives of the BSSN conformal factor  $\phi$ . Computing these derivatives numerically is difficult because this must be done in the ghost zones. I currently solve this problem by using a ghost zone twice as wide for the BSSN conformal factor  $\phi$  as for the other BSSN field variables. This works, but is somewhat cumbersome; it would be interesting to explore the alternative scheme I describe (but have not implemented) where the ghost zones can be the same width for all the field variables.

I have implemented my multiple-patch scheme in a prototype numerical code, using 4th order finite differencing in space and time, and 5th order Lagrange polynomial interpatch interpolation. I use ghost zones which are 4 points wide for the BSSN conformal factor  $\phi$ , and 2 points wide for all the other dynamical field variables. My finite differencing scheme is a straightforward generalization to 3-D of the 1-D finite differencing scheme I described in Thornburg (1999b). The overall accuracy of the scheme is 4th order in the patch interiors,

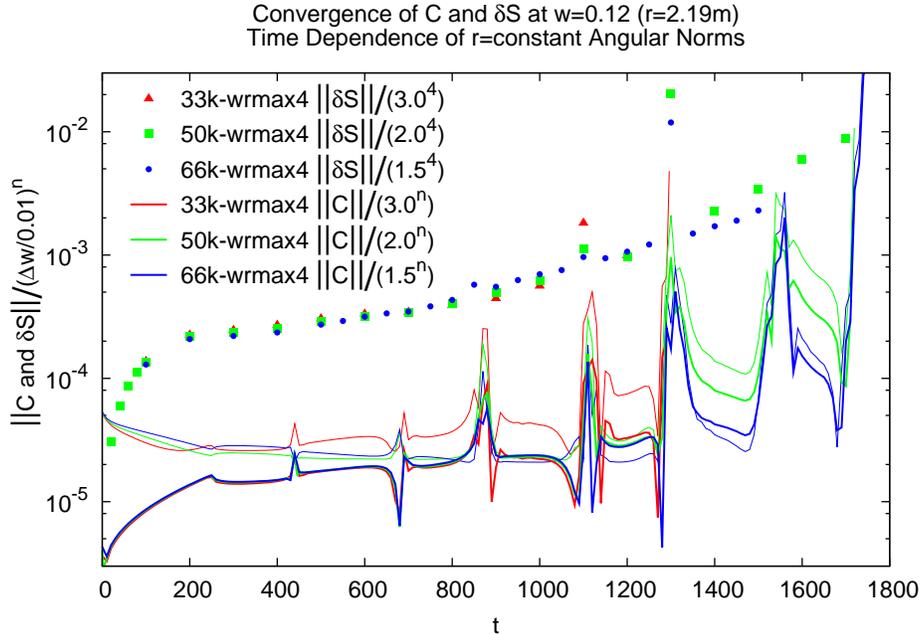


FIG. 6 This figure shows the convergence of various  $r = \text{constant}$  angular RMS-norms of the energy constraint  $C$  and the state vector error  $\delta S$  at the fixed radial position  $w = 0.12$  ( $r = 2.19m$ ), as a function of time, for the 33k-wrmax4, 50k-wrmax4, and 66k-wrmax4 models. The points show angular RMS-norms of the state vector error (scaled by the 4th power of the resolution) over all grid points in each  $r = \text{constant}$  shell. The thick [thin] lines show angular RMS-norms of the energy constraint (scaled by the 4th [3rd] power of the resolution) over all patch-interior [interpatch-boundary] grid points in each  $r = \text{constant}$  shell.

and 3rd order close to the interpatch and radial boundaries (though this could be raised to 4th order everywhere by adjusting the interpolation and radial extrapolation operators).

In tests of the evolution of Kerr initial data with various numerical parameters, this scheme performs quite well, with evolutions maintaining good convergence and preserving the energy constraint near the black hole to better than 1% for  $\gtrsim 1000m$ , and lasting up to  $\gtrsim 1500m$  before crashing. The degradation of accuracy at later times, and the eventual crashes, appear to be due to outer-boundary effects, *not* to any problems inherent to the multiple-patch scheme.

The numerical code I have used for these results is a standalone one, not integrated into any standard numerical-relativity toolkit such as Cactus (Goodale *et al.* (2003)). Such an integration will pose some software-engineering problems, but should have large benefits for furthering the practical use of multiple-patch schemes.

How could my multiple-patch scheme be extended to handle multiple (moving) black holes? The easiest way appears to be to use an inflated-cube set of patches around (and kept dynamically centered on) each (excised) black hole, a Cartesian patch or patches filling the space between the black holes, and probably an outer set of inflated-cube patches outside both black holes.<sup>15,16</sup> This scheme requires multidimensional interpatch interpola-

<sup>15</sup> Using a Cartesian patch or patches between the black holes avoids the  $r = 0$  singularities of the outer set

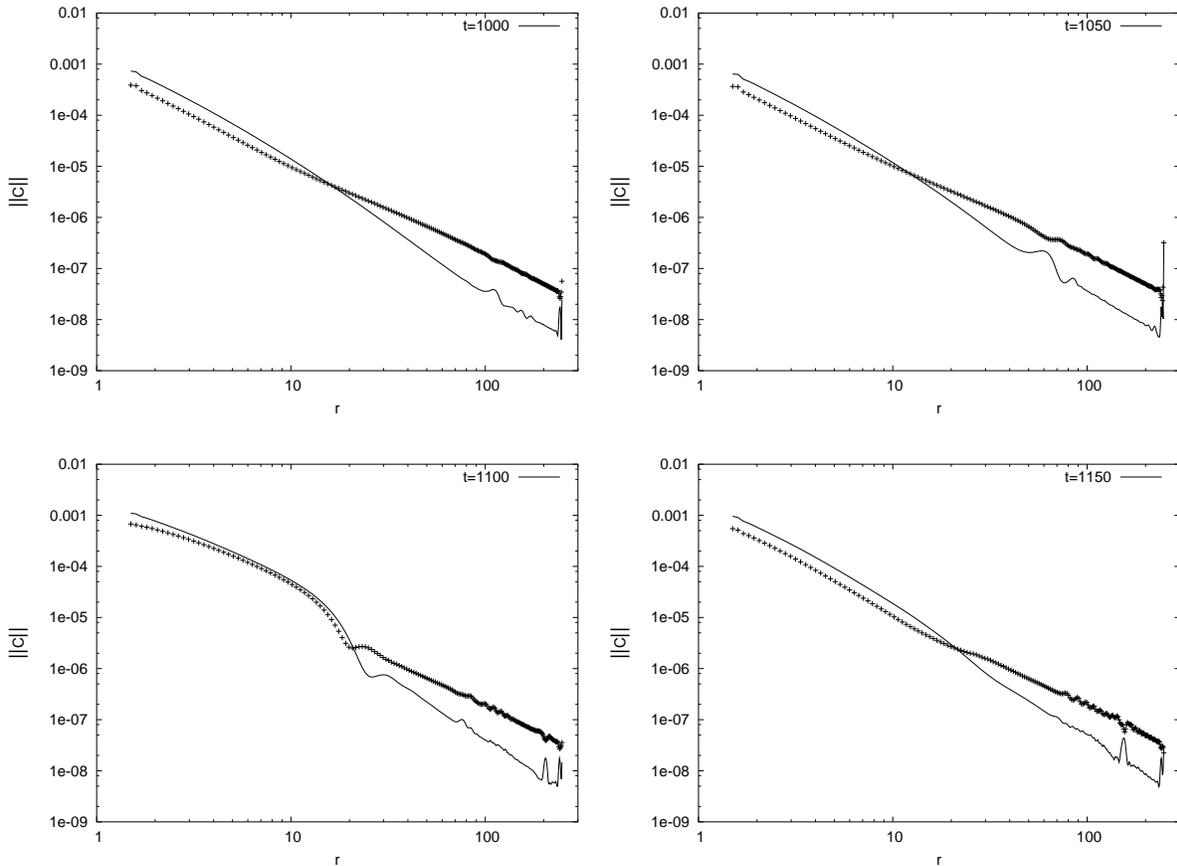


FIG. 7 This figure shows 4 frames from a movie showing the time evolution of angular RMS-norms of the energy constraint  $C$  for the 50k-wrmax4 model. The solid lines (points) show angular RMS-norms over the patch-interior (interpatch-boundary) grid points in each  $r = \text{constant}$  shell; in all cases  $\infty$ -norms are within an order of magnitude of the RMS-norms shown. At  $t = 1000m$  an inward-propagating error wave is visible at  $r \approx 100m$ ; by  $t = 1050m$  it has propagated in to  $r \approx 50m$ . At  $t = 1100m$  the wave is just falling into the black hole, and at  $t = 1150m$  the wave has almost completely vanished. The full movie is available from <http://www.aei.mpg.de/~jthorn/research/mpe/movies/>.

tion between the inflated-cube and Cartesian patches, but the experience of Calabrese and Neilsen (2004b) suggests that this probably doesn't cause any significant problems. Such schemes should be a fruitful area for further research.

## APPENDIX A: Details of the Ghost-Zone Synchronization Algorithm

In this appendix I give a detailed description of my ghost-zone synchronization algorithm.

---

of inflated-cube patches.

<sup>16</sup> Such a scheme is essentially a 3-D inflated-cube analog of the axisymmetric multiple-patch scheme I have previously used for constructing 2-black-hole conformally-flat initial data (Thornburg (1985, 1987)).

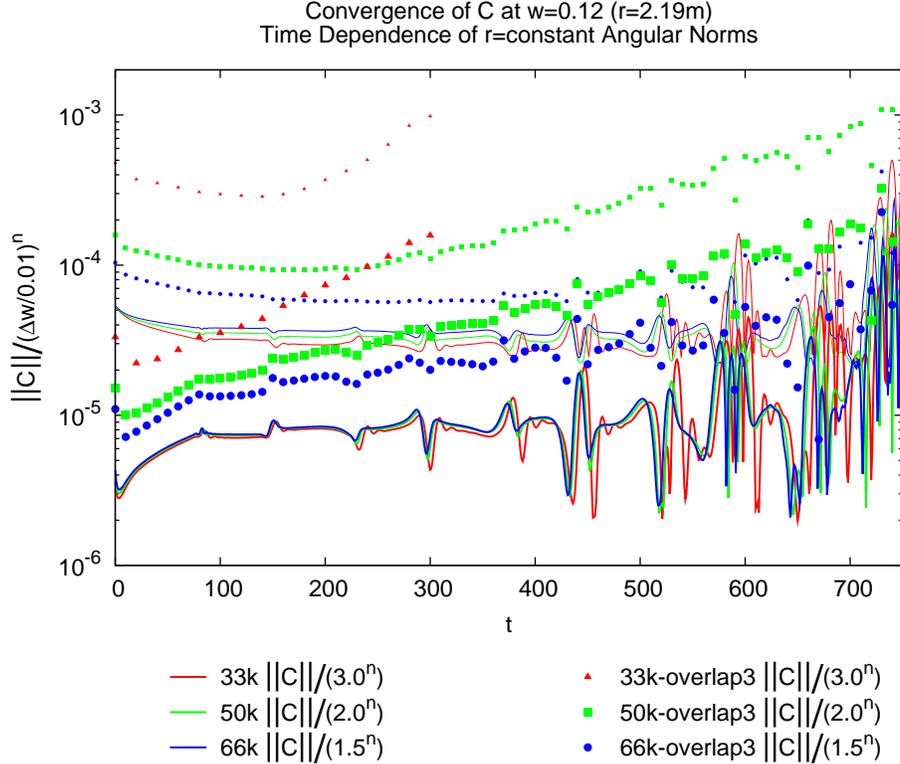


FIG. 8 This figure shows the convergence of various  $r = \text{constant}$  angular RMS-norms of the energy constraint  $C$  at the fixed radial position  $w = 0.12$  ( $r = 2.19m$ ), as a function of time, for the 33k, 50k, and 66k models, compared to the 33k-overlap3, 50k-overlap3, and 66k-overlap3 models. The thick [thin] lines and large [small] points show angular RMS-norms of the the energy constraint (scaled by the 4th [3rd] power of the resolution) over all patch-interior [interpatch-boundary] grid points in each  $r = \text{constant}$  shell.

Figure 9 shows the major steps of the algorithm. The main complexity of the algorithm (steps 2 to 5) is in the handling of the angular ghost zones. Because this is done independently in each  $r = \text{constant}$  angular shell of grid points, for the rest of this appendix I drop the  $r$  coordinate and just describe the angular computations.

Consider an interpatch boundary between patches  $\mathbf{p}$  and  $\mathbf{q}$ , and define  $(\rho, \sigma)$  angular coordinates relative to this boundary,  $\perp$  perpendicular to the boundary and (in each patch)  $\parallel$  parallel to the boundary. (Recall from section II.B that the patches share a common  $\perp$  coordinate, while each has its own  $\parallel$  coordinate.) Corresponding to  $(\perp, \parallel)$ , I define  $(\text{iperp}, \text{ipar})$  as integer grid coordinates to describe the synchronization algorithm, and also as integer array indices for subscripting grid functions.

Since these coordinates are patch-dependent, I use a notation inspired by the C++ programming language, where  $(\mathbf{p}.\text{iperp}, \mathbf{p}.\text{ipar})$  refers to the  $(\text{iperp}, \text{ipar})$  coordinates of patch  $\mathbf{p}$  relative to this boundary. Figure 10 illustrates the geometry of the patches, ghost zones, and coordinates.

1. Radially extrapolate the grid function(s) in the nominal grid to compute their values in the nominal-angular-grid part of each patch's inner and outer (radial) ghost zones.
2. Apply the appropriate symmetry mappings to compute grid function values in the non-angular-corner nominal-radial-grid part of each symmetry ghost zone.
3. Use interpatch interpolation to compute grid function values in the nominal-radial-grid part of each interpatch ghost zone.
4. Do a change-of-basis transformation on each interpatch-interpolated value. (See section II.E for details of this.)
5. Apply the appropriate symmetry mapping to compute grid function values in the angular corners of the nominal-radial-grid part of each symmetry ghost zone.
6. Radially extrapolate the grid function(s) in the nominal grid to compute their values in the angular-ghost-zones part of each patch's inner and outer (radial) ghost zones.

FIG. 9 This figure shows my algorithm for synchronizing ghost zones.

Suppose  $\mathbf{p}$ 's nominal grid is the rectangle

$$(\text{iperp}, \mathbf{p.ipar}) \in [\mathbf{p.iperp}_{\text{nominal,min}}, \mathbf{p.iperp}_{\text{nominal,max}}] \times [\mathbf{p.ipar}_{\text{nominal,min}}, \mathbf{p.ipar}_{\text{nominal,max}}] \quad (\text{A1})$$

and similarly for  $\mathbf{q}$ . Consider one of  $\mathbf{p}$ 's (angular) ghost zones  $\mathbf{gz}$ , which overlaps  $\mathbf{q}$ 's nominal grid (that is, patch- $\mathbf{p}$  grid function values in  $\mathbf{gz}$  are to be obtained by interpolating from patch  $\mathbf{q}$ ). Without loss of generality suppose that the coordinates are oriented such that  $\mathbf{gz}$  lies on  $\mathbf{p}$ 's maximum- $\text{iperp}$  boundary, so  $\mathbf{gz}$ 's extent in the  $\text{iperp}$  direction is  $\text{iperp} \in [\mathbf{p.iperp}_{\text{nominal,max}} + 1, \mathbf{p.iperp}_{\text{nominal,max}} + w]$ , where  $w$  is the ghost zone width.<sup>17</sup>

I define  $\mathbf{gz}$ 's extent in the  $\mathbf{p.ipar}$  direction to depend on the type (symmetry or interpatch) of  $\mathbf{p}$ 's adjacent ghost zones on  $\mathbf{gz}$ 's minimum- $\mathbf{p.ipar}$  and maximum- $\mathbf{p.ipar}$  ends:

- If  $\mathbf{p}$ 's adjacent ghost zone on an end is a symmetry ghost zone, then I define  $\mathbf{gz}$  to *not* include their mutual corner. That is, for example, if  $\mathbf{p}$ 's maximum- $\mathbf{p.ipar}$  ghost zone is a symmetry ghost zone, I define  $\mathbf{gz}$ 's maximum- $\mathbf{p.ipar}$  extent to be  $\mathbf{p.ipar} \leq \mathbf{p.ipar}_{\text{nominal,max}}$ . Part (a) of figure 10 illustrates this case.
- If  $\mathbf{p}$ 's adjacent ghost zone on an end is an interpatch ghost zone, then I define  $\mathbf{gz}$  to include up to the diagonal line of their mutual corner, arbitrarily breaking ties by including the diagonal line itself into the  $\rho$  ghost zone. That is, for example, if  $\mathbf{p}$ 's maximum- $\mathbf{p.ipar}$  ghost zone is an interpatch ghost zone, I define  $\mathbf{gz}$ 's maximum- $\mathbf{p.ipar}$  extent at each  $\text{iperp}$  to be

$$\mathbf{p.ipar} \leq \mathbf{p.ipar}_{\text{nominal,max}} + (\text{iperp} - \mathbf{p.iperp}_{\text{nominal,max}}) - \begin{cases} 0 & \text{if } \perp \text{ is } \mathbf{p}'\text{s } \rho \text{ coordinate} \\ 1 & \text{if } \perp \text{ is } \mathbf{p}'\text{s } \sigma \text{ coordinate} \end{cases} \quad (\text{A2})$$

Part (b) of figure 10 illustrates this case, with the diagonal line shown dashed.

<sup>17</sup>  $w = 2$  for all the numerical results presented here, and for the example in figure 10.

Next, I define  $\mathbf{gz}$ 's "patch interpolation region"  $\mathbf{R}$  in  $\mathbf{q}$ : This is the set of  $\mathbf{q}$  grid points from which  $\mathbf{gz}$ 's interpatch interpolation will use data. Since the interpolation is only done in 1-D along  $\mathbf{q.ipar}$  lines (these run vertically in figure 10),  $\mathbf{R}$  has the same extent in the  $\mathbf{iperp}$  direction (horizontally in figure 10) as  $\mathbf{gz}$ . I define  $\mathbf{R}$ 's extent in the  $\mathbf{q.ipar}$  direction to depend on the type (symmetry or interpatch) of  $\mathbf{q}$ 's adjacent ghost zones on each of  $\mathbf{q}$ 's minimum- $\mathbf{ipar}$  and maximum- $\mathbf{ipar}$  ends:

- If  $\mathbf{q}$ 's adjacent ghost zone on an end is a symmetry ghost zone, then I allow  $\mathbf{gz}$ 's interpolation to use data from that symmetry ghost zone (as well as from  $\mathbf{q}$ 's nominal grid).<sup>18</sup> That is, for example, if  $\mathbf{q}$ 's maximum- $\mathbf{q.ipar}$  ghost zone is a symmetry ghost zone, I define  $\mathbf{R}$ 's maximum- $\mathbf{q.ipar}$  extent to be  $\mathbf{q.ipar} \leq \mathbf{q.ipar}_{\text{nominal,max}} + w$ . So long as the ghost zone is wide enough, this allows  $\mathbf{gz}$ 's interpolations to be kept centered even at  $\mathbf{gz}$  points within an interpolation-molecule-radius of a symmetry boundary. Part (a) of figure 10 illustrates this case.
- If  $\mathbf{q}$ 's adjacent ghost zone on an end is an interpatch ghost zone (this can only happen at a "triple corner" where 3 patches meet), then at present I only allow  $\mathbf{gz}$ 's interpolation to use data from  $\mathbf{q}$ 's nominal grid on that end. That is, for example, if  $\mathbf{q}$ 's maximum- $\mathbf{q.ipar}$  ghost zone is an interpatch ghost zone, I define  $\mathbf{R}$ 's maximum- $\mathbf{q.ipar}$  extent to be  $\mathbf{q.ipar} \leq \mathbf{q.ipar}_{\text{nominal,max}}$ . Part (b) of figure 10 illustrates this case. Notice that for a 4-point or smaller interpatch interpolation molecule, the interpatch interpolation can always be kept centered, while for a 5-point or larger molecule, the interpolation must be off-centered for points close to the triple corner.

If interpatch ghost zones are interpolated in some sequential order in step 3 of the synchronization algorithm of figure 9, it would be possible to keep more of the interpolations centered by expanding  $\mathbf{R}$  to include points of the adjacent interpatch ghost zone if that ghost zone came earlier than  $\mathbf{gz}$  in the sequential order, and thus had already been interpolated. I haven't investigated this possibility in detail, but (by keeping more of the interpolations centered) it might well lead to better numerical stability in the time evolution.

## APPENDIX B: Transformation Properties of the $\tilde{\Gamma}^i$

In this appendix I outline the derivation of the interpatch coordinate-transformation law (7) for the BSSN conformal Christoffel symbols  $\tilde{\Gamma}^i$ .

The transformation law for the non-conformal Christoffel symbols  $\Gamma^i$  is well-known (Misner *et al.* (1973, equation (10.26))):

$$\Gamma^a(\mathbf{p}) = X^a_i \Gamma^i(\mathbf{q}) + X^a_i Y^i_{bc} g^{bc}(\mathbf{p}) \quad (\text{B1})$$

Substituting the identity (valid so long as we're in a 3-D slice)

$$\tilde{\Gamma}^a = e^{4\phi} \Gamma^a + 2\tilde{g}^{ab} \partial_b \phi \quad (\text{B2})$$

<sup>18</sup> Since  $\mathbf{gz}$  is contained in  $\mathbf{q}$ 's nominal grid in the  $\mathbf{iperp}$  direction, these symmetry-ghost-zone points have already been synchronized in step 2 of the synchronization algorithm of figure 9, so it's ok to use their values here (in step 3).

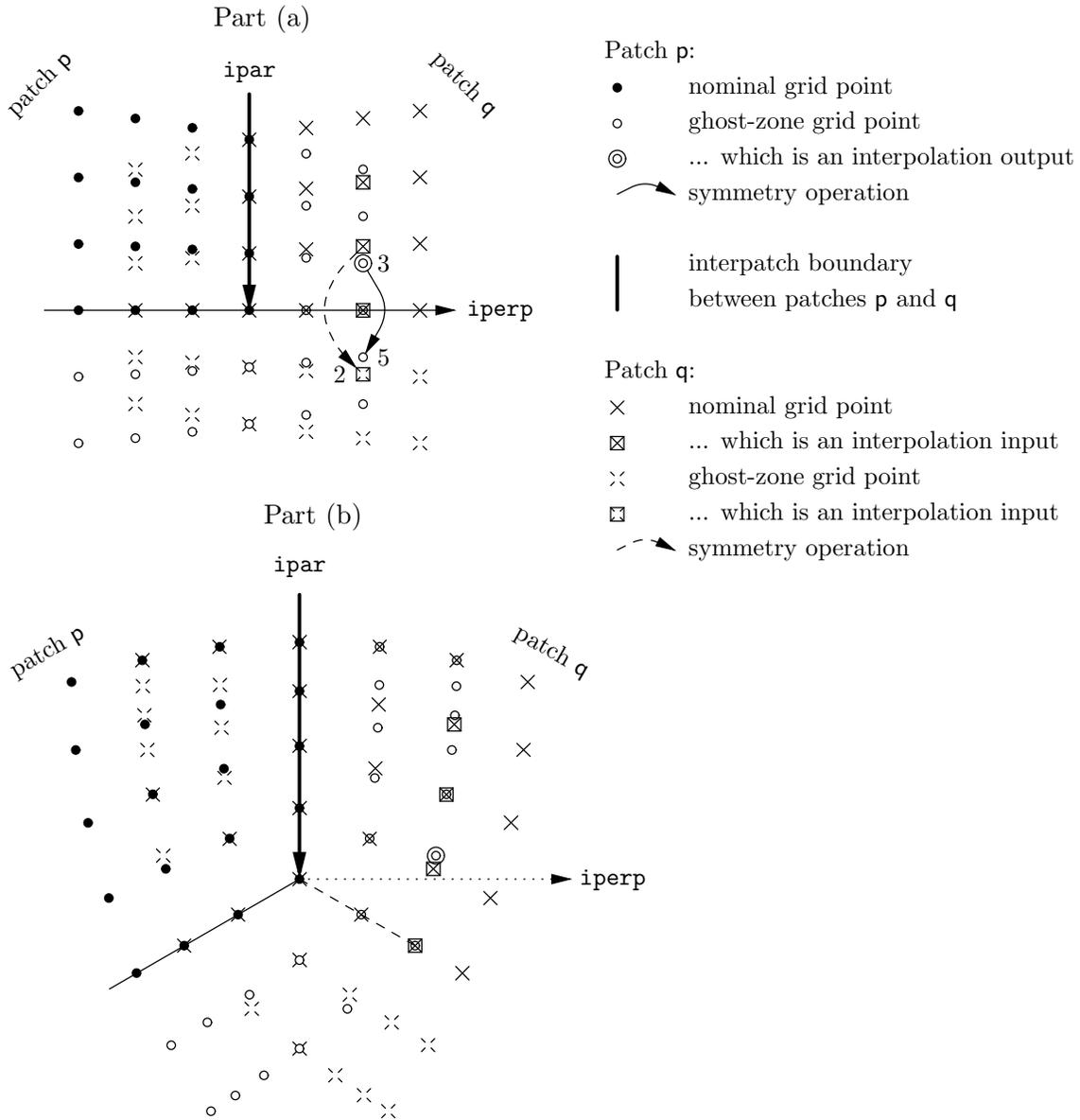


FIG. 10 This figure shows details of the interpatch interpolation process for ghost-zone synchronization. The interpatch boundary between patches  $p$  and  $q$  is shown as a heavy vertical line in each part of the figure. In both parts of the figure,  $iperp$  increases to the right and  $ipar$  increases downwards, though these coordinates aren't orthogonal. Part (a) shows the case where  $p$ 's adjacent ghost zone on the maximum- $ipar$  end is a symmetry ghost zone (for this example, one corresponding to a reflection symmetry across the  $iperp$  axis). Sample output points from various steps of the ghost-zone synchronization algorithm are labelled. Part (b) shows the case where  $p$ 's adjacent ghost zone on the maximum- $ipar$  end is an interpatch ghost zone. The arrows show symmetry operations, the boxed points are interpatch interpolation inputs, and the circled point is an interpatch interpolation output. For the case shown here (a 4-point interpatch interpolation molecule), the interpolation can be kept centered, but for a larger molecule it must be off-centered.

into each side of this, then using the  $\phi$  transformation law (5) and simplifying, gives the final  $\tilde{\Gamma}^i$  transformation law (7).

### APPENDIX C: Outer Boundary Conditions

Basically, I use Sommerfeld outer boundary conditions, applied to each coordinate component of the field variables. However, there's a complication: because the  $(r, \rho, \sigma)$  tensor basis isn't asymptotically Cartesian (for example, asymptotically there are  $r^2$  factors in the angular metric components), I need to transform to the locally Cartesian basis  $\{\bar{x}^i\}$  before applying the usual Sommerfeld condition. In this appendix I use an overbar to denote components in a locally Cartesian basis, as in  $\bar{f}$ .

For each field variable  $f$ , I actually apply a Sommerfeld condition not to  $\bar{f}$  itself, but rather to the deviation  $\delta\bar{f} \equiv \bar{f} - \bar{f}_{\text{background}}$  of  $\bar{f}$  from its (time-independent) value in a specified background slice. The Sommerfeld condition for  $f$  is thus  $\partial_t(r^n \bar{f}) = -c \partial_r(r^n \delta\bar{f})$ , where  $c$  is the propagation speed and  $n$  the falloff exponent, or equivalently

$$\partial_t \bar{f} = -c \left[ \partial_r \delta\bar{f} + \frac{n}{r} \delta\bar{f} \right] \quad (\text{C1})$$

For 3-scalars such as  $\alpha$  and  $K$  I can apply this boundary condition directly. The other BSSN variables need to be transformed to locally Cartesian coordinates for the Sommerfeld condition (C1) to be applied. Proceeding analogously to section II.E, I define the transformation matrices

$$\bar{X}^a{}_i = \frac{\partial x^a}{\partial \bar{x}^i} \quad (\text{C2a})$$

$$\bar{Y}^i{}_a = \frac{\partial \bar{x}^i}{\partial x^a} \quad (\text{C2b})$$

$$\bar{Y}^i{}_{ab} = \frac{\partial^2 \bar{x}^i}{\partial x^a \partial x^b} \quad (\text{C2c})$$

These are straightforward to compute analytically from the coordinate definitions (1). Note that the  $\bar{X}^a{}_i$  and  $\bar{Y}^i{}_a$  matrices are inverses.

The BSSN conformal factor  $\phi$  transforms as

$$\bar{\phi} = \phi - \frac{1}{6} \log |\bar{Y}| \quad (\text{C3})$$

where  $\bar{Y} \equiv \det [\bar{Y}^i{}_a]$ . Because  $Y \equiv Y(x^a)$  is a known coefficient, not a dynamical variable,  $\delta\bar{\phi} = \delta\phi$ , so for purposes of the Sommerfeld condition (C1) I can treat  $\phi$  as a 3-scalar, and apply (C1) directly.

The BSSN  $\tilde{g}_{ij}$  and  $\tilde{A}_{ij}$  transform as

$$\tilde{\tilde{S}}_{ij} = |\bar{Y}|^{2/3} \bar{X}^a{}_i \bar{X}^b{}_j \tilde{S}_{ab} \quad (\text{C4})$$

where  $\tilde{S}_{ij} \in \{\tilde{g}_{ij}, \tilde{A}_{ij}\}$ . Substituting this into both sides of the Sommerfeld condition (C1), multiplying both sides by  $|\bar{Y}|^{-2/3} \bar{Y}^i{}_c \bar{Y}^j{}_d$  and summing over  $i$  and  $j$ , and simplifying, then gives the boundary condition

$$\partial_t \tilde{S}_{ij} = -c \left[ \partial_r \delta \tilde{S}_{ij} + \frac{n}{r} \delta \tilde{S}_{ij} + \bar{Z}^k{}_i \delta \tilde{S}_{kj} + \bar{Z}^k{}_j \delta \tilde{S}_{ki} + \frac{2}{3} (\partial_r \log |\bar{Y}|) \delta \tilde{S}_{ij} \right] \quad (\text{C5})$$

where  $\bar{Z}^a_b \equiv \bar{Y}^i_b \partial_r \bar{X}^a_i$ .

The BSSN  $\tilde{\Gamma}^i$  transform in a more complicated way: Applying the interpatch transformation law (B1) to the coordinates  $x^a(\mathbf{p})$  and  $\tilde{x}^i(\mathbf{q})$ , and using the identity (B2) and its inverse in each of these coordinate systems, gives the transformation law

$$\tilde{\Gamma}^i = |\bar{Y}|^{-2/3} \left[ \bar{Y}^i_a \tilde{\Gamma}^a - 2\bar{Y}^i_a \tilde{g}^{ab} \partial_b \phi + 2\bar{Y}^i_a \bar{Y}^j_b \tilde{g}^{ab} \partial_j \phi - \bar{Y}^i_{ab} \tilde{g}^{ab} + \frac{1}{3} \bar{Y}^i_a \bar{Y}^j_b (\partial_j \log |\bar{Y}|) \tilde{g}^{ab} \right] \quad (\text{C6})$$

Substituting this into both sides of the Sommerfeld condition (C1), multiplying both sides by  $|\bar{Y}|^{2/3} \bar{X}^c_i$  and summing over  $i$ , and simplifying, then gives the boundary condition

$$\begin{aligned} \partial_t \tilde{\Gamma}^c &= 2 (\partial_t \tilde{g}^{bc} \partial_b \phi + \tilde{g}^{bc} \partial_b \partial_t \phi) - 2\bar{Y}^j_b (\partial_t \tilde{g}^{bc} \partial_j \phi + \tilde{g}^{bc} \partial_j \partial_t \phi) \\ &\quad + \bar{X}^c_i \bar{Y}^i_{ab} \partial_t \tilde{g}^{ab} + \frac{1}{3} \bar{Y}^j_b (\partial_j \log |\bar{Y}|) \partial_t \tilde{g}^{bc} \\ &\quad - c \left\{ \bar{W}^c_a \delta \tilde{\Gamma}^a + \partial_r \delta \tilde{\Gamma}^c - 2\bar{W}^c_a (\delta \tilde{g}^{ab} \partial_b \phi + \tilde{g}^{ab} \partial_b \delta \phi) \right. \\ &\quad \quad - 2 (\partial_r \delta \tilde{g}^{bc} \partial_b \phi + \delta \tilde{g}^{bc} \partial_{rb} \phi + \partial_r \tilde{g}^{bc} \partial_b \delta \phi + \tilde{g}^{bc} \partial_{rb} \delta \phi) \\ &\quad \quad + 2\bar{W}^c_a \bar{Y}^j_b (\delta \tilde{g}^{ab} \partial_j \phi + \tilde{g}^{ab} \partial_j \delta \phi) + 2(\partial_r \bar{Y}^j_b) (\delta \tilde{g}^{bc} \partial_j \phi + \tilde{g}^{bc} \partial_j \delta \phi) \\ &\quad \quad + 2\bar{Y}^j_b (\partial_r \delta \tilde{g}^{bc} \partial_j \phi + \delta \tilde{g}^{bc} \partial_{rj} \phi + \partial_r \tilde{g}^{bc} \partial_j \delta \phi + \tilde{g}^{bc} \partial_{rj} \delta \phi) \\ &\quad \quad - \bar{W}^c_{ab} \delta \tilde{g}^{ab} - \bar{X}^c_i \bar{Y}^i_{ab} \partial_r \delta \tilde{g}^{ab} \\ &\quad \quad - \frac{1}{3} \bar{W}^c_a \bar{Y}^j_b (\partial_j \log |\bar{Y}|) \delta \tilde{g}^{ab} - \frac{1}{3} \partial_r \bar{Y}^j_b (\partial_j \log |\bar{Y}|) \delta \tilde{g}^{bc} \\ &\quad \quad - \frac{1}{3} \bar{Y}^j_b (\partial_{rj} \log |\bar{Y}|) \delta \tilde{g}^{bc} - \frac{1}{3} \bar{Y}^j_b (\partial_r \log |\bar{Y}|) \partial_r \delta \tilde{g}^{bc} \\ &\quad \quad + \left( \frac{n}{r} - \frac{2}{3} \partial_r \log |\bar{Y}| \right) \left[ \delta \tilde{\Gamma}^c - 2 (\delta \tilde{g}^{bc} \partial_b \phi + \tilde{g}^{bc} \partial_b \delta \phi) + 2\bar{Y}^j_b (\delta \tilde{g}^{bc} \partial_j \phi + \tilde{g}^{bc} \partial_j \delta \phi) \right. \\ &\quad \quad \quad \left. - \bar{X}^c_i \bar{Y}^i_{ab} \delta \tilde{g}^{ab} - \frac{1}{3} \bar{Y}^j_b (\partial_j \log |\bar{Y}|) \delta \tilde{g}^{bc} \right] \left. \right\} \quad (\text{C7}) \end{aligned}$$

where  $\bar{W}^a_b \equiv \bar{X}^a_i \partial_r \bar{Y}^i_b$  and  $\bar{W}^a_{bc} \equiv \bar{X}^a_i \partial_r \bar{Y}^i_{bc}$ . The right-hand-side terms involving time derivatives can be evaluated by using the other boundary conditions: Differentiating the Sommerfeld condition (C1) for  $\phi$  gives

$$\partial_i \partial_t \phi = -c \left[ \partial_{ir} \delta \phi + \frac{n}{r} \partial_i \delta \phi + \begin{cases} -n/r^2 & \text{if } i = r \\ 0 & \text{otherwise} \end{cases} \right] \quad (\text{C8})$$

while  $\partial_t \tilde{g}^{ab}$  can be evaluated by applying the matrix identity  $d\tilde{g}^{ab} = -\tilde{g}^{ai} \tilde{g}^{bj} d\tilde{g}_{ij}$  to the boundary condition (C5).

For  $\alpha$ ,  $K$ , and  $\phi$ , I use a propagation speed  $c$  given by the Bona-Masso lapse propagation speed (9), while for  $\tilde{g}_{ij}$ ,  $\tilde{A}_{ij}$ , and  $\tilde{\Gamma}^i$  I use a propagation speed given by the outgoing light-cone speed  $\alpha/\sqrt{g_{rr}} - \beta^r$ . I use a falloff power of  $n = 1$  for all the Sommerfeld boundary conditions.

For historical reasons, all the numerical results reported here were obtained using a simple Dirichlet outer boundary condition  $\partial_t \tilde{\Gamma}^i = 0$  instead of the Sommerfeld condition (C7). This probably contributes to the outer boundary instabilities I see, and I plan to switch to the Sommerfeld condition for future computations.

## APPENDIX: Acknowledgments

I thank John Baker, Herbert Balasin, Robert Beig, Bernd Brügmann, Frank Herrmann, Denis Pollney, and Natascha Riahi for assistance with deriving the BSSN equations and the  $\tilde{\Gamma}^i$  interpatch transformations. I thank Miguel Alcubierre, Ian Hawke, Frank Löffler, Denis Pollney, Erik Schnetter, and Virginia Vitzthum for valuable advice and assistance.

I thank Goldie Rodgers, Jeremy Thorn, the Alexander von Humboldt foundation, the AEI visitors program, and the AEI postdoctoral fellowship program for financial support. I thank the AEI for hospitality and use of research facilities during my 1998-2000 AEI visits. I thank Jean Wolfgang, Peter Luckham, and Ulrich Kiermayr for assistance with computer facilities. The numerical results presented in this paper were obtained on a variety of computer systems, notably the PEYOTE cluster at the AEI.

## References

- Alcubierre, M., G. Allen, T. W. Baumgarte, C. Bona, D. Fiske, T. Goodale, F. S. Guzmán, I. Hawke, S. Hawley, S. Husa, M. Koppitz, C. Lechner, *et al.*, 2004, *Class. Quantum Grav.* **21**(2), 589.
- Alcubierre, M., B. Brügmann, P. Diener, M. Koppitz, D. Pollney, E. Seidel, and R. Takahashi, 2003a, *Phys. Rev. D* **67**, 084023.
- Alcubierre, M., B. Brügmann, T. Dramlitsch, J. Font, P. Papadopoulos, E. Seidel, N. Stergioulas, and R. Takahashi, 2000, *Phys. Rev. D* **62**, 044034.
- Alcubierre, M., A. Corichi, J. A. González, D. Núñez, and M. Salgado, 2003b, *Classical and Quantum Gravity* **20**(18), 3951.
- Alcubierre, M., and J. Massó, 1998, *Phys. Rev. D* **57**(8), 4511.
- Alcubierre, M., and B. Schutz, 1996, in *The Seventh Marcel Grossmann Meeting: On Recent Developments in Theoretical and Experimental General Relativity, Gravitation, and Relativistic Field Theories*, edited by R. T. Jantzen, G. M. Keiser, and R. Ruffini (World Scientific, Singapore), p. 611.
- Anninos, P., G. Daues, J. Massó, E. Seidel, and W.-M. Suen, 1995, *Phys. Rev. D* **51**(10), 5562.
- Baumgarte, T. W., and S. L. Shapiro, 1999, *Physical Review D* **59**, 024007.
- Berger, M. J., 1982, *Adaptive Mesh Refinement for Hyperbolic Partial Differential Equations*, Ph.D. thesis, Stanford University, University Microfilms #DA 83-01196.
- Berger, M. J., 1986, *SIAM Journal of Scientific and Statistical Computing* **7**(3).
- Berger, M. J., and P. Colella, 1989, *J. Comp. Phys.* **82**, 64.
- Berger, M. J., and J. Olinger, 1984, *Journal of Computational Physics* **53**, 484.
- Bishop, N. T., R. Gómez, L. Lehner, M. Maharaj, and J. Winicour, 1997, *Phys. Rev. D* **56**(10), 6298.
- Blum, E. K., 1962, *Mathematics of Computation* **16**.
- Bona, C., J. Massó, E. Seidel, and J. Stela, 1995, *Phys. Rev. Lett.* **75**, 600.
- Boyd, J. P., 2000, *Chebyshev and Fourier Spectral Methods (Second Edition, Revised)* (Dover Publications, New York).
- Brandt, S., and B. Brügmann, 1997, *Phys. Rev. Lett.* **78**(19), 3606.
- Brown, D. L., G. S. Chesshire, W. D. Henshaw, and D. J. Quinlan, 1997, in *Eighth SIAM Conference on Parallel Processing for Scientific Computing, held in Minneapolis, Minnesota, 14-17 March 1997*, URL <http://www.llnl.gov/CASC/Overture/>.

- Calabrese, G., L. Lehner, D. Neilsen, J. Pullin, O. Reula, O. Sarbach, and M. Tiglio, 2003a, *Class. Quant. Grav* **20**, L245.
- Calabrese, G., L. Lehner, O. Reula, O. Sarbach, and M. Tiglio, 2003b, eprint gr-qc/0308007.
- Calabrese, G., and D. Neilsen, 2004a, personal communication to Jonathan Thornburg.
- Calabrese, G., and D. Neilsen, 2004b, *Physical Review D* **69**, 044020 (21 pages).
- Chesshire, G. S., and W. D. Henshaw, 1990, *Journal of Computational Physics* **90**(1), 1.
- Chesshire, G. S., and W. D. Henshaw, 1994, *SIAM Journal of Scientific Computing* **15**(4), 819.
- Choptuik, M. W., 1986, *A Study of Numerical Techniques for Radiative Properties in General Relativity*, Ph.D. thesis, University of British Columbia.
- Choptuik, M. W., 1989, in *Frontiers in Numerical Relativity*, edited by C. Evans, L. Finn, and D. Hobill (Cambridge University Press, Cambridge, England), pp. 206–221.
- Choptuik, M. W., 1991, *Phys. Rev. D* **44**, 3124.
- Gary, J., 1975, *Boundary Conditions for the Method of Lines Applied to Hyperbolic Systems*, Technical Report CU-CS-073-75, Department of Computer Science, University of Colorado.
- Gómez, R., L. Lehner, R. Marsa, and J. Winicour, 1998, *Phys. Rev. D* **57**(8), 4778.
- Gómez, R., L. Lehner, P. Papadopoulos, and J. Winicour, 1997, *Classical and Quantum Gravity* **14**(4), 977.
- Goodale, T., G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, and J. Shalf, 2003, in *Vector and Parallel Processing - VECPAR'2002, 5th International Conference, Lecture Notes in Computer Science* (Springer, Berlin).
- Gustafsson, B., 1971, *On the Convergence Rate for Difference Approximations to Mixed Initial Boundary Value Problems*, Technical Report 33, Department of Computer Science, Uppsala University.
- Gustafsson, B., 1975, *Mathematics of Computation* **29**(130), 396.
- Gustafsson, B., 1982, *Journal of Computational Physics* **48**, 270.
- Gustafsson, B., and H. Kreiss, 1979, *J. Comp. Phys.* **30**, 333.
- Gustafsson, B., H.-O. Kreiss, and J. Olinger, 1995, *Time dependent problems and difference methods* (Wiley, New York).
- Kidder, L. E., and L. S. Finn, 2000, *Phys. Rev. D* **62**, 084026, gr-qc/9911014.
- Lehner, L., 2003, personal communication, cited in Calabrese and Neilsen (2004b).
- Lehner, L., D. Neilsen, O. Reula, and M. Tiglio, 2004a, paper in preparation.
- Lehner, L., O. Reula, and M. Tiglio, 2004b, paper in preparation.
- Lichnerowicz, A., 1944, *J. Math Pures et Appl.* **23**, 37.
- Misner, C. W., K. S. Thorne, and J. A. Wheeler, 1973, *Gravitation* (W. H. Freeman, San Francisco).
- Nakamura, T., and K. Oohara, 1989, in *Frontiers in Numerical Relativity*, edited by C. Evans, L. Finn, and D. Hobill (Cambridge University Press, Cambridge, England), pp. 254–280.
- Nakamura, T., and K. Oohara, 1998, in *Numerical Astrophysics 1998 (NAP98) - Proceedings*, eprint gr-qc/9812054, URL <http://www.a.phys.nagoya-u.ac.jp/NAP98-proceedings-official.html>.
- Nakamura, T., K. Oohara, and Y. Kojima, 1987, *Progress of Theoretical Physics Supplement* **90**, 1.
- Olsson, F., and N. A. Petersson, 1996, *Computers and Fluids* **25**(6), 583.
- Petersson, N. A., 1999, *SIAM Journal of Scientific Computing* **20**, 1995, URL <http://www.llnl.gov/CASC/Overture/henshaw/publications/Peters%sonAlgorithm.pdf>.
- Pfeiffer, H. P., L. E. Kidder, M. A. Scheel, and S. A. Teukolsky, 2003, *Comput. Phys. Commun.* **152**, 253.

- Reula, O., 2003, Novel finite-differencing techniques for numerical relativity: Application to black-hole excision, talk at the Advanced School & Conference on Sources of Gravitational Waves, held at the Abdus Salam International Centre for Theoretical Physics, Trieste, Italy, URL [http://surubi.fis.uncor.edu/~reula/Seminars/ICTP/novel\\_f.pdf](http://surubi.fis.uncor.edu/~reula/Seminars/ICTP/novel_f.pdf).
- Rubbert, P. E., and K. D. Lee, 1982, in *Numerical Grid Generation*, edited by J. F. Thompson (North-Holland, New York), pp. 235–252, ISBN 0-444-00757-1.
- Scheel, M. A., and L. Kidder, 2000, Cornell/NCSA-3+1 Group Report, talk at Miniprogram on Colliding Black Holes, held at the Institute for Theoretical Physics, University of California at Santa Barbara, 10-28 January 2004, URL <http://online.kitp.ucsb.edu/online/numrel100/>.
- Schnetter, E., S. H. Hawley, and I. Hawke, 2004, *Classical and Quantum Gravity* **21**(6), 1465.
- Seidel, E., and W.-M. Suen, 1992, *Phys. Rev. Lett.* **69**(13), 1845.
- Shibata, M., 1999, *Prog. Theor. Phys.* **101**, 1199, gr-qc/9905058.
- Shibata, M., and T. Nakamura, 1995, *Phys. Rev. D* **52**, 5428.
- Smarr, L., and J. W. York, Jr., 1978, *Phys. Rev. D* **17**(10), 2529.
- Thompson, J. F., Z. U. A. Warsi, and C. Mastin, 1985, *Numerical Grid Generation: Foundations and Applications* (North-Holland, New York), ISBN 0-444-00985-X.
- Thornburg, J., 1985, *Coordinates and Boundary Conditions for the General Relativistic Initial Data Problem*, Master's thesis, University of British Columbia, Vancouver, British Columbia.
- Thornburg, J., 1987, *Class. Quantum Grav.* **4**, 1119.
- Thornburg, J., 1993, *Numerical Relativity in Black Hole Spacetimes*, Ph.D. thesis, University of British Columbia, Vancouver, British Columbia.
- Thornburg, J., 1999a, *Phys. Rev. D* **59**, 104007 (26 pages).
- Thornburg, J., 1999b, gr-qc/9906022.
- Thornburg, J., 2003, in *The Ninth Marcel Grossman Meeting: On Recent Developments in Theoretical and Experimental General Relativity, Gravitation, and Relativistic Field Theories*, edited by V. G. Gurzadyan, R. T. Jantzen, and R. Ruffini (World Scientific, Singapore), pp. 1743–1744, eprint gr-qc/0012012.
- Thornburg, J., 2004, *Classical and Quantum Gravity* **21**(2), 743, URL <http://stacks.iop.org/0264-9381/21/743>.
- Tiglio, M., 2003, Black hole evolutions: numerical techniques, boundary conditions and constraint preservation, talk at “Gravitation: A Decennial Perspective”, a conference held at the Penn State Center for Gravitational Physics, 8-12 June 2003, URL [http://cgpg.gravity.psu.edu/events/conferences/Gravitation\\_De%ccennial/Proceedings/NR/Tuesday/Tiglio/tiglio\\_talk.pdf](http://cgpg.gravity.psu.edu/events/conferences/Gravitation_De%ccennial/Proceedings/NR/Tuesday/Tiglio/tiglio_talk.pdf).
- Tiglio, M., 2004, personal communication to Jonathan Thornburg.
- Unruh, W. G., 1984, personal communication to Jonathan Thornburg.
- Wald, R. M., 1984, *General Relativity* (The University of Chicago Press, Chicago), ISBN 0-226-87032-4 (hardcover), 0-226-87033-2 (paperback).
- Williamson, J. H., 1980, *Journal of Computational Physics* **35**(1), 48.
- York, J. W., Jr., 1979, in *Sources of Gravitational Radiation*, edited by L. L. Smarr (Cambridge University Press, Cambridge, UK), ISBN 0-521-22778-X, pp. 83–126.