Evolving a puncture black hole with fixed mesh refinement

Breno Imbiriba,[1,2] John Baker,[1] Dae-Il Choi,[1,3] Joan Centrella,[1] David R. Fiske,[2,1] J. David Brown,[4] James R. van Meter,[1] and Kevin Olson[5,6]

[1] *Laboratory for High Energy Astrophysics, NASA Goddard Space Flight Center, Greenbelt, MD 20771 USA*
[2] *Department of Physics, University of Maryland, College Park, MD 20740 USA*
[3] *Universities Space Research Association, 7501 Forbes Boulevard #206, Seabrook, MD 20706 USA*
[4] *Department of Physics, North Carolina State University, Raleigh, NC 27695 USA*
[5] *Goddard Earth Sciences and Technology Center,*
University of Maryland Baltimore County, Baltimore, MD 21250 USA
[6] *Earth and Space Data Computing Division, NASA Goddard Space Flight Center, Greenbelt, MD 20771 USA*

(Dated: Date: 2004/03/01 23:32:40)

We present a detailed study of the effects of mesh refinement boundaries on the convergence and stability of simulations of black hole spacetimes. We find no technical problems. In our applications of this technique to the evolution of puncture initial data, we demonstrate that it is possible to simulaneously maintain second order convergence near the puncture and extend the outer boundary beyond $100M$, thereby approaching the asymptotically flat region in which boundary condition problems are less difficult.

I. INTRODUCTION

Numerical relativity, which comprises the solution of Einstein's equations on a computer, is an essential tool for understanding the behavior of strongly nonlinear dynamical gravitational fields. Current grid-based formulations of numerical relativity feature ~ 17 or more coupled nonlinear partial differential equations that are solved using finite differences in 3 spatial dimensions (3-D) plus time. The physical systems described by these equations generally have a wide range of length and time scales, and realistic simulations are expected to require the use of some type of adaptive gridding in the spacetime domain.

A primary example of the type of physical system to be studied using numerical relativity is the final merger of two comparable mass inspiraling black holes, which is expected to be a strong source of gravitational radiation for ground-based detectors such as LIGO and VIRGO, as well as the space-based LISA. The individual black hole masses M_1 and M_2 set the scales for the binary in the source interaction region, and we can expect both spatial and temporal changes on these scales as the system evolves. The binary must be evolved for a time $t \sim 1000M$, $M \sim M_1 + M_2$, starting from a separation $\sim 10M$ to simulate its final few orbits followed by the plunge and ringdown. This orbital region is surrounded by the wave zone of scale $\sim 100M$, where the outgoing signals take on a wave-like character and can be measured. Accomplishing realistic simulations of binary black hole mergers on even the most powerful computers clearly requires the use of variable mesh sizes over the spatial grid.

Adaptive mesh refinement (AMR) was first applied in numerical relativity to study critical phenomena in scalar field collapse in 1-D [?]; several other related studies have also used AMR, most recently in 2-D [?]. AMR has also been used in 2-D to study the evolution of inhomogeneous cosmologies [? ?]. In the area of black hole evolution, AMR was first applied to a simulation of a Schwarzschild black hole [?]. Fixed mesh refinement (FMR) was used to evolve a short part of a (nonequal mass) binary black hole merger [?] an excised Schwarzschild black hole in an evolving gauge [?], and orbiting, equal mass black holes in a co-rotating gauge [?]. AMR has also been used to set binary black hole initial data [? ?]. The propagation of gravitational waves through spacetime has been carried out using AMR, first using a single 3-D model equation describing perturbations of a Schwarzschild black hole [?] and later in the 3-D Einstein equations [?]. More recently, gravitational waves have been propagated across fixed mesh refinement boundaries, with a focus on the interpolation conditions needed at the mesh boundaries to inhibit spurious reflected waves [?].

Realistic simulations of the final merger of binary black holes are likely to require a hierarchy of grids, using both FMR and AMR. The source region would have the finest grids, and would be surrounded by successively coarser grids, encompassing the orbital region and extending into the wave zone out to a distance $\geq 100M$. Evolving dynamical gravitational fields using such a mesh refinement hierarchy poses a number of technical challenges. For example, the gravitational waves produced by the sources will originate as signals in the near zone and need to cross fixed mesh refinement boundaries to reach the wave zone. In addition, "Coulombic" signals that may vary with time but are not wavelike in character, such as are produced by the gravitational potential around black holes, can stretch across mesh boundaries. Inappropriate interpolation conditions at refinement boundaries can lead to spurious reflection of signals at these interfaces; *c.f.* [?]. Additional complications can arise when the grid refinement is adaptive.

In this paper, we use the evolution of a single Schwarzschild black hole with FMR as a numerical laboratory. The black hole is represented as a puncture without excision. Using a hierarchy of fixed mesh refinements, we are able to resolve the strong field region near the puncture (and demonstrate the convergence of the solution in this region) while locating the outer boundary at $> 100M$. In Sec. II we describe our methodology, including the numerical implementation. The treatment of mesh refinement boundaries is discussed in Sec. III. Black hole evolutions with FMR are presented in Sec. IV; examples are given of evolutions using geodesic slicing, and 1 + log slicing with zero shift. We conclude with a discussion in Sec. V.

II. METHODOLOGY

A. Basic Equations

We use the BSSN form of the ADM equations [? ?]. These equations evolve the quantities

$$\phi = \frac{1}{12}\log\gamma \tag{1a}$$

$$K = \gamma^{ab}K_{ab} \tag{1b}$$

$$\tilde{\gamma}_{ij} = e^{-4\phi}\gamma_{ij} \tag{1c}$$

$$\tilde{A}_{ij} = e^{-4\phi}\left(K_{ij} - \frac{1}{3}\gamma_{ij}K\right) \tag{1d}$$

$$\tilde{\Gamma}^i = \tilde{\gamma}^{ab}\tilde{\Gamma}^i_{ab} \tag{1e}$$

written here in terms of the physical, spatial 3-metric γ_{ij} and extrinsic curvature K_{ij} [?], where all indices range from 1 to 3. These quantities evolve according to

$$\frac{d\phi}{dt} = -\frac{1}{6}\alpha K \tag{2a}$$

$$\frac{dK}{dt} = -\nabla^a\nabla_a\alpha + \alpha\left(\tilde{A}_{ab}\tilde{A}^{ab} + \frac{1}{3}K^2\right) \tag{2b}$$

$$\frac{d\tilde{\gamma}_{ij}}{dt} = -2\alpha\tilde{A}_{ij} \tag{2c}$$

$$\begin{aligned}\frac{d\tilde{A}_{ij}}{dt} &= e^{-4\phi}\left(-\nabla_i\nabla_j\alpha + \alpha R_{ij}\right)^{\mathrm{TF}} \\ &\quad + \alpha\left(K\tilde{A}_{ij} - 2\tilde{A}_{ia}\tilde{A}^a{}_j\right)\end{aligned} \tag{2d}$$

$$\begin{aligned}\frac{\partial\tilde{\Gamma}^i}{\partial t} &= 2\alpha\left(\tilde{\Gamma}^i_{ab}\tilde{A}^{ab} - \frac{2}{3}\tilde{\gamma}^{ia}K_{,a} + 6\tilde{A}^{ia}\phi_{,a}\right) \\ &\quad + \tilde{\gamma}^{kl}\left(-\tilde{\Gamma}^j_{kl}\beta^i{}_{,j} + \frac{2}{3}\tilde{\Gamma}^i_{kl}\beta^j{}_{,j}\right) + \beta^k\tilde{\Gamma}^i{}_{,k} \\ &\quad + \tilde{\gamma}^{jk}\beta^i{}_{,jk} + \frac{1}{3}\tilde{\gamma}^{ij}\beta^k{}_{,kj} - 2\tilde{A}^{ia}\alpha_{,a}\end{aligned} \tag{2e}$$

where here and henceforth the indices of conformal quantities are raised with the conformal metric. The lapse α and shift β^i specify the gauge, the full derivative notation $d/dt = \partial/\partial t - \mathcal{L}_\beta$ is a partial with repect to time minus a Lie derivative, and the notation "TF" indicates the trace-free part of the expression in parenthesis. These quantities are analytically subject to the physical constraints

$$H = R - K_{ab}K^{ab} + K^2 \tag{3a}$$

$$P^i = (\gamma^{im}\gamma^{jn} - \gamma^{ij}\gamma^{mn})\nabla_j K_{mn} \tag{3b}$$

for which we recompute the physical quantities on the right hand sides from the evolved quantites using Eqs. (1).

Note that the covariant derivatives of the lapse in Eq. (2b) and in Eq. (2d) are with respect to the *physical* metric, and are used here for compactness. In the code, this is computed

$$\begin{aligned}\nabla_m\nabla_n\alpha &= \partial_m\partial_n\alpha - 4\partial_{(m}\phi\partial_{n)}\alpha \\ &\quad - \tilde{\Gamma}^k_{mn}\partial_k\alpha + 2\tilde{\gamma}_{mn}\tilde{\gamma}^{kl}\partial_k\phi\partial_l\alpha\end{aligned} \tag{4}$$

only in terms of conformal BSSN quantities, and the index of the covariant derivative is raised in Eq. (2b) with the physical metric. The Ricci tensor in Eq. (2d) is also with respect to the physical metric. We compute it according to the decomposition

$$R_{ij} = \tilde{R}_{ij} + R^\phi_{ij} \tag{5}$$

with

$$\begin{aligned}\tilde{R}_{ij} &= -\frac{1}{2}\tilde{\gamma}^{lm}\tilde{\gamma}_{ij,lm} + \tilde{\gamma}_{k(i}\tilde{\Gamma}^k{}_{,j)} + \tilde{\gamma}^{lm}\tilde{\Gamma}^k_{lm}\tilde{\Gamma}_{(ij)k} \\ &\quad + \tilde{\gamma}^{lm}\left(2\tilde{\Gamma}^k_{l(i}\tilde{\Gamma}_{j)km} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right)\end{aligned} \tag{6}$$

and

$$\begin{aligned}R^\phi_{ij} &= -2\tilde{\nabla}_i\tilde{\nabla}_j\phi - 2\tilde{\gamma}_{ij}\tilde{\nabla}^k\tilde{\nabla}_k\phi \\ &\quad + 4\tilde{\nabla}_i\phi\tilde{\nabla}_j\phi - 4\tilde{\gamma}_{ij}\tilde{\nabla}^k\phi\tilde{\nabla}_k\phi\end{aligned} \tag{7}$$

giving the conformal and remaining pieces of the physical Ricci tensor.

There are many rules of thumb in the community regarding how to incorporate the constraints into the evolution equations, and, in particular, when to use the independently evolved $\tilde{\Gamma}^i$ as opposed to recomputing the equivalent quantity from the evolved metric. We have made our choices manifest in the writing of the equations here; we largely follow the rules set out in [?].

B. Numerical Implementation

For the spatial discretization of Eqs. (2), we take the data to be defined at the centers of the spatial grid cells and use standard $O(\Delta x)^2$ centered spatial differences [?]. To advance this system in time, we use the iterated Crank–Nicholson method with 2 iterations [?], which gives $O(\Delta t)^2$ accuracy. For the evolutions with geodesic slicing, an analytic solution is available; this is used to set the boundary conditions at the outer edge of the grid for these runs. Evolutions using the 1 + log slicing condition employ interpolated Sommerfeld outgoing wave conditions at the outer boundary. Overall, the code is second-order convergent; specific examples of this are given in Sec. IV below.

We explicitly enforce the algebraic constraints that $\tilde{A}_{ij}$ is trace-free and that $\tilde{\gamma} = 1$. We enforce the trace-free condition by replacing the evolved variable with

$$\tilde{A}_{ij} \rightarrow \tilde{A}_{ij} - \frac{1}{3}\tilde{\gamma}^{mn}\tilde{A}_{mn}\tilde{\gamma}_{ij}$$

and we enforce the unit determinant condition by replacing the evolved metric with

$$\tilde{\gamma}_{ij} \rightarrow \tilde{\gamma}^{-1/3}\tilde{\gamma}_{ij}$$

after each ICN interation. Both of these constraints are enforced in all of the runs presented below.

We use the Paramesh package [?] to implement both mesh refinement and parallelization in our code. Paramesh works on logically Cartesian, or structured, grids and carries out the mesh refinement on grid blocks. When refinement is needed, the grid blocks needing refinement are bisected in each coordinate direction, similar to the technique of Ref. [?].

All grid blocks have the same logical structure, with n_x zones in the x-direction, and similarly for n_y and n_z. Thus, refinement of a 1-D block produces two child blocks, each having $n_x n_y n_z$ zones but with zone sizes a factor of two smaller than in the parent block. Refinement can continue as needed on the child blocks, with the restriction that the grid spacing can change only by a factor of two, or one refinement level, at any location in the spatial domain. Each grid block is surrounded by a number of guard cell layers that are used to calculate finite difference spatial derivatives near the block's boundary. These guard cells are filled using data from the interior cells of the given block and the adjacent block.

Paramesh can be used in applications requiring FMR, AMR, or a combination of these. The package takes care of creating the grid blocks, as well as building and maintaining the data structures needed to track the spatial relationships between blocks. Paramesh handles all inter-block communications and keeps track of physical boundaries on which particular conditions are set, guaranteeing that the child blocks inherit this information from the parent blocks. In a parallel environment, Paramesh distributes the blocks among the available processors to achieve load balance, minimize inter-processor communications, and maximize block locality.

This scheme, it is worth noting, provides excellent computational scalabality. Equipped with Paramesh, the scalabality of our code has been tested for up to 256 processors and has demonstrated a consistently good scaling factor for both the unigrid and the FMR runs. For unigrid runs, we started with a uniform Cartesian grid of a certain number of grid cells, a fixed number of timesteps, and a certain number of PEs (Processing Elements), and then increased the number of PEs to run a larger job while the number of grid cells per PE remained constant. In this situation, we expect that the total CPU time taken to run the code should have a scaling factor of 1 under perfect conditions. In reality, communication overhead makes the scalability less than perfect. We define the scaling factor to be the time expected with perfect scaling divided by the actual time taken. Using FMR, we ran the same simulations as in the unigrid case except that a quarter of the computational domain was covered by a mesh with twice the resolution. Despite the more complicated communication patterns, scalability in the FMR runs is comparable to the unigrid runs. The scaling factor of our code is $\sim .92$ for unigrid runs and $\sim .90$ for FMR runs.

For the work described in this paper, we are using FMR. For simplicity, we use the same timestep, chosen for stability on the finest grid, over the entire computational domain; *c.f.* [?]. At the mesh refinement boundaries, we use a single layer of guard cells. Special attention is paid to the restriction (transfer of data from fine to coarse grids) and prolongation (coarse to fine) operations used to set the data in these guard cells, as discussed in the next section.

III. TREATMENT OF REFINEMENT BOUNDARIES

A. Guard Cell Filling

As a general rule, guard cells at mesh refinement boundaries should be filled to third order accuracy for an evolution code such as ours to maintain overall second order accuracy.[1] The current version of our code uses a third order guard cell filling scheme that is now included with the standard Paramesh package. This guard cell filling proceeds in three steps. The first step is a restriction operation in which interior fine grid cells are used to fill the coarse grid guard cells. This is depicted for the case of two spatial dimensions in the left side of Fig. 1. The restriction proceeds as a succession of one–dimensional quadratic interpolations, and is accurate to third order in the grid spacing. Note that the 3-cell-wide fine grid stencil used for this step (nine darker points in the figure) cannot be centered on the guardcell (grey square). In each dimension the stencil includes two fine grid cells on one side of the guard cell and one fine grid cell on the other. The stencil is always positioned so that its center is shifted toward the center of the block (assumed in the figure to be toward the upper left). This ensures

[1] In our terminology the "order of accuracy" refers to the order of errors in the grid spacing. Thus, third order accuracy for guard cell filling means that the guard cell values have errors of order Δx^3, where Δx is the (fine) grid spacing. (Note that third order accurate guard cell filling was termed "quadratic" guard cell filling in Ref. [?].) Second order accuracy for the evolution code means that, after a finite evolution time, the field variables have errors of order Δx^2.

that only interior fine grid points, and no fine grid guard cells, are used in this first guard cell filling step. For the second step, the fine grid guard cells are filled by prolongation from the coarse grid using the stencil shown in the right side of Fig. 1. The prolongation operation proceeds as a succession of one–dimensional quadratic interpolations, and is third order accurate. In this case, the coarse grid stencil includes a layer of guard cells (grey squares), which have just been filled in the first step.

After these first two steps the coarse and fine grid guard cells are filled to third order accuracy. However, order of accuracy is not our only concern. In practice, we find that this two–step guard cell filling scheme leads to unacceptably large reflection and transmission errors for waves passing through the interface. These errors can be suppressed by adding a third step to guard cell filling, "derivative matching" at the interface. With derivative matching the coarse grid guard cell values are recomputed so that the first derivatives at the interface, as computed on the coarse grid, match the first derivatives at the interface as computed on the fine grid. The first derivative on the coarse grid is obtained from standard second order differencing using a guard cell and its neighbor across the interface. The first derivatives on the fine grid are computed using guard cells and their neighbors across the interface, appropriately averaged to align with the coarse grid cell centers. This third guard cell filling step maintains third order accuracy and greatly improves the transmission and reflection errors at the interface.

B. Error Analysis

Why should third order guard cell filling be adequate to maintain overall second order accuracy? This is a nontrivial question, and there are certain subtleties that arise in our black hole evolutions. To help pave the way for a better understanding of these issues, let us consider the simple example of a scalar field in one spatial dimension,

$$\dot{\psi} = \pi \tag{8a}$$
$$\dot{\pi} = \psi'' \; . \tag{8b}$$

Here, the dot denotes a time derivative and primes denote space derivatives. These equations can be solved numerically using the twice–iterated Crank–Nicholson algorithm. The fields at timestep $n+1$ are given in terms of the fields at timestep n by

$$\psi_j^{n+1} = \psi_j^n + \Delta t\, \pi_j^n + \frac{\Delta t^2}{2} D^2 \psi_j^n + \frac{\Delta t^3}{4} D^2 \pi_j^n \tag{9a}$$
$$\pi_j^{n+1} = \pi_j^n + \Delta t\, D^2 \psi_j^n + \frac{\Delta t^2}{2} D^2 \pi_j^n + \frac{\Delta t^3}{4} D^4 \psi_j^n \tag{9b}$$

where D^2 is a finite difference operator approximating the second spatial derivative. Consider for the moment a uniform spatial grid. If D^2 is the usual second order accurate centered difference operator, the dominant source of error for ψ_j^{n+1} comes from the term proportional to Δt^3. This term has the "wrong" numerical coefficient as compared to the Taylor series expansion of the exact solution. The dominant sources of error for π_j^{n+1} come from the term proportional to Δt^3, which also has the "wrong" numerical coefficient, and from the second order error in $D^2\psi_j^n$. For a uniform grid the dominant error in $D^2\psi_j^n$ is $D^4\psi_j^n\,\Delta x^2/12$, so the leading errors for a single timestep are

$$(\psi_j^{n+1})_{err} = \frac{1}{12}\Delta t^3 D^2 \pi_j^n \tag{10a}$$
$$(\pi_j^{n+1})_{err} = \frac{1}{12}\Delta t\,(\Delta t^2 + \Delta x^2) D^4 \psi_j^n \; . \tag{10b}$$

For $\Delta t \sim \Delta x$, each of these one–time–step errors is proportional to Δx^3. If we evolve the initial data to a finite time T, the $\mathcal{O}(\Delta x^3)$ errors accumulate over $N = T/\Delta t$ timesteps resulting in second order errors.[2] Thus, the basic variables ψ and π are second order convergent on a uniform grid.

On a non–uniform grid, guard cell filling introduces errors of order Δx^3 in ψ at grid points adjacent to the boundary. This leads to errors of order Δx in $D^2\psi_j^n$ and $1/\Delta x$ in $D^4\psi_j^n$. From Eq. (9b) we see that in one timestep π can aquire errors of order Δx^2. The concern is that these errors might accumulate over $N = T/\Delta t$ timesteps to yield first order errors. This, in fact, does not happen. Simple numerical experiments show that ψ and π are second order convergent on a non–uniform grid with third order guard cell filling. We can understand this result with the following heuristic reasoning. The numerical algorithm of Eq. (9), as does any mathematically sound numerical scheme, approximates the exact solution of the scalar field equations (Eq. (8)) in which the fields propagate along the light cone. The "bulk" errors displayed in Eq. (10) accumulate along the past light cone to produce an overall error of order $N\,\Delta x^3 \sim \Delta x^2$ at each spacetime point. Errors in guard cell filling, which occur at a fixed spatial location, do not accumulate over multiple timesteps since the past light cone of a given spacetime point will cross the interface (typically) no more than once.

The characteristic fields for the system of Eq. (8) are $\pi + \psi'$ and $\pi - \psi'$. Numerically, the field ψ' is obtained by evolving the ψ, π system for a given number of timesteps then taking the second order accurate numerical first derivative of ψ. A more complete analysis shows that ψ', like π, is subject to one–time–step errors of order Δx^2 due to guard cell filling. Since ψ' takes its value from data along the past light cone, these errors do not

[2] This is a simplification. The dominant error after N timesteps includes other terms of order Δx^2 in addition to the product of N and the one–time–step error. These other terms include, for example, the product of an order N^3 coefficient and a one–time–step error of order Δx^5.

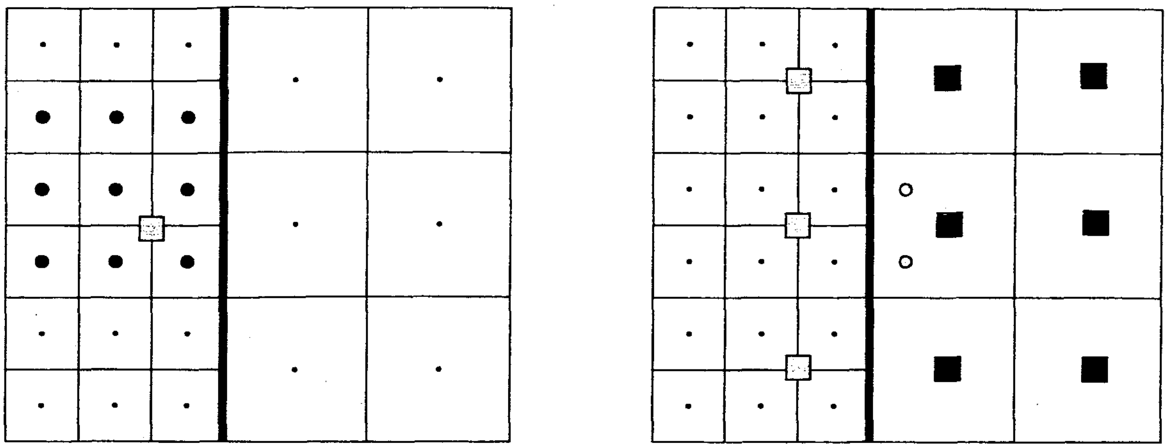

FIG. 1: Guard cell filling in two spatial dimensions. In these pictures, the thick vertical line represents a refinement boundary. The picture on the left shows the first step, in which one of the coarse grid guard cells (grey square) is filled using quadratic interpolation across nine interior fine grid cells (black circles). The other coarse grid guard cells are filled using corresponding stencils of nine interior fine grid cells. The picture on the right shows the second step in which two fine grid guard cells (grey circles) are filled using quadratic interpolation across nine coarse grid values (squares). These coarse grid values include one layer of guard cells. The final step in guard cell filling is to use "derivative matching" to refill the coarse grid guard cells. The asymmetry in the left panel is drawn with the assumption that the fine block's center is toward the top-left of the panel.

accumulate over the course of N timesteps and the error in ψ' remains second order.[3] If, after a finite evolution time, we compute ψ'' (which is the first derivative of ψ'), then the third order guard cell filling errors in ψ' will yield second order errors in ψ''. Thus, we can evolve the system of Eq. (8) for finite time on a grid with mesh refinement, numerically compute the second derivative of ψ, and find that ψ'' is second order accurate.

The BSSN equations, Eq. (2c) and Eq. (2d), are similar to the scalar field equations, Eq. (8), with $\tilde\gamma_{ij}$ playing the role of ψ and $-\tilde A_{ij}$ playing the role of π. This feature was one of the original motivations behind the BSSN system. Note that the term analogous to ψ'' in the $\dot\pi$ equation is the term $\tilde\gamma^{lm}\tilde\gamma_{ij,lm}$ contained in the trace–free part of the Ricci tensor, which appears on the right–hand side of Eq. (2d). Obviously there are many other terms that appear on the right–hand side of the $d\tilde A_{ij}/dt$ equation. We can model the affect of these terms by including a fixed function on the right–hand side of the $\dot\pi$ equation:

$$\dot\psi = \pi \qquad (11a)$$

$$\dot\pi = \psi'' - \chi'' \,. \qquad (11b)$$

We have written the fixed function as the second derivative of χ. For simplicity we choose χ to depend on x only, the most relevant dependence for our consideration of behavior across spatial resolution interfaces. The general solution of this system is then

$$\psi(t,x) = \bar\psi(t,x) + \chi(x) \qquad (12a)$$

$$\pi(t,x) = \bar\pi(t,x) \qquad (12b)$$

where $\bar\psi$, $\bar\pi$ is a solution of the homogeneous wave equation (Eq. (8)).

The extended model system of Eq. (11) can be solved numerically with the discretization

$$\psi_j^{n+1} = \psi_j^n + \Delta t\,\pi_j^n + \ldots \qquad (13a)$$

$$\pi_j^{n+1} = \pi_j^n + \Delta t\,(D^2\psi_j^n - D^2\chi_j) + \ldots \qquad (13b)$$

The higher order terms in Δt, not show here, come from the iterations in our iterated Crank–Nicholson algorithm. It is important to recognize that the χ'' term is expressed as the numerical second derivative of χ_j and not as the discretization of the analytical second derivative, $(\chi'')_j$. The reason for this choice is that $D^2\chi_j$ mimics the effect of the extra terms on the right–hand side of Eq. (2d) which, in our BSSN code, depend on the discrete first and second derivatives of the BSSN variables ϕ and $\tilde\Gamma^i$.

From the discussion of the wave equation (Eq. (8)) we can anticipate the results of numerical experiments with the model system (Eq. (11)) on a non–uniform grid. For arbitrary initial data ψ_j^0, π_j^0, the numerical solution is given by

$$\psi_j^n = \bar\psi_j^n + \chi_j \qquad (14a)$$

$$\pi_j^n = \bar\pi_j^n \qquad (14b)$$

where $\bar\psi_j^n$, $\bar\pi_j^n$ is the numerical solution of the homogeneous wave equation (Eq. (8)) with initial data $\bar\psi_j^0 - \chi_j$, $\bar\pi_j^0$. The order of convergence for ψ_j^n is determined by how

[3] The one–time–step errors for ψ due to guard cell filling are order Δx^3. These errors do accumulate over N timesteps to yield errors of order $N\,\Delta x^3 \sim \Delta x^2$.

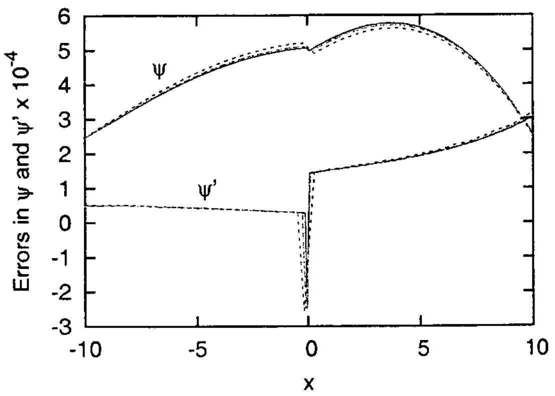

FIG. 2: Convergence tests for ψ and ψ'. The mesh interface is at $x = 0$, with the fine grid on the left and coarse grid on the right. The errors in ψ and ψ' are divided by factors of 4 and 16 for the middle and low resolution cases.

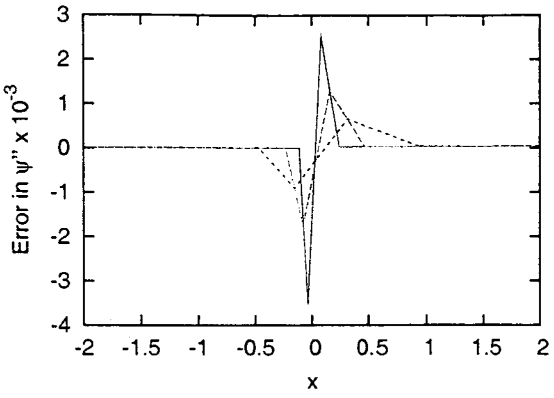

FIG. 3: Convergence test for ψ''. The solid curve is the error in ψ'' for the highest resolution run. As in Fig. 2, the interface is at $x = 0$ and the errors for the middle and low resolution cases are divided by factors of 4 and 16. The grid points adjacent to the interface do not coincide because ψ'' is only first order convergent at these points.

rapidly, as $\Delta x \to 0$, the numerical solution in Eq. (14a) approaches the exact solution $\psi(t,x) = \bar{\psi}(t,x) + \chi(x)$. Since χ_j is simply the projection of the analytic function $\chi(x)$ onto the numerical grid, the term χ_j in the numerical solution (Eq. (14a)) does not contribute any error. We have already determined that on a non–uniform grid $\bar{\psi}^n_j$ approaches $\bar{\psi}(t,x)$ with second order accuracy. Thus, we expect ψ^n_j to be second order convergent.

What about derivatives of ψ? The order of convergence for $D^1\psi^n_j$ is found by comparing the discrete derivative $D^1\psi^n_j = D^1\bar{\psi}^n_j + D^1\chi_j$ to the analytic solution $\psi' = \bar{\psi}' + \chi'$. Again, as we have discussed, $D^1\bar{\psi}^n_j$ approaches $\bar{\psi}'$ with second order errors. It is also easy to see that the numerical derivative $D^1\chi_j$ approaches χ' with second order accuracy. Away from grid interfaces this is obviously true, assuming that D^1 is the standard second order accurate centered difference operator. For points adjacent to a grid interface, guard cell values for ψ are filled with third order errors. These errors lead to second order errors in $D^1\psi^n_j$. Overall then, we expect second order convergence for $D^1\psi^n_j$.

The expected convergence rates for ψ and ψ' are confirmed by the results shown in Fig. 2. For these numerical tests, we chose $\chi(x) = \exp((x-50)/10)$ and initial data

$$\psi(0,x) = 100e^{-(x+10)^2/400} + e^{(x-50)/10} \qquad (15a)$$

$$\pi(0,x) = \frac{1}{2}(x+10)e^{-(x+10)^2/400} \ . \qquad (15b)$$

Each set of curves shows the errors at three different resolutions, $\Delta x = 5/16$, $5/32$, and $5/64$, where Δx is the fine grid spacing. The evolution time is 20.83, corresponding to 200, 400, or 800 timesteps (depending on the resolution) and a Courant factor of 1/3.

The order of convergence for the second derivative of ψ is determined from a comparison of $D^2\psi^n_j = D^2\bar{\psi}^n_j + D^2\chi_j$ and the analytic solution $\psi'' = \bar{\psi}'' + \chi''$. We have seen that $D^2\bar{\psi}^n_j$ approaches $\bar{\psi}''$ with second order accuracy. The situation for $D^2\chi_j$, however, is somewhat different. Away from any grid interface $D^2\chi_j$ will approach χ'' with second order accuracy, assuming D^2 is the standard second order accurate finite difference operator. But for points adjacent to the interface, and only those points, guard cell filling errors of order Δx^3 in χ will lead to first order errors in $D^2\chi_j$. Thus, we expect to find second order convergence for $D^2\psi^n_j$ at all points except those points adjacent to the interface. Points adjacent to the interface should be first order convergent.

Fig. 3 shows the results of our convergence test for ψ''. The spikes at the interface ($x = 0$) appear because the two grid points adjacent to the interface are only first order convergent. Elsewhere, the plot shows second order convergence.[4]

The behavior demonstrated in Fig. 3 also occurs in the BSSN system when we examine the convergence plot for the Hamiltonian constraint, Fig. 5. In graphing the Hamiltonian constraint HC, we are comparing a combination of grid functions that includes second derivatives of the BSSN variables to the exact analytical solution of HC, namely, zero. We therefore expect spikes to appear in the convergence plot for the Hamiltonian constraint, and indeed they do.

We wish to emphasize that the lack of second order convergence for second spatial derivatives at grid points adjacent to the interfaces is not due to any error in our code, or shortcoming of the numerical algorithm. Since the undifferentiated variables are second order convergent everywhere, we can always assure second order con-

[4] Although our primary concern here is in ψ and its derivatives, it is interesting to note that π and π' are second order convergent everywhere. The second derivative π'' contains "packets" of high frequency, first order error that propagate away from the interface.

vergence of their derivatives by using suitable finite difference stencils. For example, in computing $D^2\psi^n_j$ from ψ^n_j we can use a second order accurate one-sided operator D^2 that avoids using guard cell values altogether. With such a choice the spikes in Fig. 3 disappear, and $D^2\psi^n_j$ is everywhere second order convergent. In our BSSN code, it is most convenient to compute the Hamiltonian constraint using the same centered difference operator D^2 that we use for the evolution equations. As a consequence, spikes appear at the grid interfaces in the convergence plot (Fig. 5).

To summarize, we use a third order accurate algorithm to fill guard cells for the BSSN variables (1). As a consequence, second spatial derivatives of these variables aquire first order errors at grid points adjacent to mesh refinement boundaries. These first order errors show up as spikes in a convergence plot for quantities that depend on second spatial derivatives, such as the Hamiltonian constraint. The key result or our analysis is a demonstration that the first order errors in second derivatives do not spoil the overall second order convergence of the evolved variables (1), in spite of the fact that second spatial derivatives appear on the right-hand sides of the evolution equations (2).

IV. BLACK HOLE EVOLUTIONS

Black hole spacetimes are a particularly challenging subject for numerical study. Practical applications will require that our FMR implementation perform robustly under the adverse conditions which arise in black hole simulations. In this section we will test that our techniques perform convergently and accurately in the presence of strong field dynamics and singular "punctures" associated with black hole evolutions and that these methods can be stable on the timescales required for interesting simulations.

The puncture approach to black hole spacetimes generalizes the Brill-Lindquist [?] prescription of initial data for resting black holes. In this approach, the spacetime is sliced in such a way as to avoid intersecting the black hole singularity, and the spatial slices are topologically isomorphic to $I\!R^3$ minus one point, a puncture, for each hole. The punctures represent an inner asymptotic region of the slice which can be conformally transformed to data which are regular on $I\!R^3$. In this way a resting black hole of mass M located at $r = 0$ is expressed in isotropic spatial coordinates by $\gamma_{ij} = \Psi^4_{BL}\delta_{ij}$, with factor $\Psi_{BL} = 1 + M/2r$ and $K_{ij} = 0$. A direct generalization of this expression for the conformal factor can be used to represent multiple black hole punctures, and data for spinning and moving black holes can be constructed according to the Bowen-York [?] prescription. A key characteristic which makes this representation appropriate for spacetime simulations is that with suitable conditions on the regularity of the lapse and shift, the evolution equations imply that time derivatives of the data at the puncture are regular everywhere despite the blow up in Ψ_{BL} at the puncture. Numerically, we treat the punctures by the prescription similar to that given in [?]. In the BSSN formulation, this amounts to a splitting of the conformal factor $\exp(4\phi)$ into a regular part, $\exp(4\phi_r)$, and non-evolving singular part, $\exp(4\phi_s)$, given by Ψ_{BL}. The numerical grid is staggered to make sure the puncture does not fall directly on a grid point, and to avoid the large finite differencing error, the derivatives of ϕ_s are specified analytically.

We study two test problems, each representing a Schwarzschild black hole in a different coordinate system. The first case, described in Sec. IV A, is a black hole in geodesic coordinates, in which the data evolve as the slice quickly advances into the singularity. In this case we have an exact solution for the development of the spacetime with which we can compare for a direct test of the simulation. Our next test, in Sec. IV B is intended to study the performance of our FMR interfaces under the condition that strong-field spacetime features pass through the interfaces. In this test we use a variant of the "1+log" slicing condition to define the lapse, α, with vanishing shift, $\beta^i = 0$. This gauge choice allows the slice to avoid running into the singularity, but causes the black hole to appear to grow in coordinate space so that the horizon region passes though our FMR interface.

A. Geodesic Slicing Results

1. Analytic Solution

Geodesic coordinates are obtained by setting unit lapse, $\alpha = 1$, and vanishing shift, $\beta^i = 0$. This implies that the gridpoints will follow geodesic trajectories through the physical spacetime. We obtain the analytic solution in terms of these geodesics.

The Schwarzschild geometry in standard coordinates is given by

$$ds^2 = -\alpha^2 dt^2 + \alpha^{-2} dR^2 + R^2 d\Omega^2, \qquad (16)$$

where $\alpha^2 = (1 - 2M/R)$.

To express this metric in geodesic coordinates, consider a spatial Cauchy surface Σ_0 in a 4-manifold $\mathcal{M}$ and a congruence of radial geodesics crossing Σ_0. Every point (R_0, Ω_0) in Σ_0 can be considered the "initial" point of a geodesic. Let the affine parameter τ for each geodesic be zero at Σ_0. Considering subsequent slices of constant proper time, we can set a global time τ over $\mathcal{M}$ and foliate it with the slices Σ_τ.

Since each (R_0, Ω_0) labels a unique geodesic, which in turn is parameterized by τ, R_0 can be considered a new radial coordinate that naturally pairs with the time coordinate τ. To emphasize R_0's role as a free coordinate, we rename it ρ.

Now we will derive the metric components in this coordinate system. The normalization of the tangent vector,

$n^a = (\partial/\partial\tau)^a$ implies that $g_{\tau\tau} = -1$. The $g_{\tau\rho}$ metric component is the ρ component of the shift vector. By means of the geodesic equation $n^a\nabla_a n^b = 0$ we can show that $g_{\tau\rho,\tau} = 0$. As initially $\beta^i = 0$, we conclude that $g_{\tau\rho} = 0$.

Together with Eq. 16 and the above considerations we have that

$$g_{\rho\rho} = \frac{(\partial R/\partial\rho)^2\, g_{RR}}{(\partial t/\partial\tau)^2\, g_{tt}}. \tag{17}$$

The final ingredient is the conserved energy associated to each geodesic. As the Schwartzchild geometry has a timelike Killing field $\xi^{(t)} = \partial_t$, we have that the energy $E = -\langle\partial_\tau, \xi^{(t)}\rangle = -(\partial t/\partial\tau)\, g_{tt}$ is conserved along the trajectory. Initially we have $E = -\sqrt{g^0_{tt}}$, where $g^0_{ab} = g_{ab}|_{t=0}$. Therefore:

$$g_{\rho\rho} = g^0_{RR}\left(\frac{\partial R}{\partial\rho}\right)^2 = \left(1-\frac{2M}{\rho}\right)^{-1}\left(\frac{\partial R}{\partial\rho}\right)^2. \tag{18}$$

The trajectories $R(\tau,\rho)$ can be explicilty computed from the $n^2 = -1$ and the enrgy conservation for the above geodesics, and we obtain [? ?]:

$$\tau - \frac{\rho^{3/2}}{(2M)^{1/2}}\left[\arccos\sqrt{\frac{R}{\rho}} + \sqrt{\frac{R}{\rho}\left(1-\frac{R}{\rho}\right)}\right] = 0. \tag{19}$$

This expression provides an implicit definition for $R = R(\rho,\tau)$, which is easily inverted numerically to high precision.

To perform numerical evolutions the geodesic coordinates have a drawback: the physical singularity is already present on the initial slice ($\tau = 0$) at $\rho = 0$. We can avoid this problem by going to isotropic coordinates by means of the transformation $\rho = r\,(1 + M/2r)^2$. We see that both $r = 0$ and $r = \infty$ map to $\rho = \infty$, and for real r the minimum value of ρ is $\rho = 2M$ (the horizon) at $r = M/2$, now the surface closest to the physical singularity on the initial slice. Substituting into Eq. (19) we see that geodesics originating on this surface reach the physical singularity, $R = 0$, at time $\tau = \pi M$, defining the maximum temporal extent of our coordinate system.

Returning to the metric, the transformation to isotropic coordinates gives us

$$g_{\rho\rho} = \left(\frac{\partial R}{\partial\rho}\right)^2\left(1+\frac{M}{2r}\right)^4. \tag{20}$$

From Eq. (20) we can express the final metric as:

$$ds^2 = -d\tau^2 + \left(\frac{\partial R}{\partial\rho}\right)^2\left(1+\frac{M}{2r}\right)^4 dr^2 + R^2 d\Omega^2. \tag{21}$$

Expressions for the extrinsic curvature, which have not previously appeared in the literature, can be derived in a similar manner. As we know from the ADM formalism [? ?], the extrinsic curvature can be viewed as the rate of change of the spatial metric

$$K_{ab} = -\frac{1}{2\alpha}\frac{d\gamma_{ab}}{d\tau}. \tag{22}$$

Using the previous expression with unit lapse and zero shift we obtain:

$$K_{\rho\rho} = -\frac{1}{2}\left(1+\frac{M}{2r}\right)^4\frac{\partial}{\partial\tau}\left(\frac{\partial R}{\partial\rho}\right)^2. \tag{23}$$

To proceed, note that if we have a function $f = f(u,v,w) = 0$ defining u as an implicit function of v and w, we can use the chain rule and the implicit function theorem to show that

$$\frac{\partial u}{\partial v} = -\frac{\partial f/\partial v}{\partial f/\partial u} \tag{24}$$

$$\frac{\partial^2 u}{\partial w\partial v} = -\left(\frac{\partial f}{\partial u}\right)^{-1}\left[\frac{\partial^2 f}{\partial v\partial w} + \frac{\partial^2 f}{\partial u^2}\frac{\partial f}{\partial v}\frac{\partial f}{\partial w}\left(\frac{\partial f}{\partial u}\right)^{-2}\right] + \left(\frac{\partial f}{\partial u}\right)^{-2}\left(\frac{\partial^2 f}{\partial v\partial u}\frac{\partial f}{\partial w} + \frac{\partial^2 f}{\partial u\partial w}\frac{\partial f}{\partial v}\right). \tag{25}$$

Taking for f the left hand side of Eq. (19), and, noting that $\partial f/\partial\tau = 1$, we conclude that:

$$K_{\rho\rho} = \left(1+\frac{M}{2r}\right)^4\frac{\partial f}{\partial\rho}\left(\frac{\partial f}{\partial R}\right)^{-3} \times\left[\frac{\partial^2 f}{\partial\rho\partial R} - \frac{\partial^2 f}{\partial R^2}\frac{\partial f}{\partial\rho}\left(\frac{\partial f}{\partial R}\right)^{-1}\right] \tag{26}$$

and

$$K_{\theta\theta} = R\left(\frac{\partial f}{\partial R}\right)^{-1} \qquad K_{\phi\phi} = K_{\theta\theta}\sin^2(\theta). \tag{27}$$

There are no off-diagonal terms. Observe that we have only partial derivatives of f that can be obtained analytically from Eq. (19), and easily evaluated numerically.

2. *Numerical Results*

Now we turn to the numerical evolution of a single puncture black hole. As explained in the previous section, at $t = \pi M$ the slice Σ_π will reach the physical singularity and because we are not performing any excision, this sets the maximum duration of our evolution. Nevertheless, $t \approx 3M$ is enough to test the relevant features of FMR evolution and provide us with a simple analytical solution.

In these simulations we locate the puncture black hole at the origin. The spherical symmetry of the problem allows us to restrict the simulation to an octant domain with symmetry boundary conditions, thereby saving memory and speeding up the evolutions. The outer boundaries are planes of constant x, y or z at $128M$ each.

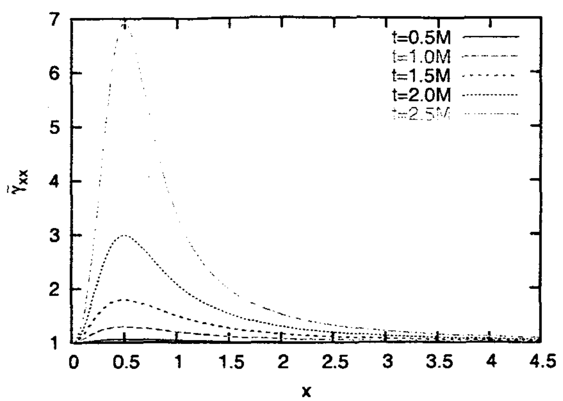

FIG. 4: Evolution of the conformal metric component $\tilde{\gamma}_{xx}$, for a geodesically sliced puncture, shown at $t = 0.5$, 1.0, 1.5, 2.0, and $2.5M$.

As noted before, one of the important features of FMR is to enable the outer boundary to be very far, of the order of $100M$. In this case we can apply our exact solution as the outer boundary condition, though this distant boundary is completely irrelevant for the most interesting strong-field region.

In this test we are mainly interested in how the FMR boundaries behave near the puncture and under strong gravitational fields. Even with the outer boundary far away we can, by applying multiple nested refinement regions, highly resolve the region near the puncture, as is required to demonstrate numerical convergence. To achieve the desired resolution near the puncture, we use 8 cubical refinement levels, locating the refinement boundaries at $64M$, $32M$, $16M$, $8M$, $4M$, $2M$ and $1M$. To test convergence we will examine the results of three runs with identical FMR grid structures, but different resolutions. The lowest resolution run has gridpoints $\Delta x_f = \Delta y_f = \Delta z_f = M/16$ apart in the finest refinement region near the puncture. The medium resolution run has double the resolution of the first run in each refinement region, and the the highest resolution run is doubled again for a maximum resolution of $\Delta x_f = M/64$. The memory demand and computational load per timestep are similar to unigrid runs of 32^3, 64^3 and 128^3 gridpoints.

In Fig. 4 we plot the conformal metric component $\tilde{\gamma}_{xx}$ along the x-axis at times $t = 0.5$, 1.0, 1.5, 2.0, and $2.5M$. The metric grows sharply near $0.5M$ blowing up completely as we approach $t = \pi M$ when, in these coordinates, the $r = .5M$ surface reaches the physical singularity. In the present context though, we are not so much interested in the field values of this well-studied spacetime, as in the difference between analytical and numerical results, to directly measure the error in our numerical simulation. At late times, the error is, not surprisingly, dominated by finite differencing error in the vicinity of the developing singularity.

The plots in Fig. 5 compare the errors at $t = 2.5M$ for the three varied resolution runs described above, demonstrating the convergence of $\tilde{\gamma}_{xx}$, $\tilde{A}_{xx}$, the Hamiltonian constraint, and the x-component of the momentum constraint. The error curve for the medium and low resolution runs have been divided by 4 and 16, respectively. That the curves shown lie nearly atop one another is an indication of second order convergence, *i.e.* that the lowest order error term depends quadratically on the gridspacing, Δx. That the remaining difference between the adjusted curves near $x = 0.5M$ seems also to decrease quadratically is an indication that the next significant error term is of order Δx^4. We achieve convergence to the analytic solution everywhere, from very near the puncture, through the peak region which is approaching the singularity, and in the weak field region.

Our particular interest is in the region near the refinement interfaces. The close-up shown in Fig. 7 demonstrates that $\tilde{\gamma}_{xx}$ converges quadratically to the exact solution. Our computation of the Hamiltonian constraint, however, requires second derivatives and, at the point nearest the refinement boundary, depends on the guardcells in such a way that the result is only 1st order convergent there. A close-up of the Hamiltonian constraint near a refinement boundary (Fig. ??) shows second order convergence at all non-adjacent points. This is as expected according to the discussion in Sec. III (*cf.* Fig. 3). We have verified for these simulations that the data for all BSSN variables do converge at second order at the refinement boundaries.

B. 1 + log slicing

Having rigorously tested the code against analytic solutions, we now use a different coordinate condition to show a longer-lived run with nontrivial, nonlinear dynamical behavior in the region of FMR interface boundaries. For this purpose, we again use zero shift but with a modified "1+log" slicing condition given by

$$\frac{\partial \alpha}{\partial t} = -2\alpha \Psi_{BL}^4 K \tag{28}$$

where insertion of the factor $\Psi_{BL} = 1 + M/2r$, originally recommended by [?], improves stability as well as convergence near the puncture. With this gauge condition, the code runs up to $t \sim 35M$. For $t > 35M$, the growth in the metric variables becomes too steep for the code to generate meaningful data.

The grid set-up is the same as in the geodesic case. Fig. ?? shows the second order convergence of the Hamiltonian constraint. The convergence of the Hamiltonian constraint is second order everywhere except at the refinement boundaries at $x \sim 1M$, $x \sim 2M$ where the convergence is only of the first order and at $x \sim 0.25M$ where the code does not converge at all. The code is second order convergent in other refinement boundaries, $x \sim 4M, 8M$, etc. At $x \sim 0.25M$, the metric component is second order convergent (Fig. ??), but the lapse

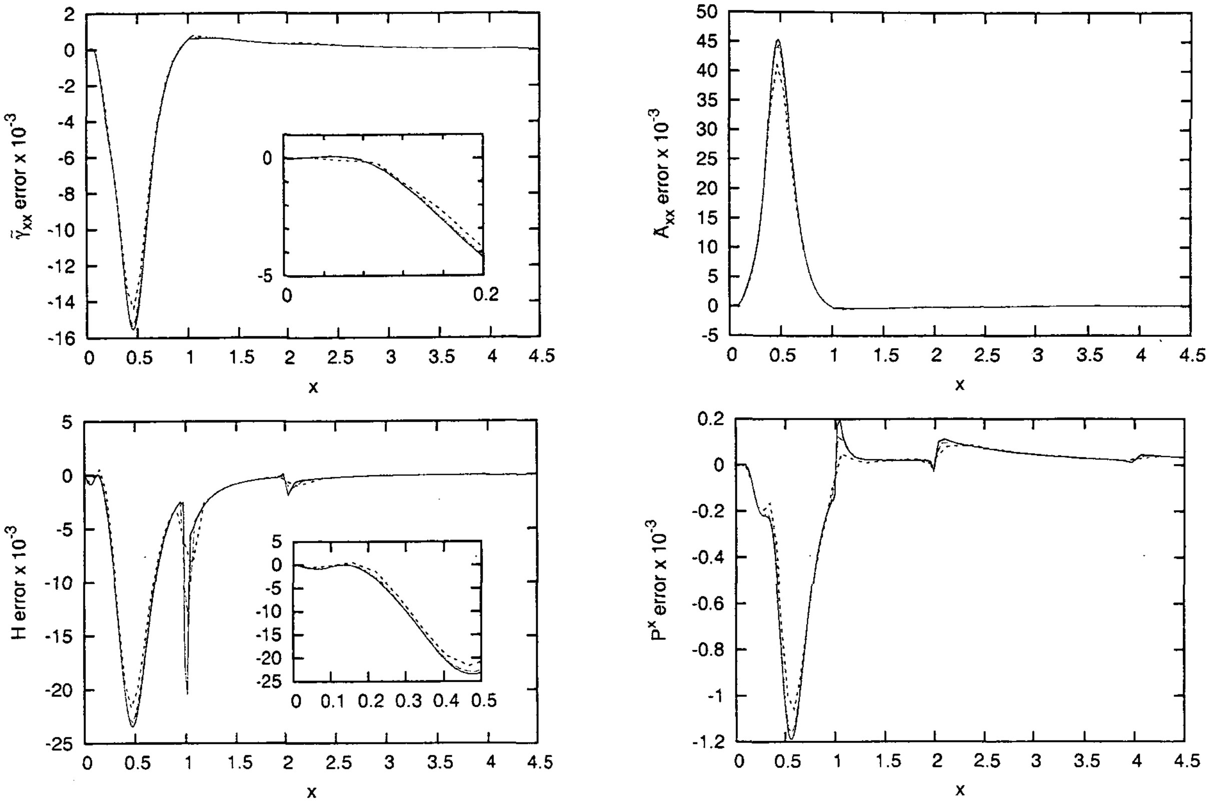

FIG. 5: Convergence of the errors (numerical values minus analytic values) in $\tilde{\gamma}_{xx}$, $\tilde{A}_{xx}$, the Hamiltonian constraint H, and the momentum constraint P^x, for a geodesically sliced puncture, all at the time $t = 2.5M$.

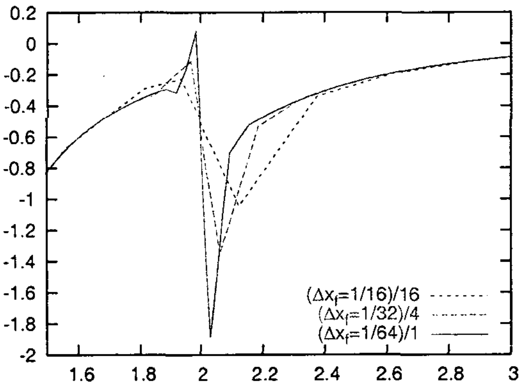

FIG. 6: Hamiltonian Constraint

(Fig. ??), exhibit high frequency noise.

This noise requires some explanation; however, it does not come from the refinement boundaries. Rather it is generated right on the spot at $x \sim 0.25M$. It becomes most evident at $t > 10M$, first appearing in the lapse and K, which are directly coupled with each other, and then eventually in all the extrinsic curvature variables. For the duration of the evolutions the noise does not propagate at all. The location of the noise is always within $x \sim 0.2M$ and $x \sim 0.4M$, independent of resolution and of the position of the FMR boundaries. The noise appears to represent a gauge shock associated with the collapse of the lapse, and is unique to 1+log slicing with zero shift. Outside this region from $0.2M$ to $0.4M$, all basic variables are second orderconvergent everywhere, including at FMR boundaries.

V. DISCUSSION AND CONCLUSIONS

This paper demonstrates that fixed mesh refinement boundaries can be located in the *strong field region* of a

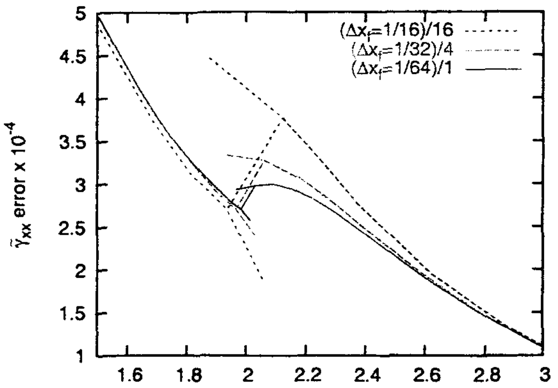

FIG. 7: Close-up of the convergence of the error (numerical value minus analytic value) in $\tilde{\gamma}_{xx}$, for a geodesically sliced puncture, at $t = 2.5M$, in the vicinity of the refinement boundary at $x = 1M$. Guardcell points are also shown.

black hole spacetime when the interface condition is handled properly, and that using mesh refinement technology, therefore, is a viable to way to use computational resources more efficiently. This result was verified through simulation of a Schwarzschild black hole in geodesic coordinates, for which we have an analytic solution for comparision, and through simulations of Schwarzschild in a variation of the 1+log (singularity avoiding) slicing. By nesting several levels of mesh refinement regions, we are able to convergently resolve the puncture while simultaneously pushing the boundary of our domain to $120M$ and keeping the computational problem size modest. Combined with our earlier results showing that gravitational waves pass through such FMR interfaces without significant reflections [?], we have now studied, in detail, the effects of FMR interfaces on the two primary features, waves and strong potentials, of astrophysically interesting spacetimes.

Our method for handing the interface condition, based in part on the Paramesh infrastructure, is detailed. We find that, in handling the interface condition between FMR levels, it is sufficient to interpolate with one order lower accuracy than would naively be expected without spoiling the convergence of our code. This leads to the somewhat counter-intuitive result that derivatives of our evolved fields appear non-convergent at grid points adjacent to the interface (*c.f.* Figs. 3, 5, 1+log), but that the evolved variables, for which these same derivatives are sources, remain convergent at all points in the interior of our domain (*c.f.* Figs 2, 5, and 1+log). Because it is the fields themselves, not their derivatives, that represent the physical solution, we find no problems with this understood behavior.

In the course of our investigation, we have also observed what we call the "point-two M problem" in 1+log runs. At intermediate times (**GIVE AN ESTIMATE**), high frequency noise appears at radial coordinate $r \approx 0.2M$ in the lapse α and the trace of the extrinsic curvature K. This noise then dissipates by later times so that, if the data is viewed outside of a certain window of time, it will not be observed. Higher resolution exacerbates this problem. We have verified that this phenomenon is present both with and without mesh refinement interfaces in the domain, and have further verified that it appears in an independently written, one-dimensional code. We therefore conclude that this is a real feature of the gauge choice used. Having chosen a generally accepted gauge, and having focused on effects of the mesh refinement interfaces in this work, we have not fully investigated the cause of nor possible remedies for this apparent pathology. We note it here, however, as an interesting topic for future investigation.

Acknowledgments

This work was performed while authors J.B. and J.V.M held National Research Council Associateship Awards at the Goddard Space Flight Center, and was funded by NASA Space Sciences grant ATP02-0043-0056. JDB also received funding from NSF grant PHY–0070892.

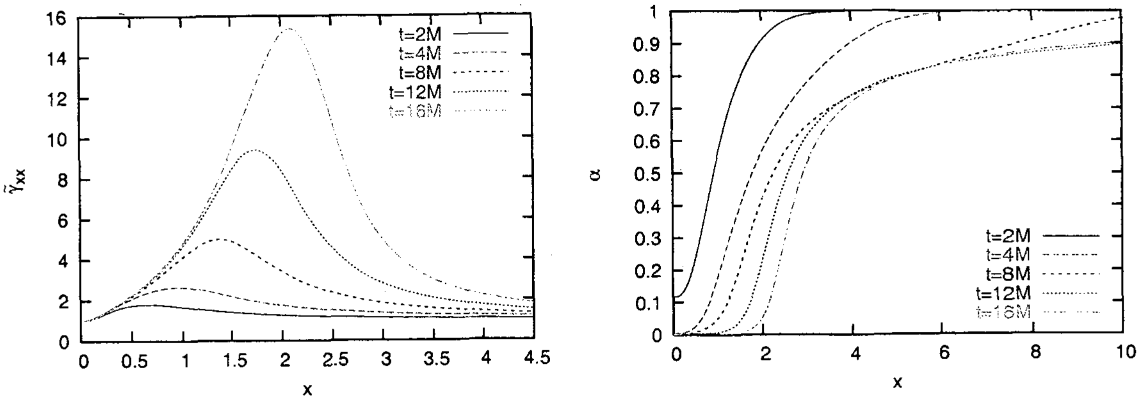

FIG. 8: γ_{xx} and α evolution.

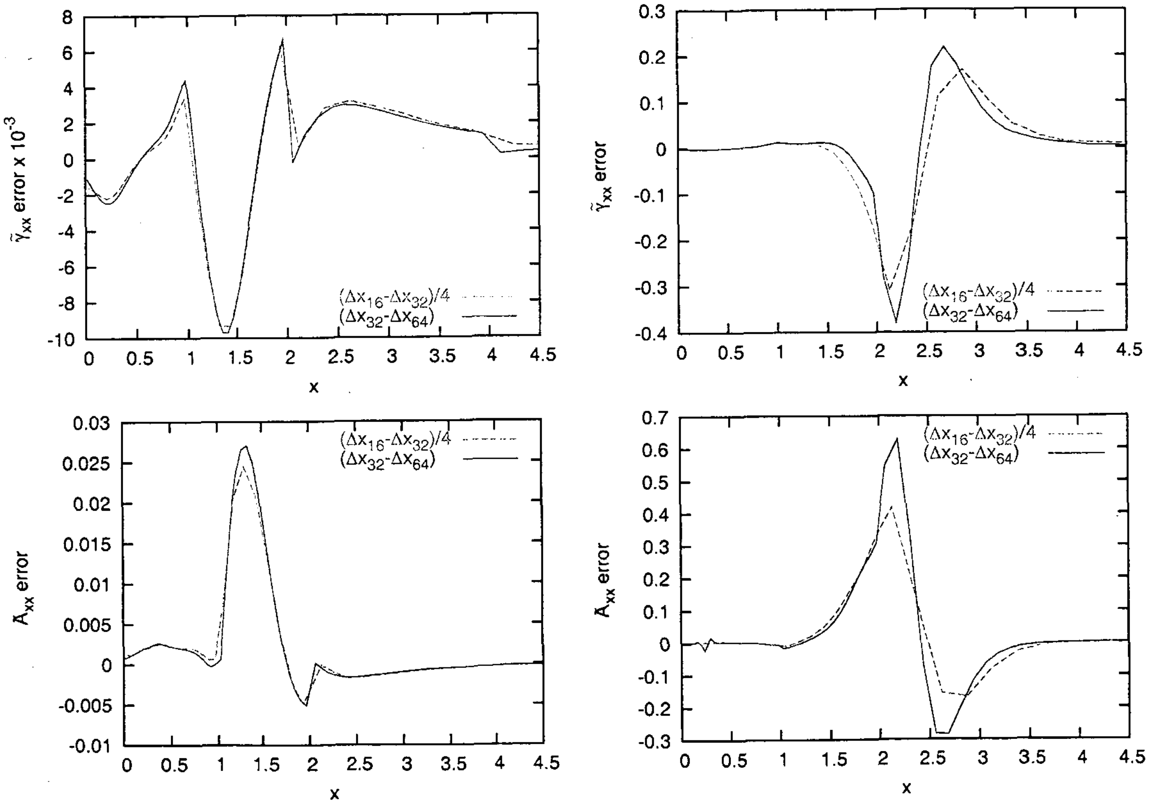

FIG. 9: $\tilde{\gamma}_{xx}$ and $\tilde{A}_{xx}$ convergence plot. T=8M and 16M

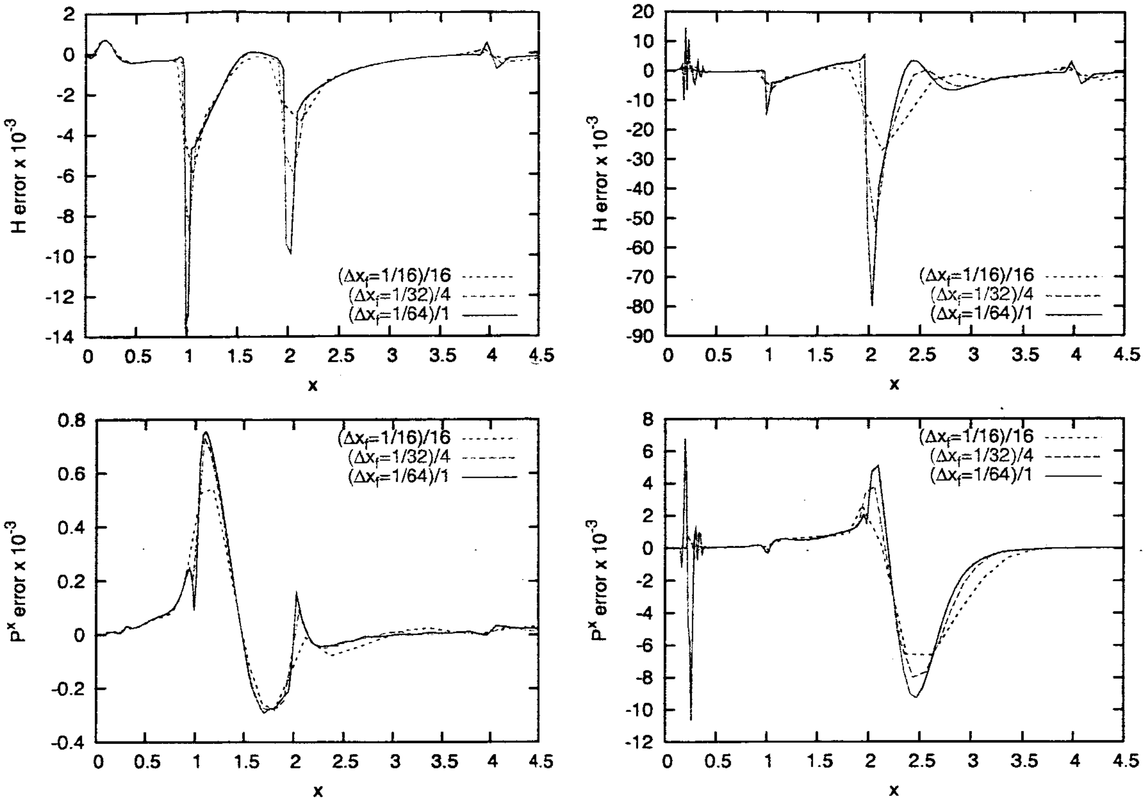

FIG. 10: Hamiltonian and Momentum constraint