



Interface conditions for wave propagation through mesh refinement boundaries

Dae-Il Choi^{a,b,*}, J. David Brown^{a,c,1}, Breno Imbiriba^{a,d},
Joan Centrella^a, Peter MacNeice^e

^a *Laboratory for High Energy Astrophysics, NASA/Goddard Space Flight Center, 8800 Greenbelt Rd., Greenbelt, MD 20771, USA*

^b *Universities Space Research Association, 7501 Forbes Boulevard, #206, Seabrook, MD 20706, USA*

^c *Department of Physics, North Carolina State University, Raleigh, NC 27695, USA*

^d *Department of Physics, University of Maryland, College Park, MD 20742, USA*

^e *Department of Physics, Drexel University, Philadelphia, PA 19104, USA*

Received 30 October 2002; received in revised form 19 June 2003; accepted 11 July 2003

Abstract

We study the propagation of waves across fixed mesh refinement boundaries in linear and nonlinear model equations in 1-D and 2-D, and in the 3-D Einstein equations of general relativity. We demonstrate that using linear interpolation to set the data in guard cells leads to the production of reflected waves at the refinement boundaries. Implementing quadratic interpolation to fill the guard cells suppresses these spurious signals.

© 2003 Elsevier B.V. All rights reserved.

Keywords: Partial differential equations; Computational techniques; Finite difference methods; Mesh generation and refinement; Numerical relativity; Gravitational waves

1. Introduction

Wave propagation is an important phenomenon throughout all areas of physics, with applications typically involving multiple spatial and temporal scales. In numerical modeling of such problems, one strategy for dealing with the disparity in spatial and temporal scales is the use of a nonuniform or adaptive computational mesh. In this case waves can cross mesh refinement boundaries as they propagate through the computational domain. This paper focuses on interface conditions that will allow waves to travel smoothly across fixed refinement boundaries, minimizing spurious reflections.

* Corresponding author. Tel.: +301-286-4066.

E-mail addresses: choi@milkyway.gsfc.nasa.gov (D.-I. Choi), david_brown@ncsu.edu (J. David Brown), imbiriba@milkyway.gsfc.nasa.gov (B. Imbiriba), jcentrel@milkyway.gsfc.nasa.gov (J. Centrella), macneice@alfven.gsfc.nasa.gov (P. MacNeice).

¹ Senior NRC Associate.

The specific application that motivated this study is modeling the emission of gravitational waves from astrophysical sources such as binary black hole and neutron star coalescences. Such systems are among the most important sources for ground-based gravitational wave detectors such as LIGO and VIRGO [1,2], as well as the planned space-based LISA mission [3]. The gravitational waves produced typically have wavelengths ~ 10 – 100 times the scales of their sources. Numerical simulations of such systems must therefore allow the signals to propagate from finely resolved regions around the sources into more coarsely resolved regions in the wave zones. Since the waveforms must be computed at large distances from their sources (i.e., effectively at infinity) for comparison with observations from gravitational wave detectors, the simulation domains must be made as large as possible. This can be achieved by incorporating several levels of successively coarser grids.

The propagation of gravitational waves is governed by the Einstein equations, which are a coupled set of nonlinear partial differential equations [4]. These equations can be written in a variety of ways, but current practice in numerical relativity favors the use of the so-called BSSN formalism [5,6]. In this formalism, the Einstein equations are written as a system of quasi-linear equations with first-order time derivatives and second-order spatial derivatives. In this paper we restrict our analysis to the “iterated Crank–Nicholson” update scheme, which is a second order accurate, explicit finite difference method that is currently in widespread use in the relativity community. It should also be noted that we consider mesh refinement only in space, not in time. In particular, for our present analysis we use a common timestep across the entire computational domain.

Adaptive mesh refinement (AMR) was first applied in numerical relativity to study critical phenomena in the 1-D collapse of a scalar field to form a black hole [7]. An early 3-D application focussed on evolving a single black hole [8]; this was followed by the use of fixed mesh refinement (FMR) to evolve a short part of a binary black hole evolution [9]. AMR was also employed to follow the propagation of gravitational waves through spacetime, first using a single model equation that describes perturbations of a non-rotating black hole [10] and later in the 3-D Einstein equations [11], and to study inhomogeneous cosmological models [12]. In these AMR studies the refinement and derefinement conditions were generally tuned so that the gravitational waves remained within the finely resolved regions.

In this paper, we address the challenge of evolving gravitational wave signals *across* mesh refinement boundaries using FMR. Success in this endeavor is an essential component of gravitational wave source modeling, due to the large disparity in the scales of the sources and the waves. Our challenge amounts to choosing a prescription for coupling adjacent grid blocks when the blocks have different resolutions. Grid blocks are coupled through their guard cells, which must be filled using data from the blocks’ interior cells. In hydrodynamics codes it is common practice to use a linear interpolation scheme for guard cell filling, with a possible adjustment for flux conservation across the interface between blocks [13–15]. We have found that this prescription is not adequate for the BSSN formulation of the Einstein equations. In particular, linear guard cell filling leads to unacceptably large reflections and distortions of the gravitational waves as they propagate from fine grid blocks to coarse grid blocks. Our solution to this problem is to use a guard cell filling procedure with quadratic-order accuracy orthogonal to the coarse–fine grid interface. The need for quadratic order guard cell filling has been previously demonstrated for elliptic boundary value problems with second order derivatives in [16,17]. With this prescription spurious wave reflections and distortions are reduced dramatically.

Given the complexity of the full system of Einstein equations, we have chosen to analyze first a set of model wave equations in 1-D and 2-D that mimic some of the properties of the Einstein equations, as expressed in BSSN form. These simplified test beds have proved essential to understanding and correcting the problems that arise in the propagation of waves across mesh refinement boundaries. Since the solution we uncovered using these model equations has proved effective in curing the difficulties encountered in the Einstein equations, we expect this work to be useful across a broad range of related wave propagation problems.

2. Linear wave equation in 1-D: evolution on a uniform grid

The linear wave equation in 1-D is generally written in the form

$$\frac{\partial^2 \phi}{\partial t^2} = \frac{\partial^2 \phi}{\partial x^2}, \quad (1)$$

where $\phi = \phi(x, t)$. Introducing the auxiliary variable $\Pi(x, t)$, we can cast Eq. (1) in a form that uses only first time derivatives:

$$\frac{\partial \phi}{\partial t} = \Pi, \quad (2)$$

$$\frac{\partial \Pi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2}. \quad (3)$$

In this section, we examine the system of Eqs. (2) and (3) to understand the interface conditions needed for smooth propagation of waves across mesh refinement boundaries. In later sections, these conditions are applied to nonlinear and multidimensional wave equations.

2.1. Discretization

For the spatial discretization of Eqs. (2) and (3), 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 [18]. To advance this system of ordinary differential equations in time we use an $O(\Delta t)^2$ iterative method first suggested by Choptuik (see [19]). In the numerical relativity literature, this explicit update scheme is referred to as “iterated Crank–Nicholson”. Each iteration has the form

$$\phi_i^{n+1} = \phi_i^n + \Delta t \bar{\Pi}_i, \quad (4)$$

$$\Pi_i^{n+1} = \Pi_i^n + \frac{\Delta t}{(\Delta x)^2} (\bar{\phi}_{i+1} - 2\bar{\phi}_i + \bar{\phi}_{i-1}) = \Pi_i^n + \frac{\Delta t}{(\Delta x)^2} F(\bar{\phi}), \quad (5)$$

where we use i to label the spatial grid, n to label the time steps, and $\bar{\phi}_i, \bar{\Pi}_i$ to indicate intermediate values calculated during the iteration process. Note that the familiar Crank–Nicholson algorithm is obtained by setting $\bar{\phi}_i$ and $\bar{\Pi}_i$ equal to their time averages, $(\phi_i^{n+1} + \phi_i^n)/2$ and $(\Pi_i^{n+1} + \Pi_i^n)/2$, respectively.

For two iterations, the specific steps are as follows. Begin by applying the discretization (4) and (5) with $\bar{\phi} = \phi^n, \bar{\Pi} = \Pi^n$ to calculate a first approximation to ϕ^{n+1} and Π^{n+1} :

$${}^{(1)}\phi_i^{n+1} = \phi_i^n + \Delta t \Pi_i^n, \quad (6)$$

$${}^{(1)}\Pi_i^{n+1} = \Pi_i^n + \frac{\Delta t}{(\Delta x)^2} F(\phi^n). \quad (7)$$

Average these new values with those at the starting time level n to get new values for $\bar{\phi}$ and $\bar{\Pi}$:

$${}^{(1)}\bar{\phi}_i = \frac{1}{2} ({}^{(1)}\phi_i^{n+1} + \phi_i^n) \quad (8)$$

$${}^{(1)}\bar{\Pi}_i = \frac{1}{2}({}^{(1)}\Pi_i^{n+1} + \Pi_i^n). \quad (9)$$

Now perform a second iteration. Again applying (4) and (5) we find a second approximation to ϕ^{n+1} and Π^{n+1} :

$${}^{(2)}\phi_i^{n+1} = \phi_i^n + \Delta t {}^{(1)}\bar{\Pi}_i, \quad (10)$$

$${}^{(2)}\Pi_i^{n+1} = \Pi_i^n + \frac{\Delta t}{(\Delta x)^2} F({}^{(1)}\bar{\phi}). \quad (11)$$

Averaging again with the values at level n yields

$${}^{(2)}\bar{\phi}_i = \frac{1}{2}({}^{(2)}\phi_i^{n+1} + \phi_i^n), \quad (12)$$

$${}^{(2)}\bar{\Pi}_i = \frac{1}{2}({}^{(2)}\Pi_i^{n+1} + \Pi_i^n). \quad (13)$$

A final update is carried out using these twice-iterated values:

$$\phi_i^{n+1} = \phi_i^n + \Delta t {}^{(2)}\bar{\Pi}_i, \quad (14)$$

$$\Pi_i^{n+1} = \Pi_i^n + \frac{\Delta t}{\Delta x^2} F({}^{(2)}\bar{\phi}). \quad (15)$$

Clearly this algorithm can be carried out for any number of iterations. In the formal limit of an infinite number of iterations, it yields the usual Crank–Nicholson scheme. However, a von Neumann stability analysis shows that this iterative scheme is stable only when the number of iterations equals 2, 3, 6, 7, 10, 11, etc., and the Courant condition $\Delta t \leq \Delta x$ is satisfied. This was shown by Teukolsky [19] for the advection equation, but the conclusion holds as well for the wave equation in the form (2) and (3). Furthermore, the accuracy of the iterative scheme is second order for any number of iterations. We must carry out at least two iterations for stability, but continuing beyond two iterations does not reduce the truncation error. In this paper we follow the common current practice in numerical relativity and carry out precisely two iterations for our tests.

2.2. Evolutions on a uniform grid

We first carried out uniform grid, or unigrid, evolutions of the discretized wave equation to provide a basis for comparison with mesh refinement runs. The initial data for ϕ is taken to be a Gaussian wave-packet,

$$\phi(x, t = 0) = A e^{-x^2/\sigma^2}, \quad \Pi(x, t = 0) = 0, \quad (16)$$

with $A = 1$ and $\sigma = 0.25$. The spatial domain extends from $x = -4$ to $x = +4$. Time evolution of this data produces two packets traveling with velocity $v = \pm 1$, each having amplitude $A = 0.5$ and the same value of σ as the original packet. Here we will consider only the packet traveling to the right, in the region $0 \leq x \leq 4$.

Fig. 1 shows the evolution of this packet for two different resolutions. The coarser resolution is given by $H = \Delta x = 0.045$ (dotted line), which has ~ 10 zones across the width of the packet at half its maximum amplitude. The solid line shows resolution $h = H/2 = 0.0225$. The time step is chosen to be $\Delta t = \Delta x/4$ for a given spatial resolution, Δx . In the last few panels of Fig. 1 one can see a slight separation between the two curves. This is primarily due to numerical dispersion, which causes the phase velocities to deviate from

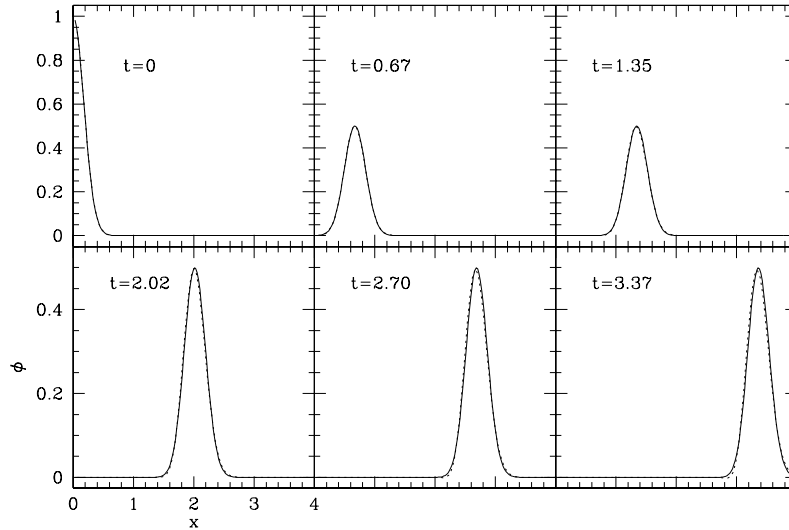


Fig. 1. The evolution of a Gaussian wavepacket on a uniform grid according to the linear 1-D wave equation is shown for two different resolutions: $H = \Delta x = 0.045$ (dotted line) and $h = H/2$ (solid line).

unity. The phase velocity for a monochromatic wave propagating on a discrete, uniform grid is calculated in Appendix A, with the result displayed in Eq. (A.11). According to this formula we expect the pulse (which has wavelength ~ 1) to propagate with speed ~ 0.999 on the fine grid and speed ~ 0.996 on the coarse grid. This translates into a separation between the two pulses of about 0.01 at time $t = 3.37$, which is the approximate separation seen in the last panel of Fig. 1.

The time evolution of the absolute errors $\epsilon \equiv |\phi_{\text{analytic}} - \phi_{\text{numerical}}|$ is shown in Fig. 2. The dotted line shows the errors ϵ_H for the coarse resolution H , and the solid line is $4 \times \epsilon_h$. Inspection of Fig. 2 shows that the two curves are nearly identical, demonstrating the second-order convergence of these runs. Note that

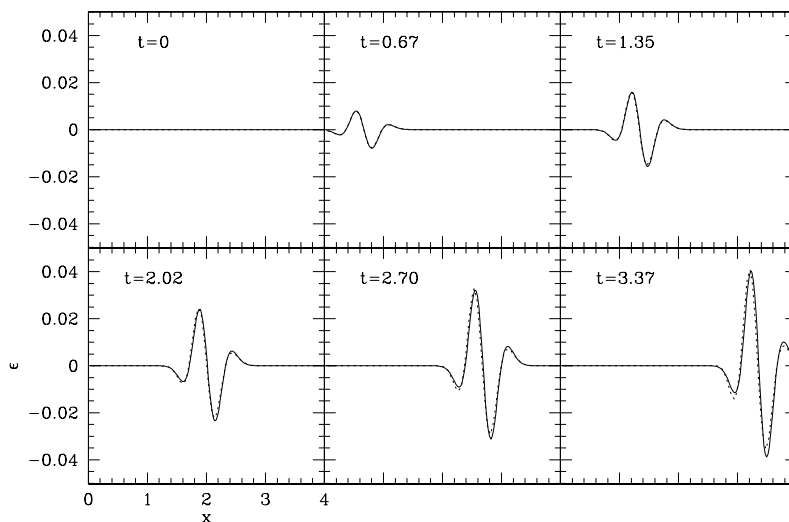


Fig. 2. The evolution of the absolute errors ϵ for the two runs in Fig. 1 is shown. The dotted line shows ϵ_H and the solid line $4 \times \epsilon_h$.

the errors are approximately antisymmetric about the location of the pulse center. This is because the dominant source of numerical error is dispersion, which has the principle effect of shifting each wave pulse relative to the exact solution.

3. Implementation of mesh refinement

We use the Paramesh package [20] to implement the mesh refinement and parallelization in our codes. All of our codes use cell-centered data. Paramesh works on logically Cartesian, or structured, grids and carries out mesh refinement on grid blocks. The underlying mesh refinement technique is similar to that of [21], in which grid blocks are bisected in each coordinate direction when refinement is needed. The grid blocks all have the same logical structure, with nxb zones in the x -direction, and similarly for nyb and nzb . Thus, refinement of a block in 1-D yields two child blocks, each having nxb zones but with zone sizes a factor of two smaller than in the parent block. When needed, refinement can continue 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 in computing finite difference spatial derivatives near the block's boundary. These guard cells must be filled using data from the interior cells of the given block and the adjacent block.

Fig. 3 shows a section of a 1-D grid in the vicinity of an interlevel boundary between two neighboring grid blocks. The fine grid covers the left half of the 1-D space, with cell-centered grid points labeled $-1/2$, $-3/2$, etc. The coarse grid covers the right half with cell-centered grid points labeled $1/2$, $3/2$, etc. The fine and coarse blocks are offset from one another for clarity of presentation. One layer of guard cells is shown, with “G” marking the coarse grid guard cell and “g” the fine grid guard cell. These guard cells are filled with data from neighboring blocks or, if the block forms part of the edge of the computational domain, from appropriate outer boundary conditions.

Paramesh can be used in applications requiring AMR, FMR, or a combination of these. It handles the creation of grid blocks, and builds and maintains the data structures needed to track the spatial relationships between blocks. It takes care of 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, maximize block locality, and minimize inter-processor communications.

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. At the mesh refinement boundaries, we use a single layer of guard cells as shown in Fig. 3; 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 subsection.

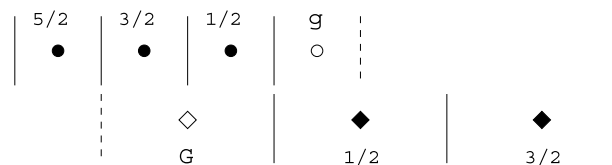


Fig. 3. An interlevel boundary between a coarse and fine grid in 1-D is shown. The coarse grid data points are marked with filled diamonds and positive half integers, the fine grid data points are marked with filled circles and negative half integers. The coarse and fine guard cells are marked with the corresponding open symbols and are denoted by “G” and “g”, respectively.

4. Linear wave equation in 1-D: evolutions with fixed mesh refinement

We now carry out evolutions of 1-D linear waves that encounter a change in the grid resolution at a fixed location. For the gravitational wave applications in which we are interested, waves will be generated in a finely resolved region and then travel out into more coarsely resolved regions. We thus start our initial wave packet, given by Eq. (16), in a region of fine resolution $h = 0.0225$ around the origin. The spatial domain is again $-4 \leq x \leq 4$. As before, the initial wave packet splits into two identical packets traveling in opposite directions. Each of these packets then encounters a fixed refinement boundary, located at $x = \pm 2.1$, and crosses into a region of coarser resolution $H = 2h$. In the following discussions, we focus only on the region $x \geq 0$.

We first use the default Paramesh linear interpolation to set the value of the data in the guard cells on both the coarse and fine grids. With this prescription for guard cell filling, the coarse grid guard cell value of any function f is given by linear interpolation,

$$f_G = \frac{1}{2}(f_{-3/2} + f_{-1/2}). \quad (17)$$

The value of f in the fine grid guard cell “g” is then given by a linear interpolation using coarse grid values, $f_g = (f_G + 3f_{1/2})/4$. Combined with Eq. (17), this gives

$$f_g = \frac{1}{8}(f_{-3/2} + f_{-1/2} + 6f_{1/2}). \quad (18)$$

Note that this guard cell filling (GCF) procedure uses the points f_G and $f_{1/2}$ on the coarse grid to obtain f_g ; this is in contrast to the direct approach, which uses the nearest points $f_{-1/2}$ and $f_{1/2}$ (cf. Eq. (23)). The prescription (17) and (18) for GCF has errors of order h^2 and is the default linear GCF method in Paramesh. We will refer to this procedure as *linear GCF* in this paper. The results of using linear GCF are displayed in Fig. 4, which shows the time evolution of the absolute errors ϵ . The dotted line shows the run with linear interpolation at the interface boundary, and the solid line the results of a unigrid run at the fine

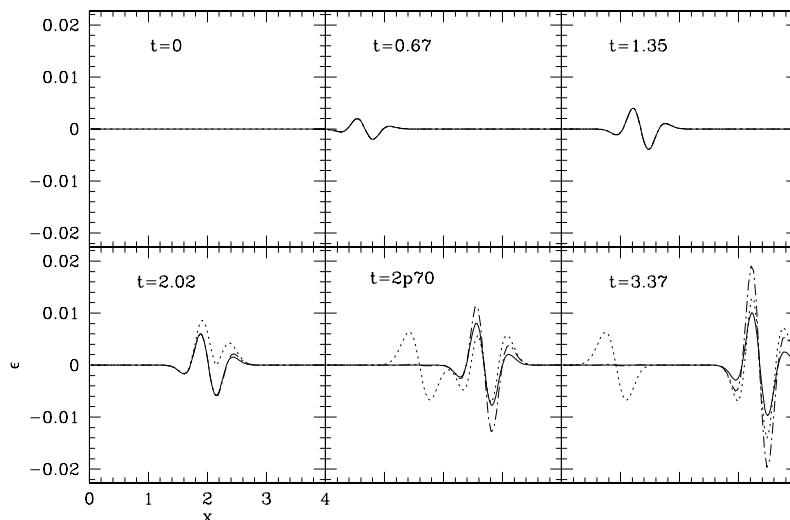


Fig. 4. The evolution of the absolute errors ϵ is shown for a Gaussian packet crossing a fixed refinement boundary at $x = 2.1$ with linear (dotted line) and quadratic (dashed line) GCF. The solid line shows ϵ for a unigrid run at the fine grid resolution.

grid resolution. As the packet passes through this boundary, a reflected wave is generated propagating to the left. The transmitted wave continues traveling to the right into the coarse grid region.

In large scale simulations of the Einstein equations with several levels of refinement, such spurious reflected waves can seriously degrade the quality of the results. Globally increasing the resolution until the reflected waves reach acceptably small amplitudes is generally not possible in 3-D. We thus need a better way to control the behavior of the signals crossing the interfaces.

To this end, we implemented direct quadratic interpolation (i.e., using the nearest 3 data points) to set the data in the coarse and fine grid guard cells. Refer again to Fig. 3. For the fine grid guardcell “g”, quadratic interpolation yields [18]

$$f_g = \frac{1}{15}(-3f_{-3/2} + 10f_{-1/2} + 8f_{1/2}). \quad (19)$$

The coarse grid guard cell “G” is filled by matching first derivatives across the interface,

$$\frac{f_{1/2} - f_G}{H} = \frac{f_g - f_{-1/2}}{h}, \quad (20)$$

where $H = 2h$. This step, which ensures that the solution is smooth across the interface, can be viewed as “flux matching” where the gradient of f plays the role of the flux. By combining the derivative matching condition with the formula for f_g we find

$$f_G = \frac{1}{15}(6f_{-3/2} + 10f_{-1/2} - f_{1/2}). \quad (21)$$

This same result for f_G can be obtained by direct quadratic interpolation. These formulae for GCF have errors of order h^3 .

The absolute errors obtained when using quadratic interpolation are shown as the dashed line in Fig. 4. Note that the reflected wave has been greatly reduced. Additional simulations, in which the size of the zones is everywhere decreased by successive factors of two, show that with quadratic GCF the code is second-order convergent. On the other hand, with linear GCF, the reflected pulse is first-order convergent. The transmitted pulse also acquires first-order errors at the interface with linear GCF. As the transmitted wave propagates through the coarse grid region, second-order errors due to dispersion and dissipation eventually dominate over the first-order errors introduced at the interface. At that point, the transmitted pulse can appear second-order convergent.

We also conducted tests using a one-dimensional periodic domain consisting of 20% fine grid and 80% coarse grid. A wave pulse was allowed to cycle through the domain multiple times. These tests clearly show that with quadratic guard cell filling, but not with linear guard cell filling, the code is second-order convergent. We also used this test code to check the stability of the interface conditions. After thousands of cycles of the wave pulse through the refined region, there were no signs of instability with either linear or quadratic guard cell filling.

In Appendix A, we present a detailed analytic treatment of wave propagation across mesh refinement boundaries that complements our numerical experiments. There we compute the reflection coefficient \mathbf{R} and transmission coefficient \mathbf{T} for a monochromatic (single frequency) wave traveling on a grid with fixed mesh refinement, for various methods of GCF. The wave travels from a fine grid region with resolution h into a coarse grid region with resolution $2h$. Fig. 5 shows the absolute value of the reflection coefficient $|\mathbf{R}|$ for linear GCF (17) and (18) (dashed curve) and quadratic GCF (19)–(21) (solid curve). The dotted curve shows the results for direct linear interpolation, defined by

$$f_G = \frac{1}{2}(f_{-3/2} + f_{-1/2}) \quad (22)$$

for the coarse grid guard cell and

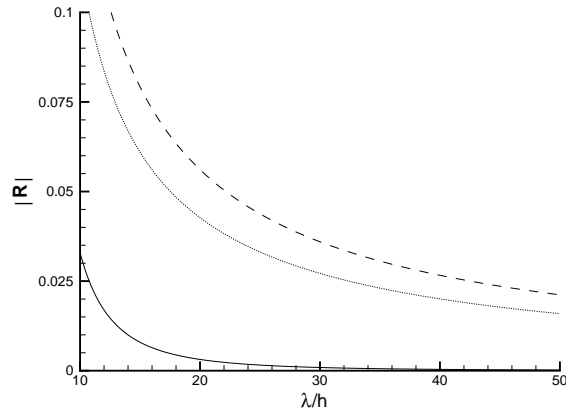


Fig. 5. Absolute values of the reflection coefficient $|\mathbf{R}|$ for linear (dashed curve), quadratic (solid curve), and direct linear (dotted curve) GCF for a wave with wavelength λ crossing a single fixed mesh boundary. The resolution of the fine grid is h .

$$f_g = \frac{1}{3}(f_{-1/2} + 2f_{1/2}) \quad (23)$$

for the fine grid guard cell. Direct linear interpolation, like the default linear GCF in Paramesh, has errors of order h^2 . The curves of Fig. 5 are plotted as functions of the wavelength in the fine grid region divided by the fine grid cell size h . Equivalently, we can interpret the horizontal-axis values as the number of fine grid points per wavelength.

For our 1-D wave equation tests, the Gaussian packet behaves roughly like a wave of wavelength $\lambda \sim 1$. With $h = 0.0225$, this corresponds to about $\lambda/h = 44$ fine grid points per wavelength. From Fig. 5 we see that the reflection coefficient for linear interpolation is about $|\mathbf{R}| = 0.02$ while that for quadratic GCF is $|\mathbf{R}| = 0.0003$. With an incident pulse amplitude of 0.5, we expect a reflected wave amplitude of about 0.01 for linear GCF and less than 0.0002 for quadratic GCF. This reflected pulse for the linear case is clearly seen in Fig. 4.

The importance of minimizing spurious reflections from grid interfaces has been emphasized above. It is equally important to minimize the distortion of waves that pass through a grid interface. The errors in the transmitted wave pulse for linear and quadratic GCF are shown in the region $x > 2.1$ of the last few panels of Fig. 4. Note that the errors for quadratic GCF are actually larger than the errors for linear GCF. This surprising result is explained as follows. Observe that the errors for the two fixed mesh refinement simulations, as well as for the unigrid run (solid curve), are approximately antisymmetric about the pulse center. The errors in each case, as in the unigrid tests discussed in Section 2, are primarily due to dispersion. Dispersion causes the wave pulses to fall behind the exact solution during propagation, giving rise to the errors shown in Fig. 4. This effect is greater for the two runs with fixed mesh refinement because, beyond $x = 2.1$, the grid resolution is lower than for the unigrid run. However, with mesh refinement, the transmitted pulse will also suffer a phase error which has the effect of artificially shifting the pulse along the x -axis. In the case of linear GCF, there is a relatively large positive phase error in the transmitted wave. This phase shift partially compensates for the negative shift caused by dispersion. As a result the size of the largest peaks in the error for the transmitted wave, for the particular test shown in Fig. 4, is smaller with linear GCF than with quadratic GCF.

Figs. 6 and 7 show the absolute value of the transmission coefficient $|\mathbf{T}|$ and the phase of the transmission coefficient $\varphi = \arctan(\Im(\mathbf{T})/\Re(\mathbf{T}))$ for a monochromatic wave, for linear, quadratic, and direct linear interpolation. These graphs are obtained from the analysis in Appendix A. From Fig. 6 it is clear that

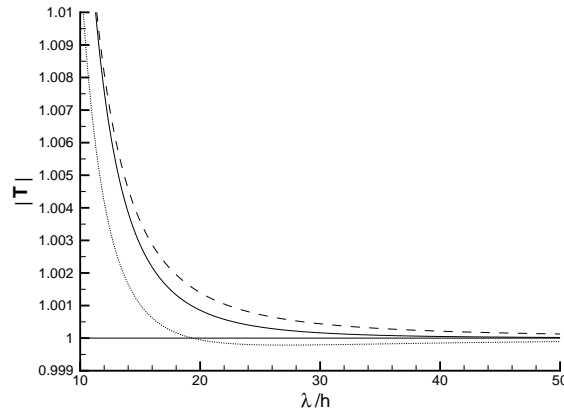


Fig. 6. Absolute values of the transmission coefficients for linear (dashed line), quadratic (solid line), and direct linear (dotted) GCF.

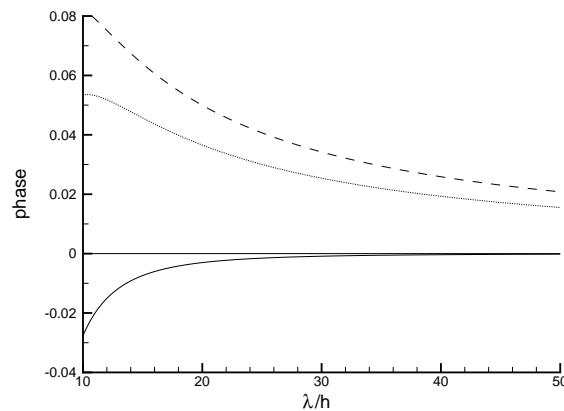


Fig. 7. The phase of the transmission coefficients for linear (dashed line), quadratic (solid line), and direct linear (dotted) GCF.

at any wavelength (any resolution) the error in amplitude for the transmitted wave is smaller for quadratic GCF than for linear GCF.² The dominant source of error for the transmitted wave is actually phase error, shown in Fig. 7. The magnitude of this error for quadratic GCF is much smaller than that for linear GCF. For a wavelength of $\lambda \sim 1$, the linear guard cell filling produces a phase shift of about $\varphi = 0.024$, while quadratic GCF gives a phase shift of about $\varphi = -0.00028$. For the tests shown in Fig. 4, the positive phase for linear GCF translates into a shift along the positive x -axis of about $\delta x = \lambda\varphi/(2\pi) \approx 0.004$. With quadratic GCF, the pulse is shifted in the negative direction, but by a much smaller amount $\delta x \approx -0.00004$. Close inspection of the data for the two transmitted pulses shows that they indeed have a separation of $\delta x \approx 0.004$. For linear GCF, this phase shift pushes the wave pulse forward and artificially compensates for

² At low resolution, that is, for wavelengths less than about $28h$, direct linear GCF has the smallest error for the transmitted wave amplitude. However, as discussed in Appendix A, as the resolution is increased $|T|$ is much closer to 1 for quadratic GCF. Also note from Fig. 7 that direct linear GCF has large phase errors for the transmitted wave.

the phase lag caused by dispersion. In general, there is no reason to expect the cumulative phase lag due to dispersion to be close in magnitude (but opposite in sign) to the phase advance caused by transmission through various grid interfaces. Thus, the relatively small transmission error seen in Fig. 4 for linear GCF should be viewed as an accident of the particular example, not a generic result.

5. Nonlinear wave equation in 1-D

The next step in developing model equations to test these interface conditions is to add nonlinear terms similar to those found in the Einstein equations. This produces the following nonlinear wave equation

$$\frac{\partial^2 \phi}{\partial t^2} = \frac{\partial^2 \phi}{\partial x^2} + d \left(\frac{\partial \phi}{\partial t} \right)^2 + e \left(\frac{\partial \phi}{\partial x} \right)^2, \quad (24)$$

where d and e are arbitrary constants. Again introducing the auxiliary variable $\Pi(x, t)$, we get the first order system

$$\frac{\partial \phi}{\partial t} = \Pi, \quad (25)$$

$$\frac{\partial \Pi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} + d \Pi^2 + e \left(\frac{\partial \phi}{\partial x} \right)^2. \quad (26)$$

Using the discretization introduced in Section 2.1, we have

$$\phi_i^{n+1} = \phi_i^n + (\Delta t) \bar{\Pi}_i \quad (27)$$

$$\Pi_i^{n+1} = \Pi_i^n + \frac{\Delta t}{(\Delta x)^2} (\bar{\phi}_{i+1} - 2\bar{\phi}_i + \bar{\phi}_{i-1}) + d(\Delta t)(\bar{\Pi}_i)^2 + e(\Delta t) \left(\frac{\bar{\phi}_{i+1} - \bar{\phi}_{i-1}}{2\Delta x} \right)^2. \quad (28)$$

Eqs. (27) and (28) are updated following the steps given in (6)–(15).

We consider the case $d = -e = 1$ and set up an initial Gaussian wave packet centered on the origin using the prescription given by Eq. (16), with $\Pi(x, t = 0) = 0$. This splits into two identical packets traveling in opposite directions, each having amplitude $A = 0.38$ and width $\sigma = 0.25$. We use the spatial domain $-4 \leq x \leq 4$ and set fixed refinement boundaries at $x = \pm 2.1$. The fine grid around the origin has resolution $h = 0.0225$ and the coarse grid regions have resolution $H = 2h$. We focus on the region $x \geq 0$.

The results are shown in Fig. 8. Since we do not have an analytic solution for Eq. (24), we display the actual solution and use unigrid runs for comparison. In addition, the vertical scale is chosen to zoom in on the region around the base of the packet (i.e., near $\phi = 0$), where the differences between the runs are the most apparent. The thin solid line shows the solution for a unigrid run at the coarse resolution H , and the thick solid line shows a unigrid run at the fine resolution h . Runs in which the packet encounters a refinement boundary are shown using a dotted line (linear GCF) and a dashed line (quadratic GCF). As we saw before, a reflected wave is generated when the packet crosses the refinement boundary using linear GCF; these effects are much less noticeable when using quadratic GCF. As in the case of the linear wave equation, the code is second-order convergent when using quadratic GCF.

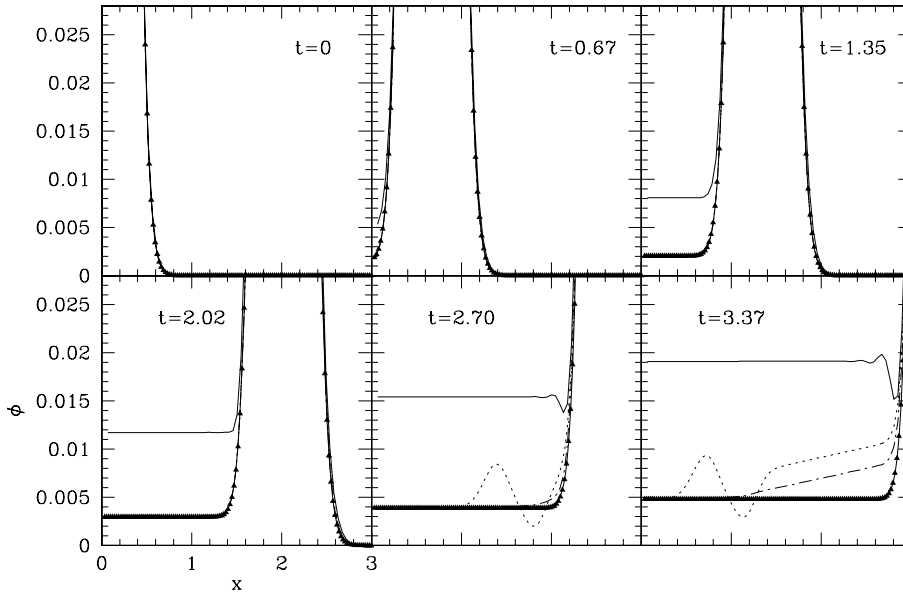


Fig. 8. The time evolution of the solution ϕ to the nonlinear 1-D wave equation is shown. Unigrid runs are shown at the coarse resolution H (thin solid line) and the fine resolution $h = H/2$ (thick solid line). Runs in which the packet encounters a mesh refinement boundary at $x = 2.1$ are shown for linear interpolation (dotted line) and quadratic (dashed line) GCF.

6. Wave equation in 2-D

As a next step, we consider the wave equation in 2-D. The evolution of cylindrically symmetric waves on a 2-D Cartesian mesh provides an ideal test problem in which the signals cross mesh refinement boundaries that are, in general, not perpendicular to their directions of propagation.

The 2-D model wave equation takes the form

$$\frac{\partial^2 \phi}{\partial t^2} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + d \left(\frac{\partial \phi}{\partial t} \right)^2 + e_1 \left(\frac{\partial \phi}{\partial x} \right)^2 + e_2 \left(\frac{\partial \phi}{\partial y} \right)^2, \tag{29}$$

where d , e_1 , and e_2 are constants. With the auxiliary variable $\Pi(x, y, t)$, we can write this in a form using only first-order time derivatives:

$$\frac{\partial \phi}{\partial t} = \Pi, \tag{30}$$

$$\frac{\partial \Pi}{\partial t} = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} + d (\Pi)^2 + e_1 \left(\frac{\partial \phi}{\partial x} \right)^2 + e_2 \left(\frac{\partial \phi}{\partial y} \right)^2. \tag{31}$$

Using the discretization introduced in Section 2.1, we have

$$\phi_{ij}^{n+1} = \phi_{ij}^n + (\Delta t) \bar{\Pi}_{ij}, \tag{32}$$

$$\begin{aligned} \Pi_{ij}^{n+1} = & \Pi_{ij}^n + \frac{\Delta t}{(\Delta x)^2} (\bar{\phi}_{i+1,j} - 2\bar{\phi}_{ij} + \bar{\phi}_{i-1,j}) + \frac{\Delta t}{(\Delta y)^2} (\bar{\phi}_{i,j+1} - 2\bar{\phi}_{ij} + \bar{\phi}_{i,j-1}) + d(\Delta t)(\bar{\Pi}_{ij})^2 \\ & + e_1(\Delta t) \left(\frac{\bar{\phi}_{i+1,j} - \bar{\phi}_{i-1,j}}{2\Delta x} \right)^2 + e_2(\Delta t) \left(\frac{\bar{\phi}_{i,j+1} - \bar{\phi}_{i,j-1}}{2\Delta y} \right)^2. \end{aligned} \quad (33)$$

As before, Eqs. (32) and (33) are updated following the steps given in Eqs. (6)–(15).

In this section we consider two types of GCF, the default Paramesh linear order GCF and a quadratic GCF scheme. The linear GCF is depicted in Fig. 9. First, each coarse grid guard cell (open diamond) is filled as a linear combination of the surrounding fine grid points (solid circles). The fine grid guard cells (open circles) are then filled using a linear combination of the surrounding coarse grid points (open and solid diamonds).

Quadratic GCF is depicted in Fig. 10. As a first step, the coarse grid guard cells are filled from a linear combination of the four surrounding fine grid guard cells. These values are only used at fine grid corners, and will soon be overwritten. Linear interpolation of the coarse grid cells (solid diamonds) parallel to the coarse–fine interface is used to compute intermediate values marked with open boxes in Fig. 10. These intermediate values, along with the two fine grid cells (solid circles) directly across the interface, are then used to obtain a quadratic fit for the fine grid guard cells marked with open circles.

Finally, as in the 1-D case, the coarse grid guard cells are filled by “flux matching”, that is, matching derivatives across the interface. Specifically, we consider the first derivative at the midpoint of Fig. 10, that is, at the point midway between the coarse grid guard cell (open diamond) and the interior cell directly across the interface (closed diamond). Derivative matching consists in equating the first derivative computed from these coarse grid cells with the second-order accurate first derivative obtained from the four fine grid cells (open and closed circles) that surround the midpoint.

The algorithm described here for quadratic GCF is similar to the one described by Martin and Cartwright [23]. The main difference is that we use linear interpolation of coarse grid values parallel to the interface to obtain intermediate values (the open boxes in Fig. 10), whereas Martin and Cartwright use quadratic interpolation. Also note that our algorithm can be applied without modification at fine grid corners, where the corner of a coarse grid block is surrounded by fine grid blocks. Recall that in the first

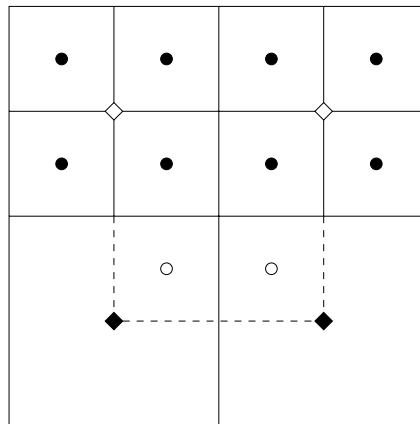


Fig. 9. Paramesh default linear GCF in 2-D. Points on the coarse grid are denoted by diamonds, and points on the fine grid by circles. First, the coarse grid guard cells (open diamonds) are filled by averaging the surrounding fine grid points. Then the fine grid guard cells (open circles) are filled by taking linear combinations of the four surrounding coarse grid points.

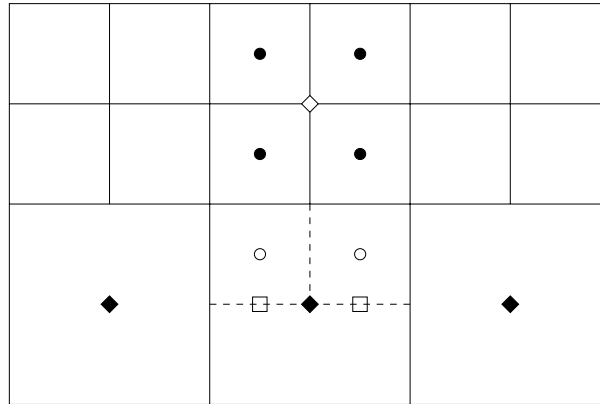


Fig. 10. Quadratic GCF in 2-D. Coarse grid guard cells (open diamonds) are temporarily filled by averaging the surrounding fine grid points. Intermediate values (open boxes) are obtained by linear interpolation of coarse grid values (solid diamonds) parallel to the interface. Fine grid guard cells (open circles) are then filled from a quadratic fit using two fine grid points and one intermediate value. Finally, the coarse grid guard cells are filled by matching the coarse and fine grid first derivatives across the interface.

step, coarse grid guard cells are filled by linear restriction from the fine grid. This allows the interpolation parallel to the interfaces to be carried out without the use of one-sided extrapolation. Finally, we point out that GCF at fine grid corners is ambiguous, since there are different ways to deal with them; either of the two coarse–fine interfaces that intersect at the corner can be used or interpolation using a stencil diagonal to the interfaces can also be used. Note that only mixed derivatives are affected by the corners when using centered differencing. In our code we do not treat the corners as special. At a corner our code naturally selects one of the two interfaces and carries out a linear interpolation parallel to that face to obtain intermediate values.

The initial data for our tests is taken to be a cylindrically symmetric wavepacket centered on the origin, with

$$\phi(x, y, t = 0) = Ae^{-(x^2+y^2)/\sigma^2} \tag{34}$$

and $\Pi(x, y, t = 0) = 0$. We choose the amplitude $A = 1$, and the width of the pulse by $\sigma = 0.25$. Quadrant symmetry is imposed by using mirror-symmetry boundary conditions along $x = 0$ and $y = 0$. The computational domain then covers the region $0 \leq x \leq 4.3125$ and $0 \leq y \leq 4.3125$. This packet is initially confined to a fine grid region of resolution $h = 0.0225$. As the packet expands, the wavefront crosses a fixed mesh refinement boundary into a region of coarser resolution $H = 2h$.

Setting $d = e_1 = e_2 = 0$ in Eqs. (30) and (31) allows this packet to evolve under a linear equation. Fig. 11 shows the results of using quadratic GCF to set the values of the guard cell data. Here, ϕ is shown at four consecutive times. The expanding wavefront encounters mesh refinement boundaries at $x = 2.1$ along the x -axis and at $y = 2.1$ along the y -axis. Note that the wave passes smoothly across the interface.

A comparison of unigrid and fixed refinement runs is shown in Fig. 12. Here, ϕ is plotted along a portion of the x -axis at a fixed time. The unigrid run (solid line) at the fine grid resolution shows the extended “tail” of the outgoing cylindrical wave front. The run with linear GCF (dotted line) shows a reflected wave traveling back into the fine grid region as the wave passes through the refinement boundary. In the run with quadratic GCF (dashed line), this spurious signal has been nearly eliminated.

Similar results are achieved when this wave packet is evolved according to a nonlinear equation, $d = -e_1 = -e_2 = 1$. The structure of the grid and location of the refinement boundary are the same as for the 2-D linear equation. Fig. 13 displays the results of ϕ along the x -axis at a fixed time. Notice that the run

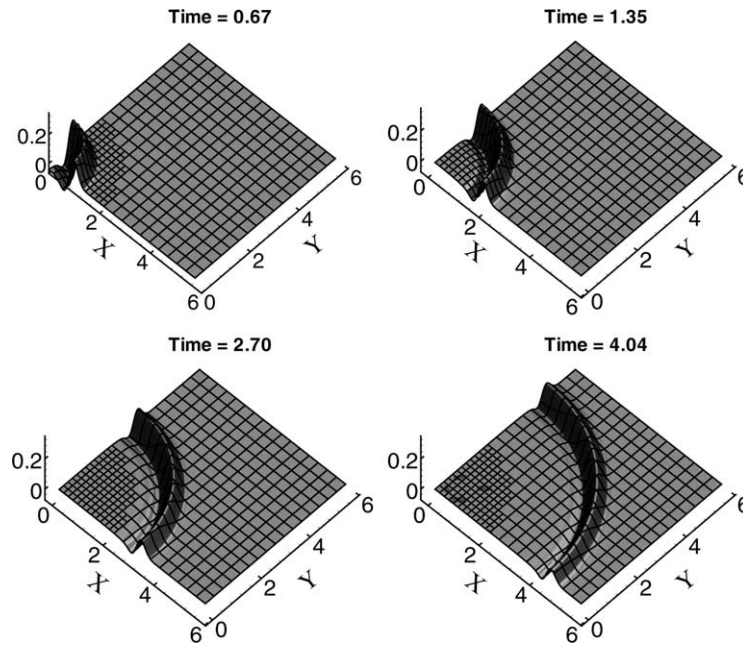


Fig. 11. The evolution of ϕ with a linear 2-D wave equation is shown at 4 consecutive times for a run using quadratic GCF to set the data in the guard cells. Each of the grid blocks shown has 8×8 zones.

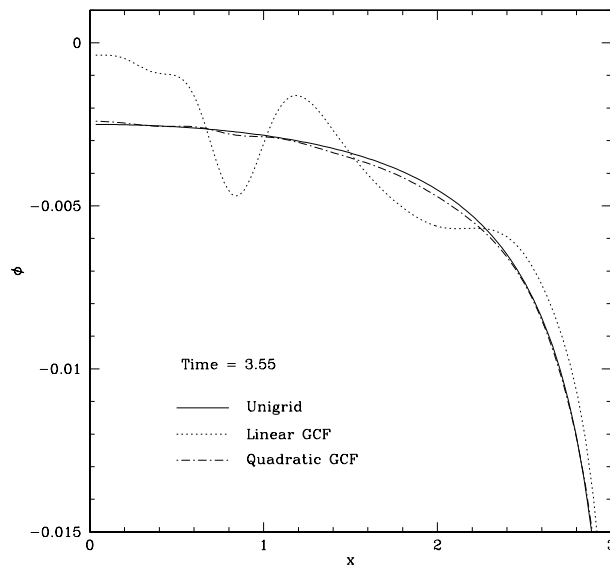


Fig. 12. ϕ is shown along a portion of the x -axis for the 2-D linear wave equation. The solid line shows the results of a unigrid run at the fine grid resolution $h = 0.0225$, the dotted line the results of a run with linear GCF, and the dashed line a run with quadratic GCF.

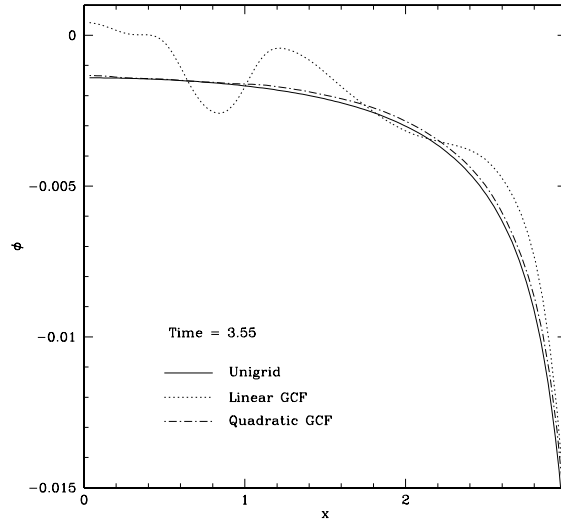


Fig. 13. ϕ is shown along a portion of the x -axis for the 2-D nonlinear wave equation. The solid line shows the results of a unigrid run at the fine grid resolution $h = 0.0225$, the dotted line the results of a run with linear GCF, and the dashed line a run with quadratic GCF.

with linear GCF (dotted line) shows a significant reflected wave. In contrast, the run with quadratic GCF (dashed line) is close to the one with a uniform grid (solid line).

7. The Einstein equations in 3-D

We are now ready to apply the techniques developed in our model equations to the propagation of gravitational waves in 3-D, which is governed by the vacuum (or source-free) Einstein equations. We write these equations in terms of the “3 + 1” spacetime split [4], in which the initial data is specified on some 3-D spacelike slice and then evolved forward in time. Within this framework, the metric takes the form

$$ds^2 = -\alpha^2 dt^2 + g_{ij}(dx^i + \beta^i dt)(dx^j + \beta^j dt). \tag{35}$$

We use units in which both the speed of light $c = 1$ and the gravitational constant $G = 1$. Lowercase Latin letters are used to denote spatial indices, so that $i, j = 1, 2, 3$. To simplify the notation throughout this section, we use the summation convention: if any expression has one index as a superscript and the same index as a subscript, summation over all values that index can take is implied [22]. The geometry of the given spacelike slice is described by the 3-metric g_{ij} . The lapse function α governs the advance of proper time across the surface, and the shift vector β^i the motion of the spatial coordinates within the hypersurface as the data is evolved forward in time. Both α and β^i are freely-specifiable functions of space and time; for the rest of this section, we use the choice $\alpha = 1$ and $\beta^i = 0$.

In the standard ADM spacetime split [4], the Einstein equations can be written in terms of g_{ij} and the extrinsic curvature of the hypersurface K_{ij} , where

$$K_{ij} = -\frac{1}{2} \frac{\partial g_{ij}}{\partial t}. \tag{36}$$

Following current practice in numerical relativity, we use the BSSN formalism [5,6] in which the Einstein equations are written in terms of conformal variables $\{\psi, K, \tilde{g}_{ij}, \tilde{A}_{ij}, \tilde{F}^i\}$ defined as follows:

$$e^{4\psi} \equiv \det(g_{ij})^{1/3}, \quad (37)$$

$$\tilde{g}_{ij} \equiv e^{-4\psi} g_{ij}, \quad (38)$$

$$K \equiv g^{ij} K_{ij}, \quad (39)$$

$$\tilde{A}_{ij} \equiv e^{-4\psi} (K_{ij} - \frac{1}{3} g_{ij} K), \quad (40)$$

$$\tilde{\Gamma}^i \equiv -\partial_j \tilde{g}^{ij}. \quad (41)$$

Here \tilde{g}^{ij} is the inverse of the conformal metric \tilde{g}_{ij} . We use the notation $\partial_j \equiv \partial/\partial x^j$ for spatial derivatives.

In terms of these conformal variables, with the gauge choices $\alpha = 1$ and $\beta^i = 0$, the vacuum Einstein equations become

$$\frac{\partial \psi}{\partial t} = -\frac{1}{6} \alpha K, \quad (42)$$

$$\frac{\partial \tilde{g}_{ij}}{\partial t} = -2\tilde{A}_{ij}, \quad (43)$$

$$\frac{\partial K}{\partial t} = \frac{1}{3} K^2 + \tilde{A}_{ij} \tilde{A}^{ij}, \quad (44)$$

$$\frac{\partial \tilde{A}_{ij}}{\partial t} = R_{ij}^{\text{TF}} + \tilde{A}_{ij} K - 2\tilde{A}_{il} \tilde{A}^l_j, \quad (45)$$

$$\frac{\partial \tilde{\Gamma}^i}{\partial t} = 2(\tilde{\Gamma}^i_{jk} \tilde{A}^{kj} - \frac{2}{3} \tilde{g}^{ij} \partial_j K + 6\tilde{A}^{ij} \partial_j \psi). \quad (46)$$

Here, $\tilde{\Gamma}^i_{jk}$ are the connection coefficients associated with \tilde{g}_{ij} , defined by

$$\tilde{\Gamma}^k_{ji} = \frac{1}{2} \tilde{g}^{mk} (\partial_i \tilde{g}_{mj} + \partial_j \tilde{g}_{mi} - \partial_m \tilde{g}_{ji}), \quad (47)$$

and

$$\tilde{A}^{ij} = \tilde{g}^{il} \tilde{g}^{jk} \tilde{A}_{lk}, \quad \tilde{A}^l_j = \tilde{g}^{li} \tilde{A}_{ij}. \quad (48)$$

The superscript ‘‘TF’’ denotes the trace-free part of a tensor, so that $R_{ij}^{\text{TF}} = R_{ij} - g_{ij} R/3$, where $R = g^{mk} R_{mk}$. The Ricci curvature tensor R_{ij} is defined by

$$R_{ij} = \partial_k \Gamma^k_{ij} - \partial_j \Gamma^k_{ik} + \Gamma^k_{mk} \Gamma^m_{ij} - \Gamma^k_{mj} \Gamma^m_{ik}. \quad (49)$$

Although the set of Eqs. (42)–(46) is considerably more complicated than our model equations, there are notable similarities. In particular, the conformal metric \tilde{g}_{ij} plays the role of the function ϕ , while A_{ij} takes the role of Π . Looking at (47) and (49), we also see that the R_{ij}^{TF} term in Eq. (45) contains second spatial derivatives of \tilde{g}_{ij} .

The lessons learned from the model equations in 1-D and 2-D can be applied successfully to the Einstein equations in 3-D, as we demonstrate by evolving a weak gravitational wave. We use the analytic solution to the linearized Einstein equations found by Teukolsky [24]; since this is given in closed form, we can then

compare the numerical results directly with this analytic solution. We choose the even parity, $L = 2, M = 0$ solution, which is given by

$$ds^2 = -dt^2 + (1 + Af_{rr})dr^2 + (2Bf_{r\theta})rdrd\theta + (2Bf_{r\phi})r\sin\theta drd\phi + \left(1 + Cf_{\theta\theta}^{(1)} + Af_{\theta\theta}^{(2)}\right)r^2 d\theta^2 + [2(A - 2C)f_{\theta\phi}]r^2 \sin\theta d\theta d\phi + \left(1 + Cf_{\phi\phi}^{(1)} + Af_{\phi\phi}^{(2)}\right)r^2 \sin^2\theta d\phi^2. \tag{50}$$

Here,

$$A = 3 \left[\frac{F^{(2)}}{r^3} + \frac{3F^{(1)}}{r^4} + \frac{3F}{r^5} \right], \tag{51}$$

$$B = - \left[\frac{F^{(3)}}{r^4} + \frac{3F^{(2)}}{r^3} + \frac{6F^{(1)}}{r^4} + \frac{6F}{r^5} \right], \tag{52}$$

$$C = \frac{1}{4} \left[\frac{F^{(4)}}{r} + \frac{2F^{(3)}}{r^2} + \frac{9F^{(2)}}{r^3} + \frac{21F^{(1)}}{r^4} + \frac{21F}{r^5} \right], \tag{53}$$

$$F = F(t - r), \quad F^{(n)} \equiv \left[\frac{d^n F(x)}{dx^n} \right]_{x=t-r}, \tag{54}$$

where F is a generating function. We use the form

$$F(x) = \frac{Ax}{\omega^2} e^{-x^2/\omega^2}, \tag{55}$$

with two free parameters, A and ω . Here we have specified an outgoing wave solution $F = F(t - r)$; an ingoing wave solution can be obtained by using $F = F(t + r)$.

For this even-parity, $M = 0$ case, the angular functions f_{ij} are:

$$f_{rr} = 2 - 3 \sin^2 \theta, \tag{56}$$

$$f_{r\theta} = -3 \sin \theta \cos \theta, \tag{57}$$

$$f_{r\phi} = 0, \tag{58}$$

$$f_{\theta\theta}^{(1)} = 3 \sin^2 \theta, \tag{59}$$

$$f_{\theta\theta}^{(2)} = -1, \tag{60}$$

$$f_{\theta\phi} = 0, \tag{61}$$

$$f_{\phi\phi}^{(1)} = -f_{\theta\theta}^{(1)}, \tag{62}$$

$$f_{\phi\phi}^{(2)} = 3 \sin^2 \theta - 1. \tag{63}$$

We present results for a gravitational wave crossing two fixed mesh refinement boundaries into regions with successively coarser resolution. We start with a wave packet composed of a linear combination of one initially ingoing and one outgoing wave, each having amplitude $A = 10^{-6}$ and width $\omega = 1$. This packet is centered on the origin in a fine grid region of resolution $h = 0.0416667$. The successively coarser regions have resolutions $2h$ and $4h$, with the first refinement boundary at $r = \sqrt{x^2 + y^2 + z^2} = 4.5$ and the second at $r = 9.0$. To complete the initial data we take $K_{ij} = 0$ so that $K = 0$ and $\dot{A}_{ij} = 0$. Octant symmetry is imposed by using mirror-symmetry boundary conditions along $x = 0$, $y = 0$, and $z = 0$. The computational domain covers the regions $0 \leq x \leq 12$ and similarly for y and z .

As the evolution proceeds, the outgoing waves travel directly toward the outer boundary of the grid. The initially ingoing waves first travel toward the origin, then reflect and move outward. As the overall signal propagates outward, it leaves flat spacetime behind.

Fig. 14 shows the evolution of these waves when linear GCF is used. The function $g_{zz} - 1$ is plotted as a function of x and y in the $z = 0$ plane at 4 successive times. Note the presence of spurious reflected signals as the waves pass through the fixed mesh boundaries. These problems are greatly reduced when quadratic GCF is used, as shown in Fig. 15. A comparison of runs with linear (dotted line) and quadratic (dashed line) GCF and the analytic solution (solid line) is shown in Fig. 16. The reflected waves are essentially eliminated by the use of quadratic GCF.

Finally, Fig. 17 demonstrates the second-order convergence of the code by comparing the results of the run in Fig. 15 with a run that differs only by having the size of the grid zones a factor of 2 larger

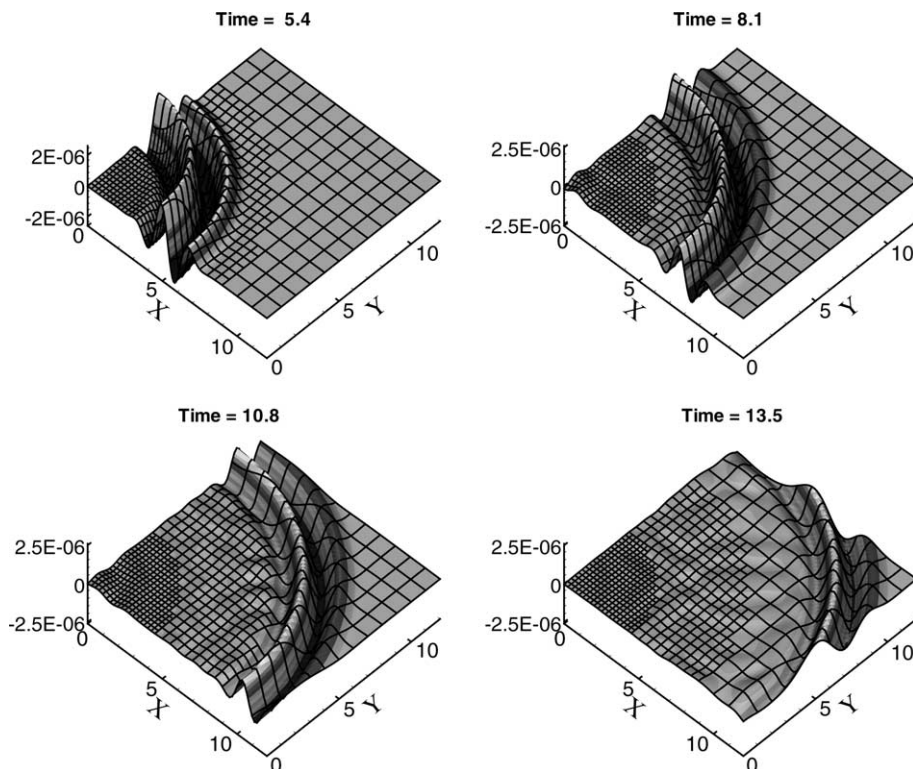


Fig. 14. Evolution of gravitational waves in the 3-D Einstein equations using linear GCF. $g_{zz} - 1$ is plotted in the $z = 0$ plane. Three levels of resolution ($h, 2h, 4h$) are used, with $h = 0.0416667$. Each of the grid blocks shown has $6 \times 6 \times 6$ zones.

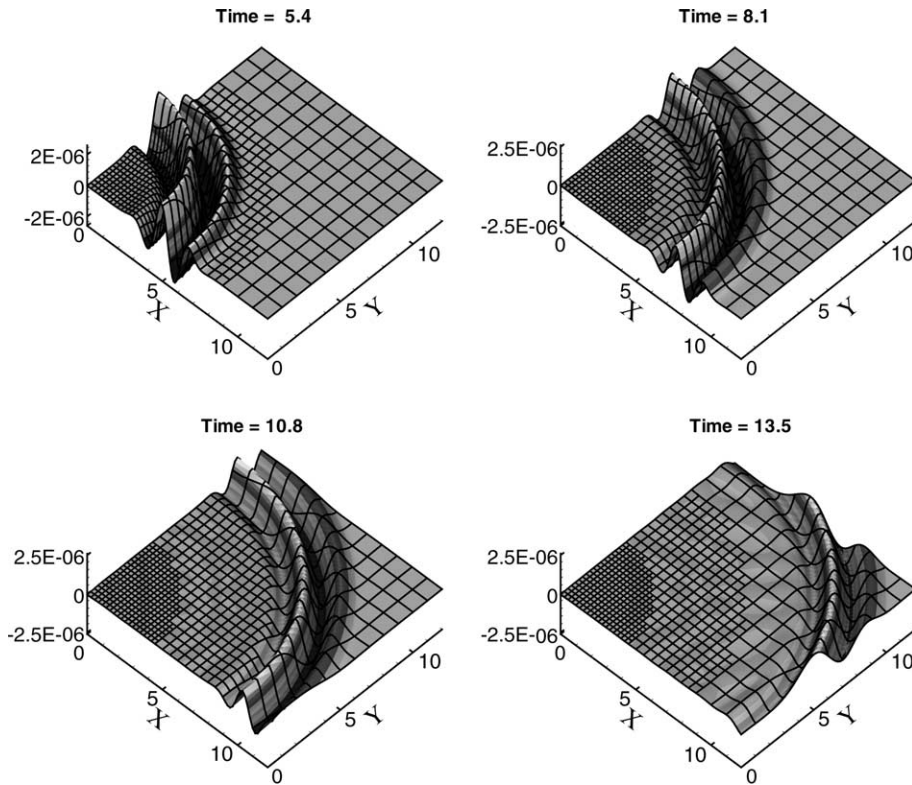


Fig. 15. Evolution of gravitational waves in the 3-D Einstein equations using quadratic GCF. $g_{zz} - 1$ is plotted in the $z = 0$ plane. Three levels of resolution ($h, 2h, 4h$) are used, with $h = 0.0416667$. Each of the grid blocks shown has $6 \times 6 \times 6$ zones.

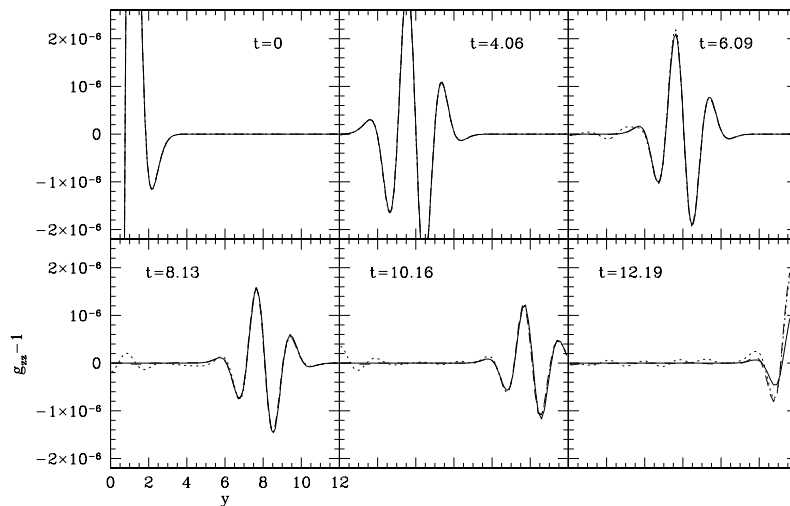


Fig. 16. The evolution of $g_{zz} - 1$ is shown along the y -axis for the 3-D Einstein equations. The analytic solution (solid line) and numerical solutions using linear (dotted line) and quadratic (dashed line) GCF are shown. The resolution levels are given by ($h, 2h, 4h$) with $h = 0.0416667$.

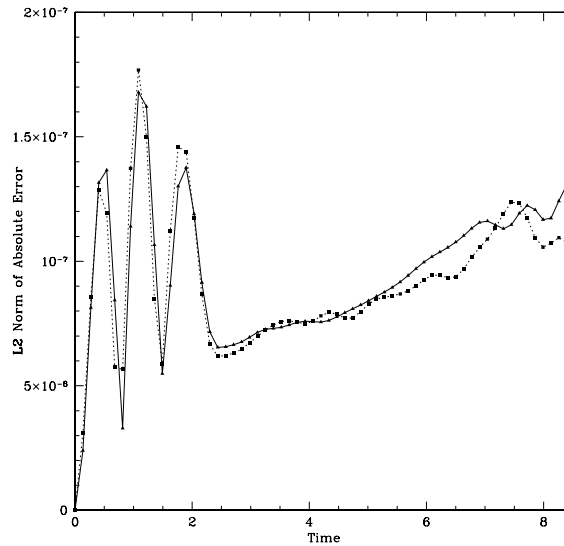


Fig. 17. The L2 norm of the absolute error ϵ is shown for two gravitational wave runs differing in resolution by a factor of 2 throughout. Both runs have two refinement boundaries and use quadratic GCF. The solid triangles connected by the solid line show ϵ for the run at higher overall resolution, and the solid boxes connected by the dotted line show $4 \times \epsilon$ for the run with lower overall resolution.

throughout. Both runs use quadratic GCF. The L2 norm of the absolute error ϵ is calculated over each simulation domain, and plotted as a function of time. The solid triangles connected by the solid line show ϵ for the run in Fig. 15, and the filled boxes connected by the dotted line show the errors for the lower resolution run multiplied by 4.

8. Summary

We have examined the propagation of waves across fixed mesh refinement boundaries, starting with simplified linear and nonlinear model equations in 1-D and 2-D, and progressing to the 3-D Einstein equations of general relativity. The numerical evolutions were carried out using centered spatial differences and the explicit iterated Crank–Nicholson time update method, giving second-order accuracy. Our results show that using linear GCF produces spurious reflected waves as the signals cross refinement boundaries, and that these are greatly suppressed by using quadratic GCF. In particular, quadratic GCF preserves the second-order convergence of the numerical evolutions. Our numerical results are complemented by a detailed analytic treatment of waves crossing refinement boundaries in 1-D in Appendix A.

While quadratic GCF is straightforward to describe and implement in 1-D, the situation becomes more complicated in 2-D. In particular, intermediate values parallel to the mesh refinement interface must be calculated in the 2-D case. We have found that using linear interpolation to obtain these intermediate values, combined with quadratic interpolation for the final values, maintains the second-order convergence. The procedure used for quadratic GCF in 2-D generalizes in a straightforward manner to the 3-D case.

The techniques presented here appear to be robust in the sense that they continue to produce excellent results as our test problems increase in complexity. Quadratic GCF successfully eliminates most of the spurious reflected waves in both linear and nonlinear model equations in 1-D and 2-D. The 3-D Einstein equations present a much larger and more complex system of equations. In the test case presented here, the

evolution of a weak gravitational wave, quadratic GCF continues to perform well, even as the signals cross two successive mesh refinement boundaries. We fully expect that these techniques will also yield excellent results for strong gravitational waves, which activate the nonlinear terms in the Einstein equations. Such evolutions require various technical differences in the gauge choices (α and β^i) as well as in the formulation of the initial data. We are currently working on such models, and will report on them elsewhere.

Acknowledgements

It is a pleasure to thank John Baker, Phillip Colella, Kevin Olson, and Steve Zalesak for helpful and stimulating discussions. The work was supported in part by NSF grant PHY-0070892.

Appendix A. Analysis of numerical wave propagation in 1-D

In this appendix, we present a more detailed analysis of the propagation of linear waves in 1-D with the discretization described in Section 2. We begin by deriving some basic results for uniform grids (Section A.1) and follow with a study of wave propagation across a fixed mesh refinement boundary (Section A.2). Here, we do not address the issue of instabilities that might arise due to coupling between fine and coarse meshes [25]. However, as noted in Section 4, our numerical tests show no signs of instability.

A.1. Wave propagation on a uniform mesh

We will employ matrix notation to facilitate the analysis in this Appendix. First, we collect the field variables ϕ , Π into the column vector

$$V = \begin{pmatrix} \phi \\ \Pi \end{pmatrix}. \quad (\text{A.1})$$

Eqs. (2) and (3) can now be written as

$$\frac{\partial V}{\partial t} = \begin{pmatrix} 0 & 1 \\ \partial^2/\partial x^2 & 0 \end{pmatrix} V. \quad (\text{A.2})$$

As usual V_j^n will denote the vector of grid functions at timestep n and grid point j . The iterated Crank–Nicholson method described in Section 2.1 is built from successive applications of the basic operator

$$Q = \begin{pmatrix} 0 & 1 \\ \partial^2 & 0 \end{pmatrix}, \quad (\text{A.3})$$

where $\partial^2 V_j^n \equiv (V_{j+1}^n - 2V_j^n + V_{j-1}^n)/\Delta x^2$. With two iterations, the update of the variables V_j^n by one full timestep is accomplished by the operator

$$M = I + \Delta t Q \left[I + \frac{\Delta t}{2} Q \left(I + \frac{\Delta t}{2} Q \right) \right]. \quad (\text{A.4})$$

The stability, dissipation, and dispersion properties are obtained by considering discrete plane wave solutions,

$$V_j^n = W e^{i\omega n \Delta t} e^{-ikj \Delta x}, \quad (\text{A.5})$$

where W is a constant vector (independent of n and j). Inserting this ansatz into the update equation $V_j^{n+1} = MV_j^n$, we find

$$e^{i\omega\Delta t}W = \begin{pmatrix} 1 - 2A^2 & \Delta t(1 - A^2) \\ -4A^2(1 - A^2)/\Delta t & 1 - 2A^2 \end{pmatrix}W, \quad (\text{A.6})$$

where

$$A \equiv \frac{\Delta t}{\Delta x} \sin(k\Delta x/2). \quad (\text{A.7})$$

Thus, W is an eigenvector with eigenvalue $e^{i\omega\Delta t}$ for the matrix that appears in Eq. (A.6). The eigenvalues are obtained in the usual way with the result $e^{i\omega\Delta t} = 1 - 2A^2 \pm 2iA(1 - A^2)$. This is the dispersion relation giving the complex frequency ω as a function of wave number k . We can, without loss of generality, consider only plane wave solutions (A.5) with positive frequency $\xi > 0$, where $\xi = \Re(\omega)$ is the real part of ω . Then the \pm sign in the dispersion relation must be set equal to the sign of the wave number k . The dispersion relation then becomes

$$e^{i\omega\Delta t} = 1 - 2A^2 + 2i|A|(1 - A^2) \quad (\text{A.8})$$

and $\xi = \Re(\omega)$ is positive. The eigenvectors W corresponding to these eigenvalues are straightforward to compute. Choosing the first component of W to be unity, we find

$$W = \begin{pmatrix} 1 \\ 2i|A|/\Delta t \end{pmatrix}. \quad (\text{A.9})$$

We note for later reference that $We^{-ikj\Delta x}$ is an eigenvector of the basic operator Q with eigenvalue $2i|A|/\Delta t$.

The finite difference scheme is unstable if the magnitude of the amplification factor, $|e^{i\omega\Delta t}|$, is greater than unity. From Eq. (A.8) we find that $|e^{i\omega\Delta t}|^2 \leq 1$ implies $A^2 \leq 1$. This inequality will be satisfied for all wave numbers k only if $\Delta t \leq \Delta x$. This is the Courant limitation on the timestep for the wave Eqs. (2) and (3) discretized with twice-iterated Crank–Nicholson. The phase velocity is found from the real part of the frequency $\xi = \Re(\omega)$. From the dispersion relation (A.8), we find

$$\xi\Delta t = \arcsin \left(\frac{2|A|(1 - A^2)}{\sqrt{1 - 4A^4(1 - A^2)}} \right). \quad (\text{A.10})$$

The phase velocity is then

$$c(\lambda) = \frac{\xi}{k} = \frac{\xi\Delta t}{2\pi\alpha} \frac{\lambda}{\Delta x}, \quad (\text{A.11})$$

where $\alpha \equiv \Delta t/\Delta x$ is the Courant factor and $\lambda = 2\pi/k$ is the wavelength. The dissipation is found from the imaginary part of the frequency, $\eta = \Im(\omega)$. Since the wave amplitude varies like $\phi \sim e^{-\eta m\Delta t}$, we see that the amplitude drops by a factor

$$e^{-\eta\Delta t} = |e^{i\omega\Delta t}| = \sqrt{1 - 4A^4(1 - A^2)} \quad (\text{A.12})$$

for each timestep.

A.2. Wave propagation with FMR

Now consider a two-level refined mesh, with fine grid Δx_f on the left and coarse grid Δx_c on the right. We will assume that the refinement jumps by a factor of 2, that is, $\Delta x_c = 2\Delta x_f$. The mesh will be labeled as shown in Fig. 3. Thus, $V_{-1/2}^n, V_{-3/2}^n$, etc. are the fine grid functions and $V_{1/2}^n, V_{3/2}^n$, etc. are the coarse grid functions.

As a first step towards analyzing the wave reflection and transmission at the interface, we relate the wave numbers in the coarse and fine grid regions. Consider a monochromatic solution that varies like $\phi \sim e^{i\xi n \Delta t}$ across the entire mesh. Specifically, we assume that the coarse and fine grid frequencies are the same, $\xi_c = \xi_f$, and that the coarse and fine grid time steps are the same, $\Delta t_c = \Delta t_f$. From the dispersion relation, Eq. (A.8), we can compute $\tan(\xi \Delta t)$ in the coarse and fine grid regions and equate the results:

$$\frac{2|A_c|(1 - A_c^2)}{1 - 2A_c^2} = \frac{2|A_f|(1 - A_f^2)}{1 - 2A_f^2}. \tag{A.13}$$

Here, $A_c = (\Delta t / \Delta x_c) \sin(k_c \Delta x_c / 2)$ and similarly for A_f . This relation has the form $f(|A_c|) = f(|A_f|)$ where $|A_c|$ and $|A_f|$ vary between 0 and 1. It is easy to show that the function $f(|A|)$ is monotonic and therefore invertible. It follows that the only solution of Eq. (A.13) is

$$|A_c| = |A_f|. \tag{A.14}$$

This equation shows that the coarse and fine grid wave numbers k_c and k_f are related by

$$k_c = \pm \frac{2}{\Delta x_c} \arcsin \left[\frac{\Delta x_c}{\Delta x_f} \sin(k_f \Delta x_f / 2) \right]. \tag{A.15}$$

The two cases corresponding to the \pm sign indicate that the wave propagation direction on the coarse and fine sides of the interface need not match. Thus, we can have a right moving wave in the coarse grid region connected to both right moving and left moving waves in the fine grid region.

From the result (A.14) we see that the rate of dissipation (A.12) of a wave, governed by $\eta = \Im(\omega)$, is the same in the coarse and fine grid regions. We also see that the relative phase between the two components ϕ and Π of the wave, Eq. (A.9), is the same in coarse and fine regions. The phase velocity (A.11), and hence the amount of dispersion, differ in the coarse and fine grid regions, since the wave numbers k_c and k_f are not equal.

According to Eq. (A.15) k_c is real only for $k_f \Delta x_f \leq \pi/3$, that is, for $\lambda_f / \Delta x_f \geq 6$. If $\lambda_f / \Delta x_f < 6$, then k_c is complex and the plane wave solution (A.5) will contain a spatial dependence in the coarse grid region that is either exponentially damped or grows exponentially. Note that, although k_c might be complex, A_c is real (assuming k_f is real) and equal to $\pm A_f$. It follows that, whether k_c is real or complex, the Courant stability condition $|e^{i\omega \Delta t}|^2 \leq 1$ is satisfied in the coarse grid region if it is satisfied in the fine grid region.

For the remainder of this appendix we will focus on the case of practical interest, where $\lambda_f / \Delta x_f \geq 6$ and k_c is real. The plots in Figs. 5–7 have been restricted to $\lambda_f / \Delta x_f \geq 10$ for clarity of presentation. Each of the curves in those plots reaches a finite value at $\lambda_f / \Delta x_f = 6$.

A.2.1. Matching solutions

At this point we have shown that waves of frequency ξ have wave number k_c on a coarse grid, wave number k_f on a fine grid, and that these values are related as in Eq. (A.15). We will now construct a solution with frequency ξ that spans the entire non-uniform grid. To begin, consider the vector

$$V_j = \begin{cases} W(e^{-ik_f j \Delta x_f} + \mathbf{R}e^{ik_f j \Delta x_f}), & j < 0, \\ W(\mathbf{T}e^{-ik_c j \Delta x_c}), & j > 0. \end{cases} \tag{A.16}$$

We will show that for an appropriate choice of the coefficients \mathbf{R} and \mathbf{T} the vector V_j is an eigenvector of the basic operator Q with eigenvalue $2i|A|/\Delta t$. For points away from the interface, namely, the points $j \leq -3/2$ and $j \geq 3/2$, this conclusion follows from our earlier observation that on a uniform grid $W e^{\pm ikj \Delta x}$ is an eigenvector of Q with eigenvalue $2i|A|/\Delta t$. The same argument cannot be applied to the points $1/2$ and $-1/2$ surrounding the interface because the stencil for the discrete derivative operator ∂^2 appearing in Q extends across the interface. Thus, when computing $\partial^2 V_j$ for $j = \pm 1/2$, we must use guard cell information.

In the main text, we discussed various choices for guard cell filling, such as the Paramesh linear GCF of Eqs. (17) and (18) and the quadratic GCF of Eqs. (19)–(21). For the purpose of presenting the analysis, we will focus instead on the direct linear GCF of Eqs. (22) and (23). In the present notation, these relations are

$$V_g^n = \frac{1}{2}(V_{-3/2}^n + V_{-1/2}^n), \quad V_g^n = \frac{1}{3}(V_{-1/2}^n + 2V_{1/2}^n). \tag{A.17}$$

Now, for grid points that are not adjacent to the interface, the operator ∂^2 takes the usual form,

$$\partial^2 V_j^n = \begin{cases} (V_{j+1}^n - 2V_j^n + V_{j-1}^n)/\Delta x_f^2, & j \leq -3/2, \\ (V_{j+1}^n - 2V_j^n + V_{j-1}^n)/\Delta x_c^2, & j \geq 3/2. \end{cases} \tag{A.18}$$

But for grid points adjacent to the interface, ∂^2 must use guard cell values given by (A.17). Consequently, we find

$$\partial^2 V_{-1/2}^n = (V_g^n - 2V_{-1/2}^n + V_{-3/2}^n)/\Delta x_f^2 = (2V_{1/2}^n - 5V_{-1/2}^n + 3V_{-3/2}^n)/(3\Delta x_f^2) \tag{A.19}$$

and

$$\partial^2 V_{1/2}^n = (V_{3/2}^n - 2V_{1/2}^n + V_g^n)/\Delta x_c^2 = (2V_{3/2}^n - 4V_{1/2}^n + V_{-1/2}^n + V_{-3/2}^n)/(2\Delta x_c^2) \tag{A.20}$$

for ∂^2 acting at grid points $j = \pm 1/2$.

We now impose the requirement that the vector V_j of Eq. (A.16) is an eigenvector of Q with eigenvalue $2i|A|/\Delta t$ at the points $j = \pm 1/2$ adjacent to the interface. Using the discretization (A.19) the relation $QV_{-1/2} = (2i|A|/\Delta t)V_{-1/2}$ yields

$$\frac{1}{3}(2\phi_{1/2} - 5\phi_{-1/2} + 3\phi_{-3/2}) = \left(\frac{2i|A|}{\alpha_f}\right)^2 \phi_{-1/2}. \tag{A.21}$$

Here, ϕ_j is the first component of the ansatz vector V_j . Similarly, with the discrete operator (A.20), we find that $QV_{1/2} = (2i|A|/\Delta t)V_{1/2}$ implies

$$\frac{1}{2}(2\phi_{3/2} - 4\phi_{1/2} + \phi_{-1/2} + \phi_{-3/2}) = \left(\frac{2i|A|}{\alpha_c}\right)^2 \phi_{1/2}. \tag{A.22}$$

These two equations can be solved for the two coefficients \mathbf{R} and \mathbf{T} . The result is

$$\mathbf{R} = \frac{3E_c^2 E_f^2 - E_f^4 - E_c^2 E_f^4 - E_f^6}{1 + E_f^2 + E_c^2 E_f^2 - 3E_c^2 E_f^4}, \tag{A.23}$$

$$\mathbf{T} = \frac{E_c(3 + 2E_f^2 - 2E_f^6 - 3E_f^8)/(2E_f)}{1 + E_f^2 + E_c^2 E_f^2 - 3E_c^2 E_f^4},$$

where the shorthand notation

$$E_c \equiv e^{ik_c \Delta x_c / 2}, \quad E_f \equiv e^{ik_f \Delta x_f / 2} \tag{A.24}$$

has been used.

At this point we have succeeded in showing that the vector V_j of Eq. (A.16), with \mathbf{R} and \mathbf{T} chosen as in Eq. (A.23), is an eigenvector of Q on the non-uniform grid. The corresponding eigenvalue is $2i|A|/\Delta t$. A short calculation shows that V_j is also an eigenvector for M , Eq. (A.4), with eigenvalue $1 - 2A^2 + 2i|A|(1 - A^2)$. Since M evolves the discrete system by one time step, we see that

$$V_j^n = \begin{cases} W e^{i\omega n \Delta t} (e^{-ik_f j \Delta x_f} + R e^{ik_f j \Delta x_f}), & j < 0, \\ W e^{i\omega n \Delta t} (T e^{-ik_c j \Delta x_c}), & j > 0. \end{cases} \tag{A.25}$$

is a solution of the finite difference equations $V_j^{n+1} = M V_j^n$ across the entire grid. Here, the complex frequency ω is given by the dispersion relation (A.8). The solution (A.25) represents a wave that travels from the fine grid region to the coarse grid region. At the interface it splits into reflected and transmitted pieces. The coefficient \mathbf{R} is the reflection coefficient, and \mathbf{T} is the transmission coefficient.

The results of this analysis can be checked numerically. For example, in Fig. 18 the solid lines show the magnitude of the reflection coefficient calculated from Eq. (A.23). The squares and triangles display the results of a numerical test, in which we modeled the propagation of a sine wave contained in a broad Gaussian envelope,

$$\phi(x, t) = A e^{-(x-x_0-t)^2/w^2} \sin k(x-t). \tag{A.26}$$

The initial data were obtained by discretizing ϕ and $d\phi/dt$ at $t = 0$. The constant x_0 was chosen so that initially the wave packet was situated in the fine grid region, far from the interface. We set the Gaussian half-width to $w = 63\Delta x$, significantly larger than the longest wavelength shown in the figures. The magnitude of \mathbf{R} was obtained by evolving the wave packet past its interaction with the interface, typically about 1000 time steps, and then extracting from the numerical data the largest value of $|\phi|$ in the reflected pulse. The Courant factor for these numerical runs was $\Delta t/\Delta x_f = 0.4$.

The squares show raw numerical data. The deviation of these data from the analytical curves is due to dissipation and dispersion of the wave pulse. Dissipation can be accounted for rather easily, using the

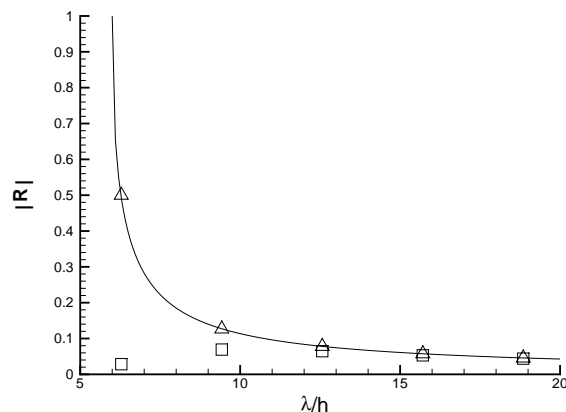


Fig. 18. Absolute value of the reflection coefficient for direct linear guard cell filling. The raw numerical data are shown as boxes, while the triangles show the data corrected for dissipation.

analytical result given in Eq. (A.12). The triangles show the numerical data with correction for dissipation. The numerical and analytical results for the reflection coefficient are in excellent agreement, as seen in Fig. 18.

The above analysis can be repeated for other choices of GCF. In particular, for the quadratic GCF of Eqs. (19)–(21), the reflection and transmission coefficients are

$$\begin{aligned} \mathbf{R} &= \frac{E_f^2(-1 + 6E_f^2 + 3E_f^4 + E_c^2(-15 + 10E_f^2 - 3E_f^4))}{3 + 6E_f^2 - E_f^4 - E_c^2(3 - 10E_f^2 + 15E_f^4)}, \\ \mathbf{T} &= \frac{2E_c(-3 - 2E_f^2 + 2E_f^6 + 3E_f^8)/E_f}{3 + 6E_f^2 - E_f^4 - E_c^2(3 - 10E_f^2 + 15E_f^4)}, \end{aligned} \quad (\text{A.27})$$

with E_c and E_f defined as before. Figs. 5–7 compare the reflection and transmission coefficients for linear, direct linear, and quadratic GCF. From Fig. 5 we see that quadratic GCF produces a much smaller reflected wave than either linear or direct linear GCF. For example, at 20 fine grid points per wavelength, linear (direct linear) GCF produces a reflected wave with amplitude about 5.6% (4.3%) that of the incident wave. With quadratic GCF the reflected wave amplitude is only about 0.31% that of the incident wave. For the transmitted wave, Fig. 7 shows that the phase error at 20 fine grid points per wavelength is smaller in magnitude by more than a factor of 10 with quadratic GCF compared to linear or direct linear GCF. Fig. 6 shows, perhaps surprisingly, that direct linear GCF does the best job of keeping the magnitude of \mathbf{T} close to 1 at wavelengths less than about $28\Delta x_f$. For longer wavelengths, $\lambda > 28\Delta x_f$, quadratic GCF is best. Note, however, that with any of these choices of GCF, the deviation of $|\mathbf{T}|$ away from unity is relatively small. Over the entire range shown in the graphs, $\lambda > 10\Delta x_f$, the maximum error for quadratic GCF is less than 1%. For the reflection coefficient, on the other hand, the maximum error for direct linear GCF is about 11%. For most situations the problem of spurious reflections at an interface will be more severe than the problem of inaccurate wave transmission through the interface.

A.2.2. Discussion of guard cell filling

The results thus far indicate that quadratic GCF is generally better than either linear or direct linear GCF at keeping the reflection coefficient small and the transmission coefficient close to unity. The question naturally arises: can one do better than quadratic GCF? We will restrict our attention to rules for GCF that use a three-point stencil. That is, the fine and coarse grid guard cell values V_g^n and V_G^n are obtained from linear combinations of three grid points,

$$V_G^n = c_1 V_{1/2}^n + c_2 V_{-1/2}^n + c_3 V_{-3/2}^n, \quad (\text{A.28})$$

$$V_g^n = f_1 V_{1/2}^n + f_2 V_{-1/2}^n + f_3 V_{-3/2}^n, \quad (\text{A.29})$$

where c_1, f_1 , etc. are constants.

The analysis of Section A.2.1 can be repeated with the guard cells defined as above. The resulting reflection and transmission coefficients are functions of the constants c_1, f_1 , etc. Although there are certain combinations of the constants that outperform quadratic GCF at low resolution, quadratic GCF is unique in the following sense. If we consider the high resolution limit, in which $k_f \Delta x_f$ is small, quadratic GCF yields

$$\mathbf{R} = \frac{3}{32}i(k_f \Delta x_f)^3 + \mathcal{O}(k_f \Delta x_f)^4, \quad (\text{A.30})$$

$$\mathbf{T} = 1 - \frac{3}{32}i(k_f \Delta x_f)^3 + \mathcal{O}(k_f \Delta x_f)^4. \quad (\text{A.31})$$

The magnitudes of the reflection and transmission coefficients behave like $|\mathbf{R}| = \mathcal{O}(k_f \Delta x_f)^3$ and $|\mathbf{T}| = 1 + \mathcal{O}(k_f \Delta x_f)^4$. For any other choice of constants c_1, f_1 , etc., the reflection and transmission coefficients approach 0 and 1, respectively, more slowly (if at all) than for quadratic GCF. For example, for direct linear GCF, we have

$$\mathbf{R} = \frac{1}{8}i(k_f \Delta x_f) + \mathcal{O}(k_f \Delta x_f)^2, \quad (\text{A.32})$$

$$\mathbf{T} = 1 + \frac{1}{8}i(k_f \Delta x_f) + \mathcal{O}(k_f \Delta x_f)^2. \quad (\text{A.33})$$

Thus, for high resolution, quadratic GCF is the best possible choice given the three-point stencil (A.28) and (A.29). Because \mathbf{R} and \mathbf{T} approach 0 and 1 rapidly, as the third power of $k_f \Delta x_f$, quadratic GCF performs well at all resolutions.

References

- [1] B. Barish, First generation interferometers, in: J. Centrella (Ed.), *Astrophysical Sources for Ground-Based Gravitational Wave Detectors*, AIP, Melville, NW, 2001, p. 3.
- [2] P. Fritschel, The second generation LIGO interferometers, in: J. Centrella (Ed.), *Astrophysical Sources for Ground-Based Gravitational Wave Detectors*, AIP, Melville, NW, 2001, p. 15.
- [3] P. Bender et al., LISA, Pre-Phase A Report, second ed., 1998, unpublished. Available from <http://lisa.jpl.nasa.gov/documents/ppa2-09.pdf>.
- [4] C. Misner, K. Thorne, J. Wheeler, *Gravitation*, W.H. Freeman, New York, 1973.
- [5] M. Shibata, T. Nakamura, Evolution of three-dimensional gravitational waves: Harmonic slicing case, *Phys. Rev. D* 52 (1995) 5428.
- [6] T. Baumgarte, S. Shapiro, Numerical integration of Einstein's field equations, *Phys. Rev. D* 59 (1998) 024007.
- [7] M. Choptuik, Universality and scaling in gravitational collapse of a massless scalar field, *Phys. Rev. Lett.* 70 (1993) 9.
- [8] B. Brügmann, Adaptive mesh and geodesically sliced Schwarzschild spacetime in 3 + 1 dimensions, *Phys. Rev. D* 54 (1996) 7361.
- [9] B. Brügmann, Binary black hole mergers in 3d numerical relativity, *Int. J. Mod. Phys. D* 8 (1999) 85.
- [10] P. Papadopoulos, E. Seidel, L. Wild, Adaptive computation of gravitational waves from black hole interactions, *Phys. Rev. D* 58 (1998) 084002.
- [11] K. New, D. Choi, J. Centrella, P. MacNeice, M. Huq, K. Olson, Three-dimensional adaptive evolution of gravitational waves in numerical relativity, *Phys. Rev. D* 62 (2000) 084039.
- [12] S. Hern, Numerical relativity and inhomogeneous cosmologies, Ph.D. Thesis, Department of Applied Mathematics and Theoretical Physics, Cambridge University, Cambridge, UK, 2000. Available from <http://arxiv.org/abs/gr-qc/0001070>.
- [13] M.J. Berger, J. Olinger, Adaptive mesh refinement for hyperbolic partial differential equations, *J. Comput. Phys.* 53 (1984) 484.
- [14] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, *J. Comput. Phys.* 82 (1989) 64.
- [15] M.J. Berger, On conservation at grid interfaces, *SIAM J. Numer. Anal.* 24 (1987) 967.
- [16] G. Chesshire, W.D. Henshaw, *J. Comput. Phys.* 90 (1990) 1–64.
- [17] W.D. Henshaw, D.W. Schwendeman, *J. Comput. Phys.* 191 (2003) 420–447.
- [18] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, *Numerical Recipes*, second ed., Cambridge University Press, New York, 1994.
- [19] S. Teukolsky, On the stability of the iterated Crank–Nicholson method in numerical relativity, *Phys. Rev. D* 61 (2000) 087501.
- [20] P. MacNeice, K. Olson, C. Mobarry, R. de Fainchtein, C. Packer, PARAMESH: a parallel adaptive mesh refinement community toolkit, *Comput. Phys. Commun.* 126 (2000) 330.
- [21] D. DeZeeuw, K. Powell, An adaptively refined cartesian mesh solver for the Euler equations, *J. Comput. Phys.* 104 (1993) 56.
- [22] B. Schutz, *A First Course in General Relativity*, Cambridge University Press, New York, 1985.
- [23] D. Martin, K. Cartwright, Solving Poisson's Equation using Adaptive Mesh Refinement, unpublished notes (1996).
- [24] S. Teukolsky, Linearized quadrupole waves in general relativity and the motion of test particles equation using adaptive mesh refinement, *Phys. Rev. D* 26 (1982) 745.
- [25] F. Olsson, N. Anders Petersson, Stability of interpolation on overlapping grids, *Comput. Fluids* 25 (1996) 583.