Lessons for adaptive mesh refinement in numerical relativity

To cite this article: Miren Radia *et al* 2022 *Class. Quantum Grav.* **39** 135006

OPEN ACCESS
IOP Publishing

Classical and Quantum Gravity

Class. Quantum Grav. **39** (2022) 135006 (43pp)

https://doi.org/10.1088/1361-6382/ac6fa9

Lessons for adaptive mesh refinement in numerical relativity

Miren Radia[1,*], Ulrich Sperhake[1,2,3], Amelia Drew[1], Katy Clough[4], Pau Figueras[4], Eugene A Lim[5], Justin L Ripley[1], Josu C Aurrekoetxea[6], Tiago França[4] and Thomas Helfer[2]

[1] Department of Applied Mathematics and Theoretical Physics, Center for Mathematical Sciences, University of Cambridge, Wilberforce Road, Cambridge CB3 0WA, United Kingdom
[2] Department of Physics and Astronomy, Johns Hopkins University, 3400 N. Charles Street, Baltimore, MD 21218, United States of America
[3] California Institute of Technology, Pasadena, CA 91125, United States of America
[4] Geometry, Analysis and Gravitation, School of Mathematical Sciences, Queen Mary University of London, Mile End Road, London E1 4NS, United Kingdom
[5] Theoretical Particle Physics and Cosmology Group, Physics Department, Kings College London, Strand, London WC2R 2LS, United Kingdom
[6] Astrophysics, Denys Wilkinson Building, University of Oxford, Keble Road, Oxford OX1 3RH, United Kingdom

E-mail: m.r.radia@damtp.cam.ac.uk, u.sperhake@damtp.cam.ac.uk, a.drew@damtp.cam.ac.uk, k.clough@qmul.ac.uk, p.figueras@qmul.ac.uk, eugene.a.lim@gmail.com, lloydripley@gmail.com, josu.aurrekoetxea@physics.ox.ac.uk, t.e.franca@qmul.ac.uk and thomashelfer@live.de

Received 19 January 2022, revised 3 May 2022
Accepted for publication 13 May 2022
Published 6 June 2022

CrossMark

Abstract

We demonstrate the flexibility and utility of the Berger–Rigoutsos adaptive mesh refinement (AMR) algorithm used in the open-source numerical relativity (NR) code GRChombo for generating gravitational waveforms from binary black-hole (BH) inspirals, and for studying other problems involving non-trivial matter configurations. We show that GRChombo can produce high quality binary BH waveforms through a code comparison with the established NR code Lean. We also discuss some of the technical challenges involved in making use

[*] Author to whom any correspondence should be addressed.

1361-6382/22/135006+43$33.00 © 2022 The Author(s). Published by IOP Publishing Ltd Printed in the UK

of full AMR (as opposed to, e.g. moving box mesh refinement), including the numerical effects caused by using various refinement criteria when regridding. We suggest several 'rules of thumb' for when to use different tagging criteria for simulating a variety of physical phenomena. We demonstrate the use of these different criteria through example evolutions of a scalar field theory. Finally, we also review the current status and general capabilities of GRChombo.

Keywords: numerical relativity, adaptive mesh refinement, gravitational waves, compact objects, computational methods

(Some figures may appear in colour only in the online journal)

1. Introduction

One of the key theoretical achievements that underpins the momentous detection of gravitational waves (GWs) from the inspiralling black-hole (BH) binary GW150914 [1] is the numerical relativity (NR) breakthrough in binary BH modeling in 2005 [2–4]. About 90 compact binary merger events have by now been observed by the GW detector network [5–7] with a wide range of total masses and mass ratios of the binary constituents. Indeed, it is a generic feature of general relativity and its character as a highly non-linear theory that its solutions often span across a large range of spatial and temporal scales. Combined with the inherent limits of computational resources, it follows that any finite difference numerical code will require some form of spatial and temporal mesh refinement to fully capture the dynamics of these solutions.

In the NR community, many codes rely on the technique of so-called 'moving boxes' for mesh refinement, where a hierarchy of nested boxes with increasingly fine meshes is centered around specified points (also sometimes referred to as the 'box-in-box' approach). Within this framework, boxes move around either dynamically or along predetermined paths, in order to track objects' trajectories[7]. This technique has proved remarkably successful, particularly in the case of generating gravitational waveforms from binaries of compact objects for the template banks for GW detectors, such as LIGO–Virgo–KAGRA [21–25]. Moving box codes have matured to allow exploration of a wide variety of physics with a plethora of diagnostic tools [26, 27].

However, there are classes of problems for which the moving boxes technique becomes impractical due to the topology of the system. Here, the use of 'fully adaptive' mesh refinement (AMR) is required where the mesh dynamically adjusts itself in response to the underlying physical system being simulated, following user-specified mesh refinement criteria. In general, there are two broad classes of AMR, depending on whether newly refined meshes are added to the grid on a cell-by-cell 'tree-structured' basis [28–30] or on a box-by-box 'block-structured' basis. In this work, we will exclusively discuss the latter.

In block-structured AMR, first described and implemented by Berger and Oliger [31], the computational domain is built from a hierarchy of increasingly fine levels, with each one containing a set of (not necessarily contiguous) boxes of meshes, with the only condition being that a finer mesh must lie on top of one *or possibly more* meshes from the next coarsest level. It is important to stress that this means it is allowed for a fine mesh to straddle more than one coarse mesh—in other words the grid structure is *level*-centric rather than box-centric.

[7] In addition, boxes may be allowed to merge if they come close enough together. These codes include those built upon the popular Cactus computational framework [8–10] such as the McLachlan [11], LazEv [12], Maya [13], Lean [14, 15] and Canuda [16] codes, and the BAM code [17–20].

In contrast to the moving boxes approach, this approach allows for highly flexible 'many-boxes-in-many-boxes' mesh topologies, enabling the study of dynamical systems where the spacetime dynamics are not driven by localized compact systems e.g. in studying non spherical collapse scenarios [32, 33], higher dimensional BHs/black string evolution [34–39], cosmic string evolution [40–42], and the behaviour of strongly inhomogeneous cosmological spacetimes [43–51].

Despite its advantages, AMR is a double-edged sword and its flexibility comes with a cost—each coarse-fine transition may introduce unwanted interpolation and prolongation errors whose magnitude depends on the order of the coarse-fine boundary operators, in addition to introducing a 'hard surface' which can generate spurious unphysical reflections. We emphasize that AMR should not be treated as a 'black box', but requires careful control and fine-tuning of refinement criteria, that often depend on the physics being simulated, in order to work effectively. In particular, the creation/destruction of a finer grid is determined by the tagging of cells for refinement, which in turn is controlled by a *tagging criterion*. Although this ability to refine regions can be incredibly powerful, in practice it can be difficult to manage the exact placement of refined grids. Furthermore, we find that the management of coarse-fine boundaries in dynamically sensitive regions of spacetime, such as near apparent horizons, is essential for producing accurate results.

In this paper we explain some of the tagging criteria and numerical techniques we have used to obtain convergent, reliable results when using block-structured AMR. We will discuss these issues in the context of the AMR NR code GRChombo [52, 53][8], which was first introduced in 2015 and uses the Chombo [76] library. While our methods apply directly to GRChombo, we believe many of the lessons we have learned are general and may be useful to researchers who work with other NR codes that make use of block-structured AMR, in particular those which rely on the Berger–Rigoutsos [77] style grid generation methods.

We demonstrate the utility of the techniques we have employed through a direct comparison of gravitational waveforms generated by binary BH inspiral and merger calculated by GRChombo and the more established Lean code which uses the aforementioned 'moving boxes' style mesh refinement, and show that GRChombo is capable of achieving comparable production-level accuracy. We secondly apply AMR techniques to the evolution of several scalar field models which exhibit dynamics on a wide range of spatial and temporal scales, to demonstrate the relative advantages of several tagging criteria implemented in GRChombo.

This paper is organized as follows:

- In section 2 we detail the computational framework of GRChombo with a focus on the AMR aspects.
- In section 3 we discuss considerations for tagging criteria in AMR grid generation.
- In sections 4 and 5 we illustrate how these techniques are applied in practice to simulations of BH binaries and spacetimes with a (self-interacting) scalar field.

[8] AMR is now being used in several other NR codes. For example, those based on the PAMR/AMRD mesh refinement libraries [54–56], the Hahndol code [4], which uses PARAMESH [57], the HAD code [58], and the pseudospectral codes, SpEC [59], bamps [60] and SFINGE [61] in which the AMR implementation is somewhat different to finite difference codes like GRChombo. More recently, CosmoGRaPH [62], and Simflowny [63] both based on the SAMRAI library [64–66], GRAthena++ [67], Gmunu [68] and Dendro-GR [69], all based on oct-tree AMR, and GRAMSES [30] have been introduced. Alternatives are problem-adapted coordinate systems, e.g. NRPy+ [70] or discontinuous Galerkin methods as in SpECTRE [71, 72] (see also [73, 74]). Furthermore, it should be noted that some code frameworks, such as those based on the Carpet mesh refinement driver [10], are technically capable of performing block-structured AMR. However, it can be cumbersome to use and these codes typically rely on moving-box type methods (e.g. the codes referenced in footnote 7). A brief overview of the history of NR codes can be found in [75].

Our notation conventions are as follows. We use Greek letters $\mu, \nu, \ldots = 0, 1, 2, 3$ for spacetime indices and Latin letters $i, j, \ldots = 1, 2, 3$ for spatial indices. We use a mostly plus signature $(- + ++)$ and geometric units $c = G = 1$. When there is a potential for ambiguity between spacetime and purely spatial tensors (e.g. the Ricci scalar R), we prepend a $^{(4)}$ to denote the spacetime quantity. In the sections on BHs we set the mass scales with respect to the ADM mass of the spacetime, whereas for the section on scalar fields we set $\mu = mc/\hbar = 1$, which then describes lengths relative to the scalar Compton wavelength[9].

2. Computational framework

This section provides a comprehensive update of the methodology discussed in section 2 of [53]. See also appendix D for details on how GRChombo is parallelized.

2.1. Mathematical equations and notational conventions

2.1.1. Evolution equations and gauge conditions. GRChombo implements the CCZ4 formulation [78, 79] in order to evolve solutions of the Einstein equations, which we review below. We have found empirically with AMR that the inclusion of constraint damping terms in this formalism can be important to maintain accuracy; these mitigate the additional noise introduced by spurious reflections off the refinement boundaries due to the more complicated grid structures [80] (when compared with moving-box mesh refinement grids). The Z4 equation of motion with constraint damping and cosmological constant is [78][10]

$$\begin{aligned} &{}^{(4)}R_{\mu\nu} - \Lambda g_{\mu\nu} + 2\nabla_{(\mu}Z_{\nu)} - 2\kappa_1 n_{(\mu}Z_{\nu)} + \kappa_1(1+\kappa_2)g_{\mu\nu}n_\alpha Z^\alpha \\ &\quad = 8\pi\left(T_{\mu\nu} - \frac{1}{2}g_{\mu\nu}T\right), \end{aligned} \tag{1}$$

where ∇ is the Levi-Civita connection of the metric $g_{\mu\nu}$, ${}^{(4)}R_{\mu\nu}$ is the Ricci tensor of ∇, Λ is the cosmological constant, Z^μ is the Z4 vector which vanishes on physical solutions of the Einstein equation, n^μ is the future-directed unit normal to the foliation of spatial slices, κ_1 and κ_2 are constant damping parameters, $T_{\mu\nu}$ is the energy–momentum tensor, and $T = g^{\alpha\beta}T_{\alpha\beta}$ is its trace.

In the standard $3+1$ decomposition of spacetime [81, 82], the metric in adapted coordinates (t, x^i) takes the form

$$\mathrm{d}s^2 = -\alpha^2\,\mathrm{d}t^2 + \gamma_{ij}(\mathrm{d}x^i + \beta^i\,\mathrm{d}t)(\mathrm{d}x^j + \beta^j\,\mathrm{d}t), \tag{2}$$

where γ_{ij} is the spatial metric, $\alpha = 1/\|\mathrm{d}t\|$ is the *lapse* function and β^i is the *shift* vector. The future-directed unit normal to the foliation is

$$n_\mu := -\alpha(\mathrm{d}t)_\mu, \qquad n^\mu = \frac{1}{\alpha}\left(\partial_t^\mu - \beta^k\partial_k^\mu\right), \tag{3}$$

[9] If we are interpreting the results in terms of a physical particle mass, this is equivalent to setting the value of $\hbar$ in the code. Note that in general $\hbar \neq 1$ in NR simulations (as this would imply that one unit in the length scale is equal to the Planck length l_{Pl}).

[10] Note the sign difference between the unit normal in [78] and here. We choose the sign which ensures the unit normal is future-directed.

and the extrinsic curvature is

$$K_{\mu\nu} := -\frac{1}{2}(\mathcal{L}_n \gamma)_{\mu\nu}, \tag{4}$$

which in adapted coordinates (2) becomes

$$K_{ij} = -\frac{1}{2\alpha}\left[\partial_t \gamma_{ij} - \beta^m \partial_m \gamma_{ij} - 2\gamma_{m(i}\partial_{j)}\beta^m\right]. \tag{5}$$

In analogy to the spacetime metric, we decompose the energy–momentum tensor according to

$$\rho := n_\alpha n_\beta T^{\alpha\beta}, \qquad S_i := -\gamma_{i\alpha} n_\beta T^{\alpha\beta}, \qquad S_{ij} := \gamma_{i\alpha}\gamma_{j\beta} T^{\alpha\beta}, \qquad S := \gamma^{ij} S_{ij}. \tag{6}$$

We conformally rescale the spatial metric as[11]

$$\tilde{\gamma}_{ij} = \chi \gamma_{ij}, \qquad \tilde{\gamma}^{ij} = \frac{1}{\chi}\gamma^{ij}, \tag{7}$$

where $\chi = [\det(\gamma_{ij})]^{-1/3}$ so that the determinant of the conformal metric $\tilde{\gamma}_{ij}$ is unity. As in the Baumgarte–Shapiro–Shibata–Nakamura–Oohara–Kojima (BSSNOK) formulation, we introduce the conformally rescaled traceless extrinsic curvature,

$$\tilde{A}_{ij} := \chi\left(K_{ij} - \frac{1}{3}K\gamma_{ij}\right), \tag{8}$$

with $K = \gamma^{ij}K_{ij}$ and the *conformal connection functions*

$$\tilde{\Gamma}^i := \tilde{\gamma}^{jk}\tilde{\Gamma}^i_{jk} = -\partial_j \tilde{\gamma}^{ij}, \tag{9}$$

where $\tilde{\Gamma}^i_{jk}$ are the Christoffel symbols with respect to the conformal metric $\tilde{\gamma}_{ij}$. Finally, we decompose the Z4 vector by defining[12]

$$\Theta := -n_\alpha Z^\alpha, \qquad \Theta^i := \gamma^i_{\ \alpha} Z^\alpha. \tag{10}$$

Rather than evolving the Θ^i directly, we instead incorporate them into a set of *modified conformal connection functions*

$$\hat{\Gamma}^i := \tilde{\Gamma}^i + 2\tilde{\gamma}^{ik}\Theta_k = \tilde{\Gamma}^i + 2\frac{\Theta^i}{\chi}. \tag{11}$$

Finally our complete set of dynamical variables are

$$\left\{\chi, \tilde{\gamma}_{ij}, K, \tilde{A}_{ij}, \Theta, \hat{\Gamma}^i\right\}, \tag{12}$$

[11] This is the main difference with the previous GRChombo paper [53], where χ^2 was used in place of χ.

[12] The spatial projection is often denoted just Z^i in the literature e.g. [78], but we have changed the symbol in order to make clear that this is the *projected* three-vector as opposed to the spatial components of the Z4 vector.

and the CCZ4 evolution equations are[13]

$$\partial_t \chi = \beta^k \partial_k \chi + \frac{2}{3}\chi(\alpha K - \partial_k \beta^k), \tag{13}$$

$$\partial_t \tilde{\gamma}_{ij} = \beta^k \partial_k \tilde{\gamma}_{ij} + \tilde{\gamma}_{ki}\partial_j \beta^k + \tilde{\gamma}_{kj}\partial_i \beta^k - 2\alpha \tilde{A}_{ij} - \frac{2}{3}\tilde{\gamma}_{ij}\partial_k \beta^k, \tag{14}$$

$$\begin{aligned}\partial_t K = {} & \beta^k \partial_k K + \alpha\left(\hat{R} + K(K - 2\Theta)\right) - 3\alpha\kappa_1(1+\kappa_2)\Theta - \gamma^{kl}D_k D_l \alpha \\ & + 4\pi\alpha(S - 3\rho) - 3\alpha\Lambda,\end{aligned} \tag{15}$$

$$\begin{aligned}\partial_t \tilde{A}_{ij} = {} & \beta^k \partial_k \tilde{A}_{ij} + \chi\left[-D_i D_j \alpha + \alpha(\hat{R}_{ij} - 8\pi S_{ij})\right]^{\mathrm{TF}} \\ & + \tilde{A}_{ij}\left[\alpha(K - 2\Theta) - \frac{2}{3}\partial_k \beta^k\right] + 2\tilde{A}_{k(i}\partial_{j)}\beta^k - 2\alpha\tilde{\gamma}^{kl}\tilde{A}_{ik}\tilde{A}_{lj},\end{aligned} \tag{16}$$

$$\begin{aligned}\partial_t \Theta = {} & \beta^k \partial_k \Theta + \frac{1}{2}\alpha\left(\hat{R} - \tilde{A}_{kl}\tilde{A}^{kl} + \frac{2}{3}K^2 - 2\Theta K\right) \\ & - \alpha\kappa_1\Theta(2 + \kappa_2) - \Theta^k \partial_k \alpha - 8\pi\alpha\rho - \alpha\Lambda,\end{aligned} \tag{17}$$

$$\begin{aligned}\partial_t \hat{\Gamma}^i = {} & \beta^k \partial_k \hat{\Gamma}^i + \frac{2}{3}\left[\partial_k \beta^k\left(\tilde{\Gamma}^i + 2\kappa_3 \frac{\Theta^i}{\chi}\right) - 2\alpha K \frac{\Theta^i}{\chi}\right] - 2\alpha\kappa_1 \frac{\Theta^i}{\chi} \\ & + 2\tilde{\gamma}^{ik}(\alpha\partial_k \Theta - \Theta\partial_k \alpha) - 2\tilde{A}^{ik}\partial_k \alpha + 2\alpha\tilde{\Gamma}^i_{kl}\tilde{A}^{kl} \\ & - \alpha\left[\frac{4}{3}\tilde{\gamma}^{ik}\partial_k K + 3\tilde{A}^{ik}\frac{\partial_k \chi}{\chi}\right] - \left(\tilde{\Gamma}^k + 2\kappa_3\frac{\Theta^k}{\chi}\right)\partial_k \beta^i \\ & + \tilde{\gamma}^{kl}\partial_k \partial_l \beta^i + \frac{1}{3}\tilde{\gamma}^{ik}\partial_l \partial_k \beta^l - 16\pi\alpha\tilde{\gamma}^{ik}S_k,\end{aligned} \tag{18}$$

where D_i is the Levi-Civita connection on the spatial slice, $[\cdot]^{\mathrm{TF}}$ denotes the trace-free part of the expression in square brackets, $\hat{R}_{ij}$ is the *modified Ricci tensor*, given in terms of the normal Ricci tensor of D_i, R_{ij} by

$$\hat{R}_{ij} := R_{ij} + 2D_{(i}\Theta_{j)}, \tag{19}$$

and κ_3 is a third damping parameter. At each right-hand side (rhs) evaluation, we construct the quantity Θ^i/χ using the evolved $\hat{\Gamma}^i$ and $\tilde{\Gamma}^i$ computed from the metric and its derivatives in (11). The covariant second derivatives of the lapse are computed via

$$\gamma^{kl}D_k D_l \alpha = \tilde{\gamma}^{kl}\chi\partial_k \partial_l \alpha - \frac{1}{2}\tilde{\gamma}^{kl}\partial_k \alpha \partial_l \chi - \chi\tilde{\Gamma}^k \partial_k \alpha, \tag{20}$$

$$D_i D_j \alpha = \partial_i \partial_j \alpha - \tilde{\Gamma}^k_{ij}\partial_k \alpha + \frac{1}{2\chi}\left(\partial_i \alpha \partial_j \chi + \partial_i \chi \partial_j \alpha - \tilde{\gamma}_{ij}\tilde{\gamma}^{kl}\partial_k \alpha \partial_l \chi\right). \tag{21}$$

[13] Note that by default we make the modification $\alpha\kappa_1 \to \kappa_1$ mentioned in [79] that allows stable evolution of BHs with $\kappa_3 = 1$. However, equations (13)–(18) do not have this modification.

The modified Ricci tensor (19) is given by[14]

$$\hat{R}_{ij} = \tilde{R}_{ij} + \frac{1}{\chi}\left(R^{\chi}_{ij} + R^{Z}_{ij}\right), \tag{22}$$

with

$$\begin{aligned}\tilde{R}_{ij} = &-\frac{1}{2}\tilde{\gamma}^{kl}\partial_k\partial_l\tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i}\partial_{j)}\hat{\Gamma}^k + \frac{1}{2}\hat{\Gamma}^k\partial_k\tilde{\gamma}_{ij} \\ &+ \tilde{\gamma}^{lm}\left(\tilde{\Gamma}^k_{li}\tilde{\Gamma}_{jkm} + \tilde{\Gamma}^k_{lj}\tilde{\Gamma}_{ikm} + \tilde{\Gamma}^k_{im}\tilde{\Gamma}_{klj}\right),\end{aligned} \tag{23}$$

$$\begin{aligned}R^{\chi}_{ij} = &\frac{1}{2}\left[\tilde{D}_i\tilde{D}_j\chi + \tilde{\gamma}_{ij}\tilde{\gamma}^{kl}\tilde{D}_k\tilde{D}_l\chi\right] \\ &- \frac{1}{4\chi}\left[\partial_i\chi\partial_j\chi + 3\tilde{\gamma}_{ij}\tilde{\gamma}^{kl}\partial_k\chi\partial_l\chi\right],\end{aligned} \tag{24}$$

$$R^{Z}_{ij} = \frac{\Theta^k}{\chi}\left(\tilde{\gamma}_{ik}\partial_j\chi + \tilde{\gamma}_{jk}\partial_i\chi - \tilde{\gamma}_{ij}\partial_k\chi\right), \tag{25}$$

and $\tilde{\Gamma}_{ijk} = \tilde{\gamma}_{il}\tilde{\Gamma}^l_{jk}$. Finally, the *modified Ricci scalar* is given by the trace of the modified Ricci tensor:

$$\hat{R} = \chi\tilde{\gamma}^{kl}\hat{R}_{kl}. \tag{26}$$

By default, GRChombo uses the damping parameters[13]

$$\alpha\kappa_1 = 0.1, \qquad \kappa_2 = 0 \quad \text{and} \quad \kappa_3 = 1. \tag{27}$$

To close the system (13)–(18), we need to specify gauge conditions for the lapse and shift. Although the structure of the code allows easy modification of the gauge, by default we use a *Bona-Masso*-type slicing condition [83] of the form

$$\partial_t\alpha = a_1\beta^k\partial_k\alpha - a_2\alpha^{a_3}(K - 2\Theta), \tag{28}$$

where a_1, a_2 and a_3 are constant parameters. Note that this reduces to the familiar $1 + \log$ slicing in the case

$$a_1 = 1, \qquad a_2 = 2 \quad \text{and} \quad a_3 = 1, \tag{29}$$

which is the default. For the shift, we use the *Gamma-driver* shift condition [3, 4] in the form:

$$\partial_t\beta^i = b_1\beta^k\partial_k\beta^i + b_2B^i, \tag{30}$$

$$\partial_tB^i = b_1(\beta^k\partial_kB^i - \beta^k\partial_k\hat{\Gamma}^i) + \partial_t\hat{\Gamma}^i - \eta B^i, \tag{31}$$

where b_1, b_2 and η are constant parameters. By default we take $b_1 = 0$ and $b_2 = 3/4$, whereas the value of η depends on the specific configuration considered (typically $\mathcal{O}(1/M_{\text{ADM}})$). Together, these gauge conditions are commonly referred to as the *moving puncture gauge*.

We enforce the tracelessness of $\tilde{A}_{ij}$ (8) before every rhs evaluation and at the end of each full timestep. Furthermore, we also enforce the condition $\chi \geqslant \chi_{\min}$ and $\alpha \geqslant \alpha_{\min}$ before

[14] Note the somewhat unconventional factor of χ^{-1} multiplying R^{χ}_{ij}.

every rhs evaluation and at the end of each full timestep, where the default values are $\chi_{\min} = 10^{-4} = \alpha_{\min}$, in order to ensure these variables do not become arbitrarily small or negative due to discretization error[15].

As is common when using the moving puncture gauge with BH spacetimes, GRChombo can track the position of the puncture(s) $\mathbf{x}_{\mathrm{p}}$ by solving [26]

$$\frac{\mathrm{d}x^i_{\mathrm{p}}}{\mathrm{d}t} = -\beta^i(\mathbf{x}_{\mathrm{p}}), \tag{32}$$

which is integrated using the trapezium rule at user-specified time intervals.

During evolutions, we monitor the violations of the Hamiltonian and momentum constraints which are given by

$$\mathcal{H} := R + \frac{2}{3}K^2 - \tilde{A}_{kl}\tilde{A}^{kl} - 2\Lambda - 16\pi\rho, \tag{33}$$

$$\mathcal{M}_i := \tilde{\gamma}^{kl}\left(\partial_k\tilde{A}_{li} - 2\tilde{\Gamma}^m_{l(i}\tilde{A}_{k)m} - 3\tilde{A}_{ik}\frac{\partial_l\chi}{2\chi}\right) - \frac{2}{3}\partial_i K - 8\pi S_i. \tag{34}$$

In the continuum limit these two quantities should vanish. From the Bianchi identities, if the constraints vanish on the initial data surface, they vanish throughout the spacetime [84]. However, given errors due to our discretization of the equations of motion, we expect there to be small violations of the constraints in our numerical simulations. We often monitor the constraint violation to assess the accuracy of a given run.

In the case of non-vacuum spacetimes, one can normalize the constraints with some measure (e.g. the maximum) of their matter sources, i.e. $\mathcal{H}$ with ρ and $\mathcal{M}_i$ with S_i.

2.1.2. Initial data. For BH binaries, we have integrated the TwoPunctures spectral solver [85] into GRChombo. This provides binary puncture data [86] of Bowen–York type [87]; the version we use also incorporates the improvements described in [88] that allows for fast spectral interpolation of the pseudospectral solution onto, e.g. a Cartesian grid. Besides TwoPunctures, GRChombo includes a class for non-spinning binary Bowen–York data with an approximate solution of the Hamiltonian constraint for the conformal factor [89], which is valid in the limit of small boosts $|\mathbf{P}| \ll M$, where $\mathbf{P}$ is the initial momentum of an individual BH and M is some measure of the total mass.

We also provide initial data for Kerr BHs using the formulation in [90], which allows for the evolution of near-extremal BHs within the moving puncture approach to BH evolution.

For matter spacetimes, one must in general solve the Hamiltonian and momentum constraints numerically to obtain valid initial conditions for the metric that correspond to the energy and momentum distributions which are chosen (including BHs using the methods above, where required). For time-symmetric spherically symmetric initial data this can be done using shooting methods, as for the axion star data used in section 5 and previous works [91–98]. In inhomogeneous cosmological spacetimes, numerical solutions for an initial matter configuration should be obtained using relaxation or multigrid methods, as in[16].

[15] Typically any values that are affected by this procedure lie behind a horizon so are causally disconnected from most of the computational domain. In cosmological simulations the conformal factor is directly related to the scale factor and thus in rapidly expanding or contracting spacetimes this limit may need to be adjusted.

[16] Most of these works assume an initial time symmetry in order to avoid solving the momentum constraints. A fully general multigrid initial condition solver for GRChombo that solves the coupled constraint system for any scalar field distribution with and without BHs is currently under development.

2.2. Code features

2.2.1. Discretization and time-stepping. GRChombo evolves the CCZ4 equations using the method of lines with a standard fourth-order Runge–Kutta method (RK4). For the spatial discretization, we typically use either fourth or sixth order centered stencils except for the advection terms, for which we switch to lopsided stencils of the same order depending on the sign of the shift vector component. For completeness, the expressions for the stencils are provided in appendix B.

Finite difference methods can often introduce spurious high-frequency modes, particularly when using AMR and regridding. To ameliorate this, GRChombo uses $N = 3$ *Kreiss–Oliger* (KO) dissipation [99]; at every evaluation of the rhs for an evolution variable F, we add the term

$$\frac{\sigma}{64\Delta x}\left(F_{i-3} - 6F_{i-2} + 15F_{i-1} - 20F_i + 15F_{i+1} - 6F_{i+2} + F_{i+3}\right) \tag{35}$$

to the rhs, where Δx is the relevant grid spacing (cf (37)). A von Neumann stability analysis [26] shows that this scheme, when applied to the trivial Partial Differential Equation (PDE) $\partial_t F = 0$, is linearly stable only if

$$0 \leqslant \sigma \leqslant \frac{2}{\alpha_C}, \tag{36}$$

where $\alpha_C = \Delta t/\Delta x$ is the *Courant–Friedrichs–Lewy* factor, which we typically set to $1/4$, and Δt is the size of the timestep[17]. Note that we always use $N = 3$ KO dissipation, independent of the order of spatial discretization. Naively, one might question this choice for sixth order spatial derivative stencils as the conventional wisdom is to pick N such that $2N - 1 > m$, where m is the order of the finite difference scheme (see, for example, [26]). However, in this case, what matters is the order of the time stepping which is still fourth order[18], hence the dissipation operator does not 'spoil' the convergence properties of the scheme. This is consistent with the approach discussed in section 3.2 of [100].

2.2.2. Berger–Rigoutsos AMR. In GRChombo, the grid comprises a hierarchy of cell-centered Cartesian meshes consisting of up to $l_{\max} + 1$ refinement levels labeled[19] $l = 0, \ldots, l_{\max}$ each with grid spacing

$$\Delta x_l = 2^{l_{\max}-l}\Delta x_{l_{\max}} = \Delta x_0/2^l. \tag{37}$$

GRChombo uses block-structured AMR, so each refinement level is split into boxes which are distributed between the CPUs as described in appendix D.

At regridding or initial grid creation, on a given refinement level l, cells are tagged for refinement according to a *tagging criterion* $C = C(\mathbf{i})$. In a given cell with indices $\mathbf{i} = (i, j, k)$ and corresponding Cartesian coordinates[20] $\mathbf{x} = (x_i, y_j, z_k)$, if $C(\mathbf{i}) > \tau_R$, where τ_R is a pre-specified threshold value (which may vary with l), then the cell is tagged for refinement.

[17] Note that the stability analysis makes a number of assumptions and problems can begin to appear towards the upper end of the range (36). We have observed that a typical symptom of an instability due to too large σ is the appearance of a checkerboard-like pattern in otherwise spatially homogeneous regions of the spacetime.

[18] Also note that theorem 9.1 in [99] only refers to the order of the time stepping.

[19] Note that the finest level that exists may be less than $l_{\max}$.

[20] Note that the indices here are of the discrete cells on the grid as opposed to spacetime components.

We discuss techniques to design a suitable tagging criterion and aspects to consider in section 3.

In block-structured AMR, the main challenge after tagging cells is finding an efficient algorithm to *partition* the cells that require refinement into blocks or boxes. GRChombo uses Chombo's implementation of the Berger–Rigoutsos grid generation algorithm [77] in order to do this.

For this purpose, we define the *block factor* as the number of cells that must divide the side lengths of all blocks; it is a specifiable parameter. Furthermore these side lengths must not exceed the specified maximum box size. In order to enforce the block factor on level $l+1$, starting with the tagged cells on level l, Chombo generates a temporary new set of tagged cells on a virtual coarser level $l-n$ where n is chosen such that the length of one cell on level $l-n$ corresponds to the block factor on level $l+1$.[21] The new set of coarser tags are derived using a global OR operation, i.e. as long as any of the l level cells corresponding to the coarser level cell is tagged, the virtual coarser level cell will be tagged. Chombo then applies the Berger–Rigoutsos partitioning algorithm on this coarser level to construct boxes of grids which obey both the desired block factor and maximum box size. We typically choose both to be a multiple of the processor vector width for optimal performance.

For completeness, we next review the Berger–Rigoutsos algorithm (see also figure 1). We find the minimum box that encloses all of the tagged cells on this level. Let $T(\mathbf{i})$ be the tagging indicator function defined by

$$T(\mathbf{i}) = \begin{cases} 1, & \text{if } C(\mathbf{i}) > \tau_{\mathrm{R}}, \\ 0, & \text{otherwise} \end{cases} \tag{38}$$

and define the *signatures* or traces of the tagging by

$$\mathcal{S}_x(i) := \sum_{j,k} T(\mathbf{i}) = \int T(\mathbf{i})\,\mathrm{d}y\,\mathrm{d}z, \tag{39}$$

$$\mathcal{S}_y(j) := \sum_{i,k} T(\mathbf{i}) = \int T(\mathbf{i})\,\mathrm{d}x\,\mathrm{d}z, \tag{40}$$

$$\mathcal{S}_z(k) := \sum_{i,j} T(\mathbf{i}) = \int T(\mathbf{i})\,\mathrm{d}x\,\mathrm{d}y. \tag{41}$$

First, we look for 'holes' in the signatures i.e. if there exist i, j or k for which $\mathcal{S}_x(i)$, $\mathcal{S}_y(j)$ or $\mathcal{S}_z(k)$ vanish which corresponds to there being no tagged cells along the plane orthogonal to the signature direction. If there are holes, we choose the one with largest index over all the dimensions (since it is more efficient to have fewer big boxes than more small boxes) as the plane of partition. If there are no holes, we next look for inflections (see [77] and their discussion of figure 10 for details) in the signatures by computing their discrete Laplacian, for example,

$$\Delta\mathcal{S}_x(i) = \mathcal{S}_x(i-1) - 2\mathcal{S}_x(i) + \mathcal{S}_x(i+1), \tag{42}$$

[21] For example if the block factor is 4, then $n=1$ since the refinement ratio is 2 and $2^{(l+1)-(l-1)} = 4$. Note that this means the block factor must be a power of 2.

and searching for zero-crossings in $\Delta\mathcal{S}_x(i)$. Heuristically, this corresponds to a rough boundary between tagged and untagged regions; cf the partitioning in step (2) of figure 1. If there are inflections, then, in each direction, we pick the inflection with the greatest difference, for example,

$$|\delta(\Delta\mathcal{S}_x(i))| = |\Delta\mathcal{S}_x(i-1) - \Delta\mathcal{S}_x(i)|. \tag{43}$$

As for the holes, we then pick the greatest inflection index over all the dimensions as our plane of partition. If there are no holes or inflections in the signatures, we simply split the box along the midpoint of the direction with the longest side.

After partitioning, we check whether the partition is sufficiently *efficient*, specifically whether the proportion of tagged cells to all cells in the partition exceeds a user-specified *fill ratio* threshold, $\epsilon_{\mathrm{FR}} < 1$ and that the lengths of the boxes are at most the pre-specified maximum box size (which we choose in order to ensure sufficient load balancing). If these tests are passed then we accept the partition and, if not, we continue to partition recursively discarding any boxes that do not contain tagged cells.

Note that, whilst a higher value of ϵ_{FR} will result in a more efficient partition in the sense that there will be a greater ratio of tagged to untagged cells, this is not always the most computationally efficient choice as there are greater overheads with smaller boxes (for example, there will be more ghost cells). There could also be more fluctuation in the structure of the grids between consecutive regrids which may result in greater noise. Although the optimal fill ratio depends on the particular physical problem and the computational resources, we typically use $\epsilon_{\mathrm{FR}} = 0.7$.

Finally the boxes in the partition are refined, that is, they are defined on the next finer level $(l+1)$ with twice the resolution (37). For newly created regions on this finer level, we interpolate the data from the coarser level using fourth-order interpolation.

The regridding process starts on the finest level that currently exist (or at most level $l_{\max} - 1$) and works up the hierarchy on increasingly coarse levels until the base level, from whose timestep the regrid was called, is reached (which need not be $l = 0$). It is therefore only possible to add a single extra level (up to $l_{\max}$) at each regrid. After the regrid on level l, the union of the set of cells in the new boxes on this level (plus an additional pre-specified *buffer region*) with the set of cells tagged on level $l-1$ is used as the final set of tagged cells on level $l-1$ in order to ensure *proper nesting*[22] [101]. This also ensures that cells on coarser levels will be tagged if any of their corresponding finer level cells are tagged.

The frequency of regridding is user-specifiable on each refinement level $l < l_{\max}$, though, since a regrid on level $l = l'$ enforces a regrid on all levels $l' \leqslant l < l_{\max}$, for problems without rapidly varying (in time) length scales, it is usually sufficient to regrid every few timesteps on one of the more coarse levels (e.g. for compact object binaries). Not only does reducing the frequency of regridding reduce the computational cost, but since regridding introduces errors/noise due to interpolation, we have also found that this can improve the accuracy of the simulation.

[22] By proper nesting we mean that

(a) The physical region corresponding to a level $l-1$ cell must be fully refined or not refined at all, that is it must be completely covered by level l cells and not partially refined.

(b) There must be at least one level l cell between the boundary of $l+1$ and the boundary of level l except at the boundary of the entire computational domain. In practice we even need two such buffer cells (corresponding to four cells on level $l+1$) for fourth and sixth-order spatial stencils.

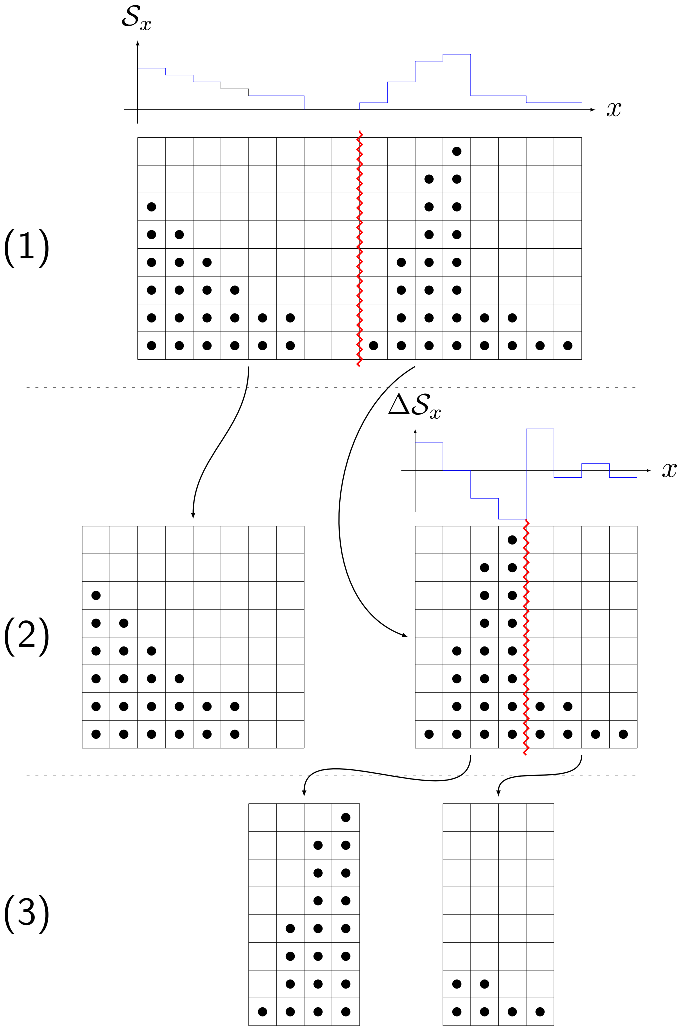


Figure 1. Schematic illustration of the partitioning algorithm. For simplicity, we show a 2D grid and only consider partitioning in the *x* direction. The cells tagged for refinement are indicated with •. In (1), the signature $\mathcal{S}_x$ is computed and two 'holes' are found where the signature vanishes. The line (plane in 3D) of partition is then at the hole with the largest index (rightmost). The result of the partitioning is shown in (2). To partition the right box in (2), the signature is computed but this time there are no holes so the algorithm looks for zero crossings of the discrete Laplacian of the signature $\Delta\mathcal{S}_x$. The zero-crossing with largest change is then selected. This algorithm terminates once all boxes have reached the required fill ratio ϵ_{FR}.

The Courant condition limits the size of the maximum time steps one can take on the finer levels. Rather than evolving all refinement levels with the same timestep, we use *subcycling* by following the Berger–Colella evolution algorithm [101], which we now review. As the algorithm is recursive, we can consider evolving a set of coarser and finer grids at level l and $l + 1$ respectively in the AMR grid hierarchy. First, one time step is taken on the coarser grids (i.e. those at level l). One then evolves the finer (level $l + 1$) grids for as many time steps until they have advanced to the same time as the coarse grid. As we have hard-coded the time steps on each level, Δt_l, so that $\Delta t_l = \Delta t_{l-1}/2$, the grids on level $l+1$ will then take two time steps after the grids on level l take one time step. After level $l+1$ has 'caught up' with level l, the mean of the data in the $[2^3 = 8]$ cells covering a single level l cell is calculated and this value is copied back onto level l. Note that this particular procedure is only second-order accurate in contrast to the restriction operation in a vertex-centered code which requires no approximation. This may partially explain some of the difference in convergence orders we observed between GRChombo and the vertex-centered code Lean in section 4.

The ghost cells at the interface between the finer and coarser grids are set by interpolating the values of the coarser grid in both space (due to the cell centered grid) and time (due to the requirement for intermediate values in the RK4 timestepping). The time interpolation is achieved by fitting the coefficients of a third order polynomial in t using the values obtained at each substep of the RK4 timestepping on the relevant cells of the coarser level (see [102] for more detail)[23].

2.2.3. Boundary conditions. GRChombo implements several classes of boundary conditions, including:

- Periodic—evolution variables φ obey $\varphi(x+L) = \varphi(x)$ in some or all Cartesian directions.
- Static—boundary values are fixed at their initial values.
- Reflective—one uses the symmetry of the spacetime to reduce the volume evolved. For example, in a simple equal mass head-on BH merger, one needs only 1/8 of the domain; the rest can be inferred from the evolved values [103]. Note that each evolution variable has a different parity across each reflective boundary.
- Extrapolating—both zeroth and first order schemes (by radial distance from a central point). These are especially useful for variables which asymptote to a spatially uniform but time varying value (see [104–106]).
- Sommerfeld/radiative—these permit (massless) outgoing radiation to leave the grid without reflections by assuming a solution of the form $\varphi = \varphi_0 + u(r-t)/r$ at the boundaries, where u can be any arbitrary function and φ_0 is a constant asymptotic value (see section 5.9 of [26]).

2.2.4. Interpolation and wave extraction. GRChombo features an AMR interpolator which allows the user to interpolate any grid variable, its first derivatives and any second derivative at an arbitrary point within the computational domain. The AMR interpolator starts searching for the requested points on the finest available level and then progresses down the hierarchy to increasingly coarse levels until all requested points are found in order to ensure the most accurate result. It supports interpolation via an arbitrary interpolation algorithm. The provided algorithms include Lagrange polynomial interpolation up to arbitrary order (although this is limited by the number of available ghost cells), using the algorithm of [107] to calculate the stencils on the fly and then memorizing them in order to increase efficiency, and 'nearest neighbour' interpolation. By default, and in the remainder of this paper, we use fourth order Lagrange polynomial interpolation.

GRChombo includes tools for the extraction of data over an arbitrary user-defined 2D surface, built on top of the AMR interpolator, with spherical and cylindrical surfaces being

[23] An alternative approach would be the use of larger ghost zones in the finer level, with the outer ones discarded at each RK step (for example, see section 2.3 of [10]). One disadvantage here is the extra memory use, especially beyond the fixed-box-hierarchy case.

implemented as examples. There is also a user-friendly interface for integrating arbitrary functions of the extracted grid variables and their derivatives over the surface using either the trapezium rule, Simpson's rule, Boole's rule or the midpoint rule in each surface coordinate direction.

Often the most important outputs of a NR simulation are the calculated GWs. To that end, GRChombo uses the Newman–Penrose formalism [108]. We describe the formulae for calculating the relevant Weyl scalar Ψ_4 (including terms arising from the Z4 vector) in appendix A. We use the extraction routines described above to interpolate the values of Ψ_4 on multiple spheres of fixed coordinate radius and then determine the modes ψ_{lm} with respect to the spin-weight -2 spherical harmonics $_{-2}Y^{lm}$ using the formula

$$r_{\mathrm{ex}}\psi_{lm}= \oint_{S^2} r_{\mathrm{ex}}\Psi_4|_{r=r_{\mathrm{ex}}} \left[{}_{-2}\bar{Y}^{lm}\right] \mathrm{d}\Omega, \tag{44}$$

where $\mathrm{d}\Omega = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$ is the area element on the unit sphere S^2. We use the trapezium rule for the integration over ϕ (since the periodicity means that any quadrature converges exponentially [109]) and Simpson's rule for the integration over θ.

3. Considerations for tagging criteria used for grid generation

We have found that the choice of tagging criteria can greatly impact the stability and accuracy of a given simulation. Here we mention several factors to consider when designing tagging criteria for use in GRChombo and other codes with similar AMR algorithms. We also provide some explicit examples of tagging criteria and discuss their relative merits.

3.1. Buffer regions

One of the problems of many tagging criteria we have tried is that they can often introduce several refinement levels over a relatively small distance in space. This leads to the boundaries of these refinement levels being particularly close to one another. Due to the errors introduced by interpolation at these boundaries, they can add spurious reflections or noise. This is exacerbated when other refinement boundaries are nearby, allowing for this noise to be repeatedly reflected and even amplified before it has time to dissipate (e.g. via KO dissipation—see section 2.2.1). A particularly simple way to mitigate this problem is to increase the *buffer regions*, i.e. the number of cells n_{B} between refinement levels. Since the regridding algorithm starts at the finest level and works up the hierarchy to coarser levels (see section 2.2.2), increasing this parameter actually increases the size of the coarser levels rather than shrinking the finer levels in order to enforce this buffer region restriction.

3.2. Considerations for black-hole spacetimes

Here we describe several techniques that we have used when creating tagging criteria to evolve BH spacetimes. The major complication with evolving BHs is that they have an event horizon. In practice, it is often challenging to find the true event horizon, which would require tracing geodesics through the full evolution of the spacetime. Therefore NR simulations typically consider the location of the apparent horizon instead, which always lies inside the event horizon [110]. Mathematically, the region within an apparent horizon is causally disconnected from its exterior. For a given numerical approximation, however, artifacts from the discretization can propagate from behind the horizon and contaminate the rest of the computational domain.

As a consequence of this superluminal propagation of numerical noise, we often find that GRChombo simulations of BHs are particularly sensitive to the presence of refinement boundaries. One should avoid adding refinement within the horizon (which in any case is unobservable and not usually of interest), but problems are particularly severe where a refinement boundary intersects the apparent horizon. In such cases we have observed significant phase inaccuracies and drifts in the horizon area (some even violating the second law of BH mechanics). Similar problems may occur if a refinement boundary is close to but does not intersect the horizon. In order to avoid these issues, we typically enforce the tagging of all cells within the horizon plus a buffer radius up to a maximum level $l_{\mathrm{BH}}^{\max}$ (which need not necessarily be $l_{\max}$ and may differ for each BH in the simulation). If r_{p} is the coordinate distance from the puncture of a BH of mass M_{BH} in a spacetime with total mass $M \sim 2M_{\mathrm{BH}}$, then, for $\eta \sim 1/M$ in the moving puncture gauge (28)–(31), after the initial gauge adjustment the apparent horizon is at approximately $r_{\mathrm{p}} = M_{\mathrm{BH}}$ (see figure 4 in [18]). Guided by this approximation, we can tag all cells with $r_{\mathrm{p}} < M_{\mathrm{BH}} + b$, where b is a pre-specified parameter. Although one might think choosing $b \propto M_{\mathrm{BH}}$ for each BH might be the most sensible choice for unequal mass configurations, we have found larger BHs less sensitive to the presence of refinement boundaries. Thus, choosing $b \propto M$ the same for each BH in a binary usually works sufficiently well.

Increasing the size of the buffer regions between refinement boundaries by adjusting n_{B} as discussed in section 3.1 can help to keep refinement boundaries sufficiently spaced apart. However, we have also separately enforced the spacing out of refinement boundaries by doubling the radius of the second and third finest levels covering a BH. This leads to tagging cells on level l (to be refined on level $l + 1$) with

$$r_{\mathrm{p}} < (M_{\mathrm{BH}} + b)2^{\min\,(l_{\mathrm{BH}}^{\max} - l - 1, 2)}. \tag{45}$$

In spacetimes where BH horizons are dynamical (often the target of AMR simulations), one can in principle use the locations of apparent horizons to define tagged regions. However, rather than incorporating the output of a horizon finder into the tagging criterion, a simpler and in most cases equally effective method can be obtained from using contours of the conformal factor χ and tagging regions with $\chi < \chi_0$, where χ_0 is a prespecified threshold value which may vary on each refinement level. This gives a reasonably robust and general method of identifying the approximate locations of horizons. Further details on precise values and their dependence on the BH spin are given in appendix C.

3.3. Asymmetric grids

The grid-generation algorithm (section 2.2.2) is inherently asymmetric, for example, it picks the 'hole' with *largest* index as the partition plane. This means that even if the tagging has symmetries, the grids themselves may not obey the same or any symmetries. For example, whilst one might expect that, for tagging cells with (45), the grids would have reflective symmetry in all three coordinate directions, this is often not the case, particularly for larger ϵ_{FR}. This asymmetry can lead to undesirable behaviour. For example, when simulating the head-on collision of two BHs with no symmetry assumptions (as described in section 2.2.3) with the tagging of (45), the punctures can deviate slightly from the collision axis. We can 'fix' this asymmetry by replacing

$$r_{\mathrm{p}} \to \varrho = \max(|x - x_{\mathrm{p}}|, |y - y_{\mathrm{p}}|, |z - z_{\mathrm{p}}|) \tag{46}$$

in (45) so that the tagged regions are boxes rather than spheres (this tagging is similar to what is done in some moving-box style mesh refinement codes). Whilst there is inevitably a loss of efficiency from this choice, this is typically outweighed by the reductions in error achieved.

Clearly, this approach pushes the AMR method in the direction of a moving boxes approach; in practice, we therefore apply it predominantly to BH simulations but not for more complex matter structures that require the full flexibility of AMR.

3.4. Using truncation error for tagging cells

Truncation error tagging was introduced by Berger and Oliger [31]. We have implemented truncation error tagging in GRChombo by using a *shadow hierarchy* (e.g. [111]). In this scheme, we estimate the truncation error on a grid at level l by comparing the solution of a specially chosen variable f on that level to the coarser level directly 'beneath' it on the grid:

$$\tau_{l,f}(\mathbf{i}) = \left| f_l(\mathbf{i}) - f_{l-1}(\mathbf{i}) \right| . \tag{47}$$

We note that the error (47) clearly must be computed before we average the finer grid values onto the coarser grid. As Chombo uses a cell-centered scheme, in order to compare the values of f on the two levels, we interpolate f from the coarser level onto the finer level using fourth order interpolation. If we compute the truncation error of multiple grid variables, we combine the error estimates for each variable at each point:

$$\tau_l(\mathbf{i}) = \sqrt{\sum_f \frac{\left(\tau_{l,f}(\mathbf{i})\right)^2}{L_f}}, \tag{48}$$

where L_f is a normalizing factor for each variable f. We then set this as our tagging criterion in (38): $C(\mathbf{i}) = \tau_l(\mathbf{i})$. The free parameters in this scheme of tagging are the choice of grid variables that one computes truncation error estimates for and the normalization factors for each variable.

The main advantages of truncation error tagging are that it allows for a conceptually straightforward way to implement convergence tests in AMR codes: as one increases the base grid resolution, one should scale the truncation error tagging threshold for grid generation with the expected convergence of the code. Additionally, truncation error tagging is a 'natural' tagging criterion as it refines regions that are most likely to be under resolved.

3.5. Tagging criteria based on grid variables and derived quantities

Some physical problems lend themselves to other tagging criteria, and GRChombo permits the user to easily specify refinement criteria based on any properties of the local grid variables or derived expressions of them, for example, derivatives or curvature scalars. We caution though that the tagging criteria we discuss below are not functions of geometric scalars, so the performance of a given criterion will depend on the formulation and gauge conditions used. Nevertheless, for the Bona-Masso-type slicing ((28)) and gamma-driver ((30) and (31)) conditions we have the most experience with, these gauge conditions have proven to be reliable and robust.

First we discuss tagging criteria based on the conformal factor of the spatial metric χ. Contours of χ can provide a good choice in dynamical BH cases as detailed in section 3.2 and appendix C, to ensure that horizons are covered. Taking differences of χ across a cell using locally evaluated derivatives, i.e. using $C = \sqrt{\delta^{ij}\partial_i\chi\partial_j\chi}\Delta x$, also provides an efficient measure to refine key areas[24]. In particular, using the second derivative of χ, i.e. $C = \sqrt{\delta^{ij}\partial_i\partial_j\chi}\Delta x$, is efficient because usually it is the regions in which gradients are changing most rapidly

[24] Imposing simply that $\partial_i\chi$ (without the factor of Δx) is higher that some threshold results in unlimited regridding, since one does not reduce the local gradient in a variable by refinement, only the difference across the cell.

that require greater resolution, rather than steep linear gradients. However, in practice any derivative can be used provided the thresholds are tuned appropriately for the problem at hand.

Alternatively, we find empirically that the sum of the absolute value of the different components of the Hamiltonian constraint proves to be an efficient tagging criteria in dynamical matter spacetimes. The condition is

$$C = \mathcal{H}_{\text{abs}} = |R| + |\tilde{A}^{kl}\tilde{A}_{kl}| + \frac{2}{3}K^2 + 16\pi|\rho| + 2|\Lambda|, \tag{49}$$

where $R = \gamma^{ij}R_{ij}$ is the Ricci scalar. As we will see in section 5.2, this quantity generally remains constant in regions of spacetime where the individual metric and matter components oscillate in a stable, time-invariant manner (as in the case of the stable axion star we present later). Thus using this measure reduces the amount of spurious regridding that occurs, which in turn reduces errors introduced by that process. Where it starts to grow in some region, this generally reflects a decrease in the local dynamical timescales and thus physically motivated regridding.

A disadvantage of using these more arbitrary criteria over error tagging is that convergence testing is more challenging—one must ensure that similar regions are refined at the appropriate resolutions in the convergence runs, which necessitates tuning of the threshold at each different base resolution. Depending on the regridding condition, halving the threshold τ_{R}, for example, may not result in double the resolution being applied. Nevertheless, if one ensures that the regions of most physical significance have an appropriate increase in refinement, convergence can usually be demonstrated, as we show below.

In order to obtain convergent results and use resolution most efficiently in a physical problem, it is often helpful to implement rules to enforce that given regions are refined for a given amount of time at least to a given level. Whilst this may seem to go against the spirit of AMR, it is easy to implement within that formalism as a secondary condition, and is often required to avoid excessive or insufficient tagging in very dynamical cases. For example, when one is not interested in resolving outgoing scalar radiation, one may choose to suppress regridding above a particular level outside of a particular radius. In the opposite sense, we often enforce extra regridding over the extraction surfaces for the Weyl scalars, to ensure that they have sufficient resolution and that noise is not introduced from grid boundaries crossing the spheres.

Several examples of the application of these criteria to BH binary inspirals and matter field evolutions are presented in the following, sections 4 and 5. These two examples cover the main considerations when using AMR in NR codes. Spacetimes with singularities have particular requirements related to the resolution of the horizons. Furthermore, in dynamical matter spacetimes achieving an optimum frequency of regridding can be crucial in obtaining convergence. In the matter case we focus on an isolated real scalar (axion) star, which provides a very good test of AMR capabilities. In particular, it tests the ability to resolve stably oscillating matter configurations without excessive gridding and ungridding, and to follow the dynamical timescales of gravitational collapse, which are the key challenges in many simulations of fundamental fields in NR, including the modeling of cosmic strings, inflationary spacetimes and exotic compact objects. Fully AMR techniques are also likely to create significant challenges for high-resolution shock capturing, but we leave this topic for future investigations.

Table 1. A list of the parameter values (cf figure 2) for the BH binary configurations simulated in this work. $M = M_1 + M_2$ denotes the total BH mass of the spacetime.

Label	q	d/M	P_t/M	P_r/M	χ_i	α	Reference
`q1-d12`	1	12.213 58	0.084 17	$5.108\,46 \times 10^{-4}$	0	—	[112]
`q2-d10`	2	10	0.085 66	0	0	—	[113]
`q1-s09`	1	11.017 68	0.075	0	0.9	30°	[114]

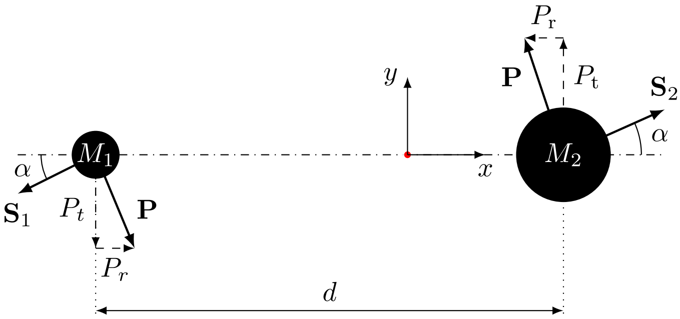


Figure 2. Schematic illustration of the parameters characterizing the BH binary configurations under consideration: the mass ratio $q = M_2/M_1 > 1$, the initial separation d, the initial tangential linear momentum P_t, the initial inward radial linear momentum P_r, the dimensionless spin of each BH $\chi_i = |\mathbf{S}_i|/M_i^2$ and the angle of the spin in the orbital plane α relative to the outward radial direction of the initial BH positions.

4. Binary black-hole simulations with adaptive mesh refinement

In this section, we demonstrate the efficacy of some of the techniques discussed in section 3 in the context of BH binaries. To do this, we select a representative sample of BH binary configurations, analyze the accuracy of the resulting gravitational waveforms and compare the results obtained with GRChombo to that obtained with a more conventional moving boxes style mesh refinement code, Lean [14].

Before we present our results, we first provide details of the explicit tagging criteria used in our GRChombo simulations and the methods we use to analyze and compare our results.

4.1. Methods

We consider three different BH binary configurations with the parameters provided in table 1 and illustrated schematically in figure 2. All simulations include an inspiral, merger and ringdown.

The first configuration consists of two equal-mass non-spinning BHs with a quasicircular inspiral lasting about ten orbits; this configuration is labeled `q1-d12` (for mass ratio $q = 1$ and distance $d \approx 12$). The parameters were computed in order to minimize the initial eccentricity of the simulation using standard techniques [112].

The second configuration involves two BHs with mass ratio 2 : 1. The inspiral is about six orbits and is approximately quasicircular. This is one of the configurations simulated in the `lq1:2` sequence of [113][25]. Here we label this configuration `q2-d10` (for mass ratio $q = 2$ and distance $d = 10$).

The final configuration consists of a mildly eccentric inspiral of two equal-mass highly-spinning BHs. The spins lie in the plane as shown in figure 2, which is the 'superkick' configuration [115–117]. Here, the quantity we analyze is the gravitational recoil of the remnant BH. This configuration is taken from the sequence simulated in [114] and we label it `q1-s09` (for mass ratio $q = 1$ and spin $\chi = 0.9$).

4.1.1. GRChombo setup and tagging criteria. For the GRChombo simulations of the BH binary configurations in table 1, we use the CCZ4 equations (13)–(18) with the default damping parameters (27) (note that in code units, $M = 1$). We use the moving puncture gauge (28)–(31) with the default lapse parameters (29) and the shift parameters $b_1 = 1, b_2 = 3/4$, $M\eta = 1$ for `q1-d12` and `q1-s09` and $b_1 = 1, b_2 = 3/4$, $M\eta = 3/4$ for `q2-d10`. For `q1-d12` and `q2-d10`, we use reflective BCs along one boundary to impose bitant symmetry (i.e. symmetry across the equatorial plane) and Sommerfeld BCs for all other boundaries. Following sections 3.2, 3.3 and 3.5, we use a tagging criterion of the following form

$$C = \max\left(C_\chi, C_{\text{punc}}, C_{\text{ex}}\right), \tag{50}$$

where the quantities on the rhs are defined below. Note that we use the value $+\infty$ to denote a large value that always exceeds the threshold τ_{R}.

(a) C_χ tags regions in which the derivatives of the conformal factor χ become steep. It is the dominant criterion for the intermediate levels $l_{\text{ex}}^{\max} \leqslant l < l_{\max} - 3$, where $l_{\text{ex}}^{\max}$ is the maximum extraction level (see item (c) below). It is given by

$$C_\chi = \Delta x_l \sqrt{\sum_{i,j}\left(\partial_i\partial_j\chi\right)^2}, \tag{51}$$

where Δx_l is the grid spacing on refinement level l.

(b) C_{punc} includes parts of the tagging criterion that use the location of the punctures. It is the dominant criterion on the finest three levels and is comprised of two parts, C_{insp} and C_{merg} that are used depending on the coordinate distance between the punctures $s_{\text{p}} = |\mathbf{x}_{\text{p},1} - \mathbf{x}_{\text{p},2}|$ as follows:

$$C_{\text{punc}} = \begin{cases} C_{\text{insp}}, & s_{\text{p}} \geqslant M + b \\ \max(C_{\text{insp}}, C_{\text{merg}}), & 10^{-3} \leqslant s_{\text{p}} < M + b \\ C_{\text{merg}}, & s_{\text{p}} < 10^{-3}, \end{cases} \tag{52}$$

where $M = M_1 + M_2$ is the sum of the individual BH masses, b is a buffer parameter (cf section 3.2), and 10^{-3} is a choice in the cutoff for the distance between the punctures s_{p} which determines when the merger has completed. The inspiral criterion is given by

$$C_{\text{insp}} = \begin{cases} +\infty, & \text{if } \varrho_1 < (M_1 + b)2^{\min(l_{\max}-l-1,2)}, \\ & \text{or } \varrho_2 < (M_2 + b)2^{\min(l_{\max}-l-1,2)}, \\ 0, & \text{otherwise}, \end{cases} \tag{53}$$

[25] Note that, in this paper, GRChombo was not used to evolve this configuration.

Table 2. GRChombo grid parameters for the configurations in table 1 There are ($l_{\max}+1$) refinement levels and the coarsest level has length (without symmetries applied) L. The grid spacing on the finest level is $\Delta x_{l_{\max}}$ and the minimum number of cells in the buffer regions between consecutive refinement level boundaries is n_{B}. The regrid threshold for the tagging criterion (50) is τ_{R} and the BH buffer parameter is b.

Configuration	$l_{\max}$	L/M	$\Delta x_{l_{\max}}/M$	n_{B}	τ_{R}	b/M
`q1-d12` low	9	1024	$1/80$	20	0.016	0.7
`q1-d12` medium	9	1024	$1/96$	24	0.0133	0.7
`q1-d12` high	9	1024	$1/128$	32	0.01	0.7
`q2-d10` low	7	512	$1/88$	48	0.01	0.467
`q2-d10` medium	7	512	$1/104$	52	0.009 23	0.467
`q2-d10` high	7	512	$1/112$	56	0.008 57	0.467
`q1-s09` low	7	512	$1/64$	16	0.02	0.7
`q1-s09` medium	7	512	$1/96$	24	0.0133	0.7
`q1-s09` high	7	512	$1/112$	28	0.0114	0.7

where ϱ_i is the ‘max’ or ‘infinity’ norm (46) of the coordinate position vector relative to puncture i. Similarly, the merger criterion is given by

$$C_{\mathrm{merg}} = \begin{cases} +\infty, & \text{if } \varrho_{\odot} < (M+b)2^{\min(l_{\max}-l-1,2)}, \\ 0, & \text{otherwise}, \end{cases} \tag{54}$$

where $\varrho_{\odot}$ is the max-norm (46) of the coordinate position vector relative to the center of mass $\mathbf{x}_{\odot}$.

(c) C_{ex} ensures the Ψ_4 extraction spheres are suitably well resolved. It is the dominant tagging criterion for $0 \leqslant l < l_{\mathrm{ex}}^{\max}$ and is given by $C_{\mathrm{ex}} = \max\limits_i\{C_{\mathrm{ex},i}\}$, where i labels the extraction spheres and

$$C_{\mathrm{ex},i} = \begin{cases} +\infty, & \text{if } r < 1.2 r_{\mathrm{ex},i} \text{ and } l < l_{\mathrm{ex},i}, \\ 0, & \text{otherwise}, \end{cases} \tag{55}$$

where $r_{\mathrm{ex},i}$ and $l_{\mathrm{ex},i}$ are the radius and level of the ith extraction sphere and $l_{\mathrm{ex}}^{\max} = \max\limits_i l_{\mathrm{ex},i}$. The factor of 1.2 is present to add a 20% buffer radius around the extraction spheres in order to reduce the effect of spurious reflections off the refinement level boundaries.

A summary of the grid configuration parameters is given in table 2.

4.1.2. Comparison code: Lean. The Lean code [14] is based on the Cactus computational toolkit [8] and uses the method of lines with fourth-order RK time stepping and sixth-order spatial stencils. The Einstein equations are implemented in the form of the BSSNOK formulation [118–120] with the moving-puncture gauge [3, 4] (cf equations (28)–(31)). The Carpet driver [10] provides mesh refinement using the method of ‘moving boxes’. For the non-spinning binary configurations `q1-d12` and `q2-d10`, we use bitant symmetry to reduce computational expense, whereas configuration `q1-s09` is evolved without symmetries. The computational domains used for these simulations are characterized by the parameters listed in table 3. The domain comprises a hierarchy of $l_{\max}+1$ refinement levels labeled from $l = 0, \ldots, l_{\mathrm{F}}, \ldots, l_{\max}$, with grid spacing given by (37). Before applying the symmetry, for $l \leqslant l_{\mathrm{F}}$ each level consists of a single fixed cubic grid of half-length $R_l = R_0/2^l$, and for $l_F < l \leqslant l_{\max}$, each level consists of two cubic components of half-length $R_l = 2^{l_{\max}-l}R_{l_{\max}}$ centered around each BH puncture.

Table 3. LEAN grid parameters for the configurations in table 1. There are $(l_{\max}+1)$ levels of which the first $(l_{\mathrm{F}}+1)$ comprises a single box that covers both BHs with the remaining levels consisting of two separate box components that cover each BH separately. The half-width of the coarsest level is R_0 and the half-width of a single component on the finest level is $R_{l_{\max}}$. The grid spacing on the finest level for the three resolutions used in the convergence analysis is $\Delta x_{l_{\max}}$.

Configuration	$l_{\max}$	l_{F}	R_0/M	$R_{l_{\max}}/M$	$\Delta x_{l_{\max}}/M$
`q1-d12`	9	5	512	1/2	1/64, 1/96, 1/128
`q2-d10`	8	4	256	1/3	1/84, 1/96, 1/108
`q1-s09`	8	3	256	1	1/80, 1/88, 1/96

We adopt this notation for consistency with that used to describe GRCHOMBO. This translates into the more conventional LEAN grid setup notation (cf [14]) as

$$\left\{(R_0,\ldots,2^{-l_{\mathrm{F}}}R_0)\times(2^{l_{\max}-l_{\mathrm{F}}-1}R_{l_{\max}},\ldots,R_{l_{\max}}),\Delta x_{l_{\max}}\right\}. \tag{56}$$

A CFL factor of $1/2$ is used in all simulations, and apparent horizons are computed with AHFinderDirect [121, 122].

For all our BH evolutions, with LEAN and GRCHOMBO, the initial data are constructed with the TwoPunctures spectral solver [85].

4.1.3. Gravitational wave analysis. One of the most important diagnostics from our simulations is the GW signal which we compute from the Weyl scalar Ψ_4. For GRCHOMBO, the calculation of Ψ_4 is explained in appendix A and technical details of the extraction procedure can be found in section 2.2.4. For LEAN, details can be found in [14]. Below, we describe further analysis we have performed in order to compare the GW output from each code.

We start with the multipolar decomposition of the Weyl scalar,

$$\Psi_4=\sum_{l=2}^{\infty}\sum_{m=-\ell}^{\ell}{}_{-2}Y^{\ell m}\psi_{\ell m}. \tag{57}$$

Next, we translate to the gravitational-wave strain h according to

$$\Psi_4=\ddot{h}=-\ddot{h}^{+}+\mathrm{i}\ddot{h}^{\times} \tag{58}$$

which gives us the strain multipoles as $\ddot{h}^{+}_{\ell m}=-\mathbf{Re}(\psi_{\ell m})$ and $\ddot{h}^{\times}_{\ell m}=\mathbf{Im}(\psi_{\ell m})$. To avoid spurious drift resulting from numerical inaccuracies, we perform the necessary integrations in time in the Fourier domain [123]. We then rewrite the strain modes in terms of their amplitude and phase

$$-h^{+}_{\ell m}+\mathrm{i}h^{\times}_{\ell m}=h^{\mathrm{A}}_{\ell m}\exp\left(\mathrm{i}h^{\phi}_{\ell m}\right), \tag{59}$$

where multiples of 2π are added to $h^{\phi}_{\ell m}$ appropriately in order to minimize the difference between consecutive data points.

The *radiated* quantities derived from Ψ_4 are affected by two main error sources; the discretization error due to finite resolution and an uncertainty arising from the extraction at finite radii instead of null infinity. We determine the former by conducting a convergence analysis of the quantities extracted at finite radius. In order to determine the second error contribution, we compute a given radiated quantity f at several finite extraction radii and extrapolate to infinity by fitting a polynomial in $1/r$ of the form

$$f_N(u,r) = \sum_{n=0}^{N} \frac{f_{n,N}(u)}{r^n}. \tag{60}$$

Here, r is the coordinate radius and $u = t - r^*$ denotes the retarded time evaluated with the tortoise coordinate

$$r^* = r + 2M\,\ln\left|\frac{r}{2M} - 1\right|. \tag{61}$$

We uniformly observe that time shifts in terms of r^* result in slightly better alignment of wave signals extracted at different coordinate radii r. If we take $f_{0,N}(u)$ as our estimate of the extrapolated quantity, we then estimate the error ϵ in our result from $r = r_{\text{ex}}$ by computing

$$\epsilon_{f,r_{\text{ex}},N} = \left|f(u,r_{\text{ex}}) - f_{0,N}(u)\right|. \tag{62}$$

Typically, and unless stated otherwise, we set $N = 1$ and drop the N subscripts. Our total error budget is then given by the sum of the discretization and extraction uncertainties.

We quantify the agreement between the two codes' results in the context of GW analysis by computing the overlap following the procedure of [124, 125]. In the following, we restrict our analysis to the dominant (2, 2) quadrupole part of the signal and drop the subscript '$\ell = 2, m = 2$'. Before computing the overlap, we extrapolate the strain to infinity using the procedure explained above.

Given the power spectral density $S_n(f)$ of a detector's strain noise as a function of frequency f, the inner product of two signals g, h on the space of waveforms is given by[26]

$$\langle g|h\rangle := 4\,\mathsf{Re}\left\{\int_0^\infty \frac{\tilde{g}^*(f)\tilde{h}(f)}{S_n(f)}\,\mathrm{d}f\right\}, \tag{63}$$

where the Fourier transform is defined by

$$\tilde{g}(f) := \int_{-\infty}^{\infty} g(t)\mathrm{e}^{-2\pi \mathrm{i} f t}\,\mathrm{d}t. \tag{64}$$

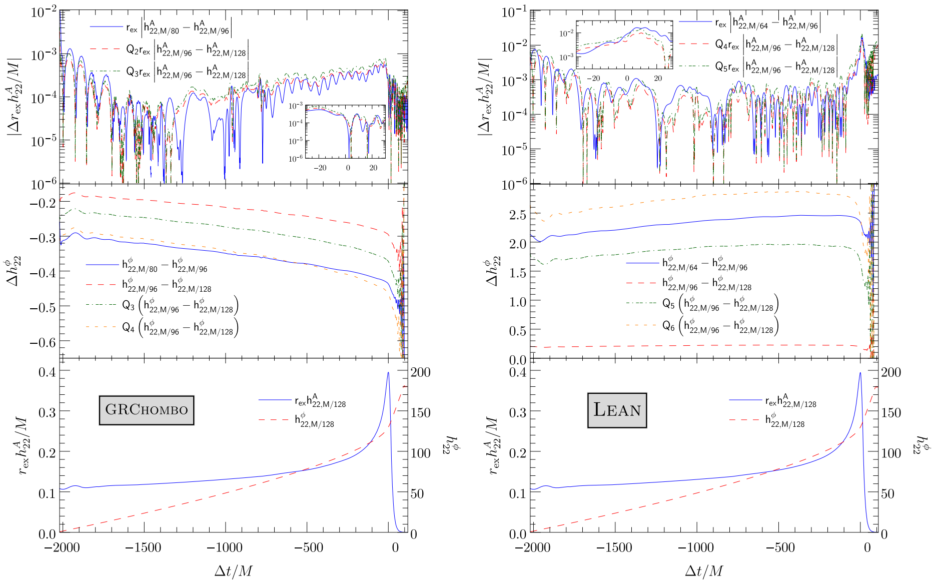
We next define the overlap of the two signals as the normalized inner product maximized over shifts Δt, Δh^ϕ in time and phase,

$$\rho(g,h) := \max_{\Delta h^\phi,\Delta t} \frac{\langle g|h\rangle}{\sqrt{\langle g|g\rangle\langle h|h\rangle}}. \tag{65}$$

The quantity $1 - \rho(g,h)$ then provides a measure for the *discrepancy* between the two waveforms, analogous to the *mismatch* introduced as a measure for signal-to-noise reduction due to model imperfections in GW data analysis [126, 127].

For `q1-s09`, we instead analyze the convergence of the linear momentum radiated in GWs in the form of the BH recoil velocity or kick. First we compute the radiated momentum $\mathbf{P}^{\text{rad}}$ using equation (7) of [113] and then compute the recoil velocity–which must lie in the

[26] We use in our calculation one-sided, as opposed to two-sided, spectral power densities, i.e. we only consider non-negative frequencies, hence the factor 4 in (63).



(a)

(b)

Figure 3. Convergence of the quadrupole mode of the strain $h_{22} = -h^+_{22} + \mathrm{i}h^\times_{22}$ calculated from the values of Ψ_4 extracted for configuration `q1-d12` at $r_{\rm ex} = 120M$ for both GRChombo with finest grid resolutions $\Delta x_{l_{\max}} = M/80$, $M/96$ and $M/128$ (a) and Lean with finest grid resolutions $\Delta x_{l_{\max}} = M/64$, $M/96$ and $M/128$ (b). Top panels: convergence of the amplitude $h^{\rm A}_{22} = |h_{22}|$. The difference between the higher resolution results is rescaled according to fourth and fifth order convergence for Lean and according to second and third-order convergence for GRChombo. In each case, the inset shows an interval around the peak amplitude. Middle panels: convergence of the phase $h^\phi_{22} = \mathrm{Arg}(h_{22})$. The difference between the higher resolution results is rescaled according to fifth and sixth order convergence for Lean and third and fourth order convergence for GRChombo. Bottom panels: for reference we plot the amplitude $h^{\rm A}_{22}$ and the phase h^ϕ_{22} of the highest resolution waveform on the same time axis. For the two lower resolution waveforms from each code, we have time-shifted each of them in order to maximize the overlap (cf (65)) with the highest resolution waveform. $\Delta t = 0$ corresponds to the maximum in $h^{\rm A}_{22}$ for the highest resolution waveform.

z-direction by symmetry—using $v = -P^{\rm rad}_z/M_{\rm fin}$. Since the radiated momentum can be written in terms of a sum, with each term involving several multipolar amplitudes $\psi_{\ell m}$ (equation (40) in [128]), analyzing this quantity has the benefit of additionally indirectly comparing the agreement of higher order multipoles (i.e. $\ell > 2$) between the codes.

4.2. Results

For each configuration in table 1, we have performed three simulations at different resolutions with both GRChombo and Lean in order to calibrate their accuracy which we discuss below. The respective grid configurations are given in tables 2 and 3.

For the first configuration `q1-d12` of an equal-mass binary, we show the convergence analysis in figure 3 with the analysis for GRChombo on the left and for Lean on the right. For Lean, we observe convergence of about fourth order in the amplitude and between fifth and sixth order in the phase of the quadrupole mode h_{22} of the strain (59). For GRChombo we observe convergence of about second order in the amplitude and about fourth order in the

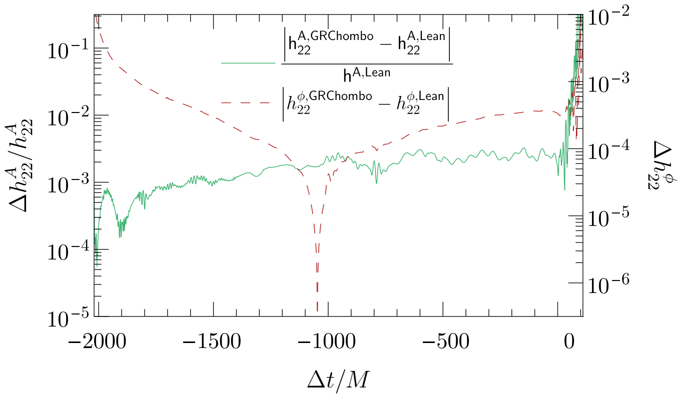


Figure 4. The relative and absolute difference between the GRChombo and Lean outputs for the amplitude and phase of h_{22} respectively from the simulation of configuration `q1-d12`. In both cases, the data comes from the simulations with finest grid spacing $\Delta x_{l_{\max}} = M/128$ with Ψ_4 extracted at $r_{\mathrm{ex}} = 120M$. As for the convergence plots in figure 3, the time has been shifted in order to maximize the overlap (cf (65)) between the two waveforms and $\Delta t = 0$ at the peak in $h_{22}^{\mathrm{A,Lean}}$.

phase of the same mode. We note that, as mentioned in [113], higher resolutions were required with GRChombo in order to enter the convergent regime.

By comparison with a Richardson extrapolation, we estimate the discretization errors in the amplitude and phase of the finest resolution simulations from both codes as follows. Excluding the early parts of the signal dominated by 'junk' radiation and the late part of the ringdown which is dominated by noise, we obtain a discretization error of $\Delta h_{22}^{\mathrm{A}}/h_{22}^{\mathrm{A}} \lesssim 1\,\%$ in the amplitude assuming fourth order convergence for Lean and second order convergence for GRChombo. Up to the late ringdown where the phase becomes inaccurate, we estimate the phase error is $\Delta h_{22}^{\phi} \leqslant 0.15$ assuming sixth order convergence for Lean and fourth order convergence for GRChombo.

Following the procedure in section 4.1.3, we estimate the error in the phase, due to finite-radius extraction, is $\epsilon_{h_{22}^{\phi},120M} \lesssim 0.4$, and, in the amplitude is $\epsilon_{h_{22}^{\mathrm{A}},120M}/h_{22}^{\mathrm{A}} \lesssim 8\,\%$ (although this steadily decreases towards $\lesssim 2\%$ near merger) for both codes. Here we have ignored the early part of the signal where the amplitude is dominated by the 'junk' radiation up to $u = 300M$.

We next directly compare the results of the two codes by computing the relative difference in the amplitude h_{22}^{A} and the absolute difference in the phase h_{22}^{ϕ} which is shown in figure 4. Again, ignoring the early part of the signal and the late ringdown, the relative difference in the amplitudes is $\lesssim 1\%$, consistent with the individual error estimates from the two codes. The discrepancy in phase remains $\mathcal{O}(10^{-3})$ or smaller throughout the inspiral, merger and early ringdown-well within the error estimates of each code.

For the first asymmetric BH binary configuration, `q2-d10`, we proceed in the same way. We study the convergence in analogy to figure 3. Ignoring again the contamination at early times, we obtain third-order convergence in the amplitude and fifth-order convergence in the phase for Lean. For GRChombo, we obtain fourth order convergence in the amplitude and mild overconvergence of about eighth order in the phase[27]. This leads to uncertainty estimates of $\Delta h_{22}^{\mathrm{A}}/h_{22}^{\mathrm{A}} \lesssim 2.5\,\%$ in the amplitude and $\Delta h_{22}^{\phi} \lesssim 0.25$ in the phase for both codes.

[27] We assume fourth order convergence for our GRChombo phase error.

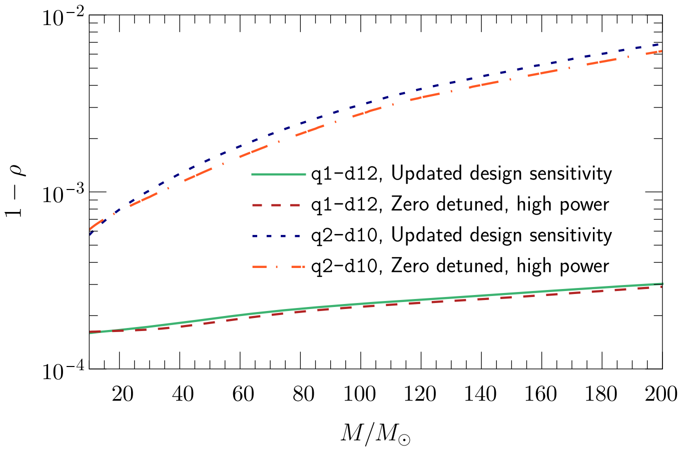


Figure 5. The discrepancy $1-\rho$ between the $(\ell, m) = (2, 2)$ mode of the GW signal from the `q1-d12` (ten orbits, non-spinning, equal mass) and `q2-d10` (six orbits, non-spinning, $2:1$ mass ratio) BH binary configurations simulated with LEAN and GRCHOMBO. For `q1-d12`, we use the simulation with resolution $\Delta x_{l_{\max}} = M/128$ for both codes and for `q2-d10`, we use the simulation with resolution $\Delta x_{l_{\max}} = M/96$ for both codes. For each configuration, we show the difference computed with the updated Advanced LIGO sensitivity design curve (`aLIGODesign.txt` in [129]) and the zero detuned, high power noise curve from the Advanced LIGO anticipated sensitivity curves (`ZERO_DET_high_P.txt` in [130]).

The error due to finite-radius extraction in the amplitudes is $\epsilon_{h^{A}_{22},86.7M}/h^{A}_{22} \lesssim 10\,\%$ in the early inspiral decreasing down to $\lesssim 2\%$ in the late inspiral, and in the phase is $\epsilon_{h^{\phi}_{22},86.7M} \sim 0.5$ for both codes.

In figure 5, we display as a function of the total mass M the discrepancy $1-\rho$ (where ρ is the overlap given by (65)) between the GRCHOMBO and LEAN waveforms for both `q1-d12` and `q2-d10`, with the spectral noise density $S_n(f)$ given by (i) the updated Advanced LIGO sensitivity design curve (`aLIGODesign.txt` in [129]) and (ii) the zero detuned, high power noise curve from the Advanced LIGO anticipated sensitivity curves (`ZERO_DET_high_P.txt` in [130]). For `q1-d12`, the figure demonstrates excellent agreement of the two waveforms for the entire range $M = 10, \ldots, 200M_\odot$ with a discrepancy $1-\rho \approx 0.03\%$ or less, whereas for `q2-d10`, the agreement is not quite as strong but nevertheless demonstrates very good consistency with a discrepancy $1-\rho \approx 0.7\%$ or less. The larger difference for `q2-d10` compared to `q1-d12` may be attributed to the slightly lower resolutions employed for this configuration, especially near the smaller BH. To put these numbers into context, Lindblom *et al* [127] estimate that a mismatch of 3.5% would result in a reduction in the GW event detection rate by about 10%.

Our final BH binary features asymmetry in the form of non-zero spins. This time, we focus on the BH recoil velocity v calculated from the linear momentum radiated in GWs, and the analysis is shown in figure 6. From the plots, we can see that LEAN exhibits convergence between third and fourth order, whilst GRCHOMBO exhibits convergence between fourth and fifth order. We illustrate our estimate of the total error for each code in figure 7. Here, the error bands—around the curve from the highest resolution simulation in each case—correspond to the difference with the Richardson extrapolated curve assuming fourth-order convergence plus

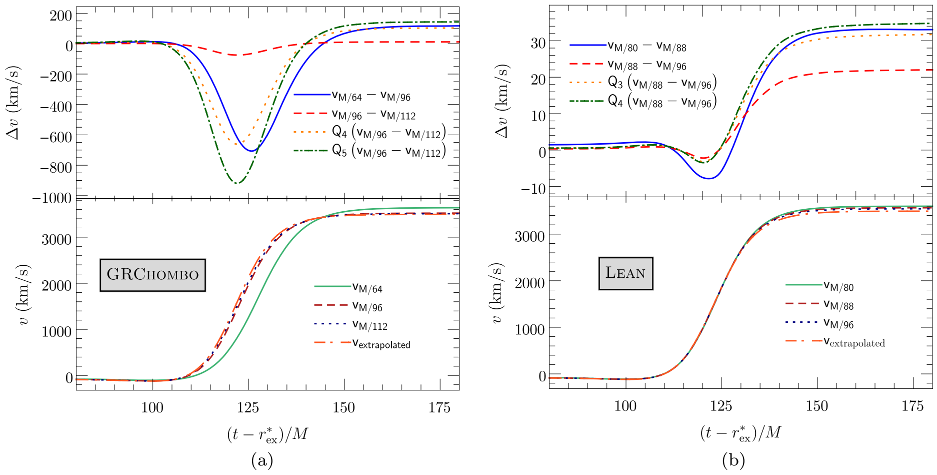


Figure 6. Convergence plots for the accumulated linear momentum radiated from configuration `q1-s09` for GRChombo with finest grid resolutions $\Delta x_{l_{\rm max}} = M/64$, $M/96$ and $M/112$ (a) and for Lean with finest grid resolutions $\Delta x_{l_{\rm max}} = M/80$, $M/88$ and $M/96$ (b). This is shown in the form of the BH recoil velocity in the bottom panels. For both codes, the radiated linear momentum is calculated from the extracted Ψ_4 values at $r_{\rm ex} = 90M$, and the extrapolated curve corresponds to a Richardson extrapolation assuming fourth order convergence. In the top panels, we show the difference between the results from different resolutions with rescalings according to third and fourth order convergence for Lean and according to fourth and fifth order convergence for GRChombo.

the estimated error due to finite-radius extraction (about $1.5\% + 2\%$ for Lean and $0.5\% + 3\%$ for GRChombo). This total error is about 3.5% for both codes.

As is not uncommon, the convergence orders obtained from NR simulations can be fickle due to the various ingredients in the codes with differing orders of accuracy which can dominate in certain regimes. This inherent complexity makes it difficult to attribute the difference in convergence orders we obtain between the two codes and we therefore do not attempt to do so.

5. Comparing tagging criteria using axion stars

In order to demonstrate the application of our techniques to problems with matter fields and dynamically varying length scales, we consider the evolution of a single axion star—a compact object composed of a real scalar bosonic field. We analyze the evolution of a star that is stable on the timescale of the simulation, as well as one in which the self-interaction is increased in order to trigger gravitational collapse to a BH. As discussed previously, this simple example tests many of the key requirements in using AMR to evolve fundamental fields coupled to gravity, in particular, the ability to follow stable oscillations and to adapt to changing dynamical timescales. Similar considerations apply, for example, to cosmological spacetimes, cosmic strings and collisions of exotic compact objects.

We demonstrate the use of two effective tagging criteria; first, tagging by the magnitude of terms in the Hamiltonian constraint (70), and second, by the numerical truncation error between refinement levels (48).

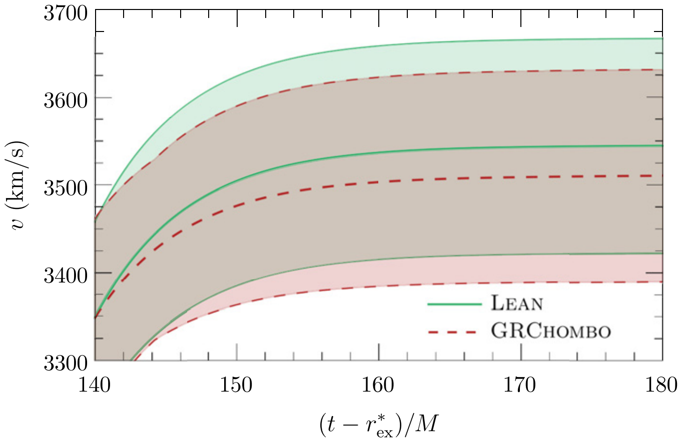


Figure 7. The accumulated radiated linear momentum at the end of the highest resolution simulations of configuration `q1-s09` from each code. The linear momentum is shown in the form of the BH recoil velocity and the error bands show our estimate of the total error coming from both discretization and finite-radius effects.

5.1. Methods

5.1.1. Setup. We consider the evolution of two different axion star configurations. Axion stars are quasi-equilibrium configurations of a self-gravitating real scalar field ϕ [131] that is subject to a periodic self-interaction potential $V(\phi)$. A canonical potential is

$$V(\phi) = \mu^2 f_{\mathrm{a}}^2[1 - \cos(\phi/f_{\mathrm{a}})], \tag{66}$$

where this form arises as a result of the spontaneously broken $U(1)$ Peccei–Quinn symmetry and subsequent 'tilting' of the potential due to instanton effects [132, 133]. The decay constant f_{a} quantifies the symmetry breaking scale and determines the strength of the scalar field self-interactions (their strength for a given central amplitude is inversely related to f_{a}) and $\mu = m_{\mathrm{a}}c/\hbar$ is an inverse length scale related to the scalar mass[28] m_{a}. Axion stars on the main stability branch are characterized by their central amplitude ϕ_0 or equivalently their ADM mass $M_{\mathrm{ADM}} \sim \mu^{-1}$. They have a physical size R (defined as the radius containing 99% of the total mass) that is approximately inversely related to their ADM mass, and thus a useful descriptor is their compactness $\mathcal{C} = M_{\mathrm{ADM}}/R$. Axion stars with $\mathcal{C} \sim 1/2$ are highly relativistic and may form BHs if they collapse or collide. For $m_{\mathrm{a}} \sim 10^{-14}$ eV, they are of a mass and size comparable to solar mass BHs, and thus potentially of astrophysical interest. Further details related to the setup used here can be found in [95, 106], and a useful general review of axion physics is provided in [134].

The equation of motion for the scalar field ϕ is given by the Klein–Gordon equation for a real scalar field minimally coupled to gravity

$$\nabla_\mu \nabla^\mu \phi - \frac{\mathrm{d}V}{\mathrm{d}\phi} = 0, \tag{67}$$

[28] In Planck units one can write $\mu = m_{\mathrm{a}}/M_{\mathrm{Pl}}^2$, where M_{Pl} is the Planck mass.

and the system is completed with the Z4 equation (1) for the metric components. To construct localised, quasi-equilibrium oscillatory (axion star) solutions, we solve the Einstein–Klein–Gordon (EKG) system of equations with a harmonic field ansatz and appropriate boundary conditions [131, 135].

Unlike the case of complex scalar boson stars, for axion star solutions the metric components $g_{\mu\nu}$ also oscillate in time, with energy being transferred between the matter and curvature terms in the Hamiltonian constraint [106]. This makes them challenging targets for dynamical refinement; simple criteria based solely on matter field gradients will fail to achieve a stable grid structure, as the gradients change over time even in the quasi-stable case. If the gradients are close in value to a tagging threshold, frequent regridding will occur, which introduces errors.

The stability of the axion star solution comes from the balance of its tendency to disperse due to gradient pressure from spatial field derivatives, with the tendency to collapse due to its energy density. The relative strengths of these effects determines whether the axion star remains stable, disperses through scalar radiation or collapses to a BH when perturbed. In particular, if the self-interaction scale f_{a} is too low, this can cause the axion star to collapse to a BH [98]. In the last case, a key AMR challenge is determining tagging criteria that progressively track the axion star collapse without triggering too frequent regridding from the more rapid field oscillations.

We consider two cases, both with central amplitude $\phi_0 = 0.020$ and $\mu M_{\mathrm{ADM}} = 0.4131$:

(a) An axion star with weak self-interactions ($f_{\mathrm{a}} = 1$),[29] where the scalar field and metric oscillate over time in a localized configuration that is stable over time periods much longer than that of the simulation. We would ideally like the refinement to remain constant, despite the oscillations of the fields.

(b) An unstable configuration where we increase the attractive self interaction by reducing the self interaction scale to $f_{\mathrm{a}} = 0.05$, such that the axion star is destabilized and undergoes collapse to a BH. We need the mesh refinement to follow this process sufficiently rapidly, but without excessive regridding.

To evolve this system in GRCHOMBO, the EKG equation (67) is decomposed into two first order equations in the 3 + 1 formulation

$$\partial_t \phi = \beta^i \partial_i \phi + \alpha \Pi, \tag{68}$$

$$\partial_t \Pi = \beta^i \partial_i \Pi + \alpha \gamma^{ij} (\partial_i \partial_j \phi + \partial_i \phi \partial_j \alpha) + \alpha \left(K\Pi - \gamma^{ij} \Gamma^k_{ij} \partial_k \phi - \frac{\mathrm{d}V}{\mathrm{d}\phi} \right), \tag{69}$$

and added to the CCZ4 evolution scheme (13)–(18). The initial data are set up as in the previous study [98] using the numerically obtained axion star profile for an $m^2\phi^2$ potential [131, 135, 136]. We choose the initial hypersurface such that $\phi = 0$ and hence $V(\phi) = 0$ everywhere. The Hamiltonian constraint is thereby satisfied for both the $V = m^2\phi^2$ and the axion potential (66) cases. Furthermore, if we impose the extrinsic curvature $K_{ij} = 0$, the momentum constraint is trivially satisfied and all the dynamical information is encoded in the kinetic term of the field Π. The system is evolved in the moving puncture gauge using the default 1 + log parameters (29) for the lapse evolution equation (28) with the exception of $a_1 = 0$ and the default Gamma-driver shift parameters for (30) and (31) with $\eta = \mu$.

[29] The decay constant f_{a} is dimensionless in geometric units as used here; to obtain its value in Planck units one simply multiplies by the Planck mass M_{Pl}.

5.1.2. Tagging criteria. We demonstrate the suitability of two different tagging methods for tracking the axion star evolution in both the stable and unstable cases. In the case of collapse to a BH, for both tagging methods, the threshold τ_{R} must be chosen such that the apparent horizon is covered entirely by the finest refinement level, as discussed in section 3.2. In our case, we choose a maximum refinement level $l_{\max} = 8$, which covers up to $\mu r \geqslant 0.5 > \mu M_{\mathrm{ADM}}$.

The first tagging criterion we consider is based on physical quantities in the simulation, as outlined in section 3.5. We choose the absolute sum of the terms in the Hamiltonian constraint $\mathcal{H}_{\mathrm{abs}}$ (49), setting the criterion

$$C(\mathbf{i}) \equiv \mathcal{H}_{\mathrm{abs}} \tag{70}$$

in the tagging indicator function (38).

We also show the efficacy of truncation error tagging, outlined in section 3.4. We choose the variables f as defined in (47) to be χ, K, ϕ and π, as these capture the information in the Hamiltonian constraint $\mathcal{H}$. We use the tagging criterion (48), explicitly

$$C(\mathbf{i}) = \sqrt{(\tau_{l,\chi}(\mathbf{i}))^2 + (\tau_{l,K}(\mathbf{i}))^2 + (\tau_{l,\phi}(\mathbf{i}))^2 + (\tau_{l,\pi}(\mathbf{i}))^2}, \tag{71}$$

where $\tau_{l,f}(\mathbf{i})$ is defined by (47) and we have set the normalizing factor $L_f = 1$ for all f.

5.1.3. Diagnostics and convergence testing. We perform convergence tests using several key physical quantities from the evolution. The first quantity is the L^2 norm of the Hamiltonian constraint violations (33) $\|\mathcal{H}\|_2$ over a coordinate volume $\mathcal{V}$

$$\|\mathcal{H}\|_2 = \sqrt{\int_{\mathcal{V}} \mathrm{d}^3x\, \mathcal{H}^2}. \tag{72}$$

For the case of a stable axion star, in order to exclude the constraint violation at the outer boundaries, we choose $\mathcal{V}$ to be the volume enclosed by a sphere of fixed coordinate radius r_{out}, $B_{r_{\mathrm{out}}}$, with a center that coincides with that of the star:

$$\mathcal{V} = B_{r_{\mathrm{out}}}. \tag{73}$$

In the case of an unstable axion star that collapses to a BH, we furthermore excise the volume enclosed by a smaller sphere of fixed coordinate radius $r_{\mathrm{in}} < r_{\mathrm{out}}$ with the same center in order to exclude the constraint violations near the puncture that arises after the collapse. The radius r_{in} is chosen such that the sphere will lie within the apparent horizon once it is formed. This means that

$$\mathcal{V} = B_{r_{\mathrm{out}}} \backslash B_{r_{\mathrm{in}}}. \tag{74}$$

For the stable axion star, we also test convergence using the total mass of the matter content M_{mat} within the same volume

$$M_{\mathrm{mat}} = -\int_{\mathcal{V}} \mathrm{d}^3x\, \sqrt{\gamma}\, T^0_0 = \int_{\mathcal{V}} \mathrm{d}^3x\, \sqrt{\gamma}\left(\alpha\rho - \beta_k S^k\right), \tag{75}$$

where ρ and S^i are defined by (6), $\gamma = \det(\gamma_{ij})$ and α and β_i are the lapse and shift as defined by (2). Further details can be found in [106].

Table 4. GRChombo grid parameters for axion star configurations using different tagging criteria. There are $(l_{\max}+1)$ refinement levels and the coarsest level has length (without symmetries applied) L. The grid spacing on the coarsest level is Δx_0 and the minimum number of cells in the buffer regions between consecutive refinement level boundaries is n_{B}. The regrid thresholds for the different tagging criteria are given by τ_{R}. We consider two cases; a stable axion star ($f_{\mathrm{a}}=1$) and an unstable collapse to a BH ($f_{\mathrm{a}}=0.05$), with $\mu=m_{\mathrm{a}}c/\hbar=1$ in code units.

f_{a}	Tagging	$l_{\max}$	μL	$\mu\Delta x_0$	n_{B}	τ_{R}
1.0	Ham	3	512	4	8	0.1
1.0	Ham	3	512	2	16	0.05
1.0	Ham	3	512	1	32	0.025
1.0	Trunc	3	512	4	8	0.0625
1.0	Trunc	3	512	2	16	3.91×10^{-3}
1.0	Trunc	3	512	1	32	2.44×10^{-4}
0.05	Ham	8	1024	4	8	0.1
0.05	Ham	8	1024	2.67	12	0.067
0.05	Trunc	8	1024	4	8	0.001
0.05	Trunc	8	1024	2.67	12	2.96×10^{-4}

5.2. Results

We have performed simulations of axion stars at different resolutions for each configuration in table 4. We evolve with GRChombo and investigate two different tagging criteria: refinement using (i) the Hamiltonian constraint $\mathcal{H}_{\mathrm{abs}}$ (70) and (ii) the truncation error of the variables χ, K, ϕ and π as defined by (71).

As outlined in section 3.5, for the stable axion star configuration ($f_{\mathrm{a}}=1.0$), we expect Hamiltonian constraint tagging to generate stable refinement levels. We set τ_{R} such that we obtain an appropriate initial grid structure, which we choose to have a maximum refinement level $l_{\max}=3$ with refinement concentrated on the axion star.

The middle panel of figure 8 shows M_{mat} for the finest grid configuration for a stable axion star in table 4, where M_{mat} is calculated as defined in (75) for a coordinate sphere with radius $\mu r_{\mathrm{out}}=25$. We observe some initial gauge evolution of M_{mat} due to the transition from the initial polar-areal gauge to puncture gauge, with subsequent regular oscillations over time, which are physical. Given that M_{mat} includes only matter contributions and the mass of an axion star is approximately constant (i.e. the total flux out of the outer spherical surface is zero), these oscillations indicate the transfer of energy between curvature and matter terms as discussed in section 5.1.1.

The upper panel of figure 8 shows the difference in mass ΔM_{mat} for the simulations in table 4. This demonstrates convergence between third and fourth order in M_{mat} at late times, but second order near the beginning of the simulation. This agrees with expectations; the initial data used is accurate to second order, and this error dominates at early times, with the fourth order convergence related to the evolution scheme only being recovered at later times. Some error is also introduced by the interpolation of the initial conditions onto the grid, which is first order (but with a high spatial resolution in the numerical solution so this is subdominant). By comparing the highest resolution simulation with a Richardson extrapolation, we obtain a discretization error estimate of $\Delta M_{\mathrm{mat}}/M_{\mathrm{mat}}\lesssim 4\times10^{-5}$ at late times (using third order extrapolation).

The lower panel of figure 8 shows the L^2 norm of the Hamiltonian constraint violations $\|\mathcal{H}\|_2$ for the same simulations. Again, we measure between third and fourth order convergence at

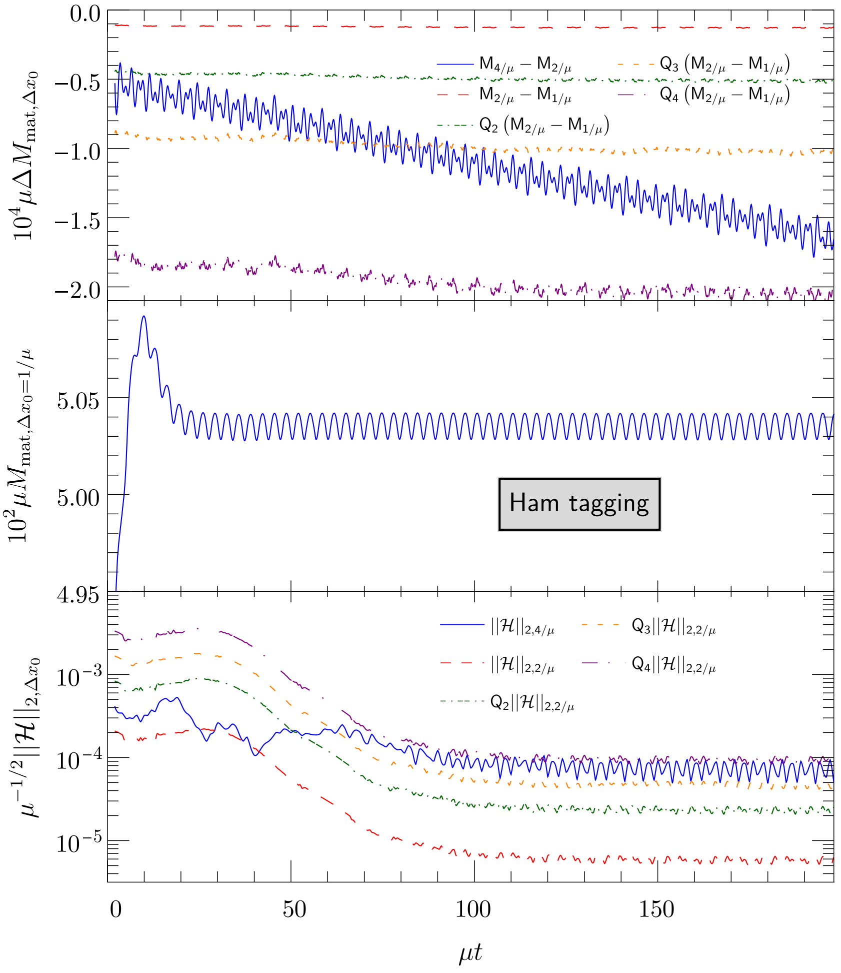


Figure 8. Convergence plots for the stable axion star configuration ($f_{\mathrm{a}} = 1.0$) with Hamiltonian constraint tagging (70) and grid configurations given in table 4. The top panel shows the difference in the calculated matter mass M_{mat} (75) within a sphere of radius $\mu r_{\mathrm{out}} = 25$ between the resolutions with rescalings according to second, third and fourth order convergence. For reference, in the middle panel, we show M_{mat} for the highest resolution simulation (with $\mu \Delta x_0 = 1$). In the bottom panel, we plot the L^2 norm of the Hamiltonian constraint $\|\mathcal{H}\|_2$ (72) for the two lower resolution simulations ($\mu \Delta x_0 = 4, 2$) in addition to rescalings according to second, third and fourth order convergence. We omit the corresponding plots for the simulations with truncation error tagging (71) as they are qualitatively very similar.

late times, and between first and second order initially. We obtain an error measure at late times of $\sqrt{M_{\mathrm{ADM}}}\,\Delta\|\mathcal{H}\|_2/16\pi \lesssim 8 \times 10^{-8}$, where we have normalized with the ADM mass to create a dimensionless measure of the spurious energy density.

We perform the same analysis of M_{mat} and $\|\mathcal{H}\|_2$ for the stable axion star using truncation error tagging with the parameters in table 4. We obtain a very similar grid structure and evolution behaviour to Hamiltonian tagging, with the same convergence and error estimates, demonstrating that both methods can achieve equivalent, accurate results.

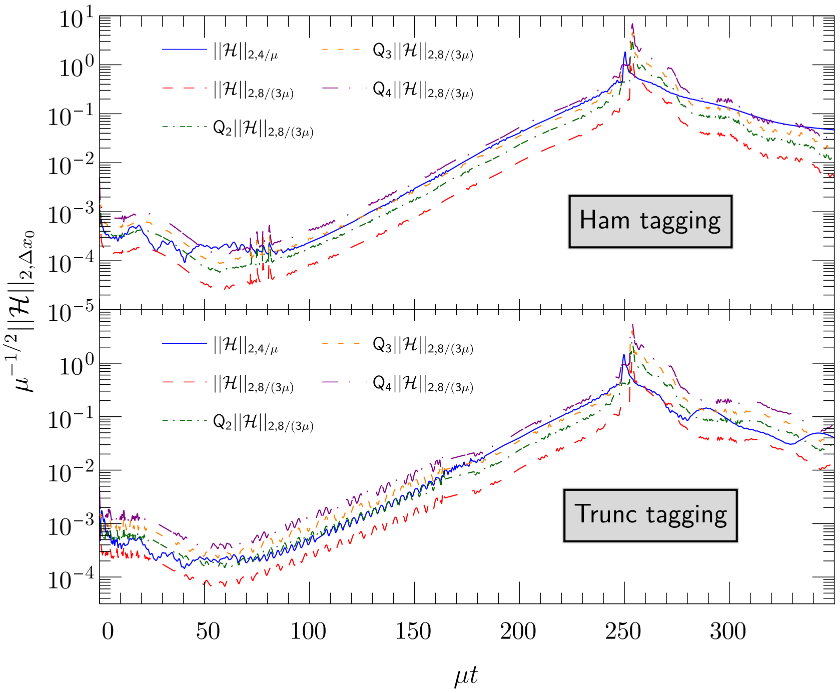


Figure 9. Convergence plots of the L^2 norm of the Hamiltonian constraint (72) contained between spheres of radius $\mu r_{\text{in}} = 0.5$ and $\mu r_{\text{out}} = 25$ for the unstable axion star configuration ($f_{\text{a}} = 0.05$) with both Hamiltonian constraint tagging (70) (top panel) and truncation error tagging (71) (bottom panel). The grid configurations are provided in table 4 and we additionally plot rescalings according to second, third and fourth order convergence.

For the unstable axion star configuration ($f_{\text{a}} = 0.05$), we perform convergence testing on $\|\mathcal{H}\|_2$ within the spatial volume with $0.5 < \mu r < 25$, excising the region $\mu r < 0.5$ where a BH is formed. We use the parameters given in table 4. We do not perform a convergence test of M_{mat}, as the quantity oscillates with a high frequency about a mean value that rapidly decreases to zero around the collapse, making such an analysis impractical.

The top panel of figure 9 shows the convergence analysis of $\|\mathcal{H}\|_2$ with Hamiltonian constraint tagging. We observe approximately third order convergence prior to collapse to a BH, then approximately fourth order convergence at late time. We obtain a maximum error measure on the finer grid of $\sqrt{M_{\text{ADM}}}\Delta\|\mathcal{H}\|_2/16\pi \lesssim 0.021$, with this value occurring approximately at the collapse.

For the same configuration with truncation error tagging in the lower panel of figure 9, we obtain similar convergence prior to the collapse with a maximum error measure $\sqrt{M_{\text{ADM}}}\Delta\|\mathcal{H}\|_2/16\pi \lesssim 0.015$. The convergence after the collapse is lower: between first and third order. In general we see that the rapid regridding that occurs during a collapse (often triggered at different times at the different resolutions) introduces errors which can reduce the convergence order. This illustrates one of the main challenges of AMR, which is to obtain good convergence in highly dynamical regimes.

6. Conclusion

In this work, we have presented a detailed discussion of the use of fully AMR in NR simulations with the GRChombo code. To avoid confusion, we first summarize how the term 'fully adaptive' is meant to distinguish AMR from the common (and often highly successful) box-in-a-box approach. This distinction consists in two main features. First, we use the term AMR in the sense that it allows for refined regions of essentially arbitrary shape. Second, it identifies regions for refinement based on a point-by-point interpretation of one or more user-specifiable functions of grid variables. Of course, a region of arbitrary shape will inevitably be approximated by a large number of boxes on Cartesian grids; the distinction from a box-in-a-box approach therefore consists in the *large number* of boxes used in AMR. Likewise, every box-in-a-box approach will ultimately base its dynamic regridding on some function of the evolved grid variables, as the apparent horizon. The key feature of AMR is the *pointwise* evaluation of grid variables or their derived quantities.

We have laid the foundations for our study in sections 2 and 3 with a comprehensive summary of the formulation of the Einstein equations and the AMR infrastructure of GRChombo. In short, we employ the CCZ4 equations (13)–(18), on a Cartesian mesh with a user-specified number of refinement levels with sixth or fourth-order spatial discretization and fourth-order RK time stepping according to the method of lines. The tagging of grid points for refinement and the corresponding regridding is performed according to the Berger–Rigoutsos AMR algorithm summarized in section 2.2.2 and figure 1.

The advantages of AMR based simulations over the simpler box-in-a-box structure evidently arise from its capability to flexibly adapt to essentially any changes in the shape or structure of the physical system under consideration. These advantages, however, do not come without new challenges; the identification of these challenges and the development of tools to overcome them are the main result of our work.

The first and most elementary result of our study is the (hardly surprising) observation that there exist no 'one size fits all' criterion for refinement that automatically handles all possible physical systems. Many of the challenges, however, can be effectively addressed with a combination of a small number of criteria for tagging and refining regions of the domain. We summarize these challenges and techniques as follows.

(a) In AMR it is more difficult to test for (and obtain) convergence, because of the loss of direct control over the resolution in a given region of spacetime. While the refinement in AMR is every bit as deterministic as it is in a box-in-a-box approach, the complexity of the underlying algorithm makes it practically impossible for a user to predict when, if and where refinement will take place. Consider for example the convergence analysis of a simulation using the truncation-error based tagging criterion of (47); in some regions of the spacetime a low resolution run may encounter a sufficiently large truncation error to trigger refinement whereas a higher-resolution run will not. To counteract this effect, one may adjust the tagging threshold in anticipation of the reduction in the truncation error, but some experimentation is often necessary because different ingredients of the code have different orders of accuracy. Additionally, one may enforce refinement using *a priori* knowledge, as for example, through enforced tagging around the spheres of wave extraction. An alternative approach would be to record the grid structure over time for one simulation (e.g. the lowest resolution run) and then 'replay' this grid structure (or as close as possible to it) for simulations at different resolutions as is done for the Had code [137].

(b) A further challenge arises from the use of too many refinement regions/boundaries over a small volume in spacetime. The interpolation at refinement boundaries is prone to generating small levels of high-frequency numerical noise that may bounce off neighbouring boundaries if these are too close in space (or time). An effective way to handle this problem is the use of buffer zones in space and to avoid unnecessarily frequent regridding.

(c) In the case of BH simulations, we often observe a degradation of numerical accuracy when refinement levels cross or even exist close to the apparent horizon. This typically manifests itself as an unphysical drift in the horizon area and, in the case of binaries, a loss of phase accuracy and/or a drift in the BH trajectory. These problems can be cured by enforced tagging of all grid points inside the apparent horizon. In practice, we add an additional buffer zone to ensure all refinement boundaries are sufficiently far away from the apparent horizon(s).

(d) The Berger–Rigoutsos algorithm detailed in section 2.2.2 does not treat the x, y and z direction on exactly equal footing; the partitioning algorithm (cf figure 1) inevitably handles the coordinate directions in a specific order. This can lead to asymmetries in the refined grids even when the underlying spacetime region is symmetric. In some simulations of BHs, we noticed this to cause a loss in accuracy. A simple way to overcome this problem is to enforce a boxlike structure around BHs.

(e) A single tagging variable (or one of its spatial derivatives) may not always be suitable to achieve appropriate refinement throughout the course of an entire simulation; for example, this may be due to gauge dependence or dramatic changes in the dynamics of the physical evolution. GRChombo allows for tagging regions based on arbitrary functions of *multiple* variables and their derivatives to overcome problems of this kind.

In order to avoid the difficulties listed here, we often combine two or more tagging criteria. The efficacy of this approach is demonstrated in sections 4 and 5 where we present in detail AMR simulations of inspiraling BH binaries and stable as well as collapsing axion stars. By comparing the BH simulations with those from the box-in-a-box based Lean code, we demonstrate that with an appropriate choice of tagging criteria, AMR simulations reach the same accuracy and convergence as state-of-the-art BH binary codes using Cartesian grids. While AMR does not directly bestow major benefits on the modeling of vacuum BH binaries (and is typically more computationally expensive), it offers greater flexibility in generalizing these to BH spacetimes with scalar or vector fields, other forms of matter, or BHs of nearly fractal shape that can form in higher-dimensional collisions [36]. The simulations of rapidly oscillating or gravitationally collapsing scalar fields demonstrate GRChombo's capacity to evolve highly compact and dynamic matter configurations of this type.

Finally, we note the potential of AMR for hydrodynamic simulations. However, high-resolution shock capturing methods present qualitatively new challenges for AMR and we leave the investigation of this topic for future work.

Acknowledgments

We thank the rest of the GRChombo collaboration (www.grchombo.org) for their support and code development work, and for helpful discussions regarding their use of AMR.

MR acknowledges support from a Science and Technology Facilities Council (STFC) studentship. KC acknowledges funding from the European Research Council (ERC) under the

European Unions Horizon 2020 research and innovation programme (Grant Agreement No. 693024), and an STFC Ernest Rutherford Fellowship. AD is supported by a Junior Research Fellowship (JRF) at Homerton College, University of Cambridge. PF is supported by the European Research Council Grant No. ERC-2014-StG 639022-NewNGR, and by a Royal Society University Research Fellowship Grant Nos. UF140319, RGF\EA\180260, URF\R\201026 and RF\ERE\210291. JCA acknowledges funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (Grant Agreement No. 693024), from the Beecroft Trust and from The Queen's College via an extraordinary Junior Research Fellowship (eJRF). TF is supported by a Royal Society Enhancement Award (Grant No. RGF\EA\180260). TH is supported by NSF Grant Nos. PHY-1912550, AST-2006538, PHY-090003 and PHY-20043, and NASA Grant Nos. 17-ATP17-0225, 19-ATP19-0051 and 20-LPS20-0011. JLR and US are supported by STFC Research Grant No. ST/V005669/1; US is supported by the H2020-ERC-2014-CoG Grant 'MaGRaTh' No. 646597.

The simulations presented in this paper used DiRAC resources under the projects ACSP218, ACTP186 and ACTP238. The work was performed using the Cambridge Service for Data Driven Discovery (CSD3), part of which is operated by the University of Cambridge Research Computing on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The DiRAC component of CSD3 was funded by BEIS capital funding via STFC capital Grants ST/P002307/1 and ST/cqgac6fa9bib2452/1 and STFC operations Grant ST/cqgac6fa9bib689X/1. In addition we have used the DiRAC at Durham facility managed by the Institute for Computational Cosmology on behalf of the STFC DiRAC HPC Facility (www.dirac.ac.uk). The equipment was funded by BEIS capital funding via STFC capital Grants ST/P002293/1 and ST/cqgac6fa9bib2371/1, Durham University and STFC operations Grant ST/cqgac6fa9bib0832/1. DiRAC is part of the National e-Infrastructure. XSEDE resources under Grant No. PHY-090003 were used on the San Diego Supercomputing Center's clusters Comet and Expanse and the Texas Advanced Supercomputing Center's (TACC) Stampede2. Furthermore, we acknowledge the use of HPC resources on the TACC Frontera cluster [138]. PRACE resources under Grant No. 2020225359 were used on the GCS Supercomputer JUWELS at the Jülich Supercomputing Center (JCS) through the John von Neumann Institute for Computing (NIC), funded by the Gauss Center for Supercomputing e.V. (www.gauss-centre.eu). The Fawcett supercomputer at the Department of Applied Mathematics and Theoretical Physics (DAMTP), University of Cambridge was also used, funded by STFC Consolidated Grant ST/P000673/1.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Appendix A. Calculation of the Weyl scalar Ψ_4

The Weyl tensor [139] in four spacetime dimensions is defined by

$$C_{\mu\nu\rho\sigma} := {}^{(4)}R_{\mu\nu\rho\sigma} - \left(g_{\mu[\rho}\,{}^{(4)}R_{\sigma]\nu} - g_{\nu[\rho}\,{}^{(4)}R_{\sigma]\mu}\right) + \frac{1}{3}g_{\mu[\rho}g_{\sigma]\nu}\,{}^{(4)}R. \tag{A.1}$$

It is completely determined by its electric and magnetic parts [140]

$$E_{\mu\nu} := n^{\alpha}n^{\beta}C_{\alpha\mu\beta\nu}, \tag{A.2}$$

$$B_{\mu\nu} := n^\alpha n^\beta (*C)_{\alpha\mu\beta\nu}, \tag{A.3}$$

where the dual Weyl tensor $({}^*C)_{\mu\nu\rho\sigma}$ is given by

$$(*C)_{\mu\nu\rho\sigma} : = \frac{1}{2}\epsilon^{\alpha\beta}{}_{\rho\sigma} C_{\mu\nu\alpha\beta}, \tag{A.4}$$

and $\epsilon_{\mu\nu\rho\sigma}$ is the volume form. Because of the symmetries of the Weyl tensor, the electric and magnetic parts are symmetric, trace-free and purely spatial.

In the Newman–Penrose formalism [108], one introduces a complex null tetrad $(l^\mu, k^\mu, m^\mu, \bar{m}^\mu)$, where we follow the notation of [26] in order to avoid confusion with the normal n^μ to the foliation (3). The Newman–Penrose, or Weyl, scalar Ψ_4 is defined by

$$\Psi_4 := C_{\alpha\beta\gamma\delta} k^\alpha \bar{m}^\beta k^\gamma \bar{m}^\delta, \tag{A.5}$$

which can be shown to reduce to [26]

$$\Psi_4 := (E_{ij} - \mathrm{i}B_{ij})\bar{m}^i \bar{m}^j. \tag{A.6}$$

We use the approach described in step (a) of section 5.1 of [141] to construct a null tetrad with the inner products

$$-l_\alpha k^\alpha = m_\alpha \bar{m}^\alpha = 1, \tag{A.7}$$

and all others vanishing. Following [18, 142], we omit the null rotations in order to bring the tetrad into a quasi-Kinnersley form (step (b) in section 5.1 of [141]). The expressions for the electric and magnetic parts of the Weyl scalar in the 3 + 1 Z4 formulation are

$$E_{ij} = \left[R_{ij} - K_i^m K_{jm} + K_{ij}(K - \Theta) - 4\pi S_{ij} + D_{(i}\Theta_{j)}\right]^{\mathrm{TF}}, \tag{A.8}$$

$$B_{ij} = \epsilon_{mn(i} D^m K_{j)}{}^n, \tag{A.9}$$

where $\epsilon_{ijk} = n^\alpha \epsilon_{\alpha ijk}$ is the volume form on the hypersurface. Note that unlike the usual 3 + 1 expressions, for example (8.3.15) and (8.3.16) in [26], these expressions are manifestly symmetric and trace-free.

Appendix B. Spatial derivative stencils

We use the formulae in [12] for the fourth order stencils. Using the conventional notation for finite differences where,

$$F_i = F|_{x=x_i}, \qquad F_{i,j} = F|_{x=x_i, y=y_j}, \tag{B.1}$$

and x_i and y_i are coordinates of the discrete points on a uniform grid, the centered stencils are

$$\partial_x F = \frac{1}{12h}\left(F_{i-2} - 8F_{i-1} + 8F_{i+1} - F_{i+2}\right), \tag{B.2}$$

$$\partial_x^2 F = \frac{1}{12h^2}\left(-F_{i-2} + 16F_{i-1} - 30F_i + 16F_{i+1} - F_{i+2}\right), \tag{B.3}$$

$$\partial_{xy}^2 F = \frac{1}{144h^2}\big(F_{i-2,j-2} - 8F_{i-2,j-1} + 8F_{i-2,j+1} - F_{i-2,j+2} - 8F_{i-1,j-2}$$
$$+ 64F_{i-1,j-1} - 64F_{i-1,j+1} + 8F_{i-1,j+2} + 8F_{i+1,j-2} - 64F_{i+1,j-1}$$

$$
\begin{aligned}
&+ 64F_{i+1,j+1} - 8F_{i+1,j+2} - F_{i+2,j-2} + 8F_{i+2,j-1} - 8F_{i+2,j+1} \\
&+ F_{i+2,j+2}\big) ,
\end{aligned} \tag{B.4}
$$

and, for the advection term, the lopsided stencils are

$$
\partial_x F = \frac{1}{12h}\left(-3F_{i-1} - 10F_i + 18F_{i+1} - 6F_{i+2} + F_{i+3}\right) \quad \text{if } \beta^x > 0, \tag{B.5}
$$

$$
\partial_x F = \frac{1}{12h}\left(-F_{i-3} + 6F_{i-2} - 18F_{i-1} + 10F_i + 3F_{i+1}\right) \quad \text{if } \beta^x \leqslant 0. \tag{B.6}
$$

We follow [100] for the sixth order stencils. The centered stencils are

$$
\partial_x F = \frac{1}{60h}\left(-F_{i-3} + 9F_{i-2} - 45F_{i-1} + 45F_{i+1} - 9F_{i+2} + F_{i+3}\right) , \tag{B.7}
$$

$$
\begin{aligned}
\partial_x^2 F = \frac{1}{180h^2}\big(&2F_{i-3} - 27F_{i-2} + 270F_{i-1} - 490F_i + 270F_{i+1} - 27F_{i+2} \\
&+ 2F_{i+3}\big) ,
\end{aligned} \tag{B.8}
$$

$$
\begin{aligned}
\partial_{xy}^2 F = \frac{1}{3600h}\big(&F_{i-3.j-3} - 9F_{i-3,j-2} + 45F_{i-3,j-1} - 45F_{i-3.j+1} \\
&+ 9F_{i-3,j+2} - F_{i-3,j+3} - 9F_{i-2,j-3} + 81F_{i-2,j-2} \\
&- 405F_{i-2,j-1} + 405F_{i-2,j+1} - 81F_{i-2,j+2} + 9F_{i-2,j+3} \\
&+ 45F_{i-1,j-3} - 405F_{i-1,j-2} + 2025F_{i-1,j-1} - 2025F_{i-1,j+1} \\
&+ 405F_{i-1,j+2} - 45F_{i-1,j+3} - 45F_{i+1,j-3} + 405F_{i+1,j-2} \\
&- 2025F_{i+1,j-1} + 2025F_{i+1,j+1} - 405F_{i+1,j+2} + 45F_{i+1,j+3} \\
&+ 9F_{i+2,j-3} - 81F_{i+2,j-2} + 405F_{i+2,j-1} - 405F_{i+2,j+1} \\
&+ 81F_{i+2,j+2} - 9F_{i+2,j+3} - F_{i+3,j-3} + 9F_{i+3,j-2} \\
&- 45F_{i+3,j-1} + 45F_{i+3,j+1} - 9F_{i+3,j+2} + F_{i+3,j+3}\big) ,
\end{aligned} \tag{B.9}
$$

and, for the advection terms, the lopsided stencils are

$$
\begin{aligned}
\partial_x F = \frac{1}{60h}&\left(2F_{i-2} - 24F_{i-1} - 35F_i + 80F_{i+1} - 30F_{i+2} + 8F_{i+3} - F_{i+4}\right) \\
&\text{if } \beta^x > 0,
\end{aligned} \tag{B.10}
$$

$$
\begin{aligned}
\partial_x F = \frac{1}{60h}&\left(F_{i-4} - 8F_{i-3} + 30F_{i-2} - 80F_{i-1} + 35F_i + 24F_{i+1} - 2F_{i+2}\right) \\
&\text{if } \beta^x < 0.
\end{aligned} \tag{B.11}
$$

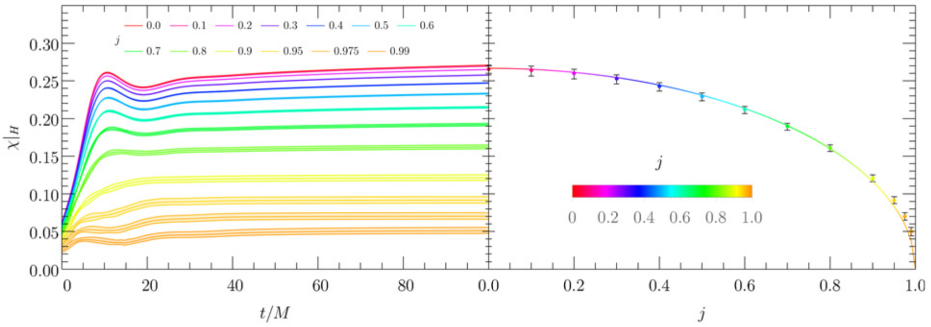


Figure C1. Plots illustrating the dependence of the value of the conformal factor χ on the apparent horizon surface H in the moving puncture gauge (28), (30) and (31) for different values of the dimensionless spin j. For all plots, we use the quasi-isotropic Kerr initial data described in section 2.1.2 and the default values of the gauge parameters (29) with $M\eta = 1$. Although we would expect the plots to vary for different gauge parameters (in particular, as η is varied), these plots provide a rough rule-of-thumb. The left panel shows the mean value of χ as a function of time with the error bands around each curve corresponding to the maximum and minimum on H. The right panel shows the mean value of χ over the interval $t/M \in [40, 100]$ for each j with the error bars corresponding to the minimum and maximum values of χ over the same interval. Furthermore, we show a fit of the mean value of χ against j which takes the form $\langle\chi\rangle|_H \simeq 0.2666\sqrt{1-j^2}$.

Appendix C. Approximate horizon locations as a tagging criteria

As noted in section 3, the use of the horizon location can be an essential part of an adaptive mesh scheme for refinement. In particular, one does not usually want to put additional refinement within a horizon (where effects are unobservable anyway), and should take care to avoid grid boundaries overlapping the horizon, since this can lead to instabilities that strongly affect the physical results. Whilst using an apparent horizon finder for this is a possibility, often a more ‘quick and dirty’ scheme using contours of the conformal factor can be just as effective, and significantly easier to implement. Whilst in principle there is a dependence on simulation and gauge parameters (in particular η [18]), in general the approximate values are quite robust.

The key dependence is on the (dimensionless) spin of the BH j, as illustrated in figure C1, with a good fit obtained from the relation

$$\langle\chi\rangle|_H = 0.2666\sqrt{1-j^2}. \tag{C.1}$$

One key advantage is that one does not need to know *a priori* the mass of the BH spacetime which forms, and it can be seen that simply using the $j = 0$ values will give a conservative coverage of the horizon. These types of criteria were used extensively in the higher dimensional black ring spacetimes studies in [36–39] and for the investigation into gravitational collapse in a modified gravity theory in [35].

Appendix D. Parallelization in GRChombo

Like other NR codes and, more generally, scientific computing codes, GRChombo exploits parallelization at several different levels in order to achieve good performance and scaling on modern supercomputers (see section 2.4 in [143] for scaling results).

For each AMR level, GRChombo splits the domain into boxes and these boxes are shared between processes running on multiple distributed-memory nodes using the message passing interface (MPI). In practice, even though the memory is shared within a node, we typically still use multiple MPI processes per node in order to achieve optimal performance. For example, if a node has n cores, we might choose to use between $n/4$–$n/2$ MPI processes per node. At every *regrid*, we use a load balancing routine in Chombo in order to evenly distribute the boxes. We sort the boxes using a Morton ordering, as this minimizes communication by increasing the chance that neighbouring boxes are on the same or nearby MPI processes.

One of the most common operations in an NR code is looping through all the cells/points on the grid, calculating some expression and then storing its value in a grid variable. An example is the calculation of the rhs at every RK4 substep which is often where a code spends a large proportion of its time. Within an MPI process, GRChombo uses OpenMP to thread these loops over the z and y coordinates of the boxes. For the x direction, GRChombo relies on SIMD (single instruction, multiple data) /vector intrinsics[30] in order to utilize the full vector-width of the targeted architecture. We use intrinsics because the complexity of the CCZ4 equations (13)–(18) means that compilers will usually fail to auto-vectorize these loops. The main disadvantage of using SIMD intrinsics is that they are complex and difficult to implement properly. In order to hide the technical implementation from users, many NR codes rely on code-generation scripts to convert more familiar Mathematica/Python expressions to optimized and vectorized Fortran/C/C++ code, for example, Kranc [144] and NRPy+ [70]. GRChombo takes a different approach, keeping the programming at the lower level but relying on C++14 templates to provide a somewhat more user-friendly interface for writing optimized code. Vectorized expressions can be enforced by replacing the C++ type `double` with a template type `data_t`, which represents a vector of values of the variables on the grid of arbitrary length (e.g. the value of χ at the points with x index $i_x = 0, 1, 2, 3, \ldots$, and constant y and z). In order to make this functionality work, the user is required to write their code in a *compute class* with a `compute` member function which can then be instantiated as an object and then passed to a loop function which calls the `compute` member function in each vector of cells. Multiple compute objects can be combined into a *compute pack* which can then be called by the loop function for added efficiency. For a more detailed description and examples, see section 2.5 in [143].

ORCID iDs

Miren Radia https://orcid.org/0000-0001-8861-2025
Ulrich Sperhake https://orcid.org/0000-0002-3134-7088
Amelia Drew https://orcid.org/0000-0001-8252-602X
Katy Clough https://orcid.org/0000-0001-8841-1522
Pau Figueras https://orcid.org/0000-0001-6438-315X
Eugene A Lim https://orcid.org/0000-0002-6227-9540
Justin L Ripley https://orcid.org/0000-0001-7192-0021
Josu C Aurrekoetxea https://orcid.org/0000-0001-9584-5791
Tiago França https://orcid.org/0000-0002-1718-151X
Thomas Helfer https://orcid.org/0000-0001-6880-1005

[30] For the x86_64 architecture, GRChombo currently supports SSE2, AVX and AVX-512 instructions.

References

[1] Abbott B P *et al* 2016 *Phys. Rev. Lett.* **116** 061102
[2] Pretorius F 2005 *Phys. Rev. Lett.* **95** 121101
[3] Campanelli M, Lousto C O, Marronetti P and Zlochower Y 2006 *Phys. Rev. Lett.* **96** 111101
[4] Baker J G, Centrella J, Choi D-I, Koppitz M and van Meter J 2006 *Phys. Rev. Lett.* **96** 111102
[5] Abbott B P *et al* (LIGO Scientific, Virgo) 2019 *Phys. Rev.* X **9** 031040
[6] Abbott R *et al* (LIGO Scientific, Virgo) 2021 *Phys. Rev.* X **11** 021053
[7] Abbott R *et al* 2021 GWTC-3: compact binary coalescences observed by LIGO and Virgo during the second part of the third observing run (arXiv:2111.03606)
[8] Goodale T, Allen G, Lanfermann G, Massó J, Radke T, Seidel E and Shalf J 2003 The CACTUS framework and toolkit: design and applications *Vector and Parallel Processing—VECPAR'2002, 5th International Conference* (Lecture Notes in Computer Science) (Berlin: Springer) http://edoc.mpg.de/3341
[9] Löffler F *et al* 2012 *Class. Quantum Grav.* **29** 115001
[10] Schnetter E, Hawley S H and Hawke I 2004 *Class. Quantum Grav.* **21** 1465–88
[11] Brown J, Diener P, Sarbach O, Schnetter E and Tiglio M 2009 *Phys. Rev.* D **79** 044023
[12] Zlochower Y, Baker J, Campanelli M and Lousto C 2005 *Phys. Rev.* D **72** 024021
[13] Herrmann F, Hinder I, Shoemaker D and Laguna P 2007 *Class. Quantum Grav.* **24** S33–42
[14] Sperhake U 2007 *Phys. Rev.* D **76** 104015
[15] Zilhao M, Witek H, Sperhake U, Cardoso V, Gualtieri L, Herdeiro C and Nerozzi A 2010 *Phys. Rev.* D **81** 084052
[16] Witek H, Zilhao M, Bozzola G, Elley M, Ficarra G, Ikeda T, Sanchis-Gual N and Silva H 2021 Canuda: a public numerical relativity library to probe fundamental physics https://doi.org/10.5281/zenodo.3565474
[17] Brügmann B 1996 *Phys. Rev.* D **54** 7361–72
[18] Bruegmann B, Gonzalez J A, Hannam M, Husa S, Sperhake U and Tichy W 2008 *Phys. Rev.* D **77** 024027
[19] Thierfelder M, Bernuzzi S and Bruegmann B 2011 *Phys. Rev.* D **84** 044012
[20] Marronetti P, Tichy W, Brügmann B, González J, Hannam M, Husa S and Sperhake U 2007 *Class. Quantum Grav.* **24** S43–58
[21] Abbott B P *et al* KAGRA, LIGO Scientific, Virgo) 2018 *Living Rev. Relativ.* **21** 3
[22] Szilágyi B 2014 *Int. J. Mod. Phys.* D **23** 1430014
[23] Kumar P *et al* 2014 *Phys. Rev.* D **89** 042002
[24] Boyle M *et al* 2019 *Class. Quantum Grav.* **36** 195006
[25] Szilágyi B, Blackman J, Buonanno A, Taracchini A, Pfeiffer H P, Scheel M A, Chu T, Kidder L E and Pan Y 2015 *Phys. Rev. Lett.* **115** 031102
[26] Alcubierre M 2008 *Introduction to 3 + 1 Numerical Relativity* (*International Series of Monographs on Physics*) (Oxford: Oxford University Press)
[27] Baumgarte T W and Shapiro S L 2010 *Numerical Relativity: Solving Einstein's Equations on the Computer* (Cambridge: Cambridge University Press)
[28] Kravtsov A V, Klypin A A and Khokhlov A M 1997 *Astrophys. J. Suppl.* **111** 73
[29] Teyssier R 2002 *Astron. Astrophys.* **385** 337–64
[30] Barrera-Hinojosa C and Li B 2020 *J. Cosmol. Astropart. Phys.* JCAP01(2020)007
[31] Berger M J and Oliger J 1984 *J. Comput. Phys.* **53** 484–512
[32] East W E 2019 *Phys. Rev. Lett.* **122** 231103
[33] Pretorius F and East W E 2018 *Phys. Rev.* D **98** 084053
[34] Lehner L and Pretorius F 2010 *Phys. Rev. Lett.* **105** 101102
[35] Figueras P and França T 2020 *Class. Quantum Grav.* **37** 225009
[36] Andrade T, Figueras P and Sperhake U 2022 *J. High Energy Phys.* JHEP03(2022)111
[37] Bantilan H, Figueras P, Kunesch M and Panosso Macedo R 2019 *Phys. Rev.* D **100** 086014
[38] Figueras P, Kunesch M, Lehner L and Tunyasuvunakool S 2017 *Phys. Rev. Lett.* **118** 151103
[39] Figueras P, Kunesch M and Tunyasuvunakool S 2016 *Phys. Rev. Lett.* **116** 071102
[40] Drew A and Shellard E P S 2022 *Phys. Rev.* D **105** 063517
[41] Helfer T, Aurrekoetxea J C and Lim E A 2019 *Phys. Rev.* D **99** 104028
[42] Aurrekoetxea J C, Helfer T and Lim E A 2020 *Class. Quantum Grav.* **37** 204001
[43] East W E, Kleban M, Linde A and Senatore L 2016 *J. Cosmol. Astropart. Phys.* JCAP09(2016)010

[44] Cook W G, Glushchenko I A, Ijjas A, Pretorius F and Steinhardt P J 2020 *Phys. Lett.* B **808** 135690
[45] Ijjas A, Cook W G, Pretorius F, Steinhardt P J and Davies E Y 2020 *J. Cosmol. Astropart. Phys.* JCAP08(2020)030
[46] Ijjas A, Sullivan A P, Pretorius F, Steinhardt P J and Cook W G 2021 *J. Cosmol. Astropart. Phys.* JCAP06(2021)013
[47] Joana C and Clesse S 2021 *Phys. Rev.* D **103** 083501
[48] Aurrekoetxea J C, Clough K, Flauger R and Lim E A 2020 *J. Cosmol. Astropart. Phys.* JCAP05(2020)030
[49] Clough K, Flauger R and Lim E A 2018 *J. Cosmol. Astropart. Phys.* JCAP05(2018)065
[50] Clough K, Lim E A, DiNunno B S, Fischler W, Flauger R and Paban S 2017 *J. Cosmol. Astropart. Phys.* JCAP09(2017)025
[51] de Jong E, Aurrekoetxea J C and Lim E A 2022 *J. Cosmol. Astropart. Phys.* JCAP03(2022)029
[52] Andrade T *et al* 2021 *J. Open Source Softw.* **6** 3703
[53] Clough K, Figueras P, Finkel H, Kunesch M, Lim E A and Tunyasuvunakool S 2015 *Class. Quantum Grav.* **32** 24
[54] Pretorius F 2005 *Class. Quantum Grav.* **22** 425–51
[55] Pretorius F and Choptuik M W 2006 *J. Comput. Phys.* **218** 246–74
[56] East W E, Pretorius F and Stephens B C 2012 *Phys. Rev.* D **85** 124010
[57] MacNeice P, Olson K M, Mobarry C, de Fainchtein R and Packer C 2000 *Comput. Phys. Commun.* **126** 330
[58] Anderson M, Hirschmann E W, Liebling S L and Neilsen D 2006 *Class. Quantum Grav.* **23** 6503–24
[59] Pfeiffer H P, Kidder L E, Scheel M A and Teukolsky S A 2003 *Comput. Phys. Commun.* **152** 253–73
[60] Hilditch D, Weyhausen A and Brügmann B 2016 *Phys. Rev.* D **93** 063006
[61] Meringolo C, Servidio S and Veltri P 2021 *Class. Quantum Grav.* **38** 075027
[62] Mertens J B, Giblin J T and Starkman G D 2016 *Phys. Rev.* D **93** 124059
[63] Palenzuela C, Miñano B, Viganò D, Arbona A, Bona-Casas C, Rigo A, Bezares M, Bona C and Massó J 2018 *Class. Quantum Grav.* **35** 185007
[64] Hornung R D and Kohn S R 2002 *Concurr. Comput.: Pract. Exp.* **14** 347–68
[65] Gunney B T N and Anderson R W 2016 *J. Parallel Distrib. Comput.* **89** 65–84
[66] Elliott N S, Anderson R W, Arrighi B and Gunney B T N SAMRAI project website 2021 https://computing.llnl.gov/projects/samrai
[67] Daszuta B, Zappa F, Cook W, Radice D, Bernuzzi S and Morozova V 2021 *Astrophys. J. Supp.* **257** 25
[68] Cheong P C-K, Lam A T-L, Ng H H-Y and Li T G F 2021 *Mon. Not. R. Astron. Soc.* **508** 2279–301
[69] Fernando M, Neilsen D, Lim H, Hirschmann E and Sundar H 2019 *SIAM J. Sci. Comput.* **41** C97–C138
[70] Ruchlin I, Etienne Z B and Baumgarte T W 2018 *Phys. Rev.* D **97** 064036
[71] Deppe N *et al* 2021 Spectre https://doi.org/10.5281/zenodo.4734670
[72] Kidder L E *et al* 2017 *J. Comput. Phys.* **335** 84–114
[73] Cao Z, Fu P, Ji L W and Xia Y 2019 *Int. J. Mod. Phys.* D **28** 1950014
[74] Ji L W, Cai R G and Cao Z 2018 Binary black hole simulation with an adaptive finite element method + III: evolving a single black hole (arXiv:1805.10642)
[75] Sperhake U 2015 *Class. Quantum Grav.* **32** 124011
[76] Adams M *et al* 2019 Chombo software package for AMR applications—design document *Technical Report LBNL-6616E* Lawrence Berkeley National Laboratory https://commons.lbl.gov/download/attachments/73468344/chomboDesign.pdf?version=1 & modificationDate=1554672305006 & api=v2
[77] Berger M and Rigoutsos I 1991 *IEEE Trans. Syst. Man. Cybern.* **21** 1278–86
[78] Alic D, Bona-Casas C, Bona C, Rezzolla L and Palenzuela C 2012 *Phys. Rev.* D **85** 064040
[79] Alic D, Kastaun W and Rezzolla L 2013 *Phys. Rev.* D **88** 064049
[80] Baker J G and van Meter J R 2005 *Phys. Rev.* D **72** 104010
[81] Arnowitt R, Deser S and Misner C W 1962 The dynamics of general relativity *Gravitation an Introduction to Current Research* ed L Witten (New York: Wiley) pp 227–65
[82] York J W Jr 1979 Kinematics and dynamics of general relativity *Sources of Gravitational Radiation* ed L Smarr (Cambridge: Cambridge University Press) pp 83–126
[83] Bona C, Massó J, Seidel E and Stela J 1995 *Phys. Rev. Lett.* **75** 600–3

[84] Frittelli S 1997 *Phys. Rev.* D **55** 5992–6

[85] Ansorg M, Bruegmann B and Tichy W 2004 *Phys. Rev.* D **70** 064011

[86] Brandt S and Brügmann B 1997 *Phys. Rev. Lett.* **78** 3606–9

[87] Bowen J M and York J W 1980 *Phys. Rev.* D **21** 2047–56

[88] Paschalidis V, Etienne Z B, Gold R and Shapiro S L 2013 An efficient spectral interpolation routine for the TwoPunctures code (arXiv:1304.0457)

[89] Dennison K A, Baumgarte T W and Pfeiffer H P 2006 *Phys. Rev.* D **74** 064016

[90] Liu Y T, Etienne Z B and Shapiro S L 2009 *Phys. Rev.* D **80** 121503

[91] Nazari Z, Cicoli M, Clough K and Muia F 2021 *J. Cosmol. Astropart. Phys.* JCAP05(2021)027

[92] Muia F, Cicoli M, Clough K, Pedro F, Quevedo F and Vacca G P 2019 *J. Cosmol. Astropart. Phys.* JCAP07(2019)044

[93] Widdicombe J Y, Helfer T and Lim E A 2020 *J. Cosmol. Astropart. Phys.* JCAP01(2020)027

[94] Widdicombe J Y, Helfer T, Marsh D J E and Lim E A 2018 *J. Cosmol. Astropart. Phys.* JCAP10(2018)005

[95] Clough K, Dietrich T and Niemeyer J C 2018 *Phys. Rev.* D **98** 083020

[96] Dietrich T, Ossokine S and Clough K 2019 *Class. Quantum Grav.* **36** 025002

[97] Helfer T, Lim E A, Garcia M A G and Amin M A 2019 *Phys. Rev.* D **99** 044046

[98] Helfer T, Marsh D J E, Clough K, Fairbairn M, Lim E A and Becerril R 2017 *J. Cosmol. Astropart. Phys.* JCAP03(2017)055

[99] Kreiss H and Oliger J 1973 *Methods for the Approximate Solution of Time Dependant Problems* (*GARP Publication Series No. 10*) (Geneva: WMO) https://library.wmo.int/index.php?lvl=notice_display & id=6937

[100] Husa S, González J A, Hannam M, Brügmann B and Sperhake U 2008 *Class. Quantum Grav.* **25** 105006

[101] Berger M J and Colella P 1989 *J. Comput. Phys.* **82** 64–84

[102] McCorquodale P and Colella P 2011 *Commun. Appl. Math. Comput. Sci.* **6** 1–25

[103] Bondarescu M 2006 Orbifolds in numerical relativity (arXiv:gr-qc/0603047)

[104] Bamber J, Clough K, Ferreira P G, Hui L and Lagos M 2021 *Phys. Rev.* D **103** 044059

[105] Traykova D, Clough K, Helfer T, Berti E, Ferreira P G and Hui L 2021 *Phys. Rev.* D **104** 103014

[106] Clough K 2021 *Class. Quantum Grav.* **38** 167001

[107] Fornberg B 1998 *SIAM Rev.* **40** 685–91

[108] Newman E and Penrose R 1962 *J. Math. Phys.* **3** 566–78

[109] Poisson S D 1827 *Mémoires de l'Académie Royale des sciences de l'Institut de France.* **T.6** 571–602

[110] Hawking S and Ellis G 1973 *The Large Scale Structure of Space–Time* (*Cambridge Monographs on Mathematical Physics*) (Cambridge: Cambridge University Press) https://books.google.co.uk/books?id=QagG_KI7Ll8C

[111] Pretorius F 2002 Numerical simulations of gravitational collapse *PhD Thesis* University of British Columbia

[112] Khan S 2019 private communications

[113] Radia M, Sperhake U, Berti E and Croft R 2021 *Phys. Rev.* D **103** 104006

[114] Sperhake U, Rosca-Mead R, Gerosa D and Berti E 2020 *Phys. Rev.* D **101** 024044

[115] Gonzalez J A, Hannam M D, Sperhake U, Bruegmann B and Husa S 2007 *Phys. Rev. Lett.* **98** 091101

[116] Campanelli M, Lousto C O, Zlochower Y and Merritt D 2007 *Phys. Rev. Lett.* **98** 231102

[117] Campanelli M, Lousto C, Zlochower Y and Merritt D 2007 *Astrophys. J.* **659** L5–8

[118] Nakamura T, Oohara K and Kojima Y 1987 *Prog. Theor. Phys. Suppl.* **90** 1–218

[119] Shibata M and Nakamura T 1995 *Phys. Rev.* D **52** 5428–44

[120] Baumgarte T W and Shapiro S L 1998 *Phys. Rev.* D **59** 024007

[121] Thornburg J 1996 *Phys. Rev.* D **54** 4899–918

[122] Thornburg J 2004 *Class. Quantum Grav.* **21** 743–66

[123] Reisswig C and Pollney D 2011 *Class. Quantum Grav.* **28** 195015

[124] Damour T, Iyer B R and Sathyaprakash B S 1998 *Phys. Rev.* D **57** 885–907

[125] Miller M A 2005 *Phys. Rev.* D **71** 104016

[126] Owen B J 1996 *Phys. Rev.* D **53** 6749–61

[127] Lindblom L, Owen B J and Brown D A 2008 *Phys. Rev.* D **78** 124020

[128] Ruiz M, Alcubierre M, Núñez D and Takahashi R 2008 *Gen. Relativ. Gravit.* **40** 2467

[129] Barsotti L, Fritschel P, Evans M and Gras S 2018 Updated Advanced LIGO sensitivity design curve *Technical Report T1800044-v5* The LIGO Scientific Collaboration https://dcc.ligo.org/LIGO-T1800044/public

[130] Shoemaker D 2010 Advanced LIGO anticipated sensitivity curves *Technical Report T0900288-v3* The LIGO Scientific Collaboration https://dcc.ligo.org/LIGO-T0900288/public

[131] Alcubierre M, Becerril R, Guzman F S, Matos T, Nunez D and Urena-Lopez L A 2003 *Class. Quantum Grav.* **20** 2883–903

[132] Peccei R D and Quinn H R 1977 *Phys. Rev. Lett.* **38** 1440–3

[133] Weinberg S 1978 *Phys. Rev. Lett.* **40** 223–6

[134] Marsh D J E 2016 *Phys. Rep.* **643** 1–79

[135] Urena-Lopez L A, Matos T and Becerril R 2002 *Class. Quantum Grav.* **19** 6259–77

[136] Urena-Lopez L A 2002 *Class. Quantum Grav.* **19** 2617–32

[137] Liebling S L 2004 *Class. Quantum Grav.* **21** 3995

[138] Stanzione D, West J, Evans R T, Minyard T, Ghattas O and Panda D K 2020 Frontera: the evolution of leadership computing at the national science foundation *Practice and Experience in Advanced Research Computing PEARC '20* (New York: Association for Computing Machinery) pp 106–11

[139] Weyl H 1918 *Math. Z.* **2** 384–411

[140] Stephani H 2004 *Relativity: An Introduction to Special and General Relativity* (Cambridge: Cambridge University Press)

[141] Baker J G, Campanelli M and Lousto C O 2002 *Phys. Rev.* D **65** 044001

[142] Fiske D R, Baker J G, van Meter J R, Choi D I and Centrella J M 2005 *Phys. Rev.* D **71** 104036

[143] Kunesch M 2018 Numerical simulations of instabilities in general relativity *PhD Thesis* University of Cambridge

[144] Husa S, Hinder I and Lechner C 2006 *Comput. Phys. Commun.* **174** 983–1004