

PAPER

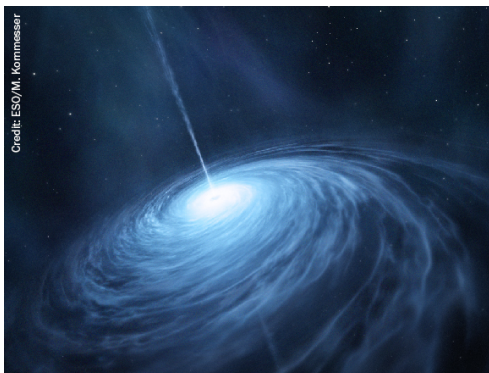
Axisymmetric hydrodynamics in numerical relativity using a multipatch method

To cite this article: Jerred Jesse *et al* 2020 *Class. Quantum Grav.* **37** 235010

View the [article online](#) for updates and enhancements.

Recent citations

- [Comparison of momentum transport models for numerical relativity](#)
Matthew D. Duez *et al*







AMERICAN
ASTRONOMICAL
SOCIETY

IOP | ebooks™

Your first choice for astronomy, astrophysics,
solar physics, and planetary science ebooks.

Start exploring the collection—download the
first chapter of every title for free.

Axisymmetric hydrodynamics in numerical relativity using a multipatch method

Jerred Jesse¹ , Matthew D Duez^{1,*} , Francois Foucart²,
Milad Haddadi¹ , Alexander L Knight², Courtney L
Cadenhead¹, Francois Hébert³, Lawrence E Kidder⁴ ,
Harald P Pfeiffer⁵ and Mark A Scheel³

¹ Department of Physics and Astronomy, Washington State University, Pullman, Washington 99164, United States of America

² Department of Physics and Astronomy, University of New Hampshire, 9 Library Way, Durham NH 03824, United States of America

³ TAPIR, Walter Burke Institute for Theoretical Physics, MC 350-17, California Institute of Technology, Pasadena, California 91125, United States of America

⁴ Center for Radiophysics and Space Research, Cornell University, Ithaca, New York 14853, United States of America

⁵ Max-Planck-Institut für Gravitationsphysik, Albert-Einstein-Institut, D-14476 Golm, Germany

E-mail: m.duez@wsu.edu

Received 6 July 2020, revised 9 September 2020

Accepted for publication 29 September 2020

Published 10 November 2020



CrossMark

Abstract

We describe a method of implementing the axisymmetric evolution of general-relativistic hydrodynamics and magnetohydrodynamics through modification of a multipatch grid scheme. In order to ease the computational requirements required to evolve the post-merger phase of systems involving binary compact massive objects in numerical relativity, it is often beneficial to take advantage of these system's tendency to rapidly settle into states that are nearly axisymmetric, allowing for 2D evolution of secular timescales. We implement this scheme in the spectral Einstein code and show the results of application of this method to four test systems including viscosity, magnetic fields, and neutrino radiation transport. Our results show that this method can be used to quickly allow already existing 3D infrastructure that makes use of local coordinate system transformations to be made to run in axisymmetric 2D with the flexible grid creation capabilities of multipatch methods. Our code tests include a simple model of a binary neutron star postmerger remnant, for which we confirm the formation of a massive torus which is a promising source of post-merger ejecta.

* Author to whom any correspondence should be addressed.

Keywords: numerical relativity, relativistic hydrodynamics, neutrino transport, compact binary postmerger evolution

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

1. Introduction

The detection of the gravitational wave signal resulting from the merger binary black hole systems by the LIGO and VIRGO collaborations [1–7] along with the detection of simultaneous electromagnetic and gravitational wave signals from binary neutron star mergers [8–11], and the corresponding need for theoretical predictions with which to compare them, has given renewed urgency to the goal of accurately modeling these systems throughout the merger process. For systems involving at least one neutron star, it is the post-merger state that is primarily responsible for the observable electromagnetic signals. Modeling of these systems through numerical relativity simulations provides critical insight into the dependencies of the signals on binary parameters and nuclear physics. Unfortunately, running these simulations in the high-resolution required to get accurate predictions can present large computational resource barriers in simulated time or size scales. However, the post-merger environment has a useful property: by taking advantage of these systems' tendency to approach an axisymmetric state, we can ease the computational resources required to simulate these systems over extended scales of both time and space. Although the dynamical timescales of remnant neutron stars and accretion disks, of the order \sim ms at most, are reasonably accessible to 3D simulations, secular effects that drive the subsequent evolution can operate on much longer timescales. Particularly important are angular momentum transport effects that can act on a wide range of timescales of up to hundreds of milliseconds [12, 13], and neutrino cooling effects that operate on timescales of up to several seconds [14].

The use of axisymmetry in numerical relativity simulations has been explored by several groups. This typically involves evolving Einstein's equations using the cartoon method [15] while evolving hydrodynamics by writing the relevant equations in a cylindrical coordinate system [16–18]. The cartoon method does involve some loss of accuracy due to interpolations required in the method, and some effort has been made to avoid these [19]. Additionally, evolution problems due to the coordinate singularities that arise from the use of polar coordinate systems have been avoided by use of a reference metric [20–22]. Methods also exist that help with issues of spatial resolution on large scales, such as adaptive mesh refinement [19, 23], which is able to concentrate resolution where it is most needed, while in most cases still building the grid from Cartesian domains. In multipatch methods [24–29], one introduces coordinate patches, each with its own local coordinate system in which it takes a simple shape (e.g. a Cartesian block), but which can be deformed in the global coordinate system and fit together into a grid to match the geometry of the problem. A number of methods used in numerical relativity not usually called 'multipatch' have local coordinate systems and therefore fit into this general category, including the multidomain pseudospectral sector of the spectral Einstein code (SpEC) [30] and the multielement discontinuous Galerkin methods [31, 32] which many hope will form the basis of the next generation of numerical relativity codes.

In this paper, we describe a method of implementing the axisymmetric evolution of the general-relativistic equations of ideal radiation hydrodynamics and magnetohydrodynamics through modification of a multipatch grid scheme, applicable to any method using the local patch coordinates framework, which we implement in the SpEC [30]. While other codes for

carrying out axisymmetric relativistic hydrodynamics evolutions exist, there are several notable new features of our methods and results. First, using the multipatch framework, we automate the conversion to an axisymmetry-friendly coordinate system, which we demonstrate by using the same recipe to make working axisymmetric 2D versions of our relativistic 3D hydrodynamics, magnetohydrodynamics, neutrino transport, and shear viscosity codes. We point out that our neutrino transport method, which evolves number density as well as energy density, is somewhat different from other gray M1 schemes, so this is the first time this particular system has been converted to 2D axisymmetry. Second, we inherit the flexibility of multipatch methods to construct grids from patches of different shapes to optimally match the geometry of a problem. As a demonstration of this, we present the 2D evolution of a viscous differentially rotating star using a combination of square patches for the stellar interior and circular wedges for the outflow zone. As well as serving as a code test, this viscous rotating star system is of great astrophysical interest because it is a reasonable model of the remnant of a binary neutron star merger. We evolve it using a different subgrid momentum transport model than has been applied to it in prior work [33], providing an important qualitative check on the previous results. Finally, we present several minor enhancements of the SpEC–Hydro code, including a generalization of our auxiliary entropy evolution [29] to the thermal gamma-law class of equations of state, an altered form of the divergence cleaning magnetohydrodynamics equations, and an altered treatment of neutrino fluxes in the optically thick limit.

This paper is organized as follows. In section 2, we describe the evolution equations for our (magneto)hydrodynamic variables and the application of our axisymmetry method to them. In section 3 several tests of this axisymmetry method are presented: a stationary Tolman–Oppenheimer–Volkoff (TOV) star, a viscous differentially rotating star, a magnetized accretion disk, and neutrino radiation in a spherically symmetric supernova collapse profile, each showing good agreement with 3D results or previous axisymmetric simulations. Concluding remarks are given in section 4, where we summarize our results and discuss future plans.

2. Formulation

2.1. Evolution equations

We use SpEC to evolve Einstein’s equations and the general relativistic equations of ideal radiation (magneto)hydrodynamics. SpEC evolves Einstein’s equations and the general relativistic hydrodynamics equations on two separate computational grids. A multidomain grid of collocation points is used to evolve Einstein’s equations pseudospectrally in a generalized harmonic formulation [34] while the general relativistic (magneto)hydrodynamics equations in conservative form are evolved on a finite difference grid. The finite difference grid uses an HLL approximate Riemann solver [35]. Reconstruction of values at cell faces from their cell-average values is done using a high-order shock capturing method, a fifth-order WENO scheme [36, 37]. Time evolution is performed using a third-order Runge–Kutta algorithm with an adaptive time-stepper which compares integration errors to specified absolute and relative thresholds, lowering the timestep if errors exceed thresholds, increasing the timestep if they are all below thresholds. At the end of each time step any necessary source term information is then communicated between the two grids, using a third-order accurate spatial interpolation scheme [38]. Both grids are forced to evolve with the same timestep, determined by the grid which requires the shorter step. Interpolation between grids is carried out once per step, and second-order interpolation in time is used for source terms during Runge–Kutta substeps.

The following sections make use of the 3 + 1 decomposition of the spacetime metric

$$ds^2 = g_{\alpha\beta} dx^\alpha dx^\beta \quad (1)$$

$$= -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt), \quad (2)$$

where α is the lapse, β^i the shift, and γ_{ij} is the three-metric on a spacelike hypersurface of constant coordinate t . The three-metric is the projection onto spatial hypersurfaces of the four-metric:

$$\gamma_{ij} = g_{ij} + n_i n_j, \quad (3)$$

where $n_\mu = (-\alpha, 0, 0, 0)$ is the unit normal to the $t = \text{constant}$ hypersurface. Additionally, we use the units with $G = c = 1$ throughout.

2.1.1. Fluid. We begin by treating our fluid as a perfect fluid with the stress–energy tensor

$$T_{\mu\nu} = \rho_0 h u_\mu u_\nu + P g_{\mu\nu}, \quad (4)$$

where ρ_0 is the baryon density, $h = 1 + P/\rho_0 + \epsilon$ is the specific enthalpy, P is the pressure, u_μ the four-velocity, and ϵ the specific internal energy.

The general relativistic hydrodynamics equations are evolved using the conservative variables

$$\rho_* = -\sqrt{\gamma} n_\mu n^\mu \rho_0 = \rho_0 W \sqrt{\gamma}, \quad (5)$$

$$\tau = \sqrt{\gamma} n_\mu n_\nu T^{\mu\nu} - \rho_* = \rho_* (hW - 1) - P \sqrt{\gamma}, \quad (6)$$

$$S_i = -\sqrt{\gamma} n_\mu T_{i\mu} = \rho_* h u_i, \quad (7)$$

where $W = \sqrt{1 + \gamma^{ij} u_i u_j}$ is the Lorentz factor and γ is the determinant of γ_{ij} . Using conservation of energy and momentum, $\nabla_\nu T^{\mu\nu} = 0$, and baryon number conservation, $\nabla_\mu (\rho_0 u^\mu) = 0$, we get the evolution equations for the conservative variables:

$$\partial_t \rho_* + \partial_j (\rho_* v_T^j) = 0, \quad (8)$$

$$\partial_t \tau + \partial_j (\alpha^2 \sqrt{\gamma} T^{0i} - \rho_* v_T^i) = -\alpha \sqrt{\gamma} T^{\mu\nu} \nabla_\mu n_\nu, \quad (9)$$

$$\partial_t S_i + \partial_j (\alpha \sqrt{\gamma} T_{ij}) = \frac{1}{2} \alpha \sqrt{\gamma} T^{\mu\nu} \partial_i g_{\mu\nu}, \quad (10)$$

where the Eulerian velocity v^i is related to the fluid transport velocity v_T^i by $v_T^i = \alpha v^i - \beta^i$. Additionally, for simulations involving nuclear matter and neutrinos, we evolve the electron fraction of the fluid, Y_e ,

$$\partial_t (\rho_* Y_e) + \partial_j (\rho_* Y_e v_T^j) = 0. \quad (11)$$

To close these equations we must also supply an equation of state for the pressure and enthalpy: $P = P(\rho_*, T, Y_e)$ and $h = h(\rho_*, T, Y_e)$.

2.1.2. Magnetic fields. To handle magnetic fields, we begin by adding the electromagnetic contribution, $T_{\text{EM}}{}^{\mu\nu}$, to the fluid stress–energy tensor, where

$$T_{\text{EM}}{}^{\mu\nu} = F^{\mu\alpha}F^{\nu\alpha} - \frac{1}{4}F_{\alpha\beta}F^{\alpha\beta}g^{\mu\nu}, \quad (12)$$

and $F^{\mu\nu}$ is the Faraday tensor. We treat the fluid as a perfect conductor, $F^{\mu\nu}u_\nu = 0$, which gives an electric field which can be computed from velocity and magnetic field.

We use two different methods for evolving the magnetic field, as described in [29]. The first method evolves the magnetic vector potential A_i and scalar potential Φ . In the generalized Lorentz gauge [39], the most robust gauge choice we have explored, the evolution equations are

$$\partial_t A_i + \partial_i (\alpha\Phi - \beta^j A_j) = \epsilon_{ijk} v^j B^k, \quad (13)$$

$$\partial_t (\sqrt{\gamma}\Phi) + \partial_j (\alpha\sqrt{\gamma}A^j - \sqrt{\gamma}\beta^j\Phi) = -\xi\alpha\sqrt{\gamma}\Phi, \quad (14)$$

where ξ is a specifiable constant of the order of the mass of the system.

The second method evolves the magnetic field using a covariant hyperbolic divergence cleaning method [40–42] in which an auxiliary scalar evolution variable $\tilde{\Psi}$ is introduced in order to propagate and damp monopole formation. In this method, the induction equation takes the form

$$\partial_t \tilde{B}^i - \partial_i (v^j \tilde{B}^j - v^i \tilde{B}^i) = \alpha\gamma^{ij}\partial_j \tilde{\Psi} + \beta^i \partial_j \tilde{B}^j, \quad (15)$$

$$\partial_t \tilde{\Psi} + \partial_i (\alpha\tilde{B}^i - \beta^i \tilde{\Psi}) = \tilde{B}^i \partial_i \alpha - \alpha (K^i_i + \lambda) \tilde{\Psi}, \quad (16)$$

where $\tilde{B}^i = \sqrt{\gamma}B^i$, $\tilde{\Psi} = \sqrt{\gamma}\Psi$, K^i_i is the trace of the extrinsic curvature, and λ is a specifiable damping constant. Previously [29], we had used Ψ rather than $\tilde{\Psi}$ as an evolution variable, but we find the new choice to be slightly more robust near excision inner boundaries.

2.1.3. Neutrinos. Neutrino evolution is handled using the gray two-moment scheme as described in [43, 44]. This method provides evolution of neutrino average energy densities, flux densities, and number densities. We define three neutrino species that we evolve: electron neutrinos ν_e , electron antineutrinos $\bar{\nu}_e$, and the heavy lepton neutrinos ν_x . The heavy lepton neutrino species groups together the four heavy lepton neutrinos and antineutrinos: $\nu_\mu, \bar{\nu}_\mu, \nu_\tau,$ and $\bar{\nu}_\tau$.

We can describe each of our three species of neutrinos ν_i using each species' distribution function $f_\nu(x^\mu, p^\mu)$, where $x^\mu = (t, x^i)$ gives the time and position of the neutrinos and p^μ is the four-momentum of the neutrinos. f_ν evolves in phase space according to the Boltzmann transport equation:

$$p^\alpha \left[\frac{\partial f_\nu}{\partial x^\alpha} - \Gamma_\gamma^{\beta\alpha} p^\gamma \frac{\partial f_\nu}{\partial p^\beta} \right] = C[f_\nu], \quad (17)$$

where the term $C[f_\nu]$ includes all collisional processes (emissions, absorptions, and scatterings).

We simplify the radiation evolution by taking the gray approximation (integrating over the neutrino spectrum) and evolving the lowest two moments of the distribution functions of each neutrino species, truncating the moment expansion by imposing the Minerbo closure [45]. Our

evolved quantities are projections of the stress–energy tensor of the neutrino radiation, $T_{\text{rad}}{}^{\mu\nu}$. The decomposition of $T_{\text{rad}}{}^{\mu\nu}$ in the fluid frame is

$$T_{\text{rad}}{}^{\mu\nu} = Ju^\mu u^\nu + H^\mu u^\nu + H^\nu u^\mu + S^{\mu\nu} \quad (18)$$

with $H^\mu u_\mu = S^{\mu\nu} u_\mu = 0$. The energy density J , flux density H^μ , and stress density $S^{\mu\nu}$ of the neutrino radiation as observed in the frame comoving with the fluid are related to the distribution functions by

$$J = \int_0^\infty d\nu \nu^3 \iint d\Omega f_{(\nu)}(x^\alpha, \nu, \Omega), \quad (19)$$

$$H^\mu = \int_0^\infty d\nu \nu^3 \iint d\Omega f_{(\nu)}(x^\alpha, \nu, \Omega) l^\mu, \quad (20)$$

$$S^{\mu\nu} = \int_0^\infty d\nu \nu^3 \iint d\Omega f_{(\nu)}(x^\alpha, \nu, \Omega) l^\mu l^\nu, \quad (21)$$

where ν is the neutrino energy in the fluid frame, $\int d\Omega$ denotes integrals over solid angle in momentum space, and

$$p^\alpha = \nu(u^\alpha + l^\alpha), \quad (22)$$

where $l^\alpha u_\alpha = 0$ and $l^\alpha l_\alpha = 1$. We also make use of the decomposition of the neutrino radiation stress–energy tensor as observed by a normal observer,

$$T_{\text{rad}}{}^{\mu\nu} = En^\mu n^\nu + F^\mu n^\nu + F^\nu n^\mu + P^{\mu\nu}, \quad (23)$$

with $F^\mu n_\mu = P^{\mu\nu} n_\mu = F^t = P^{tt} = 0$. Additionally, for each species of neutrino we consider the number current density:

$$N^\mu = Nn^\mu + \mathcal{F}^\mu, \quad (24)$$

where N is the neutrino number density, and \mathcal{F}^μ is the number density flux. The decomposition of N^μ relative to the fluid frame can be expressed in terms of J , H^μ , and the fluid-frame average neutrino energy $\langle \nu \rangle$ as

$$N^\mu = \frac{Ju^\mu + H^\mu}{\langle \nu \rangle}. \quad (25)$$

We define a projection operator onto the reference frame of an observer comoving with the fluid,

$$h_{\alpha\beta} = g_{\alpha\beta} + u_\alpha u_\beta. \quad (26)$$

This allows us to then use the fluid-frame variables to write equations for the energy, flux, and stress tensor in the normal frame (i.e. the frame with four-velocity equal to the normal vector)

$$E = W^2 J + 2W v_\mu H^\mu + v_\mu v_\nu S^{\mu\nu}, \quad (27)$$

$$F_\mu = W^2 v_\mu J + W (g_{\mu\nu} - n_\mu v_\nu) H^\nu + W v_\mu v_\nu H^\nu + (g_{\mu\nu} - n_\mu v_\nu) v_\rho S^{\nu\rho}, \quad (28)$$

$$P_{\mu\nu} = W^2 v_\mu v_\nu J + W (g_{\mu\rho} - n_\mu v_\rho) v_\nu H^\rho + (g_{\mu\rho} - n_\mu v_\rho) (g_{\nu\kappa} - n_\nu v_\kappa) S^{\rho\kappa} + W (g_{\rho\nu} - n_\rho v_\nu) v_\mu H^\rho, \quad (29)$$

by making use of the decomposition of the four-velocity, $u^\mu = W(n^\mu + v^\mu)$.

Evolution equations for $\tilde{E} = \sqrt{\gamma}E$, $\tilde{F}^i = \sqrt{\gamma}F^i$, and $\tilde{N} = \sqrt{\gamma}N$ can then be written in conservative form:

$$\partial_t \tilde{E} + \partial_j (\alpha \tilde{F}^j - \beta^j \tilde{E}) = \alpha \left(\tilde{P}^{ij} K_{ij} - \tilde{F}^j \partial_j \ln \alpha - \tilde{S}_{\text{rad}}^\alpha n_\alpha \right) \quad (30)$$

$$\partial_t \tilde{F}_i + \partial_j \left(\alpha \tilde{P}_i^j - \beta^j \tilde{F}_i \right) = -\tilde{E} \partial_i \alpha + \tilde{F}_k \partial_i \beta^k + \frac{\alpha}{2} \tilde{P}^{jk} \partial_i \gamma_{jk} + \alpha \tilde{S}_{\text{rad}}^\alpha \gamma_{i\alpha}, \quad (31)$$

$$\partial_t \tilde{N} + \partial_j (\alpha \sqrt{\gamma} \mathcal{F}^j - \beta^j \tilde{N}) = \alpha \sqrt{\gamma} C_{(0)}, \quad (32)$$

where $\tilde{P}_{ij} = \sqrt{\gamma} P_{ij}$. Complete treatment of these equations requires prescriptions for the closure relation that computes $P^{ij}(E, F_i)$, the computation of \mathcal{F}^j and the collisional source terms $\tilde{S}_{\text{rad}}^\alpha$ and $C_{(0)}$ which couple the neutrinos to the fluid (and introduce corresponding source terms to the right-hand side of equations (9)–(11)). Details on the treatment for these are beyond the scope of this paper, and are available in [43, 44].

One notable detail of our moment scheme that has to be handled carefully in axisymmetry, however, is the treatment of high-opacity regions. As written above, the two-moment equations lead to excessive diffusion in that regime. When the optical depth of a grid cell becomes ≥ 1 , it would be more accurate to switch to a one-moment scheme ('M0'), with a closure set by the known value of the momentum density and pressure tensor in that regime:

$$H_\mu^{\text{M0}} = \frac{1}{3\kappa_t} \partial_\mu J^{\text{M0}}; \quad S_{\mu\nu}^{\text{M0}} = \frac{1}{3} J^{\text{M0}} (g_{\mu\nu} + u_\mu u_\nu) \quad (33)$$

with κ_t the total opacity of the fluid to neutrinos (absorption and scattering) and

$$J^{\text{M0}} = \frac{3E}{4W^2 - 1}. \quad (34)$$

We follow a slight modification of the scheme proposed in [46], and instead correct the numerical fluxes (divergence terms) in the evolution equations so that \tilde{E}, \tilde{N} evolve as solutions of the diffusion equation in the limit of high κ_t . Let us assume that $F_{\tilde{E}}^{\text{M1}}, F_{\tilde{F}}^{\text{M1}}, F_{\tilde{N}}^{\text{M1}}$ are the numerical fluxes in the two-moment scheme (calculated using our standard closure $P(E, F^i)$ and the HLL Riemann solver), and $F_{\tilde{E}}^{\text{M0}}, F_{\tilde{F}}^{\text{M0}}, F_{\tilde{N}}^{\text{M0}}$ are the same fluxes calculated using the 'M0' closure (i.e. calculating \tilde{F}, \tilde{P} from $J^{\text{M0}}, H^{\text{M0}}, S^{\text{M0}}$). We use as numerical fluxes

$$F_{\tilde{E}, \tilde{N}} = a F_{\tilde{E}, \tilde{N}}^{\text{M1}} + (1 - a) F_{\tilde{E}, \tilde{N}}^{\text{M0}} \quad (35)$$

with $a = \min(1, \tanh A)$, $A = (\kappa_t \Delta x)^{-1}$, and

$$F_{\tilde{F}} = \tilde{A}^2 F_{\tilde{F}}^{\text{M1}} + (1 - \tilde{A}^2) F_{\tilde{F}}^{\text{M0}}, \quad (36)$$

with $\tilde{A} = \min(1, A)$.

Numerical implementation of these M0 fluxes must be done with care. Our numerical methods are designed for conservation-type equations, and the evolution of fields at cell centers is calculated from source terms and from the flux into and out of cells at the faces where they

intersect adjacent cells. Thus, the numerical fluxes have to be estimated on cell faces (halfway between grid points). In fact, we reconstruct flux values on each side of a face (to be combined by the approximate Riemann solver for a shock-capturing scheme). Terms linear in J^{M0} , S^{M0} are *advection* and *pressure gradient terms* that can be computed either using the ‘left’ or ‘right’ state of E on a face. If both states agree on the sign of the advection speed, we use the upstream value of E to calculate these terms. If they do not, we set all advection/pressure terms to zero. Terms linear in H^{M0} , on the other hand, are *diffusion terms* that require the knowledge of $\partial_\mu J^{M0}$ on cell faces. On a cell in direction ‘ μ ’, this can easily be estimated from the value of J^{M0} at neighboring cell centers. For other directions, we (a) calculate $\partial_\mu J^{M0}$ on cell edges by averaging its value on the neighboring faces where it can be evaluated using simple finite differencing; and (b) calculate $\partial_\mu J^{M0}$ on the cell faces where the simple finite differencing method does not work by taking the smallest value of $|\partial_\mu J^{M0}|$ on neighboring cell edges (if both neighbors agree on the sign of the derivative), or setting it to zero (if they do not agree on that sign). This method is inspired from the treatment of derivatives entering the viscous stress tensor in Parrish *et al* [47].

2.1.4. Viscosity. Viscosity is implemented using the approach of [48] that extends the Newtonian large-eddy simulation framework to general relativistic systems. In the large-eddy simulation framework, we recognize that although the equations for energy and momentum evolution allow for evolving modes at all scales, in numerical simulations on a discrete grid we can only evolve modes for which we have sufficient resolution to cover. Thus each computational cell deals with averaged values, while any modes smaller than the cell are removed.

We therefore average over and filter out small scales in the velocity field, leaving equations for the resolved fields:

$$\partial_i \bar{\tau} + \partial_j \left(\overline{\tau v_T^j} + P \sqrt{\gamma} \alpha \bar{v}^j \right) = \alpha \sqrt{\gamma} \left(K_{ij} \bar{S}^{jk} - \bar{S}^i \partial_i \log \alpha \right) \quad (37)$$

$$\begin{aligned} \partial_i \bar{S}_i + \partial_j \left(\overline{S_i v_T^j} + \alpha P \sqrt{\gamma} \delta_{ij} \right) &= \alpha \sqrt{\gamma} \left(\frac{1}{2} \bar{S}^{jk} \partial_i \gamma_{jk} + \frac{1}{\alpha} \bar{S}_k \partial_i \beta^k \right. \\ &\quad \left. - \frac{(\bar{\tau} + \bar{p}_*)}{\sqrt{\gamma}} \partial_i \log \alpha \right), \end{aligned} \quad (38)$$

where K^{ij} is the extrinsic curvature and $S^{ij} = S^i v^j + P \gamma^{ij}$. For our simulations, we will take the averages to be cell averages. In order to complete this set of mean-field equations, we must provide a closure condition for the quantity $\overline{S^i v^j}$:

$$\overline{S_i v_j} = \bar{S}_i \bar{v}_j + \tau_{ij}. \quad (39)$$

τ_{ij} is the subgrid scale stress tensor, that captures the turbulent modes unresolved by our grid. We model this tensor using

$$\tau_{ij} = -2\nu_T \rho h W^2 \left[\frac{1}{2} (\nabla_i \bar{v}_j + \nabla_j \bar{v}_i) - \frac{1}{3} \nabla_k \bar{v}^k \gamma_{ij} \right], \quad (40)$$

where ∇ is the covariant derivative compatible with γ_{ij} . The quantity ν_T possesses a dimension of a viscosity, which leads us to the assumption

$$\nu_T = \ell_{\text{mix}} c_s \quad (41)$$

where c_s is the sound speed of the local fluid. ℓ_{mix} is the characteristic length over which our

subgrid scale turbulence occurs and is known as the mixing length.

As explained in [49], we find that, to maintain the relations equations (5)–(7) for resolved fields, equation (37) must be altered. In this paper, we use the energy equation with the correction to 2nd order in v , which is

$$\partial_i \bar{\tau} + \partial_j \left(\bar{\tau} \bar{v}_T^j + P \sqrt{\gamma} \alpha \bar{v}^j \right) = \alpha \sqrt{\gamma} \left(K_{ij} \bar{S}^{jk} - \bar{S}^i \partial_i \log \alpha \right) - \partial_j \left(\sqrt{\gamma} \tau^{jk} \bar{v}_k \right). \quad (42)$$

2.2. Multipatch axisymmetry

Multipatch methods work by dividing the computational domain into separate domain patches, each of which may have its own local coordinate system x_L^i related to the global coordinate system x_G^i by a map which controls the embedding of the domain in global space. In local coordinates, the patch is (for all applications in this paper) a simple Cartesian grid. The basis vectors $\partial/\partial x_L^i$ and $\partial/\partial x_G^i$ are then related by the Jacobian transformation matrix of the map. Importantly, the patches may have differing shapes in the global coordinate system that can be tailored to better capture the desired features of the simulation. Since our evolution equations for the conservative variables are generally covariant, evolution can be performed directly in the local coordinate system of each individual patch and then the result can be transformed back to the global coordinate system for any necessary communication of information between patches.

SpEC is parallelized using MPI, and the division of grids into patches is used to divide the simulation work. Each domain patch is assigned to a particular processor. For the simulations in this paper, we choose to assign one fluid patch to each processor and, for runs with concurrent spectral evolution of the metric, one pseudospectral patch.

Communication between domain patches occurs through synchronizing values in the ghost zones of each patch at the end of each timestep. In the case that these subdomain patches overlap but do not have directly matching points we communicate data by interpolating values between points. When evolving a vector potential A_i , an additional ghost zone synchronization must be carried out on the magnetic field after it is computed from the curl of A_i . In this case, only the outermost layer of ghost zone points, not a full stencil, should be synchronized. Our curl operator is second-order and only uses nearest neighbors, so interpolated synchronization of additional layers results in magnetic monopole artifacts that grow quickly [29].

Additionally, we create ghost zone points that extend beyond any symmetry boundaries that we have defined in order to impose boundary conditions. During the communication phase, these ghost zone points are filled with data from the live points using the appropriate symmetry conditions. In particular, we must impose symmetry conditions on the symmetry axis. Let us define basis vectors as follows. Imagine a 2D plane, which will represent the computational domain, that crosses the symmetry axis and introduce Cartesian coordinates and basis vectors on the plane. The direction parallel to the axis is ∂_z . The cylindrical radius giving the coordinate distance to the axis is called ϖ , and the corresponding Cartesian coordinate on the plane is ϖ_C . Set $\varpi_C = \varpi$ when $\varpi_C > 0$, but on the other side of the axis, $\varpi_C = -\varpi$. Finally, there is a third Cartesian axis ∂_y points out of the plane and *on the plane* is related to the azimuthal direction $\partial_y = \varpi_C^{-1} \partial_\phi$. To impose the axisymmetry condition, we add a stencil of ghost zones across the axis at negative ϖ . Note that ∂_{ϖ_C} and ∂_ϖ are antiparallel in the ghost zone region, as are ∂_y and ∂_ϕ . We use cell-centered grids in the ϖ_C direction, so the first live point has center half a grid spacing offset from the axis, and no point (live or ghost) is centered exactly on the axis. For scalar quantities, the axisymmetry condition is $f(-\varpi) = f(\varpi)$. For vectors, $v^{\varpi_C}(-\varpi_C) = v^{\varpi_C}(\varpi_C)$, $v^z(-\varpi_C) = v^z(\varpi_C)$, $v^y(-\varpi_C) = -v^y(\varpi_C)$. Below, we will ignore the distinction between ϖ and ϖ_C , since it is only relevant for ghost zones.

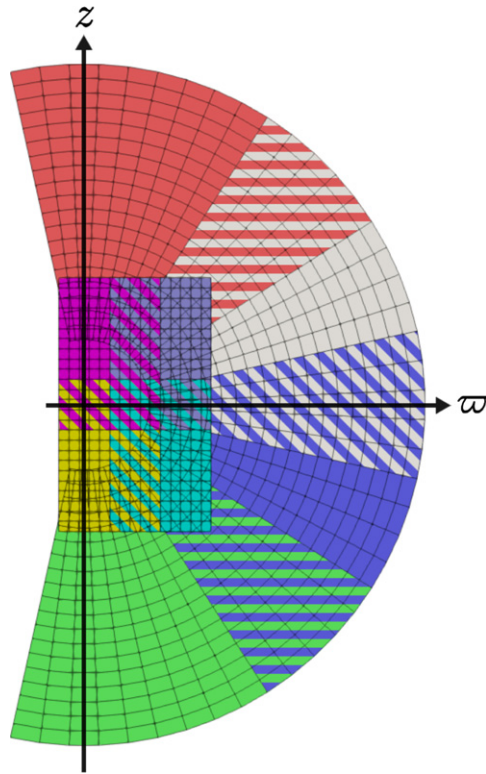


Figure 1. Example of a 2D multipatch grid, for use with axisymmetry, composed of overlapping square (cylindrical-polar) and wedge (spherical-polar) grid shapes. Grid points are arranged so that the coordinate singularity at the symmetry axis falls between grid points. Points extending beyond the symmetry axis are ghost zone points used to impose boundary conditions. Striped regions show portions of the grid where two or more patches are overlapping with matching points.

When evolving a three-dimensional system using a two-dimensional computational domain, each gridpoint represents a ring labeled by two nonazimuthal coordinates. Quite general 2D maps are possible to relate local to global coordinates, but two are particularly useful. A linear map ($x_G^i = a_i x_L^i + b_i$) corresponds to patches that are globally rectangular blocks, covering cylinders in 3D. A polar map [e.g. $x_G^1 = x_L^1 \cos(x_L^2)$, $x_G^2 = x_L^1 \sin(x_L^2)$] corresponds to patches that are globally wedges of circles, covering a specified range of polar r , θ . A combination of wedges covering $0 < \theta < \pi$ in 2D covers a spherical shell domain in 3D. A general 2D grid can contain arbitrary combinations of rectangular blocks and wedges, as shown in figure 1.

Although the grid is 2D, the tangent space on which vectors live is still 3D; even axisymmetric systems can have azimuthal velocity and magnetic field components, for example. The third coordinate in the local coordinate system is set to be the global azimuthal ϕ . Then the local coordinates for a rectangular block will be (up to linear transformation) cylindrical-polar, while the local coordinates for a wedge patch will be (up to linear transformation) spherical-polar. By modifying the map Jacobian, we can make the existing transformation between local and global coordinates handle transforming the third coordinate into an azimuthal coordinate

that can be used to perform axisymmetric evolutions. To do this we expand the elements of the Jacobian matrix using the chain rule to add in the effects of the polar transformation:

$$J^i_j = \frac{\partial x_G^i}{\partial x_L^j} = \frac{\partial x_G^i}{\partial x_A^n} \frac{\partial x_A^n}{\partial x_L^j}, \quad (43)$$

where x_G are the global coordinates, x_L are the local coordinates of a given grid patch, and x_A are a set of global polar coordinates. Since the global and polar coordinates only differ in terms involving the azimuthal direction, the final change from the original Jacobian, J^i_j , to the new axisymmetry Jacobian, $J_{\text{axi}}^i_j$, will be straightforward:

$$J^i_j = \begin{pmatrix} \frac{\partial x_G^1}{\partial x_L^1} & \frac{\partial x_G^1}{\partial x_L^2} & 0 \\ \frac{\partial x_G^2}{\partial x_L^1} & \frac{\partial x_G^2}{\partial x_L^2} & 0 \\ \frac{\partial x_G^3}{\partial x_L^1} & \frac{\partial x_G^3}{\partial x_L^2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \rightarrow J_{\text{axi}}^i_j = \begin{pmatrix} \frac{\partial x_G^1}{\partial x_L^1} & \frac{\partial x_G^1}{\partial x_L^2} & 0 \\ \frac{\partial x_G^2}{\partial x_L^1} & \frac{\partial x_G^2}{\partial x_L^2} & 0 \\ \frac{\partial x_G^3}{\partial x_L^1} & \frac{\partial x_G^3}{\partial x_L^2} & 0 \\ 0 & 0 & \varpi \end{pmatrix}, \quad (44)$$

where ϖ is the coordinate distance from the rotational symmetry axis and we have chosen coordinate directions 1 and 2 to correspond to the two coordinates defined by our two-dimensional computational domain and coordinate direction 3 is transformed to the axisymmetric azimuthal direction ϕ . We also make use of the Hessian matrix in the transformation of the derivatives of metric-related quantities to the local coordinates, and must likewise make similar adjustments to the Hessian:

$$H^i_{jk} = \frac{\partial}{\partial x_L^j} \left(\frac{\partial x_G^i}{\partial x_L^k} \right) = \frac{\partial}{\partial x_L^j} \left(\frac{\partial x_G^i}{\partial x_A^n} \frac{\partial x_A^n}{\partial x_L^k} \right). \quad (45)$$

Explicitly,

$$H^3_{31} = H^3_{13} = J^2_1, \quad (46)$$

$$H^3_{23} = J^2_2, \quad (47)$$

$$H^2_{33} = -\varpi. \quad (48)$$

Generally the evolution of Einstein's equations using SpEC's pseudospectral grid tends to use much less computing time than the hydrodynamics evolution, so our axisymmetry method is primarily aimed at implementing axisymmetric evolution on the hydrodynamics grid while evolving Einstein's equations in 3D. For spherical shell pseudospectral domains, whose collocation points correspond to an expansion of functions in terms of spherical harmonics, azimuthal information can be reduced by reducing azimuthal resolution, corresponding to a lowering of the azimuthal mode number m retained in spectral expansions. It cannot be lowered to $m_{\text{max}} = 0$ because the spectral evolution uses Cartesian components of tensors. We find, however, that the speed increase from doing so is modest, and the resulting spectral grids are more prone to constraint-violating instabilities, so we have not used azimuthal resolution reduction on pseudospectral grids for the simulations in this paper.

Information required by the pseudospectral grid from the hydrodynamics grid is expanded back to 3D during communication. 2D planar fluid data can be extended to 3D for the metric source terms by rotating appropriately about symmetry plane. For the metric data needed for the fluid evolution, as long as the pseudospectrally evolved metric nearly respects the axisymmetry, it is sufficient to take metric data from the xz plane. Deviations from axisymmetry in the

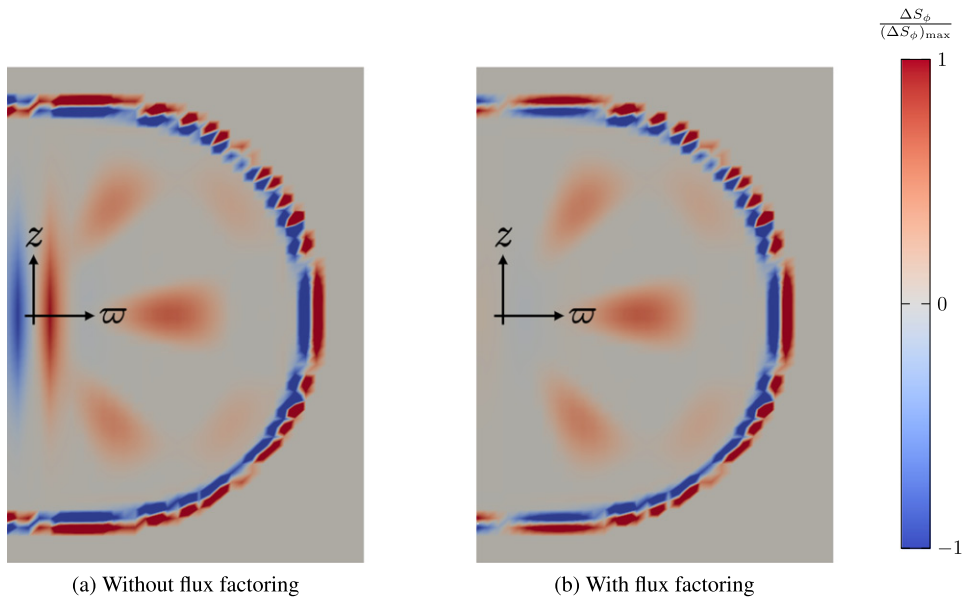


Figure 2. Example of error growth in the evolution of S_ϕ (after transformation back to the global coordinate system) near the symmetry axis of a low resolution, differentially rotating star in a stationary state. Both images plot the difference of S_ϕ between the initial state and the end of the first time step, with the same color scale used for both images chosen to enhance the appearance of errors inside the star. Since the initial conditions are an equilibrium state, all deviations from zero are due to numerical error. A good handling of the symmetry axis leads to errors not being particularly large there. The left image shows the multipatch axisymmetry method applied without factoring of flux terms, while the right shows the star with factoring enabled.

evolved spacetime functions could arise from three sources. The first is numerical error. We have found this small enough to be ignorable, but it could artificially be removed when using spherical shell domains by filtering modes in m . Alternatively, azimuthally-averaged values could be used for the fluid evolution, but we would find the idea of significant uncontrolled violation of the axisymmetry assumption troubling. Second, even if the spacetime retains an azimuthal Killing vector, the coordinate system might evolve in a way that breaks the assumed coordinate form of axisymmetry. Hopefully, the gauge choice will minimize such effects. Merger simulations in SpEC use the damped harmonic gauge [50], and we have found that metric profiles following black hole-neutron star mergers are indeed nearly axisymmetric in the evolved coordinates. This gauge also maintains axisymmetry for the live-metric single-star problem described in section 3.3 below. Finally, the metric could conceivably be subject to some violent non-axisymmetric instability, in which case the methods of this paper are obviously inappropriate.

The conservative form of radiation magnetohydrodynamics evolves variables that are densities and thus proportional to $\sqrt{\gamma}$. Under local to global transformation, the metric determinant transforms as $\sqrt{\gamma_L} = J\sqrt{\gamma_G}$, where J is the determinant of the Jacobian. Note that J is zero on the axis, and indeed would naturally change sign there because the orientation of the basis vectors switches there. SpEC always takes a positive square root, but the only points on the other side of the axis are ghost zone points (needed to impose the symmetry boundary conditions), and non-smooth functions like $\sqrt{\gamma}$ are not interpolated or reconstructed.

Unfortunately, when evolving, this method is prone to producing errors near the symmetry axis that, without correction, grow over time. Vector and tensor valued quantities are most heavily affected due to direct transformation of components in the azimuthal coordinate direction introducing singular terms. An example of this type of error is shown in figure 2. Eventually though, all of our evolved quantities, including scalar quantities, will suffer from errors due to also picking up a singular term in the determinant of the three-metric.

The problem primarily occurs during the computation of the divergence of the flux term, F_A , in the evolution equation of a given quantity A

$$\partial_t A + \partial_i F_A^i = S_A \quad (49)$$

with S_A being any source terms appearing on the right-hand side of the equation.

Some early 2D general relativistic hydrodynamic simulations stabilized the axis evolution using dissipation [51, 52]. Our solution, inspired by [22], is to factor out singular terms that have been introduced to F_A during the transformation to the local coordinates prior to computing the divergence. Depending on the specific component of the flux F_A corresponding to A , there may be multiple factors of ϖ that need to be removed:

$$F_A^i = \varpi^n \tilde{F}_A^i, \quad (50)$$

where \tilde{F}_A is just the ϖ -factored form of the flux, and the integer n will depend on A . We can now instead take the divergence of this factored form of the flux and apply the chain rule, which gives

$$\partial_i (\varpi^n \tilde{F}_A^i) = \varpi^n \partial_i \tilde{F}_A^i + n \varpi^{n-1} \frac{\partial \varpi}{\partial x_L^i} \tilde{F}_A^i. \quad (51)$$

We can also take advantage of the property that if the coordinate specified by ϖ corresponds to one of the directions in the global coordinate system, for example if the global coordinates are Cartesian, the derivatives of ϖ with respect to the local coordinates can be directly taken from components of the Jacobian dealing with the direction associated with ϖ . With this, all of the singular terms introduced from the polar Jacobian are removed from the divergence. Importantly though, the divergence of \tilde{F}_A in the first term on the right side of this equation will need to be computed using the value of \tilde{F}_A at cell faces using the Riemann solver, while \tilde{F}_A in the second term on the right side will use the value at cell centers.

Additionally, since all components have now been transformed into a polar coordinate system, from the definition of axisymmetry we have

$$\partial_\phi F_A^\phi = 0, \quad (52)$$

where the ϕ -index indicates the coordinate of the axisymmetric azimuthal direction. This allows us to ignore the azimuthal portion of the divergences so that we only need to apply the factoring to the two components of the flux that lie in the plane of the computational grid ($i = 1$ and 2 in the below factoring).

All of our evolved quantities carry a factor of $\sqrt{\gamma}$, which will also acquire a singular term, from the transformation of γ_{ij} to the local coordinate system, that also needs to be handled analytically. The flux factoring thus falls into three broad categories for our current evolution equations. Factoring of fluxes for scalar density quantities [equations (8), (9), (11), (14), (16), (30) and (32), and the added term in (42)], takes the form

$$F_A^i = \varpi \tilde{F}_A^i, \quad (53)$$

$$\partial_i F_A^i = \varpi \partial_i \tilde{F}_A^i + \frac{\partial \varpi}{\partial x_L^i} \tilde{F}_A^i. \quad (54)$$

Factoring for covariant vector density quantities [equations (10), (13) and (31)], takes the form

$$F_{A_j}^i = \begin{cases} \varpi \tilde{F}_{A_j}^i & , \text{for } j \neq \phi \\ \varpi^2 \tilde{F}_{A_j}^i & , \text{for } j = \phi, \end{cases} \quad (55)$$

$$\partial_i F_{A_j}^i = \begin{cases} \varpi \partial_i \tilde{F}_{A_j}^i + \frac{\partial \varpi}{\partial x_L^i} \tilde{F}_{A_j}^i & , \text{for } j \neq \phi \\ \varpi^2 \partial_i \tilde{F}_{A_j}^i + 2\varpi \frac{\partial \varpi}{\partial x_L^i} \tilde{F}_{A_j}^i & , \text{for } j = \phi. \end{cases} \quad (56)$$

Factoring for contravariant vector density quantities [equation (15)], takes the form

$$F_{A_j}^i = \begin{cases} \varpi \tilde{F}_{A_j}^i & , \text{for } j \neq \phi, \\ \tilde{F}_{A_j}^i & , \text{for } j = \phi, \end{cases} \quad (57)$$

$$\partial_i F_{A_j}^i = \begin{cases} \varpi \partial_i \tilde{F}_{A_j}^i + \frac{\partial \varpi}{\partial x_L^i} \tilde{F}_{A_j}^i & , \text{for } j \neq \phi, \\ \partial_i \tilde{F}_{A_j}^i & , \text{for } j = \phi. \end{cases} \quad (58)$$

In each of these, the index i only covers coordinates 1 and 2 due to equation (52). SpEC and most other relativistic hydrodynamics codes use conservative shock capturing techniques with approximate Riemann solvers. For codes of this type, a convenient way to implement this factoring program is to use a different coordinate basis, with $\frac{1}{\varpi} \frac{\partial}{\partial x^\phi}$ instead of $\frac{\partial}{\partial x^\phi}$, on cell faces than on cell centers. That is, one simply reconstructs factored quantities.

When evolving a magnetic vector potential, it is also necessary to factor A_ϕ when computing B^i .

$$\partial_i A_\phi = \varpi \partial_i \tilde{A}_\phi + \tilde{A}_\phi \frac{\partial \varpi}{\partial x^i}, \quad (59)$$

where $\tilde{A}_\phi = A_\phi / \varpi$ ⁶.

In their factored form, the principle part of the fluid equations matches 2D Cartesian hydrodynamics (with both sides of the equations divided by a ϖ factor) and so should have similar stability properties. A similar factoring scheme has been extensively tested in 3D [20–22].

The neutrino variables in the M1 scheme are handled in the same way. Even in axisymmetry, the first moment \tilde{F}^i is three-dimensional, and for any transport scheme axisymmetry imposes no constraint on the momentum-space dependence of the distribution function at any spatial point. Factoring of scalar and vector densities is carried out as above. This requires the weighted averages of M1 and M0 fluxes from equations (35) and (36) computed at cell centers. The value of $\partial_\mu J^{\text{M0}}$ on cell centers is estimated by a centered second-order finite

⁶In fact, only factoring for the coordinate i nearly parallel to ϖ on the axis is necessary.

difference using center values of neighboring cells. Since the sign of each component of the advection speed at a cell center is always unambiguous (as opposed to cell faces, for each of which there are two reconstructions), we always add the advective and pressure gradient components of M0 fluxes at cell centers. In fact, this contribution is needed to avoid axis artifacts. The $\tilde{P}^{jk} \partial_i \gamma_{jk}$ source term in equation (31) contains a singular term (from the ϖ^2 factor in γ_{33}) which is canceled by a matching term in the flux from $\alpha \tilde{P}_{ij}$ evaluated at cell centers. In the optically thick limit, this matching term is formally in the advective and pressure gradient part of the M0 flux of \tilde{F}^i . If over an extended optically thick region of the grid the advection speeds on the left and right of cell faces either vanish or differ in sign, then there can be an inconsistency between how the flux is computed at cell faces (for which the advective term would be absent) and cell centers (for which it would be present), which we find also creates axis artifacts when using non-rectangular grids. Such a situation is not likely to occur in realistic simulations, but it does occur in the test problem in section 3.5 below, in which velocities are set to zero. It can be dealt with in a number of ways. One simple way is to add advective fluxes and radiation pressure gradient terms, or at least the latter, on faces even when advective speeds are zero, in that case using the average of values calculated from the two reconstructions. Another simple way is to fall back to M1 fluxes for \tilde{F}^i and handle these as in [43].

Metric-related quantities $(\gamma_{ij}, \alpha, \beta)$ are evolved on their own separate spectral grid in 3D and are communicated to the hydrodynamics grid at the end of each time step. Spatial derivatives of these metric quantities are computed while on the metric grid and then communicated to the hydrodynamics grid, at which point they can be transformed into the local coordinate system as needed. The transformation to local coordinates uses the analytic Jacobian and Hessian, so metric derivatives automatically have their singular factors treated analytically. The transformation equations for global to local components of metric derivatives are

$$\beta_L^j{}_{,i} = (J^{-1})^j{}_j J^i{}_{\bar{i}} \beta_G^{\bar{j}}{}_{\bar{i}} - \beta_L^k (J^{-1})^j{}_j H^{\bar{j}}{}_{ik}, \quad (60)$$

$$\begin{aligned} \gamma_L^{ij}{}_{,k} &= (J^{-1})^i{}_{\bar{i}} (J^{-1})^j{}_j J^{\bar{k}}{}_k \gamma_G^{\bar{i}\bar{j}}{}_{\bar{k}} \\ &\quad - H^{\bar{m}}{}_{kn} [\gamma_L^{nj} (J^{-1})^i{}_{\bar{m}} + \gamma_L^{in} (J^{-1})^j{}_{\bar{m}}]. \end{aligned} \quad (61)$$

2.3. Auxiliary entropy variable

After each substep, the evolved variables $(\rho_*, \tau, S_i, \rho_* Y_e, \tilde{B}^i)$ must be used to recover the primitive variables $(\rho_0, T, Y_e, u_i, B^i)$, a process that involves multi-dimensional root-finding. In particular, if the internal energy is small compared to kinetic or magnetic energy, the temperature recovered from total energy and momentum densities will be unreliable. Due to numerical error, recovered T and u_i , especially at very low densities, may be unphysical, or there may not even be a set of primitive variables corresponding to the evolved variables at a point.

As in [29, 53], we introduce an auxiliary entropy density evolution variable $\rho_* S$, where S is the specific entropy. The variable $\rho_* S$ obeys a continuity equation (viscous and neutrino source terms being unimportant for its purpose) which can be treated in axisymmetry like the other scalar density evolution equations. After each substep in time, SpEC first attempts to recover primitive variables using the standard evolution variables. If this is not possible, or if the recovered specific entropy decreases by more than a fixed percentage compared to its

advected value⁷, primitive variables are recovered disregarding τ and using $\rho_* S$. At the end of each substep, the primitive variables are used to reset all evolution variables, so that τ and $\rho_* S$ are synchronized to each other. The entropy variable allows more reasonable recovery of fluid internal energy at points where this component of the energy is a subdominant contribution to τ and S_i . A particularly challenging problem is accurate primitive variable recovery in magnetically dominated regions like jets. (See [54–56] on inversion methods for this case).

For physical equations of state (e.g. finite-temperature nuclear-theory based EoS), the actual statistical mechanical entropy per baryon can be used to define S . However, in numerical relativity, equations of state are commonly used which have no uniquely defined entropy or temperature, although with absolute zero specified from outside (e.g. for gamma-law EoS, a value of the polytropic constant is defined to be ‘cold’). A common case is an EoS with nuclear physics-motivated cold component plus a simple thermal gamma-law component added on. In terms of baryonic number density $n = \rho_0/m_{\text{amu}}$ and internal energy density u ,

$$P(n, u) = P_c(n) + (\Gamma_{\text{th}} - 1)(u - u_c), \quad (62)$$

where

$$P_c(n) = n^2 \frac{d[U_c/n]}{dn}. \quad (63)$$

The first law gives

$$nT \, dS = -(u + P)dn + n \, du. \quad (64)$$

Combining the three above equations yields, after a short calculation,

$$nT \, dS = \rho_0^{\Gamma_{\text{th}}} \, d[(u - u_c)\rho_0^{-\Gamma_{\text{th}}}], \quad (65)$$

so $(u - u_c)\rho_0^{-\Gamma_{\text{th}}}$ advects for adiabatic change, indicating that this is an acceptable S variable. For gamma-law EoS, one can set $u_c = 0$, yielding the standard auxiliary entropy variable (up to a scaling factor) for this case.

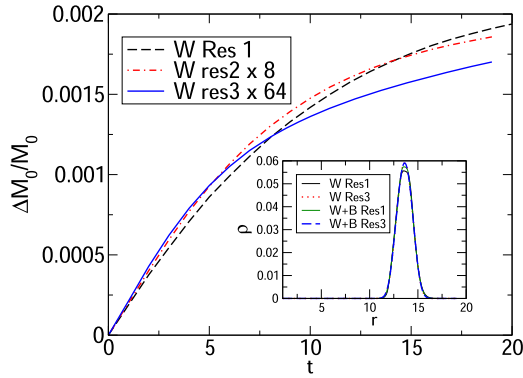
2.4. Low-density treatment

We impose a density floor in low-density regions outside stars, which is necessary to avoid division by zero in our finite difference solver. Also, for densities ~ 2 decades above the floor and below, we impose limits on temperature and velocity; see [57] for details. In the presence of a magnetic field, we do not limit the components of the velocity normal to the field lines even at the lowest densities, to avoid altering the electric field [58]. Test problems with low-density regions also use the auxiliary entropy variable to assist recovery of primitive variables from conservative variables.

⁷The rationale is that shocks, magnetic reconnection, and viscosity can only increase entropy. If the loss of a significant percentage of the entropy at a gridpoint in one timestep by neutrino cooling is considered plausible in a given simulation, this condition would have to be relaxed, for instance to require only that the entropy remain positive. For this paper, we allow the entropy to decrease to 0.97 times its value advected from the previous timestep for problems without neutrino radiation. The 3% buffer lets us avoid only allowing heating errors, which could lead to a systematic drift of temperature. For the neutrino transport problem (section 3.5), the entropy variable was not used.

Table 1. Change in rest mass during evolution of density pulse propagation for pure wedge (W) and wedge plus block (W + B) grids.

Grid	W Res1	W Res2	W Res3	W + B Res1	W + B Res2	W + B Res3
$10^4 \Delta M_0 / M_0$	19	2.3	0.27	26	1.6	0.085

**Figure 3.** Fractional error in rest mass M_0 as a function of time for 3 resolutions of wedge grids for the outgoing density pulse problem. Inset: the equatorial profile of the pulse at $t = 19$ for the lowest and highest resolution wedge (W) and wedge plus block (W + B) grids. Our integration routine does not handle overlapping grids accurately, so we do not plot the time evolution of M_0 for W + B grids, but the final mass error (after the pulse leaves the overlap region) is reported in table 1.

3. Tests

3.1. Mass conservation of an advected pulse

Factoring the continuity equation does lead to truncation error in rest mass conservation. So too does the interpolation needed to fill ghost zones at non-matching patch boundaries for grids like that shown in figure 1. For our applications, we have found these errors to be quite small, but we quantify them here for a simple test. We introduce a density pulse $e^{(r-5)^2}(\cos^2 \theta + 1)$ moving radially outward at $0.5c$ in Minkowski spacetime. We evolve until the center of the pulse reaches $r = 14$. For a first set of runs (W), we use spherical-polar wedges covering $1 < r < 20$. The lowest resolution has 100 radial points and 40 polar points; each subsequent resolution has double the points in each direction as the previous one. For a second set of runs (W + B), we use rectangular blocks surrounded by wedges, with the interface in the $7 < r < 10$ region. The lowest resolution has 100 radial points and 40 polar points in the wedges and 100×50 points in the blocks; each subsequent resolution doubles the points in each direction in each patch.

The patch is able to propagate through the interpolation region without picking up noticeable artifacts or deviation from the purely wedge grid. This is not surprising, since in an earlier paper we were able to propagate a shock through a similar grid in 3D [29]. In table 1 we report the change in rest mass between initial and final times. We see roughly 3rd-order convergence thanks to our 5th-order WENO reconstruction scheme, 3rd-order ghost zone interpolation, and 3rd-order time differencing. This is also demonstrated in figure 3.

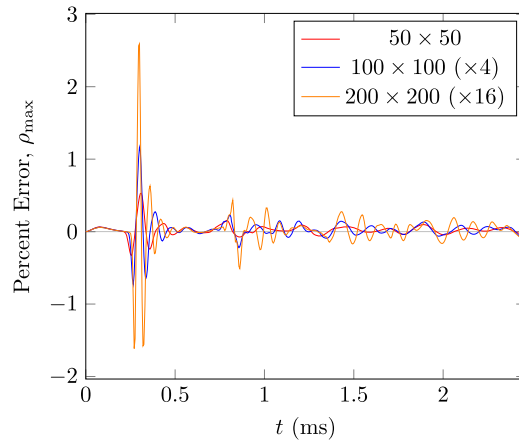


Figure 4. Percent error in the maximum density for the TOV star. Error is shown for grid resolutions of 50×50 , 100×100 , and 200×200 . The error for the 100×100 and 200×200 resolutions have been scaled up by the square of the change in resolution from the 50×50 case.

3.2. TOV star

Initial stability testing was performed using a TOV star in a stationary state. The star was created using a polytropic equation of state with polytropic index $\Gamma = 2$, polytropic constant $\kappa = 100G^3c^{-4}M_\odot^2 = 1.82 \times 10^{10} \text{ cm}^5 \text{ g}^{-1} \text{ s}^{-2}$, and a central density of $7.72 \times 10^{14} \text{ g cm}^{-3}$. This resulted in a gravitational mass of $1.38M_\odot$, a baryonic rest mass of $1.49M_\odot$ and a circumferential radius of 14.22 km. The star was evolved for 2.46 ms in 2D, using both axisymmetry and equatorial symmetry. This is 20 dynamical timescales, using for the dynamical timescale the free-fall time $\sqrt{R^3/(GM)}$. The computational domain was a square grid $14.7 \text{ km} \times 14.7 \text{ km}$ in size, and was evolved using four different resolutions with uniform grid spacing: 50×50 , 100×100 , and 200×200 grid points. For this test, we evolve using the Cowling approximation, meaning the metric is held fixed.

In figure 4 we plot the percent error in the maximum density of the star over time for each resolution, rescaled. We see an initial spike in density at the center of the star. This is caused by relaxation of the surface of the star, creating a disturbance that moves inward. Density is continuous but not smooth at the surface, so this feature exhibits approximately first-order convergence. After this initial peak settles, we see second-order convergence. Rest mass error converges away with resolution and is conserved to $\Delta M_0/M_0 \approx 5 \times 10^{-6}$ for the highest resolution.

3.3. Differentially rotating star

We choose a star with very similar profiles and global properties as the differentially rotating star used in Shibata *et al* [33] and likewise use this star to test the evolution of a system under the influence of an effective viscosity. We point out that this system is not only a useful test of the effective viscosity code, but is designed to resemble the outcome of a binary neutron star merger. Global quantities such as mass, compaction, and spin resemble such a system, although the form of the rotation profile differs from what is found by merger simulations [59–66]. Thus, the simulations in [33] indicate that the interior of the post-merger remnant approaches uniform rotation on a timescale of milliseconds, with the outer layers expanding to form a torus around the central star. Over the next $\sim 10^2$ ms, viscous effects acting on the

outer star and torus drive an outflow of $\sim 10^{-2}M_{\odot}$ (for sufficiently strong viscosity). Below, we demonstrate stable hydrodynamic evolution for 100 ms, and we confirm the formation of the envelope and massive torus structure able to give rise to outflows using an independent code and different viscosity treatment than [33].

The star has an initial baryonic rest mass of $2.64M_{\odot}$ and an equatorial radius $R_e = 10.2$ km. We use a piecewise polytropic equation of state, in two pieces, of the form

$$P = \begin{cases} \kappa_1 \rho^{\Gamma_1}, & \rho \leq \rho_t \\ \kappa_2 \rho^{\Gamma_2}, & \rho \geq \rho_t, \end{cases} \quad (66)$$

where κ_1 and κ_2 are polytropic constants, Γ_1 and Γ_2 are the polytropic indices, and ρ_t is the density at which we transition between the two pieces. For this star, we choose the polytropic indices to be $\Gamma_1 = 4/3$ and $\Gamma_2 = 11/4$; we set the transition density between the two to be $\rho_t = 1.91 \times 10^{14} \text{ g cm}^{-3}$ and the low-density polytropic constant to $\kappa_1 = 0.15GM_{\odot}^{2/3}$.

The initial rotation profile for the star is given by $u^t u_{\phi} = \hat{A}(\Omega_0 - \Omega)$ where Ω_0 is the angular velocity along the rotation axis and we choose $\hat{A} = 0.8R_e$. The initial equilibrium state is supplied by the code of Cook, Shapiro, and Teukolsky [67].

In order to handle outflows that will occur when viscosity is enabled, we create a computational grid better suited for resolving both the central star and low density outflowing material. Since any outflows that occur will rapidly drop in density and are not expected to have any small detail features of concern after they leave the region of the star, we leverage the utility of the multipatch technique to apply differing grid structures to each zone of interest. In the central region containing the star we employ the same rectangular grid structure as seen in the previous TOV star test, with a resolution of 100×100 grid points. In the outflow region we switch to a polar grid with constant latitude resolution (so that the proper spacing between angularly adjacent points increases with distance from the star). The polar grid has 50 points in the angular direction (covering $0 < \theta < \pi/2$) and 400 points in the radial direction. We apply a map to the entire grid that allows us to reduce radial resolution at large distances:

$$R = r + 2 e^{-\gamma\beta} \sinh(\gamma r), \quad (67)$$

where r is the radius in grid coordinates (the coordinates in which radial grid spacing is uniform), and R is the radius in the original quasi-isotropic, asymptotically-Minkowski coordinates. The map provides an approximately linear grid spacing for radii less than β , which we have chosen to be at 25.85 km, and then switches to an exponential grid spacing based on γ , which is chosen such that $r_{\text{outer}} = 73.5$ km is mapped to $R_{\text{outer}} = 2205$ km. The pseudospectral grid used for the evolution of Einstein's equations is composed of an inner ball at the center of the star surrounded by a series of spherical shells extending to a distance of 2940 km. For the gauge choice, we freeze the generalized harmonic gauge function H_{α} to minimize coordinate dynamics.

In this test, we have modified the density floor from our previous implementations to use a floor dependent on radius:

$$\rho_0 > \frac{A}{1 + R^2} + B, \quad (68)$$

where we have chosen $A = 1.62 \times 10^4 \text{ g cm}^{-3}$ and $B = 1.62 \times 10^{-2} \text{ g cm}^{-3}$.

For this test, we employ the viscosity treatment described in section 2. To make a comparison with the results of the α -viscosity model used in [12], we devise a mixing length ℓ_{mix} corresponding to the same kinematic viscosity as a constant α . The α -viscosity model is generalized to differentially rotating stars in [12] by setting

$$\nu_\alpha = \frac{\alpha c_s^2}{\Omega_e}, \quad (69)$$

where c_s is again the local sound speed and Ω_e is the angular velocity of the star at the surface on the equator. Equating this to equation (41), we can get an approximate relation between the strength of a given mixing length to that of an α -viscosity parameter:

$$\ell_{\text{mix}} = \frac{\alpha c_s}{\Omega_e}. \quad (70)$$

For the current test we set the viscous mixing length to $\ell_{\text{mix}} = 147$ m, giving a comparable viscous strength to $\alpha \approx 0.01$ in the interior of the star. (For constant ℓ_{mix} , α as defined in equation (69) is not exactly constant.) The timescale for viscous angular momentum transport is approximately R^2/ν . Using equation (41) gives a timescale on the order of

$$t_{\text{visc}} \sim 10 \text{ ms} \left(\frac{r}{10 \text{ km}} \right)^2 \left(\frac{\ell_{\text{mix}}}{147 \text{ m}} \right)^{-1} \left(\frac{c_s}{0.3c} \right)^{-1}. \quad (71)$$

As evolution begins the star quickly begins to transport angular momentum outward causing the rotational velocity profile to become flatter. Although the rotation profile does flatten, we see from figure 6 that the profile never completely settles into a rigidly rotating state, and retains some differential rotation. This is a feature of this viscosity method [49].

Qualitatively the outflow near the star produces the expected distribution of material producing a short, low density burst of material as viscosity is enabled, and at later times as more material leaves the star a disk begins to form. Other material blows farther outward, indicating the beginnings of a viscous-driven outflow noted in [33] which we do not follow. The density profiles in figure 5 are to be compared to figure 4 in Shibata *et al* [33]. We note that even the qualitative agreement we see in the density plots is nontrivial; it requires the correct treatment of the energy equation described in section 2.1.4. Subgrid momentum transport is modeled in [33] via an Israel–Stewart-type formulation of the relativistic Navier–Stokes equations, which is analytically quite different from our treatment. Our qualitative agreement with this previous work gives confidence that its results will not prove very sensitive to details of the momentum transport modeling. As a further check, we supply convergence tests of the spacetime and fluid evolution in figure 7.

3.4. Magnetized disk

We evolve a standard axisymmetric MHD test problem: a magnetized torus around a Kerr black hole. The initial conditions for this test are matched to the ‘fiducial model’ of McKinney and Gammie [68]. A black hole with dimensionless spin $J/M^2 = 0.938$ is surrounded by a Fishbone–Moncrief torus [69] with inner edge at $r_{\text{BL}} = 6M$ and specific angular momentum determined by $u^t u_\phi = 4.281$. The torus has initial maximum density $\rho_0 = 1$ and a $\Gamma = 4/3$ equation of state. The torus initially has pressure $P = \kappa \rho^\Gamma$, with $\kappa = 0.00425$, $\Gamma = 4/3$. A confined poloidal seed field is introduced via the initial vector potential one-form $\tilde{A} = A_0 \max(\rho_0 - 1, 0) d\phi$, with A_0 chosen to make the maximum ratio of magnetic to gas pressure be around 0.01. We evolve for $3000M$ on a spherical-polar grid with inner radius at $r_{\text{BL}} = 1.32M$ and maximum radius at $60M$. We evolve with 170^2 , 256^2 , 384^2 , and 512^2 grids, for which the initial fraction of the magnetic energy at points where the fastest-growing MRI mode is resolved by at least 10 points is 0.194, 0.44, 0.66, 0.79, respectively.

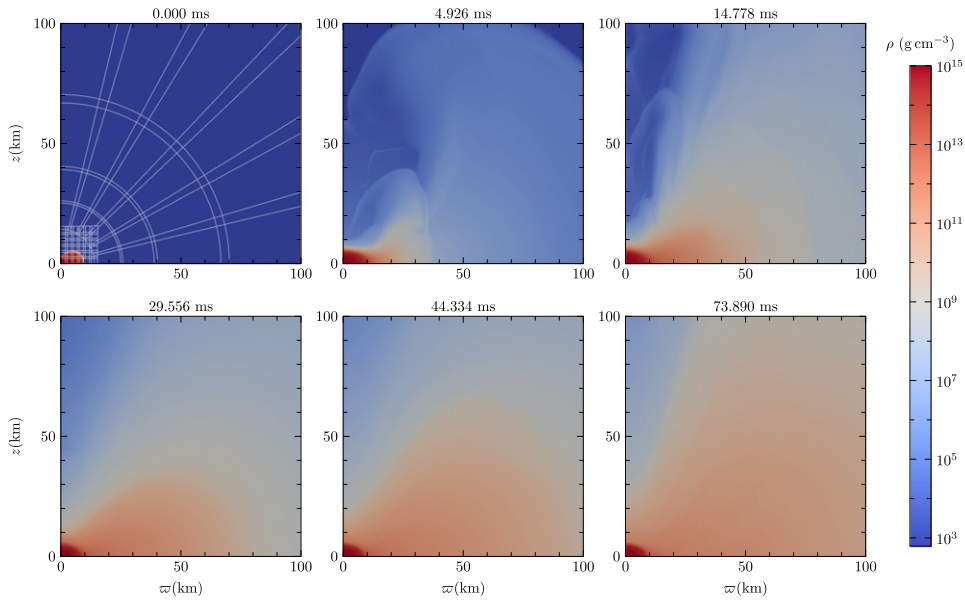


Figure 5. Evolution of density profiles of the differentially rotating viscous star with mixing length $\ell_{\text{mix}} = 147$ m. The first panel shows the outlines of each of the overlapping subdomain patches used to construct the computational domain.

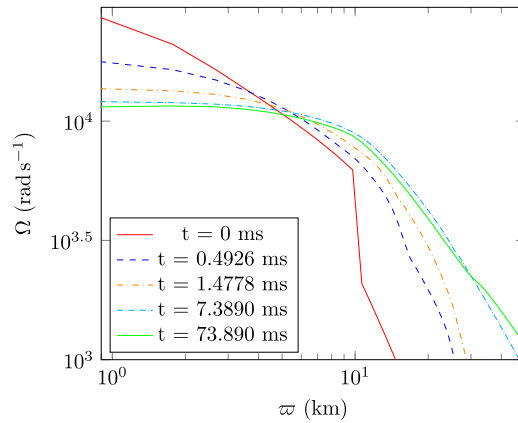


Figure 6. Rotational velocity profile of the viscous differentially rotating star in the equatorial plane at multiple times. We see the rotation profile begin to flatten as viscous effects redistribute angular momentum inside the star.

The Kerr spacetime is written in in Kerr–Schild coordinates. We make the standard change of variables for spherical-polar disk simulations:

$$r = \sqrt{x^2 + z^2} = e^{x_1}, \quad (72)$$

$$\theta = \pi x_2 + \frac{1}{2}(1 - h) \sin(2\pi x_2). \quad (73)$$

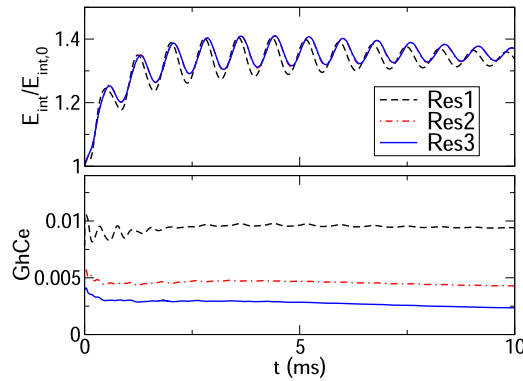


Figure 7. Global quantities for three resolutions during the first 10 ms, the most dynamic phase of the differentially rotating star evolution due to effective viscosity. Each resolution has 30% more points in each direction on each subdomain patch of the fluid grid and one extra radial collocation point, two extra angular collocation points for each subdomain of the spectral grid. Top: the total internal energy normalized to its initial value. The two highest resolutions nearly coincide. Overall, the star heats due to viscosity, but adiabatic fluctuations due to oscillations of the star are also visible. These modes are excited by a combination of imperfection of the numerically-generated initial data and because the dynamical and viscous timescales are only separated by a factor of $\sim 10^2$. Bottom: the normalized generalized harmonic constraint violation.

Setting a uniform grid in x_1, x_2 concentrates resolution near the black hole and on the equator. We set $h = 0.5$. Finally, because $r \neq r_{\text{BL}}$ we compose with a final coordinate map to map the coordinate spheres $(x^2 + z^2)^{1/2} = C$ to surfaces of constant Kerr radius $r_{\text{BL}} = C$. This allows an excision inner boundary inside the horizon $r_{\text{BL}} = r_+$ that conforms better to the horizon shape.

For this run, we use a position-dependent density floor $\rho_0 > 10^{-5} r^{-3/2} (\kappa/0.00425)^{-3} c^6$. We also increase ρ_0 and P in the magnetically-dominated region as needed to maintain $b^2/\rho_0 < 10$ and $b^2/P < 500$, which significantly improves the step size chosen by the adaptive timestepper. We evolve both with hyperbolic divergence cleaning and vector potential evolution. For the vector potential evolution, we use the generalized Lorentz gauge [39]. Simpler gauges, such as the algebraic $\partial_t \tilde{A} = \vec{v} \cdot \tilde{B}$ and advective $\partial_t \tilde{A} = -\mathcal{L}_v \tilde{A}$ give the same evolution of gauge-invariant quantities but, after a while, at a drastically reduced timestep, presumably because the vector potential does not remain as smooth.

The vector potential evolution benefits from added explicit dissipation. We apply Kreiss–Oliger dissipation [70] to the evolution of A_i and Φ with a coefficient of 0.001. (Our dissipation operator is defined as a sum of fourth derivatives with respect to local coordinates but applied to global components of the relevant evolved variables.) Without dissipation, grid-scale ripples appear in the magnetic field atop an otherwise reasonable field structure. If the coefficient is increased to 10^{-2} , the main difference is a slightly lower asymptotic speed in the polar jets. Kreiss–Oliger dissipation is not needed for divergence cleaning runs; in fact, it destabilizes the magnetic field evolution near the excision zone. Instead, extra dissipation for divergence cleaning simulations is obtained by setting the maximum signal speeds in our HLL Riemann solver for the evolution of \tilde{B} and Ψ to the null speeds.

The qualitative expectations for this problem are well-known and are reproduced for our runs for both types of B field evolution. Magnetic winding generates a toroidal magnetic field,

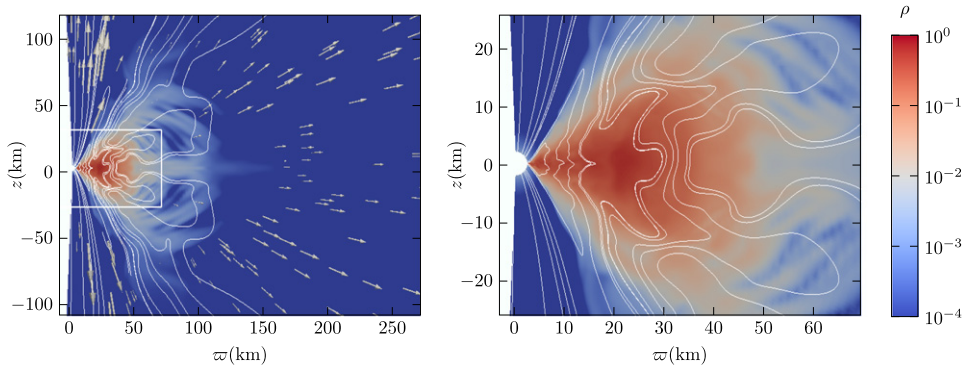


Figure 8. Magnetized disk with field lines (for field with toroidal component projected out) at time $t = 1500M$, with M the mass of the black hole (which we set equal to one), for evolution on a 256^2 grid. Note that the grid extends slightly to the left of the axis because of the symmetry ghost zones. The right panel shows the region highlighted by the white box in the left panel. Magnetic field and velocity fields are averaged over the time $1000 < t/M < 1500$. The initial maximum density in the torus is chosen to be unity. Profiles are plotted in Kerr–Schild coordinates. The longest velocity vector arrows close to the poles far from the black hole correspond to speed very close to $1 = c$.

while the magnetorotational instability triggers turbulence in the disk. Matter falls into the black hole at an average rate of about $\dot{M} \approx 10^{-1}c^3/G$. The poles become magnetically dominated. An outgoing Poynting flux can be found in this region, and gas accelerates to near the speed of light on the poles away from the black hole. In parts of the polar jet region, b^2/P and the Lorentz factor grow to the limits imposed by our atmosphere algorithm (500 and 20, respectively). The magnetic field energy grows for the first $1000M$, then saturates, then begins to die away at a steady rate. This decrease of the magnetic field is not physical but it is expected in any axisymmetric simulation (at least one not enhanced by dynamo-modeling additions to the induction equation [71]) because of the anti-dynamo theorem. Outside the region close to the poles, a mildly relativistic wind is seen. The configuration of the system at $t = 1500M$ is shown in figure 8. All resolutions produce essentially this same configuration. However, because the MRI is resolved in more of the disk, the effect of increasing resolution is to enhance turbulence, in that eddies appear more distinctly and at finer scales for higher resolutions and the accretion rate increases. Also, while the total magnetic energy is initially resolution-independent (when it is dominated by magnetic winding and linear MRI growth), it saturates at a higher level at higher resolution. (The lowest resolution saturates at a factor of two lower energy than the highest resolution; the other resolutions are, of course, closer to the highest, with saturation energy increasing monotonically with resolution, although because it is a turbulent problem no clear order of convergence can be identified).

None of this is newsworthy, although it is reassuring to confirm for the first time that SpEC can produce magnetically-dominated jets when they are expected. For our purposes, the main value of this test is that we can check, for a complex, astrophysically interesting MHD problem, that our code produces no unphysical axis artifacts in any quantity we have checked (ρ_0 , v^i , B^i , b^2/P). Of course, the axis actually is a special region in this problem, which is clearly seen in the solution, but this can easily be distinguished from artifacts of the coordinate singularity because the latter, when they appear (as they do not in this case), have grid-spacing width. The absence of such glitches is, in fact, a nontrivial accomplishment. For divergence cleaning evolutions without factoring of the evolution equations, grid-scale axis artifacts in the velocity

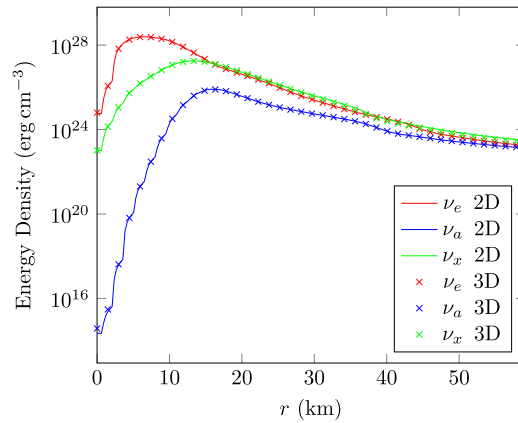


Figure 9. Comparison of neutrino energy density along the z -axis for the 2D and 3D spherically symmetric supernova collapse profile at $t = 1.5$ ms. ν_e , ν_a , and ν_x represent the electron neutrinos, electron antineutrinos, and heavy lepton neutrinos respectively.

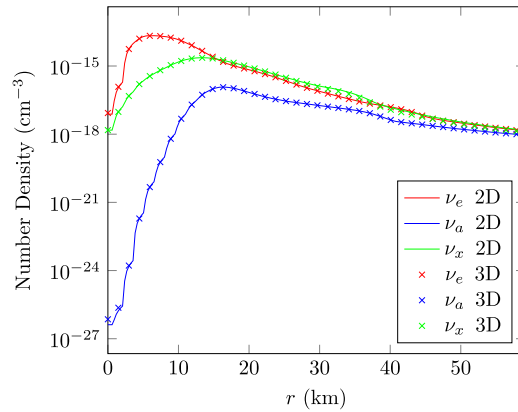


Figure 10. Comparison of neutrino number density along the z -axis for the 2D and 3D spherically symmetric supernova collapse profile at $t = 1.5$ ms. ν_e , ν_a , and ν_x represent the electron neutrinos, electron antineutrinos, and heavy lepton neutrinos respectively.

are easily seen, although they can be suppressed by using low-order reconstruction (MC2 [72]) near the axis. For vector potential evolutions without factoring the computation of \vec{B} from A , axis glitches become so severe that simulations crash shortly after accretion onto the black hole begins.

Although the results are qualitatively similar, we consider the vector potential method superior for this problem, at least with our current implementations. In divergence cleaning methods, Ψ builds up at boundaries, particularly the excision boundary. The amount tends to grow with time and we fear would eventually endanger the simulation. Because of it, magnetic energy fluxes are not reliable in the inner layer of points (while in vector potential evolutions, the inner layer shows no problems). Presumably the solution would be to improve the treatment of the magnetic variables at boundaries.

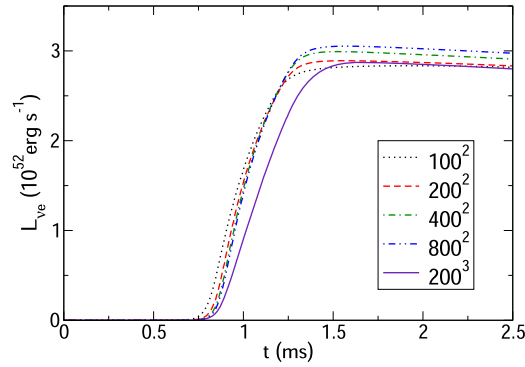


Figure 11. Luminosity in electron neutrinos for 3D and 2D evolutions at various resolutions. The luminosity is calculated by integrating the energy flux through the outer boundary. Thus, the 3D curve reaches its peak slightly later, since the outer corner of the 3D grid is at $\sqrt{3} \times 300$ km as opposed to the 2D corner at $\sqrt{2} \times 300$ km.

3.5. Neutrino radiation

Initial testing of the axisymmetric neutrino code was performed by comparing the results obtained from the spherically symmetric post-bounce supernova profile used in [44] in both 2D axisymmetry with equatorial symmetry and in 3D using octant symmetry. The density ranges from $4 \times 10^{14} \text{ g cm}^{-3}$ at the center of the protoneutron star to 10^7 g cm^{-3} at the outer boundary. The temperature peaks at 22 MeV at a radius of 13 km. The profile covers a wide range of neutrino opacity from very opaque to very transparent; the neutrinosphere is at a radius of around 30 km. In this test we evolve the moments of the neutrino distribution function, fluid temperature, and fluid composition (the electron fraction Y_e) for a 1D profile constructed as a spherical average of a 2D core collapse simulation 160 ms after bounce. The velocity of the fluid is set to zero.

We perform this test in 2D on a square grid with length 300 km and a resolution of 200×200 grid points. In 3D, we use a cube with the same length of 300 km and a resolution of $200 \times 200 \times 200$ grid points. Both systems were evolved for 1.5 ms using a fixed timestep of 4.9×10^{-3} ms to ensure that no error was introduced from possible differences between 2D and 3D in the adaptive timestepper. The 2D test used 24 processing cores and required 2.41 core-hours of run time, whereas the 3D test on 48 cores required 470.27 core-hours, achieving a speed up factor of ~ 195 . We see very strong agreement in results between the 2D and 3D results, as seen in figures 9 and 10. We also see acceptable agreement in the neutrino luminosity; for the electron neutrinos (the neutrino flavor whose luminosity shows a clearest settled value), both 2D and 3D settle to within about a percent of each other at $L_{\nu_e} = 3 \times 10^{52} \text{ erg s}^{-1}$. By carrying out 2D simulations at other resolutions, we confirm that the 2D vs 3D agreement is within the truncation error, as shown in figure 11. Although L_{ν_e} is consistent between resolutions up to differences under 10%, demonstrating a clear order of convergence is difficult. This is because of the very sharp behavior of the neutrino fluxes near the neutrinosphere, for which one would require extremely high resolution to be in the convergence regime.

The agreement between 2D and 3D might seem trivial since the 2D and 3D grids are closely matched, but the polar transformation significantly alters the flux divergence and metric derivative source terms considered separately. Also, factoring is essential for avoiding strong axis artifacts.

4. Conclusion

We have implemented an axisymmetric evolution of the general relativistic hydrodynamics equations through modification of the local coordinate transformations of a multipatch scheme. Without the appropriate factoring of singular terms from spatial derivatives near the symmetry axis, we find that unphysical errors grow in evolved quantities. Testing of this method, with factoring of singular terms applied, produces results that compare favorably to full 3D simulations at a fraction of the required computational time. Since only minimal modification of the implementation of the evolution equations in 3D was required, this method provides a path for a quick application of axisymmetric evolution to codes that make use of computational domains with local coordinate transformations.

We plan to move forward using this method in order to study the effects of a wide variety of physical parameters on binary post-merger environments that require evolution on secular timescales that we have been unable to explore in the past. Additionally, although our method currently evolves Einstein's equations in 3D using spectral methods, we would also like to extend axisymmetry to the evolution of those equations as well.

Acknowledgments

JJ would like to acknowledge Guy Worthey and Sukanta Bose for providing useful comments on an earlier draft of this paper. MD would like to acknowledge useful discussions with Thomas Baumgarte, which helped MD overcome his prejudice against analytical treatment of coordinate singularities. JJ gratefully acknowledges support from the Washington NASA Space Grant Consortium, NASA Grant NNX15AJ98H. MD gratefully acknowledges support from the NSF through Grant PHY-1806207. FF and AK gratefully acknowledge support from the NSF through Grant PHY-1806278, from NASA through Grant 80NSSC18K0565, and DOE-CAREER Grant DE-SC0020435. HP gratefully acknowledges support from the NSERC Canada. LK acknowledges support from NSF Grant PHY-1606654 and PHY-1912081. FH and MS acknowledge support from NSF Grants PHY-170212 and PHY-1708213. FH, LK and MS also thank the Sherman Fairchild Foundation for their support.

ORCID iDs

Jerred Jesse  <https://orcid.org/0000-0001-5266-4536>
Matthew D Duez  <https://orcid.org/0000-0002-0050-1783>
Milad Haddadi  <https://orcid.org/0000-0001-7548-9099>
Lawrence E Kidder  <https://orcid.org/0000-0001-5392-7342>

References

- [1] Abbott B P *et al* (LIGO Scientific, Virgo) 2016 *Phys. Rev. Lett.* **116** 061102
- [2] Abbott B P *et al* (LIGO Scientific, Virgo) 2016 *Phys. Rev. Lett.* **116** 241102
- [3] Abbott B P *et al* (LIGO Scientific, Virgo) 2016 *Phys. Rev. Lett.* **116** 241103
- [4] Abbott B P *et al* (LIGO Scientific, Virgo) 2017 *Phys. Rev. Lett.* **118** 221101
- [5] Abbott B P *et al* (LIGO Scientific, Virgo) 2017 *Astrophys. J. Lett.* **851** L35
- [6] Abbott B P *et al* (LIGO Scientific, Virgo) 2017 *Phys. Rev. Lett.* **119** 141101

- [7] Abbott B *et al* 2019 *Phys. Rev. X* **9** 031040
- [8] Abbott B P *et al* (LIGO Scientific, Virgo) 2017 *Phys. Rev. Lett.* **119** 161101
- [9] Abbott B P *et al* (LIGO Scientific, Virgo, Fermi GBM, INTEGRAL, IceCube, AstroSat Cadmium Zinc Telluride Imager Team, IPN, Insight-Hxmt, ANTARES, Swift, AGILE Team, 1M2H Team, Dark Energy Camera GW-EM, DES, DLT40, GRAWITA, Fermi-LAT, ATCA, ASKAP, Las Cumbres Observatory Group, OzGrav, DWF (Deeper Wider Faster Program), AST3, CAASTRO, VINROUGE, MASTER, J-GEM, GROWTH, JAGWAR, CaltechNRAO, TTU-NRAO, NuSTAR, Pan-STARRS, MAXI Team, TZAC Consortium, KU, Nordic Optical Telescope, ePESSTO, GROND, Texas Tech University, SALT Group, TOROS, BOOTES, MWA, CALET, IKI-GW Follow-up, H.E.S.S., LOFAR, LWA, HAWC, Pierre Auger, ALMA, Euro VLBI Team, Pi of Sky, Chandra Team at McGill University, DFN, ATLAS Telescopes, High Time Resolution Universe Survey, RIMAS, RATIR, SKA South Africa/MeerKAT) 2017 *Astrophys. J. Lett.* **848** L12
- [10] Abbott B P *et al* (LIGO Scientific, Virgo, FermiGBM, INTEGRAL) 2017 *Astrophys. J. Lett.* **848** L13
- [11] Abbott B P *et al* (LIGO Scientific, Virgo) 2020 Gw190425: observation of a compact binary coalescence with total mass $\sim 3.4m_{\odot}$ (arXiv:2001.01761)
- [12] Shibata M and Kiuchi K 2017 *Phys. Rev. D* **95** 123003
- [13] Hotokezaka K, Kiuchi K, Kyutoku K, Muranushi T, Sekiguchi Y i, Shibata M and Taniguchi K 2013 *Phys. Rev. D* **88** 044026
- [14] Paschalidis V, Etienne Z B and Shapiro S L 2012 *Phys. Rev. D* **86** 064032
- [15] Alcubierre M, Brüggmann B, Holz D, Takahashi R, Brandt S, Seidel E and Thornburg J 2001 *Int. J. Mod. Phys. D* **10** 273–89
- [16] Duez M D, Liu Y T, Shapiro S L and Stephens B C 2004 *Phys. Rev. D* **69** 104030
- [17] Duez M D, Liu Y T, Shapiro S L, Shibata M and Stephens B C 2006 *Phys. Rev. D* **73** 104015
- [18] Shibata M 2003 *Astrophys. J.* **595** 992–9
- [19] Pretorius F 2005 *Class. Quantum Grav.* **22** 425–51
- [20] Baumgarte T W, Montero P J, Cordero-Carrión I and Müller E 2013 *Phys. Rev. D* **87** 044026
- [21] Montero P J, Baumgarte T W and Müller E 2014 *Phys. Rev. D* **89** 084043
- [22] Baumgarte T W, Montero P J and Müller E 2015 *Phys. Rev. D* **91** 064035
- [23] Anderson M, Hirschmann E W, Lehner L, Liebling S L, Motl P M, Neilsen D, Palenzuela C and Tohline J E 2008 *Phys. Rev. D* **77** 024006
- [24] Koldoba A V, Romanova M M, Ustyugova G V and Lovelace R V E 2002 *Astrophys. J.* **576** L53–6
- [25] Schnetter E, Diener P, Dorband E N and Tiglio M 2006 *Class. Quantum Grav.* **23** S553–78
- [26] Reisswig C, Bishop N T, Lai C W, Thornburg J and Szilágyi B 2007 *Class. Quantum Grav.* **24** S327–40
- [27] Zink B, Schnetter E and Tiglio M 2008 *Phys. Rev. D* **77** 103015
- [28] Fragile P C, Lindner C C, Anninos P and Salmonson J D 2009 *Astrophys. J.* **691** 482–94
- [29] Hossein Nouri F *et al* 2018 *Phys. Rev. D* **97** 083014
- [30] Simulating eXtreme Spacetimes (SXS) collaboration 2020 SpEC: spectral Einstein code <https://black-holes.org/code/SpEC.html> (accessed 25 February 2020)
- [31] Hébert F, Kidder L E and Teukolsky S A 2018 *Phys. Rev. D* **98** 044041
- [32] Kidder L E *et al* 2017 *J. Comput. Phys.* **335** 84–114
- [33] Shibata M, Kiuchi K and Sekiguchi Y i 2017 *Phys. Rev. D* **95** 083005
- [34] Lindblom L, Scheel M A, Kidder L E, Owen R and Rinne O 2006 *Class. Quantum Grav.* **23** S447–62
- [35] Harten A, Lax P D and Leer B v 1983 *SIAM Rev.* **25** 35–61
- [36] Liu X D, Osher S and Chan T 1994 *J. Comput. Phys.* **115** 200–12
- [37] Jiang G S and Shu C W 1996 *J. Comput. Phys.* **126** 202–28
- [38] Duez M D, Foucart F, Kidder L E, Pfeiffer H P, Scheel M A and Teukolsky S A 2008 *Phys. Rev. D* **78** 104015
- [39] Farris B D, Gold R, Paschalidis V, Etienne Z B and Shapiro S L 2012 *Phys. Rev. Lett.* **109** 221102
- [40] Liebling S L, Lehner L, Neilsen D and Palenzuela C 2010 *Phys. Rev. D* **81** 124023
- [41] Penner A J 2011 *Mon. Not. Roy. Astron. Soc.* **414** 1467–82
- [42] Mösta P *et al* 2013 *Class. Quantum Grav.* **31** 015005
- [43] Foucart F *et al* 2015 *Phys. Rev. D* **91** 124021
- [44] Foucart F, O’Connor E, Roberts L, Kidder L E, Pfeiffer H P and Scheel M A 2016 *Phys. Rev. D* **94** 123016
- [45] Minerbo G N 1978 *J. Quant. Spectrosc. Radiat. Transfer* **20** 541–5
- [46] Audit E, Charrier P, Chièze J and Dubroca B 2002 arXiv:astro-ph/0206281

- [47] Parrish I J, McCourt M, Quataert E and Sharma P 2012 *Mon. Not. Roy. Astron. Soc.* **422** 704–18
- [48] Radice D 2017 *Astrophys. J.* **838** L2
- [49] Duez M D et al 2020 in preparation
- [50] Szilágyi B 2014 *Int. J. Mod. Phys. D* **23** 1430014
- [51] Shibata M 2000 *Prog. Theor. Phys.* **104** 325–58
- [52] Duez M D, Liu Y T, Shapiro S L and Stephens B C 2004 *Phys. Rev. D* **69** 104030
- [53] Noble S C, Krolik J H and Hawley J F 2009 *Astrophys. J.* **692** 411–21
- [54] Noble S C, Gammie C F, McKinney J C and Del Zanna L 2006 *Astrophys. J.* **641** 626–37
- [55] Siegel D M, Mösta P, Desai D and Wu S 2018 *Astrophys. J.* **859** 71
- [56] Kastaun W, Kalinani J V and Ciolfi R 2020 arXiv:2005.01821
- [57] Foucart F et al 2013 *Phys. Rev. D* **87** 084006
- [58] Muhlberger C D, Nouri F H, Duez M D, Foucart F, Kidder L E, Ott C D, Scheel M A, Szilágyi B and Teukolsky S A 2014 *Phys. Rev. D* **90** 104014
- [59] Kastaun W and Galeazzi F 2015 *Phys. Rev. D* **91** 064027
- [60] Endrizzi A, Ciolfi R, Giacomazzo B, Kastaun W and Kawamura T 2016 *Class. Quantum Grav.* **33** 164001
- [61] Kastaun W, Ciolfi R and Giacomazzo B 2016 *Phys. Rev. D* **94** 044060
- [62] Ciolfi R, Kastaun W, Giacomazzo B, Endrizzi A, Siegel D M and Perna R 2017 *Phys. Rev. D* **95** 063016
- [63] Hanauske M, Takami K, Bovard L, Rezzolla L, Font J A, Galeazzi F and Stöcker H 2017 *Phys. Rev. D* **96** 043004
- [64] Endrizzi A, Logoteta D, Giacomazzo B, Bombaci I, Kastaun W and Ciolfi R 2018 *Phys. Rev. D* **98** 043015
- [65] Kiuchi K, Kyutoku K, Sekiguchi Y and Shibata M 2018 *Phys. Rev. D* **97** 124039
- [66] Ciolfi R, Kastaun W, Kalinani J V and Giacomazzo B 2019 *Phys. Rev. D* **100** 023005
- [67] Cook G B, Shapiro S L and Teukolsky S A 1992 *Astrophys. J.* **398** 203–23
- [68] McKinney J C and Gammie C F 2004 *Astrophys. J.* **611** 977
- [69] Fishbone L G and Moncrief V 1976 *Astrophys. J.* **207** 962–76
- [70] Kreiss H and Oliger J 1973 *Methods for the Approximate Solution of Time Dependent Problems (Global Atmospheric Research Programme (GARP): GARP Publication Series vol 10)* (Geneva: International Council of Scientific Unions, World Meteorological Organization)
- [71] Sadowski A, Narayan R, Tchekhovskoy A, Abarca D, Zhu Y and McKinney J C 2015 *Mon. Not. Roy. Astron. Soc.* **447** 49–71
- [72] van Leer B 1977 *J. Comput. Phys.* **23** 276