Chapter 6
Einstein Toolkit

Samuel Cupp, Steven R. Brandt, Lucas T. Sanches, Roland Haas, Leonardo R. Werneck, and Philipp Mösta

Abstract The Einstein Toolkit is one of the oldest open-source, freely-accessible, community-driven code frameworks for studying fully general relativistic systems such as black holes, neutron stars, and supernovae. The toolkit was developed by an international collaboration of students and professors working at many institutions. The toolkit is built in layers, offering generic computational infrastructure components at its lowest level—e.g., adaptive mesh refinement, method of lines, and I/O—and base modules which provide commonly-defined variables for an evolution system at a higher level. The base level provides a communication interface for more specific spacetime formulations, hydrodynamical schemes, and analysis tools like horizon finders, gravitational wave extractors, ejecta trackers, and others.

S. Cupp
Department of Physics, University of Idaho, Moscow, USA
e-mail: scupp1@my.apsu.edu

S. R. Brandt
Center for Computation and Technology, Luoisiana State Univesity, Baton Rouge, USA
e-mail: sbrandt@cct.lsu.edu

L. T. Sanches
Programa De Engenharia De Sistemas E Computação, Universidade Federal Do Rio De Janeiro (UFRJ), Rio De Janeiro, Brazil

R. Haas (✉)
National Center for Supercomputing Applications, Univesity of Illinois, Champaign, USA
e-mail: rhaas@illinois.edu

L. R. Werneck
Department of Physics, University of Idaho, Moscow, USA

P. Mösta
GRAPPA, Anton Pannekoek Institute for Astronomy and Institute of High-Energy Physics, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands
e-mail: p.moesta@uva.nl

C. Bambi et al. (eds.), *New Frontiers in GRMHD Simulations*, Springer Series in Astrophysics and Cosmology, https://doi.org/10.1007/978-981-97-8522-3_6

6.1 Cactus/Carpet

The Einstein Toolkit (ET) [37] is based on the Cactus code [25, 49], originally developed at the NCSA and the Max Planck Institute for Gravitational Physics by Paul Walker and Joan Massó. Cactus has been used on high-performance computing (HPC) systems for nearly three decades while being continuously updated to support new hardware technologies [89]. During all this time, it has been regularly used for production-level science involving complex astrophysical simulations of colliding black hole (BH) [44, 53, 54], colliding neutron star (NS) [27, 48, 66, 84, 88, 106, 108, 109], supernovae (SN) [74], and cosmological simulations [17, 65] on some of the largest supercomputers in the world [72].

6.1.1 Basic Infrastructure

At its core, Cactus is a code framework designed to facilitate solving of partial differential equations (PDE) using finite difference and finite volume methods. Cactus itself only implements the connective tissue required to support these features, with additional modules implementing all other features. It provides a minimal infrastructure called the *flesh*, which acts like a kind of glue to bind the various computational infrastructure and science modules together. Following the biological metaphor provided by its name, these modules are called *thorns*. Indeed, the flesh does not even provide memory allocation for its arrays, delegating that task to a "driver thorn." A key benefit to the interfaces established by Cactus is that its library of science thorns was developed independently of numerical and infrastructure thorns (examples include time integration via the MoL thorn [99], HDF5 output, and checkpointing).

One of the key functionalities provided by the flesh is a basic workflow scheduler. Looking at the scheduler from a high level, Cactus "schedules" C, C++, and Fortran routines in specific "scheduling bins" that represent different phases in the simulation workflow. Cactus begins by running routines in setup and initial data bins, continues by running the scheduled functions in its evolution bins in a loop until a terminating condition is reached (such as number of iterations, final simulation time or compute walltime), and finally executes the scheduled functions in its shutdown and analysis bins. By default, Cactus provides a standard set of schedule bins, but new bins can be defined by thorns to be used by other thorns.

Cactus also provides users with an "inheritance" mechanism. A thorn's variables and functions can be marked as publicly available. Other thorns can then import these variables and routines from their parent thorns, using and modifying them. This mechanism allows for the implementation of "interface-like" thorns, that provide a common way of doing a computation or storing variables.

Since 2003, Carpet [91] has been the principle driver used by Cactus. It provides access to full Berger-Oliger adaptive mesh refinement (AMR), data distribution, communication through MPI, and memory management.

6.1.2 Reproducibility of Results and Regression Testing

Whenever new thorns are added to the ET, tests are provided to ensure the reproducibility of results obtained. These tests are run and checked rigorously as part of the regular release schedule of the ET (every six months, typically in May and November). Because reproducibility is such a high priority for this framework, thorn Formaline is typically compiled into each executable. Formaline captures all details about a simulation, including local edits in the source code that developers may have forgotten to commit and push to a repository and embeds them as a tar-archive in the executable. Thus, simulations performed with the ET retain a snapshot of the source code and compilation options used. Formaline also collects runtime information and records all runtime parameters used during the simulation.

6.2 Thorn Library Overview

The ET contains a wide variety of scientific modules. Support for curvilinear coordinates is provided by the Llama [81] code, general relativistic magnetohydrodynamics (GRMHD) by either the GRHydro [61, 71] or IllinoisGRMHD [39], and spacetime evolution by McLachlan [24], Baikal [87], BaikalVacuum [87], and lean_public [95, 107] thorns. McLachlan uses code automatically generated and optimized by the Kranc [55] code generator which includes loop-tiling, vectorization, and kernel splitting to ensure that compute kernels fit into L1 cache. Similarly, Baikal and BaikalVacuum use code generated by the NRPy+ [86] code generator, which takes symbolic expressions in Python and converts them to highly optimized C code using SIMD, common subexpression elimination (CSE), and other techniques. lean_public uses manually coded Fortran subroutines, providing a very straightforward implementation of the evolution equations.

GRHydro and IllinoisGRMHD support a variety of tabulated and analytical equations of state. These are implemented by the EOS_Omni and GRHayLib thorns, respectively. Both thorns support a tabulated equation of state (EOS), which imports a table of data for the different hydrodynamic quantities like pressure and specific internal energy, among others. Usually, these quantities are known for some values of the fluid density, temperature, and electron fraction, but one can determine them at any point that is within the bounds of the table data via interpolation. The analytic EOS approximates these tables by defining the pressure as a function of the density. While a simple polytropic EOS can be used, both thorns also provide implementations of the more accurate piecewise polytropic EOS, which more closely matches the table data at fixed temperature and electron fraction.

Despite its imperfections, Cactus has had enormous success in uniting the numerical relativity community in achieving science goals for several decades. It has won awards [2], been the basis of SPEC [1] benchmarks, and spawned countless Ph.D. theses and scientific papers.

Detailed descriptions and documentation for each module are included in the source distribution [30, 61].

6.2.1 Attribution of Credit

One difficulty of open-source projects is the proper attribution of credit, particularly with regard to citations. Instead of relying on a static publication for giving credit to the people who contribute to the ET, a Zenodo DOI is provided. With each release of the ET, new authors and contributors are added, and the Zenodo reference is updated. As of this writing, the current DOI is `doi:10.5281/zenodo.10380404`. This allows for newer contributors to be added to the ET's author list, so that all thorn authors are properly acknowledged for their contributions.

6.2.2 Base Modules

ET thorn variables are documented to be used within a system of units such that the solar mass, the speed of light, and the gravitational constant are all set to one (i.e., $G = c = M_\odot = 1$).

The ADMBase thorn provides grid functions for the spatial 3-metric γ_{ij}, the unit normal of the spacelike slices of the 4D spacetime manifold $n^\mu = (\alpha^{-1}, \beta^i/\alpha)$, the lapse function α, the shift vector β^i, and the extrinsic curvature associated with the embedding k_{ij} in the Arnowitt-Deser-Misner (ADM) formulation [12]. In addition, it also provides a grid function for the determinant of the spatial metric $\gamma = \det(\gamma_{ij})$.

Similar to ADMBase, the HydroBase thorn provides a definition for the fluid rest-mass density ρ, the pressure P, the specific internal energy density ϵ, the contravariant three-velocity in the fluid rest frame v^i, the electron fraction Y_e, the temperature T, the specific entropy s, and the contravariant magnetic field vector B^i.

Finally, the TmunuBase thorn defines the components of the 4D energy-momentum tensor $T_{\mu\nu}$, as well as scheduling bins to determine when and how to calculate $T_{\mu\nu}$.

These "Base" modules make it possible to couple various independently coded initial data (ID) schemes, spacetime solvers, and hydrodynamical schemes, providing a great deal of generality and computational flexibility.

6.2.3 CarpetRegrid2

Carpet is capable of full Berger-Oliger structure mesh refinement with arbitrary shapes for the refined regions and subcycling in time. For compact object merger simulations however, the regions of the simulation domain that require high resolution, can be computed based on the known location of the objects themselves. This results in a simpler regridding scheme, less reliant on ad-hoc prescriptions for a refinement criterion, a result also reported in [9]. To provide this "steered" mesh refinement, Carpet provides the CarpetRegrid2 thorn, which allows users to define sets of nested refined regions that can dynamically change in size, location and depth of nesting during the evolution. Coupled with object tracking capabilities provided by PunctureTracker, which tracks black holes, and VolumeIntegrals_GRMHD, which tracks NS and other material objects this allows for efficient refinement around the region of interest while ensuring simple geometries with low surface to volume ratio, thus limiting communication overhead.

6.2.4 Coordinate Systems/Llama

One of the other major concerns for any general relativistic code is the coordinate system employed for discretizing and evolving a set of PDEs. Depending on the needs of the programmer and the physical problem in question, one may use either 2 or 3 dimensions. In the former case, Cactus offers support for using 3D Cartesian Coordinates in "2D" through a thorn called Cartoon2D. Based on a concept originally proposed by Paul Walker, Cartoon2D fills in ghost zone values by rotations (see [8] for further information).

For large scale 3D simulations, the use of non-Cartesian coordinate systems is often employed. Such systems can offer advantages over Cartesian ones. Certain problems are naturally better-suited for certain coordinate systems. In the early days of numerical relativity, Čadež coordinates [105], which naturally described two spheroidal throats and a spheroidal outer boundary, helped the first collisions of black holes on a computer to succeed. Similarly, fisheye coordinates were used by the early inspiral black hole codes (before AMR was used).

An advantage of non-Cartesian coordinate systems becomes apparent when one desires to obtain the gravitational wave (GW) signal output from a numerical simulation. In order to do so, it is necessary to extract the wave signal very far away from the physical system producing the spacetime perturbations. In practice, this amounts to placing the outer boundaries of the simulation domain very far away from the center of interest in order to minimize the influence from the artificial outer boundary conditions that must be imposed in any simulation on the extracted wave signal. Using a spherical coordinate system, for instance, disentangles the angular and radial resolutions of a domain, allowing them to extend further away with smaller memory requirements when compared to their Cartesian counterpart.

Implementing non-Cartesian coordinate systems without introducing new coordinate singularities is paramount for any numerical simulation code, otherwise all potential advantages derived from such coordinate systems may nullified (but see [69] for a scheme to handle certain types of coordinate singularities in the simulation grid). These coordinate singularities can be avoided by decomposing the coordinate system in question into multiple coordinate "patches," each covering a piece of the complete domain of interest. As an example, let us consider spherical coordinates, which are known to be singular at the poles. It is possible to remove these singularities and cover the entire surface of the sphere by utilizing 6 coordinate patches, each covering a section of the whole domain. To visualize how such a system can be constructed, picture a malleable cube in 3D which is uniformly filled with air. Each of the six faces of the cube will round as the cube inflates and turn into spherical sections. The union of these 6 sections can uniquely describe the surface of a sphere without any coordinate singularities. Given the nature of the theory of General Relativity, decomposing a spacetime (a 4D manifold) into multiple sub-regions, each described by their own coordinate systems is mathematically very sound and can be easily done.

In the ET, when utilizing the Carpet driver, the Llama [81, 84] thorn allows users to utilize multiple coordinate patches for achieving non-Cartesian simulation domains. Most often, Llama is used on spherical domains, both with either spherical hole or Cartesian "core."

Figure 6.1 is a schematic representation of the currently supported multipatch coordinate systems. The figure represents a constant 2D slice of the simulation domain where each patch is colored differently. The Llama thorn supports spherical coordinates with a spherical core (left panel) and spherical coordinates with a Cartesian core (right panel). Llama handles the spherical-Cartesian inner boundary by making sure that the Cartesian core and the spherical patches overlap. Ghost zones are filled by interpolating values between patches.

From the user's perspective, using the aforementioned thorns involves simply modifying the derivative operators used (due to technical details on how tensor components are stored). Generally speaking, users interested in coding a spacetime or hydrodynamics formulation need not be concerned about multiple patches, AMR regions, or parallelization. In each of these cases, the Cactus infrastructure calls their code for a logical box, a 2 or 3-dimensional array with start and end indexes,

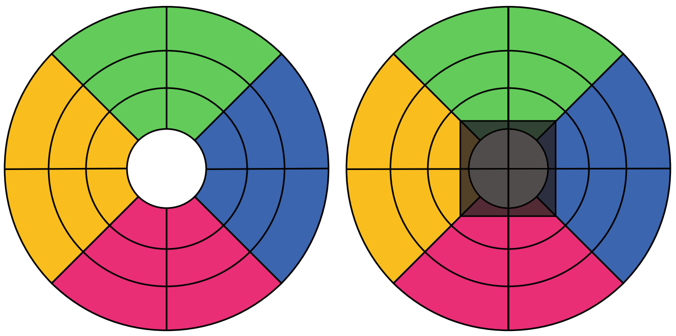

Fig. 6.1 Multipatch coordinates currently supported in the ET. Left panel: Spherical patches with spherical inner boundary. Right panel: Spherical patches with spherical inner boundary and an overlapping Cartesian core

grid spacings and other required information. For an example of GRMHD simulations utilizing multiple grid patches with Llama, see for instance [84].

6.2.5 Time Stepping

The actual time integration (e.g., RK4 and other Runge-Kutta schemes, Iterative Crank-Nicholson, Adams-Bashforth, multi-rate Runge-Kutta schemes, etc.) of any Cactus simulation is implemented by the MoL (Method of Lines) thorn. To use MoL, the user writes a small C routine which registers the variables to be evolved and the corresponding time-derivative of that value (which MoL calls the RHS value). One then schedules a routine to run in the MoL_CalcRHS bin to fill in the RHS values.

Therefore, new formulations of the hydrodynamical quantities can be created by (1) implementing a thorn which schedules a new function in MoL_CalcRHS, and (2) providing routines in the MoL_PostStep bin to communicate needed information between the thorn-specific evolution variables and the variables in ADMBase, HydroBase, and TmunuBase.

To implement a new spacetime module, one would need (in addition to writing and scheduling the RHS function) to provide a way to convert any new variables to the ADM variables.

Users choose which timestepping method to use via runtime parameters, based on the specific needs of the simulated system.

In either case, users benefit from Cactus's time integration, I/O modules, checkpointing, and AMR capabilities. They do not need to constantly re-implement these core functionalities, but can instead rely on the already existing, well-tested, and optimized implementations. Similarly, all modules using MoL immediately have access to any new time stepping scheme implemented in MoL, showcased, for example, by the introduction of the Adams-Bashforth multistep method, or, with some extra declarations, the multi-rate Runge-Kutta schemes.

6.2.6 Spacetime Evolution

The ET contains multiple alternative modules to evolve spacetime using the Baumgarte-Shapiro-Shibata-Nakamura (BSSN) and ccZ4 formulation of numerical relativity. McLachlan [24], Baikal [87], and lean_public [95, 107] are included in ET by default, and users can switch between them at runtime. The modular nature of the ET makes it possible to use other custom spacetime evolution modules. For example, the CTGamma [81] thorn is optimized to use curvilinear, multi-patch coordinates employing the Z4c formulation using Llama.

McLachlan, and lean_bssn both support curvilinear coordinates provided by Llama, and all codes use NewRad to implement approximate radiative boundary conditions at the outer boundary of the simulation domain.

Typically ID for spacetime metric and gauge quantities is either provided along with ID for fluid quantities, or, in the case a vacuum binary black hole (BBH) simulations by TwoPunctures [10], which constructs ID for a BBH system with arbitrary spin using the Brandt-Bruegmann method [21] in the Bowen-York formulation [20].

To support the modularity of the ET, ID is provided through a set of spacetime variables set based on the ADM decomposition, which can then be used by any of these evolution thorns regardless of the specific formulation. Similarly, these ADM variables are used to communicate between the evolution thorns and any analysis and fluid evolution thorns.

6.3 Hydrodynamics

Following the Valencia Formulation [11, 16, 67], all the relativistic hydrodynamic thorns use a combination of the primitive variables

$$\boldsymbol{P} = \begin{bmatrix} \rho \\ \varepsilon \\ T \\ P \\ v^i \\ B^i \\ Y_\mathrm{e} \\ S \end{bmatrix}, \tag{6.1}$$

and conserved variables

$$\boldsymbol{C} = \begin{bmatrix} \tilde{D} \\ \tilde{\tau} \\ \tilde{S}_i \\ \tilde{B}^i \\ \tilde{Y}_\mathrm{e} \\ \tilde{S} \end{bmatrix} \equiv \sqrt{\gamma} \begin{bmatrix} D \\ \tau \\ S_i \\ B^i \\ DY_\mathrm{e} \\ WS \end{bmatrix} \equiv \sqrt{\gamma} \begin{bmatrix} W\rho \\ n_\mu n_\nu T^{\mu\nu} - D \\ -n_\mu T^\mu_i \\ B^i \\ DY_\mathrm{e} \\ WS \end{bmatrix} = \sqrt{\gamma} \begin{bmatrix} W\rho \\ \alpha^2 T^{00} - D \\ \alpha T^0_i \\ B^i \\ DY_\mathrm{e} \\ WS \end{bmatrix}, \tag{6.2}$$

to evolve the system. The conserved variables $\boldsymbol{C}$ are an alternative representation of the primitive variables $\boldsymbol{P}$ that are better-suited for hyperbolic evolution.

In (6.1), ρ is the baryon density, P is the pressure, ε is the specific internal energy, T is the temperature, v^i is the fluid 3-velocity, Y_e is the electron fraction (possibly many such composition variables exist for more complex EOS), S is the entropy per baryon, and B^i are the spatial components of the magnetic field (B^μ). The fluid 3-velocity is related to the 4-velocity u^μ by

$$v^i = \frac{u^i}{W} + \frac{\beta^i}{\alpha}.$$

The frame of B^μ is that of normal (or Eulerian) observers with 4-velocity n^μ and is normal to the spatial hypersurface ($B^\mu n_\mu = 0$). The set of primitive variables is not fully independent, being related to each other through the EOS, for example, $P = P(\rho, \varepsilon)$ or $P = P(\rho, T, Y_e)$ depending on the EOS. Different codes may make different choices as to which primitive variables are considered independent and which ones are considered dependent variables, possibly even based on the temperature, density or magnetization regime the fluid is in.

In (6.2), D is the rest-mass density, τ is the energy variable, and $W = \alpha u^0 = \left(1 - \gamma_{ij} v^i v^j\right)^{-1/2}$ is the Lorentz factor. The GRMHD energy-momentum tensor, $T^{\mu\nu}$, is given by

$$T^{\mu\nu} = \left(\rho h + b^2\right) u^\mu u^\nu + \left(P + P_{\text{mag}}\right) g^{\mu\nu} - b^\mu b^\nu \,, \tag{6.3}$$

where $h = 1 + \epsilon + P/\rho$ is the specific enthalpy, and $P_{\text{mag}} = b^2/2$, $b^2 = g_{\mu\nu} b^\mu b^\nu$. For the magnetic field in the lab frame associated with unit normal n^μ, we choose the definition

$$B^\mu = \frac{1}{\sqrt{4\pi}} n_\nu {}^*\!F^{\mu\nu}. \tag{6.4}$$

where ${}^*\!F^{\mu\nu}$ is the dual of the Faraday tensor. Using this definition, $b^\mu = \frac{1}{\sqrt{4\pi}} u_\nu {}^*\!F^{\mu\nu}$ is the magnetic field in the co-moving frame u_μ. This gives the relation

$$b^0 = \frac{u_i B^i}{\alpha}, \tag{6.5}$$

$$b^i = \frac{B^i + B^j u_j u^i}{W}, \tag{6.6}$$

which completes our prescription for the magnetohydrodynamics (MHD) fields. The evolution equations are then given by the equation of rest-mass conservation, conservation of energy-momentum, and Maxwell's equation:

$$\begin{aligned} (D u^\mu)_{;\mu} &= 0 \\ T_\mu{}^\nu{}_{;\nu} &= 0 \\ {}^*\!F^{\mu\nu}{}_{;\nu} &= 0, \end{aligned} \tag{6.7}$$

where a semicolon indicates covariant derivatives with respect to $g_{\mu\nu}$.

At this point, we note that some of the evolution equations are optional and dependent on the EOS chosen to model nuclear matter. The electron fraction is only used when using a finite-temperature EOS, and $\tilde{Y}_e$ is not evolved when using a polytropic hybrid or simple gamma-law EOS. Similarly, entropy evolution is entirely optional and usually only used for some conservative-to-primitive recovery schemes.

6.3.1 Initial Data

To begin a simulation, we first initialize $\boldsymbol{P}$ using one of the ID thorns. Many options exist for this in the ET depending on the system of interest. Most ID solvers are outside of and independent of the Cactus framework. They are standalone programs for which specialized Cactus importer thorns exist.

Cactus typically is able to install a version of the initial data tool using its "External Library" mechanism. This is something like a very simple package manager designed to help install (or link to) Cactus dependencies.

6.3.1.1 TOVSolver

The Tolman-Oppenheimer-Volkoff (TOV) star [77, 103] is a spherically symmetric solution to the Einstein Equations that has, historically, been used to constrain the possible equations of state for NSs. It remains useful as a test of the various 3D machinery required to evolve more complex systems, and as a toy problem for students and workshops. Our standard tutorial used in our portal [23] and workshops involves a short simulation of a TOV star at low resolution.

6.3.1.2 Hydro_RNSID

The Rotating Neutron Star ID thorn uses the Komatsu-Eriguch-Hachisu (KEH) method [60] to construct a NS with a single killing vector. As such, this thorn generates 2D rotationally symmetric data. The RNSID solver is, therefore, a slightly more complex data set than the TOV star and is also useful for testing. Beyond testing evolution codes, single rotating NS are also used to study EOS, and to serve as a building block to construct NS-disk ID. This solver can be run within Cactus, or as a standalone utility that generates data files to be imported by Cactus at a later time.

6.3.1.3 FishboneMoncriefID

The Fishbone-Moncrief Initial Data creates ID for a black hole surrounded by a polytropic disk. Input parameters to the solution include the radius at which the density of the disk is at its maximum as well as the polytropic parameters. The code for this dataset is generated using NRPy, a Python based code generator. It also has a random noise parameter for use in testing.

The solver follows the prescription outlined in [45], describing a way to produce axisymmetric, isentropic disks with constant angular momentum per baryon.

6.3.1.4 LORENE/Meudon

LORENE is perhaps the oldest tool used to generate ID for BBH or binary neutron star (BNS) mergers [18]. LORENE uses spectral methods to generate data files which ET is able to import using the Meudon thorns Meudon_Bin_NS, Meudon_Bin_BH, and Meudon_Mag_NS.

Several options exist to initialize magnetic fields (or the vector potential) for a given hydrodynamic setup. The Meudon_Mag_NS thorn can directly set up both the hydrodynamic and magnetic variables for a single star. Seed_Magnetic_Fields provides several vector potential ID configurations for both single and binary NSs.

LORENE generally supplies a cold equation of state. This is fine for the ID when the NSs are widely separated. Later, during the merger, thermal effects can become important.

6.3.1.5 SGRID/DNSdata

SGRID is a similar to LORENE in purpose, but it can generate data for more extreme corners of the NS collision parameter space, e.g. total mass of $\approx 3.4M_{\odot}$, highly spinning and precessing systems close to breakup, high mass ratio systems for soft EOSs, elliptical orbits, etc. SGRID [101] solves the conformal thin-sandwich equations using pseudo-spectral methods, so it has exponential convergence on its patchwork of grids.

The DNSdata thorn was designed by the creators of SGRID to read SGRID-generated ID files into Cactus and facilitate the use of SGRID for BNS collisions in Cactus.

6.3.1.6 KadathThorn/KadathImporter

The Kadath thorns are an interface for using files generated by the FUKA ID solver [50, 79]. The Frankfurt University/Kadath solver is capable not only of generating BNS data, but also black hole/neutron star (BHNS) binaries. In all these cases, Kadath can, like SGRID, solves for NSs with tabulated equations of state and spins near the mass-shedding limit.

A limitation is that the Kadath thorns require a strict z-symmetry, so spins can only be aligned or anti-aligned with the orbital axis.

6.3.1.7 FLRWSolver

FLRWSolver [64] constructs ID suitable for a dust-filled spatially flat Friedman-n-Lemaître-Roberston-Walker (FLRW) cosmological spacetime evolved in full general relativity. Optionally, a linear perturbation can be seeded to study its linear and non-linear growth.

6.3.1.8 Vacuum Initial Data

A number of analytic spacetime solutions are provided by the Exact, and EinsteinExact thorns, useful to both test evolution codes and as building blocks for BH-disk, or similar, systems. TwoPunctures [10] and NRPyEllipticET [13] construct ID for BBH, with arbitrary spin and linear momentum values. The former also accepts a matter source to construct constraint satisfying ID data in presence of (prescribed) matter. Finally, NRPyEllipticPN [51] implements post-Newtownian (PN) expressions to compute parameters resulting in low-eccentricity BBH inspiral simulations.

6.3.2 Equation of State

The thorn EOS_Omni can be used to implement the equation of state. It is capable of modeling cold or hot EOSs. The thorn assumes nuclear statistical equilibrium with rest-mass density ρ, specific internal energy ϵ (alternatively, temperature T), and electron fraction Y_e as independent variables. The equation of state type can be specified as a polytropic, Γ-law, piecewise-polytropic with thermal part, cold tabulated with thermal part, hot tabulated, and barotropic tabulated EOS. Tables of some EOS are available at [6].

GRHayL contains its own independent EOS routines supporting Γ-law, piecewise-polytropic with thermal part, and hot tabulated EOS using the same tables as EOS_Omni. The details of this library and its implementation in the ET are included in the description of GRHayL and the new GRHayL-based IllinoisGRMHD in Sect. 6.6.

RePrimAnd [59] provides a complete EOS framework and conservative to primitive inversion method with strong guarantees on the existence and validity of solutions. A version is included in ET but is not currently in use due to various technical issues. It supports non-magnetized fluids (a version suitable for magnetized fluids exists [57]) described by a Γ-law, or cold tabulated EOS with a thermal part.

One can use these in conjunction with GRHydro or GRHayL to continue the evolution of NS ID created using such tools as LORENE, SGRID, or FUKA.

6.3.3 GRMHD Evolution

After initializing the primitives and spacetime quantities, the conserved variables are calculated. During each step of the evolution, the primitive variables are used to compute the RHS values used to numerically integrate the evolution equations.

The basic workflow for computing the time derivatives of the evolved quantities is depicted in Fig. 6.2 (adapted from [4]).

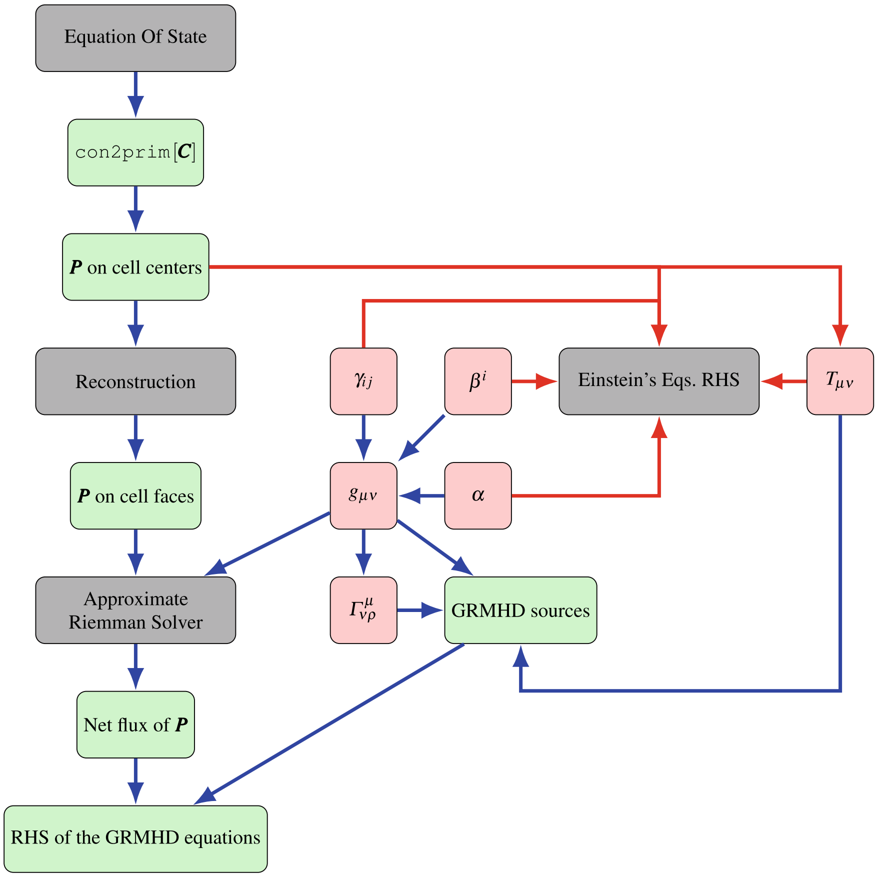
Equation Of State
con2prim[$\boldsymbol{C}$]
$\boldsymbol{P}$ on cell centers
Reconstruction
γ_{ij}
β^i
Einstein's Eqs. RHS
$T_{\mu\nu}$
$\boldsymbol{P}$ on cell faces
$g_{\mu\nu}$
α
Approximate Riemman Solver
$\Gamma^{\mu}_{\nu\rho}$
GRMHD sources
Net flux of $\boldsymbol{P}$
RHS of the GRMHD equations

Fig. 6.2 Diagram depicting the conceptual workflow of a GRMHD code. In the figure, elements outlined by green boxes denote GRMHD variables, red boxes denote Einstein's equations' components required for the GRMHD evolution and gray boxes denote functional steps in the GRMHD workflow. Concurrently, blue arrows denote data flow for computing the GRMHD RHS while those depicted by red boxes and arrows pertain to relevant segments of the spacetime evolution scheme

In an initial step, the force terms appearing in the 3+1 decomposed energy-momentum conservation equation (6.7) are computed from the spacetime metric and Christoffel symbols, setting the stage to compute the numerical fluxes for the fluid variables. The primitive variables $\boldsymbol{P}$ are then computed from the conserved variables $\boldsymbol{C}$, which generally involves EOS-specific methods using iterative solvers. This step is in general only second order convergent, since all GRMHD codes in ET (except WhiskyTHC and Sprtiz which employ higher order finite *difference* schemes, but are not shipped as part of ET) do not distinguish between the cell-centered value of $\boldsymbol{P}$ and their cell averages. After primitive recovery, the values of the recovered primitive variables are then reconstructed to cell faces, where they form the input to a Riemann problem that is used to compute numerical fluxes. Finally, the net fluxes are used to compute an average rate of change for the conserved variables that is passed to the time stepper. It is worth noting that, when used with the time steppers provided by thorn MoL, this averaged rate of change needs only be computed to first order to achieve the full convergence order of the time stepper.

Simultaneously the fluid variables are used to compute the energy-momentum tensor $T_{\mu\nu}$, which is used as a source term by the spacetime evolution thorns.

6.4 Non-Einstein Toolkit Hydrodynamic Codes Leveraging the Einstein Toolkit

Before describing the tools available within the official distribution of the ET, we would like to point out that there are other GRMHD codes which use the ET framework but have never become part of the ET. These include WhiskyTHC [82], Spritz [28] and FIL [70]. WhiskyTHC is available under the GPLv3 license and provides advanced microphysics. It has been used for NS collisions [82, 83]. Spritz [46], which borrows some ideas from Whisky, is available under the Creative Commons Attribution 4.0 International license. In order to study gamma-ray bursts resulting from BNS, this code employs a tabulated and temperature-dependent equation of state to describe NS matter as well as neutrino emission and absorption. FIL is the private "Frankfurt Illinois" code derived from IllinoisGRMHD and supports realistic microphysics and neutrino cooling using high-order numerical methods.

As these exist outside the ET umbrella, their testing and validation methods vary from group to group and do not leverage the ET's continuous integration (CI) testing framework. That does not mean that these codes are not tested or maintained by their original developers. However, the standards of support and testing of these other codes depend on the needs of their developers for their own published research projects. The existence of these codes helps to illustrate the strength of the open source software model for science in general as well as the vibrancy of the ET community in particular.

The creation of these codes is facilitated by the open source license of the ET. When creating codes based on the ET, researchers may opt to (and this is frequently the case) create a closed-source code that leverages the Toolkit, keeping it private until they have validated and published results using it. At this stage, researchers then may decide to move their code to open source and petition for inclusion in the ET. It is very important to point out, however, that developers are never *required* to do this.

6.5 GRHydro

The original support for hydrodynamics within the ET was the GRHydro thorn [71], which itself descends from the EU Network on Sources of Gravitational Radiation code, Whisky [14]. It was also the first publicly available, fully GRMHD code [46]. Initially, GRHydro used a relativistic polytropic or Γ-law gas equation of state, with support for tabulated EOS added via EOS_Omni later. In [71], MHD was added to the capabilities of GRHydro. In order to maintain the divergence-free character of the magnetic field, GRHydro implements both constrained transport [47] and hyperbolic divergence cleaning [80] schemes.

The current, open-sourced GRhydro [14, 71] implements GRMHD with a high-resolution, shock-capturing, finite-volume scheme with (e)PPM [29, 68], (W)ENO(-Z) [19, 94, 98], and MP5 [96] reconstruction schemes and HLLE [36, 52], Roe [85], and Marquina [31] approximate Riemann solvers. Conservative to primitive conversion uses a non-linear Newton-Raphson iteration scheme based on [75], though specific applications, for example, SN simulations [72, 73], have instead explored use of the method of [26], and other variations may exist. In combination with the Llama [81] infrastructure, it supports curvilinear coordinates allowing it to handle large simulation volumes at moderate cost [84].

GRHydro evolves the set of conserved variables $\boldsymbol{C}$ in (6.2) using the Valencia formulation. Similar to most existing codes, it usually reconstructs the primitive variables (6.1) to cell interfaces when setting up the Riemann problem computing numerical fluxes, resulting in a second order accurate evolution scheme. By default GRHydro will reconstruct the 3-velocity v^i and specific internal energy ϵ (in addition to rest mass density ρ). For a hot tabulated EOS it may be more appropriate to reconstruct temperature T, and Wv^i instead of v^i to guarantee that the reconstructed speed remains subluminal. Both variants are offered by GRHydro as runtime options.

GRHydro implements the "flux-CT" scheme of [47, 104], which employs a *cell* averaged magnetic field B^i rather than the face-averaged field used in [41]. As a consequence of this choice, the (exactly) conserved divergence operator is an edge-centered operator instead of a cell-centered operator. This choice simplifies implementation in Carpet which lacks full support for face-centered variables. As of this writing, GRHydro does not include methods to preserve the divergence-free condition across mesh refinement boundaries which results in a layer of constraint violating data near the edges of mesh-refinement levels. Due to the constraint preserving nature of the evolution equation, this data remains fixed in space and does not propagate. For SN simulations, where the grid does not move, this is sufficient to ensure that simulations succeed.

As an alternative, GRHydro also implements a hyperbolic divergence cleaning scheme following [80], which introduces an additional dynamic variable ψ that is driven to 0 to transport and damp away any violation of the divergence-free condition. This method is well-suited for AMR simulations since any constraint violation is damped away no matter whether it is sourced by finite resolution effects or by AMR data movement. This method, however, has proven difficult to control in curved spacetime simulations beyond simple TOV star and Bondi accretion test cases. Flat spacetime simulations, which are where the method originates from, are successful.

Beyond compact object simulations, GRHydro has recently been used for cosmological simulations [64] and has been regularly used for core-collapse SNe simulations [73, 74, 78].

6.6 GRHayL/IllinoisGRMHD

IllinoisGRMHD [39] is an open-source GRMHD code originating from the closed-source code developed by the Illinois numerical relativity group [35]. Several modified versions of this code also exist [38, 70, 106] outside of the ET, as different research groups have extended the original code to facilitate their own research interests and needs. Notably, it was the first public code to use the vector potential of the magnetic field as an evolution variable. The original release implements the piecewise parabolic method (PPM) [29], hybrid EOS, and HLLE [36, 52] Riemann solver. The IllinoisGRMHD family of thorns has been used by many groups, primarily for BNS simulations [33–35, 38, 39, 62, 70, 106, 108, 109]. It has also been used for circumbinary disk evolution [42, 43]. A recent development is to base IllinoisGRMHD on a common library of modules (GRHayL) that provides re-usable building blocks for GRMHD codes. In cases where this difference is relevant the newer version of IllinoisGRMHD is dubbed GRHIGM below.

One notable difference between IllinoisGRMHD and other codes in the ET is that it chooses a different primitive velocity. The more common choice is the Valencia 3-velocity v^i, used by HydroBase, but IllinoisGRMHD instead uses the 3-velocity appearing in the induction equation, i.e.,

$$\tilde{v}^i = \frac{u^i}{u^0} = \alpha v^i - \beta^i, \tag{6.8}$$

as its primitive variable. IllinoisGRMHD also evolves the vector potential A_i instead of directly evolving the magnetic field B^i. This has the advantage of ensuring that $\nabla \cdot B^i = 0$ (i.e. no magnetic monopoles) automatically. Additionally, IllinoisGRMHD uses a different definition of B^i than what is given in (6.4). However, GRHIGM uses the definition of (6.4), making the various ET codes now agree in their definition of B^i. IllinoisGRMHD's vector potential evolution method is particularly well suited for simulations involving moving meshes, as happens for example in magnetized BNS merger simulations, for which constraint transport methods that fail to preserve the divergence-free condition across mesh refinement boundaries can rapidly fill the simulation domain with constraint violating data [40].

6.6.1 GRHayL

IllinoisGRMHD's popularity has led to a number of forks implementing advanced functionality, e.g. in the FIL [70] code. However, these improvements often led to mutually incompatible code variants that made community collaboration more difficult. To better support community efforts to improve and extend the code, the core algorithms of IllinoisGRMHD were converted into a C library called the General Relativistic Hydrodynamics Library (GRHayL). GRHayL provides an infrastructure-agnostic, extensible, open-source library of building blocks to simplify the task of extending or improving IllinoisGRMHD's core features. As discussed in Sect. 6.3, there exist several discrete steps to integrate the evolution equation. These steps can be broken into independent operations which are used to compartmentalize the internal features of GRHayL. As such, GRHayL divides its features into Atmosphere, Con2Prim, EOS, Flux_Source, Induction, Neutrinos, and Reconstruction modules.

GRHayL provides all the features of IllinoisGRMHD through functions acting on pointwise or stencil-wise data within its modules. Most modules refer to a specific stage of the evolution procedure and do not depend on each other, allowing users to freely pick and choose which modules to use. However, EOS contains all the code implementing features for specific choices of the EOS and is used by the other modules. Similar to RePrimAnd, it defines a common application programming interface (API) to access the EOS. Codes wishing to use new EOS implement that API, potentially re-using functionality provided by existing EOS implementations. This allows for modular and customizable code behavior.

Starting with the ET_2024_05 "Lev Landau" release of ET, IllinoisGRMHD uses GRHayL, which itself is also included.

6.7 Data Analysis

The ET has a number of built-in thorns for diagnostics and analysis. The most basic of these are the I/O thorns, which include not only 1D, 2D, and 3D data output but also scalar reductions such as the norm, average, maximum, etc. As there are many analysis and diagnostics thorns in the ET, we do not provide an exhaustive list of these thorns but instead provide descriptions of several widely-used analysis thorns. Some of these thorns—such as AHFinderDirect—are focused on black hole physics, but they are also frequently used during matter simulations that can lead to the formation of a black hole, such as BNS mergers.

6.7.1 AHFinderDirect

The AHFinderDirect [100] thorn provides a way to search for trapped surfaces, 2-surfaces within a spacetime that obey the equation:

$$0 = D_i \Sigma^i + K_{ij} \Sigma^i \Sigma^j - K, \tag{6.9}$$

where Σ^i is an outgoing vector on a 2D surface within a 3D slice of the spacetime, and D_i is the covariant derivative associated with γ_{ij}.

These surfaces correspond to regions in the spacetime where outgoing null rays do not expand. They are often used as proxies for event horizons because they are easier to compute, since they can be computed locally rather than starting from future null infinity and working backward. Apparent horizons, if they are present, will be coincident with or inside event horizons, so their appearance provides evidence that a black hole has formed.

Towards the end of any numerical simulation, apparent horizons are expected to settle down to being the same as the event horizon (which is true for the stationary Kerr solution).

Unlike event horizons, e.g. during a black hole collision, apparent horizons can suddenly appear during the course of a simulation. This means that to find them, one most provide some initial guess for their location.

AHFinderDirect discretizes (6.9) on a star-shaped trial-surface directly solving the resulting systems of non-linear equations for the trapped surface's shape using Newton's method starting from the initial guess. This is in contrast to curvature flow based methods [102], which solve a parabolic equation whose steady state solution is the trapped horizon surface. For BBH simulations, initial guesses for the trapped surfaces are provided by users based on initial location, mass and spins of the initial BHs, while initial guesses for parameters for a BH that forms during a collapse are estimated based on the expected mass of the BH and its location at the center of mass of the system.

6.7.2 QuasiLocalMeasures

This module implements the calculation of mass and spin multipoles from the isolated and dynamical horizon formalism [32, 92], as well as a number of other proposed formulae for quasilocal mass, linear momentum, angular momentum, etc. Despite their approximate nature, they have been surprisingly helpful in numerical simulations (see, e.g., [63]), and are therefore an indispensable tool in numerical relativity. QuasiLocalMeasures takes as input a horizon surface (or any other surface that the user specifies, e.g. a large coordinate sphere) and calculates quantities such as the Weyl or Ricci scalars, the 3-volume element of the horizon world tube, as well as other physical observables such as mass and momenta. While initially designed for

BH simulations, it was found in [97] that the methods used by QuasiLocalMeasures can also be used to estimate spins of NSs.

6.7.3 *WeylScal4*

A common output produced by compact object merger simulations is data about the GWs produced during the event. For this the Newman-Penrose scalar Ψ_4

$$\Psi_4 = C_{\rho\sigma\mu\nu}\tilde{m}^\rho \tilde{n}^\sigma \tilde{m}^\mu \tilde{n}^\nu, \tag{6.10}$$

which encodes outgoing radiation contained in the Weyl tensor $C_{\rho\sigma\mu\nu}$ is often convenient. WeylScal4 computes this quantity, as well as the remaining Newman-Penrose components of the Weyl tensor and the curvature invariants. Ψ_4 is computed by separating time (0 index) and space (latin indices)

$$\begin{aligned}\Psi_4 = {} & R_{ijkl}n^i\tilde{m}^j n^k\tilde{m}^l + 2R_{0jkl}(n^0\tilde{m}^j\tilde{n}^k\tilde{m}^l - \tilde{m}^0 n^j\tilde{n}^k\tilde{m}^l) \\ & + R_{0j0l}(n^0\tilde{m}^j\tilde{n}^0\tilde{m}^l + \tilde{m}^0\tilde{n}^j\tilde{m}^0\tilde{\tilde{n}}^l - 2\tilde{n}^0\tilde{m}^j\tilde{m}^0\tilde{n}^l),\end{aligned} \tag{6.11}$$

where $R_{\rho\sigma\mu\nu}$ is the spacetime Riemmann tensor and $\tilde{n}^\mu$, $\tilde{m}^\mu$ are part of a complex valued fiducial null-tedrad defined in [15]. WeylScal4 uses Kranc [55], which generates thorns from Mathematica input, reducing development effort and error rate when transcribing expressions into code.

6.7.4 *Multipole*

Multipole is responsible for performing a mode decomposition of Cactus grid functions via a projection onto the spin weighted spherical harmonics ${}_sY_{\ell m}$. This is extremely useful while extracting GW components of Ψ_4, whose dominant modes are the quadrupolar $\ell = |m| = 2$ spin-weight $s = -2$ modes. Multipole produces output as either ASCII or more compact HDF5 files.

6.7.5 *VolumeIntegrals*

The VolumeIntegrals_GRMHD and VolumeIntegrals_vacuum thorns compute a number of volume integrals of GRMHD and spacetime quantities, respectively. The thorns accept a number of integration volumes consisting of spherical shells, including balls and hollowed out balls. For integrals that compute a location, the integration volume can be set to itself track this location and also track it with a refined region defined in CarpetRegrid2. In this manner, results from some of these integrals can be used to track NSs and BHs, for example.

VolumeIntegrals_GRMHD computes the center of mass

$$X^i = \frac{\int Dx^i dV}{\int DdV}, \tag{6.12}$$

as well as rest mass

$$M_0 = \int DdV \tag{6.13}$$

for each region.

VolumeIntegral_vacuum and ADMMass calculate volume integrals for the ADM mass using

$$M_{ADM} = \frac{1}{16\pi} \int \left(\alpha\sqrt{\gamma}\gamma^{ij}\gamma^{kl}(\gamma_{ik,j} - \gamma_{ij,k})\right)_{,l} dV, \tag{6.14}$$

which follows its definition using a surface integral [76], assuming $\alpha \to 1$ at infinity. VolumeIntegral_vacuum also computes a quantity

$$C^i_\alpha = \int \left[(1-\alpha)^{80}\right] dV \tag{6.15}$$

dubbed “center of lapse”, which, for material objects, has qualitatively the same result as the center of mass.

VolumeIntegral_vacuum and VolumeIntegrals_GRMHD permit regions of the simulation to be excluded from the integral, which allows, e.g., the interior of BHs being masked when computing rest mass.

6.7.6 Outflow

Outflow calculates the flow of rest mass density across a fixed, star-shaped surface, e.g. an apparent horizon or a sphere at fixed radius. Starting from (6.7) the evolution equation for D is integrated over the enclosed volume V and transformed into a surface integral to yield

$$\dot{M}_0 = -\int_{\partial V} \sqrt{\gamma}\alpha D(v^i - \beta^i/\alpha)d\sigma_i, \tag{6.16}$$

where σ_i is the outward-pointing flat-space surface element of the enclosing surface. Additionally, conditions can be used to count only unbound fluid elements that will eventually escape gravity and be ejected from the system. This provides a convenient way to monitor ejecta during BNS mergers, for example.

6.7.7 *Particle_tracerET*

particle_tracerET tracks the motion of test particles that comove with fluid flows. Designed to visualize the evolution of magnetic field lines in ideal fluid simulations, tracers can also be used to create evolution traces of ejecta particles. It evolves each particle's location according to

$$\frac{dX^i}{dt} = \alpha v^i - \beta^i , \tag{6.17}$$

seeding initial particle locations such as to preferentially sample high density regions of the simulation grid.

6.7.8 *SmallbPoynET*

The smallbPoynET thorn computes the fluid four-velocity u^i, the magnetic field measured by a co-moving observer b^μ, and the Poynting vector S^i. While a relatively simple thorn, this provides useful diagnostic and analysis information about the magnetic evolution of the system, and possible regions of jet formation. Currently, this thorn only supports the old IllinoisGRMHD definition of B^i, though this is planned to change in future releases.

6.8 Current and Future Development

The ET has been very successful in enabling numerical astrophysics, resulting in close to 200 publications within the last 4 years. It has provided a stable basis for research by scientists all over the world, incorporating newly developed methods while providing a backwards compatible core set of functionality that allows users to maintain their own codes with minimal effort. Over the past decade, however, computational research has shifted from single socket scalar architectures (what Cactus was originally designed for) to NUMA architectures with vectorization, and GPUs. With the end of Dennard scaling increased parallelization at all levels of the software stack and support for accelerator architectures is crucial to tackle immensely resource intensive GRMHD simulations. These include highly turbulent fluid motion in large, asymmetric domains and realistic treatment of the EOS, and microphysical interaction such as neutrino transport, and nuclear reactions. All of these are required to properly model the physical processes involved in NS collisions such as the "kilonova event" GW170817/GRB 170817A observed in [7].

While there had been attempts to incorporate support for GPUs into ET in the past [89], these have been hampered by the need to maintain full backwards compatibility with existing code and need to restructure the underlying driver thorn Carpet to adequately support GPUs. More recently, efforts have been underway to develop a new driver, CarpetX [90], that is built on the AMReX [110] mesh refinement library developed by the DOE as part of the Exascale Computing Project. AMReX provides the necessary abstraction layers to support both CPU and GPU resources, providing a portability layer for CUDA, RoCM, and oneAPI.

As of this writing, CarpetX is mature enough to be used by early adopters to port codes and benchmark its algorithms on a variety of large CPU and GPU clusters. While functionality that is not immediately required, such as subcycling in time (local time-stepping) are still being developed, the framework is functional for a number of scientific applications such as SN explosion and BNS merger simulations.

There are now 3 independently developed GRMHD codes based on CarpetX [3, 58, 93] that are gearing up for their first production simulations. Each targets different problems and prioritizes different areas of the physics. GRaM-X is designed to simulate core-collapse SNe, targeting very large simulations, with fully energy dependent neutrino transport being among the implementation goals, while dealing with a fairly straightforward geometry requiring no advanced regridding schemes or complicated domain decomposition. AsterX, on the other hand, targets BNS merger simulations (for which an energy averaged neutrino transport scheme may be sufficient) but has to contend with two moving objects and ejecta whose geometry cannot be predicted in advance. AsterX implements the advanced primitive variable recovery scheme of RePrimAnd [59]. Finally, GRHayL, which currently only supports CPU based evolution, provides a method for preserving core code for reuse in new infrastructures, providing infrastructure-agnostic building blocks that incorporate scientific code modules and can be combined with architecture-dependent infrastructure modules. This also simplifies "swapping out" of individual modules for more advanced ones, without having to re-design the evolution code from scratch.

Major improvements [56] in the microphysics support in CarpetX-based codes will be implemented in the upcoming years, along with support for more realistic equations of state [5], and more initial data readers for advanced initial data solvers.

The ET has heavily relied on automated code generation [55, 87] for the Einstein field equations and other tensorial expressions. CarpetX provides enhanced correctness checking capabilities that a CarpetX aware version of the NRPy+ code generator can target to catch common errors and mistakes [22], beyond what is possible in Carpet.

In the ET the Llama thorns provide support for multi-patch domains with curvilinear coordinates used in each patch. They currently only support overlapping patch boundaries, which can be a source of numerical noise and complicate implementing refluxing schemes that guarantee conservation of rest mass and other evolved quantities between patches. For CarpetX, the CapyrX thorn is being developed to provide this functionality while also simplifying the process of adding new types of patch sets. Once fully implemented, CapyrX will support both interpatch boundary communication via interpolation or via characteristic variables exchanges.

While the list of thorns supporting CarpetX is currently much shorter than those using the Carpet driver, their number is steadily growing and the transition to a new driver also offers an opportunity to correct past design decisions that are hindering current research due to changing architecture constraints.

In conclusion, the ET provides a rich, mature software ecosystem for astrophysical GRMHD simulations that has stood the test of time. With CarpetX there is a leap forward in performance and level of support for modern GPU accelerators, ensuring continued high performance of ET-based simulations on the largest compute cluster and targeting simulations at the forefront of scientific research.

Acknowledgements The ET has been supported by NSF 2004157/2004044/2004311/2227105/2004879/2003893/2114582. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

1. 507.cactuBSSN_r. SPEC CPU®2017 benchmark description. https://www.spec.org/cpu2017/Docs/benchmarks/507.cactuBSSN_r.html
2. Awards won by cactus and the Einstein toolkit. https://www.cactuscode.org/media/prizes/index.html
3. GRHayL/GRHayLET: Einstein Toolkit thorns which require GRHayL. https://github.com/GRHayL/GRHayLET.git
4. GRMHD workflow: 10.5281/zenodo.10988388. https://zenodo.org/records/10988388
5. Modular unified solver of the equation of state. https://musesframework.io/
6. stellarcollapse.org. https://stellarcollapse.org/
7. B.P. Abbott, R. Abbott, T.D. Abbott, F. Acernese, K. Ackley, C. Adams, T. Adams, P. Addesso, R.X. Adhikari, V.B. Adya, C. Affeldt, M. Afrough, B. Agarwal, M. Agathos, K. Agatsuma, N. Aggarwal, O.D. Aguiar, L. Aiello, A. Ain, P. Ajith, B. Allen, G. Allen, A. Allocca, P.A. Altin, A. Amato, A. Ananyeva, S.B. Anderson, W.G. Anderson, S.V. Angelova, S. Antier, S. Appert, K. Arai, M.C. Araya, J.S. Areeda, N. Arnaud, K.G. Arun, S. Ascenzi, G. Ashton, M. Ast, S. M. Aston, P. Astone, D.V. Atallah, P. Aufmuth, C. Aulbert, K. AultONeal, C. Austin, A. Avila-Alvarez, S. Babak, P. Bacon, M.K.M. Bader, S. Bae, M. Bailes, P.T. Baker, F. Baldaccini, G. Ballardin, S.W. Ballmer, S. Banagiri, J.C. Barayoga, S.E. Barclay, B.C. Barish, D. Barker, K. Barkett, F. Barone, B. Barr, L. Barsotti, M. Barsuglia, D. Barta, S.D. Barthelmy, J. Bartlett, I. Bartos, R. Bassiri, A. Basti, J.C. Batch, M. Bawaj, J.C. Bayley, M. Bazzan, B. Bécsy, C. Beer, M. Bejger, I. Belahcene, A.S. Bell, B.K. Berger, G. Bergmann, S. Bernuzzi, J.J. Bero, C.P.L. Berry, D. Bersanetti, A. Bertolini, J. Betzwieser, S. Bhagwat, R. Bhandare, I.A. Bilenko, G. Billingsley, C.R. Billman, J. Birch, R. Birney, O. Birnholtz, S. Biscans, S. Biscoveanu, A. Bisht, M. Bitossi, C. Biwer, M.A. Bizouard, J.K. Blackburn, J. Blackman, C.D. Blair, D.G. Blair, R.M. Blair, S. Bloemen, O. Bock, N. Bode, M. Boer, G. Bogaert, A. Bohe, F. Bondu, E. Bonilla, R. Bonnand, B.A. Boom, R. Bork, V. Boschi, S. Bose, K. Bossie, Y. Bouffanais, A. Bozzi, C. Bradaschia, P.R. Brady, M. Branchesi, J.E. Brau, T. Briant, A. Brillet, M. Brinkmann, V. Brisson, P. Brockill, J.E. Broida, A.F. Brooks, D.A. Brown, D.D. Brown, S. Brunett, C.C. Buchanan, A. Buikema, T. Bulik, H.J. Bulten, A. Buonanno, D. Buskulic, C. Buy, R.L. Byer, M. Cabero, L. Cadonati, G. Cagnoli, C. Cahillane, J. Calderón Bustillo, T.A. Callister, E. Calloni, J.B. Camp, M. Canepa, P. Canizares, K.C. Cannon, H. Cao, J. Cao, C.D. Capano, E. Capocasa, F. Carbognani, S. Caride, M.F. Carney, G. Carullo, J. Casanueva Diaz, C. Casentini, S. Caudill, M. Cavaglià, F. Cavalier, R. Cavalieri, G. Cella, C.B. Cepeda,

P. Cerdá-Durán, G. Cerretani, E. Cesarini, S.J. Chamberlin, M. Chan, S. Chao, P. Charlton, E. Chase, E. Chassande-Mottin, D. Chatterjee, K. Chatziioannou, B.D. Cheeseboro, H.Y. Chen, X. Chen, Y. Chen, H.P. Cheng, H. Chia, A. Chincarini, A. Chiummo, T. Chmiel, H.S. Cho, M. Cho, J.H. Chow, N. Christensen, Q. Chu, A.J.K. Chua, S. Chua, A.K.W. Chung, S. Chung, G. Ciani, R. Ciolfi, C.E. Cirelli, A. Cirone, F. Clara, J.A. Clark, P. Clearwater, F. Cleva, C. Cocchieri, E. Coccia, P.F. Cohadon, D. Cohen, A. Colla, C.G. Collette, L.R. Cominsky, M. Constancio, L. Conti, S.J. Cooper, P. Corban, T.R. Corbitt, I. Cordero-Carrión, K.R. Corley, N. Cornish, A. Corsi, S. Cortese, C.A. Costa, M.W. Coughlin, S.B. Coughlin, J.P. Coulon, S.T. Countryman, P. Couvares, P.B. Covas, E.E. Cowan, D.M. Coward, M.J. Cowart, D.C. Coyne, R. Coyne, J.D.E. Creighton, T.D. Creighton, J. Cripe, S.G. Crowder, T.J. Cullen, A. Cumming, L. Cunningham, E. Cuoco, T. Dal Canton, G. Dálya, S.L. Danilishin, S. D'Antonio, K. Danzmann, A. Dasgupta, C.F. Da Silva Costa, V. Dattilo, I. Dave, M. Davier, D. Davis, E.J. Daw, B. Day, S. De, D. DeBra, J. Degallaix, M. De Laurentis, S. Deléglise, W. Del Pozzo, N. Demos, T. Denker, T. Dent, R. De Pietri, V. Dergachev, R. De Rosa, R.T. DeRosa, C. De Rossi, R. DeSalvo, O. de Varona, J. Devenson, S. Dhurandhar, M.C. Díaz, T. Dietrich, L. Di Fiore, M. Di Giovanni, T. Di Girolamo, A. Di Lieto, S. Di Pace, I. Di Palma, F. Di Renzo, Z. Doctor, V. Dolique, F. Donovan, K.L. Dooley, S. Doravari, I. Dorrington, R. Douglas, M. Dovale Álvarez, T.P. Downes, M. Drago, C. Dreissigacker, J.C. Driggers, Z. Du, M. Ducrot, R. Dudi, P. Dupej, S.E. Dwyer, T.B. Edo, M.C. Edwards, A. Effler, H.B. Eggenstein, P. Ehrens, J. Eichholz, S.S. Eikenberry, R.A. Eisenstein, R.C. Essick, D. Estevez, Z.B. Etienne, T. Etzel, M. Evans, T.M. Evans, M. Factourovich, V. Fafone, H. Fair, S. Fairhurst, X. Fan, S. Farinon, B. Farr, W.M. Farr, E.J. Fauchon-Jones, M. Favata, M. Fays, C. Fee, H. Fehrmann, J. Feicht, M.M. Fejer, A. Fernandez-Galiana, I. Ferrante, E.C. Ferreira, F. Ferrini, F. Fidecaro, D. Finstad, I. Fiori, D. Fiorucci, M. Fishbach, R.P. Fisher, M. Fitz-Axen, R. Flaminio, M. Fletcher, H. Fong, J.A. Font, P.W.F. Forsyth, S.S. Forsyth, J.D. Fournier, S. Frasca, F. Frasconi, Z. Frei, A. Freise, R. Frey, V. Frey, E.M. Fries, P. Fritschel, V.V. Frolov, P. Fulda, M. Fyffe, H. Gabbard, B.U. Gadre, S.M. Gaebel, J.R. Gair, L. Gammaitoni, M.R. Ganija, S.G. Gaonkar, C. Garcia-Quiros, F. Garufi, B. Gateley, S. Gaudio, G. Gaur, V. Gayathri, N. Gehrels, G. Gemme, E. Genin, A. Gennai, D. George, J. George, L. Gergely, V. Germain, S. Ghonge, A. Ghosh, A. Ghosh, S. Ghosh, J.A. Giaime, K.D. Giardina, A. Giazotto, K. Gill, L. Glover, E. Goetz, R. Goetz, S. Gomes, B. Goncharov, G. González, J.M. Gonzalez Castro, A. Gopakumar, M.L. Gorodetsky, S.E. Gossan, M. Gosselin, R. Gouaty, A. Grado, C. Graef, M. Granata, A. Grant, S. Gras, C. Gray, G. Greco, A.C. Green, E.M. Gretarsson, P. Groot, H. Grote, S. Grunewald, P. Gruning, G.M. Guidi, X. Guo, A. Gupta, M.K. Gupta, K.E. Gushwa, E.K. Gustafson, R. Gustafson, O. Halim, B.R. Hall, E.D. Hall, E.Z. Hamilton, G. Hammond, M. Haney, M. M. Hanke, J. Hanks, C. Hanna, M.D. Hannam, O.A. Hannuksela, J. Hanson, T. Hardwick, J. Harms, G.M. Harry, I.W. Harry, M.J. Hart, C.J. Haster, K. Haughian, J. Healy, A. Heidmann, M.C. Heintze, H. Heitmann, P. Hello, G. Hemming, M. Hendry, I.S. Heng, J. Hennig, A.W. Heptonstall, M. Heurs, S. Hild, T. Hinderer, W.C.G. Ho, D. Hoak, D. Hofman, K. Holt, D.E. Holz, P. Hopkins, C. Horst, J. Hough, E.A. Houston, E.J. Howell, A. Hreibi, Y.M. Hu, E.A. Huerta, D. Huet, B. Hughey, S. Husa, S.H. Huttner, T. Huynh-Dinh, N. Indik, R. Inta, G. Intini, H.N. Isa, J.M. Isac, M. Isi, B.R. Iyer, K. Izumi, T. Jacqmin, K. Jani, P. Jaranowski, S. Jawahar, F. Jiménez-Forteza, W.W. Johnson, N.K. Johnson-McDaniel, D.I. Jones, R. Jones, R.J.G. Jonker, L. Ju, J. Junker, C.V. Kalaghatgi, V. Kalogera, B. Kamai, S. Kand hasamy, G. Kang, J. B. Kanner, S.J. Kapadia, S. Karki, K.S. Karvinen, M. Kasprzack, W. Kastaun, M. Katolik, E. Katsavounidis, W. Katzman, S. Kaufer, K. Kawabe, F. Kéfélian, D. Keitel, A.J. Kemball, R. Kennedy, C. Kent, J.S. Key, F.Y. Khalili, I. Khan, S. Khan, Z. Khan, E.A. Khazanov, N. Kijbunchoo, C. Kim, J.C. Kim, K. Kim, W. Kim, W.S. Kim, Y.M. Kim, S.J. Kimbrell, E.J. King, P.J. King, M. Kinley-Hanlon, R. Kirchhoff, J.S. Kissel, L. Kleybolte, S. Klimenko, T.D. Knowles, P. Koch, S.M. Koehlenbeck, S. Koley, V. Kondrashov, A. Kontos, M. Korobko, W.Z. Korth, I. Kowalska, D.B. Kozak, C. Krämer, V. Kringel, B. Krishnan, A. Królak, G. Kuehn, P. Kumar, R. Kumar, S. Kumar, L. Kuo, A. Kutynia, S. Kwang, B.D. Lackey, K. H. Lai, M. Landry, R.N. Lang,

J. Lange, B. Lantz, R.K. Lanza, S.L. Larson, A. Lartaux-Vollard, P.D. Lasky, M. Laxen, A. Lazzarini, C. Lazzaro, P. Leaci, S. Leavey, C.H. Lee, H.K. Lee, H.M. Lee, H.W. Lee, K. Lee, J. Lehmann, A. Lenon, E. Leon, M. Leonardi, N. Leroy, N. Letendre, Y. Levin, T.G.F. Li, S.D. Linker, T.B. Littenberg, J. Liu, X. Liu, R.K.L. Lo, N.A. Lockerbie, L.T. London, J.E. Lord, M. Lorenzini, V. Loriette, M. Lormand, G. Losurdo, J.D. Lough, C.O. Lousto, G. Lovelace, H. Lück, D. Lumaca, A.P. Lundgren, R. Lynch, Y. Ma, R. Macas, S. Macfoy, B. Machenschalk, M. MacInnis, D.M. Macleod, I. Magaña Hernandez, F. Magaña-Sandoval, L. Magaña Zertuche, R.M. Magee, E. Majorana, I. Maksimovic, N. Man, V. Mandic, V. Mangano, G.L. Mansell, M. Manske, M. Mantovani, F. Marchesoni, F. Marion, S. Márka, Z. Márka, C. Markakis, A.S. Markosyan, A. Markowitz, E. Maros, A. Marquina, P. Marsh, F. Martelli, L. Martellini, I.W. Martin, R.M. Martin, D.V. Martynov, J.N. Marx, K. Mason, E. Massera, A. Masserot, T.J. Massinger, M. Masso-Reid, S. Mastrogiovanni, A. Matas, F. Matichard, L. Matone, N. Mavalvala, N. Mazumder, R. McCarthy, D.E. McClelland, S. McCormick, L. McCuller, S.C. McGuire, G. McIntyre, J. McIver, D.J. McManus, L. McNeill, T. McRae, S.T. McWilliams, D. Meacher, G.D. Meadors, M. Mehmet, J. Meidam, E. Mejuto-Villa, A. Melatos, G. Mendell, R.A. Mercer, E.L. Merilh, M. Merzougui, S. Meshkov, C. Messenger, C. Messick, R. Metzdorff, P.M. Meyers, H. Miao, C. Michel, H. Middleton, E.E. Mikhailov, L. Milano, A.L. Miller, B.B. Miller, J. Miller, M. Millhouse, M.C. Milovich-Goff, O. Minazzoli, Y. Minenkov, J. Ming, C. Mishra, S. Mitra, V.P. Mitrofanov, G. Mitselmakher, R. Mittleman, D. Moffa, A. Moggi, K. Mogushi, M. Mohan, S.R.P. Mohapatra, I. Molina, M. Montani, C.J. Moore, D. Moraru, G. Moreno, S. Morisaki, S.R. Morriss, B. Mours, C.M. Mow-Lowry, G. Mueller, A.W. Muir, Arunava Mukherjee, D. Mukherjee, S. Mukherjee, N. Mukund, A. Mullavey, J. Munch, E.A. Muñiz, M. Muratore, P.G. Murray, A. Nagar, K. Napier, I. Nardecchia, L. Naticchioni, R.K. Nayak, J. Neilson, G. Nelemans, T.J.N. Nelson, M. Nery, A. Neunzert, L. Nevin, J.M. Newport, G. Newton, K.K.Y. Ng, P. Nguyen, T.T. Nguyen, D. Nichols, A.B. Nielsen, S. Nissanke, A. Nitz, A. Noack, F. Nocera, D. Nolting, C. North, L.K. Nuttall, J. Oberling, G.D. O'Dea, G.H. Ogin, J.J. Oh, S.H. Oh, F. Ohme, M.A. Okada, M. Oliver, P. Oppermann, R.J. Oram, B. O'Reilly, R. Ormiston, L.F. Ortega, R. O'Shaughnessy, S. Ossokine, D.J. Ottaway, H. Overmier, B.J. Owen, A.E. Pace, J. Page, M.A. Page, A. Pai, S.A. Pai, J.R. Palamos, O. Palashov, C. Palomba, A. Pal-Singh, H. Pan, H.-W. Pan, B. Pang, P.T.H. Pang, C. Pankow, F. Pannarale, B.C. Pant, F. Paoletti, A. Paoli, M.A. Papa, A. Parida, W. Parker, D. Pascucci, A. Pasqualetti, R. Passaquieti, D. Passuello, M. Patil, B. Patricelli, B.L. Pearlstone, M. Pedraza, R. Pedurand, L. Pekowsky, A. Pele, S. Penn, C.J. Perez, A. Perreca, L.M. Perri, H.P. Pfeiffer, M. Phelps, O.J. Piccinni, M. Pichot, F. Piergiovanni, V. Pierro, G. Pillant, L. Pinard, I.M. Pinto, M. Pirello, M. Pitkin, M. Poe, R. Poggiani, P. Popolizio, E.K. Porter, A. Post, J. Powell, J. Prasad, J.W.W. Pratt, G. Pratten, V. Predoi, T. Prestegard, M. Prijatelj, M. Principe, S. Privitera, R. Prix, G.A. Prodi, L.G. Prokhorov, O. Puncken, M. Punturo, P. Puppo, M. Pürrer, H. Qi, V. Quetschke, E.A. Quintero, R. Quitzow-James, F.J. Raab, D.S. Rabeling, H. Radkins, P. Raffai, S. Raja, C. Rajan, B. Rajbhandari, M. Rakhmanov, K.E. Ramirez, A. Ramos-Buades, P. Rapagnani, V. Raymond, M. Razzano, J. Read, T. Regimbau, L. Rei, S. Reid, D.H. Reitze, W. Ren, S.D. Reyes, F. Ricci, P.M. Ricker, S. Rieger, K. Riles, M. Rizzo, N.A. Robertson, R. Robie, F. Robinet, A. Rocchi, L. Rolland, J.G. Rollins, V.J. Roma, J.D. Romano, R. Romano, C.L. Romel, J.H. Romie, D. Rosińska, M.P. Ross, S. Rowan, A. Rüdiger, P. Ruggi, G. Rutins, K. Ryan, S. Sachdev, T. Sadecki, L. Sadeghian, M. Sakellariadou, L. Salconi, M. Saleem, F. Salemi, A. Samajdar, L. Sammut, L.M. Sampson, E.J. Sanchez, L.E. Sanchez, N. Sanchis-Gual, V. Sand berg, J.R. Sanders, B. Sassolas, B.S. Sathyaprakash, P.R. Saulson, O. Sauter, R.L. Savage, A. Sawadsky, P. Schale, M. Scheel, J. Scheuer, J. Schmidt, P. Schmidt, R. Schnabel, R.M.S. Schofield, A. Schönbeck, E. Schreiber, D. Schuette, B.W. Schulte, B.F. Schutz, S.G. Schwalbe, J. Scott, S. M. Scott, E. Seidel, D. Sellers, A.S. Sengupta, D. Sentenac, V. Sequino, A. Sergeev, D.A. Shaddock, T.J. Shaffer, A.A. Shah, M.S. Shahriar, M.B. Shaner, L. Shao, B. Shapiro, P. Shawhan, A. Sheperd, D.H. Shoemaker, D.M. Shoemaker, K. Siellez, X. Siemens, M. Sieniawska, D. Sigg, A.D. Silva, L.P. Singer, A. Singh, A. Singhal, A.M. Sintes, B.J.J. Slagmolen, B. Smith, J.R. Smith,

R.J.E. Smith, S. Somala, E.J. Son, J.A. Sonnenberg, B. Sorazu, F. Sorrentino, T. Souradeep, A.P. Spencer, A.K. Srivastava, K. Staats, A. Staley, M. Steinke, J. Steinlechner, S. Steinlechner, D. Steinmeyer, S.P. Stevenson, R. Stone, D.J. Stops, K.A. Strain, G. Stratta, S.E. Strigin, A. Strunk, R. Sturani, A.L. Stuver, T.Z. Summerscales, L. Sun, S. Sunil, J. Suresh, P.J. Sutton, B.L. Swinkels, M.J. Szczepańczyk, M. Tacca, S.C. Tait, C. Talbot, D. Talukder, D.B. Tanner, M. Tápai, A. Taracchini, J.D. Tasson, J.A. Taylor, R. Taylor, S.V. Tewari, T. Theeg, F. Thies, E.G. Thomas, M. Thomas, P. Thomas, K.A. Thorne, K.S. Thorne, E. Thrane, S. Tiwari, V. Tiwari, K.V. Tokmakov, K. Toland, M. Tonelli, Z. Tornasi, A. Torres-Forné, C.I. Torrie, D. Töyrä, F. Travasso, G. Traylor, J. Trinastic, M.C. Tringali, L. Trozzo, K.W. Tsang, M. Tse, R. Tso, L. Tsukada, D. Tsuna, D. Tuyenbayev, K. Ueno, D. Ugolini, C.S. Unnikrishnan, A.L. Urban, S.A. Usman, H. Vahlbruch, G. Vajente, G. Valdes, M. Vallisneri, N. van Bakel, M. van Beuzekom, J.F.J. van den Brand, C. Van Den Broeck, D.C. Vand er-Hyde, L. van der Schaaf, J.V. van Heijningen, A.A. van Veggel, M. Vardaro, V. Varma, S. Vass, M. Vasúth, A. Vecchio, G. Vedovato, J. Veitch, P.J. Veitch, K. Venkateswara, G. Venugopalan, D. Verkindt, F. Vetrano, A. Viceré, A.D. Viets, S. Vinciguerra, D.J. Vine, J.Y. Vinet, S. Vitale, T. Vo, H. Vocca, C. Vorvick, S.P. Vyatchanin, A.R. Wade, L.E. Wade, M. Wade, R. Walet, M. Walker, L. Wallace, S. Walsh, G. Wang, H. Wang, J.Z. Wang, W.H. Wang, Y.F. Wang, R.L. Ward, J. Warner, M. Was, J. Watchi, B. Weaver, L. W. Wei, M. Weinert, A.J. Weinstein, R. Weiss, L. Wen, E.K. Wessel, P. Weßels, J. Westerweck, T. Westphal, K. Wette, J.T. Whelan, S.E. Whitcomb, B.F. Whiting, C. Whittle, D. Wilken, D. Williams, R.D. Williams, A.R. Williamson, J.L. Willis, B. Willke, M.H. Wimmer, W. Winkler, C.C. Wipf, H. Wittel, G. Woan, J. Woehler, J. Wofford, K.W.K. Wong, J. Worden, J.L. Wright, D.S. Wu, D.M. Wysocki, S. Xiao, H. Yamamoto, C.C. Yancey, L. Yang, M.J. Yap, M. Yazback, H. Yu, H. Yu, M. Yvert, A. ZadroŻny, M. Zanolin, T. Zelenova, J.P. Zendri, M. Zevin, L. Zhang, M. Zhang, T. Zhang, Y.H. Zhang, C. Zhao, M. Zhou, Z. Zhou, S.J. Zhu, X.J. Zhu, A.B. Zimmerman, M.E. Zucker, J. Zweizig, LIGO Scientific Collaboration, and Virgo Collaboration. GW170817: observation of gravitational waves from a binary neutron star inspiral. Phys. Rev. Lett. **119**(16), 161101 (2017)

8. M. Alcubierre, S. Brandt, B. Brügmann, D. Holz, E. Seidel, R. Takahashi, J. Thornburg, Symmetry without symmetry: numerical simulation of axisymmetric systems using cartesian grids. Int. J. Mod. Phys. D **10**, 273–290 (2001)
9. T. Andrade, L.A. Salo, J.C. Aurrekoetxea, J. Bamber, K. Clough, R. Croft, E. de Jong, A. Drew, A. Duran, P.G. Ferreira, P. Figueras, H. Finkel, T. França, B.-X. Ge, C. Gu, T. Helfer, J. Jäykkä, C. Joana, M. Kunesch, K. Kornet, E.A. Lim, F. Muia, Z. Nazari, M. Radia, J. Ripley, P. Shellard, U. Sperhake, D. Traykova, S. Tunyasuvunakool, Z. Wang, J.Y. Widdicombe, K. Wong, GRChombo: an adaptable numerical relativity code for fundamental physics. J. Open Source Softw. **6**(68), 3703 (2021)
10. M. Ansorg, B. Brügmann, W. Tichy, A single-domain spectral method for black hole puncture data. Phys. Rev. D **70**, 064011 (2004)
11. L. Antón, O. Zanotti, J.A. Miralles, J.M. Martí, J.M. Ibáñez, J.A. Font, J.A. Pons, Numerical 3+1 general relativistic magnetohydrodynamics: a Local characteristic approach. Astrophys. J. **637**, 296–312 (2006)
12. R.L. Arnowitt, S. Deser, C.W. Misner, The dynamics of general relativity. Gen. Relativ. Gravit. **40**, 1997–2027 (2008)
13. T. Assumpcao, L.R. Werneck, T.P. Jacques, Z.B. Etienne, Fast hyperbolic relaxation elliptic solver for numerical relativity: conformally flat, binary puncture initial data. Phys. Rev. D **105**(10), 104037 (2022)
14. L. Baiotti, I. Hawke, P.J. Montero, F. Löffler, L. Rezzolla, N. Stergioulas, J.A. Font, E. Seidel, Three-dimensional relativistic simulations of rotating neutron star collapse to a Kerr black hole. Phys. Rev. D **71**, 024035 (2005)
15. J.G. Baker, M. Campanelli, C.O. Lousto, The Lazarus project: A Pragmatic approach to binary black hole evolutions. Phys. Rev. D **65**, 044001 (2002)
16. F. Banyuls, J.A. Font, J.M.A. Ibanez, J.M.A. Marti, J.A. Miralles, Numerical 3+1 general relativistic hydrodynamics: a local characteristic approach. Astrophys. J. **476**, 221 (1997)

17. E. Bentivegna, M. Korzynski, Evolution of a periodic eight-black-hole lattice in numerical relativity. Class. Quant. Grav. **29**, 165007 (2012)
18. S. Bonazzola, E. Gourgoulhon, J.-A. Marck, Numerical approach for high precision 3d relativistic star models. Phys. Rev. D **58**(10), 104020 (1998)
19. R. Borges, M. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J. Comp. Phys. **227**, 3191–3211 (2008)
20. J.M. Bowen, J.W. York Jr., Time asymmetric initial data for black holes and black hole collisions. Phys. Rev. D **21**, 2047–2056 (1980)
21. S. Brandt, B. Brügmann, Black hole punctures as initial data for general relativity. Phys. Rev. Lett. **78**, 3606–3609 (1997)
22. S.R. Brandt, Automatic generation of Carpet/CarpetX thorns with NRPy+, 2022. https://youtu.be/d6VZYiAKvXc
23. S.R. Brandt, H. Roland, The einstein toolkit tutorial server, in *Science Gateways 2023 Annual Conference* (2023)
24. J. David Brown, P. Diener, O. Sarbach, E. Schnetter, M. Tiglio, Turduckening black holes: an analytical and computational study. Phys. Rev. D **79**, 044023 (2009)
25. Cactus Computational Toolkit
26. P. Cerda-Duran, J.A. Font, L. Anton, E. Muller, A new general relativistic magnetohydrodynamics code for dynamical spacetimes. Astron. Astrophys. **492**, 937 (2008)
27. R. Ciolfi, W. Kastaun, B. Giacomazzo, A. Endrizzi, D.M. Siegel, R. Perna, General relativistic magnetohydrodynamic simulations of binary neutron star mergers forming a long-lived neutron star. Phys. Rev. D **95**(6), 063016 (2017)
28. F. Cipolletta, J.V. Kalinani, B. Giacomazzo, R. Ciolfi, Spritz: a new fully general-relativistic magnetohydrodynamic code. Class Quantum Gravity **37**(13), 135010 (2020)
29. P. Colella, P.R. Woodward, The Piecewise Parabolic Method (PPM) for gas dynamical simulations. J. Comp. Phys. **54**, 174–201 (1984)
30. S. Cupp, S.R. Brandt, G. Bozzola, P. Diener, Z. Etienne, D. Ferguson, R. Haas, L. Ji, H. Park, F.G. Lopez Armengol, M. Chabanov, C.-H. Cheng, A. Dima, J. Doherty, L. Ennoggi, T.P. Jacques, J. Kalinani, P. Moesta, M. Pirog, L.T. Sanches, E. Schnetter, S. Shankar, W. Tichy, L. Werneck, M. Alcubierre, D. Alic, G. Allen, M. Ansorg, M. Babiuc-Hamilton, L. Baiotti, W. Benger, E. Bentivegna, S. Bernuzzi, T. Bode, B. Brendal, B. Bruegmann, M. Campanelli, F. Cipolletta, G. Corvino, R. De Pietri, H. Dimmelmeier, R. Dooley, N. Dorband, M. Elley, Y. El Khamra, J. Faber, G. Ficarra, T. Font, J. Frieben, B. Giacomazzo, T. Goodale, C. Gundlach, I. Hawke, S. Hawley, I. Hinder, E.A. Huerta, S. Husa, T. Ikeda, S. Iyer, D. Johnson, A.V. Joshi, A. Kankani, W. Kastaun, T. Kellermann, A. Knapp, M. Koppitz, P. Laguna, G. Lanferman, P. Lasky, F. Löffler, H. Macpherson, J. Masso, L. Menger, A. Merzky, J.M. Miller, M. Miller, P. Montero, B. Mundim, P. Nelson, A. Nerozzi, S.C. Noble, C. Ott, L.J. Papenfort, R. Paruchuri, D. Pollney, D. Price, D. Radice, T. Radke, C. Reisswig, L. Rezzolla, C.B. Richards, D. Rideout, M. Ripeanu, L. Sala, J.A. Schewtschenko, B. Schutz, E. Seidel, E. Seidel, J. Shalf, K. Sible, U. Sperhake, N. Stergioulas, W.-M. Suen, B. Szilagyi, R. Takahashi, M. Thomas, J. Thornburg, C. Tian, M. Tobias, A. Tonita, S. Tootle, P. Walker, M.-B. Wan, B. Wardell, H. Witek, M. Zilhão, B. Zink, Y. Zlochower, The Einstein toolkit (2023)
31. R. Donat, A. Marquina, Capturing shock reflections: an improved flux formula. J. Comp. Phys. **125**, 42–58 (1996)
32. O. Dreyer, B. Krishnan, D. Shoemaker, E. Schnetter, Introduction to isolated horizons in numerical relativity. Phys. Rev. D **67**, 024018 (2003)
33. M.D. Duez, F. Foucart, L.E. Kidder, H.P. Pfeiffer, M.A. Scheel, S.A. Teukolsky, Evolving black hole-neutron star binaries in general relativity using pseudospectral and finite difference methods. Phys. Rev. D **78**, 104015 (2008)
34. M.D. Duez, Y.T. Liu, S.L. Shapiro, M. Shibata, B.C. Stephens, Collapse of magnetized hypermassive neutron stars in general relativity. Phys. Rev. Lett. **96**, 031101 (2006)
35. M.D. Duez, Y.T. Liu, S.L. Shapiro, B.C. Stephens, Relativistic magnetohydrodynamics in dynamical spacetimes: numerical methods and tests. Phys. Rev. D **72**, 024028 (2005)

36. B. Einfeldt, On Godunov-type methods for gas dynamics. SIAM J. Numer. Anal. **25**(2), 294–318 (1988)
37. Einstein Toolkit: Open software for relativistic astrophysics
38. P.L. Espino, G. Bozzola, V. Paschalidis, Quantifying uncertainties in general relativistic magnetohydrodynamic codes. Phys. Rev. D **107**(10), 104059 (2023)
39. Z.B. Etienne, V. Paschalidis, R. Haas, P. Mösta, S.L. Shapiro, IllinoisGRMHD: an open-source, user-friendly GRMHD code for dynamical spacetimes. Class. Quant. Grav. **32**(17), 175009 (2015)
40. Z.B. Etienne, V. Paschalidis, Y.T. Liu, S.L. Shapiro, Relativistic MHD in dynamical spacetimes: improved EM gauge condition for AMR grids. Phys. Rev. D **85**, 024013 (2012)
41. C.R. Evans, J.F. Hawley, Simulation of magnetohydrodynamic flows: a Constrained transport method. Astrophys. J. **332**, 659–677 (1988)
42. B.D. Farris, R. Gold, V. Paschalidis, Z.B. Etienne, S.L. Shapiro, Binary black-hole mergers in magnetized disks: simulations in full general relativity. Phys. Rev. Lett. **109**, 221102 (2012)
43. B.D. Farris, Y.T. Liu, S.L. Shapiro, Binary black hole mergers in gaseous disks: Simulations in general relativity. Phys. Rev. D **84**, 024024 (2011)
44. D. Ferguson et al., Second MAYA catalog of binary black hole numerical relativity waveforms 9 (2023)
45. L.G. Fishbone, V. Moncrief, Relativistic fluid disks in orbit around kerr black holes. Astrophys. J. **207**, 962–976 (1976)
46. B. Giacomazzo, Numerical simulations of coalescing neutron star binaries part i-numerical relativity (2023)
47. B. Giacomazzo, L. Rezzolla, WhiskyMHD: a new numerical code for general relativistic magnetohydrodynamics. Class. Quantum Grav. **24**, S235–S258 (2007)
48. R. Gold, V. Paschalidis, Z.B. Etienne, S.L. Shapiro, H.P. Pfeiffer, Accretion disks around binary black holes of unequal mass: General relativistic magnetohydrodynamic simulations near decoupling. Phys. Rev. D **89**(6), 064060 (2014)
49. T. Goodale, G. Allen, G. Lanfermann, J. Massó, T. Radke, E. Seidel, J. Shalf, *The Cactus framework and toolkit: design and applications, in Vector and Parallel Processing - VECPAR'2002, 5th International Conference* Lecture Notes in Computer Science. (Springer, Berlin, 2003)
50. P. Grandclement, Kadath: a spectral solver for theoretical physics. J. Comput. Phys. **229**, 3334–3357 (2010)
51. S. Habib, A. Ramos-Buades, E.A. Huerta, S. Husa, R. Haas, Z. Etienne, Initial data and eccentricity reduction toolkit for binary black hole numerical relativity waveforms 11 (2020)
52. A. Harten, High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**, 357–393 (1983)
53. J. Healy, C.O. Lousto, Fourth RIT binary black hole simulations catalog: extension to eccentric orbits. Phys. Rev. D **105**(12), 124010 (2022)
54. E.A. Huerta, R. Haas, S. Habib, A. Gupta, A. Rebei, V. Chavva, D. Johnson, S. Rosofsky, E. Wessel, B. Agarwal, D. Luo, W. Ren, Physics of eccentric binary black hole mergers: A numerical relativity perspective. Phys. Rev. D **100**(6), 064003 (2019)
55. S. Husa, I. Hinder, C. Lechner, Kranc: a Mathematica application to generate numerical codes for tensorial evolution equations. Comput. Phys. Commun. **174**, 983–1004 (2006)
56. M.R. Izquierdo, J. Fernando Abalos, C. Palenzuela, Guided moments formalism: a new efficient full-neutrino treatment for astrophysical simulations. Phys. Rev. D **109**(4), 043044 (2024)
57. J.V. Kalinani, R. Ciolfi, W. Kastaun, B. Giacomazzo, F. Cipolletta, L. Ennoggi, Implementing a new recovery scheme for primitive variables in the general relativistic magnetohydrodynamic code Spritz. Phys. Rev. D **105**(10), 103031 (2022)
58. J.V. Kalinani et al., AsterX: a new open-source GPU-accelerated GRMHD code for dynamical spacetimes 6 (2024)
59. W. Kastaun, J.V. Kalinani, R. Ciolfi, Robust recovery of primitive variables in relativistic ideal magnetohydrodynamics. Phys. Rev. D **103**(2), 023018 (2021)

60. H. Komatsu, Y. Eriguchi, I. Hachisu, Rapidly rotating general relativistic stars. I - Numerical method and its application to uniformly rotating polytropes. Mon. Not. Roy. Astron. Soc. **237**, 355–379 (1989)
61. F. Löffler, J. Faber, E. Bentivegna, T. Bode, P. Diener, R. Haas, I. Hinder, B.C. Mundim, C.D. Ott, E. Schnetter, G. Allen, M. Campanelli, P. Laguna, The Einstein toolkit: a community computational infrastructure for relativistic astrophysics. Class. Quantum Grav. **29**(11), 115001 (2012)
62. F.G. Lopez Armengol, Z.B. Etienne, S.C. Noble, B.J. Kelly, L.R. Werneck, B. Drachler, M. Campanelli, F. Cipolletta, Y. Zlochower, A. Murguia-Berthier, L. Ennoggi, M. Avara, R. Ciolfi, J. Faber, G. Fiacco, B. Giacomazzo, T. Gupte, T. Ha, J.H. Krolik, V. Mewes, R. O'Shaughnessy, J.M. Rueda-Becerril, J. Schnittman, Handing off the outcome of binary neutron star mergers for accurate and long-term postmerger simulations. Phys. Rev. D **106**, 083015 (2022)
63. G. Lovelace, Y. Chen, M. Cohen, J.D. Kaplan, D. Keppel, K.D. Matthews, D.A. Nichols, M.A. Scheel, U. Sperhake, Momentum flow in black-hole binaries. II. Numerical simulations of equal-mass, head-on mergers with antiparallel spins. Phys. Rev. D **82**, 064031 (2010)
64. H.J. Macpherson, P.D. Lasky, D.J. Price, Inhomogeneous cosmology with numerical relativity. Phys. Rev. D **95**(6), 064028 (2017)
65. H.J. Macpherson, P.D. Lasky, D.J. Price, The trouble with Hubble: local versus global expansion rates in inhomogeneous cosmological simulations with numerical relativity. Astrophys. J. Lett. **865**(1), L4 (2018)
66. F. Maione, R. De Pietri, A. Feo, F. Löffler, Binary neutron star merger simulations with different initial orbital frequency and equation of state. Class. Quant. Grav. **33**(17), 175009 (2016)
67. J.M. Martí, J.M. Ibáñez, J.M. Miralles, Numerical relativistic hydrodynamics: local characteristic approach. Phys. Rev. D **43**, 3794–3801 (1991)
68. P. McCorquodale, P. Colella, A high-order finite-volume method for conservation laws on locally refined grids. Comm. Appl. Math. Comput. Sci. **6**, 1 (2011)
69. V. Mewes, Y. Zlochower, M. Campanelli, I. Ruchlin, Z.B. Etienne, T.W. Baumgarte, Numerical relativity in spherical coordinates with the Einstein Toolkit. Phys. Rev. D **97**(8), 084059 (2018)
70. E.R. Most, L. Jens Papenfort, L. Rezzolla, Beyond second-order convergence in simulations of magnetized binary neutron stars with realistic microphysics. Mon. Not. Roy. Astron. Soc. **490**(3), 3588–3600 (2019)
71. P. Mösta, B.C. Mundim, J.A. Faber, R. Haas, S.C. Noble, T. Bode, F. Löffler, C.D. Ott, C. Reisswig, E. Schnetter, GRHydro: a new open source general-relativistic magnetohydrodynamics code for the Einstein Toolkit. Class. Quantum Gravity **31**(1), 015005 (2014)
72. P. Mösta, C.D. Ott, D. Radice, L.F. Roberts, E. Schnetter, R. Haas, A large scale dynamo and magnetoturbulence in rapidly rotating core-collapse supernovae. Nature **528**, 376 (2015)
73. P. Mösta, S. Richers, C.D. Ott, R. Haas, A.L. Piro, K. Boydstun, E. Abdikamalov, C. Reisswig, E. Schnetter, Magnetorotational core-collapse supernovae in three dimensions. Astrophys. J. Lett. **785**, L29 (2014)
74. P. Mösta, L.F. Roberts, G. Halevi, C.D. Ott, J. Lippuner, R. Haas, E. Schnetter, R-process nucleosynthesis from three-dimensional magnetorotational core-collapse supernovae. Astrophys. J. **864**(2), 171 (2018)
75. S.C. Noble, C.F. Gammie, J.C. McKinney, L. Del Zanna, Primitive variable solvers for conservative general relativistic magnetohydrodynamics. Astrophys. J. **641**, 626–637 (2006)
76. N.O. Murchadha, J.W. York, Gravitational energy. Phys. Rev. D **10**, 2345–2357 (1974)
77. J.R. Oppenheimer, G.M. Volkoff, On Massive Neutron Cores. Phys. Rev. **55**, 374–381 (1939)
78. C.D. Ott, E. Abdikamalov, P. Mösta, R. Haas, S. Drasco, E.P. O'Connor, C. Reisswig, C.A. Meakin, E. Schnetter, General-relativistic simulations of three-dimensional core-collapse supernovae. Astrophys. J. **768**, 115 (2013)
79. L. Jens Papenfort, S.D. Tootle, P. Grandclément, E.R. Most, L. Rezzolla, New public code for initial data of unequal-mass, spinning compact-object binaries. Phys. Rev. D **104**(2), 024057 (2021)

80. A.J. Penner, General relativistic magnetohydrodynamic Bondi-Hoyle accretion. Mon. Not. Roy. Astron. Soc. **414**, 1467–1482 (2011)
81. D. Pollney, C. Reisswig, E. Schnetter, N. Dorband, P. Diener, High accuracy binary black hole simulations with an extended wave zone. Phys. Rev. D **83**, 044045 (2011)
82. D. Radice, L. Rezzolla, F. Galeazzi, High-order fully general-relativistic hydrodynamics: new approaches and tests. Class. Quant. Grav. **31**, 075012 (2014)
83. D. Radice, L. Rezzolla, F. Galeazzi, High-order numerical-relativity simulations of binary neutron stars. ASP Conf. Ser. **498**, 121–126 (2015)
84. C. Reisswig, R. Haas, C.D. Ott, E. Abdikamalov, P. Mösta, D. Pollney, E. Schnetter, Three-dimensional general-relativistic hydrodynamic simulations of binary neutron star coalescence and stellar collapse with multipatch grids. Phys. Rev. D **87**, 064023 (2013)
85. P.L. Roe, Characteristic-based schemes for the Euler equations. Ann. Rev. Fluid Mech. **18**, 337–365 (1986)
86. I. Ruchlin, Z.B. Etienne, T.W. Baumgarte, SENR /NRPy +ÃƒÂ¢Ã¢âŁšÂ¬Ã‚Â¯: Numerical relativity in singular curvilinear coordinate systems. Phys. Rev. D **97**(6), 064036 (2018)
87. I. Ruchlin, Z.B. Etienne, T.W. Baumgarte, SENR/NRPy+: numerical relativity in singular curvilinear coordinate systems. Phys. Rev. D **97**(6), 064036 (2018)
88. M. Ruiz, A. Tsokaros, S.L. Shapiro, Jet launching from merging magnetized binary neutron stars with realistic equations of state. Phys. Rev. D **104**(12), 124049 (2021)
89. E. Schnetter, M. Blazewicz, S.R. Brandt, D.M. Koppelman, F. Löffler, *Chemora: a PDE solving framework for modern HPC architectures* (Comput. Sci, Engin, 2014). ((arXiv:1410.1764))
90. E. Schnetter, S. Brandt, S. Cupp, R. Haas, P. Mösta, S. Shankar, Carpetx, February 2022. Grants: 10.13039/100000001::2004157–10.13039/100000001::2004879
91. E. Schnetter, S.H. Hawley, I. Hawke, Evolutions in 3-D numerical relativity using fixed mesh refinement. Class. Quantum Grav. **21**, 1465–1488 (2004)
92. E. Schnetter, B. Krishnan, F. Beyer, Introduction to dynamical horizons in numerical relativity. Phys. Rev. D **74**, 024028 (2006)
93. S. Shankar, P. Mösta, S.R. Brandt, R. Haas, E. Schnetter, Y. de Graaf, GRaM-X: a new GPU-accelerated dynamical spacetime GRMHD code for exascale computing with the Einstein toolkit. Class Quantum Gravity **40**(20), 205009 (2023)
94. C.W. Shu, High order ENO and WENO schemes for computational fluid dynamics, in *High Order Methods for Computational Physics*. ed. by J. Timothy, H. Barth (Array Deconinck (Springer, New York, 1999), pp.439–582
95. U. Sperhake, Binary black-hole evolutions of excision and puncture data. Phys. Rev. D **76**, 104015 (2007)
96. A. Suresh, H.T. Huynh, Accurate monotonicity-preserving schemes with Runge-Kutta time stepping. J. Comput. Phys. **136**(1), 83–99 (1997)
97. N. Tacik, F. Foucart, H.P. Pfeiffer, R. Haas, S. Ossokine, J. Kaplan, C. Muhlberger, M.D. Duez, L.E. Kidder, M.A. Scheel, B. Szilágyi, Binary neutron stars with arbitrary spins in numerical relativity. Phys. Rev. D **92**(12), 124012 (2015). [Erratum: Phys. Rev. D **94**, 049903 (2016)]
98. A. Tchekhovskoy, J.C. McKinney, R. Narayan, WHAM: a WENO-based general relativistic numerical scheme I: Hydrodynamics. Mon. Not. Roy. Astron. Soc. **379**, 469–497 (2007)
99. J. Thornburg, A 3+ 1 computational scheme for dynamic spherically symmetric black hole spacetimes–ii: time evolution (1999). arXiv preprint gr-qc/9906022
100. J. Thornburg, A fast apparent-horizon finder for 3-dimensional cartesian grids in numerical relativity. Class. Quantum Grav. **21**, 743–766 (2004)
101. W. Tichy, A. Rashti, T. Dietrich, R. Dudi, B. Brügmann, Constructing binary neutron star initial data with high spins, high compactnesses, and high mass ratios. Phys. Rev. D **100**(12), 124046 (2019)
102. K.P. Tod, Looking for marginally trapped surfaces. Class. Quantum Gravity **8**(5), L115 (1991)
103. R.C. Tolman, Static solutions of Einstein's field equations for spheres of fluid. Phys. Rev. **55**, 364–373 (1939)

104. G. Tóth. The $\nabla.B = 0$ constraint in shock-capturing magnetohydrodynamics codes. J. Comput. Phys. **161**, 605–652 (2000)
105. A. Čadež, Ph.d. thesis, University of North Carolina at Chapel Hill, 1971
106. L.R. Werneck et al., Addition of tabulated equation of state and neutrino leakage support to illinoisgrmhd. Phys. Rev. D **107**(4), 044037 (2023)
107. H. Witek, M. Zilhao, G. Bozzola, M. Elley, G. Ficarra, T. Ikeda, N. Sanchis-Gual, H. Silva, Canuda: a public numerical relativity library to probe fundamental physics, October 2021
108. Y. Zenati, J. Krolik, L. Werneck, Z. Etienne, S. Noble, A. Murguia-Berthier, J. Schnittman, The dynamics of debris disk creation in neutron star mergers 4 (2024)
109. Y. Zenati, J.H. Krolik, L.R. Werneck, A. Murguia-Berthier, Z.B. Etienne, S.C. Noble, T. Piran, Bound debris expulsion from neutron star merger remnants. Astrophys. J. **958**(2), 161 (2023)
110. W. Zhang, A. Almgren, V. Beckner, J. Bell, J. Blaschke, C. Chan, M. Day, B. Friesen, K. Gott, D. Graves, M. Katz, A. Myers, T. Nguyen, A. Nonaka, M. Rosso, S. Williams, M. Zingale, AMReX: a framework for block-structured adaptive mesh refinement. J. Open Sour. Softw. **4**(37), 1370 (2019)