Journal of Computational Physics 231 (2012) 718–744

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Parallel, grid-adaptive approaches for relativistic hydro and magnetohydrodynamics

R. Keppens [a,b,c,*], Z. Meliani [a], A.J. van Marle [a], P. Delmont [a], A. Vlasis [a], B. van der Holst [d]

[a] *Centre for Plasma-Astrophysics, K.U.Leuven, Celestijnenlaan 200B, 3001 Heverlee, Belgium*
[b] *FOM-Institute for Plasma Physics Rijnhuizen, P.O. Box 1207, 3430 BE Nieuwegein, The Netherlands*
[c] *Astronomical Institute, Utrecht University, P.O. Box 80000, 3508 TA Utrecht, The Netherlands*
[d] *Department of Atmospheric, Oceanic and Space Sciences, University of Michigan, Ann Arbor, MI 48109, USA*

ARTICLE INFO

Article history:
Available online 15 January 2011

Keywords:
AMR
Relativistic HD
Relativistic MHD
Shock-capturing schemes

ABSTRACT

Relativistic hydro and magnetohydrodynamics provide continuum fluid descriptions for gas and plasma dynamics throughout the visible universe. We present an overview of state-of-the-art modeling in special relativistic regimes, targeting strong shock-dominated flows with speeds approaching the speed of light. Significant progress in its numerical modeling emerged in the last two decades, and we highlight specifically the need for grid-adaptive, shock-capturing treatments found in several contemporary codes in active use and development. Our discussion highlights one such code, MPI-AMRVAC (Message-Passing Interface-Adaptive Mesh Refinement Versatile Advection Code), but includes generic strategies for allowing massively parallel, block-tree adaptive simulations in any dimensionality. We provide implementation details reflecting the underlying data structures as used in MPI-AMRVAC. Parallelization strategies and scaling efficiencies are discussed for representative applications, along with guidelines for data formats suitable for parallel I/O. Refinement strategies available in MPI-AMRVAC are presented, which cover error estimators in use in many modern AMR frameworks. A test suite for relativistic hydro and magnetohydrodynamics is provided, chosen to cover all aspects encountered in high-resolution, shock-governed astrophysical applications. This test suite provides ample examples highlighting the advantages of AMR in relativistic flow problems.

1. Contemporary relativistic simulations

1.1. Overview of current coding efforts

Relativistic flows play a fundamental role in highly energetic astrophysical scenarios, ranging from accretion flows in the vicinity of compact stars and black holes, over pulsar wind environments, to the jetted outflows in X-ray binary systems (micro-quasars), in association with Active Galactic Nuclei, and inferred for catastrophic explosions of long gamma-ray-burst (GRB) progenitor stars. In all these areas, spectacular progress in physical insight emerged from increasingly sophisticated numerical simulations, where the large-scale dynamics is governed by relativistic hydrodynamic and magnetohydrodynamic equations.

* Corresponding author at: Centre for Plasma-Astrophysics, K.U.Leuven, Celestijnenlaan 200B, 3001 Heverlee, Belgium.

E-mail addresses: Rony.Keppens@wis.kuleuven.be (R. Keppens), Zakaria.Meliani@wis.kuleuven.be (Z. Meliani), AllardJan.vanMarle@wis.kuleuven.be (A.J. van Marle), delmont@mathcces.rwth-aachen.de (P. Delmont), Alkis.Vlasis@wis.kuleuven.be (A. Vlasis), bartvand@umich.edu (B. van der Holst).

0021-9991/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2011.01.020

These advances in numerical relativity are best documented in living review articles: Martí and Müller [53] provide a thorough discussion of high-resolution shock capturing schemes in special relativistic hydrodynamics. They emphasize the crucial role played by the exact knowledge of the solution to the Riemann problem, where two uniform states start off in direct contact. The various schemes that do justice to the conservation law character of the relativistic hydro equations make use of this knowledge to varying degree. Their review [53] provides pointers to other approaches, and contains a test bench of stringent 1D shock problems. The rapid development of the field is evident from the observation that all aspects indicated as future developments (improving the equation of state, coupling special relativistic hydro with adaptive mesh refinement – AMR, and going to general relativistic magnetohydrodynamics) have meanwhile been realized. The state-of-the-art in general relativistic hydro and magnetohydrodynamic simulations, where some have progressed to perform fully coupled simulations where the Einstein equations are solved to dynamically update the metric used for the flow problems, is covered in the review by Font [26]. This latter review provides once more a clear history of algorithm evolution in numerical relativity, gives an overview of codes that handle 'test fluid' evolutions where the metric is held fixed, and discusses applications ranging from gravitational collapse (with predictions for gravitational wave signals) of single stars, to accretion disk/jet physics, to neutron star physics, even compact binary coalescence.

In this paper, we focus on special relativistic hydro and MHD applications, motivated mainly from our own continued effort in augmenting the wealth of community codes available for astrophysical research. We start with listing some of the many related coding efforts, meant to highlight the commonalities/differences between these codes first, to then follow in the remainder of this paper with an in-depth discussion of shock-capturing schemes for relativistic (M)HD and AMR strategies, that form the recurring ingredients of modern code developments.

In [1], the Genesis code for 3D special relativistic hydro was introduced, which has set the scene for a whole generation of future coding efforts. It used Riemann solver methodologies [25] from the start, and has been used extensively, e.g. to study relativistic jet dynamics [52].

A code that has made significant progress in recent years is PLUTO [62], which improved shock-capturing methodologies for relativistic flow problems [60,61] and performed basic research on relativistic jet dynamics [64,75]. It shares philosophy with the MPI-AMRVAC code discussed in this paper, in the sense that it provides versatility in physics modules and their solvers, covering Newtonian Euler to MHD, up to special relativistic (M)HD, allowing for different equations of state (EOS) [59]. Its design allows both Cartesian up to curvilinear grids, and efforts are underway to incorporate patch-based AMR by coupling the code to the public Chombo [14] library.

In [34], an Harten–Lax–van Leer–Einfeldt (HLLE) discretization [32] combined with patch-based AMR was used for special relativistic hydro flows, in up to 3D Cartesian boxes using an ideal gas constant gamma EOS. The RAM [95] code also focuses on special relativistic hydro, employing parallel AMR borrowed from the PARAMESH [49] development, also exploited by the (non-relativistic) FLASH [13] code. RAM uses higher order accurate weighted essentially nonoscillatory (ENO) or piecewise parabolic (PPM) reconstructions, combined with 3rd order total variation diminishing (TVD) Runga–Kutta schemes. The AMR is a pure block-tree, and along with the parallelization inherited from PARAMESH [49]. In [95], a constant adiabatic index EOS is employed. In (r)enzo [91], special relativistic hydro schemes from local Lax-Friedrich, HLL to HLLC have been used (versions for gas dynamics of these solvers are discussed in [81]), in combination with AMR, following closely the original (patch-based) Berger and Collela [9] prescriptions.

The RAISHIN [68] code works with a fixed grid, is designed to handle general relativistic MHD in a fixed metric, combined with an ideal gas, constant adiabatic index law. It has been exploited in 3D Cartesian, special relativistic MHD jet simulations, where it uses an HLL scheme combined with a constrained transport magnetic field treatment. The HARM [27] scheme uses an HLL type central flux, with different linear limited reconstructions, using the flux-CT scheme by Tóth [85] for handling numerical monopoles in general relativistic MHD in fixed metrics (Schwarzschild and Kerr). The grid resolution is fixed. Yet another development is known as ECHO [20,19,21], where up to general relativistic MHD and force-free electrodynamics modules are implemented, using higher order reconstructions, with ambitions to couple to an Einstein equation solver. Special relativistic problems with applications to pulsar wind nebulae have been investigated. In contrast to these latter fixed grid codes, Cosmos++ [4] exploits pure cell-based AMR, suitable for fully unstructured grid settings, while the general relativistic MHD equations are solved in a fixed metric. In [3], a patch-based AMR strategy is used for general relativistic MHD computations, using ENO-type reconstructions and hyperbolic divergence cleaning for the magnetic field. Finally, arguably the most advanced code in use today is the WhiskyMHD [30] development. It represents a large collaborative effort employing the Cactus [11] toolkit, along with its carpet AMR driver, to perform general relativistic MHD simulations in dynamically evolving spacetimes. In the fluid parts, reconstructions (PPM and ENO) combine with HLLE, Roe-type and Marquina flux formulae, and thus resemble many of the previously mentioned strategies.

1.2. MPI-AMRVAC philosophy

As is clear from the previous section, a whole suite of codes exist to date, which to varying degree invest in developing modern discretizations for relativistic (M)HD, improve on grid refinement strategies, and focus on challenging applications in high energy astrophysics. The MPI-AMRVAC code has as a guiding principle its versatility to handle any set of equations of the generic type

$$\partial_t \mathbf{U} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{S}_{phys}(\mathbf{U}, \partial_i \mathbf{U}, \partial_i \partial_j \mathbf{U}, \mathbf{x}, t), \tag{1}$$

where **U** denotes a set of (conserved) variables controlled by fluxes **F**(**U**), with strict conservation possibly destroyed by the presence of physical source terms **S**. The nature of this latter term is general enough to cover many situations commonly encountered in astrophysical settings. While any set of equations written in the above form can be handled in principle, the code currently has following functionalities:

- An advection equation module, $\mathbf{U} = \rho$ and $\mathbf{F} = \rho\mathbf{v}$ with **v** a uniform velocity vector. This module is suited for testing purposes, whenever novel functionality in terms of discretization or grid adaptivity is added;
- An adiabatic HD set, $\mathbf{U} = (\rho, \rho\mathbf{v})$ and $\mathbf{F} = (\rho\mathbf{v}, \rho\mathbf{v}\mathbf{v} + c_{\mathrm{ad}}\rho^{\gamma})$ with parameters c_{ad} and γ flexible enough to handle isothermal or polytropic scenarios;
- The usual Euler equations of hydrodynamics;
- The (resistive or ideal) classical magnetohydrodynamic equations;
- Special relativistic hydro and ideal MHD modules, in variants where differing equations of state are used. They form the topic of this overview article.

All these modules can be used in any dimensionality N_D, combined with any physically meaningful combination in terms of the number of vector components thereby employed. The dimensionality of the code, and the number of vector components, as well as the selected physics module, the basic grid block size, and a problem-specific user module is selected before the code is compiled by a (Perl) preprocessor [82]. This is the exact same strategy pioneered with the versatile advection code (VAC), initiated by Tóth [83]. The functionality further encompasses the use of Cartesian, polar, cylindrical (to which polar belongs), or spherical grids, with possibilities to treat axisymmetric or translationally symmetric ignored dimensions. In terms of source terms, precoded modules contain external gravity, both for planar atmospheres and as a point source, and optically thin radiative losses [88], besides the resistive terms present in the classical MHD routines. For every physics module, a number of high-resolution shock-capturing strategies are available, which can be chosen to differ from grid level to grid level in multi-level AMR simulations. In the remainder of this paper, we will illustrate this flexibility for special relativistic applications. As a final note on code philosophy, it is written in Fortran, employs MPI (Message-Passing Interface standard) for parallelization as discussed in detail below, and apart from requiring full MPI-2 functionality (including parallel I/O), does not rely on further libraries, making it easy to install on any kind of computer system, from laptops and desktops to massively parallel Unix-based platforms.

In the remainder of this paper, we provide details on all basic steps illustrated in the following skeleton code, generic enough to hold for any AMR code. The numbers in square brackets indicate section numbers where this aspect is discussed further below.

```
time_evolution: do
  exit time loop by user-defined criteria
  [2.2] compute timestep constraint for all grids
  [3.6] conditionally save data
  advance all grids for one time step dt
    distinguish source and flux addition strategies
      [2] select numerical scheme, geometric sources, ...
      [3.3 3.4] collect fluxes at fine—coarse boundaries
    after every parallel update of grids on all processors
      [2.2 3.4] fill all ghost cells for all grid blocks
    [3.3 3.4] at final temporal update, fix for conservation
  regrid
    [3.2] Quantify the error estimator on all grids
    Create new grid structure, ensure proper nesting
      [3.4] Load balance the new grid structure
    update time counters
end do time_evolution
```

2. Shock-capturing strategies for relativistic (M)HD

2.1. Governing equations

The special relativistic, ideal MHD equations, are of the general form given in Eq. (1), with the conserved variables **U** and fluxes **F**(**U**) obtained from the following expressions

$$\begin{aligned}\mathbf{U} &= (D, \mathbf{S}, \tau, \mathbf{B})^T = \left(\Gamma\rho, (\xi + B^2)\mathbf{v} - (\mathbf{v}\cdot\mathbf{B})\mathbf{B}, \xi + \frac{B^2}{2} + \frac{1}{2}(v^2B^2 - (\mathbf{v}\cdot\mathbf{B})^2) - p - D, \mathbf{B}\right)^T,\\ \mathbf{F} &= \left(D\mathbf{v}, \mathbf{S}\mathbf{v} - \frac{\mathbf{B}\mathbf{B}}{\Gamma^2} - (\mathbf{v}\cdot\mathbf{B})\mathbf{v}\mathbf{B} + Ip_{\mathrm{tot}}, \tau\mathbf{v} + p_{\mathrm{tot}}\mathbf{v} - (\mathbf{v}\cdot\mathbf{B})\mathbf{B}, \mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v}\right)^T.\end{aligned} \tag{2}$$

In these, the proper density ρ and pressure p are local rest frame values, while the three-vectors for velocity $\mathbf{v}$ and magnetic field $\mathbf{B}$ are viewed in the laboratory Lorentzian frame, which we exploit for the computations. The magnetic field is constrained by the Maxwell equation $\nabla \cdot \mathbf{B} = 0$. We indicate the Lorentz factor as $\Gamma = 1/\sqrt{1-v^2}$, with $v^2 = \mathbf{v} \cdot \mathbf{v}$ and the usual light speed $c = 1$ convention. The fluxes contain the total pressure p_{tot}, which reads

$$p_{\mathrm{tot}} = p + \frac{(\mathbf{v}\cdot\mathbf{B})^2}{2} + \frac{B^2}{2\Gamma^2}. \tag{3}$$

In the case of an ideal equation of state with constant adiabatic index γ, the specific enthalpy (h) related variable $\xi = \Gamma^2 \rho h$ becomes

$$\xi = \Gamma^2\left(\rho + \frac{\gamma p}{\gamma - 1}\right), \tag{4}$$

while the sound speed follows from $c_{\mathrm{g}} = \sqrt{\gamma p/\rho h}$. In our examples that follow, we also exploit the more general Mathews equation of state [50,54], where

$$1 + \epsilon = \frac{p}{(\gamma-1)\rho} + \sqrt{\left(\frac{p}{(\gamma-1)\rho}\right)^2 + 1}, \tag{5}$$

defines ϵ, appearing in the specific enthalpy as

$$h = \frac{1}{2}\left[(\gamma+1)(1+\epsilon) - (\gamma-1)\frac{1}{1+\epsilon}\right]. \tag{6}$$

In this approximation, the squared sound speed is given by

$$c_{\mathrm{g}}^2 = \frac{p}{\rho h}\left[\frac{\gamma+1}{2} + \frac{\gamma-1}{2}\frac{1}{(1+\epsilon)^2}\right], \tag{7}$$

while the locally effective polytropic index is then given by

$$\gamma_{\mathrm{eff}} = \gamma - \frac{\gamma-1}{2}\left(1 - \frac{1}{(1+\epsilon)^2}\right). \tag{8}$$

In these latter expressions, setting the parameter $\gamma = 5/3$ yields an excellent approximation to the Synge gas variation, which is the relativistically correct perfect gas description. The Matthews prescription even allows to mimic non-adiabatic effects, by using different γ values. A recent textbook discussing these governing special relativistic (M)HD equations, along with providing their covariant formulation that form the starting point for general relativistic treatments is [31].

We note that modern codes for handling relativistic (M)HD may offer various prescriptions for the governing equation of state (EOS). In [59], four equations of state were mentioned: (1) the ideal gas, constant polytropic index case from Eq. (4); (2) the actual Synge gas EOS involving the evaluation of modified Bessel functions [80]; (3) an interpolated EOS as introduced in [78]; and (4) their so-called Taub EOS (TM in their notation), which is the Mathews prescription [50,54] from Eq. (6) for $\gamma = 5/3$. The role of using a more realistic EOS in numerical relativistic MHD simulations was investigated in [63], where a constant polytropic index EOS was compared with the Mathews prescription on a selection of stringent test problems. A similar comparison for selected relativistic hydro shock tubes was undertaken in [77], where yet another, better approximation to the Synge EOS was proposed as well. In various recent relativistic hydro jet studies [56,57], the role of allowing for such locally varying effective polytropic index has been emphasized. Several ultra-relativistic tests were given in the appendix to [56], quantifying gamma ray burst shock physics where it impacts local compression rates (see also Section 4.1.3). In [57], it was discovered how this variation directly influences stability to rotation-induced, Rayleigh–Taylor type instabilities, when a rotating, hot jet is surrounded by a cold flow in a two-component structure (see also Section 4.1.4). The computationally more expensive Synge EOS is frequently avoided, a notable exception is found in the case study of a relativistic hydro jet by Perucho and Martí [72], where a two species prescription for a mixed lepton–baryon jet was taking the Synge variation for each species into account.

2.2. TVDLF, HLL, HLLC, HLLD and Roe-type Riemann solvers

The shock-capturing schemes to advance the system (1) with variables and fluxes from Eq. (2) all depend, in a minor up to major manner, on the knowledge of the characteristic wave speeds of the relativistic MHD system. We indicate schematically the most common strategies in use today, and give all details for the simplest among these, which directly generalize to other equations of the form (1).

The integration scheme advances the conserved variables $\mathbf{U}$, which represent grid cell values in the standard finite volume sense. In our implementation, a variety of multi-step, explicit schemes is available, including a fourth order Runge–Kutta scheme. For the most frequently exploited two-step predictor–corrector scheme, assuming that we use directionally unsplit flux additions, the numerical update takes the following form. A representative Hancock-TVDLF combination (with TVDLF denoting Total Variation Diminishing Lax-Friedrichs [84,93]) computes

$$\begin{aligned} U_{\mathbf{x}}^{n+\frac{1}{2}} &= U_{\mathbf{x}}^{n} + \frac{\Delta t}{2}\mathcal{S}(\mathbf{U}_{\mathbf{x}}^{n}) - \frac{\Delta t}{2\Delta V_{\mathbf{x}}}\sum_{1}^{N_D}\left(A_{i+1/2}F(\mathbf{U}_{i+1/2}^{L,n}) - A_{i-1/2}F(\mathbf{U}_{i-1/2}^{R,n})\right), \\ U_{\mathbf{x}}^{n+1} &= U_{\mathbf{x}}^{n} + \Delta t\mathcal{S}(\mathbf{U}_{\mathbf{x}}^{n+1/2}) - \frac{\Delta t}{\Delta V_{\mathbf{x}}}\sum_{1}^{N_D}\left(A_{i+1/2}F_{i+1/2}^{LF,n+\frac{1}{2}} - A_{i-1/2}F_{i-1/2}^{LF,n+\frac{1}{2}}\right), \end{aligned} \tag{9}$$

with $U_{\mathbf{x}}$ denoting a component of $\mathbf{U}$ in grid point $\mathbf{x}$. Source terms are indicated here with $\mathcal{S}$, which could e.g. be geometrical source terms in non-Cartesian settings (given below for a cylindrical case), and/or relate to physical sources $\mathbf{S}_{phys}$ from Eq. (1). The grid cell at $\mathbf{x}$ has cell volume $\Delta V_{\mathbf{x}}$, and is bounded in the N_D dimensions by edge centered surfaces/areas indicated by $A_{i\pm1/2}$. For Cartesian cases, the ratio $A_{i\pm1/2}/\Delta V_{\mathbf{x}} = 1/\Delta x_i$ with Δx_i the grid spacing in direction $i \in 1,\ldots,N_D$, and there are no geometrical source terms. The F then denotes the corresponding normal (to these surfaces) physical flux component read off from $\mathbf{F}$, which are in the above (non-conservative) Hancock predictor step, evaluated at cell interface states $\mathbf{U}_{i+1/2}^{L,n}$, $\mathbf{U}_{i-1/2}^{R,n}$. In the corrector step, the conservative total variation diminishing Lax-Friedrichs (TVDLF) [84,93] spatial discretization exploits fluxes $F_{i+1/2}^{LF}$ according to

$$F_{i+1/2}^{LF} = \frac{1}{2}\left[F(\mathbf{U}_{i+1/2}^{L}) + F(\mathbf{U}_{i+1/2}^{R}) - \epsilon^{LF}c_{i+1/2}^{\max}\left(U_{i+1/2}^{R} - U_{i+1/2}^{L}\right)\right]. \tag{10}$$

The locally computed $c_{i+1/2}^{\max}$ denotes the maximum physical propagation speed (in direction i) at the (averaged) cell interface state, hence

$$c_{i+1/2}^{\max} \equiv c^{\max}\left(\frac{\mathbf{U}_{i+1/2}^{L} + \mathbf{U}_{i+1/2}^{R}}{2}\right), \tag{11}$$

and is given by the fastest characteristic speed $c^{\max}$ in the system at hand. For MHD, this is then $c^{\max} = \max(|\lambda_F^-|,|\lambda_F^+|)$ where $\lambda_F^{\pm}$ indicate fast wave speeds. The computation of these characteristic speeds for relativistic MHD is explained in the following section. In the latter, dissipative part of the flux from Eq. (10), one can reduce dissipation by the parameter $\epsilon^{LF}(\leqslant 1)$. This is particularly relevant for computing steady states, where improved solutions can be obtained by gradual reduction of this parameter. One may also exploit $\max(c_{i+1/2}^{\max}(\mathbf{U}_{i+1/2}^{L}),\ c_{i+1/2}^{\max}(\mathbf{U}_{i+1/2}^{R}))$ in case the arithmetic averaging of conservative state values as in Eq. (11) would introduce a non-physical new state by numerical inaccuracies. It is worthwhile mentioning that the first order variant of this scheme was already in use in 1961 for 2D Euler problems [76], and the same scheme has been rediscovered and analysed in detail in [46].

To obtain the cell interface states from the cell center values, a limited linear reconstruction determines the $U_{i+1/2}^{L}$ and $U_{i+1/2}^{R}$ states as

$$\begin{aligned} U_{i+1/2}^{L} &= U_i + \overline{\Delta U}_i/2, \\ U_{i+1/2}^{R} &= U_{i+1} - \overline{\Delta U}_{i+1}/2, \end{aligned} \tag{12}$$

which involves a limited slope $\Delta\bar{U}_i = \Delta U_i\phi(r_i) \equiv \Delta U_i\phi(\Delta U_{i-1}/\Delta U_i)$. The limiter is here written to act on the conservative variable differences $\Delta U_i = U_{i+1} - U_i$ in the direction i, but in `MPI-AMRVAC` these can also be employed on the corresponding primitive variables $(\rho, \Gamma\mathbf{v}, p, \mathbf{B})$ (note the use of the spatial four-velocity $\Gamma\mathbf{v}$) or other user selected (e.g. logarithmically stretched) combinations. Many flavors are implemented, with two representative ones, realizing TVD on scalar problems, given by

$$\begin{cases} \overline{\Delta U_i} = \Delta U_i \max[0,\min(1,r_i)] & \text{minmod}, \\ \overline{\Delta U_i} = \Delta U_i \max\left[0,\min\left(\beta_{mc},\beta_{mc}r_i,\frac{1+r_i}{2}\right)\right] & \text{MC } \beta_{mc} \in [1.2,2]. \end{cases} \tag{13}$$

The minmod limiter is rather diffusive, but extremely robust. Sharper results can be obtained using the monotonized central (MC) [89] limiter. An even better (achieving third order accuracy on smooth profiles) limiter is the one introduced by Koren [45], recently generalized in [12]. This limiter actually distinguishes left and right interpolations in Eq. (12) such that

$$\begin{cases} \overline{\Delta U_i} = \Delta U_i \max\left[0,\min\left(2r_i,\frac{2+r_i}{3},2\right)\right] & \text{Koren in Left } U^L, \\ \overline{\Delta U_i} = \Delta U_i \max\left[0,\min\left(2r_i,\frac{1+2r_i}{3},2\right)\right] & \text{Koren at Right } U^R. \end{cases} \tag{14}$$

The reconstruction can also be done exploiting piecewise parabolic reconstructions (PPM) [15], and for the relativistic MHD case, formulae for such third-order reconstructions follow the prescriptions outlined in [59,67]. Switches for flattening the interpolation profiles for the primitive variables in the presence of strong shocks can then be selected, to prevent unphysical oscillations (in analogy to the dissipation algorithm in the appendix to Ref. [59] with the flattening parameter from [67]).

The geometrical source terms stem directly from the divergence formulae on the tensor $\mathbf{F}(\mathbf{U})$, and in relativistic MHD for the cylindrical case where we exploit (S_R, S_ϕ, S_Z) components for the $\mathbf{S}$ variable (similar for $\mathbf{B}$, $\mathbf{v}$) they are

$$\mathcal{S}^{\rm geo}_{S_R} = \frac{1}{R}\left(S_\phi v_\phi - \frac{B_\phi^2}{\Gamma^2} - (\mathbf{v}\cdot\mathbf{B})B_\phi v_\phi + p_{\rm tot}\right), \tag{15}$$

$$\mathcal{S}^{\rm geo}_{S_\phi} = -\frac{1}{R}\left(S_\phi v_R - \frac{B_\phi B_R}{\Gamma^2} - (\mathbf{v}\cdot\mathbf{B})B_\phi v_R\right), \tag{16}$$

$$\mathcal{S}^{\rm geo}_{B_\phi} = -\frac{1}{R}\left(v_\phi B_R - B_\phi v_R\right). \tag{17}$$

The expressions for the spherical case are found by analogy.

The TVDLF scheme hence uses minimal info on the wave speeds, and in combination with AMR and its inherent robustness due to its diffusive nature, it is readily usable for any system of conservation laws at minimal implementation costs. This method is used for relativistic MHD modeling in [86,87]. Our implementation has options for split or unsplit dimensional treatments, similar variety in adding source term strategies, and various more advanced schemes than the Hancock-TVDLF combination detailed above. In the rest of this section, some more schematic info is provided on other shock-capturing methods in active use today.

Maximal wave speed info is used in a full Roe-type approximate Riemann solver, where all characteristic wave speeds λ_p (7 in total for relativistic MHD) as well as the wave strengths α_p are deduced from the eigenvalues, as well as right and left eigenvector pairs of the flux Jacobian $\mathbf{F_U}$. Such a Godunov-type method for relativistic MHD was presented by Komissarov [43], and his paper collects all details needed for its implementation, such as the eigenvalue and eigenvector expressions and means to handle degenerate cases. Following this, improved, but purely 1D, approximate Riemann solver implementations are found in [6] and in [42], where the former is of TVD type. It is worth noting that recent research in full linearized Roe-type Riemann solvers presents renormalized sets of the eigenvector pairs, which behave properly in degenerate situations where some wave speeds coincide [5]. At the same time, the more involved scheme was found to have accuracy similar to more central evaluations, such as the HLLD solver adapted to relativistic MHD in [66]. HLLD, and the simpler HLL, HLLC solvers, make further approximations to their corresponding representation of the Riemann fan, as schematically illustrated in Fig. 1. These type of solvers originated in gas dynamics and Newtonian MHD, and have meanwhile been adapted to relativistic (M)HD. Depending on the amount of waves used to approximate the actual 7-wave fan, a corresponding amount of different fluxes are computed. One switches between their expressions according to the relative orientation of the wave signals in the (x, t) diagram. Appropriate recipes for computing meaningful intermediate states ensure desirable properties like positivity (positive pressures and densities), the ability to capture isolated discontinuities, etc.

Multi-dimensional relativistic MHD schemes exploiting these more central approximations to the Riemann fan (HLL and HLLC) emerged in [20,61,33]. While HLL uses only two wave speeds, HLLC includes the contact discontinuity as well. A five wave variant, HLLD, has recently been adapted to relativistic MHD in [66]. The 2-wave approximation boils down to the HLL solver (originally due to Harten et al. [32]). Employing both fastest wave speeds λ_F^- and λ_F^+, HLL switches the flux evaluation between $\mathbf{F}(\mathbf{U}^L_{i+1/2})$ when $\lambda_F^- \geqslant 0$, $\mathbf{F}(\mathbf{U}^R_{i+1/2})$ when $\lambda_F^+ \leqslant 0$, or a weighted flux expression

$$F^{hll}_{i+1/2} = \frac{\lambda_F^+ F(\mathbf{U}^L_{i+1/2}) - \lambda_F^- F(\mathbf{U}^R_{i+1/2}) + \epsilon^{hll}\lambda_F^-\lambda_F^+\left(U^R_{i+1/2} - U^L_{i+1/2}\right)}{\lambda_F^+ - \lambda_F^-}, \tag{18}$$

for the case where the Riemann fan has maximal and minimal waves traveling in opposite directions. In practical application, this HLL method has proven to work very similarly to the TVDLF method, demonstrating minor improvements over the latter, to which it reduces when $\lambda_F^- = -\lambda_F^+$. The parameter ϵ^{hll} has the same role as the ϵ^{LF}, to reduce dissipation. In MPI-AMRVAC, we implemented this HLL as well as the HLLC solver following [61], and introduced a hybrid variant of the latter. This hybrid variant, already applied in relativistic jet simulations in [56], in essence reduces locally to TVDLF in a region of user-set extent around any cell where the physical flux expressions for left and right edge-based reconstructed states have opposite sign

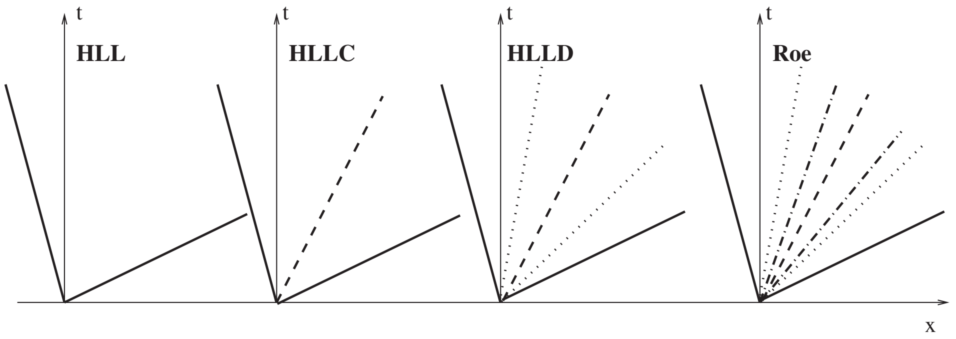

Fig. 1. A schematic view on the Riemann fan approximation for (relativistic) MHD in the (x, t) space employed by HLL, HLLC, HLLD, to full Roe-type approximate Riemann solvers. The amount of waves employed introduces a corresponding amount of different flux expressions, depending on the actual orientation of the wave signals with respect to the vertical (time) axis.

for any variable. For all non-relativistic modules, we also provide full Roe-type schemes, namely the single step TVD, as well as the TVD-MUSCL schemes, explained in detail in [84].

The time stepping being explicit, we encounter the usual Courant–Friedrichs–Lewy (CFL) constraint on the time step Δt. In fact, on the grid hierarchy, what is by default determined is

$$\Delta t = \min_{1,\ldots,n_{\text{leaf}}} \left(\frac{C}{\max\limits_{\text{block}} \left(\sum_1^{N_D} \frac{c_i^{\max}}{\Delta x_i} \right)} \right). \tag{19}$$

The $c_i^{\max}$ still denotes the maximal signal propagation speed in direction i, now determined at cell centers (in contrast to its cell-edge based value used in Eq. (10)). Note that in parallel simulations, we need to make a global communication to collect the overall instantaneous minimal value over all active grid leafs (see the data structure discussion below on this tree terminology), while per grid block, we determine the maximal physical propagation speed, and sum the contributions over all dimensions $i \in 1,\ldots,N_D$. The communication across processors is of a `MPI_ALLREDUCE` kind. Source terms introduce their own specific time step limits, in addition to this CFL constraint. To handle the possible occurence of sudden initial transients, a ramp-up to full CFL limited timesteps is provided as well. The Courant number C in the time step formula is allowed up to 1, values below 0.9 are commonly adopted.

Most discretizations discussed above have a stencil which involves two cells on each side of the cell to update. This stems directly from the limited linear reconstruction strategies, but when one selects the parabolic reconstruction method, up to three cells are needed on each side. This is important for providing boundary conditions, for which each grid block gets surrounded by a layer of ‘ghost cells’. The width of this layer can correspondingly be set to two grid cells, while wider stencil methods will need a larger ghost cell border. While the actual number of ghost cells needed directly relates to the method stencil, we restricted the implementation to contain an even number of ghost cells on each side (hence we use four ghost cells per side when PPM reconstruction is used, although 3 would suffice). On all interior boundaries, these cells are filled by copy/prolongation/restriction operations, depending on the refinement level of bounding grid blocks. The communication involved with that is aided by the data structures discussed in Section 3.3. At physical domain boundaries, a number of precoded possibilities can be selected per variable of the set $\mathbf{U}$, per domain boundary ($2N_D$ in N_D dimensions). Naturally, they can be user-manipulated as well, in which functionality for setting (parts of a) domain boundary using prescriptions on primitive variables instead of conservative variables proves particularly useful for relativistic problems. Precoded possibilities on conservative variables are

- (a)Symmetric mirroring of cell centered values with respect to the boundary, useful for reflective wall treatments;
- Continuous extrapolation in the direction perpendicular to the boundary (Neumann-type zero derivative), which mimics open boundaries (in characteristic wave based schemes, in effect realizing minimal to no reflection);
- Limited inflow conditions, meant to operate on velocity components perpendicular to (open) boundaries, where an extrapolation of the adjacent mesh cell value $U_{\mathbf{x}}$ is done according to max($U_{\mathbf{x}}$, $U_{\mathbf{x}} r_{\text{clip}}$) or min($U_{\mathbf{x}}$, $U_{\mathbf{x}} r_{\text{clip}}$) for a maximal or minimal boundary, respectively. Setting the clipping parameter r_{clip} to small positive or zero values allows turbulent flow structures to cross boundaries unhampered;
- Periodic boundaries, which then necessarily need to act on all quantities, on both directional sides of the domain.

Worth mentioning separately is the π-periodicity employed across ‘poles’ of the polar, cylindrical or spherical grids [86]. These fill ghost cells from 180° opposite internal mesh cells, using suitable (a)symmetric mirroring. This has the restriction that the number of grid blocks surrounding such a pole must be even. When specifying user-supplied boundary conditions, the user may also provide a further limit on the timestep, as the code does not include the boundaries when calculating its timestep restriction according to the CFL condition.

2.3. Conservative to primitive variables, characteristic speeds

In relativistic treatments, the conservative to primitive variable computation is a numerical problem by itself. It must be stressed that the accuracy with which this problem is solved numerically is a crucial element of modern solvers. Indeed, we have the obvious physical restrictions that $v < 1$, which means $\Gamma \geqslant 1$, while $p > 0$, $\rho > 0$, and we also want the electric field density $E^2 > 0$, where $E^2 \sim B^2 v^2 - (\mathbf{v} \cdot \mathbf{B})^2$. The latter is obviously true analytically, but numerical precision is finite. All these constraints must be consistent with $\tau > 0$ and $D \geqslant \rho$, and it helps to take explicit account of the numerical accuracy with which factors like $1 - v^2$ can actually be distinguished from zero. We build in our Newton–Raphson solver an appropriate “upper limit” on velocities, up to which conversions are computationally feasible. Similar tricks can be employed to guarantee consistent conversions from primitive to conservative variables, while ensuring a lower limit on attainable densities and pressures. This is similar to classical MHD, where this conversion is algebraic and trivial, but still requires that all contributions to total energy are positive separately, which may introduce numerical inaccuracies in regions of very low plasma beta. This problem returns in augmented form in relativistic settings, since rest mass, internal energy, kinetic energy, magnetic and electric field densities all may dominate in localized regions of the computational domain. The transformation from conservative to primitive variables for relativistic MHD has been studied separately in [69], where up to six different methods were compared. It was pointed out that sufficiently accurate and computationally fast treatments are available nowadays, although occasional unphysical primitive variable outcomes and/or lack of convergence in iterative inversions still require some additional correction procedure (see Section 2.4). The calculation complexity is also influenced by the EOS in use, and Ryu et al. [77] summarized how (1) a constant polytropic index case can be reduced to solving a single quartic for the velocity magnitude v; (2) that the Mathews prescription for $\gamma = 5/3$ boils down to a cubic equation in $\Gamma^2 - 1$; and (3) that their novel approximation to the Synge EOS prescription requires the solution of a computationally well-behaved eight-power expression for Γ. The transformation between primitive and conservative variables is completely analogous for general relativistic MHD, and e.g. the `ECHO` code adopted to general relativistic MHD has three inversion methods described in [21].

Our method for conservative to primitive variable transformation has been described in [87], which we sum up here. It parallels the recovery process given in [42,61]. To calculate the instantaneous values of the primitive variables (ρ, $\Gamma\mathbf{v}$, p, $\mathbf{B}$), it is helpful to define and carry along the auxiliary variables (Γ, ξ). Actually our relativistic hydro solver uses the set (Γ, p), while the MHD one takes the (Γ, ξ) combination. Note that following relations hold generally, independent of equation of state:

$$\mathbf{v} = \frac{\mathbf{S} + \xi^{-1}(\mathbf{S}\cdot\mathbf{B})\mathbf{B}}{\xi + B^2}, \tag{20}$$

$$\mathbf{v}\cdot\mathbf{B} = \frac{\mathbf{S}\cdot\mathbf{B}}{\xi}, \tag{21}$$

$$\frac{1}{\Gamma^2} = 1 - \frac{\left|\mathbf{S} + \xi^{-1}(\mathbf{S}\cdot\mathbf{B})\mathbf{B}\right|^2}{(\xi + B^2)^2}, \tag{22}$$

$$0 = \xi - p - \tau - D + B^2 - \frac{1}{2}\left[\frac{B^2}{\Gamma^2} + \frac{(\mathbf{S}\cdot\mathbf{B})^2}{\xi^2}\right]. \tag{23}$$

The first three relations merely re-express primitive variable expressions like $\mathbf{v}$, $\mathbf{v}\cdot\mathbf{B}$ and Γ in terms of conservative variables ($D, \mathbf{S}, \tau, \mathbf{B}$) and one auxiliary variable ξ. This is then used in the last expression, which rewrites the definition for the variable τ from Eq. (2). This latter rewrite now represents a non-linear transcendental equation which determines the local ξ value for given conservative variables ($D, \mathbf{S}, \tau, \mathbf{B}$). Recalling that $\xi = \Gamma^2\rho h$ and that $D = \Gamma\rho$, the final interdependence between ξ and the pressure p is fully specified by the equation of state used, such as Eq. (4) for an ideal gas, or relations (6) for the Mathews prescription. The roots of the latter transcendental equation for ξ are then determined by Newton–Raphson iterations. For details on possible bracketing and consistency checks, we refer to [87]. For completeness, we note that we locally switch to a viewpoint solving for p instead of ξ, in all points which have no magnetic field (this is the strategy everywhere in pure relativistic hydro runs), in that sense realizing multi-physics simulations where hydro and MHD regions coexist in the physical domain.

The more advanced solvers exploit knowledge of the wave speeds, which in relativistic MHD obey

$$-1 < \lambda_{\mathrm{F}}^- \leqslant \lambda_{\mathrm{A}}^- \leqslant \lambda_{\mathrm{S}}^- \leqslant \lambda_{\mathrm{E}} \leqslant \lambda_{\mathrm{S}}^+ \leqslant \lambda_{\mathrm{A}}^+ \leqslant \lambda_{\mathrm{F}}^+ < 1. \tag{24}$$

For direction i, entropy waves travel at $\lambda_{\mathrm{E}} = v_i$, while Alfvén signals are found from

$$\lambda_{\mathrm{A}}^{\pm} = v_i \pm \frac{1}{\Gamma^2}\frac{B_i}{\sqrt{\rho h + 2(p_{\mathrm{tot}} - p)} \pm (\mathbf{v}\cdot\mathbf{B})}. \tag{25}$$

The magneto-acoustic speeds are found from the quartic

$$0 = \rho h(1 - c_{\mathrm{g}}^2)\Gamma^4(\lambda - v_i)^4 - (1 - \lambda^2) \times \left\{\Gamma^2(\rho h c_{\mathrm{g}}^2 + 2(p_{\mathrm{tot}} - p))(\lambda - v_i)^2 - c_{\mathrm{g}}^2\left[\Gamma(\mathbf{v}\cdot\mathbf{B})(\lambda - v_i) - \frac{B_i}{\Gamma}\right]^2\right\}. \tag{26}$$

In relativistic hydro, this simplifies to a single quadratic expression, and then the use of exact analytic formulae for the root evaluation is straightforward. For the quartic, although such expressions exist, it is more easily obtained by numerical iteration, for which we use a Laguerre method. In the formulation above, the roots are known to be in $]-1, 1[$, but they can be awfully close to eachother or one. We implemented three equivalent formulations of this quartic expression, one for solving λ directly, one for solving in terms of $1/(1 - \lambda)$ which lies in $]-0.5, \infty[$ [10], and one which solves for $\Gamma(\lambda - v_i)$. A fourth option in `MPI-AMRVAC` does not directly work with the quartic above, but uses approximate expressions as in [27]. While being considerably faster, the inherent diffusion due to the approximations can be a severe limitation.

The calculation of the wave speeds is a cell-localized procedure. In case regions with pure hydro and pure MHD conditions exist within the simulated domain, we distinguish locally between five different means to quantify the wave speeds, depending on any combination of finite or vanishing magnetic field energy density B^2, finite or vanishing flow found from S^2, and finite or vanishing normal magnetic field component B_i. When either of these quantities attains a zero value, the quartic reduces to a simpler expression, allowing exact formulae for root evaluation instead of the Laguerre iteration.

2.4. Solenoidal field treatments and positivity fix options

For relativistic MHD, just as in classical MHD, the magnetic field obeys the additional constraint $\nabla \cdot \mathbf{B} = 0$. Many strategies exist to control numerically generated deviations from exact solenoidal fields, which would in the worst case lead to numerical instability. An extensive comparison of seven magnetic divergence control methods on nine test problems in multi-dimensional, non-relativistic MHD scenarios for fixed grids can be found in Tóth [85]. The treatments vary from insisting on machine precision accuracy in *one* particular pre-chosen discretization (constrained transport [23], hereafter CT), to global elliptic corrections (projection scheme), over local source term treatments. In [7], another assessment of four 'divergence cleaning' methods was made for non-relativistic MHD, and critical comments regarding their extension to AMR simulations were mentioned: elliptic corrections may form bottlenecks to parallel AMR simulations due to the global nature of the Poisson problem, while one staggered formulation was reported that naturally extends to AMR hierarchies. Obviously, local source term treatments are most easily transported to AMR frameworks, and a comparison between parabolic cleaning, the use of Powell source terms for momentum, energy and induction equations, or for the induction equation alone on classical MHD tests is in [40]. The parabolic method boils down to adding a diffusion-type term [51] to the induction equation as

$$\frac{\partial \mathbf{B}}{\partial t} = \frac{C_{\mathrm{d}}}{\Delta t}\left(\sum^{N_D} \frac{1}{\Delta x_i^2}\right)^{-1} \nabla\nabla \cdot \mathbf{B}, \tag{27}$$

where the coefficient $C_d \approx 1$. In the evaluation of the divergence, as well as for the evaluation of its gradient, we can exploit either mere central difference type evaluations, or use an evaluation based on the limited linear reconstructed cell edge magnetic field values. This parabolic method acts to locally damp errors at a rate which does not introduce a stricter CFL limit. A more general method that also advects, while damping, is the hyperbolic cleaning scheme from [16], which we implemented for classical MHD as well. This latter scheme augments the MHD system with an additional variable, and is referred to as a Generalized Lagrangian multiplier or GLM formulation. This approach also extends to relativistic MHD, and a recent GLM type scheme for non-relativistic MHD has been discussed in [65]. Other source term treatments, available in MPI-AMRVAC for both classical and relativistic modules are the addition of $\nabla \cdot \mathbf{B}$ proportionate terms. In the classical MHD case, one can choose to follow the original Powell [74] procedure adding sources to momentum, energy and induction equations, or only add a $-(\nabla \cdot \mathbf{B})\mathbf{v}$ term to the induction equation alone [36,17]. The latter option is in accord with Lorentz invariance, and is the one adopted to relativistic MHD. In the MHD testcases shown below, we indicate which strategy is taken for this type of error control.

Although we use source term treatments to handle $\nabla \cdot \mathbf{B}$ in our grid-adaptive relativistic MHD simulations, other strategies have proven effectiveness in related coding efforts. A constrained transport scheme is used in [20], which exploits finite difference discretizations for cell-centered variables, combined with cell corner based vector potential $\nabla \times \mathbf{A} = \mathbf{B}$ components. In their approach, the flux expressions for the electric field components which then govern $\partial \mathbf{A}/\partial t = \mathbf{v} \times \mathbf{B}$ are defined in a fully upwinded manner. Simpler constrained transport discretizations are discussed in [61]. The 'flux-CT' scheme introduced for classical MHD in [85] was carried over to relativistic MHD in the HARM code [27].

Another, similarly corrective, action is referred to as *positivity fixing*. This is merely an additional means to handle the supposedly rare instances where due to all nonlinearities of the scheme employed, the local conservative to primitive transformation signals a non-physical state. Our positivity fix approach can then be activated, and one such strategy operates as follows: identify all cells (within the same grid block) that represent physical states surrounding a faulty cell in a rectangular zone up to n_{pos} cells away; (2) convert those cells to primitive variables; and (3) for all but the magnetic field components, replace the faulty cell values by the average of surrounding physical state cells. Finally, revert to conservative variables where needed. Obviously, in this form, strict conservation may be violated. We will indicate for the tests below whether positivity fixing was allowed. Other strategies used by similar codes involve locally switching to first order fluxes, such as mentioned in [95].

As a final note on discretization method related details, improvements in high-resolution, shock-capturing schemes designed for multi-dimensional AMR simulations in relativistic MHD can still be expected. They typically will first mature in classical MHD, single grid codes, to then find their way to modern AMR frameworks. These may include extensions of Godunov type schemes to fully multi-dimensional, higher order accurate versions, such as those presented for classical MHD in [28]. The improvements can be in terms of higher-order limiters for finite volume treatments, like those found in [12]. Specific challenges may remain to carry these over to nested AMR grid hierarchies, while ensuring solenoidal field treatments, in combination with robustness and efficiency properties. Thereby allowing the schemes to vary from AMR level to level poses yet another level of complexity to this research.

3. Adaptive mesh refinement strategies

3.1. Introduction

Adaptive mesh refinement (AMR) comes in a fair variety of strategies, and can vary from cell-based AMR as exploited in fully unstructured grid codes, to the patch- or block-based strategies where on every grid level l, one in essence works with a collection of structured grids characterized by the same resolution Δx_i^l, with $i = 1,\ldots,N_D$. We will discuss pure octree, block-based oriented data structures in the sections that follow. This block-based refinement strategy lends itself well for parallelization, and details are given for MPI-AMRVAC. Previous AMRVAC code variants, not exploiting MPI, have experimented with patch-based over hybrid-block based strategies, as well as OpenMP parallelism [41], and we start by pointing out the main differences between those variants and the present AMR philosophy.

In an originally pure patch-based, dimension-independent implementation for Cartesian grids, we closely followed the pioneering work of Berger and Collela [9]. Their AMR error estimator uses a Richardson type comparison between two future, coarse grid solutions (see also the discussion in Section 3.2), and a sophisticated algorithm would organize those grid cells identified for refinement, into non-overlapping, logically rectangular, hierarchically nested grid patches by a series of group, bisect and merge operations [8]. The algorithm involved in this grid processing creates different sized grids, and tries to optimize the efficiency of newly created grids, defined as the number of points flagged for refinement, versus the total cell number in the new grid patch. The grid hierarchy would allow (1) any fixed even refinement ratio, possibly differing per grid level l; (2) ensure proper nesting, such that at least one cell at level $l-1$ occurs between level l and $l-2$ grid cells; and (3) to advance each grid level with its own time step Δt_l, proportional to the level l resolution Δx_i^l. This AMRVAC was used to quantify AMR efficiency obtained on a number of Newtonian hydro and ideal MHD test problems, and compared source term treatments for solenoidal magnetic field control on selected tests in [40]. OpenMP parallelism for virtually shared memory parallelization worked well with moderate CPU numbers, where one then parallelizes loops executed over all grids at a certain AMR level using combined static and dynamic scheduling [41].

This patch-based AMR implementation introduces relatively complex spatio-temporal interpolation formulae, to provide ghost cell values for individual patches from underlying coarser level grids. The time-stepping then uses finer timesteps on finer grids, and has to be executed in a well-defined order, such that temporal interpolations for intermediate time levels from coarser grids can be performed. The regridding can be executed from any grid level onwards, therefore change during the update of a single coarsest time step, all the while taking care of exact conservation properties. The latter is paramount for any AMR simulation targeting shock-dominated flow evolutions and implies that (1) new grids must be filled in a way ensuring conservation (the *prolongation* step), (2) the same must be done when coarsening grids (the *restriction* step), and (3) one must employ the more accurate fluxes pertaining to the finest grid level at both sides of fine–coarse boundaries. This flux-fixing update involves to update the coarsest cells immediately adjacent to fine–coarse level transitions at specific times.

In transforming AMRVAC to its current full octree MPI-AMRVAC code, an intermediate hybrid block-based AMR strategy was developed [86,55]. In this variant, we made the simplification that the grid reflects an incomplete octree, where possibly not all 2^{N_D} level $l+1$ blocks are created from a single level l block. The octree-type AMR structure removes a lot of the complexity associated with creating a properly nested grid structure, now ensuring proper nesting on the block level, but still involves spatio-temporal interpolations for ghost cell filling. This hybrid block approach combines (1) the optimal fit of new grids to evolving, localized patterns in fluid flow (which is a key feature of the patch-based approach), with (2) the more balanced block-based organization of the computations. Therefore, it was found that even for simple 2D advection problems, this hybrid block-based strategy is most efficient.

Currently, the MPI-AMRVAC variant employs pure octree block refinement, where all 2^{N_D} child blocks are activated when a block is identified for refinement. The strategy for refine/coarsen operations is explained further on. To simplify the parallelization, we gave up flexibility to allow different sized refinement ratios between grid levels, fixing it to 2. Moreover, to allow for direct parallelization over all grids at all levels, we now use the same time step Δt for all levels, dramatically simplifying the filling of ghost cells (no more temporal interpolations required). The flux-fixing operations are still identical to those in the other AMR strategies, while the prolongation and restriction formulae are designed to handle curvilinear grids as well.

In this section, we discuss those aspects of our AMR implementation that are generic enough to apply to any dimensional, parallel block-tree AMR code. In particular, we provide details on useful data structures, indicating their global versus local (per processor) character. All of these are suited for any curvilinear coordinate system, as it will be clear that none rely on explicit knowledge of the coordinates actually in use (Cartesian, polar, cylindrical or spherical), but merely reflect the tree structure of the block-AMR. We implicitly assume a fixed refinement ratio of two, as commonly assumed. Schematic figures for a 2D Cartesian case generalize straightforwardly to higher or lower dimensionality, and the discussion uses N_D to indicate the dimension at hand.

The actual prolongation formula, written out for a 2D case but instantly generalizing to any dimensionality, employs the $2^{N_D}=4$ fine cell volumes at the $l+1$ grid level, as well as the cell center coordinates in the curvilinear coordinate space (e.g. (r,ϕ) for a polar grid) as follows:

$$U^{l+1}_{i_f j_f} = U^{l}_{ij} + \sum_{1}^{N_D=2} \Delta\bar{U}_i{}^{l} 2 \underbrace{\frac{x^{l+1}_{i_f j_f} - x^{l}_{ij}}{\Delta x^{l}_{i}}}_{\pm 1/4} \left(1 - \frac{\Delta V^{l+1}_{i_f j_f}}{\sum_{(i)} \Delta V^{l+1}_{i'_f j'_f}}\right). \tag{28}$$

The volume summation indicated with $\sum_{(i)}$ sums up the 2 fine cell volumes along direction i. Fig. 2 schematically indicates the coarse grid points involved in this prolongation, and shows graphically the fine cell volumes $\Delta V^{l+1}_{i_f j_f}$ which in essence enter the prolongation formula, to determine the cell values at a fine grid point such as $x^{l+1}_{i_f j_f}$. The limited slope for both directions i

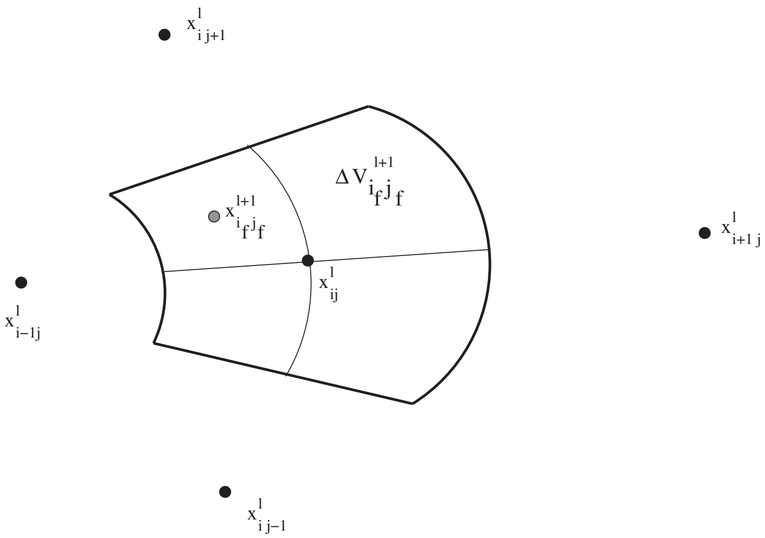

Fig. 2. The surrounding coarse cell coordinates and the partial cell volumes involved in the prolongation formula given by Eq. (28).

is determined on the coarse level l, based on a comparison between central, left and right slopes, which in the notation introduced after Eq. (12) when discussing slope limited linear reconstructions writes as

$$\overline{\Delta U_i} = \frac{\Delta U_i + \Delta U_{i-1}}{2} \max\left(0, \min\left(1, \frac{2}{1+r_i}, \frac{2r_i}{1+r_i}\right)\right). \tag{29}$$

This rewrites the MC limiter from Eq. (13) for $\beta_{mc} = 1$. One could also exploit any of the other TVD slope limiters in this prolongation process, but in practice the above choice is the one used in both the prolongation for ghost cell filling at fine–coarse boundaries, as well as for filling newly created level $l + 1$ grids from available level l data. Optionally, either of these two prolongation processes can be performed on the original conservative variables, on conservative variables where temporarily the energy density variable is replaced by a similarly conserved entropy density quantity, or even on the primitive variables (although this latter choice does not ensure strict conservation). This reconstruction can also be avoided, when a pure first order copying of coarse cell values suffices. They all ensure conservation in the sense that

$$\sum_{i_f}^{i_f+1} \sum_{j_f}^{j_f+1} U^{l+1}_{i_f j_f} \Delta V^{l+1}_{i_f j_f} = U^l_{ij} \Delta V^l_{ij} = U^l_{ij} \sum_{i_f}^{i_f+1} \sum_{j_f}^{j_f+1} \Delta V^{l+1}_{i_f j_f}. \tag{30}$$

The restriction formula merely involves this same expression, using it to obtain the coarse cell values U^l_{ij}. As pointed out in [86], these prolongation-restriction formulae can be used in curvilinear coordinate systems where the Jacobian involves separable coordinates.

3.2. Refinement criteria

The efficiency gain and accuracy preservation of AMR computations, as compared to equivalent uniform effective resolution runs, ultimately relies on the criteria controlling grid refinement and coarsening operations. In block-tree, fixed refinement ratio two simulations, a fair variety of error estimators are used in existing frameworks. We give a summary of the strategies available in `MPI-AMRVAC`, which are sufficiently general to apply to other similar coding efforts.

The block-tree nature implies that a decision for refining/coarsening is to be made on a block-by-block basis. This automated block-based regridding procedure involves three steps:

- Consider all blocks at level $1 \leqslant l < l_{\max}$, with $l_{\max}$ the maximal grid level selected;
- Quantify the local error $E_{\mathbf{x}}$ at each gridpoint $\mathbf{x}$ in a certain grid block;
- If any point has this error exceeding a user-set tolerance $E_{\mathbf{x}} > \epsilon_l$, refine this block (and ensure proper nesting);
- If all points have their error below a user-set fraction of the tolerance used in the previous step, $E_{\mathbf{x}} \leqslant f^{\epsilon}_l \epsilon_l$, coarsen the block (for $l > 1$).

This involves two sets of possibly level-dependent parameters, namely the error tolerance per level ϵ_l, and the coarsening fraction $f^{\epsilon}_l < 1$. The local error estimator can be one of the following three options, which differ in the amount of time levels involved in the estimator. One can employ a Richardson-based estimator, quantifying errors at a future time level t^{n+1}, using variables stored both at previous t^{n-1} and current t^n time levels. A second choice involves only these latter two time levels, and employs local comparisons. The third choice available is a Löhner type estimator [48] (also used in FLASH [13], RAM [95], among others) which only uses the present time level t^n, and in effect computes normalized second derivatives to locate strong variations. Any of these estimators, given in formulae below, uses a user-selected subset of the conserved or auxiliary variables (or even variables that are computed dynamically at the time of regridding), through the formula

$$E_{\mathbf{x}} = \sum_{iw} \sigma_{iw} E^{\mathrm{Rel}}_{\mathbf{x},iw}. \tag{31}$$

This implies the quantification of local relative variable errors $E^{\mathrm{Rel}}_{\mathbf{x},iw}$ with iw a variable index, and weights obeying $\sum_{iw}\sigma_{iw} = 1$. It should be noted that all error estimators, which thus ultimately quantify $E^{Rel}_{\mathbf{x},iw}$ (the local relative error per grid point, per variable) can easily be augmented with user-coded (de) refinement. The latter is of particular relevance for aiding the grid refinement at time $t = 0$, when e.g. the lowest grid level would not even resolve the actual variation in the initial condition. It also proves useful when introducing temporally varying, moving source terms, or specific boundary treatments which require a minimal level resolution, etc. All estimators that use instantaneous critical gradient values, as commonly encountered in AMR applications, fall in this user-augmented refinement category.

The estimators mentioned then work as follows. The Richardson procedure computes two future solutions at t^{n+1}, by using the t^{n-1} solution, coarsening it to a $2\Delta x_i$ grid, and then integrating this solution using $2\Delta t$. This coarsened-integrated solution is denoted as $w^{\mathbf{CI}}$. We also start with the t^n solution, integrate this with Δt, and subsequently coarsen it to $2\Delta x_i$. This integrated-coarsened solution is referred to as $w^{\mathbf{IC}}$. The local relative variable errors $E^{\mathrm{Rel}}_{\mathbf{x},iw}$ are then found from

$$E^{\mathrm{Rel}}_{\mathbf{x},iw} = \frac{|w^{\mathbf{CI}}_{iw} - w^{\mathbf{IC}}_{iw}|}{\sum_{iw}\sigma_{iw}|w^{\mathbf{IC}}_{iw}|}. \tag{32}$$

The integrator used in this Richardson estimator is for our purposes a first order, dimensionally unsplit scheme, only incorporating unsplit source terms.

The local comparison merely employs the availability of the t^{n-1} and t^n solution vectors. It estimates the local relative variable errors as

$$E^{\mathrm{Rel}}_{\mathbf{x},iw} = \frac{|w^{n-1}_{iw} - w^{n}_{iw}|}{|w^{n-1}_{iw}|}. \tag{33}$$

This is obviously computationally cheaper, but has the disadvantage that it in essence uses historical info, which may be insufficient for rapidly moving, strong shock cases. In our experience, both Richardson or local error estimators work satisfactorily on a variety of test problems, but both may need an added, user-set buffer zone around each grid point flagged for refinement in this manner. This zone sets the buffer width in numbers of grid cells (per dimension) n^i_{buff}, about flagged grid cells. A non-zero buffer width introduces a difference in the communication pattern associated with this hierarchical grid creation. To that end, it is helpful to have a logical array of refine flags, covering all possible grids for all processors. When no buffer zone is present, grids actually behave independent as far as their flagging for refinement/coarsening is concerned, and the decision to refine a grid is then communicated by a MPI_ALLGATHER operation on these logical refine flags. When buffering is active, we first make the local refine flags of the neighbor grids consistent within the local memory, and next perform an MPI_ALLREDUCE operation on these arrays using a 'logical or' comparison.

The third estimator, which we advocate by default, is the Löhner [48] prescription as adjusted in the PARAMESH [49] library or the FLASH3 [24] code. In our experience, it does not require any of the buffering just discussed, and is computationally efficient as it employs only instantaneous values from t^n. It discretizes a weighted second derivative in grid point $\mathbf{x}$ given by

$$\sqrt{\frac{\sum_{i_1}\sum_{i_2}\left(\Delta x_{i_1}\Delta x_{i_2}\frac{\partial^2 w}{\partial x_{i_1}\partial x_{i_2}}\Big|_{\mathbf{x}}\right)^2}{\sum_{i_1}\sum_{i_2}\left[\Delta x_{i_1}\left\{\left|\frac{\partial w}{\partial x_{i_1}}\Big|_{\mathbf{x}-\Delta x_{i_1}}\right| + \left|\frac{\partial w}{\partial x_{i_1}}\Big|_{\mathbf{x}+\Delta x_{i_1}}\right|\right\} + f_{\mathrm{wave,l}}|\bar{w}|_{i_1,i_2}\right]^2}}. \tag{34}$$

The indices run over all dimensions i_1, $i_2 = 1,\ldots,N_D$, and the denominator contains an average value obtained from all surrounding 'corners', i.e. for $N_D = 2$ this writes

$$|\bar{w}|_{i_1,i_2} = |\bar{w}|_{\mathbf{x}+\Delta x_{i_1}+\Delta x_{i_2}} + |\bar{w}|_{\mathbf{x}+\Delta x_{i_1}-\Delta x_{i_2}} + |\bar{w}|_{\mathbf{x}-\Delta x_{i_1}+\Delta x_{i_2}} + |\bar{w}|_{\mathbf{x}-\Delta x_{i_1}-\Delta x_{i_2}}. \tag{35}$$

Note that Eq. (34) multiplies this average with a level dependent 'wavefilter' parameter $f_{\mathrm{wave,\,l}}$, which can be of order 10^{-2} typically. The discrete formulae used to evaluate Eq. (34) then write as follows. One computes the local relative variable errors ${E_{\mathbf{x},iw}}^{\mathrm{Rel}}$ from

$$E^{\mathrm{Rel}}_{\mathbf{x},iw} = \sqrt{\frac{N_{iw}}{\max(D_{iw},\epsilon)}}, \tag{36}$$

with numerator

$$N_{iw} = \sum_{i_1}\sum_{i_2}\left[\Delta_{i_1}(\Delta_{i_2} w_{iw})\right]^2, \tag{37}$$

and denominator

$$D_{iw} = \sum_{i_1}\sum_{i_2}\left[S_{i_1}(|\Delta_{i_1} w_{iw}|) + f_{\mathrm{wave,l}} S_{i_2}(S_{i_1}|w_{\mathrm{iw}}|)\right]^2. \tag{38}$$

These formulae contain discrete central differences Δ_i, introduced per dimension i, as well as sum operators S_i. In a 2D case, they realize $\Delta_{i_1=1}v(i,j) \equiv v(i+1,j) - v(i-1,j)$ and $S_{i_1=1}v(i,j) \equiv v(i+1,j) + v(i-1,j)$.

As a final note on these error estimator procedures, all of them may use logarithmically stretched values for (positive) variables, of interest for gravitationally stratified situations. Furthermore, the tolerance ϵ_l parameter in our Löhner estimator is usually of order 0.1, but needs to be set to a smaller value when the same problem is run using Richardson or local comparisons.

3.3. Data structures for parallel, block-tree AMR

In this section, we discuss those aspects of our AMR implementation that are generic enough to apply to any dimensional, parallel block-tree AMR code. In particular, we provide details on useful data structures, indicating their global versus local (per processor) character. All of these are suited for any curvilinear coordinate system, as it will be clear that none rely on explicit knowledge of the coordinates actually in use (Cartesian, polar, cylindrical or spherical), but merely reflect the tree structure of the block-AMR. We implicitly assume a fixed refinement ratio of two, as commonly assumed. Schematic figures for a 2D Cartesian case generalize straightforwardly to higher or lower dimensionality, and the discussion uses N_D to indicate the dimension at hand.

The overall domain is considered to be 'rectangular', i.e. bounded by coordinate pairs $[x^i_{\text{prob, min}},\ x^i_{\text{prob, max}}]$ in each dimension $i \in \{1,\ldots,N_D\}$. On the lowest grid level $l = 1$, one controls the coarsest resolution $\Delta x^i_{l=1}$ as well as a suitable domain decomposition, by specifying both the total number of level 1 grid cells $N^i_{l=1}$ along with the individual block size N^i_{b}, per dimension i. Obviously, the total cell number must be an integer multiple of the block size. Following the refinement strategies explained in the previous section, a hypothetical 2D domain is shown in Fig. 3, which corresponds to a domain where 4 × 3 blocks are exploited in this domain decomposition, and where local refinement was activated in four out of these level $l = 1$ blocks, here in the top right domain corner, as well as in one level $l = 2$ grid. Global, integer grid indices are introduced per dimension, in a manner where knowledge of these grid indices, combined with AMR level knowledge, instantly allows one to localize the grid when needed. Following Fig. 3 (left panel), the grid on level $l = 2$ indicated by global grid indices (5, 3) is indeed the fifth grid block horizontally, and the third vertically, when the domain would be resolved fully with level $l = 2$ blocks. Note that for level $l > 1$ grid blocks, the global indices for the underlying parent block can be deduced by a simple integer operation, e.g. for all 4 level $l = 3$ grids shown, the underlying parent level $l = 2$ grid indices, which are (6, 4), follow from $g^i_{l-1} = [(g^i_l - 1)/2]_{\text{int}} + 1$, where $[\cdot]_{\text{int}}$ is Fortran's `int` operation. The right panel of Fig. 3 reflects the tree representation of the hypothetical grid hierarchy shown at left, where the presence of a grid leaf at a certain grid level is identified through a boolean variable. As indicated before, the total number of active grid leafs n_{leaf} may change from timestep to timestep.

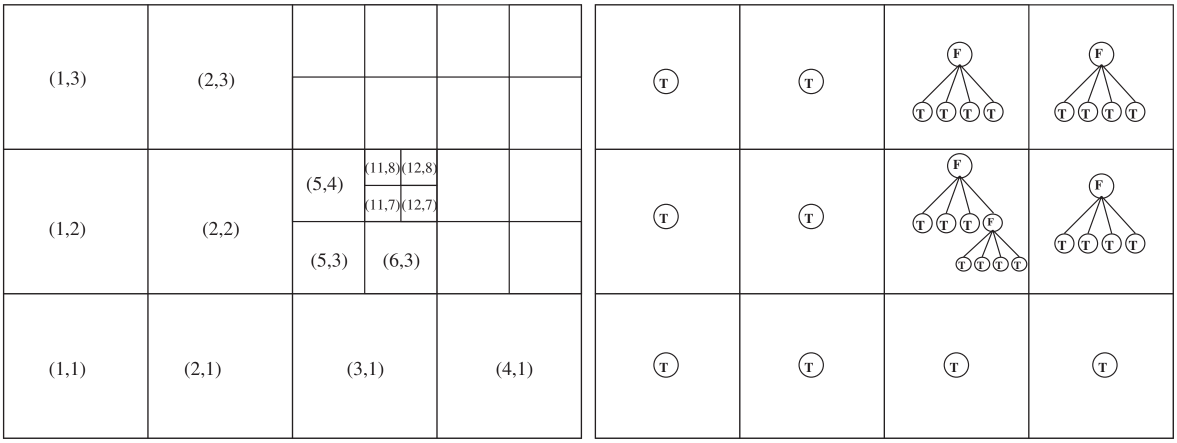

Fig. 3. For a hypothetical grid arrangement exploiting 4 × 3 grid blocks at level $l = 1$, the left panel shows the global grid indices, while the right panel gives the tree representation with boolean variables indicating grid 'leafs'.

To facilitate the communication patterns associated with flux fixing across level boundaries, as well as ghost cell exchange between blocks, it is convenient to work with two sets of local grid indices. The first set of local grid indices identifies a block and its neighbor blocks on the same grid level l by N_D integers from $\{-1, 0, 1\}$. This is illustrated for the 2D case in the left panel of Fig. 4, where the central grid block is shown as (0, 0) and the local grid indices are given for all eight neighbor blocks. Using these local grid indices, one directly computes the corresponding global grid indices for selected neighbor blocks by integer additions. For each block, we also introduce a set of local child indices $g^i_{\mathrm{loc},c}$, which are for the 2D case (1,1), (2,1), (1,2), (2,2) for bottom left, bottom right, top left, top right child grid, respectively. These local children indices form part of a larger pattern shown in the right panel of Fig. 4, which continues this local child index scheme across a coarse–fine interface. Note that one can compute the local indices by modulo operations on the global grid indices: e.g. the near-center level $l = 2$ grid indicated by (5,3) in Fig. 3 corresponds to a (1,1) child of its parent level $l = 1$ grid, in accord with $g^i_{\mathrm{loc},c} = 1 + \mathrm{mod}_2(g^i_{l=2} - 1)$. In a completely analogous manner, the combined use of global indices, the two sets of local indices, and integer operations introduce a flexible and efficient way to gather knowledge on grid neighbors, their location in the tree structure and in physical space, and deduce the needed communication patterns for boundary filling and flux fixing.

To determine these communication patterns, pointers traverse the tree structure in a variety of ways. Any grid block at level l will store a selection of pointers, and this in addition to (1) its N_D global grid indices g^i_{l}; (2) its level l; and (3) its presence or absence as an active leaf in the tree. These pointers give info on (1) the $2N_D$ directional neighbors of the grid block at the same grid level; (2) its parent grid (if its level $l > 1$); and (3) its 2^{N_D} children grid blocks (if $l < l_{\max}$). The $2N_D$ directional neighbors are shown schematically in 1D, 2D and 3D in Fig. 5, along with the parent and children pointers for a 2D case. Exploiting these parent and children grid block pointers, together with the directional neighbor knowledge, full grid neighbor information can be collected. Every grid block in fact differentiates the neighboring grids according to their type. One distinguishes up to 4 differing types, all requiring separate ghost cell treatment. These four neighbor types are illustrated in Fig. 5 as well, and correspond to (1) a grid-domain boundary and (2) a fine–coarse transition, both of which use a null neighbor pointer; while (3) same refinement level, as well as (4) coarse–fine interface transitions require a pointer to the same level neighbor in that direction. Due to the proper nesting of grids, this then suffices to identify all surrounding grid blocks. These pointers are actively used when ghost cell filling and flux fixing steps (happening for all grid blocks simultaneously) invoke inter-processor communication.

In a parallel implementation, the details of which are provided next, we also introduce for each grid block a processor-specific grid number g_{ipe}, ranging up to a fixed parametrized value g_{high}. These are useful in loops that must be performed over all grids residing on the local processor, like advancing all grids over a (partial) timestep, or adding split source terms, or filling all ghost cells, etc. The tree structure known to all processors then includes the processor number *ipe* on which the grid is dynamically allocated in memory. Finally, some operations benefit from having a linear, linked list possibility to traverse the tree on a level by level basis. To that end, each grid also contains a pointer to the previous and next grid created in the same AMR level, taking all grids on all processors into account. This linked list is complemented with a globally known pointer to the first (head) and last (tail) grid on each level. For the hypothetical grid structure used in Figs. 3 and 4, this corresponding linked list representation is shown in Fig. 6.

3.4. Parallelization details

The parallelization of the AMR simulation introduces yet one more means to traverse the tree structure. This comes about from the load balancing strategy, for which a space-filling curve is used that goes through all active grid blocks. Such space-filling curves exist in quite a variety, and in effect provide a linear arrangement of all grid blocks along them, introducing a new set of global grid numbers $g_{\mathrm{sfc}} \in [1, n_{\mathrm{leaf}}]$. In MPI-AMRVAC, we adopted a fairly straightforward *Z*-order or Morton-order

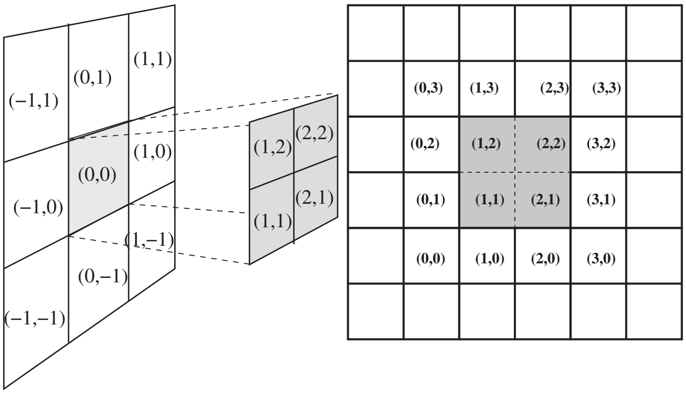

Fig. 4. Left: the local grid indices used to identify neighbor grid blocks at the same grid level, and for the $2^{N_D} = 4$ children. The latter pattern is used in enlarged form across coarse–fine boundaries, as indicated at right.

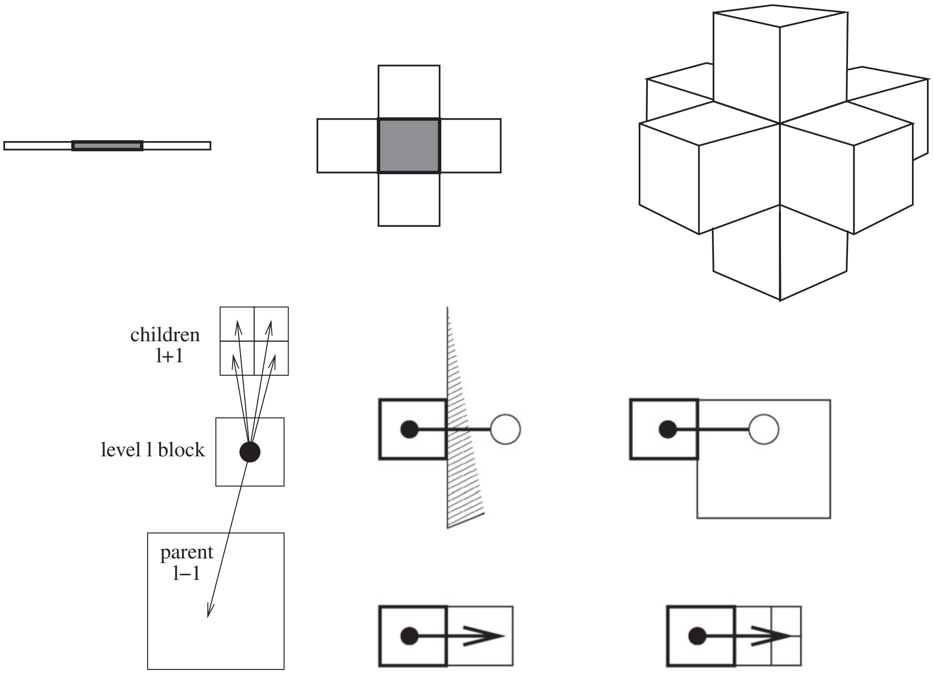

Fig. 5. Top: the directional neighbors, for 1D, 2D and 3D configurations. Bottom left: the pointers used to keep track of parent and children grid blocks. Bottom right: the four types of neighbor blocks, depending on physical boundary presence, fine–coarse, same level, or coarse–fine transitions. The latter two introduce a neighbor pointer.

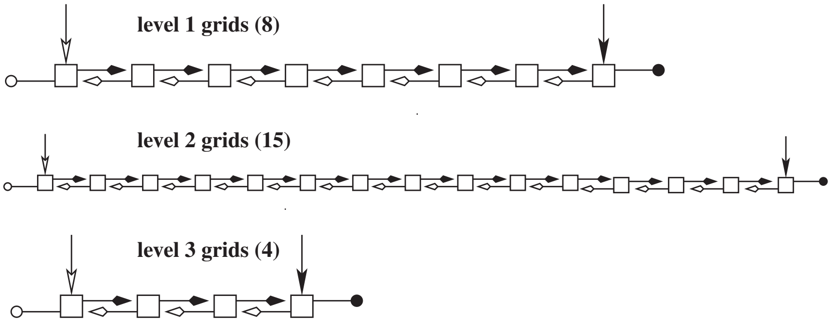

Fig. 6. For the same grid structure as in Fig. 3, the linked list representation.

space filling curve, which has the distinct advantage that the computation of the grid Morton number g_{sfc} is easily accomplished by a recursive subroutine. This recursively descends from the root (level $l = 1$) to the leaf levels down the tree structure, as identified by the boolean tree representation shown in Fig. 3. For this same hypothetical grid structure, the Morton space-filling curve is illustrated in Fig. 7, along with the resulting distribution of these 27 grid blocks on four CPUs. Load-balancing is done after every timestep, following the adaptive remeshing. When exploiting N_{p} CPUs, our strategy for load balancing merely ensures that each CPU has at least $[n_{\mathrm{leaf}}/N_{\mathrm{p}}]_{\mathrm{int}}$ grid blocks, while the remainder $\mathrm{mod}_{N_{\mathrm{p}}}(n_{\mathrm{leaf}})$ grid blocks increase this number by 1 for the first as many CPUs. In the example of Fig. 7, this implies that the first three CPUs each contain seven grid blocks, while the fourth has 6. The load balancing strategy implies the use of `MPI_ISEND`, `MPI_IRECV` pairs, which use block types created by `MPI_TYPE_CREATE_SUBARRAY`, `MPI_TYPE_COMMIT` pairs. These latter make the send/receive operations aware of the block size, the number of ghost cell layers, and the number of physical variables in the solution arrays.

It is of interest to point out that each active grid leaf is now known through three integer valued identifiers: we mentioned how each grid has N_D global indices, g_l^i, that we have a processor-specific grid number g_{ipe}, and another global index along the space filling curve g_{sfc}. For convenience, we introduced pointers that allow easy transition from one to another index scheme. On each processor, we further store the begin and end index along the space filling curve of the grid blocks residing in local memory.

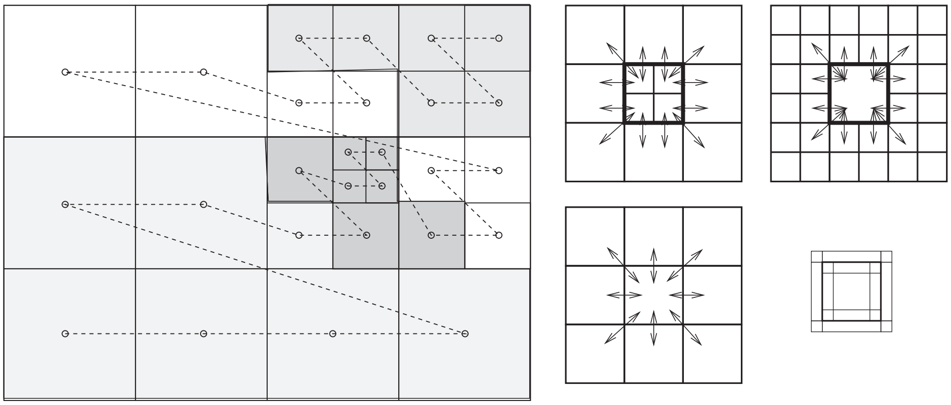

Fig. 7. For the grid arrangement from Figs. 3–6, the Morton ordering of the grid blocks, as well as its resulting distribution over four CPUs, indicated with different grayscales. Right: the communication patterns associated with boundary filling for different grid nestings.

Besides the communication just discussed for load-balancing, the parallel AMR needs send/receive operations for ghost cell filling, when neighboring grid blocks (which differ at most by one grid level) reside on different processors. It also requires collection of fluxes to restore global conservation at each fine–coarse, or coarse–fine interface. The possible communication patterns are in essence straightforward, due to the enforced proper nesting. The three possibilities are schematically indicated in Fig. 7, right panels. For clarity, we idealized this schematic to reflect a fine–coarse situation, a coarse–fine one, and a same refinement level case all around the central grid(s), but it should be evident that in practice mixed situations arise (consider e.g. the case of the level 2 grid (5, 3) from Fig. 3). A fourth neighbor type, as already discussed with the pointer strategy shown in Fig. 5, arises adjacent to a physical domain boundary, i.e. at $x^i = x^i_{\mathrm{prob,min}}$ or at $x^i = x^i_{\mathrm{prob,max}}$, for which no interprocessor communication is needed. Fine-coarse transitions illustrated require the central finer grids shown to receive prolongations, and send restrictions to their surrounding coarser grids. Same level neighbor grids exchange boundary layers by copy, while coarse–fine transitions ask the central coarse grid to send prolongations, and receive restrictions. Note that each of the arrows representing a specific communication pattern in the right panels of Fig. 7 in our implementation introduces the corresponding block `MPI_ISEND`, `MPI_IRECV` patterns, with differing block types introduced by `MPI_TYPE_CREATE_SUBARRAY`, `MPI_TYPE_COMMIT` pairs. In the case of a same refinement level exchange, these types identify the ghost cell zones appropriate at a minimal or maximal (considered per dimension) boundary, or across the corners, as illustrated in the bottom right panel of Fig. 7. At the finalization of a (partial) timestep, a boundary filling for all grids on all processors is done, which on entry creates all the required block types for the differing boundary patterns, and on completion releases all the block types, using `MPI_TYPE_FREE`. Since all grid blocks have the same size, each processor can generate the needed block types on entry of the ghost cell filling procedure. Thereafter, each processor loops over all its locally residing grid blocks, and determines from the grid block neighbor types the needed (physical boundary, copied, prolonged, or restricted) boundary communications. All processors thus send and receive their own locally needed ghost cell values, from and for all their local grid blocks, with carefully positioned `MPI_WAITALL` synchronization points.

For the flux-fixing operations, active across the fine–coarse or coarse–fine boundaries only (indicated with thicker lines in Fig. 7, top right panels), the finer meshes send, while the coarser mesh receives the more accurate fluxes to update the coarse cells immediately within one cell layer adjacent to these boundaries. Contrary to the restriction, prolongation, or copy operations mentioned in the previous paragraph which all in essence work with N_D-dimensional patches of the grid blocks, as they exploit cell-centered quantities, the fluxes are cell edge based, and need the exchange of $N_D - 1$ dimensional edges. All shock-capturing methods collect and pass the fluxes used to update the cell centered quantities adjacent to fine–coarse cells to flux pointers, introduced per grid block, per dimension, and one for minimal and maximal sides, respectively. These can be used directly for flux fix operations on the local processor, and are the ones which are send to neighboring grids on remote CPUs. For the latter, at the receiving coarse grid, we collect these accumulated fluxes in a linear buffer array. When correcting the coarse cells next to level transitions, this linear buffer is then reshaped to the corresponding mesh positions (i.e. we exploit Fortran `reshape (source = ···, shape = ···)` operations).

3.5. Efficiency quantification

To demonstrate code performance, Fig. 8 quantifies speedup behavior for some representative simulations. The left panel uses the AMR as a domain decomposition strategy (1 grid level), for a full 2D, classical MHD simulation taking roughly 1000 timesteps with a 384 × 384 resolution, including I/O. The setup is the frequently encountered Orszag-Tang test problem (top panel shows the pressure at the endtime simulated), for which we discuss a relativistic MHD variant in Section 4.2.3. Note that we can model this planar, non-relativistic MHD test in less than 5 s on 64 processors. The diagonal dotted line corresponds to perfect scaling, and the full simulation (indicated as 'run' in the figure) matches this perfect scaling, with cache

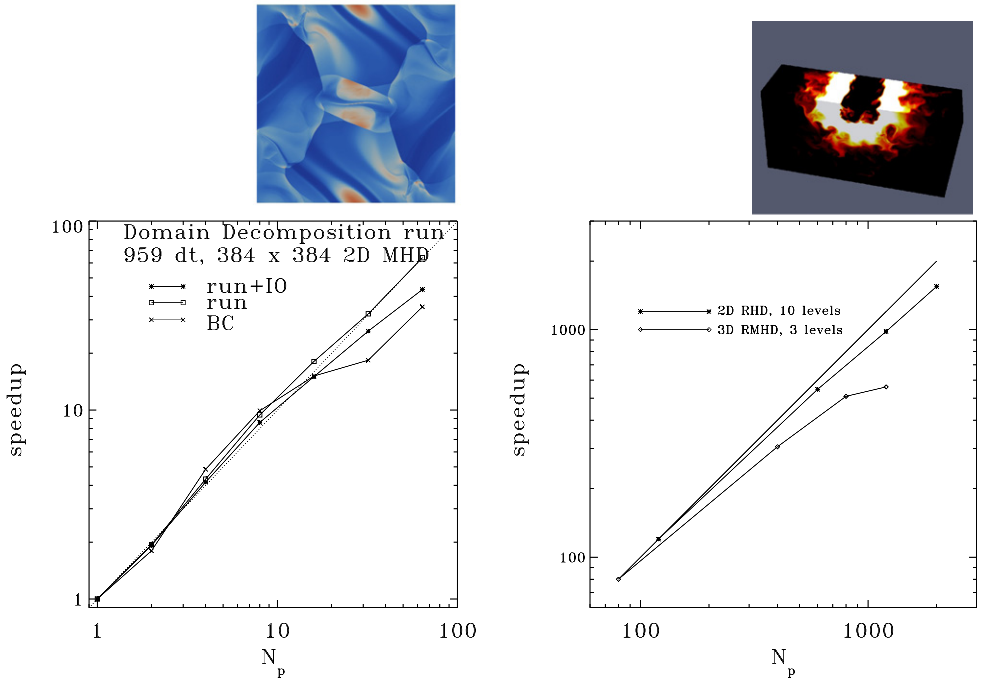

Fig. 8. Parallel performance for MPI-AMRVAC in terms of obtained speedup when using increasing numbers of CPUs (N_p). Left panels are for a domain decomposition run of a 2D classical MHD test, for which the final ($t = 3.1$) pressure distribution is shown on top. Right panels are for both a 2D and 3D relativistic (M)HD run, where up to 2000 processors are exploited at 80% efficiency. An impression of the 3D jet simulation is shown on top right.

effects influencing the scaling at low processor numbers. The 'run + IO' gives scaling for a simulation where both logfile and full data dumps occur repeatedly during the simulation (both are described in Section 3.6 below), while the 'BC' curve just reports the scaling of the boundary conditions alone, which in this domain decomposition case covers all internal copy operations and the treating of the periodic sides. Note that all curves are rescaled to unity for a single CPU: the actual time needed for realizing the boundary conditions does not form a major part of the total execution time.

The right panel in Fig. 8 contains timings, part from actual production runs on the CINES Jade supercomputer, for:

- A 3-level, 3D relativistic MHD run at 480 × 480 × 384 effective resolution (see inset, modeling a two-component jet, for which we give a 2.5D test in cylindrical geometry in Section 4.1.4);
- A 10-level 2D relativistic hydro run (targeting GRB shock dynamics [58], for which we provide a 1D stringent test in Section 4.1.3).

The latter problem is sufficiently large in overall block number to demonstrate sustained 80% efficiency up to 2000 CPUs. It should be noted that having sufficiently many, large enough blocks per processor is the key point to demonstrate scaling for larger CPU numbers. All scalings reported here kept the problem size fixed when going to more processor numbers, hence report strong scaling experiments. The space filling curve employed may also influence scaling on specific tests (see e.g. the discussion on their locality properties in [92]), as the example CPU layout from Fig. 7 shows that more optimal distribution patterns are possible in certain cases.

3.6. I/O specifics

For large-scale simulations, the I/O operations themselves require special attention. In any simulation, one may distinguish among three types of I/O procedures: one ensuring sufficiently complete data file dumps for restarts or later post-processing, one collecting globally computed monitor quantities useful for later data analysis and interpretation, and a third category related to visualization/post-processing only.

As the latter two are application specific, we merely mention that a significant part of the I/O subroutines within MPI-AMRVAC allow a broad variety of post-processing on the data file format used for restart. The restart dumps are in fact the only type of output file that MPI-AMRVAC writes during the simulation, but the same executable can postprocess afterwards for analysis and visualization. This even provides the possibility to do a restart in more dimensions than the original simulation. For pure postprocessing, we provide possibilities to convert to vtu (unstructured VTK) format, in parallel (using a master–slave pattern) or single CPU execution mode, which are directly readable by open-source (parallel) visualization software like `Paraview` [71] or `Visit` [90]. In the conversion, one can optionally switch to cell corner representations, use provided functionality to add (even gradient-based) additional variables computed from the conserved (or auxiliary) variables, and if needed enforce the code to create a uniform grid interpolation at any level desired, when the post-processing can not handle quadtree/octree grids. Other data formats available for conversion relate to `Idl` [35], `Tecplot` [79] or `OpenDX` [70] visualization possibilities.

The I/O associated with complete data file dumps is worth mentioning separately: the Morton-ordered space filling curve, together with the fixed block size, allows us to perform fully parallel I/O read and write operations. In fact, each processor writes all grids stored in its local memory in the order defined by the Morton number g_{sfc}. All processors write simultaneously to a single file, using `MPI_FILE_IWRITE_AT` constructions that know about the offset corresponding to the block size, and the g_{sfc} index. An `MPI_WAITALL` operation precedes the final writing, by the master CPU, of all extra info required to reconstruct/restart the simulation. This restart is possible on a differing number of CPUs, and may suddenly allow more refinement levels. The extra info consists of (1) the boolean representation of the grid structure as depicted in Fig. 3; and (2) info on grid block size per dimension, equation-specific variable values (like γ in the ideal relativistic MHD module), number of active tree leafs n_{leaf}, maximal refinement level present, dimensionality N_D, number of vector components ($\geqslant N_D$, $\leqslant 3$), number of variables, number of equation-specific variables, integer time counter, and global time t. This represents the most compact, and efficient manner of writing out huge data sets typically executing on massively parallel platforms. As a backup option for those computer platforms where full parallel I/O from all processors is unavailable, we also implemented a fully equivalent, pure master–slave data file dump optionality. The latter option first lets all processors communicate their local data to a master processor, which subsequently writes the data to a single file. This is implemented in two variants, one which uses the `MPI_FILE_WRITE_AT` construct, and one using mere Fortran write statements.

For monitoring the temporal evolution of scalar-valued, global properties, like integral quantities over (part of) the domain, or extremal values of thermodynamic parameters or vector magnitudes, a logfile exists storing their values at user-selected intervals. The global character of these introduces additional I/O communication, typically involving `MPI_REDUCE` operations of type `MPI_SUM`, `MPI_MAX`, `MPI_MIN` after each CPU has collected the info pertaining to local grid blocks. This logfile is written entirely by the master CPU, using standard `MPI_FILE_WRITE` constructs, equivalent to Fortran write statements.

4. Test suite for relativistic plasma modeling

In the remainder of this paper, a selection of representative test problems are described in detail, as far as initial conditions, employed discretization, and AMR strategy is concerned. We cover relativistic hydro problems using constant adiabatic index as well as Mathews EOS, up to relativistic MHD cases with varying effective polytropic index.

4.1. Relativistic hydro problems

4.1.1. Wall heating

A first test considers reproducing the specific challenge of shock heating, where a relativistically flowing gas is forced to convert its kinetic to internal energy completely, by hitting a wall (in the Cartesian case), or by converging on the axis (polar) or origin (spherical). As pointed out in the review by Martí and Müller [53], this has been a recurring test for newly developed solvers, and it can be compared to the analytic solution. A Cartesian version was already presented in [55] at an extreme inflow Lorentz factor, but we will here redo the milder reference case with inflow v_{in} at $\Gamma_{\mathrm{in}} = 10$. Initial condition has $\gamma = 4/3$ and uses the simple EOS from Eq. (4), setting the proper density $\rho = 1$ and pressure $p = 10^{-7}\Gamma_{\mathrm{in}}(\gamma-1)\rho$. We position the wall at left. The test is a 1D problem, with conserved variables D, S_1, τ. Using a Hancock-TVDLF scheme employing minmod limiting, we use a grid covering $x_1 = [0,1]$, have 40 cells on level 1, and allow for four grid levels in total. Refinement uses Löhner, on conserved variables D, S_1 and on auxiliary variable Γ, in a 0.5, 0.25, 0.25 weighting scheme. The tolerance $\epsilon_l = 0.2$. We use in all cases a Courant number $C = 0.4$, without ramp-up, and the right boundary fixes the primitive variables to the exact solution known to be $\rho = (1+\frac{|v_{\mathrm{in}}|t}{x_1})^{\alpha}$, $v_1 = v_{\mathrm{in}}$ and $p = 10^{-7}\Gamma_{\mathrm{in}}(\gamma-1)\rho(x_1,t)$. The index α is zero for Cartesian, 1 for cylindrical, and 2 for spherical inflow, and in the latter two cases these boundaries are spatio-temporally varying. The AMR solutions at $t = 2$ for 1D Cartesian, spherical, and cylindrical (polar) grids, in terms of the D-variable, with the exact solution overplotted, are given in Fig. 9. The 1D non-Cartesian cases add geometrical sources at the same time as the fluxes (unsplit). The shock front is known to be at $t(\gamma-1)\Gamma_{\mathrm{in}}|v_{\mathrm{in}}|/(\Gamma_{\mathrm{in}}+1)$ and is accurately captured, while the polar and spherical case show a small error, mostly pronounced near $x_1 = 0$. No positivity fixing is activated, but non-Cartesian cases reflect (for start-up only) at an artificial left boundary $x_1 = 0.0001$. We also show in Fig. 9 the same test performed on a full 2D polar grid. It serves to show that the simulation recovers the corresponding 1D case, with a grid that behaves as in a 1D case. For this 2D run, we add contributions of all directions simultaneously (dimensionally unsplit treatment).

4.1.2. Richtmyer–Meshkov in planar relativistic HD

A second test considers a 2D planar configuration, targeting the ability to simulate highly dynamic, shock and fluid instability dominated scenarios. We realize an initial condition setting up a shock impinging on an inclined density discontinuity, and choose parameters such that the Richtmyer–Meshkov instability, due to vorticity deposition on the shocked contact, sets

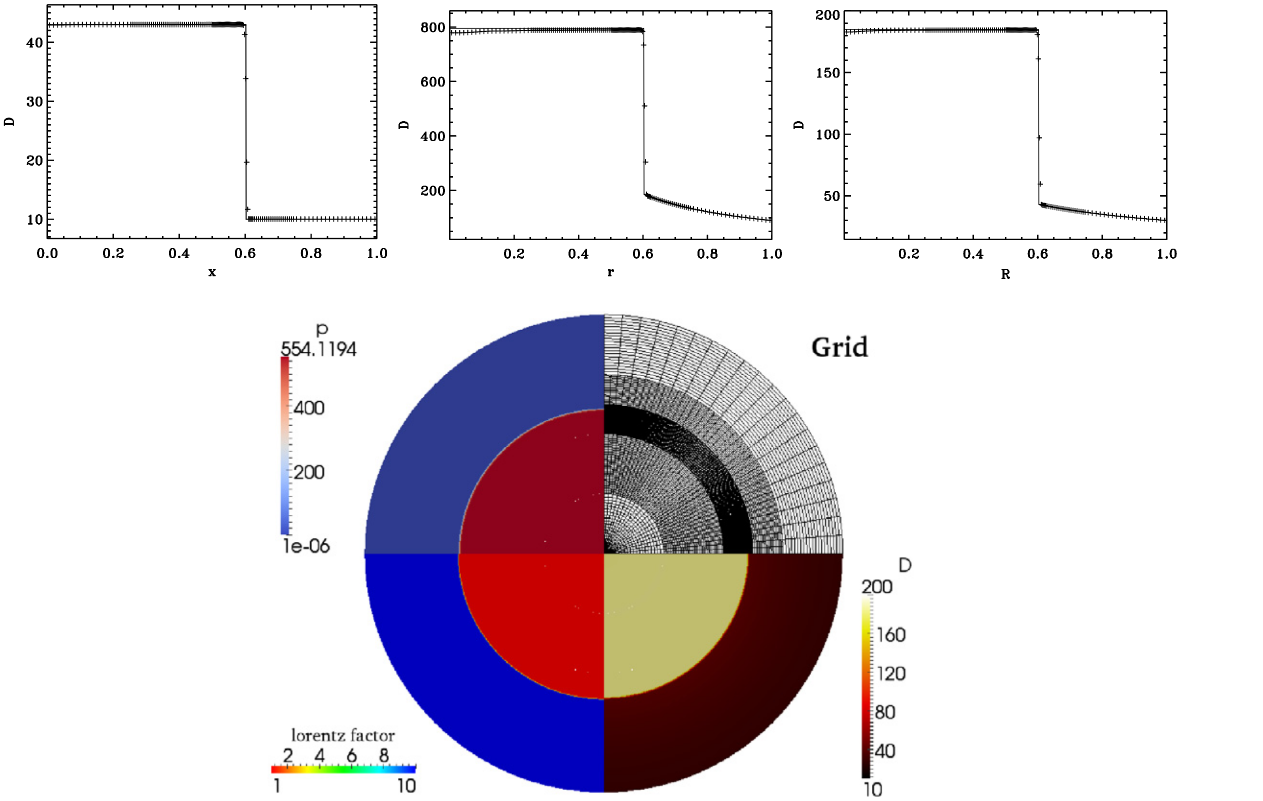

Fig. 9. Top: Cartesian, spherical and cylindrical wall heating, for a Lorentz factor 10 inflow at $t = 2$. The cylindrical case is repeated on a full 2D polar grid, shown as a composite below, illustrating clockwise: the grid, D, Γ and p variations.

in. A non-relativistic hydro variant was studied in [86], and in the Newtonian case, the initial refraction pattern created when the shock passes across the inclined contact can even be predicted using Riemann-solver-based methods [18]. Here, we introduce a test in relativistic hydro settings (with relatively ‘low’ speeds) that demonstrates quite some complexity in the evolving flow patterns.

The simulation is performed on a 2D Cartesian grid $[0, 2.5] \times [0, 0.5]$, using 24×120 grid points at level $l = 1$, allowing seven refinement levels. A Löhner estimator is used, tolerance $\epsilon_l = 0.03$, with refinement triggered on D, S_x and auxiliary variable p, in a 0.7, 0.2, 0.1 ratio. We use Hancock-TVDLF with MC limiting, and can use a Courant number $C = 0.9$ throughout, with a ramp-up of 10 timesteps. We simulate till $t = 6$, and for this long-term, high resolution run we activate positivity fixing. The simulation treats all directions simultaneously when adding fluxes. The initial condition realizes a Mach $M_{\rm sh} = 100$ planar shock, positioned at $x = 0.1$, which propagates into a static medium that contains an inclined contact discontinuity at $x = 0.4$, inclined at an angle $\pi/4$ to the horizontal. Across this contact, a density contrast of 1000 is assumed. A constant $\gamma = 4/3$ is used. In particular, with a parametrization using the postshock velocity $v_{\rm post}$ and the Mach number,

$$\rho = \begin{cases} \rho_1(1+\gamma\Gamma_{\rm post})/(\gamma-1) & x < 0.1, \\ \rho_1 = 1 & 0.1 < x < 0.4 + y/\tan(\pi/4), \\ 1000\rho_1 & x > 0.4 + y/\tan(\pi/4). \end{cases}$$

$$v_x = \begin{cases} v_{\rm post} = 0.5 & x < 0.1, \\ 0 & 0.1 < x. \end{cases}$$

$$p = \begin{cases} \rho_1(\Gamma_{\rm post}-1)(1+\gamma\Gamma_{\rm post}) & x < 0.1, \\ \frac{\frac{\gamma-1}{\gamma}\rho_1 v_{\rm sh}^2}{\left((\gamma-1)M_{\rm sh}^2 - v_{\rm sh}^2\right)} & 0.1 < x. \end{cases}$$

The pressure expression uses the strong shock limit to quantify the shock velocity from $v_{\rm post} v_{\rm sh} = 1 - 1/(\Gamma_{\rm post}(1 + \gamma(\Gamma_{\rm post} - 1)))$. These expressions correspond to the Rankine–Hugoniot relations in the strong shock case. Boundary conditions use reflective wall symmetries at top and bottom, and a continuous extrapolation of all conserved variables left and right. Fig. 10 shows a snapshot at time $t = 6$ of the density, in a Schlieren plot which quantifies an exponentially stretched

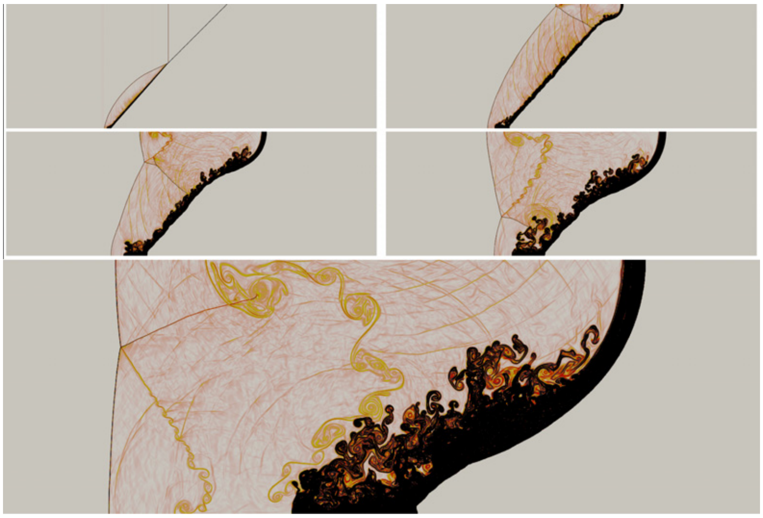

Fig. 10. Relativistic Richtmyer–Meshkov instability and shock refraction. Shown is a Schlieren plot of the density, at times $t = 1, 2, 3, 4$ at top and for $t = 6$ at bottom. See text for details.

gradient evaluation, here taken at $\exp[-0.00025(|\nabla\rho| - 200)]$. A few frames leading up to this endstate are shown as well, showing the initial refraction, the development of the Richtmyer–Meshkov instability, and the intricate shock reflection patterns propagating backwards, in turn developing localized Kelvin–Helmholtz patterns along the various sliplines.

4.1.3. Ultra-relativistic shock dynamics

For special relativistic hydro, the knowledge of the exact solution to the Riemann problem became complete with the work by Pons et al. [73], where the effect of arbitrary tangential flow speeds was incorporated. Those authors presented a set of nine stringent tests for a constant gamma ideal EOS. In [55], it was demonstrated that one can recover their solution, but especially the cases with a high tangential velocity in the left state, where the tail of the leftgoing rarefaction and the contact discontinuity are very close, pose severe numerical challenges in terms of the effective resolution required for agreement. The same is true when higher Lorentz factor regimes are of interest. Lorentz factor $\Gamma = 100$ regimes are known to be important in GRB explosions, which have been simulated in 1D as well as 2D in [55]. Lorentz factor 100 tests, in combination with the Mathews EOS from Eq. (6), were presented in [56]. Correspondence with the exact Riemann problem solution could be demonstrated, when full AMR capabilities were exploited, up to 12 levels with base level grids of 256 cells.

While those latter tests impose Lorentz factor 100 flow from the beginning, it is known that such high Lorentz factor flows can arise naturally from more milder initial conditions, due to the dependence of the Lorentz factor on all velocity components. This ‘boost’ scenario has been topic of renewed interest, either using exact Riemann solvers (for constant gamma EOS) in [2], or reduced 1D MHD simulations in [94]. Here, we give a 1D test that achieves up to Lorentz factor $\Gamma_{\max} \approx 108$ flows with the following initial conditions

$$(\rho, p, v_x, v_y)_L = \left(10^{-4}, 1000, 0.01, 0.998699\ldots\right),$$
$$(\rho, p, v_x, v_y)_R = (1, 1, 0, 0).$$

The left state is set to have Lorentz factor 20 initially, and we use the Mathews equation of state with parameter $\gamma = 5/3$. At time $t = 0.28$, the solution is shown in Fig. 11, where a zoom on the Lorentz factor variation demonstrates the challenge for the solver (the exact variation is overplotted as a solid line there, a slight overshoot occurs at the tail of the rarefaction). The right panel shows the corresponding variations in the logarithm of Γh, ρ, as well as the effective polytropic index γ_{eff} and tangential speed v_y. The simulation uses a 2-stage Runge–Kutta (RK2) with HLLC flux function, using the MC limiter at $\beta_{mc} = 1.2$, Courant parameter $C = 0.9$ and ramp-up with 100 timesteps. Twelve AMR levels are activated on a base grid of 100 cells in $[-0.05, 0.2]$, and Löhner error estimation with $\epsilon_l = 0.1$ is augmented with forced refinement to trigger all levels at the initial discontinuity. No positivity fix is needed, and continuous extrapolation suffices for boundary treatment.

4.1.4. Astrophysical two-component jet models

A final hydrodynamic test combines the various ingredients demonstrated in the previous cases, by performing a simulation where high Lorentz factor flows coexist with milder flow conditions, in a test in cylindrical geometry, using the Mathews EOS. In this test, we revisit the recent findings by Meliani and Keppens [57] which revealed the existence of a purely relativistic instability in so-called two-component jets. We use 2.5D relativistic hydro, using (R, ϕ, Z) ordering of the vector

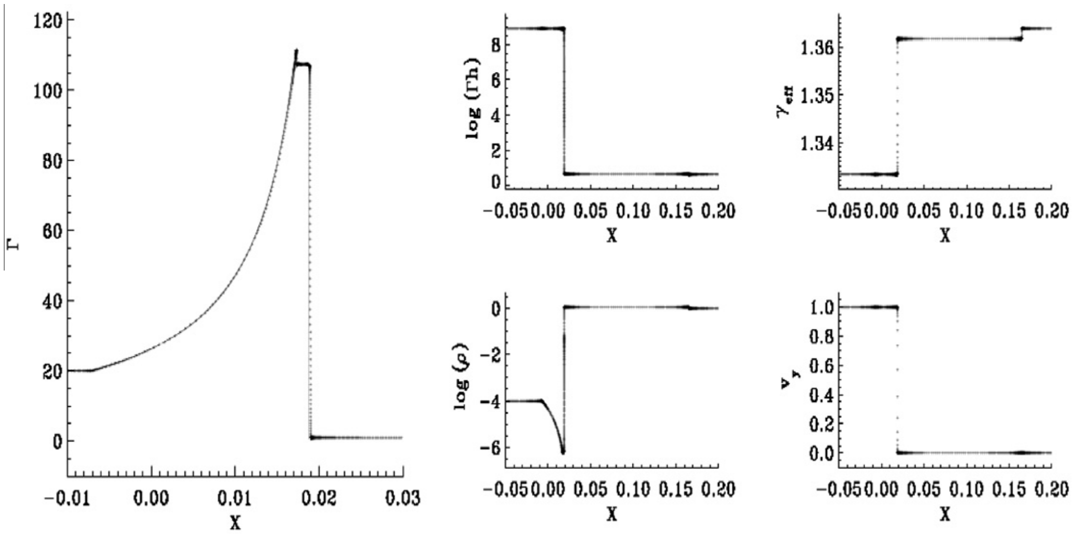

Fig. 11. A Riemann problem demonstrating relativistic boost, showing Lorentz factor at left, and several flow quantities at right. Note the difference in the scale of the horizontal axis.

components to handle the (R, ϕ) plane in polar coordinates, allowing for translational invariance in Z. The initial conditions are

$$\rho = \begin{cases} 0.0692 & R < 0.0333 = R_{in}, \\ 119.94 & R_{in} < R < 0.1, \\ 0.01 & R > 0.1. \end{cases} \qquad v_Z = \begin{cases} 0.99939 & R < R_{in}, \\ 0.9428 & R_{in} < R < 0.1, \\ 0 & R > 0.1. \end{cases}$$

$$v_\phi = \begin{cases} 0.01(R/R_{in})^{\frac{1}{4}} & R < R_{in}, \\ 0.001(R_{in}/R) & R_{in} < R < 0.1, \\ 0 & R > 0.1. \end{cases} \quad p = \begin{cases} p_{in}(R) & R < R_{in}, \\ p_{out}(R) & R_{in} < R < 0.1, \\ p_{out}(R = 0.1) & R > 0.1. \end{cases}$$

The pressure profiles enforce radial equilibrium, using a relation of the form

$$p(R) = \xi_{i,o}\left(1 - \alpha_{i,o}(R/R_{in})^{a_{i,o}}\right)^{-p_{i,o}} - \rho_{i,o}\frac{\gamma_{i,o} - 1}{\gamma_{i,o}}, \tag{39}$$

in which parameters are used that realize an on axis pressure of $p_{in}(0) = 2.3$, and an actual variation of the effective polytropic index that corresponds to $\gamma_i = 4/3$ for the inner and outer medium, and $\gamma_o = 5/3$ for the dense ring in between $R_{in} < R < 0.1$. Precise formulae for even toroidally magnetized cases have been given in [57]. This case recovers the hydro case from that paper, but now on a cylindrical grid, at overall reduced effective resolution. The initial profile for the radial velocity uses a white noise perturbation of maximal amplitude 0.001, localized about the inner radius R_{in}. The evolution is characterized by the development of a relativistic Rayleigh–Taylor type mode at this interface, ultimately causing full mixing between the hot inner jet, which has Lorentz factor 30, and the surrounding cold, outer jet at Lorentz factor 3. In Fig. 12, snapshots of the logarithm of the proper density ρ are shown at selected times, as well as a composite plot at $t = 17.5$. A base resolution of 60×60 with four levels was used, on a circle of radius 0.3. A Courant parameter $C = 0.9$ is used, with 100 timesteps ramp-up. We use HLLC with RK2 and PPM reconstruction, and for the long-term run, the positivity fix is activated. To overcome the CFL limit of the small, central grid cells, we augment the Löhner estimator with coarsening active near the origin only. Boundary conditions use the π-periodicity at the origin, and continuous extrapolation at $R = 0.3$ on all but the normal flow component, where we clip all inflow ($r_{\text{clip}} = 0$). The evolution qualitatively resembles the previous run performed on a Cartesian grid [57], except for the absence of the even smaller-scale Rayleigh–Taylor fingering at the outer $R = 0.1$ (their higher resolution, and the doubled adding of noise at this outer boundary can explain this difference).

4.2. Relativistic MHD problems

For relativistic MHD, modern algorithm developments once more benefit from the knowledge of the exact solution to the full 7-wave Riemann problem, which emerged only recently [29]. That paper contained up to 10 stringent tests, collected from previous literature and augmented with a test targeting specific challenges with Alfvénic discontinuities. All 10 cases were reproduced with the hybrid-block variant of AMRVAC, using Richardson-type error estimation, in [87]. With typically 60

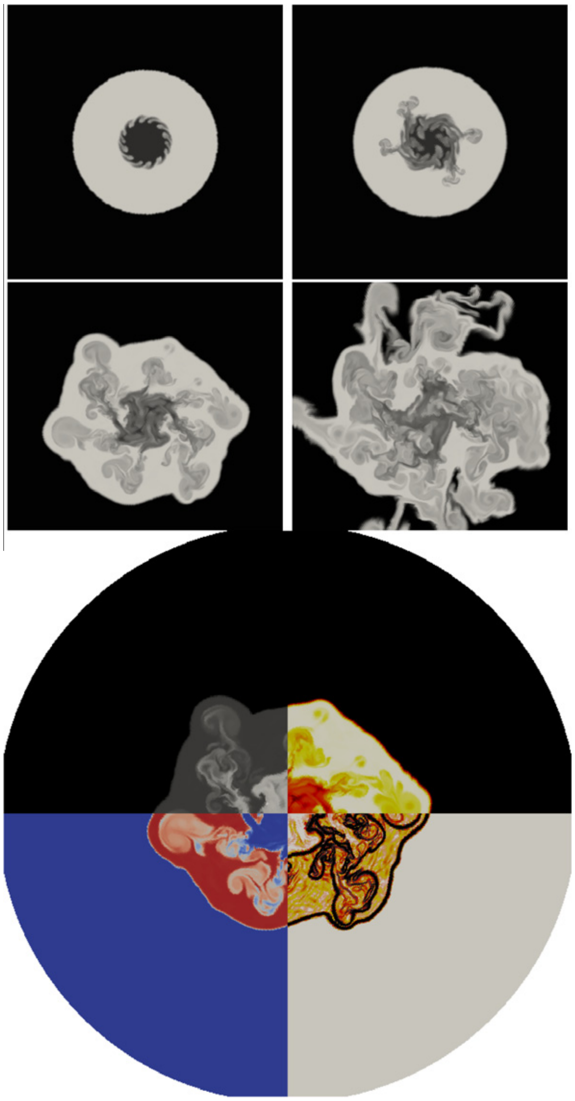

Fig. 12. Top: at times $t = 5, 12.5, 17.5, 22.5$, the logarithm of density in a grayscale plot. The bottom is a composite plot at $t = 17.5$, showing clockwise $\log(\rho)$, a schlieren plot of density, the effective polytropic index and the Lorentz factor.

base level $l = 1$ grid cells, allowing eight levels, a TVDLF scheme could reproduce the exact solutions accurately. In [61], the HLLC solver was tested on four representative shock tube problems as well, and a set of 2D tests highlighted subtle improvements between an HLL and HLLC scheme, using a constrained transport type magnetic field evolution. In this section, we first reproduce a reference relativistic MHD problem, and then provide novel tests using the Mathews EOS in multi-dimensional MHD settings. All tests use dimensionally unsplit treatments in the flux addition, but employ a split strategy for adding the monopole-error related source terms. The latter splitting adds sources using a half time step $\Delta t/2$ prior to the flux additions, to be followed by another $\Delta t/2$ source update.

4.2.1. Relativistic rotor

In literature on algorithmic improvements for (relativistic) MHD solvers, a number of tests emerged as reference cases which novel code efforts then use as benchmark tests. For multidimensional MHD in particular, two recurring problems are the so-called blast wave setups and the rotor test. The blast wave benchmarks start from a uniform magnetic field and impose a localized pressure and/or density change in a circle (2D) or spherical region (3D). Recent 3D variants for fixed grid, classical MHD are found in e.g. [28,65], while an early 3D run with AMR for an implosion, rather than an explosion, was reported in [40]. For relativistic MHD, a similar 2D circular blast wave explosion can be found in early work by Dubal [22] and Komissarov [43], as well as in more recent algorithm comparisons [20,61,21,69,5] for up to 3D variants [62]. In the following section, we report on a novel test with similar simplicity in setup, that shifts the attention to the linear waves generated when a more localized perturbation is imposed. In that latter case, we can compare the result to the analytically known group speed diagram.

The rotor test, for classical MHD found in e.g. [40,62,65], has been revisited in relativistic MHD in [20,5]. In [87], we simulated this test using our hybrid-block based AMRVAC code [86], and we repeat this test here using MPI-AMRVAC following [20]. Due to the initial Lorentz factor 10 conditions in part of the domain, this test is far more stringent than the typical blast wave setup. We use a constant polytropic index EOS with $\gamma = 5/3$, and set up a rest mass density $\rho = 10$ within a disk of radius 0.1 positioned at the center of our domain $[0,1] \times [0,1]$. The whole domain has initially a uniform horizontal magnetic field $\mathbf{B} = \hat{e}_x$ and pressure $p = 1$. The region outside the disk is static, while the disk inner region has a rigid rotation

$$v_x = -9.95(y - 0.5) \quad v_y = 9.95(x - 0.5). \tag{40}$$

We simulate till $t = 0.4$, and we use a four-step, fourth order Runge–Kutta temporal advance, using TVDLF with Koren limiting in the reconstructions. Positivity fixing is allowed, and a rampup of 1000 timesteps to a maximal Courant number of 0.4 allows to capture the initial phase containing a sharp discontinuity at the disk radius. We use a base grid of 60×60, the Löhner estimator on variables D, τ, B_x, B_y in a 6/2/1/1 weight ratio, allowing five refinement levels to reach effective resolution of 960×960. The end result is shown in Fig. 13, which can be compared with Fig. 5 in [20], or Fig. 14 from [5]. We deliberately show a Schlieren plot of density as well, actually plotting $\exp[-0.02525(|\nabla\rho| - 2)]$, which shows much more detail than any of the previously published figures. This makes the many wave front interactions more readily visible. Note that we do not suffer from artificial squaring of the central regions, seen in the low resolution runs from the bottom panels of Fig. 14 from [5]. At the same time, the details in the various wave fronts may still be influenced strongly by the discretizations employed. For the monopole control, the diffusive approach with $C_d = 1$ was exploited. The total amount of grid blocks (each of size 12×12) increases up to 3634, from an initial 304. While initially only 2% of the domain is covered by level five grids, the final state shown here has roughly 50% of the domain on level five grids. It is worth noting that we simulated this test on

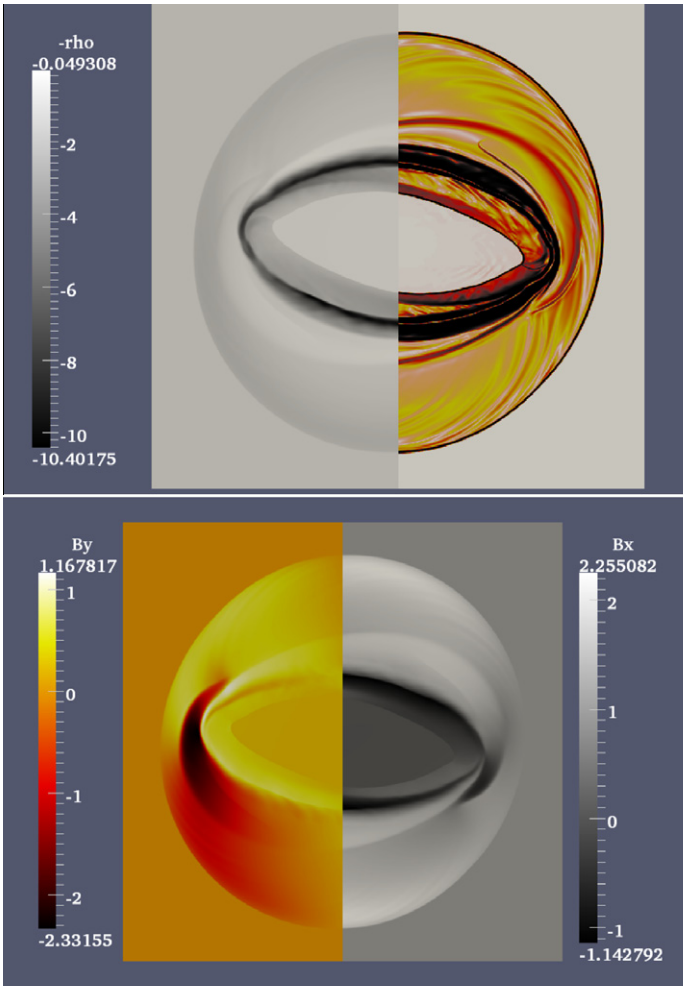

Fig. 13. The relativistic rotor problem. At time $t = 0.4$, we show in the top panel the density structure (both in a linear scale as well as using a Schlieren plot) and below the magnetic field components. The minimal and maximal values are indicated in the color legends.

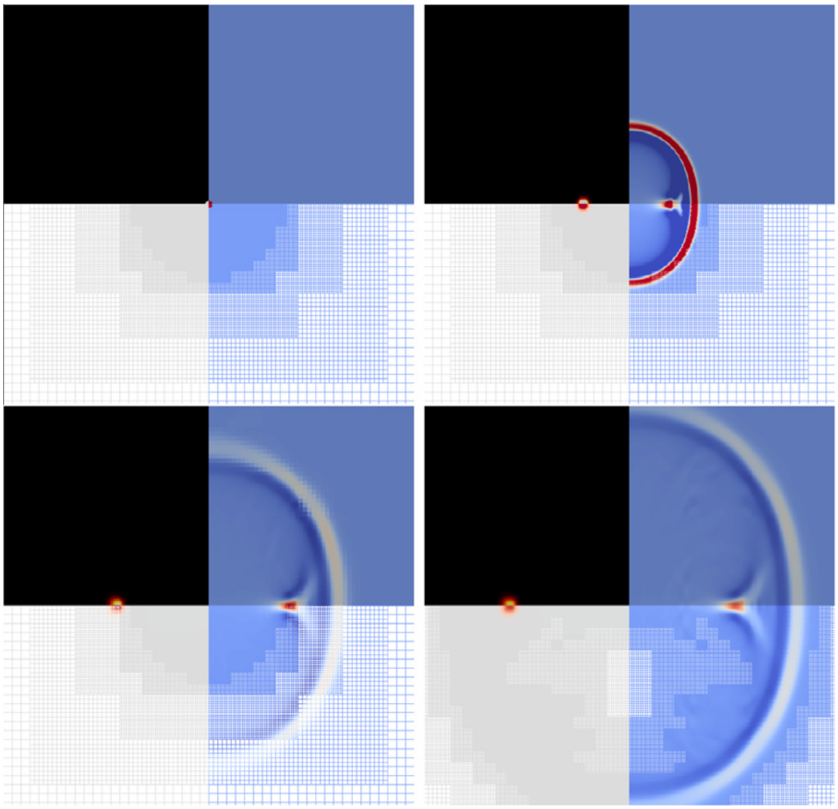

Fig. 14. The group diagram emerging from a localized perturbation. The frames corresponding to $t = 0$, $t = 0.25$, and $t = 0.5$ have deliberately kept the grid fixed, and the fast wavefront has left the finest grid levels. We then turn on AMR facilities, and the frame at $t = 0.65$ shows the expected behavior. In all frames: right sides show pressure variations, left shows B_z (below) and v_z (above).

a quadcore desktop computer (Intel (R) Core (TM)2 Quad CPU at 2.4 GHz, with about 7.8 GB RAM memory) in 4 h, with all I/O included (21 full datafile dumps, and every timestep collecting global logfile info).

4.2.2. Wave diagrams

A known aspect of MHD wave propagation is their anisotropic means to propagate, with increased anisotropic behavior when going from fast, over slow, to Alfvénic disturbances. This is, as in the Newtonian case, best quantified by the knowledge of the phase and group diagrams in relativistic MHD [37,31]. Especially the latter, group diagram representation is a powerful tool for code testing, since it represents the actual outcome of an evolution starting as a pointlike perturbation. It complements the traditional 1D Riemann problem tests in (1) providing an exact solution for multi-D scenarios; and (2) brings in aspects of linear waves. When performed in non-static initial conditions, it also tests wave aberration effects.

Fig. 14 shows four snapshots of the evolution from a simple initial condition where $\rho = 1,\ p = 1,\ \mathbf{B} = 0.3\hat{e}_x$ in a static environment, with a perturbation of $\delta p = 0.1$ and $\delta v_z = 0.1$ imposed at $t = 0$ on the region $\sqrt{x^2 + y^2} < 0.005$. Note that we assume translational invariance in z. Shown in all snapshots is in clockwise fashion: the pressure in colourscale plot, idem below but in a wireframe view showing the grid, a wireframe view of the out-of-plane magnetic field component B_z, and a colourscale view of the out-of-plane velocity v_z. The latter two demonstrate the pointwise Alfvénic disturbances, while the pressure fronts show both the fast wave front, and the cusplike wavefront from the (constructive interference of) slow waves.

The domain covered is $[-1, 1] \times [-1, 1]$ and we use 160×160 grid points, using five levels in total. Note that the initial condition is only resolved from level 3 onwards, so we actually have to enforce grid refinement manually at first. In fact, using the Löhner estimator under standard parameter settings, one would typically not capture linear waves, since most of the estimators perform best on strong nonlinear jumps. Therefore, we fix the grid, triggered about the origin, up to time $t = 0.5$. Three instances of the evolution are shown in Fig. 14 corresponding to $t = 0$, $t = 0.25$, and this $t = 0.5$ time. At that point, the fast wave has already traversed a number of grid levels, without distortion. Then, we turn on automated grid refinement, but with a wavefilter value set to $f_{\text{wave,l}} = 0.0001$. This is more than sufficient to let the AMR realize it has to refine on all wavefronts, thereby prolonging the wave features that have left the finest level grid. At the final time $t = 0.65$ shown, one can note the accurate reproduction of all wave features.

This test used a RK2 with TVDLF flux function combination with minmod limiting, used the diffusion approach of Eq. (27) with $C_d = 1$, and for the quartic formulation for the wavespeeds worked with the $1/(1 - \lambda)$ implementation. A Courant

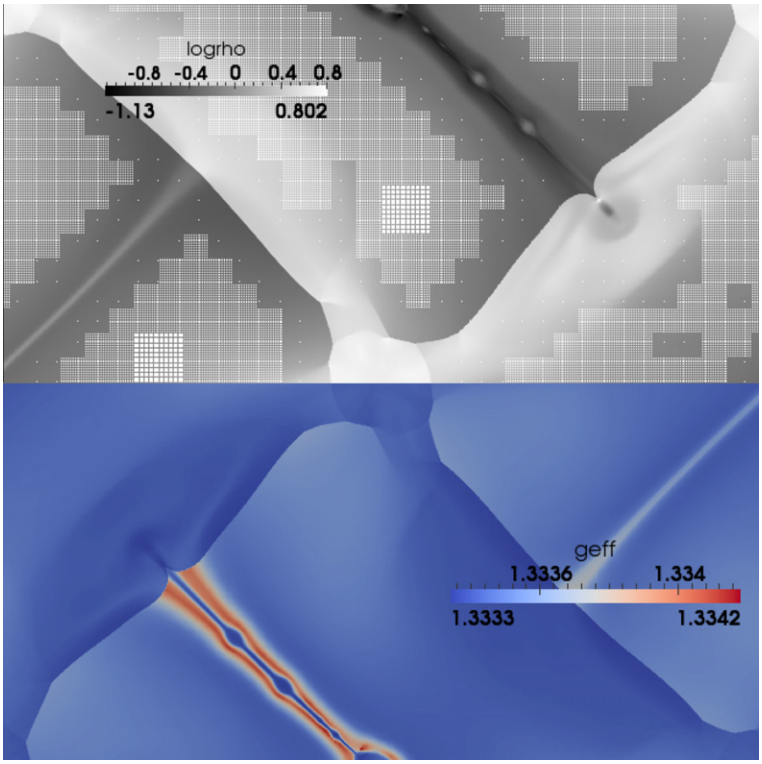

Fig. 15. A snapshot of a shock-dominated magnetized plasma evolution, where ultimately a current sheet fragments due to tearing events. Top is the logarithm of proper density ρ, bottom shows effective polytropic index γ_{eff}, at time $t = 5.6$. The combined picture shows the full square domain.

parameter of 0.4 with 100 timesteps ramp-up was employed. It can be worthwhile to use this type of test, under a fair number of governing parameter values, to further assess the different schemes in modern MHD codes.

4.2.3. Relativistic Orszag–Tang problem

A final test reported here considers a relativistic variant of the classical Orszag–Tang problem, as also presented for relativistic MHD in [87]. While that paper assumed a polytropic EOS, we here revisit the problem using the Mathews EOS. Initial conditions on a 2D periodic domain $[0, 2\pi] \times [0, 2\pi]$ realize a vortical flow at Lorentz factors up to 7, and a double magnetic island structure, setting

$$v_x = -\frac{0.99}{\sqrt{2}} \sin(y) \quad B_x = -\sin(y),$$
$$v_y = +\frac{0.99}{\sqrt{2}} \sin(x) \quad B_y = +\sin(2x),$$

in otherwise uniform $p = 10$, $\rho = 1$ plasma. The parameter $\gamma = 5/3$, but these conditions then represent hot plasma (effective polytropic index close to 4/3). We use 40 × 40 cells on level 1, allowing seven levels. Löhner estimator on the D variable alone is sufficient, with $\epsilon_l = 0.25$. A Courant parameter $C = 0.9$ with 20 ramp-up timesteps, together with TVDLF and minmod limiting is used, and the same diffusion strategy for solenoidal field control as in the previous test. The evolution was described in detail in [87], a representative figure at time $t = 5.6$ is Fig. 15. It shows the proper density in a logarithmic fashion on top, and the effective polytropic index variation below. At this time, the shock-compressed current sheet which intersects bottom (and hence, top) boundary has become subject to tearing instability, developing island structures. It must be stressed that these reconnection events are due to numerical diffusion, as we simulate ideal MHD. Revisiting these test conditions in resistive relativistic MHD, with solvers as presented recently in [44], would be able to quantify reconnection rates. Still, it seems reasonable to act as in classical MHD, where the practice of using ideal MHD, shock-capturing solvers for also reconnecting field evolutions has given much insight in global MHD scenarios.

5. Outlook and future developments

This paper focused on modern code aspects in special relativistic plasma modeling, where shock-capturing schemes have become standard, while AMR facilities are finding their way to most codes in active use and development. While the discussion used the MPI-AMRVAC code for providing implementation details, it is clear that many contemporary efforts share ideas on AMR criteria, methodology, and parallelization aspects. We gave all details for the latter, which forms an important aspect to allow ever more sophisticated simulations in astrophysics.

A specific example concerns the study of relativistic jet dynamics, where jets originally studied in 2D hydro settings [52] can now be done with magnetic fields included, in either constrained topologies [47,38] or in fully helical field situations [39], up to 3D [64,68]. This is an area of active research, where simulations in full general relativistic setting can give insight in jet launch conditions and accretion disk physics near black holes, while special relativistic codes do more detailed analysis of the many MHD instabilities to which the jets are prone. Also, the realm of ultra-relativistic flows pertinent to GRB physics is likely to gain significant further insight from targeted numerical studies, where the fluid/plasma behavior at very high Lorentz factor flows is modeled. Given the enormous progress in the field of relativistic computations over the last two decades, we can expect continued active research in decades ahead.

Acknowledgments

We acknowledge access to the K.U.Leuven High Performance Computing cluster VIC. These results were obtained in the framework of the Projects GOA/2009/009 (K.U.Leuven) and G.0277.08 (FWO-Vlaanderen). Z. Meliani acknowledges financial support from HPC Europa, Project 228398. We thank Bruno Giacomazzo for making his Riemann solver available, and Gábor Tóth for pointing out a generalization of the MC limiting. Visualization used Paraview, see www.paraview.org.

References

[1] M.A. Aloy, J.M. Ibáñez, J.M. Martí, E. Müller, GENESIS: a high-resolution code for three-dimensional relativistic hydrodynamics, ApJS 122 (1999) 151.
[2] M.A. Aloy, L. Rezzolla, A powerful hydrodynamic booster for relativistic jets, Astrophys. J. 640 (2006) L115.
[3] M. Anderson, E.W. Hirschmann, S.L. Liebling, D. Neilsen, Relativistic MHD with adaptive mesh refinement, CQGra 23 (2006) 6503.
[4] P. Anninos, P.C. Fragile, J.D. Salmonson, Cosmos++: relativistic magnetohydrodynamics on unstructured grids with local adaptive refinement, ApJ 635 (2005) 723.
[5] L. Antón, J.A. Miralles, J.M. Martí, J.M. Ibáñez, M.A. Aloy, P. Mimica, Relativistic magnetohydrodynamics: renormalized eigenvectors and full wave decomposition Riemann solver, ApJS 188 (2010) 1.
[6] D.S. Balsara, Total variation diminishing scheme for relativistic magnetohydrodynamics, Astrophys. J. Suppl. Ser. 132 (2001) 83.
[7] D.S. Balsara, J. Kim, A comparison between divergence-cleaning and staggered-mesh formulations for numerical magnetohydrodynamics, Astrophys. J. 602 (2004) 1079.
[8] M.J. Berger, Data structures for adaptive grid generation, SIAM J. Sci. Stat. Comput. 7 (1986) 904.
[9] M.J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1989) 64.
[10] J. Bergmans, R. Keppens, D.E.A. van Odyck, A. Achterberg, Simulations of relativistic astrophysical flows, in: T. Plewa, T. Linde, V.G. Weirs (Eds.), Adaptive Mesh Refinement – Theory and Applications, Lecture Notes in Computational Science and Engineering, vol. 41, 2005, p. 223.
[11] Cactus toolkit at <http://cactuscode.org>.
[12] M. Cada, M. Torrilhon, Compact third-order limiter functions for finite volume methods, J. Comput. Phys. 228 (2009) 4118.
[13] A.C. Calder, B. Fryxell, T. Plewa, R. Rosner, L.J. Dursi, V.G. Weirs, T. Dupont, H.F. Robey, J.O. Kane, B.A. Remington, R.P. Drake, G. Dimonte, M. Zingale, F.X. Timmes, K. Olson, P. Ricker, P. MacNeice, H.M. Tufo, On validating an astrophysical simulation code, Astrophys. J. Suppl. Ser. 143 (2002) 201.
[14] Chombo library at <http://seesar.lbl.gov/ANAG/chombo/>.
[15] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas dynamical simulations, J. Comput. Phys. 54 (1984) 174.
[16] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, Hyperbolic divergence cleaning for the MHD equations, J. Comput. Phys. 175 (2002) 645.
[17] P.J. Dellar, A note on magnetic monopoles and the one-dimensional MHD Riemann problems, J. Comput. Phys. 172 (2001) 392.
[18] P. Delmont, R. Keppens, B. van der Holst, An exact Riemann-solver-based solution for regular shock refraction, J. Fluid Mech. 627 (2009) 33.
[19] L. Del Zanna, N. Bucciantini, An efficient shock-capturing central-type scheme for multidimensional relativistic flows. I. Hydrodynamics, Astron. Astrophys. 390 (2002) 1177.
[20] L. Del Zanna, N. Bucciantini, P. Londrillo, An efficient shock-capturing central-type scheme for multidimensional relativistic flows. II. Magnetohydrodynamics, Astron. Astrophys. 400 (2003) 397.
[21] L. Del Zanna, O. Zanotti, N. Bucciantini, P. Londrillo, ECHO: a Eulerian conservative high-order scheme for general relativistic magnetohydrodynamics and magnetodynamics, Astron. Astrophys. 473 (2007) 11.
[22] M.R. Dubal, Numerical simulations of special relativistic, magnetic gas flows, Comput. Phys. Commun. 64 (1991) 221.
[23] C.R. Evans, J.F. Hawley, Simulation of magnetohydrodynamic flows – a constrained transport method, Astrophys. J. 332 (1988) 659.
[24] FLASH3 user guide at <http://flash.uchicago.edu>.
[25] J.A. Font, J.M. Ibáñez, A. Marquina, J.M. Martí, Multidimensional relativistic hydrodynamics: characteristic fields and modern high-resolution shock-capturing schemes, Astron. Astrophys. 282 (1994) 304.
[26] J.A. Font, Numerical hydrodynamics and magnetohydrodynamics in general relativity, Living Rev. Relat. 11 (2008) 7.
[27] C.F. Gammie, J.C. McKinney, G. Tóth, HARM: a numerical scheme for general relativistic magnetohydrodynamics, Astrophys. J. 589 (2003) 444.
[28] T.A. Gardiner, J. Stone, An unsplit Godunov method for ideal MHD via constrained transport in three dimensions, J. Comput. Phys. 227 (2008) 4123.
[29] B. Giacomazzo, L. Rezzolla, The exact solution of the Riemann problem in relativistic MHD, J. Fluid Mech. 562 (2006) 223.
[30] B. Giacomazzo, L. Rezzolla, WhiskyMHD: a new numerical code for general relativistic magnetohydrodynamics, CQGra 24 (2007) S235.
[31] J.P. Goedbloed, R. Keppens, S. Poedts, Advanced MHD. With Application to Laboratory and Astrophysical Plasmas, Cambridge University Press, Cambridge, 2010.
[32] A. Harten, P.D. Lax, B. van Leer, On upstream differencing and Godunov-type schemes for hyperbolic conservation laws, SIAM Rev. 25 (1983) 35.
[33] V. Honkkila, P. Janhunen, HLLC solver for ideal relativistic MHD, J. Comput. Phys. 223 (2007) 643.
[34] P.A. Hughes, M.A. Miller, G.C. Duncan, Three-dimensional hydrodynamic simulations of relativistic extragalactic jets, Astrophys. J. 572 (2002) 713.
[35] Idl is a product of ITT Visual Information Solutions <http://www.ittvis.com>.
[36] P. Janhunen, A positive conservative method for magnetohydrodynamics based on HLL and Roe methods', J. Comput. Phys. 160 (2000) 649.
[37] R. Keppens, Z. Meliani, Linear wave propagation in relativistic magnetohydrodynamics, Phys. Plasmas 15 (2008) 102103.
[38] R. Keppens, Z. Meliani, Grid-adaptive simulations of relativistic flows, in: H. Deconinck, E. Dick (Eds.), Computational Fluid Dynamics 2006, vol. 335, Springer, Berlin, 2009.
[39] R. Keppens, Z. Meliani, B. van der Holst, F. Casse, Extragalactic jets with helical magnetic fields: relativistic MHD simulations, Astron. Astrophys. 486 (2008) 663.

[40] R. Keppens, M. Nool, G. Tóth, J.P. Goedbloed, Adaptive mesh refinement for conservative systems: multi-dimensional efficiency evaluation, Comput. Phys. Commun. 153 (2003) 317.
[41] R. Keppens, G. Tóth, OpenMP parallelism for multi-dimensional grid-adaptive magnetohydrodynamic simulations, Lect. Notes Comput. Sci. 2329 (2002) 940.
[42] A.V. Koldoba, O.A. Kuznetsov, G.V. Ustyugova, An approximate Riemann solver for relativistic magnetohydrodynamics, MNRAS 333 (2002) 932.
[43] S.S. Komissarov, A Godunov-type scheme for relativistic magnetohydrodynamics, MNRAS 303 (1999) 343.
[44] S.S. Komissarov, Multi-dimensional numerical scheme for resistive relativistic MHD, MNRAS 382 (2007) 995.
[45] B. Koren, A robust upwind discretization method for advection, diffusion and source terms, in: C.B. Vreugdenhil, B. Koren (Eds.), Numerical Methods for Advection–Diffusion Problems, Notes on Numerical Fluid Mechanics, vol. 45, Vieweg, Braunschweig, 1993, p. 117.
[46] A. Kurganov, E. Tadmor, New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations, J. Comput. Phys. 160 (2000) 241.
[47] T. Leismann, L. Antón, M.A. Aloy, E. Müller, J.M. Martí, J.A. Miralles, J.M. Ibáñez, Relativistic MHD simulations of extragalactic jets, Astron. Astrophys. 436 (2005) 503.
[48] R. Löhner, An adaptive finite element scheme for transient problems in CFD, Comput. Methods Appl. Mech. Eng. 61 (1987) 323.
[49] P. MacNeice, K.M. Olson, C. Mobarry, R. deFainchtein, C. Packer, PARAMESH: a parallel adaptive mesh refinement community toolkit, Comput. Phys. Commun. 126 (2000) 330.
[50] W.G. Mathews, The hydromagnetic free expansion of a relativistic gas, Astrophys. J. 165 (1971) 147.
[51] B. Marder, A method for incorporating Gauss' law into electromagnetic PIC codes, J. Comput. Phys. 68 (1987) 48.
[52] J.M. Martí, E. Müller, J.A. Font, J.M. Ibáñez, A. Marquina, Morphology and dynamics of relativistic jets, Astrophys. J. 479 (1997) 151.
[53] J.M. Martí, E. Müller, Numerical hydrodynamics in special relativity, Living Rev. Relat. 6 (2003) 7.
[54] Z. Meliani, C. Sauty, K. Tsinganos, N. Vlahakis, Relativistic Parker winds with variable effective polytropic index, Astron. Astrophys. 425 (2004) 773.
[55] Z. Meliani, R. Keppens, F. Casse, D. Giannios, AMRVAC and relativistic hydrodynamic simulations for gamma-ray burst afterglow phases, MNRAS 376 (2007) 1189.
[56] Z. Meliani, R. Keppens, B. Giacomazzo, Fanaroff-Riley type I jet deceleration at density discontinuities, Astron. Astrophys. 491 (2008) 321.
[57] Z. Meliani, R. Keppens, Decelerating relativistic two-component jets, Astrophys. J. 705 (2009) 1594.
[58] Z. Meliani, R. Keppens, Dynamics and stability of relativistic GRB blast waves, Astron. Astrophys. 520 (2010) L3.
[59] A. Mignone, T. Plewa, G. Bodo, The piecewise parabolic method for multidimensional relativistic fluid dynamics, Astrophys. J. Suppl. Ser. 160 (2005) 199.
[60] A. Mignone, G. Bodo, An HLLC Riemann solver for relativistic flows – I. Hydrodynamics, MNRAS 364 (2005) 126.
[61] A. Mignone, G. Bodo, An HLLC Riemann solver for relativistic flows – II. Magnetohydrodynamics, MNRAS 368 (2006) 1040.
[62] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni, A. Ferrari, PLUTO: a numerical code for computational astrophysics, Astrophys. J. Suppl. Ser. 170 (2007) 228.
[63] A. Mignone, J.C. McKinney, Equation of state in relativistic magnetohydrodynamics: variable versus constant adiabatic index, MNRAS 378 (2007) 1118.
[64] A. Mignone, P. Rossi, G. Bodo, A. Ferrari, S. Massaglia, High resolution 3D relativistic MHD simulations of jets, MNRAS 402 (2010) 7.
[65] A. Mignone, P. Tzeferacos, A second-order unsplit Godunov scheme for cell-centered MHD: the CTU–GLM scheme, J. Comput. Phys. 229 (2010) 2117.
[66] A. Mignone, M. Ugliano, G. Bodo, A five-wave Harten–Lax–van Leer Riemann solver for relativistic magnetohydrodynamics, MNRAS 393 (2009) 1141.
[67] G.H. Miller, P. Colella, A conservative three-dimensional Eulerian method for coupled solid–fluid shock capturing, J. Comput. Phys. 183 (2002) 26.
[68] Y. Mizuno, P. Hardee, K.I. Nishikawa, Three-dimensional relativistic magnetohydrodynamic simulations of magnetized spine-sheath relativistic jets, Astrophys. J. 662 (2007) 835.
[69] S.C. Noble, C.F. Gammie, J.C. McKinney, L. del Zanna, Primitive variable solvers for conservative general relativistic magnetohydrodynamics, Astrophys. J. 641 (2006) 626.
[70] OpenDX is an open source tool available at <http://www.opendx.org>.
[71] Paraview is an open source tool available at <http://www.paraview.org>.
[72] M. Perucho, J.M. Martí, A numerical simulation of the evolution and fate of a Fanaroff-Riley type I jet, MNRAS 382 (2007) 526.
[73] J.A. Pons, J.M. Martí, E. Müller, The exact solution of the Riemann problem with non-zero tangential velocities in relativistic hydrodynamics, J. Fluid Mech. 422 (2000) 125.
[74] K.G. Powell, An Approximate Riemann Solver for Magnetohydrodynamics (That Works in More than One Dimension), ICASE Report No. 94-24, Langley VA, 1994.
[75] P. Rossi, A. Mignone, G. Bodo, S. Massaglia, A. Ferrari, Formation of dynamical structures in relativistic jets: the FRI case, Astron. Astrophys. 488 (2008) 795.
[76] V.V. Rusanov, The calculation of the interaction of non-stationary shock waves and obstacles, USSR Comput. Math. Math. Phys. 1 (1961) 304.
[77] D. Ryu, I. Chattopadhyay, E. Choi, Equation of state in numerical relativistic hydrodynamics, Astrophys. J. Suppl. Ser. 166 (2006) 410.
[78] I.V. Sokolov, H.-M. Zhang, J.I. Sakai, Simple and efficient Godunov scheme for computational relativistic gas dynamics, J. Comput. Phys. 172 (2001) 209.
[79] Tecplot is a product of Tecplot, Inc., see <http://www.tecplot.com>.
[80] J.L. Synge, The Relativistic Gas, North Holland, Amsterdam, 1957.
[81] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, Springer-Verlag, Berlin, 1997.
[82] G. Tóth, The LASY preprocessor and its application to general multi-dimensional codes, J. Comput. Phys. 138 (1997) 981.
[83] G. Tóth, A general code for modeling MHD flows on parallel computers: versatile advection code, Astrophys. Lett. Commun. 34 (1996) 245. See <http://www.phys.uu.nl/~toth>.
[84] G. Tóth, D. Odstrčil, Comparison of some flux corrected transport and total variation diminishing numerical schemes for hydrodynamic and magnetohydrodynamic problems, J. Comput. Phys. 128 (1996) 82.
[85] G. Tóth, The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes, J. Comput. Phys. 161 (2000) 605.
[86] B. van der Holst, R. Keppens, Hybrid block-AMR in cartesian and curvilinear coordinates: MHD applications, J. Comput. Phys. 226 (2007) 925.
[87] B. van der Holst, R. Keppens, Z. Meliani, A multidimensional grid-adaptive relativistic magnetofluid code, Comput. Phys. Commun. 179 (2008) 617.
[88] A.J. van Marle, R. Keppens, Radiative cooling in numerical astrophysics: the need for adaptive mesh refinement, Comput. Fluids, in press, doi:10.1016/j.compfluid.2010.10.022.
[89] B. van Leer, Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101.
[90] Visit is available at <https://wci.llnl.gov/codes/visit/home.html>.
[91] P. Wang, T. Abel, W. Zhang, Relativistic hydrodynamic flows using spatial and temporal adaptive structured mesh refinement, Astrophys. J. Suppl. Ser. 176 (2008) 467.
[92] T. Wen, J. Su, P. Colella, K. Yelick, N. Koen, An adaptive mesh refinement benchmark for modern parallel programming languages, in: SC'07 Proceedings of the 2007 ACM/IEEE Conference on Supercomputing, doi:10.1145/1362622.1362676.
[93] H.C. Yee, A class of high-resolution explicit and implicit shock-capturing methods, NASA TM-101088, 1989.
[94] S. Zenitani, M. Hesse, A. Klimas, Scaling of the anomalous boost in relativistic jet boundary layer, Astrophys. J. 712 (2010) 951.
[95] W. Zhang, A.I. MacFadyen, RAM: a relativistic adaptive mesh refinement hydrodynamics code, Astrophys. J. Suppl. Ser. 164 (2006) 255.