

Special relativistic hydrodynamics with CRONOS

D. Huber and R. Kissmann

Institut für Astro- und Teilchenphysik, University of Innsbruck, 6020 Innsbruck, Austria e-mail: contact@david-huber.eu, ralf.kissmann@uibk.ac.at

Received 21 May 2021 / Accepted 21 July 2021

ABSTRACT

We describe the special relativistic extension of the CRONOS code, which has been used for studies of gamma-ray binaries in recent years. The code was designed to be easily adaptable, allowing the user to easily change existing functionalities or introduce new modules tailored to the problem at hand. Numerically, the equations are treated using a finite-volume Godunov scheme on rectangular grids, which currently support Cartesian, spherical, and cylindrical coordinates. The employed reconstruction technique, the approximate Riemann solver, and the equation of state can be chosen dynamically by the user. Further, the code was designed with stability and robustness in mind, detecting and mitigating possible failures early on. We demonstrate the code's capabilities on an extensive set of validation problems.

Key words. hydrodynamics – relativistic processes – methods: numerical

1. Introduction

Relativistic hydrodynamics is one of the most successful tools to describe complex dynamical phenomena ranging from cosmological scales to those of colliding particles. General relativity is thereby a necessary ingredient to understand systems with strong gravitational fields, such as accretion discs and the launch of jets in active galactic nuclei (e.g. Mościbrodzka et al. 2014, 2016; Komissarov & Porth 2021) or the mergers of compact objects producing gravitational waves (e.g. Siegel & Metzger 2018). Many other phenomena, however, do not involve strongly bent spacetime but still flow velocities near the speed of light, for example gamma-ray bursts (e.g. Willingale & Mészáros 2017; Zhang 2018), systems with pulsar winds (e.g. Barkov et al. 2019), such as gamma-ray binaries (e.g. Huber et al. 2021a,b) and pulsar wind nebulae (e.g. Olmi & Bucciantini 2019), or interacting heavy-ion beams in collider experiments (e.g. Busza et al. 2018). Their dynamics can therefore be more suitably described in special relativistic frameworks, which we focus on in this work.

Due to the high complexity of many problems in current research, analytical approaches are no longer feasible to explain observed data. Numerical approaches (see Martí & Müller 2003, 2015, for a review) are therefore essential to further our understanding of the related systems. For this, many modern codes have been presented that treat the equations of special relativistic hydrodynamics (SRHD) and/or magnetohydrodynamics (SRMHD), such as PLUTO (Mignone et al. 2007, 2012), RAMSES (Teyssier 2002; Lamberts et al. 2013), ATHENA++ (Stone et al. 2008; White et al. 2016), AMRVAC (Meliani et al. 2007), and others.

In this paper, we present the special relativistic extension of CRONOS (Kissmann et al. 2018), which was used in comprehensive models for the emission of gamma-ray binaries (see Huber et al. 2021a,b). CRONOS is a finite-volume, Godunov-type code that was developed to solve hyperbolic systems of partial differential equations with astrophysical applications in mind. A key feature of CRONOS consists in its easy adaptability and

extensibility, providing a framework with several interfaces for customisations by the user that allows for the straightforward incorporation of additional physics, such as the transport of energetic particles (see e.g. Reitberger et al. 2017), radiative cooling of the plasma (see e.g. Scherer et al. 2020), or the use of more general reference frames (see e.g. Huber et al. 2021a). Here we describe the extension of CRONOS to SRHD, where we did not introduce any new algorithms, but selected and combined different techniques that have been proven successful in literature.

The paper is structured as follows: in Sect. 2 we introduce the governing equations; the notations and general structure of the code are defined in Sect. 3; in Sect. 4 we describe the numerical scheme implemented in CRONOS; and we validate it on an extensive set of numerical tests in Sect. 5. Finally, we comment on the parallel performance of the code in Sect. 6 and give a summary in Sect. 7.

2. System of equations

The CRONOS code (Kissmann et al. 2018) was in general developed to solve a set of hyperbolic conservation laws of the form

$$\partial_t U + \nabla \cdot \boldsymbol{F}(U) = S,\tag{1}$$

where U denotes a set of conserved quantities, F the respective fluxes, and S an additional source term.

2.1. Special relativistic hydrodynamics

In analogy to classical hydrodynamics, the dynamics of a relativistic fluid can be expressed as such a system (Landau & Lifshitz 1959). The respective equations for the conservation of rest-mass, energy, and momentum read as follows:

$$\partial_t D + \nabla \cdot (D \mathbf{v}) = S_D, \tag{2a}$$

$$\partial_t \tau + \nabla \cdot ((\tau + p) \mathbf{v}) = S_{\tau}, \tag{2b}$$

$$\partial_t \mathbf{m} + \nabla \cdot (\mathbf{m} \otimes \mathbf{v} + p \, \mathbb{1}) = \mathbf{S}_{\mathrm{m}},\tag{2c}$$

where $U=(D,\tau,m^j)$ correspond to the conserved variables, that are the mass, energy, and momentum density in the *j*th coordinate direction, respectively, and $S=(S_D,S_\tau,S_{m^j})$ correspond to their respective source terms. The fluxes further involve the fluid's three-velocity \boldsymbol{v} and pressure p. All quantities are given in the laboratory frame and in natural units, that is with the speed of light c=1.

It should be noted that the momentum fluxes in Eq. (2c) are given by the rank-2 stress tensor. Additional geometrical source terms might therefore arise implicitly when evaluating the divergence in non-Cartesian coordinates (see Sect. 3.1).

Other codes treat the conservation of energy by solving an equivalent equation for $E = \tau + D$ (see e.g. Mignone et al. 2007; Lamberts et al. 2013). In the low pressure, non-relativistic regime, the variables D and E, however, can take very similar values, which can lead to severe truncation errors. This can be avoided by evolving the variable $\tau = E - D$, instead, which we therefore also used in CRONOS. This choice was solely motivated by numerical reasons and has therefore been used in many codes (see e.g. Banyuls et al. 1997; Aloy et al. 1999; Baiotti et al. 2005; Meliani et al. 2007; Rezzolla 2013).

The conserved variables and the corresponding fluxes are further expressed in terms of the primitive fluid variables $w = (\rho, p, u^i)$, which are the fluid's mass density ρ , pressure p, and the spatial part of the four-velocity vector u^i , respectively. The relations read as follows:

$$U(w) = (D, \tau, m^{j})$$

= $(\rho u^{0}, \rho u^{0}(hu^{0} - 1) - p, \rho hu^{0}u^{j}),$ (3)

where the fluid's bulk Lorentz factor is denoted by u^0 and its specific enthalpy by h.

In contrast to other implementations that employ the three-velocity vector as a primitive quantity (e.g. Mignone et al. 2007; Lamberts et al. 2013), we used the four-velocity vector instead and inferred the former via $\mathbf{v} = \mathbf{u}/u^0$. This choice is more natural for numerical purposes since $|\mathbf{u}|$ is not bound by the speed of light – a constraint that one would have to guarantee for $|\mathbf{v}|$ at all times. The computation of the Lorentz factor via $(u^0)^2 = 1 + |\mathbf{u}|^2$ is also more robust by naturally avoiding truncation errors as compared to $(u^0)^2 = (1 - |\mathbf{v}|^2)^{-1}$.

The presented system of five equations in six unknowns is closed by specifying an additional equation of state (EoS), which is usually provided in the form of $h=h(\Theta)$ with $\Theta=\frac{p}{\rho}$. In CRONOS we considered two EoSs: the ideal one, assuming a constant adiabatic index Γ and the Taub-Mathews (TM) EoS (Mathews 1971), which approaches $\Gamma \to \frac{4}{3}$ in the high-temperature limit and $\Gamma \to \frac{5}{3}$ in the low-temperature limit. The specific enthalpies are given by

$$h(\Theta) = \begin{cases} 1 + \frac{\Gamma}{\Gamma - 1}\Theta & \text{ideal EoS} \\ \frac{5}{2}\Theta + \sqrt{\frac{9}{4}\Theta^2 + 1} & \text{TM EoS}, \end{cases}$$
(4)

respectively. For our current purposes, these EoSs are sufficient, although it should be noted that many more are available in the literature (see e.g. Synge 1957; Rezzolla 2013). Due to the modular design of CRONOS, additional EoSs can be easily added in the future by supplying the relevant routines.

2.2. Passive tracer fields

CRONOS provides the option to solve user-defined conservation laws alongside the ones governing the fluid dynamics (see also Kissmann et al. 2018). A prominent use case for this is the treatment of tracers fields ψ , which are passively advected with the fluid flow. Their value remains constant along fluid lines, following the transport equation

$$(\partial_t + \mathbf{v} \cdot \nabla) \psi = 0. \tag{5}$$

A corresponding conservation equation for $\Psi = \psi D$ can be found by combining Eqs. (2a) and (5), yielding

$$\partial_t \Psi + \nabla \cdot (\Psi \mathbf{v}) = 0. \tag{6}$$

This equation was treated in analogy to the mass-conservation equation Eq. (2a) by the fluid solvers.

3. Notation and general structure

3.1. Coordinate systems

In CRONOS, we employed orthogonal, curvilinear coordinate systems in three dimensions, which are expressed in the generic variables (x^1, x^2, x^3) and the corresponding spatial metric tensor $\gamma = \text{diag}(h_1^2, h_2^2, h_3^2)$ given by the scaling factors h_i . Specifically, we have implemented Cartesian, cylindrical, and spherical coordinates (see also Table A.1). For the latter two, the divergence of the rank-2 stress tensor in the momentum-conservation equation Eq. (2c) yields non-zero geometrical source terms G_{m^i} . The expression can be expanded component-wise in terms of the standard vector-divergence in the respective coordinate system, yielding

$$\partial_t m^j + \nabla \cdot \left(m^j \mathbf{v} + p \, \mathbf{e}_j \right) = S_{m^j} + G_{m^j},\tag{7}$$

where e_j denotes the unit base vector along the *j*th coordinate direction (see Appendix A).

3.2. Normalisation

For the numerical treatment of the equations, CRONOS uses normalised quantities internally in the form $X = X_0 \hat{X}$, where X is a physical quantity, X_0 a normalisation constant, and \hat{X} the resulting unit-free, normalised quantity that is treated by the solvers (Kissmann et al. 2018). Our implementation foresees four base quantities that can be normalised independently, which correspond to the length, time, mass, and number density. The latter is not strictly necessary to have a complete system of units. However, it has been proven to be useful in practice since employing a normalisation derived from the length-scale can lead to unwieldy large values for the number density.

The normalisations can be chosen freely by the user, depending on the problem at hand by specifying the normalisation constants for four different quantities, which can also include derived quantities (e.g. energy density, velocity, etc.). If constants for derived quantities are supplied, the normalisations for the non-specified base quantities are obtained automatically via physical relations. If the specified quantities form an inconsistent system of units, the procedure is aborted and the user is notified. In contrast to classical hydrodynamics, velocities are always normalised to the speed of light in the relativistic case, as implicitly imposed by Eqs. (2a)–(2c). The user is hence only allowed to specify the three remaining normalisations to complete the system.

Finally, the normalised equations, corresponding to Eqs. (2a)–(2c), that are solved by CRONOS are

$$\partial_{\hat{i}}\hat{D} + \hat{\nabla} \cdot (\hat{D}\,\hat{\mathbf{v}}) = \hat{S}_{D} \tag{8a}$$

$$\partial_{\hat{\tau}}\hat{\tau} + \hat{\nabla} \cdot ((\hat{\tau} + \hat{p})\hat{\mathbf{v}}) = \hat{S}_{\tau} \tag{8b}$$

$$\partial_{\hat{r}}\hat{m}^{j} + \hat{\nabla} \cdot \left(\hat{m}^{j}\hat{\boldsymbol{v}} + \hat{p}\,\hat{\boldsymbol{e}}_{j}\right) = \hat{S}_{m^{j}} + \hat{G}_{m^{j}}. \tag{8c}$$

For the rest of this work, we exclusively use normalised quantities and omit the hat sign on these for better readability. Once a solution in normalised quantities was obtained, physical quantities could be easily computed in a post-processing step using the corresponding normalisation constants, which are therefore stored together with the data in the output files.

3.3. Grid structure

To solve the set of equations Eqs. (8a)–(8c), CRONOS employs a finite-volume scheme, decomposing the computational volume into a set of disjoint cells (see e.g. LeVeque 2002). While the shapes of the cells are in principle irrelevant for such a scheme, we restricted ourselves to three-dimensional, rectangular grids for practical reasons.

The division into cells is achieved by subdividing each coordinate axis $d \in \{1, 2, 3\}$ spanning the computational domain (x_b^d, x_e^d) into N^d intervals $\omega_{i^d}^d$, which are addressed by the index $i^d \in \{0, \dots, N^d - 1\}$. The edges of the individual intervals $\omega_{i^d}^d = \left(x_{i^d-1/2}^d, x_{i^d+1/2}^d\right)$ can be freely chosen as long as they are strictly increasing and first and last edges match the beginning and the end of the computational domain, that is $x_{-1/2}^d = x_b^d$ and $x_{N^d-1/2}^d = x_e^d$, respectively. Usually, the edges are chosen to be spaced linearly, however, non-linear options are also implemented (see Kissmann et al. 2018).

The individual cells are then given by $C_I = \omega_{i^1}^1 \times \omega_{i^2}^2 \times \omega_{i^3}^3$ and are correspondingly addressed by the index triplet $I = (i^1, i^2, i^3)$. The centre of a cell $x_I^d := (x_{i^d-1/2}^d + x_{i^d+1/2}^d)/2$ is defined to be located at the midpoint of each coordinate interval.

A cell C_I borders on a set of neighbouring cells with indices $J \in N(I)$. The interfaces to its neighbours $\Sigma_{I,J}$ can therefore be used to decompose the cell's surface as $\partial C_I = \bigcup_{J \in N(I)} \Sigma_{I,J}$.

For rectangular grids, the cell interface normals $n_{I,J}$ are further (anti-)parallel to the $d_{I,J}$ th coordinate basis vectors, yielding $n_{I,J} \cdot e_{d_{I,J}} =: \sigma_{I,J} = \pm 1$.

In contrast to finite difference schemes, where the values of dynamical variables are stored as point values on a grid, in a finite-volume approach, cell-averaged values are stored. The averaged primitive variables \bar{w} , which are initially set by the user, are defined by

$$\bar{w}_I := \frac{1}{\Delta V_I} \int_{C_I} w \, dV \quad \text{with} \quad \Delta V_I := \int_{C_I} dV.$$
 (9)

The corresponding set of averaged conserved variables \bar{U} is defined in analogy.

3.4. Semi-discrete finite-volume scheme

To obtain a numerical scheme to update the cell averages \overline{w} or respectively \overline{U} , the set of conservation equations Eq. (1) was integrated over the extent of a cell C_I . Substituting the respective cell averages and applying Gauss's theorem yields

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{U}_{I} = \overline{S}_{I} - \frac{1}{\Delta V_{I}} \int_{\partial C_{I}} \mathbf{F} \cdot \,\mathrm{d}\mathbf{A},\tag{10}$$

where \overline{S} denotes the cell averaged source term. Splitting the cell's surface into disjoint interfaces to its neighbours and using the interface-averaged fluxes defined as

$$\overline{F}_{I,J} := \frac{\sigma_{I,J}}{\Delta A_{I,J}} \int_{\Sigma_{I,J}} F^{d_{I,J}} \, \mathrm{d}A, \quad \Delta A_{I,J} := \int_{\Sigma_{I,J}} \, \mathrm{d}A \tag{11}$$

allows Eq. (10) to be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\overline{U}_{I} = \bar{\mathcal{L}}_{I} := \overline{S}_{I} - \frac{1}{\Delta V_{I}} \sum_{I \in \mathcal{N}(I)} \Delta A_{I,J} \, \overline{F}_{I,J}. \tag{12}$$

With this, the initial system of partial differential equations in Eq. (1) has been cast into a system of ordinary differential equations (ODEs) for each grid cell. This is usually referred to as a semi-discrete scheme, since the problem was discretised in space but not in time, having the advantage that the spatial and temporal parts of the problem are conceptually decoupled (see e.g. also Kurganov & Tadmor 2000; LeVeque 2002).

4. Numerical scheme

In CRONOS we followed the general idea presented by Godunov (1959), but used higher-order schemes in both space and time, instead. First, we obtained interface-centred point values of the fluid variables at both sides of a given cell interface using a second-order spatial reconstruction (see Sect. 4.1). The resulting jump in the reconstructed quantities can be viewed as a local, one-dimensional Riemann problem, which was solved by approximate Riemann solvers (see Sect. 4.2) to obtain an estimation for the fluxes $\overline{F}_{I,J}$ in Eq. (12). Finally, advancing the solution in time was handled by a second-order Runge-Kutta scheme (see Sect. 4.3).

4.1. Reconstruction

To provide the initial data for the Riemann problems to be solved at the cell interfaces, a spatial reconstruction of the dynamical variables was used. The order of the reconstruction thereby critically determines the spatial accuracy of the overall method. In CRONOS we applied piecewise-linear reconstructions for every coordinate direction independently, using minmod (see van Leer 1979; Harten et al. 1983) and van-Leer (see van Leer 1977) slope limiters. The reconstruction methods employed for special relativistic hydrodynamics are identical to the ones for classical hydrodynamics (see Kissmann et al. 2018).

The reconstruction was applied for primitive variables, yielding the left- (L) and right-handed (R) states $w_{I,J}^{\rm L}$ and $w_{I,J}^{\rm R}$, respectively, at a given cell interface $\Sigma_{I,J}$. Therefrom, the corresponding point values of the conserved variables $U_{I,J}^{\rm L/R}$ were computed subsequently. Since the flux function in Eqs. (2a)–(2c) is expressed in terms of primitive variables, this approach has the additional advantage that the more expensive variable conversion from conserved to primitive ones (see Sect. 4.4) can be avoided.

As opposed to Newtonian hydrodynamics, where the fluid speed is generally not bound, it has to remain subluminal at all times for relativistic hydrodynamics. This constraint can be easily violated if three-velocity components are reconstructed independently of each other. In our case, related problems were avoided from the outset since we used and reconstructed spatial four-velocity components instead, which are not subject to such a constraint.

4.2. Approximate Riemann solvers

For a given cell interface $\Sigma_{I,J}$, the reconstructed states at both sides were used to define a local Riemann problem that can be formalised as

$$w(t_0, \xi) = \begin{cases} w_{I,J}^{L} & \xi < \xi_{I,J} \\ w_{I,I}^{R} & \xi > \xi_{I,J}, \end{cases}$$
 (13)

with $\xi = x^{d_{IJ}}$ denoting the coordinate aligned with the interface normal. Assuming the cell-interface to be flat, the conservation Eq. (1) governing the evolution of the Riemann problem reduces to

$$\partial_t U + \partial_{\mathcal{E}} f(w) = 0, \tag{14}$$

where $f = \mathbf{e}_{d_{I,J}} \cdot \mathbf{F} = F^{d_{I,J}}$ corresponds to the flux through the interface.

Although relativistic Riemann problems cannot be solved in a closed analytic form, semi-analytic iterative solvers have been developed that can compute their solution to arbitrary precision (see e.g. Marti & Müller 1994; Pons et al. 2000). Employing such solvers, however, is cumbersome and computationally very expensive, which makes them unsuitable for an application in simulation codes.

Instead, approximate Riemann solvers have been developed, making simplifying assumptions on the waves emerging from the initial discontinuity. This significantly reduces the computational cost, while still yielding good approximations for the fluxes at the cell interfaces. In CRONOS we employed the HLL and the HLLC special-relativistic Riemann solvers (Mignone & Bodo 2005). Since every cell interface is treated individually and identically by the solver, we omit the cell indices, for example $w_L \equiv w_{I,J}^L$, for the remainder of this section for the sake of better readability.

4.2.1. HLL solver

One of the most commonly employed Riemann solvers is the HLL solver first proposed by Harten et al. (1983) for the classical Euler equations. The generalisation to the relativistic case is straightforward and was used in Schneider et al. (1993) for the first time. Here, we briefly summarise the expressions relevant for the implementation in CRONOS.

The solver assumes that the Riemann problem produces two shock waves propagating with speeds $\lambda_{\rm L}$ and $\lambda_{\rm R}$ computed from Eqs. (27a) and (27b). Only one constant intermediate state U^* is considered to be formed between both shocks, while the states ahead of the shocks remain unchanged. The intermediate state can be computed directly from the Rankine-Hugoniot jump conditions at both shocks

$$U^* = \frac{\lambda_R U_R - \lambda_L U_L + f_L - f_R}{\lambda_R - \lambda_L},$$
(15)

using the fluxes of the undisturbed states at either side $f_{L/R} = f(w_{L/R})$. The corresponding fluxes in the intermediate region are then given by

$$f^* = \frac{\lambda_R f_L - \lambda_L f_R + \lambda_L \lambda_R (U_R - U_L)}{\lambda_R - \lambda_L}$$
 (16)

Depending on the signal speeds $\lambda_{L/R}$, the numerical cell-interfaces fluxes are ultimately given by

$$F^{\text{HLL}} = \begin{cases} f_{\text{L}} & \text{for } 0 < \lambda_{\text{L}} \\ f^* & \text{for } \lambda_{\text{L}} < 0 < \lambda_{\text{R}} \\ f_{\text{R}} & \text{for } \lambda_{\text{R}} < 0. \end{cases}$$
 (17)

Although the HLL solver is quite simple and computationally inexpensive, it yields good and robust approximations for the cell-interface fluxes. In general, the solver has been found to perform well at shock and rarefaction waves. However, since contact discontinuities were not considered in the simplified wavefan, the HLL solver introduces additional numerical dissipation at such waves.

4.2.2. HLLC solver

A natural improvement to the HLL solver is to restore the neglected contact discontinuity. The corresponding approximate Riemann solver, which considers all physical intermediate states, is known as the HLLC solver and was first presented by Toro (1997) for the classical Euler equations and later extended to special relativistic hydrodynamics by Mignone & Bodo (2005). Again, we briefly summarise the quantities relevant for our implementation in CRONOS.

The solver assumes the emerging wavefan to be composed of two shocks together with a contact discontinuity in between, separating two intermediate states U_s^* with $s \in \{L, R\}$, respectively. Using the Rankine-Hugoniot jump conditions for both shock waves and assuming that the intermediate-state fluxes take the following form $f_s^* = f(w_s^*)$, which is not necessarily the case in general (Mignone & Bodo 2005), the speed of the contact discontinuity λ^* is given by

$$(\lambda^*)^2 [\lambda_L A_R - \lambda_R A_L] + [B_R - B_L] - \lambda^* [A_R - A_L + \lambda_L B_R - \lambda_R B_L] = 0, \quad (18)$$

with

$$A_{\rm s} := \lambda_{\rm s}(\tau_{\rm s} + D_{\rm s}) - m_{\rm s}^{\rm x} \tag{19}$$

$$B_{\rm s} := m_{\rm s}^{\rm x}(\lambda_{\rm s} - v_{\rm s}^{\rm x}) - p_{\rm s}. \tag{20}$$

We solved this equation analytically, taking only the root with the minus sign into account since this is the only physically acceptable one (Mignone & Bodo 2005).

After computing the intermediate-state pressure for one side

$$p_{\rm s}^* = \frac{\lambda^* A_{\rm s} - B_{\rm s}}{1 - \lambda_{\rm s} \lambda^*} \tag{21}$$

and using the constant pressure over the contact discontinuity $p^* \equiv p_{\rm L}^* \equiv p_{\rm R}^*$, the remaining quantities of the intermediate states are given by

$$m_{\rm s}^{x*}(\lambda_{\rm s} - \lambda^*) = m_{\rm s}^{x}(\lambda_{\rm s} - v_{\rm s}^{x}) - p_{\rm s} + p^*$$
 (22)

$$\tau_s^*(\lambda_s - \lambda^*) = \tau_s(\lambda_s - v_s^x) - v_s^x p_s + \lambda^* p^*$$
(23)

$$\zeta_{\rm s}^*(\lambda_{\rm s} - \lambda^*) = \zeta_{\rm s}(\lambda_{\rm s} - v_{\rm s}^{\rm x}). \tag{24}$$

Here, ζ represents any passively advected, conserved quantity such as the mass density D, the momentum densities m^{t_1} , m^{t_2} tangential to the interface, or any generic tracer field Ψ . Finally, the numerical fluxes at the cell interfaces were obtained from

$$F^{\text{HLLC}} = \begin{cases} f_{\text{L}} & \text{for } 0 < \lambda_{\text{L}} \\ f_{\text{L}}^* & \text{for } \lambda_{\text{L}} < 0 < \lambda^* \\ f_{\text{R}}^* & \text{for } \lambda^* < 0 < \lambda_{\text{R}} \\ f_{\text{R}} & \text{for } \lambda_{\text{R}} < 0. \end{cases}$$
(25)

To avoid a direct evaluation of the flux function $f(w_s^*)$ for the intermediate states, which would require a variable inversion,

the intermediate-state numerical fluxes were computed from the Rankine-Hugoniot conditions instead:

$$f_{s}^{*} = f_{s} + \lambda_{s} (U_{s}^{*} - U_{s}). \tag{26}$$

The HLLC solver has been shown to have superior performance compared to the more simple HLL solver. However, it has also been reported to potentially suffer from the carbuncle phenomenon (see e.g. Quirk 1994; Dumbser et al. 2004).

4.2.3. Wave speed estimates

Both the HLL and HLLC Riemann solvers require an estimate for the fastest signal speeds to the left and to the right of the discontinuity λ_L , λ_R , respectively (see also Toro 1997). They can be obtained from the smallest and largest eigenvalues of the system in Eq. (14) and were first computed by Davis (1988) for the relativistic case. Although this is not the only possible estimate (e.g. the Roe average, see Roe 1981), it is commonly used in many codes (see e.g. Mignone et al. 2007; Lamberts et al. 2013). In Cronos we used

$$\lambda_{L} = \min\left(\lambda_{-}(w_{L}), \lambda_{-}(w_{R})\right) \tag{27a}$$

$$\lambda_{\mathbf{R}} = \max\left(\lambda_{+}(w_{\mathbf{L}}), \lambda_{+}(w_{\mathbf{R}})\right),\tag{27b}$$

where λ_{\pm} are the smallest and largest eigenvalues of the system, respectively, which take the form

$$\lambda_{\pm}(w) = \frac{u^0 u^d \pm \sqrt{u_s^2 \left(u^{0^2} - u^{d^2} + u_s^2\right)}}{u^{0^2} + u_s^2},\tag{28}$$

with $d = d_{I,J}$ corresponding to the direction of the interface normal, $u_s^2 := c_s^2/(1 - c_s^2)$, and the sound speed (taken from Mignone & McKinney 2007)

$$c_{\rm s}^2(\Theta) = \begin{cases} \frac{\Gamma\Theta}{h} & \text{ideal EoS} \\ \frac{\Theta}{3h} + \frac{5h - 8\Theta}{h - \Theta} & \text{TM EoS}, \end{cases}$$
 (29)

with $h(\Theta)$ being computed from Eq. (4).

4.3. Time integration

In CRONOS, we treated the system of ODEs, arising in the semi-discrete framework in Eq. (12) in time, using a second-order total-variation-diminishing Runge-Kutta scheme, that is Heun's method (see Shu 1988)

$$\overline{U}_{I}^{*} = \overline{U}_{I}^{n} + \bar{\mathcal{L}}_{I}(w^{n}) \Delta t \tag{30a}$$

$$\overline{U}_I^{n+1} = \frac{1}{2} \left(\overline{U}_I^n + \overline{U}_I^* + \overline{\mathcal{L}}_I(w^*) \Delta t \right), \tag{30b}$$

where the indices n, n + 1, and * indicate values at the current, the next, and an intermediate time-step.

The evaluation of the dimensionally unsplit operator $\bar{\mathcal{L}}_I$ for a given cell requires the computation of the numerical fluxes at every cell interface. This involves the solution of Riemann problems, which is the most expensive part of the scheme. Since $\bar{F}_{I,J} = -\bar{F}_{J,I}$ by construction, we iterated over interfaces in the computational domain instead of over cells to avoid unnecessary double computations. In practice, we further iterated over one-dimensional pencils for each coordinate, which allows for the efficient utilisation of the coordinate-wise nature of the reconstructions. This is followed by another iteration over the

whole domain regarding the source term, where we evaluated the respective expressions at the cell centres as

$$\overline{S}_I = S(t, x_I, \overline{w}_I). \tag{31}$$

The step size for the numerical time integration was constrained by the CFL condition (see Courant et al. 1928) and was obtained from

$$\Delta t = C_{\text{CFL}} \min_{I,J} \left(\frac{\Delta L_I^{d_{I,J}}}{\lambda_{I,J}^{\text{max}}} \right), \tag{32}$$

where C_{CFL} is the Courant number and $\lambda_{I,J}^{\max}$ the largest absolute value of the signal speeds computed at the cell interface $\Sigma_{I,J}$, which was retrieved from the characteristic speeds that have been computed for the Riemann solvers. We allowed the user to set a specific value for C_{CFL} in each parameter file, individually. For dimensionally unsplit schemes, such as the one described here, the Courant number was chosen to be $C_{\text{CFL}} \leq 1/N_d$ to maintain stability in any case (see also Lamberts et al. 2013), where N_d is the dimension of the problem. Depending on the problem, however, the condition might be eased; for example, for radially symmetric flows, a Courant number of $C_{\text{CFL}} \leq 1/\sqrt{N_d}$ is sufficient. The time step is further adapted if an output checkpoint or the end of the simulation is reached. After obtaining a solution at t^{n+1} , user-supplied routines are invoked, for example, to reset the solution in a given region, before the time integration procedure starts again.

In contrast to Newtonian hydrodynamics, where the signal speeds λ^{\max} are not limited in general, they are bound by the speed of light in relativity. In practice, this implies that the step size computed from Eq. (32) does not become smaller than $\Delta t_{\min} = C_{\text{CFL}} \min_{d,l} \left(\Delta L_l^d \right) / c$, which only depends on the extension of the smallest cell. Such a lower bound has been proven to be useful in practice for the estimation of maximum simulation runtimes.

4.4. Variable inversion

Since the fluxes are given in terms of the primitive fluid variables $w=(\rho,p,u^j)$ but the time integration has to be performed on conserved variables $U=(D,\tau,m^j)$ to obtain a shock-capturing scheme, transformations between the two sets are required. While the conversion from primitive quantities to conserved ones is trivial and can be implemented in a straightforward manner, the conversion in the other direction is not and it involves solving non-linear equations for at least one of the quantities. For this, we closely followed the approach presented by Mignone & McKinney (2007), which was initially developed for relativistic MHD, and we adapted it for our purposes (similar to Lamberts et al. 2013). We solved the non-linear equation

$$f(W') := W' - \tau - p(\chi, \rho) = 0, \tag{33}$$

where $W' = D(hu^0 - 1)$ and $\chi = \rho(h - 1)$. The function $p(\chi, \rho)$ depends on the chosen EoS and can be found in Table 1 for the different cases implemented in CRONOS. To evaluate f for a given W', we first computed the corresponding four-speed via

$$|\mathbf{u}|^2(W') = \frac{|\mathbf{m}|^2}{(W'+D)^2 - |\mathbf{m}|^2},\tag{34}$$

which was used to obtain (Mignone & McKinney 2007)

$$\chi(W', |\mathbf{u}|^2) = \frac{W'}{u^{02}} - \frac{D|\mathbf{u}|^2}{u^{02}(1+u^0)}$$
(35)

Table 1. Thermodynamical quantities for different EoSs implemented in CRONOS (see also Sect. 2) necessary for the variable inversion (compiled from Mignone & McKinney 2007).

	$p(\chi, \rho)$	$\partial_{\chi} p$	$\partial_{ ho} p$
Ideal EoSs TM EoSs	$\frac{\frac{\Gamma-1}{\Gamma}\chi}{2\chi(\chi+2\rho)}$ $\frac{5(\chi+\rho)+\sqrt{9\chi^2+18\rho\chi+25\rho^2}}{5(\chi+\rho)+\sqrt{9\chi^2+18\rho\chi+25\rho^2}}$	$\frac{\frac{\Gamma-1}{\Gamma}}{\frac{2\chi+2\rho-5p}{5\rho+5\chi-8p}}$	$0 \\ \frac{2\chi - 5p}{5\rho + 5\chi - 8p}$

and

$$\rho(W', |\boldsymbol{u}|^2) = \frac{D}{u^0} \tag{36}$$

employing the identity $(u^0)^2 = 1 + |\mathbf{u}|^2$.

We solved Eq. (33) numerically using an iterative Newton-Raphson scheme until the relative change was below a certain threshold ϵ , which can be set by the user depending on the problem (typically $\epsilon=10^{-8}$). It can be shown analytically that the scheme is accurate in the ultra-relativistic limit as long as $u^0 \lesssim \epsilon^{-1/2}$ and $p/(\rho u^{0^2}) \gtrsim \epsilon$ (see Mignone & McKinney 2007), which might require an adjustment of ϵ for certain problems.

The derivative df/dW' required by the algorithm is expressed as

$$\frac{\mathrm{d}f}{\mathrm{d}W'} = 1 - \frac{\partial p}{\partial \chi} \frac{\mathrm{d}\chi}{\mathrm{d}W'} - \frac{\partial p}{\partial \rho} \frac{\mathrm{d}\rho}{\mathrm{d}W'},\tag{37}$$

where the kinematical derivatives are given by

$$\frac{\mathrm{d}\chi}{\mathrm{d}W'} = \frac{1}{u^{0^2}} + u^0 \left(D + 2u^0 \chi\right) \frac{|\mathbf{m}|^2}{\left(W' + D\right)^3} \tag{38}$$

$$\frac{d\rho}{dW'} = \frac{Du^0 |\mathbf{m}|^2}{(W' + D)^3}.$$
(39)

The derivatives involving the pressure $\partial_{\chi}p$ and $\partial_{\rho}p$ again depend on the chosen EoS and are also listed in Table 1.

The Newton-Raphson iteration is started using the initial value

$$W_0' = \frac{1}{3} \left(2E + \sqrt{4E^2 - 3|\mathbf{m}|^2} \right) - D. \tag{40}$$

This choice guarantees pressure positivity given the conservative state is physically admissible (Mignone & McKinney 2007).

Once a sufficiently precise solution for W' is found, the primitive variables are given by Eq. (36), $p(\chi, \rho)$ supplied by the EoS, and by

$$u(W') = \frac{m}{(W'+D)^2 - |m|^2}.$$
(41)

4.4.1. Admissible conserved states

If a given conserved state is unphysical, the variable inversion fails and produces unphysical primitive states, for example, with negative density or pressure, superluminal speeds, etc., or the inversion does not converge at all. To avoid related problems, we checked the admissibility of a given conserved state $U \in \mathcal{G}_1$ before every inversion, relying on the set of physically admissible states formulated by Wu & Tang (2015):

$$G_1 = \{ U \mid D > 0, \ q(U) := \tau + D - \sqrt{D^2 + |\mathbf{m}|^2} > 0 \}.$$
 (42)

If the constraint is violated ($U \notin \mathcal{G}_1$), the user is informed and the simulation is aborted by default. Alternatively, the user can choose to overwrite the states in the concerned cells with new ones by supplying user-defined routines, for example, setting a lower limit for the pressure and computing other quantities accordingly.

4.4.2. Entropy inversion

In situations where the thermal energy is negligible compared to the kinetic energy, simple discretisation errors can lead to unphysical states $(U \notin \mathcal{G}_1)$ for which the variable inversion will fail. To avoid related problems, we offer the user the possibility to solve an additional equation for the conservation of entropy in smooth flows

$$\partial_t \Sigma + \nabla \cdot (\Sigma \mathbf{v}) = 0, \tag{43}$$

next to the hydrodynamic equations (in analogy to our procedure for classical HD and MHD Kissmann et al. 2018), with $\Sigma = \sigma D$ and the entropy (Mignone & McKinney 2007)

$$\sigma = \begin{cases} \frac{p}{\rho^{T}} & \text{ideal EoS} \\ \frac{p}{\rho^{5/3}}(h - \Theta) & \text{TM EoS}. \end{cases}$$
 (44)

Numerically, the equation was treated in analogy to passive tracer fields (see Sect. 2.2), which guaranteed that we maintained positive values for Σ using the implemented solvers. Although the conservation of energy might be slightly violated by employing Σ instead of τ in the variable inversion, the resulting scheme is much more stable, which can be necessary for certain applications.

When using Σ , the inversion was performed by numerically solving

$$q(\rho) = \rho u^{0}(h(\rho)) - D = 0, \tag{45}$$

where

$$u^{0}(h) = \sqrt{1 + \frac{|\boldsymbol{m}|^{2}}{D^{2}h^{2}}} \tag{46}$$

and

$$h^{2}(\rho) = \begin{cases} \left(1 + \frac{\Gamma}{\Gamma - 1}\sigma\rho^{\Gamma - 1}\right)^{2} & \text{ideal EoS} \\ \frac{\left(1 + 4\rho^{2/3}\sigma\right)^{2}}{1 + 3\rho^{2/3}\sigma} & \text{TM EoS.} \end{cases}$$
(47)

Although the derivative of Eq. (45) can be easily found and implemented for a Newton-Raphson scheme to solve the equation, we used a hybrid, bracketing, root-finding method, that is Brent's method (Brent 1972), instead. For our use cases, we found a comparable performance for both solvers, while the latter one is guaranteed to converge in any case. As an initial bracket for the scheme, we used $\rho_{\text{max}} = D$ and $\rho_{\text{min}} = D^2/\sqrt{D^2 + |\mathbf{m}|^2}$. These limits were obtained from $\rho = D/u^0(h)$ using the asymptotic values of the specific enthalpy $1 < h < \infty$ in the ultra-cold and ultra-hot limit, respectively.

Once a sufficiently accurate value for ρ was found, the inversion was completed by computing the fluid velocity from $u = m/Dh(\rho)$ and the pressure according to

(42)
$$p = \begin{cases} \sigma \rho^{\Gamma} & \text{ideal EoS} \\ \frac{1}{2}\rho \left(h(\rho) - \sqrt{h^2(\rho) - 4\sigma\rho^{2/3}}\right) & \text{TM EoS.} \end{cases}$$

Table 2. Parameters for the Riemann problems according to Eq. (50) solved in Sect. 5.1 (taken from Mignone & Bodo 2005) together with the L_1 errors of the numerical solutions for the density and the pressure, respectively.

	P1	P2	Р3	P4
$w_{\rm L}$ $w_{ m R}$		(1, -0.6, 10) (10, 0.5, 20)	$(10, 0, 40/3)$ $(1, 0, 2/3 \times 10^{-6})$	$(1,0,10^3)$ $(1,0,10^{-2})$
$L_1^ ho \ L_1^p$	4.05×10^{-2} 6.88×10^{-2}		5.94×10^{-2} 3.23×10^{-2}	1.62×10^{-1} 2.94

The conservation of entropy, however, only applies for smooth, non-compressive flows. To identify those cells for which this is the case and the variable inversion can be feasibly performed using Eq. (45), we flagged cells according to the following criteria (similar to Balsara & Spicer 1999; Mignone et al. 2020):

$$\mathcal{F}_{I}^{\Sigma} = \left(\|\nabla p_{I}\| \max_{d} \Delta L_{I}^{d} < \alpha_{1} \min_{I, N(I)} p \right) \vee$$

$$\left(-\alpha_{2} \max_{I, N(I)} c_{s} < (\nabla \cdot \mathbf{v}) \min_{d} \Delta L_{I}^{d} \right).$$

$$(49)$$

The occurring gradient and divergence were approximated using central, finite differences and c_s denotes the local sound speed computed according to Eq. (29). We typically used values of $\alpha_1=0.05$ and $\alpha_2=0$. Further, if a given cell I did not qualify for the entropy inversion ($\mathcal{F}_I^\Sigma=0$), we also disabled the flag for its direct neighbours for safety ($\mathcal{F}_J^\Sigma=0$ for $J\in N(I)$). Wherever $\mathcal{F}_I^\Sigma=1$, we performed the variable inversion using Eq. (45). Otherwise, Eq. (33) was used and the correspond-

Wherever $\mathcal{F}_I^{\Sigma} = 1$, we performed the variable inversion using Eq. (45). Otherwise, Eq. (33) was used and the corresponding value of σ was recomputed from the resulting state. The described procedure can be enabled or disabled by a compiletime switch set by the user.

5. Validation

In the following sections, we present the results from a variety of numerical tests to verify our implementations within the CRONOS code.

5.1. 1D Riemann problems

One-dimensional Riemann problems are particularly useful numerical tests for hydrodynamic codes since they can be solved by semi-analytical methods to arbitrary precision (Marti & Müller 1994). The Riemann problems for the following tests are posed in the general form

$$(\rho, v^x, p) = \begin{cases} w_{\rm L} & \text{for } x < 0.5 \\ w_{\rm R} & \text{for } x > 0.5. \end{cases}$$
 (50)

For comparison, we considered the same set of Riemann problems as in Mignone & Bodo (2005), summarised in Table 2 with zero fluid velocity tangential to the discontinuity and an ideal EoS with $\Gamma = \frac{5}{3}$. All numerical tests were performed using extrapolating boundary conditions.

In Fig. 1, we present the numerical evolution of the considered Riemann problems together with their exact solutions¹.

In P1 and P2, two shock waves and two rarefaction waves are formed, respectively, which are in good agreement with the exact solutions. The tests P3 and P4 are characterised by large differences in pressure between the initial states and they evolve into a leftward-moving rarefaction and a rightward-moving shock wave, forming dense shells of material. These shells are also known as relativistic blast waves and pose notoriously difficult problems for numerical schemes. For P4, the numerical solution yields a shock position and a compression ratio that slightly differ from their exact locations and values. This is caused by an insufficient resolution of the blast wave in combination with the spatial reconstruction effectively reducing to first order in the vicinity of shocks. Overall, the numerical solutions obtained using CRONOS are in good agreement with the exact, semi-analytic solutions, which validates the implemented solvers.

5.2. 1D Riemann problems with non-zero tangential velocity

While velocities tangential to the initial discontinuity v^t do not affect the evolution of Riemann problems in classical hydrodynamics, they do have a severe impact on the emerging wavefan in relativistic hydrodynamics because of the non-linear coupling introduced by the Lorentz factor. We demonstrated that these effects are also correctly handled by our implementations by solving the Riemann problem

$$(\rho, v^x, v^t, p) = \begin{cases} (1, 0.5, 0, 1) & \text{for } x < 0.5\\ (0.125, 0, v_R^t, 0.1) & \text{for } x > 0.5 \end{cases}$$
 (51)

for different values of the right-sided, tangential velocity v_R^t in analogy to the ones in Sect. 5.1. Exact semi-analytic solutions with non-zero tangential velocities are again available in the literature (see Pons et al. 2000; Rezzolla et al. 2003), which were used to produce the reference solutions shown in Fig. 2.

We investigated two scenarios with $v_{\rm R}^t=0.7$ and $v_{\rm R}^t=0.9$ and show the numerical solutions in Fig. 2. The numerical results agree well with the exact solutions and show the transition from a rarefaction-shock to a shock-shock pattern expected at $v_{\rm R}^t\sim0.85$ (Rezzolla et al. 2003). The numerical L_1 errors are $L_1^\rho=6.20\times10^{-3}$; 9.76×10^{-3} for the density and $L_1^p=6.61\times10^{-3}$; 9.04×10^{-3} for the pressure, corresponding to $v_{\rm R}^t=0.7$; 0.9, respectively.

5.3. 2D Riemann problem

After the successful tests in one dimension, we now move to tests in higher dimensions. First, we consider a two-dimensional Riemann problem, which is posed by the interaction of four different initial states; one in each quadrant of the two-dimensional plane. For the sake of comparison, we solved the same problem as investigated by Del Zanna & Bucciantini (2002) that is parameterised as

$$(\rho, v^{x}, v^{y}, p) = \begin{cases} (0.1, 0, 0, 0.01) & \text{for } x > 0, y > 0\\ (0.1, 0.99, 0, 1) & \text{for } x < 0, y > 0\\ (0.5, 0, 0, 1) & \text{for } x < 0, y < 0\\ (0.1, 0, 0.99, 1) & \text{for } x > 0, y < 0. \end{cases}$$
(52)

Again, we used extrapolating boundary conditions, the minmod slope limiter, and an ideal FoS with $\Gamma = \frac{5}{2}$.

slope limiter, and an ideal EoS with $\Gamma = \frac{5}{3}$. In Fig. 3, we present the numerical results using the HLL and HLLC solver, respectively. The initial discontinuities separating the two top states and the two right states evolve into curved shock waves that enclose a drop-shaped region in the

¹ See github.com/cxkoda/srrp for an implementation of an exact Riemann solver for relativistic hydrodynamics in PYTHON.

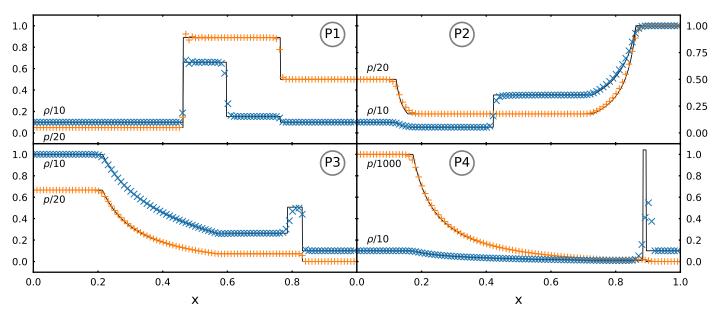


Fig. 1. Numerical results for one-dimensional Riemann problems at time t = 0.4. The density (blue crosses) and pressure (orange plus signs) are shown. The underlying solid black line corresponds to the exact solution. The numerical solutions have been obtained using 400 gridpoints, the minmod slope limiter, the HLLC solver, and CFL = 0.4. To improve visibility, we show only every fourth point of the numerical results and scaled the different quantities as annotated. The parametrisations of the Riemann problems for the numerical test can be found in Table 2.

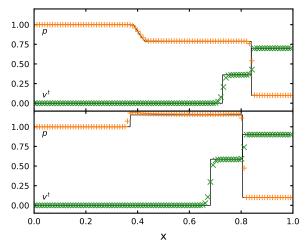


Fig. 2. Same as in Fig. 1, but for Riemann problems involving non-zero tangential velocities in the right state with $v_R^t = 0.7$ (*top*) and $v_R^t = 0.9$ (*bottom*) as parameterised in Eq. (51). The solutions for the tangential velocity (green crosses) and the pressure (orange plus signs) are shown.

top-right quadrant. The lower-left region remains confined by two contact discontinuities and forms a jet-like structure. While the contact discontinuities remain sharp and steady using the HLLC solver, they degenerate into spurious waves using the HLL solver because of the increased numerical dissipation of the latter, demonstrating the better performance of the HLLC solver. Our numerical results agree well with the ones in the literature (e.g. Del Zanna & Bucciantini 2002; Mignone & Bodo 2005). Having verified the implementation on Cartesian grids, we moved to curvilinear coordinates in the subsequent test.

5.4. Axisymmetric relativistic jet

In this numerical test, we demonstrate the implementation of geometric source terms arising in non-Cartesian coordinate systems (see Sect. 3.1). For this, we simulated the evolution of

an axisymmetric relativistic jet in 2D cylindrical geometry. The jet was set up by initialising the computational volume with an ambient, background medium (respective quantities are denoted with an index m) and injecting the supersonic beam material (denoted with an index b) by imposing constant boundary conditions at z=0 for radii r<1. At the symmetry axis of the jet r=0, we imposed reflecting boundary conditions and outflowing boundary conditions everywhere else. The computational domain covers the region $0 \le r \le 12$ and $0 \le z \le 35$. For the sake of comparison, we adopted the same set of parameters as Del Zanna & Bucciantini (2002), which are given by the beam density $\rho_b=0.1$, the beam speed $v_b^z=0.99$, the ambient medium density $\rho_m=10$, matched pressures $p_b=p_m=10^{-2}$, and an ideal EoS with $\Gamma=\frac{5}{3}$. The ambient medium is at rest, $v_m=0$, and the beam has no radial velocity, $v_p^r=0$.

The results at time t=80 are shown in Fig. 4 and exhibit typical features of relativistic jets, such as the bow-like shock of the ambient medium, a hull of shocked material, a central relativistic stream, and turbulence excited by the Kelvin-Helmholtz instability due to the high velocity shear. Simulations using the HLLC solver show more small-scale fluctuations than those using the HLL solver because of the latter's higher numerical dissipation. This again demonstrates the superior performance of the HLLC solver. The results are in good qualitative agreement with the numerical solutions obtained by Del Zanna & Bucciantini (2002) and Mignone & Bodo (2005). The slight extension of the head using the HLLC solver is potentially caused by the carbuncle problem, which usually manifests itself as an extended nose on the jet axis (Quirk 1994) and has also been seen in other implementations of this test (e.g. see Lamberts et al. 2013).

Here, we exclusively considered a single ideal EoS. The evolution of relativistic jets, however, is heavily affected by the particular choice of the EoS, which is exploited in the next tests.

5.5. Equation of state

Here we verify the different EoSs implemented in CRONOS (see also Sect. 2). For this, we again simulated the propagation of

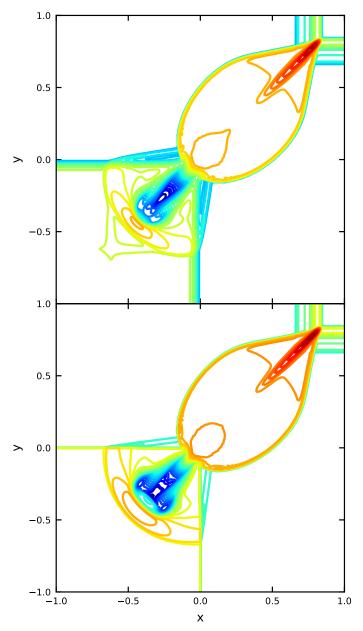


Fig. 3. Logarithmically equidistant contours of the rest-mass density for numerical solutions of the two-dimensional Riemann problem given in Eq. (52) at time t = 0.8. The results using the HLL solver (*top*) and HLLC solver (*bottom*) are shown, respectively. The solutions were obtained on a grid with 400^2 cells using a CFL number of 0.4.

relativistic jets in analogy to the setup in Sect. 5.4. For a given equation of state, the properties of a jet are determined by three parameters: the density ratio $\eta = \rho_{\rm b}/\rho_{\rm m}$, the beam's Lorentz factor $u^0_{\rm b}$, and its Mach number $M_{\rm b} = v_{\rm b}/c_{\rm s}$. Following the tests performed in Mignone et al. (2007), we adopted the same jet parameters $\eta = 10^{-3}, u^0_{\rm b} = 10, M_{\rm b} = 1.77$ and investigated the jet evolution under three different EoSs: two ideal ones with $\Gamma = \frac{5}{3}$ and $\Gamma = \frac{4}{3}$ and the TM EoS. We further chose the pressures $p_{\rm b} = p_{\rm m} = 10^{-2}$ resulting in the ambient medium densities $\rho_{\rm m}$ listed in Table 3, which were used together with $\rho_{\rm b} = \eta \, \rho_{\rm m}$ and $v_{\rm b} = 0.995$ to specify the setup for the simulation.

In Fig. 5 we show the resulting densities for the simulated jets at t = 60. Owing to the different EoSs, the head of the jets propagate at different speeds along the z-axis, which is visible from the different jet extents in the simulation.

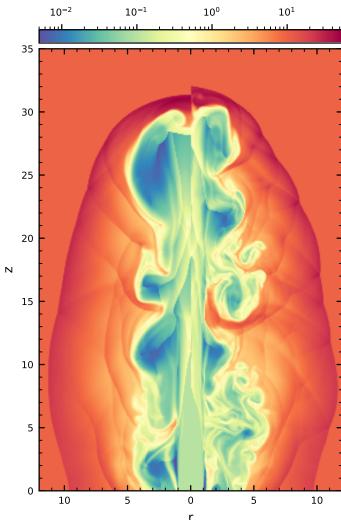


Fig. 4. Rest-mass density for the relativistic jet given in Sect. 5.4 at time t=80. The results using the HLL solver (left half) and HLLC solver (right half) are shown, respectively. The numerical solutions were obtained on a grid with 512×1536 cells in cylindrical axisymmetry using a CFL number of 0.4.

Table 3. Ambient medium densities and one-dimensional estimates for the jet speed according to Eq. (53) for the jets considered in Sect. 5.5.

	$\Gamma = \frac{5}{3}$	$\Gamma = \frac{4}{3}$	TM
$ ho_{ m m} V_{ m j}$	27.7424	2.19394	10.3337
	0.30354	0.57883	0.39243

In one dimension, the propagation of relativistic jets was investigated by Martí et al. (1997) who derived the following analytical expression for the speed of the head of a jet:

$$V_{\rm j} = \frac{u^0_{\rm b} \sqrt{\eta \, h_{\rm b}/h_{\rm m}}}{1 + u^0_{\rm b} \sqrt{\eta \, h_{\rm b}/h_{\rm m}}}$$
 (53)

The values corresponding to our setups can again be found in Table 3, which serve as a test for our implementations. In Fig. 6 we show the locations of the jet heads in our numerical simulations, which are in good agreement with the analytical estimates from Eq. (53). As expected from similar simulations performed by Mignone et al. (2007), the simulated jets become slower and

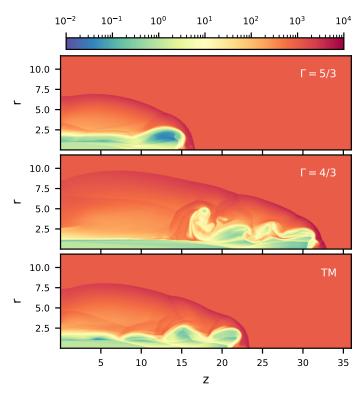


Fig. 5. Rest-mass densities for the jets simulated with different equations of state (as annotated) at t = 60. The computations were performed on a grid with 512×1536 cells in cylindrical symmetry using the HLLC solver and CFL = 0.4. The jets were set up using the parameters given in the text and Table 3.

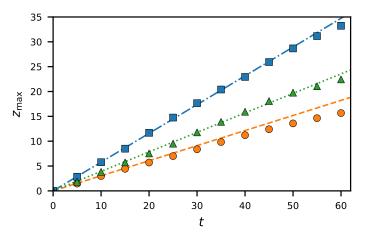


Fig. 6. Position of the head of the jet over time for different equations of state: $\Gamma = \frac{5}{3}$ (orange, circles, dashed), $\Gamma = \frac{4}{3}$ (blue, squares, dash-dotted), TM-EoS (green, triangles, dotted). Markers correspond to numerical results and lines to one-dimensional estimates given by Eq. (53). The simulated jets correspond to the ones depicted in Fig. 5.

start to deviate from the linear propagation in one dimension for later times, which follows from the two-dimensional nature of the numerical tests.

5.6. Spatial order

Finally, we verified the order of our scheme using similar tests as those performed for classical hydrodynamics with CRONOS (see Ryu & Goodman 1994; Kissmann et al. 2018). These tests

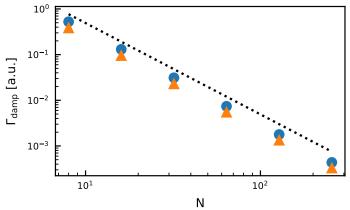


Fig. 7. Damping rate $\Gamma_{\rm decay}$ for the decaying, sound-waves test (see Sect. 5.6) as a function of the number of grid cells N using the HLL (blue circles) and the HLLC solver (orange triangles), respectively. The black dotted line indicates a N^{-2} dependence to guide the eye.

investigate the decay of two-dimensional standing sound waves by the numerical dissipation of the scheme.

In analogy to classical hydrodynamics, also in special relativity, acoustic waves can be excited by adding velocity perturbations to a constant background state, owing to the similar structure of the eigenvectors of the linearised equations (see Rezzolla 2013). We initialised sinusoidal sound waves using

$$\delta u^{x} = \delta u^{y} = \delta_{\text{amp}} \sin(k_{x}x + k_{y}y), \tag{54}$$

with the perturbation amplitude $\delta_{\rm amp}=10^{-5}$, the wavenumbers $k_x=k_y=\frac{2\pi}{L}$, and the side length of the computational domain L=1. This produces two acoustic waves that propagate with velocities $\lambda_{\pm}(w_0)$ into opposite directions, which can be computed from Eq. (28) for a given constant background state w_0 . For the purpose of this test, we chose the background density $\rho_0=1$ and the four-velocity $u_0^x=u_0^y=0$, $u_0^z=$ const. to ensure that both waves travel at the same speed, yielding

$$\lambda_{\pm} = \pm c_{\rm s} \sqrt{\frac{1 - (v_0^z)^2}{1 - (v_0^z)^2 c_{\rm s}^2}},\tag{55}$$

where $v_0^z = \frac{u_0^z}{\sqrt{1+(u_0^z)^2}}$ is the three-speed in *z*-direction and c_s the sound speed of the background medium, respectively.

In a periodic domain, this produces a standing wave that oscillates with frequency $f=\frac{\sqrt{2}}{L\lambda_+}$. To introduce relativistic effects, we set the background speed to a relativistic value of $u_0^z=\frac{2}{\sqrt{3}}$. Choosing $c_s=\frac{1}{2}$, which implies $p_0=\frac{3}{4\rho_0}$ for an ideal EoS with $\Gamma=\frac{4}{3}$, therefore leads to one oscillation in two units of time. By simulating until t=20, we thus captured ten full oscillations.

From the simulations, we retrieved the spatial root-mean-squared average of the velocity perturbation that oscillates with the double frequency. Due to the presence of numerical viscosity, its amplitude decreases exponentially $\propto \exp(-\Gamma_{\rm decay}\,t)$. The decay rate $\Gamma_{\rm decay}$ was determined from the simulation results as a function of spatial resolution and is shown in Fig. 7. The second-order nature of our scheme is obvious from $\Gamma_{\rm decay} \propto N^{-2}$. As expected, the HLLC solver introduces less numerical viscosity as compared to the HLL solver. In both cases we used the minmod slope limiter and CFL = 0.4.

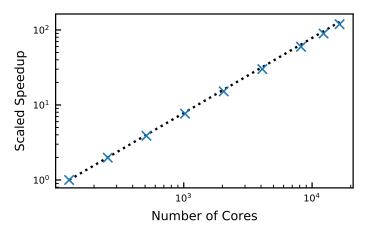


Fig. 8. Scaled speedup of simulations performed using CRONOS as a function of the number of computation cores in a weak-scaling test. The dotted line indicates perfect scaling.

6. Parallel performance

The relativistic extension of CRONOS has been successfully employed in various computational environments using up to $\sim 10^4$ computing cores. The code was parallelised using a halo-communication approach that was realised employing the Message Passing Interface (MPI) library.

To test the parallel efficiency of our code, we performed a weak-scaling test on the Joliot-Curie Rome HPC environment with AMD Rome (Epyc) 7H12 2.6GHz bi-processor compute nodes. As a test setup, we considered the colliding winds of two spherically symmetric, static sources similar to the simulation of gamma-ray binaries in Huber et al. (2021a). Initially, the system was divided into two regions, each filled by one of the winds, using an approximate curve for the contact discontinuity in the system. The simulation was run sufficiently long to obtain a representative average of the time step duration. While the domain size was increased with each test, the workload handled by each computation core was kept constant at 64^3 cells. The results are shown in Fig. 8. As expected, the parallel efficiency drops with an increasing number of cores, but remains above 93% for more than 1.6×10^4 cores.

7. Summary

We have presented and verified an extension to the CRONOS code treating special relativistic hydrodynamics for astrophysical applications. The code is based on a semi-discrete finite-volume framework, ensuring the conservation of relevant quantities to efficiently capture shocks. CRONOS employs a second-order spatial reconstruction combined with a second-order Runge-Kutta time-integrator. The extension was developed with a focus on robustness, providing the user with additional interfaces to prevent, detect, and possibly counteract failures.

Our implementation was verified against a set of standard problems, for which either an analytical solution or established numerical solutions from the literature are available. Parallel scaling for up to $\sim 10^4$ cores has been demonstrated.

As the rest of the CRONOS framework, this extension was also developed with modularity in mind, making it easy to consider additional physics supplied by the user. This was, for example, used to simulate an additional transport of energetic particles governed by a Fokker-Planck type equation alongside the fluid dynamics (as demonstrated in Huber et al. 2021a,b).

Acknowledgements. We thankfully acknowledge the access to the research infrastructure of the Institute for Astro- and Particle Physics at the University of Innsbruck (Server Quanton AS-220tt-trt8n16-g11 x8), the LEO HPC infrastructure of the University of Innsbruck and PRACE resources, that were used throughout the development of the code. We acknowledge PRACE for granting us access to Joliot-Curie at GENCI@CEA, France. D.H. acknowledges financial support by the University of Innsbruck through the stipend grant 2020/2/MIP-15 titled "Doktoratsstipendium aus der Nachwuchsförderung". This research made use of Cronos (Kissmann et al. 2018); GNU Scientific Library (GSL) (Galassi et al. 2018); matplotlib, a Python library for publication quality graphics (Hunter 2007); Scipy (Virtanen et al. 2020); and NumPy (Harris et al. 2020). We thank the anonymous referee for the thoughtful comments and suggestions that allowed us to improve our manuscript.

References

```
Aloy, M. A., Ibáñez, J. M., Martí, J. M., & Müller, E. 1999, ApJS, 122, 151
Altman, W., & de Oliveira, A. M. 2011, Mech. Adv. Mater. Struct., 18, 454
Baiotti, L., Hawke, I., Montero, P. J., et al. 2005, Phys. Rev. D, 71, 024035
Balsara, D. S., & Spicer, D. 1999, J. Comput. Phys., 148, 133
Banyuls, F., Font, J. A., Ibáñez, J. M., Martí, J. M., & Miralles, J. A. 1997, ApJ,
Barkov, M. V., Lyutikov, M., Klingler, N., & Bordas, P. 2019, MNRAS, 485,
  2041
Brent, R. P. 1972, Algorithms for Minimization without Derivatives (Englewood
   Cliffs: Prentice-Hall)
Busza, W., Rajagopal, K., & van der Schee, W. 2018, Annu. Rev. Nucl. Part. Sci.,
   68, 339
Courant, R., Friedrichs, K., & Lewy, H. 1928, Math. Ann., 100, 32
Davis, S. F. 1988, SIAM J. Sci. Stat. Comput., 9, 445
Del Zanna, L., & Bucciantini, N. 2002, A&A, 390, 1177
Dumbser, M., Moschetta, J.-M., & Gressier, J. 2004, J. Comput. Phys., 197, 647
Galassi, M., Davies, J., Theiler, J., et al. 2018, GNU Scientific Library Reference
   Manual. https://www.gnu.org/software/gsl/
Godunov, S. K. 1959, Mat. Sb., 3, 271
Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357
Harten, A., Lax, P. D., & van Leer, B. 1983, SIAM Rev., 25, 35
Huber, D., Kissmann, R., Reimer, A., & Reimer, O. 2021a, A&A, 646, A91
Huber, D., Kissmann, R., & Reimer, O. 2021b, A&A, 649, A71
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Kissmann, R., Kleimann, J., Krebl, B., & Wiengarten, T. 2018, ApJS, 236, 53
Komissarov, S., & Porth, O. 2021, New Astron. Rev., 92, 101610
Kurganov, A., & Tadmor, E. 2000, J. Comput. Phys., 160, 241
Lamberts, A., Fromang, S., Dubus, G., & Teyssier, R. 2013, A&A, 560, A79
Landau, L. D., & Lifshitz, E. M. 1959, Fluid Mechanics (Oxford: Pergamon
   Press)
LeVeque, R. J. 2002, Finite Volume Methods for Hyperbolic Problems,
   Cambridge Texts in Applied Mathematics (Cambridge: Cambridge University
Marti, J. M., & Müller, E. 1994, J. Fluid Mech., 258, 317
Martí, J. M., Müller, E., Font, J. A., Ibáñez, J. M. Z., & Marquina, A. 1997, ApJ,
Martí, J. M., & Müller, E. 2003, Liv. Rev. Relativ., 6, 7
Martí, J. M., & Müller, E. 2015, Liv. Rev. Comput. Astrophys., 1, 3
Mathews, W. G. 1971, ApJ, 165, 147
Meliani, Z., Keppens, R., Casse, F., & Giannios, D. 2007, MNRAS, 376, 1189
Mignone, A., & Bodo, G. 2005, MNRAS, 364, 126
Mignone, A., & McKinney, J. C. 2007, MNRAS, 378, 1118
Mignone, A., Bodo, G., Massaglia, S., et al. 2007, ApJS, 170, 228
Mignone, A., Zanni, C., Tzeferacos, P., et al. 2012, ApJS, 198, 7
Mignone, A., Zanni, C., Vaidya, B., et al. 2020, Pluto User's Guide v4.4, http:
   //plutocode.ph.unito.it
Mościbrodzka, M., Falcke, H., Shiokawa, H., & Gammie, C. F. 2014, A&A, 570,
Mościbrodzka, M., Falcke, H., & Shiokawa, H. 2016, A&A, 586, A38
Olmi, B., & Bucciantini, N. 2019, MNRAS, 484, 5755
Pons, J. A., Ma Martí, J., & Müller, E. 2000, J. Fluid Mech., 422, 125
Quirk, J. J. 1994, Int. J. Numer. Meth. Fluids, 18, 555
Reitberger, K., Kissmann, R., Reimer, A., & Reimer, O. 2017, ApJ, 847, 40
Rezzolla, L. 2013, Relativistic Hydrodynamics (Oxford: Oxford University
   Press)
Rezzolla, L., Zanotti, O., & Pons, J. A. 2003, J. Fluid Mech., 479, 199
Roe, P. L. 1981, in The Use of the Riemann Problem in Finite Difference
   Schemes, eds. W. C. Reynolds, & R. W. MacCormack, 141, 354
Ryu, D., & Goodman, J. 1994, ApJ, 422, 269
```

Scherer, K., Baalmann, L. R., Fichtner, H., et al. 2020, MNRAS, 493,

4172

Schneider, V., Katscher, U., Rischke, D. H., et al. 1993, J. Comput. Phys., 105,

Shu, C.-W. 1988, SIAM J. Sci. Stat. Comput., 9, 1073

Siegel, D. M., & Metzger, B. D. 2018, ApJ, 858, 52

Stone, J. M., & Norman, M. L. 1992, ApJS, 80, 753

Stone, J. M., Gardiner, T. A., Teuben, P., Hawley, J. F., & Simon, J. B. 2008, ApJS, 178, 137

Synge, J. L. 1957, The Relativistic Gas (Amsterdam: North-Holland Pub)

Teyssier, R. 2002, A&A, 385, 337

Toro, E. F. 1997, Riemann Solvers and Numerical Methods for Fluid Dynamics (Berlin, Heidelberg: Springer-Verlag)

van Leer, B. 1977, J. Comput. Phys., 23, 276

van Leer, B. 1979, J. Comput. Phys., 32, 101

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nat. Meth., 17, 261

White, C. J., Stone, J. M., & Gammie, C. F. 2016, ApJS, 225, 22

Willingale, R., & Mészáros, P. 2017, Space Sci. Rev., 207, 63

Wu, K., & Tang, H. 2015, J. Comput. Phys., 298, 539

Zhang, B. 2018, The Physics of Gamma-Ray Bursts (Cambridge: Cambridge Univeristy Press)

Appendix A: Geometrical source terms

The geometrical source terms G_{m^j} arising in Eq. (7) can be obtained using the anholonomic connection symbol Ω_{ik}^{J} in the given coordinate system as

$$G_{m^j} = -\Omega_{ik}^j T^{ik},\tag{A.1}$$

where $T^{ij} = m^i v^j + p \, \delta^{ij}$ denotes the stress-tensor components. The connection symbol corresponds to the anholonomic components of the Christoffel symbol of second kind Γ^j_{ik} (also known as natural connection) in physical units and can be computed by (see Altman & de Oliveira 2011)

$$\Omega_{ik}^{j} = \frac{h_j}{h_i h_k} \Gamma_{ik}^{j} - \frac{h_j}{h_k} \partial_k (1/h_i) \, \delta_i^{j}. \tag{A.2} \label{eq:alpha_ik}$$

This yields the geometrical source terms

$$G_{m^j} = -\sum_{i,k} \frac{h_j}{h_i h_k} \Gamma^j_{ik} T^{ik} + \sum_i \frac{\partial_i h_j}{h_i h_j} T^{ij}. \tag{A.3}$$

Table A.1. Coordinate variables x^i , scale factors h_i , and geometrical source terms G_{m^j} for the different coordinate systems implemented in

	Cartesian	Cylindrical	Spherical
(x^1, x^2, x^3)	(x, y, z)	(r, ϕ, z)	(r, θ, ϕ)
(h_1, h_2, h_3)	(1, 1, 1)	(1, r, 1)	$(1, r, r \sin \theta)$
G_{m^1}	0	$\frac{p+m^{\phi}v^{\phi}}{r}$	$\frac{2p+m^{\theta}v^{\theta}+m^{\phi}v^{\phi}}{r}$
G_{m^2}	0	$\frac{-m^{\phi}v^{r}}{r}$	$\frac{-m^{\theta}v^{r} + (p + m^{\phi}v^{\phi})\cot\theta}{r}$
G_{m^3}	0	0	$\frac{-m^{\phi}(v^r + v^{\theta} \cot \theta)}{r}$

The Christoffel symbol can be readily obtained from the metric tensor $g_{ij} = h_i h_j \delta_{ij}$ and its derivatives via

$$\Gamma_{bc}^{a} = \frac{1}{2} g^{ai} \left(\partial_{b} g_{ci} + \partial_{c} g_{ib} - \partial_{i} g_{bc} \right). \tag{A.4}$$

For orthogonal coordinates (Stone & Norman 1992), the only non-zero entries are given by

$$\Gamma^{i}_{ii} = \frac{\partial_{i}h_{i}}{h_{i}}; \ \Gamma^{i}_{jj} = \frac{h_{j}}{h_{i}^{2}}\partial_{i}h_{j}; \ \Gamma^{i}_{ij} = \Gamma^{i}_{ji} = \frac{\partial_{j}h_{i}}{h_{i}}$$
(A.5)

The resulting source terms for the implemented coordinate systems are summarised in Table A.1. We note that the source terms arising in special relativistic hydrodynamics are formally identical to the ones in Newtonian hydrodynamics. This is intuitive since both sets of equations are also formally identical except for different definitions of the conserved variables.

A similar procedure can also be used to derive the source terms for co-rotating coordinate systems in special relativistic hydrodynamics. This has been done in Huber et al. (2021a) in the context of simulation of gamma-ray binaries and is also available in CRONOS.