

MHDUeT: an HPC code to perform large-eddy-simulations of binary mergers involving neutron stars, using realistic EoS and neutrinos

Carlos Palenzuela^{1,2}

¹*Departament de Física, Universitat de les Illes Balears, Palma de Mallorca, E-07122, Spain and*

²*Institute of Applied Computing & Community Code (IAC3),
Universitat de les Illes Balears, Palma de Mallorca, E-07122, Spain*

Internal notes providing the details of the evolution equations and numerical methods implemented in MHDueT. Use them at your own risk!

CONTENTS

I. Introduction	1	H. Refinement criteria, restriction and prolongation	26
II. Evolution equations	1	I. Sub-cycling in time	26
A. Spacetime	2	Appendix A. Conversion to physical units	27
1. Einstein equations	2	Appendix B. WENO SCHEMES	29
2. CCZ4 formalism	2	1. Third-order WENO	29
3. Gauge conditions	3	2. Fifth-order WENO	29
4. Boundary conditions	3	Appendix C. Berger-Oliger without order reduction (BOR)	30
B. Matter: magnetized fluid	4	3. Standard fourth-order RK	31
C. Neutrino leakage	6	4. Strong Stability Preserving third-order RK	31
D. Neutrinos: truncated moments (M1)	7	5. Dense output interpolator	32
1. Optically thin and thick limits	8	A. Tabulated EOSs	32
2. Minerbo closure	8	1. Table format	32
3. Characteristic structure	9	2. Shen EOS	32
4. Neutrino number current	9	a. Table Construction	32
5. M1 final set of equations	9	b. Chemical Potentials	32
6. Time integration of equation with stiff source	10	c. Energy Shift	32
7. Jacobian	11	3. LS EOSs	32
8. Simplified system	12	a. Chemical Potentials	32
9. Calculating the emission and absorption coefficients	12	b. Energy Shift	33
10. Coupling neutrino matter	13	References	33
11. Neutrino processes	14		
E. LES	14		
III. Numerical implementation	16		
A. EoS	16		
1. Hybrid EoS	16		
2. Microphysical EoS	16		
B. Ejecta	18		
C. Eikonal	19		
D. Numerical methods	20		
1. The Method of Lines	21		
2. The Runge-Kutta time integrator	21		
3. IMEX	22		
4. Spatial Discretization for smooth solutions	22		
5. Spatial Discretization for non-smooth solutions	23		
E. Linear reconstruction for advection-diffusion equations	24		
F. Monotonic Preserving MP5	24		
G. AMR	25		

I. INTRODUCTION

Here we explain the problems we want to address within this code and compare with similar ones in the market.

II. EVOLUTION EQUATIONS

Our previous studies on neutron stars used a magnetized perfect fluid with an hybrid (i.e., a combination of a polytropic and an ideal gas) equation of state. In this paper we detail the recent changes to our code to use finite temperature equations of state and the leakage scheme. However, we also briefly present the Einstein and fluid equations for the sake of completeness and to define our notation.

From a physical point of view, we follow the dynamics of gravitational and magnetohydrodynamical fields, plus the contribution of neutrinos. The MHD equations must be modified to account for the effect of neutrinos in both energy and lepton numbers. Since the leakage scheme is essentially a local calculation (the exception being the optical depth) providing lepton and energy rates of change as measured by a co-moving observer, its extension to the general relativistic case is straightforward. For the sake of completeness we describe next the basic strategy.

The covariant system of equations is

$$G_{ab} = 8\pi(T_{ab} + \mathcal{R}_{ab}) \quad (1)$$

$$\nabla_a T_b^a = \mathcal{G}_b \quad (2)$$

$$\nabla^a (T_{ab} n^b) = 0 \quad (3)$$

$$\nabla_a (Y_e \rho u^a) = \rho R_Y \quad (4)$$

$$\nabla_a {}^* F^{ab} = 0. \quad (5)$$

where the first is Einstein equation coupling the matter and the spacetime and the others state the conservation laws for the stress-energy tensor, matter and lepton number respectively where $Y_e \equiv n_e/n_b$ is the electron fraction, the ratio of electrons to baryons. In the absence of lepton source terms, Eq. (42) follows closely the conservation law for the rest mass density, i.e. Y_e is a mass scalar. The sources \mathcal{G}_a ($\equiv -\nabla_c \mathcal{R}_a^c$) and R_Y are the radiation four-force density and lepton sources, which are determined here via the leakage scheme.

A. Spacetime

Here we write down again the CCZ4 formalism, the gauge conditions (with the damping parameters values).

1. Einstein equations

The Einstein equations in the presence of both matter and radiation are

$$G_{ab} = 8\pi(T_{ab} + \mathcal{R}_{ab}) \quad (6)$$

where T_{ab} is the stress energy tensor of a perfect fluid, \mathcal{R}_{ab} is the contribution from the radiation field, and we have adopted geometrized units where $G = c = M_\odot = 1$. Eq. 6, coupled with appropriate prescriptions for the dynamics of T_{ab} and \mathcal{R}_{ab} , defines the system of equations. In what follows we briefly describe how each is implemented.

We solve the Einstein equations by adopting a 3+1 decomposition in terms of a spacelike foliation. The hypersurfaces that constitute this foliation are labeled by a time coordinate t with unit normal n^a and endowed with spatial coordinates x^i . We express the spacetime metric as

$$ds^2 = -\alpha^2 dt^2 + \gamma_{ij} (dx^i + \beta^i dt) (dx^j + \beta^j dt), \quad (7)$$

where α is the lapse function, β^i the shift vector, γ_{ij} the induced 3-metric on each spatial slice, and $\sqrt{\gamma}$ is the square root of its determinant.

In this work, we use the covariant conformal Z4 formulation of the evolution equations [1, 2]. A summary of the final set of evolution equations for the spacetime fields, together with the gauge conditions setting the choice of coordinates, can be found in [3].

2. CCZ4 formalism

In this section we are going to study the particular case where $p = -1$ and the fields can be rewritten so that the evolution can be reduced exactly to our BSSN implementation when $Z_a = 0$. Let us start first from the general equations, with $p = -1$ and $\psi = \chi$, so that:

$$\gamma_{ij} = \frac{1}{\chi} \tilde{\gamma}_{ij}, \quad \gamma^{ij} = \chi \tilde{\gamma}^{ij}, \quad \chi = \gamma^{-1/3} \rightarrow \tilde{\gamma} = 1 \quad (8)$$

$$\tilde{A}_{ij} = \chi [K_{ij} - \gamma_{ij} \text{tr} K / 3] \rightarrow \text{tr} \tilde{A} = 0 \quad (9)$$

$$\hat{\Gamma}^i = \tilde{\Gamma}^i + \frac{2}{\chi} Z^i, \quad \tilde{\Gamma}^i = \tilde{\gamma}^{ij} \tilde{\gamma}^{kl} \partial_l \tilde{\gamma}_{jk} \rightarrow Z^i = \frac{\chi}{2} (\hat{\Gamma}^i - \tilde{\Gamma}^i)$$

$$\hat{K} \equiv K - 2\Theta \quad (11)$$

The evolution equations in terms of this new fields reads

$$\partial_t \tilde{\gamma}_{ij} = \beta^k \partial_k \tilde{\gamma}_{ij} + \tilde{\gamma}_{ik} \partial_j \beta^k + \tilde{\gamma}_{kj} \partial_i \beta^k - \frac{2}{3} \tilde{\gamma}_{ij} \partial_k \beta^k - 2\alpha \left(\tilde{A}_{ij} - \frac{1}{3} \tilde{\gamma}_{ij} \tilde{A} \right) - \frac{\alpha}{3} \kappa_c \tilde{\gamma}_{ij} \ln \tilde{\gamma} \quad (12)$$

$$\partial_t \tilde{A}_{ij} = \beta^k \partial_k \tilde{A}_{ij} + \tilde{A}_{ik} \partial_j \beta^k + \tilde{A}_{kj} \partial_i \beta^k - \frac{2}{3} \tilde{A}_{ij} \partial_k \beta^k - \frac{\alpha}{3} \kappa_c \tilde{\gamma}_{ij} \tilde{A} \quad (13)$$

$$+ \chi \left[\alpha \left({}^{(3)}R_{ij} + D_i Z_j + D_j Z_i - 8\pi G S_{ij} \right) - D_i D_j \alpha \right]^{\text{TF}} + \alpha \left(\hat{K} \tilde{A}_{ij} - 2\tilde{A}_{ik} \tilde{A}^k_j \right)$$

$$\partial_t \chi = \beta^k \partial_k \chi + \frac{2}{3} \chi \left[\alpha (\hat{K} + 2\Theta) - \partial_k \beta^k \right] \quad (14)$$

$$\partial_t \hat{K} = \beta^k \partial_k \hat{K} - D_i D^i \alpha + \alpha \left[\frac{1}{3} (\hat{K} + 2\Theta)^2 + \tilde{A}_{ij} \tilde{A}^{ij} + 4\pi G (\tau + S) + \kappa_1 (1 - \kappa_2) \Theta \right] + 2Z^i \partial_i \alpha \quad (15)$$

$$\partial_t \Theta = \beta^k \partial_k \Theta + \frac{\alpha}{2} \left[{}^{(3)}R + 2D_i Z^i + \frac{2}{3} \hat{K}^2 + \frac{2}{3} \Theta (\hat{K} - 2\Theta) - \tilde{A}_{ij} \tilde{A}^{ij} \right] - Z^i \partial_i \alpha - \alpha \left[8\pi G \tau + \kappa_1 (2 + \kappa_2) \Theta \right] \quad (16)$$

$$\partial_t \hat{\Gamma}^i = \beta^j \partial_j \hat{\Gamma}^i - \hat{\Gamma}^j \partial_j \beta^i + \frac{2}{3} \hat{\Gamma}^i \partial_j \beta^j + \tilde{\gamma}^{jk} \partial_j \partial_k \beta^i + \frac{1}{3} \tilde{\gamma}^{ij} \partial_j \partial_k \beta^k \quad (17)$$

$$- 2\tilde{A}^{ij} \partial_j \alpha + 2\alpha \left[\tilde{\Gamma}^i_{jk} \tilde{A}^{jk} - \frac{3}{2\chi} \tilde{A}^{ij} \partial_j \chi - \frac{2}{3} \tilde{\gamma}^{ij} \partial_j \hat{K} - 8\pi G \tilde{\gamma}^{ij} S_i \right] \quad (18)$$

$$+ 2\alpha \left[-\tilde{\gamma}^{ij} \left(\frac{1}{3} \partial_j \Theta + \frac{\Theta}{\alpha} \partial_j \alpha \right) - \frac{1}{\chi} Z^i (\kappa_1 + \frac{2}{3} (\hat{K} + 2\Theta)) \right] \quad (19)$$

where the expression $[\dots]^{\text{TF}}$ indicate the trace-less part with respect to the metric $\tilde{\gamma}_{ij}$. Besides, the Ricci terms can be written now as

$${}^{(3)}R_{ij} + 2D_{(i} Z_{j)} = {}^{(3)}\hat{R}_{ij} + \hat{R}_{ij}^{\chi} \quad (20)$$

$$\chi \hat{R}_{ij}^{\chi} = \frac{1}{2} \partial_i \partial_j \chi - \frac{1}{2} \tilde{\Gamma}_{ij}^k \partial_k \chi - \frac{1}{4\chi} \partial_i \chi \partial_j \chi + \frac{2}{\chi} Z^k \tilde{\gamma}_{k(i} \partial_{j)} \chi + \frac{1}{2} \tilde{\gamma}_{ij} \left[\tilde{\gamma}^{km} \left(\partial_k \partial_m \chi - \frac{3}{2\chi} \partial_k \chi \partial_m \chi \right) - \hat{\Gamma}^k \partial_k \chi \right] \quad (21)$$

$$\hat{R}_{ij} = -\frac{1}{2} \tilde{\gamma}^{mn} \partial_m \partial_n \tilde{\gamma}_{ij} + \tilde{\gamma}_{k(i} \partial_{j)} \hat{\Gamma}^k + \hat{\Gamma}^k \tilde{\Gamma}_{(ij)k} + \tilde{\gamma}^{mn} \left(\tilde{\Gamma}_{mi}^k \tilde{\Gamma}_{jkn} + \tilde{\Gamma}_{mj}^k \tilde{\Gamma}_{ikn} + \tilde{\Gamma}_{mi}^k \tilde{\Gamma}_{knj} \right) \quad (22)$$

and the Laplacian of the lapse is just

$$D_i D^i \alpha = \chi \tilde{\gamma}^{ij} D_i D_j \alpha = \chi \tilde{\gamma}^{ij} \partial_i \partial_j \alpha - \chi \tilde{\Gamma}^k \partial_k \alpha - \frac{1}{2} \tilde{\gamma}^{ij} \partial_i \alpha \partial_j \alpha \quad (23)$$

The matter terms can be written in terms of the T^{ab} and the conformal metric, namely

$$\tau = n_a n_b T^{ab} = \alpha^2 T^{00}$$

$$S_i = n_a (g_{bi} - n_b n_i) T^{ab} = \alpha g_{bi} T^{0b} \quad ,$$

$$S_i = \frac{\tilde{S}_i}{\chi} \quad , \quad \tilde{S}_i = \alpha \tilde{\gamma}_{ik} (T^{0k} + \beta^k T^{00})$$

$$S_{ij} = g_{ai} g_{bj} T^{ab} \quad , \quad S_{ij} = \frac{\tilde{S}_{ij}}{\chi^2} \quad ,$$

$$\tilde{S}_{ij} = \tilde{\beta}_i \tilde{\beta}_j T^{00} + \left(\tilde{\gamma}_{ik} \beta_j + \tilde{\gamma}_{jk} \beta_i \right) T^{0k} + \tilde{\gamma}_{ik} \tilde{\gamma}_{jm} T^{km} \quad ,$$

$$S = \gamma^{ij} S_{ij} = \frac{\tilde{S}}{\chi} \quad .$$

where the shift indexes has been lowered with the conformal metric $\tilde{\beta}_i \equiv \tilde{\gamma}_{ij} \beta^j$.

3. Gauge conditions

We are using 1+log slicing condition with a simplified version of the Gamma-freezing shift condition, namely

$$\partial_t \alpha = \lambda_1 \beta^i \partial_i \alpha - 2\alpha f_\alpha(\alpha) (\hat{K} - \hat{K}_0) \quad (24)$$

$$\partial_t \beta^i = \lambda_2 \beta^j \partial_j \beta^i + \frac{3}{4} f_\beta(\alpha) \hat{\Gamma}^i - \eta (\beta^i - \beta_0^i) \quad (25)$$

where η is a parameter scaling with the total mass $\approx 2/M$ and λ_i can be either 0 or 1. The gauge functions $f(\alpha)$ depend of α and can be freely chosen, although we write them in the code as $f_\beta(\alpha) = f_0 + f_1 \alpha$ and $f_\alpha(\alpha) = f_2 + f_3 \alpha$. The standard gauge conditions that we use have $\lambda_1 = \lambda_2 = f_0 = f_2 = 1$ and $f_1 = f_3 = \hat{K}_0 = \beta_0^i = 0$. The Z-damping also depend on the total mass, and we usually set them as $\kappa_1 \leq 1/M$ and $\kappa_2 = 0$.

4. Boundary conditions

The main part of the boundary conditions assumes that there is an outgoing radial wave with some speed v_0 ,

$$X = X_0 + \frac{u(r - v_0 t)}{r} \quad (26)$$

where X is any of the tensor components of the evolved variables, X_0 its value at infinity and u a spherically symmetric perturbation. Notice that $\{X_0, v_0\}$ depend on the

particular variable. The time derivative can be written as

$$\partial_t X = -v^i \partial_i X - v_0 \frac{X - X_0}{r} \quad (27)$$

where $v^i = v_0 x^i / r$ and ∂_i are evaluated using centered finite differencing where possible and one-sided finite differencing elsewhere.

A similar boundary condition is provided by the original Sommerfeld boundary conditions, that for a field X can be written in general as

$$\partial_t X = \beta^j \partial_j X - v_0 \left(s^i \partial_i X + \frac{X}{r} \right) \quad (28)$$

where v_0 is the propagation velocity of the eigenmode and s^i is the normal vector of the corresponding boundary face. We usually take instead the radial vector $s^i = x^i / r$. Applied to the Einstein equations, these conditions imply

$$\partial_t \hat{K} = \beta^k \partial_k \hat{K} - \sqrt{\frac{2f_\alpha}{\alpha}} \left(s^k \partial_k \hat{K} + \frac{\hat{K}}{r} \right) \quad (29)$$

$$\partial_t \hat{\Gamma}^i = \beta^k \partial_k \hat{\Gamma}^i - \sqrt{f_\beta} \left(s^k \partial_k \hat{\Gamma}^i + \frac{\hat{\Gamma}^i}{r} \right) \quad (30)$$

$$\partial_t \Theta = \beta^k \partial_k \Theta - \alpha \left(s^i \partial_i \Theta + \frac{\Theta}{r} \right) \quad (31)$$

$$\partial_t \tilde{A}_{ij} = \beta^k \partial_k \tilde{A}_{ij} - \alpha \left(s^i \partial_i \tilde{A}_{ij} + \frac{\tilde{A}_{ij}}{r} \right) \quad (32)$$

Notice that, within this approach, the spacetime metric fields $\{\alpha, \beta^i, \chi, \tilde{\gamma}_{ij}\}$ are evolved at the boundaries with the same evolution equations than in the interior.

B. Matter: magnetized fluid

Equations for the perfect magnetized fluid stress-energy tensor. 3+1 evolution equations for the magnetized fluid. Theoretical details on the EoS.

Let us assume that the perfect fluid and the electromagnetic fields are minimally coupled, that is, that there are no mixed terms and so the total stress-energy tensor can be written just by addition

$$T_{\mu\nu} = T_{\mu\nu}^{fluid} + T_{\mu\nu}^{em} \quad (33)$$

so explicitly it can be written as

$$T_{\mu\nu} = [\rho(1 + \epsilon) + p] u_\mu u_\nu + p g_{\mu\nu} + F_\mu^\lambda F_{\nu\lambda} - \frac{1}{4} g_{\mu\nu} F^{\lambda\alpha} F_{\lambda\alpha} \quad (34)$$

The rest mass density ρ is the density of the fluid measured by a comoving observer. The total energy density of the fluid also contains contributions from the internal degrees of freedom of the particles,

$$e = \rho(1 + \epsilon) \quad (35)$$

where the internal energy ϵ accounts, for instance, for the thermal energy, the binding energy,... The pressure p is described by an Equation of State as a function of the rest mass density and the internal energy, and it is a property of the type of fluid which is being considered. Within the pressure we can construct the enthalpy h ,

$$h = \rho(1 + \epsilon) + p \quad (36)$$

It is important to stress that the set of thermodynamic quantities $\{\rho, \epsilon, P\}$ are all measured in the rest frame of the fluid element, although in general we will use an Eulerian perspective where the coordinates are not tied to the flow of the fluid. Therefore, we will need the four-velocity u^μ to describe how the fluid moves with respect to the Eulerian observers. The 4-velocity follows the usual normalization relation

$$u^\mu u_\mu = -1 \quad (37)$$

It is more useful to deal with the standard velocity vectors, so the velocity u^μ will be decomposed into spatial and temporal components, namely

$$u^\mu = W n^\mu + W v^\mu \quad (38)$$

where v^μ corresponds to the familiar three-dimensional quantities as measured by Eulerian observers (ie, $v^\mu n_\mu = 0$). Notice that the time component is not independent due to the normalization relation (37), so

$$W = -n_\mu u^\mu = (1 - v_i v^i)^{-1/2}, \quad u^i = W(v^i - \frac{\beta^i}{\alpha}) \quad (39)$$

where we can recognize now that W is the standard Lorentz factor. The set of fluid variables $U = (\rho, \epsilon, p, v^i)$ are the primitive quantities which describe the state of a perfect fluid.

The evolution of matter must comply with the conservation of the stress-energy tensor

$$\nabla_\nu T^{\mu\nu} = 0, \quad (40)$$

which can be expressed as a system of conservation laws for the energy and momentum densities. In addition to the conservation of energy and momentum, when there are neither creation nor destruction of particles, the fluids has to conserve also the total number of baryons. This law is expressed in terms of the baryon number density ρu^μ , and is written as

$$\nabla_\mu (\rho u^\mu) = 0 \quad (41)$$

which is just the relativistic generalization of the conservation of mass. A similar conservation law is followed by the lepton number, with a source term allowing for production/annihilation mechanisms

$$\nabla_a (Y_e \rho u^a) = \rho R_Y \quad (42)$$

where we remind that

$$Y_e = \frac{n_p}{n_p + n_n} \quad (43)$$

being n_p and n_n the proton and neutron number densities, respectively. The net electron number density $n_{e^-} - n_{e^+} = n_p$ due to charge neutrality in the fluid.

Then, the primitive quantities U has to be computed from the conservation of energy, momentum and baryonic number. The EOS for the pressure close the system of equations.

In order to capture properly the weak solutions of the non-linear equations in the presence of shocks it is important to write them in local conservation law form. The GRMHD equations for a magnetized, non-viscous and perfectly conducting fluid [4] provide a set of evolution equations for the conserved variables

$\{\sqrt{\gamma}D, \sqrt{\gamma}S^i, \sqrt{\gamma}U, \sqrt{\gamma}B^i\}$. These conserved fields are functions of the rest-mass density ρ , the specific internal energy ϵ , the velocity vector v^i and the magnetic field B^i (primitive fields), namely:

$$D = \rho W \quad (44)$$

$$D_Y = \rho W Y_e \quad (45)$$

$$S_i = (hW^2 + B^2)v_i - (B^k v_k)B_i \quad (46)$$

$$\tau = hW^2 - p + B^2 - \frac{1}{2} \left[(B^k v_k)^2 + \frac{B^2}{W^2} \right] - D \quad (47)$$

The evolution equations for these conserved fields can be written as

$$\partial_t(\sqrt{\gamma}D) + \partial_k[\sqrt{\gamma}(-\beta^k + \alpha v^k)D] = 0 \quad (48)$$

$$\partial_t(\sqrt{\gamma}D_Y) + \partial_k[\sqrt{\gamma}(-\beta^k + \alpha v^k)D_Y] = \frac{\alpha}{W} \sqrt{\gamma} D R_Y \quad (49)$$

$$\partial_t(\sqrt{\gamma}\tau) + \partial_k[\sqrt{\gamma}(-\beta^k \tau + \alpha(S^k - Dv^k))] = \sqrt{\gamma}[\alpha S^{ij} K_{ij} - S^j \partial_j \alpha] \quad (50)$$

$$\partial_t(\sqrt{\gamma}S_i) + \partial_k[\sqrt{\gamma}(-\beta^k S_i + \alpha S^k{}_i)] = \sqrt{\gamma}[\alpha \Gamma^j{}_{ik} S^k_j + S_j \partial_i \beta^j - (\tau + D) \partial_i \alpha]$$

$$\partial_t(\sqrt{\gamma}B^i) + \partial_k[\sqrt{\gamma}\{B^i(\alpha v^k - \beta^k) - B^k(\alpha v^i - \beta^i) + \alpha \gamma^{ki} \phi\}] = \sqrt{\gamma} \phi [\gamma^{ik} \partial_k \alpha - \alpha \gamma^{jk} \Gamma^i{}_{jk}]$$

$$\partial_t(\sqrt{\gamma}\phi) + \partial_k[\sqrt{\gamma}(-\beta^k \phi + \alpha c_h^2 B^k)] = \sqrt{\gamma}[c_h^2 B^k \partial_k \alpha - \alpha \phi \text{tr} K - \alpha \kappa \phi] \quad (51)$$

where the fluxes are

$$S_{ij} = \frac{1}{2} (v_i S_j + v_j S_i) + \gamma_{ij} p - \frac{1}{2W^2} \left[2B_i B_j - \gamma_{ij} B^2 \right] \\ - \frac{1}{2} (B^k v_k) \left[B_i v_j + B_j v_i - \gamma_{ij} (B^m v_m) \right]$$

Since the fluxes depend also on the primitive fields \mathbf{u} , one needs to calculate them before computing the rhs by using the definitions and the Equation of State $p = p(\rho, Y_e, \epsilon)$, as detailed in Sec. III A.

Notice that for the Einstein equations we need to construct the stress-energy tensor of a perfect fluid from these quantities

$$T^{00} = \frac{1}{\alpha^2} (D + \tau) \quad (52)$$

$$T^{0i} = \frac{S^i}{\alpha} - \beta^i T^{00} \quad (53)$$

$$T^{ij} = \gamma^{ik} \gamma^{jm} S_{km} - S^i \frac{\beta^j}{\alpha} - \beta^i T^{0j} \quad (54)$$

Let us define the densitized conserved quantities as

$$\bar{D} = \sqrt{\gamma} D, \quad \bar{D} Y_e = \sqrt{\gamma} D Y_e, \quad \bar{S}_i = \sqrt{\gamma} S_i, \quad (55)$$

$$\bar{\tau} = \sqrt{\gamma} \tau, \quad \bar{B}^i = \sqrt{\gamma} B^i, \quad \bar{\phi} = \sqrt{\gamma} \phi \quad (56)$$

The evolution equations, in terms of the BSSN quantities and the densitized conserved fields, can be written as

$$\begin{aligned}
\partial_t \bar{D} + \partial_k [(-\beta^k + \alpha v^k) \bar{D}] &= 0 \\
\partial_t \bar{D} Y_e + \partial_k [(-\beta^k + \alpha v^k) \bar{D} Y_e] &= \frac{\alpha}{W} \bar{D} R_Y \\
\partial_t \bar{\tau} + \partial_k [-\beta^k \bar{\tau} + \alpha (\bar{S}^k - \bar{D} v^k)] &= \frac{\alpha}{\chi} \bar{S}^{ij} \tilde{A}_{ij} + \frac{\alpha}{3} \text{tr} \bar{S} \text{tr} K - \bar{S}^j \partial_j \alpha \\
\partial_t \bar{S}_i + \partial_k [-\beta^k \bar{S}_i + \alpha \bar{S}_i^k] &= \frac{\alpha}{2\chi} (\bar{S}^{jk} \partial_i \tilde{\gamma}_{jk} - \text{tr} \bar{S} \partial_i \chi) + \bar{S}_j \partial_i \beta^j - (\bar{\tau} + \bar{D}) \partial_i \alpha \\
\partial_t \bar{B}^i + \partial_k [\bar{B}^i (\alpha v^k - \beta^k) - \bar{B}^k (\alpha v^i - \beta^i) + \alpha \chi \tilde{\gamma}^{ki} \bar{\phi}] &= \bar{\phi} [-\alpha \chi \tilde{\Gamma}^i + \tilde{\gamma}^{ki} (-\frac{\alpha}{2} \partial_k \chi + \chi \partial_k \alpha)] \\
\partial_t \bar{\phi} + \partial_k [-\beta^k \bar{\phi} + \alpha c_h^2 \bar{B}^k] &= -\alpha \bar{\phi} \text{tr} K + c_h^2 \bar{B}^k (\partial_k \alpha) - \alpha \kappa \bar{\phi}
\end{aligned} \tag{57}$$

where $\text{tr} \bar{S} = \gamma_{jk} \bar{S}^{jk} = \tilde{\gamma}_{jk} \bar{S}^{jk} / \chi$.

C. Neutrino leakage

The leakage scheme provides the fluid rest frame energy sink \mathcal{Q} and lepton sink/source R_Y due to neutrino processes. R_Y is the source term for a scalar quantity and therefore is the same in all frames. We express the source term for the energy and momentum in an arbitrary frame as

$$\mathcal{G}_a = \mathcal{Q} u_a. \tag{58}$$

Since the effect of neutrino pressure is small [5] and difficult to accurately capture with a neutrino leakage scheme, we ignore its contribution in the fluid rest frame. Now, defining $\mathcal{H} \equiv n^a \mathcal{G}_a = -\mathcal{Q} W$ and $\mathcal{H}_b \equiv h_{bc} \mathcal{G}^c = \mathcal{Q} W v_b$, the modified relativistic MHD equations become

$$\partial_t (\sqrt{\gamma} D_Y) + \dots = \alpha \sqrt{\gamma} \rho R_Y \tag{59}$$

$$\partial_t (\sqrt{\gamma} \tau) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{Q} W \tag{60}$$

$$\partial_t (\sqrt{\gamma} S_i) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{Q} W v_i \tag{61}$$

Neutrino interaction rates depend sensitively on the matter temperature and composition. Therefore, in order to model the effect of neutrinos with reasonable accuracy, we require an equation of state beyond that of a polytrope or an ideal gas. We use publicly available EoS tables from www.stellarcollapse.org and described in O'Connor and Ott (2010) [6]. We have rewritten some of the library routines for searching the table to make them faster and more robust. In this paper we use the Shen-Horowitz-Teige (SHT) [7] EoS with the NL3 relativistic mean-field parametrization, the Lattimer-Swesty (LS) [8] EoS with $K = 220$ MeV, and the H. Shen (HS) [9] for the single neutron star simulations, and the HS EoS for the neutron star binary.

The leakage scheme seeks to account for (i) the changes to the (electron) lepton number and (ii) the loss of energy from the emission of neutrinos. As discussed, since the dynamical timescale for the post-merger of binary neutron star systems is relatively short, radiation momentum transport and diffusion effects are expected to be subleading. Our scheme is based on the open-source neutrino leakage scheme from [6] and available

at www.stellarcollapse.org. At low optical depths, the leakage scheme relies on calculating the emission rate of energy (Q_{free}) and lepton number (R_{free}) directly from the rates of relevant processes. We consider three species of neutrinos, represented here by: ν_e for electron neutrinos, $\bar{\nu}_e$ for electron antineutrinos, and ν_x for both tau and muon neutrinos and their respective antineutrinos. As discussed in [10, 11], the dominant processes are those that

- produce electron flavour neutrinos and antineutrinos: charged-current, electron and positron capture reactions
 $e^+ + n \rightarrow p + \bar{\nu}_e$, $e^- + p \rightarrow n + \nu_e$.
- produce all flavours of neutrinos: electron-positron pair-annihilation
 $e^+ + e^- \rightarrow \bar{\nu}_i + \nu_i$
 and plasmon decay
 $\gamma \rightarrow \bar{\nu}_i + \nu_i$.

Notice that nucleon-nucleon bremsstrahlung can also be an important source of ν_x neutrinos, dominating over electron-positron annihilation at low temperatures and high densities. We will include such a process in future work.

At high optical depths on the other hand, because the equilibrium time scales are much shorter than either neutrino diffusion or hydrodynamic time scales, neutrinos are assumed to be at their equilibrium abundances and the rates of energy loss (Q_{diff}) and lepton loss (R_{diff}) are taken to proceed at the diffusion timescale. The equilibrium abundances can be trivially calculated, however the calculation of the diffusion timescale is more involved as it requires the knowledge of non-local optical depths. The computation of these optical depths lies at the core of the leakage strategy and, because we are interested in general (non-spherically symmetric) scenarios, we describe how to compute them from the local opacities in section III C. (We refer the reader to [6] for full details about the calculation of the local opacity and diffusion time scale.) The emission rates are then interpolated between the behavior at low and high optical depths in order to achieve an efficient way to incorporate neutrino effects that is correct in both regimes and applicable in between. In

our implementation, we interpolate the energy and lepton number emission rates between these two regimes via the following formula

$$X_{\text{eff}} = \frac{X_{\text{diff}} X_{\text{free}}}{X_{\text{diff}} + X_{\text{free}}}, \quad (62)$$

where X is either Q or R .

D. Neutrinos: truncated moments (M1)

A radiation transport scheme needs to evolve the distribution function $f_{(\nu)}(x^a, p^a)$ of neutrinos or photons, where $x^a = (t, x^i)$ are the spacetime coordinates and $p^a = dx^a/d\lambda$ are the components of the 4-momentum of the neutrinos/photons, with λ some affine parameter. Here we neglect neutrino masses, and the 4-momentum is thus a null vector, i.e. $p^a p_a = 0$. The distribution function of photons and of each species of neutrinos evolves according to Boltzmann's equation

$$p^a \left[\frac{\partial f_{(\nu)}}{\partial x^a} - \Gamma_{ac}^b p^c \frac{\partial f_{(\nu)}}{\partial p^b} \right] = \left[\frac{\partial f_{(\nu)}}{\partial \tau} \right]_{\text{coll}} \quad (63)$$

where Γ_{bc}^a are the Christoffel symbols, and the right-hand side includes all collisional processes (i.e., emissions, absorptions, scatterings). Solving Boltzmann's equation thus requires the evolution in time of a 6-dimensional function, a very steep computational challenge. The main objective of this work is to provide a relatively cheap algorithm for general relativistic radiation transport which, while approximate for the numerical resolutions that we can currently afford, asymptotes to Boltzmann's equation in the limit of infinite computational resources. While our algorithm can theoretically be used in various problems involving general relativistic photon or neutrino transport, we implement and test it with the problem of neutrino transport in merger simulations in mind.

Different approximations to the Boltzmann equation have thus been developed for numerical applications. First, let us consider the moment formalism developed by Thorne, in which only the lowest moments of the distribution function in momentum space are evolved. We use this formalism in the *gray* approximation, that is we only consider energy-integrated moments. Although the moment formalism can in theory be used with a discretization in neutrino energies, this makes the simulations significantly more expensive and involves additional technical difficulties in the treatment of the gravitational and velocity redshifts, particularly for applications such as compact binary mergers in which we have both relativistic speeds and large gravitational redshifts. We consider three independent neutrino species: the electron neutrinos ν_e , the electron antineutrinos $\bar{\nu}_e$, and the heavy-lepton neutrinos ν_x . The latter is the combination of 4 species ($\nu_\mu, \bar{\nu}_\mu, \nu_\tau, \bar{\nu}_\tau$). This merging is justified because the temperatures and neutrino energies reached in

our merger calculations are low enough to suppress the formation of the corresponding heavy leptons whose presence would require including the charged current neutrino interactions that differentiate between these individual species.

In the gray approximation, and considering only the first two moments of the distribution function, we evolve for each species projections of the stress-energy tensor of the neutrino radiation T_{rad}^{ab} . One possible decomposition of this tensor is given by

$$T_{rad}^{ab} = J u^a u^b + H^a u^b + H^b u^a + Q^{ab} \quad (64)$$

with $H^a u_a = Q^{ab} u_b = 0$ and u^a the 4-velocity of the fluid. The energy J , flux H^a and stress tensor Q^{ab} of the neutrino radiation as observed by an observer comoving with the fluid are related to the neutrino distribution function by

$$J = \int_0^\infty d\nu \nu^3 \int d\Omega f_{(\nu)}(x^a, \nu, \Omega) \quad (65)$$

$$H^a = \int_0^\infty d\nu \nu^3 \int d\Omega f_{(\nu)}(x^a, \nu, \Omega) l^a \quad (66)$$

$$Q^{ab} = \int_0^\infty d\nu \nu^3 \int d\Omega f_{(\nu)}(x^a, \nu, \Omega) l^a l^b \quad (67)$$

where ν is the neutrino energy in the fluid frame, $\int d\Omega$ denotes integrals over solid angle on a unit sphere in momentum space, and $p^a = \nu(u^a + l^a)$ with $l^a u_a = 0$ and $l^a l_a = 1$. We also consider the decomposition of T_{rad}^{ab} in terms of the energy, flux and stress tensor observed by an inertial observer,

$$T_{rad}^{ab} = E n^a n^b + F^a n^b + F^b n^a + P^{ab} \quad (68)$$

with $F^a n_a = P^{ab} n_b = 0$ (i.e., $F^t = P^{ta} = 0$) and n^a the unit normal to the hypersurfaces $t = \text{constant}$.

Conservation of energy and angular momentum implies

$$\nabla_b T^{ab} = -\nabla_b T_{MHD}^{ab} \quad (69)$$

which in the 3+1 decomposition can be written as

$$\partial_t(\sqrt{\gamma}E) + \partial_i [\sqrt{\gamma}(\alpha F^i - \beta^i E)] = \quad (70)$$

$$\alpha \sqrt{\gamma} [P^{ik} K_{ij} - F^i (\partial_i \alpha) / \alpha - \mathcal{S}^a n_a]$$

$$\partial_t(\sqrt{\gamma}F_i) + \partial_k [\sqrt{\gamma}(\alpha P^k{}_i - \beta^k F_i)] = \quad (71)$$

$$\sqrt{\gamma} \left[-E \partial_i \alpha + F_k \partial_i \beta^k + \frac{\alpha}{2} P^{jk} \partial_i \gamma_{jk} + \alpha \mathcal{S}^a \gamma_{ia} \right]$$

where \mathcal{S}^a is the term representing the interaction between the neutrino radiation and the fluid. This interaction can be written as

$$\mathcal{S}^a = (\eta - \kappa_a J) u^a - (\kappa_a + \kappa_s) H^a \quad (72)$$

where $(\eta, \kappa_a, \kappa_s)$ are respectively the (energy-averaged) neutrino emissivity, absorption opacity and scattering opacity, to be calculated depending on the fluid state and the EoS table information. Scattering is assumed

to be isotropic and elastic. Inelastic scattering could in principle be treated within this formalist as absorption events immediately followed by emission.

It is important to remark that these equations are exact but they are not in a closed form, since P^{ik} can not be expressed in terms of (E, F^i) . The key idea of the M1 scheme is to introduce an approximate analytic closure for these equations, namely a relation $P^{ik} = P^{ik}(E, F^i)$. If such a relation could be found exactly P^{ik} , the M1 scheme would provide a exact solution of the transport equation. However, as it depends on the global geometry of the radiation field, such a closer can not be exact in general. The M1 scheme usually adopts the so-called *minerbo* closure, which is exact in both the optically thick limit (with matter and radiation in thermodynamic equilibrium) and in the propagation of radiation in a transparent medium from a single point source.

1. Optically thin and thick limits

In the optically thin limit we assume that radiation is streaming at the speed of light in the direction of the radiation flux, namely

$$P_{ij}^{thin} = \frac{F_i F_j}{F^k F_k} E \quad (73)$$

which is a explicit relation.

In the optically thick limit we consider that

$$Q_{thick}^{ab} = \frac{J_{thick}}{3} (g^{ab} + u^a u^b) \quad (74)$$

where

$$J_{thick} = \frac{3}{2W^2 + 1} [(2W^2 - 1)E - 2W^2 F^k v_k] \quad (75)$$

$$\gamma_b^a H_{thick}^b = \frac{F^a}{W} + \frac{W v^a}{2W^2 + 1} [(4W^2 + 1)v_k F^k - 4W^2 E]$$

$$H_{thick}^n = -H^a n_a = W(E - J_{thick} - F^k v_k) \quad (76)$$

$$\begin{aligned} H_{thick}^i &= \gamma_b^i H_{thick}^b = \frac{F^i}{W} - \frac{4}{3} J_{thick} W v^i - H_{thick}^n v^i \\ &= \frac{F^i}{W} + \frac{W v^i}{2W^2 + 1} [(4W^2 + 1)F^k v_k - 4W^2 E] \end{aligned} \quad (77)$$

$$\begin{aligned} P_{ij}^{thick} &= \frac{4}{3} J_{thick} W^2 v_i v_j \\ &+ W(H_i^{thick} v_j + H_j^{thick} v_i) + \frac{1}{3} J_{thick} \gamma_{ij} \end{aligned} \quad (78)$$

2. Minerbo closure

We combine both limits to construct the Minerbo closure

$$P_{ij} = \frac{3\chi - 1}{2} P_{ij}^{thin} + \frac{3(1 - \chi)}{2} P_{ij}^{thick} \quad (79)$$

where $\chi \in [\frac{1}{3}, 1]$ is the Eddington factor

$$\chi(\xi) = \frac{1}{3} + \xi^2 \left(\frac{6 - 2\xi + 6\xi^2}{15} \right) \quad , \quad \xi^2 = \frac{H_a H^a}{J^2} \quad (80)$$

where with $H_a H^a = (g^{ab} + n^a n^b - n^a n^b) H_a H_b = H_k H^k - H_n^2$. In the optically thick regions the flow $H^a \approx 0$ (i.e., $\xi \approx 0$ and $\chi \approx 1/3$), so $P_{ij} \approx P_{ij}^{thick}$. On the other hand, in the optically thin regions $H^a \approx J$ (i.e., $\xi \approx 1$ and $\chi \approx 1$), so $P_{ij} \approx P_{ij}^{thin}$.

Generic relations between the quantities in fluid and observer frames are:

$$P_{ab} = W^2 v_a v_b J + W(g_{ac} - n_a v_c) v_b H^c \quad (81)$$

$$+ W(g_{bc} - n_c v_b) v_a H^c + (g_{ac} - n_a v_c)(g_{bd} - n_b v_d) Q^{cd}$$

$$J = EW^2 - 2W^2 F^i v_i + W^2 P^{ij} v_i v_j \quad (82)$$

$$H^a = W(E - F^k v_k) h_b^a n^b + W h_b^a F^b - W h_b^a v_j P^{ij} \quad (83)$$

(remember that $F^a n_a = P^{ab} n_b = F^0 = P^{0b} = 0$ and $h_{ab} = g_{ab} + u_a u_b$ and $\gamma_{ab} = g_{ab} + n_a n_b$, such that $h_b^a n^b = n^a - W u^a$ and $n_a h^{ab} \gamma_{bi} = -W u_i$. We can combine these relations as follows

$$-JW + H^a n_a = -EW + W F^k v_k \quad (84)$$

$$JW v_i + H_i = W F_i - W P_i^k v_k \quad (85)$$

such that it is possible to reconstruct $H^a = -(H^b n_b) n^a + \gamma_k^a H^k$, with $-H^a n_a = H^n$ and $\gamma_{ai} H^a = \gamma_{ai} \gamma_k^a H^k = H_i$.

Therefore, the idea is that at each point (i.e., after the con2prim for the fluid fields) one can compute P_{ij} using the Minerbo closure (79) and the evolved fields (E, F^i) . However, since $\chi(\xi(J, H^a))$ as described by equation (80), and those fields depend also P_{ij} through equations (84,85), we obtain an transcendental equation which needs to be solved numerically. The function to be solved in this case is

$$R = \frac{\xi^2 J^2 - H_a H^a}{E^2} = 0 \quad (86)$$

and we need to take derivatives, either analytical or numerical, of this function. We perform the root finding by adopting the Brent-Dekker method as implemented in the gnu scientific library, which uses a combination of secant and bisection and does not require the analytical knowledge of the derivatives. The procedure to calculate this function would be

1. compute $(P_{ij}^{thin}, P_{ij}^{thick})$ following eqs(73-78).
2. Begin Iteration: use as a guess for χ the previous step to compute the total P_{ij} using equation (79).
3. compute the components of H^a following these relations

$$J = W^2(E - 2F^k v_k + P^{ij} v_i v_j) \quad (87)$$

$$H_n = -H^a n_a = W(E - J - F^k v_k) \quad (88)$$

$$H_i = W(F_i - J v_i - P_i^k v_k) \quad (89)$$

4. compute $H_a H^a = H_k H^k - H_n^2$ and (χ, ξ) using eq.(80). closure
5. check if the $R = 0$ as given by equation (86). Otherwise, use this values to compute a new guess for χ using the Brent-Dekker method and iterate until convergence.

$$\lambda_i = \frac{3\chi - 1}{2} \lambda_{thin}^i + \frac{3(1 - \chi)}{2} \lambda_{thick}^i \quad (90)$$

3. Characteristic structure

The velocities can be calculated both in the thin and thick regimes, and then interpolated using the Minerbo

where

$$\begin{aligned} \lambda_{thin}^i &= |\beta^i| + \alpha \max \left[\frac{|F^i|}{\sqrt{F_k F^k}}, \frac{E |F^i|}{F_k F^k} \right] \\ \lambda_{thick}^i &= |\beta^i| + \alpha \max \left[\frac{2W^2 |v^i| + \sqrt{(2W^2 + 1)\gamma^{ii} - 2W^2 v^i v^i}}{2W^2 + 1}, |v^i| \right]. \end{aligned} \quad (91)$$

4. Neutrino number current

Although weak reactions conserve the total lepton number of the system, they can alter the electron fraction of the matter. For this reason, it is desirable to also evolve the number density of neutrinos. To this aim, we follow a phenomenological approach proposed by Foucart, where for each neutrino species we introduce a neutrino number current $N_{(\nu)}^a$ with $\nu = (\nu_e, \bar{\nu}_e, \nu_x)$, following a conservation equation

$$\nabla_a N^a = \sqrt{-g} \mathcal{C} = \sqrt{-g} (\eta^0 - \kappa_a^0 n) \quad (92)$$

where $n = -N^a u_a$ is the neutrino density in the fluid frame and (κ_a^0, η^0) are the neutrino number absorption and emission coefficients, also to be computed from the fluid state and the information in the EoS tables.

Assuming that the neutrino number density and the radiation flux are aligned (i.e., which is a reasonable assumption but not true in general), one can define the following closure relation

$$N^a = n f^a = n \left(u^a + \frac{H^a}{J} \right). \quad (93)$$

The conservation equation for the neutrino number current, within the 3+1 decomposition, can be written as

$$\partial_t(\sqrt{\gamma} n \Gamma) + \partial_i(\alpha \sqrt{\gamma} n f^i) = \alpha \sqrt{\gamma} (\eta^0 - \kappa_a^0 n) \quad (94)$$

where

$$\Gamma = \alpha f^0 = W - \frac{1}{J} H^a n_a = W \left(\frac{E - F_a v^a}{J} \right), \quad (95)$$

$$f^i = W \left(v^i - \frac{\beta^i}{J} \right) + \frac{H^i}{J} \quad (96)$$

We can set $N = -n_a N^a = \alpha n f^0 = n \Gamma$. With the N0+M1 approach one can compute the average energy of the neutrinos ϵ_ν , since in the fluid frame approximately $J \approx n \langle \epsilon_\nu \rangle$, that is

$$\langle \epsilon_\nu \rangle = \frac{J}{n} = \frac{\Gamma J}{N} = \frac{W(E - F^i v_i)}{N} \quad (97)$$

which gives a way to calculate an average energy that varies with space and time.

5. M1 final set of equations

Let us define the densitized conserved quantities as

$$\bar{E} = \sqrt{\gamma} E, \quad \bar{F}_i = \sqrt{\gamma} F_i, \quad \bar{N} = \sqrt{\gamma} n \Gamma = \sqrt{\gamma} N, \quad (98)$$

The evolution equations, in terms of the BSSN quantities and the densitized conserved fields, can be written as

$$\partial_t \bar{N} + \partial_k [(-W\beta^k + \alpha Wv^k + \alpha \frac{H^k}{J}) \frac{\bar{N}}{\Gamma}] = \alpha \sqrt{\gamma} \mathcal{C} \quad (99)$$

$$\partial_t \bar{E} + \partial_k [-\beta^k \bar{E} + \alpha \bar{F}^k] = \frac{\alpha}{\chi} \bar{P}^{ij} \tilde{A}_{ij} + \frac{\alpha}{3} \text{tr} \bar{P} \text{tr} K - \bar{F}^j \partial_j \alpha + \alpha \sqrt{\gamma} \mathcal{S}_n \quad (100)$$

$$\partial_t \bar{F}_i + \partial_k [-\beta^k \bar{F}_i + \alpha \bar{P}_i^k] = \frac{\alpha}{2\chi} (\bar{P}^{jk} \partial_i \tilde{\gamma}_{jk} - \text{tr} \bar{P} \partial_i \chi) + \bar{F}_j \partial_i \beta^j - \bar{E} \partial_i \alpha + \alpha \sqrt{\gamma} \mathcal{S}_i \quad (101)$$

where $\bar{P}_{ij} = \sqrt{\gamma} P_{ij}$ and $\text{tr} \bar{P} = \gamma_{jk} \bar{P}^{jk} = \tilde{\gamma}_{jk} \bar{P}^{jk} / \chi$.

The source terms in the neutrino (which appears exactly in the same form but with the opposite sign in the fluid MHD equations) are given by:

$$\mathcal{C} = \eta^0 - \kappa_a^0 n = \eta^0 - \kappa_a^0 \frac{N}{\Gamma} \quad (102)$$

$$\mathcal{S}^a = W(\eta - \kappa_a J) v^a - (\kappa_a + \kappa_s) H^a \quad (103)$$

$$\begin{aligned} \mathcal{S}_n &= -n_a \mathcal{S}^a = W(\eta - \kappa_a J) - (\kappa_a + \kappa_s) H^n \\ &= W [(\eta + \kappa_s J) - (\kappa_a + \kappa_s)(E - F_i v^i)] \end{aligned} \quad (104)$$

$$\mathcal{S}^i = \gamma_b^i \mathcal{S}^b = W(\eta - \kappa_a J) v^i - (\kappa_a + \kappa_s) H^i \quad (105)$$

To close this system of equations, we need three additional ingredients: a prescription for the computation of $P_{ij}(E, F_i)$ (i.e., a closure relation like the Minerbo described above), a prescription for the computation of the number flux, specific to the evolution of the number density, and the collisional source terms ($\mathcal{S}^a, \mathcal{C}$).

6. Time integration of equation with stiff source

Notice that we can split the vector of fields \mathbf{U} in two parts, one containing some stiff terms and other which all the rhs can be treated explicitly, respectively $\mathbf{U} = (\mathbf{V}, \mathbf{W})$. So, in general the evolution equations can be written as

$$\partial_t \mathbf{W} = F_W(\mathbf{V}, \mathbf{W}) \quad (106)$$

$$\partial_t \mathbf{V} = F_V(\mathbf{V}, \mathbf{W}) + \frac{1}{\epsilon(\mathbf{W})} R_V(\mathbf{V}, \mathbf{W}) \quad (107)$$

where we have considered that the relaxation parameter ϵ can depend also on the \mathbf{W} fields. The evolution procedure to compute each step $U^{(i)}$ can be split in two substeps

1. compute the explicit intermediate values $\{\mathbf{V}^*, \mathbf{W}^*\}$, that is,

$$\begin{aligned} \mathbf{W}^* &= \mathbf{W}^n + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} F_W(\mathbf{U}^{(j)}) \\ \mathbf{V}^* &= \mathbf{V}^n + \Delta t \sum_{j=1}^{i-1} \tilde{a}_{ij} F_V(\mathbf{U}^{(j)}) \\ &\quad + \Delta t \sum_{j=1}^{i-1} a_{ij} \frac{1}{\epsilon^{(j)}} R_V(\mathbf{U}^{(j)}) \end{aligned} \quad (108)$$

where we have defined $\epsilon^{(j)} = \epsilon(\mathbf{W}^{(j)})$.

2. compute the implicit part, involving only \mathbf{V} , by solving the implicit equation

$$\begin{aligned} \mathbf{V}^{(i)} &= \mathbf{V}^* + a_{ii} \frac{\Delta t}{\epsilon^{(i)}} R_V(\mathbf{V}^{(i)}, \mathbf{W}^{(i)}) \\ \mathbf{W}^{(i)} &= \mathbf{W}^* \end{aligned} \quad (109)$$

There are different ways to solve the equation in the second step (109).

If the source terms depend linearly on the evolution fields, then the stiff part can be written in the following way,

$$R_V(\mathbf{V}, \mathbf{W}) = A(\mathbf{W})\mathbf{V} + S(\mathbf{W}) \quad (110)$$

In this way, the implicit equation can be solved just by inverting the matrix, namely

$$\begin{aligned} \mathbf{V}^{(i)} &= [I - a_{ii} \frac{\Delta t}{\epsilon^{(i)}} A(\mathbf{W}^*)]^{-1} (\mathbf{V}^* + a_{ii} \frac{\Delta t}{\epsilon^{(i)}} S(\mathbf{W}^*)) \\ \mathbf{W}^{(i)} &= \mathbf{W}^* \end{aligned} \quad (111)$$

If the sources depend non-linearly on the evolution fields, then one need to solve the full non-linear implicit problem

$$\mathbf{G} = \mathbf{V}^{(i)} - \mathbf{V}^* - a_{ii} \frac{\Delta t}{\epsilon^{(i)}} R_V(\mathbf{V}^{(i)}, \mathbf{W}^{(i)}) \quad (112)$$

which will require the Jacobian $\mathcal{J} = \frac{\partial R_V}{\partial \mathbf{V}^{(i)}}$, which can be found explicitly in Radice paper, and it will be discussed in detail in the next subsection.

Another option would be to linearize the stiff term around $\{\mathbf{V}'\}$ (assuming $\mathbf{W}^{(i)}$ is known), namely

$$\begin{aligned} R_V(\mathbf{V}^{(i)}, \mathbf{W}^{(i)}) &\approx R_V(\mathbf{V}', \mathbf{W}^{(i)}) \\ &\quad + \left(\frac{\partial R_V}{\partial \mathbf{V}} \right)_{\mathbf{V}', \mathbf{W}^{(i)}} (\mathbf{V}^{(i)} - \mathbf{V}') \end{aligned} \quad (113)$$

By defining $A(\mathbf{V}', \mathbf{W}^{(i)}) \equiv \left(\frac{\partial R_V}{\partial \mathbf{V}} \right)_{\mathbf{V}', \mathbf{W}^{(i)}}$, substituting the previous expansion (113) in (109), adding and subtracting \mathbf{V}' on the right-hand-side, and rearranging the terms, it is obtained an expression that can be solved explicitly by inverting the matrix $M \equiv [I - a_{ii} \frac{\Delta t}{\epsilon^{(i)}} A(\mathbf{V}', \mathbf{W}^{(i)})]^{-1}$, namely

$$\mathbf{V}^{(i)} = \mathbf{V}' + M[\mathbf{V}^* - \mathbf{V}' + a_{ii} \frac{\Delta t}{\epsilon^{(i)}} R_V(\mathbf{V}', \mathbf{W}^{(i)})] \quad (114)$$

7. *Jacobian*

We will solve the full non-linear equation

$$\mathbf{G} = \mathbf{V}^{(i)} - \mathbf{V}^* - a_{ii} \frac{\Delta t}{\epsilon^{(i)}} R_V(\mathbf{V}^{(i)}, \mathbf{W}^{(i)}) = 0 \quad (115)$$

The N0 equation is just a scalar equation with a linear stiff term, which can be solved analytically, namely

$$\bar{N} = \bar{N}^* + a_{ii} \Delta t \alpha \sqrt{\gamma} \left(\eta^0 - \kappa_a^0 \frac{N}{\Gamma} \right) \quad (116)$$

By dividing by $\sqrt{\gamma}$ and defining $N^* = \bar{N}^*/\sqrt{\gamma}$, this relation can be inverted directly as

$$N = \frac{N^* + a_{ii} \Delta t \alpha \eta^0}{1 + a_{ii} \Delta t \alpha \kappa_a^0 / \Gamma} \quad (117)$$

Since Γ depends on (E, F_i) as it is specified in eq.(123), namely

$$\Gamma = W \left(\frac{E - F_i v^i}{J} \right) \quad (118)$$

this equation for N will be solved only after these fields have been updated.

The M1 system of equations is coupled and can not be solved directly. Instead, it forms a 4D system which needs to be solved simultaneously with an iterative method. Again, for the undensitized variables (i.e., dividing by $\sqrt{\gamma}$), the equations to be solved are

$$G = E - E^* \quad (119)$$

$$\begin{aligned} & - a_{ii} \Delta t \alpha W [(\eta + \kappa_s J) - (\kappa_a + \kappa_s)(E - F_i v^i)] \\ G_i &= F_i - F_i^* \quad (120) \\ & - a_{ii} \Delta t \alpha [W(\eta - \kappa_a J) v_i - (\kappa_a + \kappa_s) H_i] \end{aligned}$$

The multidimensional NR solver involves the inverse of the Jacobian of \mathbf{G} with respect to $\mathbf{V} = (E, F_i)$. The solution at the iteration n can be calculated from the solution at the iteration $n - 1$ as follows

$$\mathbf{V}^n = \mathbf{V}^{n-1} - \left(\frac{\partial \mathbf{G}}{\partial \mathbf{V}} \right)^{-1} \mathbf{G} \quad (121)$$

and where the Jacobian of G can be written in terms of the Jacobian of R_V easily, namely

$$\left(\frac{\partial \mathbf{G}}{\partial \mathbf{V}} \right) = \mathbf{I} - a_{ii} \frac{\Delta t}{\epsilon^{(i)}} \left(\frac{\partial R_V}{\partial \mathbf{V}} \right) \quad (122)$$

The stiff problem involves the fields $\mathbf{V} = (E, F_i, N)$, although actually it can be decoupled into a part involving $\mathbf{V} = (E, F_i)$ and a simple implicit equation for N to be solved after those fields have been updated. Let us define $\kappa_{as} = \kappa_a + \kappa_s$ and write J, H_i as a function of E, J_i, χ for the Minerbo closure, namely

$$J(E, F_i) = B_0 + d_{thin} B_{thin} + d_{thick} B_{thick} \quad (123)$$

$$\begin{aligned} H_i(E, F_i) &= -(a_{v0} + d_{thin} a_{vthin} + d_{thick} a_{vthick}) v_i \\ & - d_{thin} a_{fthin} \hat{f}_i - (a_{F0} + d_{thick} a_{Fthick}) F_i \end{aligned} \quad (124)$$

where we have also defined

$$\hat{f}_i = \frac{F_i}{\sqrt{F_k F^k}}, \quad d_{thick} = \frac{3}{2}(1 - \chi), \quad d_{thin} = 1 - d_{thick},$$

$$B_0 = W^2 [E - 2v_k F^k], \quad (125)$$

$$B_{thin} = W^2 E (v_k \hat{f}^k)^2, \quad (126)$$

$$B_{thick} = \frac{W^2 - 1}{2W^2 + 1} [4W^2 (v_k F^k) + (3 - 2W^2)E], \quad (127)$$

$$a_{v0} = W B_0, \quad (128)$$

$$a_{vthin} = W B_{thin}, \quad (129)$$

$$a_{vthick} = W B_{thick} \quad (130)$$

$$+ \frac{W}{2W^2 + 1} [(2W^2 - 1)(v_k F^k) + (3 - 2W^2)E],$$

$$a_{fthin} = W E (v_k \hat{f}^k), \quad (131)$$

$$a_{F0} = -W, \quad (132)$$

$$a_{Fthick} = W v^2. \quad (133)$$

The Jacobian $\mathcal{J} \equiv \left(\frac{\partial R_V}{\partial \mathbf{V}} \right)$ of the undensitized fields is then given by

$$\mathcal{J}_{00} = -\alpha W \left(\kappa_{as} - \kappa_s \frac{\partial J}{\partial E} \right), \quad (134)$$

$$\mathcal{J}_{0j} = +\alpha W \left(\kappa_s \frac{\partial J}{\partial F_j} + \kappa_{as} v^j \right), \quad (135)$$

$$\mathcal{J}_{i0} = -\alpha \left(\kappa_{as} \frac{\partial H_i}{\partial E} + W \kappa_a \frac{\partial J}{\partial E} v_i \right), \quad (136)$$

$$\mathcal{J}_{ij} = -\alpha \left(\kappa_{as} \frac{\partial H_i}{\partial F_j} + W \kappa_a v_i \frac{\partial J}{\partial F_j} \right). \quad (137)$$

where the necessary derivatives are

$$\frac{\partial J}{\partial E} = W^2 + d_{thin}(v_k \hat{f}^k)^2 W^2 + d_{thick} \frac{(3 - 2W^2)(W^2 - 1)}{1 + 2W^2}, \quad (138)$$

$$\frac{\partial J}{\partial F_j} = 2W^2 \left(-1 + d_{thin} \frac{E(v_k \hat{f}^k)}{F} + 2d_{thick} \frac{W^2 - 1}{1 + 2W^2} \right) v^j - 2d_{thin} \frac{W^2 E(v_k \hat{f}^k)^2}{F} \hat{f}^j, \quad (139)$$

$$\frac{\partial H_i}{\partial E} = W^3 \left(-1 - d_{thin}(v_k \hat{f}^k)^2 + d_{thick} \frac{2W^2 - 3}{1 + 2W^2} \right) v_i - d_{thin} W(v_k \hat{f}^k) \hat{f}_i, \quad (140)$$

$$\begin{aligned} \frac{\partial H_i}{\partial F_j} = & W \left(1 - d_{thin} \frac{E(v_k \hat{f}^k)}{F} - d_{thick} v^2 \right) \delta_i^j + 2W^3 \left[1 - d_{thin} \frac{E(v_k \hat{f}^k)}{F} - d_{thick} \left(v^2 + \frac{1}{2W^2(1 + 2W^2)} \right) \right] v_i v^j \\ & + 2d_{thin} \frac{WE(v_k \hat{f}^k)}{F} \hat{f}_i \hat{f}^j + 2d_{thin} \frac{W^3 E(v_k \hat{f}^k)^2}{F} v_i \hat{f}^j - d_{thin} \frac{WE}{F} \hat{f}_i v^j. \end{aligned} \quad (141)$$

A simple case, very useful for testing purposes, can be recovered at the zero fluid's velocity limit $v_i = 0 \rightarrow J = E, H_i = F_i$ and the stiff terms are just

$$R_E = \alpha(\eta - \kappa_a E), \quad R_F = -\alpha \kappa_{as} F_i \quad (142)$$

such that the Jacobian matrix is diagonal

$$\mathcal{J}_{00} = -\alpha \kappa_a, \quad \mathcal{J}_{ij} = -\alpha \kappa_{as} \delta_{ij} \quad (143)$$

and very easy to invert, either assuming that it is linear or using the full non-linear expression.

8. Simplified system

Let us consider a flat spacetime such that $(\alpha = 1, \beta^i = 0, \gamma_{ij}, \delta_{ij})$. It is straightforward to check that the M1 equations reduce to

$$\begin{aligned} \partial_t N + \partial_k \left[(Wv^k + \frac{H^k}{J}) \frac{N}{\Gamma} \right] &= \mathcal{C}^0 \\ \partial_t E + \partial_k [F^k] &= \mathcal{S}_n \\ \partial_t F_i + \partial_k [P^k{}_i] &= \mathcal{S}_i \end{aligned}$$

Let us consider also the simplification $v_i = 0$. In that case, the optically thick limit reduces to

$$J_{thick} = E \rightarrow P_{ij}^{thick} = \frac{1}{3} E \delta_{ij}. \quad (144)$$

while that the optically thin remains the same

$$P_{ij}^{thin} = \frac{F_i F_j}{F^k F_k} E \quad (145)$$

We combine both limits to construct the Minerbo closure

$$P_{ij} = \frac{3\chi - 1}{2} P_{ij}^{thin} + \frac{3(1 - \chi)}{2} P_{ij}^{thick} \quad (146)$$

where ξ is the Eddington factor. Since with $v_i = 0$ we have the following relations

$$J = E, \quad H_n = 0, \quad H_i = F_i \quad (147)$$

which imply $H_a H^a = H_k H^k - H_n^2 = F_i F^i$ and

$$\chi(\xi) = \frac{1}{3} + \xi^2 \left(\frac{6 - 2\xi + 6\xi^2}{15} \right), \quad \xi^2 = \frac{F_i F^i}{E^2} \quad (148)$$

For the source terms, the first observation is that $\Gamma = 1$, and that most of the terms vanish for $v_i = 0$, namely

$$\partial_t N + \partial_k \left[\frac{F^k}{E} N \right] = (\eta^0 - \kappa_a^0 N) \quad (149)$$

$$\partial_t E + \partial_k [F^k] = (\eta - \kappa_a E) \quad (150)$$

$$\partial_t F_i + \partial_k [P^k{}_i] = -(\kappa_a + \kappa_s) F_i \quad (151)$$

9. Calculating the emission and absorption coefficients

Emissivity, absorption and scattering coefficients are kept fixed throughout the implicit time integration. This can cause numerical scheme to oscillate if matter is thrown out of equilibrium over a timescale $\tau = (c\sqrt{\kappa_a(\kappa_a + \kappa_s)})^{-1}$ small compared with Δt . We assume that $\tau \geq \Delta t$ and that neutrinos and fluid are in thermodynamical equilibrium (i.e., equilibrium of temperature, chemical potential,...). That means that the temperature T is taken to be the fluid temperature, while the neutrino chemical potential are evaluated at equilibrium using the EoS at the fluid density, temperature and electron fraction, separately for each neutrino flavor, as follows

$$\mu_{\nu_e} = \mu_e + \mu_p - \mu_n, \quad \mu_{\bar{\nu}_e} = -\mu_{\nu_e}, \quad \mu_{\nu_x} = 0. \quad (152)$$

The value of these chemical potentials, for neutrinos in equilibrium with the fluid, are obtained directly from the equation-of-state table. Notice also that in general there should be an extra term $-Q = 1.2935 \text{ MeV}$ taking into account the rest-mass-energy difference between a neutron and a proton. We believed that term is already included in the table information.

We compute the source terms in the neutrino equations by assuming that the neutrinos obey a Fermi-Dirac distribution with temperature T_ν and chemical potential μ_ν . In that case, we have

$$f(\epsilon_\nu) = \frac{1}{1 + \exp[(\epsilon_\nu - \mu_\nu)/(K_B T_\nu)]} \quad (153)$$

where ϵ_ν is the neutrino energy, μ_ν is the neutrino chemical potential, k_B is the Boltzmann constant, and T_ν is the temperature of the neutrinos, calculated above.

Within these assumptions, the black-body function for the neutrino energy density is

$$B_\nu(T_\nu) = g_i \frac{4\pi}{(hc)^3} (k_B T_\nu)^4 F_3(\eta_\nu) \quad (154)$$

where F_3 is the Fermi function of order 3 and $\eta_\nu = \mu_\nu/(k_B T)$ is the degeneracy parameter of the neutrinos. Since $[hc] = \text{ergs cm}$ and $[k_B T] = \text{ergs}$, then $[B] = \text{ergs/cm}^3$. Here $g_e = 1$ and $g_X = 4$. The equilibrium number density of neutrinos is computed as

$$\mathcal{B}_\nu(T_\nu) = g_i \frac{4\pi}{(hc)^3} (k_B T_\nu)^3 F_2(\eta_\nu) \quad (155)$$

which has units of $[\mathcal{B}] = 1/\text{cm}^3$. Notice that these expressions could be used to compute a neutrino temperature T_ν by setting $J_\nu = B_\nu$, $n_\nu = \mathcal{B}_\nu$ and inverting those two relations to find T_ν . At this point it is important to stress few things about the units. The units of the opacities goes as the inverse of the mean free path, then $[\kappa] = 1/\text{cm}$. The emissivities are basically the emission rates, that for the number of neutrinos have units $[\eta_0] = 1/(\text{cm}^3 \text{s})$ and for the neutrino energy $[\eta_1] = \text{ergs}/(\text{cm}^3 \text{s})$. Notice that we can get a relation between the average energy and the temperature of the neutrinos as

$$\langle \epsilon_\nu \rangle = \frac{B_\nu}{\mathcal{B}_\nu} = \frac{F_3(\eta_\nu)}{F_2(\eta_\nu)} (k_B T_\nu) \quad (156)$$

We can estimate then the neutrino temperature as

$$k_B T_\nu = \frac{F_2(\eta_\nu)}{F_3(\eta_\nu)} \langle \epsilon_\nu \rangle$$

where the average energy of the neutrinos $\langle \epsilon_\nu \rangle$, using that in the fluid frame approximately $J \approx n \langle \epsilon_\nu \rangle$, can be calculated as

$$\langle \epsilon_\nu \rangle = \frac{J}{n} = \frac{\Gamma J}{N} = \frac{W(E - F^i v_i)}{N} \quad (157)$$

In the gray approximation, one uses $\{B_\nu(T), \mathcal{B}_\nu(T)\}$ with T the temperature of the fluid. The mismatch between the fluid temperature and the neutrino temperature introduces a correction factor in the energy-integrated opacities for $(\nu_e, \bar{\nu}_e)$, namely

$$\kappa_a = \kappa_a^{eq} \left[\max\left(1, \frac{T_\nu}{T}\right) \right]^2 = \kappa_a^{eq} \left[\max\left(1, \frac{\langle \epsilon_\nu \rangle}{\langle \epsilon_{eq} \rangle}\right) \right]^2 \quad (158)$$

$$\kappa_s = \kappa_s^{eq} \left[\max\left(1, \frac{T_\nu}{T}\right) \right]^2 = \kappa_s^{eq} \left[\max\left(1, \frac{\langle \epsilon_\nu \rangle}{\langle \epsilon_{eq} \rangle}\right) \right]^2 \quad (159)$$

where $\langle \epsilon_{eq} \rangle = B_\nu(T)/\mathcal{B}_\nu$, that is, the average energy of the BB spectra of the neutrinos in equilibrium with the fluid. We will try to use in general the expression with

the temperature, to have a feeling of the temperature of the neutrinos.

In order to guarantee that the neutrinos are in equilibrium with the fluid in the optically thick regions, we then compute the free-streaming emission due to charged-current reactions and the absorption due to pair processes through an energy-integrated version of Kirchhoff's law. At a given energy ϵ_ν , Kirchhoff's law

$$\eta^{eq} = \kappa^{eq} \int B(T, \mu_\nu) d\nu, \quad (160)$$

gives us a relation between the emissivity, the absorption opacity, and the equilibrium spectrum of the neutrino radiation. Given the blackbody function, we compute the ν_e and $\bar{\nu}_e$ emission coefficients and the ν_x absorption coefficients using Kirchhoff's law, namely

$$\eta_{\nu_e} = c \kappa_{a,\nu_e} B_{\nu_e}(T), \quad (161)$$

$$\eta_{\bar{\nu}_e} = c \kappa_{a,\bar{\nu}_e} B_{\bar{\nu}_e}(T), \quad (162)$$

$$c \kappa_{a,\nu_x} = \frac{\eta_{\nu_x}}{B_{\nu_x}(T)} \quad (163)$$

We apply the same treatment to the neutrino number emissivities and opacities, but using \mathcal{B} instead of B .

Notice that the Ruffert and Janka opacity and emissivity are probably not compatible with Kirchhoff's law out of the box. In the long term it would be better to have reactions that have detailed balance built in, rather than using Kirchhoff. In the case of the heavy-lepton neutrinos, the main production channel is nucleon-nucleon bremsstrahlung, while there is no absorption through charge current reactions (no muons), so we need to compute an effective absorption from the emission and not the other way around.

In summary, the logical procedure would be

$$(\rho, T, Y_e) \rightarrow (\mu_e, \mu_p, \mu_n) \rightarrow \mu_\nu \quad (164)$$

$$(J, n) \rightarrow \langle \epsilon_\nu \rangle, T_\nu \quad (165)$$

$$(\mu_\nu, \epsilon_\nu, T, T_\nu) \rightarrow (B_\nu, \mathcal{B}_\nu) \quad (166)$$

$$(\mu_\nu, \epsilon_\nu, T) \rightarrow (\kappa_{s,\nu_i}, \kappa_{a,\nu_e}, \kappa_{a,\bar{\nu}_e}, \eta_{\nu_x}) \quad (167)$$

10. Coupling neutrino matter

The M1+N0 equations, for each neutrino specie, reads as

$$\partial_t(\sqrt{\gamma} N^{\nu_e}) + \dots = \alpha \sqrt{\gamma} \mathcal{C}^{\nu_e} \quad (168)$$

$$\partial_t(\sqrt{\gamma} E^{\nu_e}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_n^{\nu_e} \quad (169)$$

$$\partial_t(\sqrt{\gamma} F_i^{\nu_e}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_i^{\nu_e} \quad (170)$$

$$\partial_t(\sqrt{\gamma} N^{\bar{\nu}_e}) + \dots = \alpha \sqrt{\gamma} \mathcal{C}^{\bar{\nu}_e} \quad (171)$$

$$\partial_t(\sqrt{\gamma} E^{\bar{\nu}_e}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_n^{\bar{\nu}_e} \quad (172)$$

$$\partial_t(\sqrt{\gamma} F_i^{\bar{\nu}_e}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_i^{\bar{\nu}_e} \quad (173)$$

$$\partial_t(\sqrt{\gamma} N^{\nu_x}) + \dots = \alpha \sqrt{\gamma} \mathcal{C}^{\nu_x} \quad (174)$$

$$\partial_t(\sqrt{\gamma} E^{\nu_x}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_n^{\nu_x} \quad (175)$$

$$\partial_t(\sqrt{\gamma} F_i^{\nu_x}) + \dots = \dots + \alpha \sqrt{\gamma} \mathcal{S}_i^{\nu_x} \quad (176)$$

while that the modified GRMHD equations become

$$\partial_t(\sqrt{\gamma}D_Y) + \dots = -\text{sgn}(\nu_i)m_N\alpha\sqrt{\gamma}\mathcal{C}^{\nu_i} \quad (177)$$

$$\partial_t(\sqrt{\gamma}\tau) + \dots = \dots - \alpha\sqrt{\gamma}(\mathcal{S}_n^{\nu_e} + \mathcal{S}_n^{\bar{\nu}_e} + \mathcal{S}_n^{\nu_x}) \quad (178)$$

$$\partial_t(\sqrt{\gamma}S_i) + \dots = \dots - \alpha\sqrt{\gamma}(\mathcal{S}_i^{\nu_e} + \mathcal{S}_i^{\bar{\nu}_e} + \mathcal{S}_i^{\nu_x}) \quad (179)$$

where $\text{sgn}(\nu_i)$ is 1 for ν_e , -1 for $\bar{\nu}_e$ and 0 for heavy-lepton neutrinos ν_x . The mass of the nucleon is $m_N \approx 1.6749 \times 10^{-30}$ g.

11. Neutrino processes

Neutrino emission is dominated by the following charged-current interactions:

$$e^- + p \rightarrow n + \nu_e \quad , \quad (180)$$

$$e^+ + n \rightarrow p + \bar{\nu}_e \quad , \quad (181)$$

and three pair process interactions: electron-positron pair annihilation

$$e^+ + e^- \rightarrow \bar{\nu}_i + \nu_i \quad (182)$$

plasmon decay,

$$\gamma + \gamma \rightarrow \bar{\nu}_i + \nu_i \quad (183)$$

and nucleon-nucleon bremsstrahlung,

$$N + N \rightarrow N + N + \bar{\nu}_i + \nu_i \quad (184)$$

Here the subscript ‘‘i’’ represents all neutrino species, and N stands for nucleons. The inverse reaction of the interactions above are responsible for neutrino absorption. Therefore, we need to compute both κ_a and η .

Lastly, we also consider the neutral-current neutrino scattering off nucleons,

$$\nu_i + N \rightarrow \nu_i + N \quad (185)$$

Note that we neglect electron-neutrino/antineutrino scattering for simplicity, noting that its contribution to the opacity is smaller than that for nucleon scattering by a typical factor of ≈ 100 in the HMNS remnant.

E. LES

extension of the LES for a microphysical EoS (new, calculated by Federico)

Notice that we hereafter have simplified the notation by removing the tildes and bars from the filtered fields and fluxes, for the sake of clarity. All fields in the equations are implicitly meant to be the filtered values (i.e., simply resolved by the discretized equations, as in any simulation).

$$\partial_t(\sqrt{\gamma}D) + \partial_k[-\beta^k\sqrt{\gamma}D + \alpha\sqrt{\gamma}(N^k - \tau_N^k)] = 0 \quad ,$$

$$\partial_t(\sqrt{\gamma}D_Y) + \partial_k[-\beta^k\sqrt{\gamma}D_Y + \alpha\sqrt{\gamma}(N_Y^k - \tau_{N_Y}^k)] = \frac{\alpha}{W}\sqrt{\gamma}DR_Y$$

$$\partial_t(\sqrt{\gamma}S_i) + \partial_k[-\beta^k\sqrt{\gamma}S_i + \alpha\sqrt{\gamma}(T_i^k - \gamma_{ij}\tau_T^{jk})] = \sqrt{\gamma}R^S{}_i \quad ,$$

$$\partial_t(\sqrt{\gamma}U) + \partial_k[-\beta^k\sqrt{\gamma}U + \alpha\sqrt{\gamma}S^k] = \sqrt{\gamma}R^U \quad ,$$

$$\partial_t(\sqrt{\gamma}B^i) + \partial_k[\sqrt{\gamma}(-\beta^k B^i + \beta^i B^k)$$

$$+ \alpha\sqrt{\gamma}(\gamma^{ki}\phi + M^{ki} - \tau_M^{ki})] = \sqrt{\gamma}R_B{}^i \quad ,$$

$$\partial_t(\sqrt{\gamma}\phi) + \partial_k[-\beta^k\sqrt{\gamma}\phi + \alpha c_h^2\sqrt{\gamma}B^k] = \sqrt{\gamma}R^\phi \quad . \quad (186)$$

we can define the energy without the rest-mass density $\tau = U - D$ with evolution equation

$$\partial_t(\sqrt{\gamma}\tau) + \partial_k[-\beta^k\sqrt{\gamma}\tau + \alpha\sqrt{\gamma}(S^k - N^k + \tau_N^k)] = \sqrt{\gamma}R^U \quad (187)$$

The fluxes consist of the following standard terms:

$$N^k = v^k D \quad , \quad (188)$$

$$N_Y^k = v^k D_Y \quad , \quad (189)$$

$$M^{ki} = B^i v^k - B^k v^i \quad , \quad (190)$$

$$\begin{aligned} T^{ki} &= hW^2 v^k v^i - E^k E^i - B^k B^i + \gamma^{ki} \left[p + \frac{1}{2}(E^2 + B^2) \right] \\ &= \frac{1}{2} (v^i S^j + v^j S^i) + \gamma^{ij} p - \frac{1}{W^2} \left(B^i B^j - \frac{1}{2} \gamma^{ij} B^2 \right) \\ &\quad - \frac{1}{2} (B^k v_k) \left[B^i v^j + B^j v^i - \gamma^{ij} (B^m v_m) \right] \quad , \quad (191) \end{aligned}$$

(where $E^i = -\epsilon^{ijk} v_j B_k$), and of the additional SGS terms:

$$\begin{aligned} \tau_N^k &= -\mathcal{C}_N \xi H_N^k \quad , \\ \tau_{N_Y}^k &= -\mathcal{C}_N \xi H_{N_Y}^k \quad , \\ \tau_T^{ki} &= -\mathcal{C}_T \xi H_T^{ki} \quad , \\ \tau_M^{ki} &= -\mathcal{C}_M \xi H_M^{ki} \quad . \quad (192) \end{aligned}$$

The cumbersome expressions of the tensors H have been obtained in detail for the special [12] and general relativistic [13] cases.

Here we apply the latter, extending the expressions reported in the Appendix A of Ref. [14] as to accommodate for the additional variables Y_e and D_Y (primitive and conserved, respectively). Besides the new SGS tensor $H_{N_e}^k$, the only other modification of the previous results arise on the term $H_p \equiv \nabla \frac{dp}{dC^a} \cdot \nabla C^a$ from the new dependence in the pressure, i.e., $p(\rho, \epsilon, Y_e)$.

$$\frac{dp}{dC^a} = \frac{dp}{d\rho} \frac{d\rho}{dC^a} + \frac{dp}{d\epsilon} \frac{d\epsilon}{dC^a} + \frac{dp}{dY_e} \frac{dY_e}{dC^a} \quad (193)$$

The only non-zero additional elements of the Jacobian (conserved-to-primitive) $\frac{dC^a}{dP^b}$ and its inverse $\frac{dP^a}{dC^b}$ ¹ are,

¹ This inversion is the only non-trivial new calculation, performed essentially using Mathematica.

respectively,

$$\begin{aligned} \frac{dD_Y}{dY_e} &= D \quad , \quad \frac{dD_Y}{dP^{a'}} = Y_e \frac{dD}{dP^{a'}} \\ \frac{dY_e}{dD_Y} &= \frac{1}{D} \quad , \quad \frac{dY_e}{dD} = -\frac{Y_e}{D} \end{aligned}$$

where the tilde in $P^{a'}$ denote the ‘‘old’’ set of primitive variables (i.e., excluding Y_e). Hence, we note that the new variables are only partially coupled to the system through the field D . In particular, we notice that $\frac{dp}{dD_Y} = \frac{d\epsilon}{dD_Y} = 0$. And we are now in conditions of computing

(193) and, therefore, to obtain the new expression for H_p .

$$\begin{aligned} H_p &= \nabla \left(\frac{dp}{d\rho} \frac{d\rho}{dC^{a'}} + \frac{dp}{d\epsilon} \frac{d\epsilon}{dC^{a'}} \right) \cdot \nabla C^{a'} + \nabla \left(\frac{dp}{dY_e} \frac{dY_e}{dC^a} \right) \cdot \nabla C^a \\ &= H_p^{\text{old}} + \nabla \left(\frac{1}{D} \frac{dp}{dY_e} \right) \cdot \nabla D_Y - \nabla \left(\frac{Y_e}{D} \frac{dp}{dY_e} \right) \cdot \nabla D \\ &= H_p^{\text{old}} + \nabla \frac{dp}{dY_e} \cdot \nabla Y_e - \frac{2}{D} \frac{dp}{dY_e} \nabla Y_e \cdot \nabla D \end{aligned} \quad (194)$$

The coefficient $\xi = \gamma^{1/3} \Delta^2 / 24$ has the proportionality to the spatial grid squared, which is typical of SGS models and ensures by construction the convergence to the continuous limit (vanishing SGS terms for an infinite resolution). Importantly, for each equation there is a pre-coefficient \mathcal{C}_i , which is meant to be of order one for a numerical scheme having a mathematically ideal Gaussian filter kernel and neglecting higher-order corrections. However, finite-difference numerical methods are usually more dissipative (and dispersive).

The relations for $\tilde{\Psi}$ and H s are given by:

$$\begin{aligned} \Psi_v^k &= \frac{2}{\Theta} \left\{ \nabla(v \cdot B) \cdot \nabla B^k - \nabla \Theta \cdot \nabla v^k + \frac{B^k}{\mathcal{E}} [\Theta \nabla B^j \cdot \nabla v_j + B_j \nabla B^j \cdot \nabla(v \cdot B) - B^j \nabla v_j \cdot \nabla \Theta] \right\} , \\ \Psi_M^{ki} &= \frac{4}{\Theta} \left[\Theta \nabla B^{[i} \cdot \nabla v^{k]} + B^{[i} \nabla B^{k]} \cdot \nabla(v \cdot B) - B^{[i} \nabla v^{k]} \cdot \nabla \Theta \right] , \\ \Psi_\Theta &= \frac{\Theta}{\Theta - E^2} \left\{ \nabla B_j \cdot \nabla B^j - \nabla E_j \cdot \nabla E^j - B_{[i} v_{k]} \Psi_M^{ki} \right\} , \quad \Psi_A = W^2 \left(p \frac{dp}{d\epsilon} + \rho^2 \frac{dp}{d\rho} \right) , \\ H_p &= \frac{\mathcal{E} W^2 (\Theta - E^2)}{(\rho \mathcal{E} - \Psi_A)(\Theta - E^2) W^2 + \Psi_A \Theta} \left\{ \rho \left(\nabla \frac{dp}{d\rho} \cdot \nabla \rho + \nabla \frac{dp}{d\epsilon} \cdot \nabla \epsilon \right) - 2 \frac{dp}{d\epsilon} \nabla \rho \cdot \nabla \epsilon \right. \\ &\quad - \left(\mathcal{E} \frac{dp}{d\epsilon} - \Psi_A \right) \left[\frac{W^2}{4} \nabla W^{-2} \cdot \nabla W^{-2} + \nabla W^{-2} \cdot \nabla(\ln \rho) \right] - \frac{2}{W^2} \frac{dp}{d\epsilon} [\nabla B_j \cdot \nabla B^j - W^4 \nabla W^{-2} \cdot \nabla h] \\ &\quad - \left(\mathcal{E} \frac{dp}{d\epsilon} + \Psi_A \right) [v_j \Psi_v^j + \nabla v_j \cdot \nabla v^j + W^2 \nabla W^{-2} \cdot \nabla W^{-2}] + \frac{\Psi_\Theta}{\mathcal{E} \Theta} \left[\left(\mathcal{E} \frac{dp}{d\epsilon} + \Psi_A \right) (\Theta - E^2) - \frac{\Psi_A \Theta}{W^2} \right] \left. \right\} \\ &\quad + \nabla \frac{dp}{dY_e} \cdot \nabla Y_e - \frac{2}{D} \frac{dp}{dY_e} \nabla Y_e \cdot \nabla D \\ H_\Theta &= \Psi_\Theta + \frac{\Theta}{\Theta - E^2} H_p , \quad (196) \\ H_v^k &:= \Psi_v^k - \left(v^k + \frac{v \cdot B}{\mathcal{E}} B^k \right) \frac{H_\Theta}{\Theta} , \quad (197) \\ H_N^k &= 2 \nabla D \cdot \nabla v^k + D H_v^k , \quad H_{N_e}^k = 2 \nabla D_Y \cdot \nabla v^k + D_Y H_v^k , \quad (198) \\ H_M^{ki} &= 2 B^{[i} H_v^{k]} + 4 \nabla B^{[i} \cdot \nabla v^{k]} \rightarrow H_E^i = \frac{1}{2} \epsilon^i_{jk} H_M^{jk} , \quad (199) \\ H_T^{ki} &= 2 \left[\nabla \mathcal{E} \cdot \nabla(v^k v^i) + \mathcal{E} \left(v^{(k} H_v^{i)} + \nabla v^k \cdot \nabla v^i \right) + v^k v^i H_p \right] - 2 \left[\nabla B^k \cdot \nabla B^i + \nabla E^k \cdot \nabla E^i + E^{(k} H_E^{i)} \right] \\ &\quad + (\gamma^{ki} - v^k v^i) \left[H_p + \nabla B_j \cdot \nabla B^j + \nabla E_j \cdot \nabla E^j + E_j H_E^j \right] , \quad (200) \end{aligned}$$

Notice that we need the derivatives ($dp/d\rho, dp/d\epsilon, dp/dY_e$) in order to compute the gradient SGS terms. They can be computed analytically for

hybrid EoS, but only numerically for the tabulated ones.

III. NUMERICAL IMPLEMENTATION

A. EoS

discuss how we are reading the info from the table, what units we use, what quantities we compute (dp/deps, dp/drho, cs2), and how to perform the con2prim.

High-resolution shock-capturing schemes integrate the fluid equations in conservation form for the conservative variables, while the fluid equations are written in a mixture of conserved and primitive variables. It is well known that the calculation of primitive variables from conserved variables for relativistic fluids requires solving a transcendental set of equations, which are only closed once an equation of state (EoS) is provided.

The dominant energy condition places constraints on the allowed values of the conserved and primitive variables

$$D \geq 0, \quad \tau \geq 0, \quad S^2 \leq (D + \tau)^2, \quad p \leq (\tau + D) \quad (201)$$

and depending on the EoS the second condition can be sharpened to $S^2 \leq (2D + \tau)\tau$, as far as τ remains positive.

1. Hybrid EoS

We consider that both the pressure and the internal energy have a cold and a thermal part, namely

$$p = p_{cold}(\rho) + p_{th}, \quad \epsilon = \epsilon_{cold}(\rho) + \epsilon_{th} \quad (202)$$

where that thermal pressure is modeled by using the ideal gas EoS $p_{th} = (\Gamma_{th} - 1)\rho\epsilon$, such that the total pressure can be written as

$$p = p_{cold}(\rho) + (\Gamma_{th} - 1)\rho[\epsilon - \epsilon_{cold}(\rho)] \quad (203)$$

The cold contribution to pressure and the internal energy only depend on the density and can be modeled with a piece-wise polytrope, namely:

$$p_{cold}(\rho) = K_i \rho^{\Gamma_i}, \quad \epsilon_{cold}(\rho) = a_i + \frac{K_i}{\Gamma_i - 1} \rho^{\Gamma_i - 1} \quad (204)$$

if $\rho_{i-1} \leq \rho \leq \rho_i$. The constant K_i and a_i can be found imposing continuity, leading to the following recursive formulas

$$a_i = a_{i-1} + \frac{K_{i-1}}{\Gamma_{i-1} - 1} \rho_{i-1}^{\Gamma_{i-1} - 1} - \frac{K_i}{\Gamma_i - 1} \rho_{i-1}^{\Gamma_i - 1} \quad (205)$$

$$K_i = K_{i-1} \rho_{i-1}^{\Gamma_{i-1} - \Gamma_i} \quad (206)$$

where $a_0 = 0$ and $\{K_0, \Gamma_0\}$ are given values. In general we will use an EoS with 4 different pieces, such that $i = 0, 1, 2, 3$.

One can calculate explicitly the derivatives of the pressure with respect to the density and the internal energy,

needed for the gradient SGS model in the LES. The latter is straightforward

$$\frac{dp}{d\epsilon} = (\Gamma_{th} - 1)\rho \quad (207)$$

For the other one we need to calculate the intermediate

$$\frac{dp_{cold}}{d\rho} = \frac{\Gamma_i p_{cold}}{\rho}, \quad \frac{d\epsilon_{cold}}{d\rho} = \frac{p_{cold}}{\rho^2} \quad (208)$$

Combining all this info, we obtain the final expression

$$\frac{dp}{d\rho} = (1 + \Gamma_i - \Gamma_{th}) \frac{p_{cold}}{\rho} + (\Gamma_{th} - 1)(\epsilon - \epsilon_{cold}) \quad (209)$$

We also need to modify the calculation of the sound speed just by using the general definition

$$c_s^2 = \frac{\rho}{h} \left[\frac{\partial p}{\partial \rho} + \frac{p}{\rho^2} \frac{\partial p}{\partial \epsilon} \right]_{s=cte} = \frac{\Gamma_{th} p + (\Gamma_i - \Gamma_{th}) p_{cold}}{h} \quad (210)$$

Although we have our own solver to recover the primitive fields from the conserved within this EoS, we have found that the procedure in Reprimand is more robust, so we have incorporated those libraries to solve for this case.

2. Microphysical EoS

Our method for solving these equations with a finite-temperature EoS is a modification of the algorithm that we use for the ideal gas EoS; the most significant change being that the internal energy must be calculated separately from the pressure using the table.

The list of primitive fields is given by $\{\rho, T, Y_e, p, v^j\}$, namely the rest-mass density, the temperature of the fluid, the electron fraction (i.e., the amount of electrons wrt neutrons), the pressure $p = p(\rho, T, Y_e)$ that is usually given in a tabulated form and the fluid velocity.

The evolved conserved variables are defined as

$$D \equiv \rho W \quad (211)$$

$$S_i \equiv (hW^2 + B^2) v_i - (B^j v_j) B_i \quad (212)$$

$$\tau \equiv hW^2 + B^2 - P - \frac{1}{2} \left((B^i v_i)^2 + \frac{B^2}{W^2} \right) \quad (213)$$

$$D_Y \equiv \rho W Y_e. \quad (214)$$

where the enthalpy is given as a function of the internal energy ϵ , so we need to extract it from the EoS, namely $\epsilon = \epsilon(\rho, T, Y_e)$.

The dominant energy condition places constraints on the allowed values of the conserved variables

$$D \geq 0, \quad S^2 \leq (D + \tau)^2, \quad D_Y \geq 0, \quad (215)$$

and depending on the EoS the second condition can be sharpened to $S^2 \leq (2D + \tau)\tau$, although it is not clear that it will work for this case. These constraints may

be violated during the evolution due to numerical error, and they are enforced before solving for the primitive variables. A minimum allowable value of the conserved density D_{vac} is chosen, and if D falls below this value we set $v^i = 0$ and $D \rightarrow D_{\text{vac}}$. We choose D_{vac} as low as possible for the unmagnetized neutron star binary, which is about nine orders of magnitude smaller than the initial central density of the stars. For the magnetized case our solver fails for such tenuous atmospheres, and it is increased by two orders of magnitude. If the second inequality is violated, then the magnitude of S_i is rescaled to satisfy the inequality. Finally, D_Y is required to satisfy the constraint on D , and the computed value of Y_e must be in the equation of state table.

The primitives to be found are the density, ρ , pressure P , electron fraction Y_e , internal energy density ϵ (i.e., or the temperature T , once the EoS is known) and velocity three-vector v^i . The magnetic field is at the same time conserved and primitive field. We write the transcendental equations in terms of the new rescaled variable

$$x \equiv \frac{hW^2}{\rho W}, \quad (216)$$

where h is the *total* enthalpy $h \equiv \rho(1 + \epsilon) + P$, and calculate Y_e from the evolution variables D_Y/D . Following [15], we rescale the conserved fields in order to get order-unity quantities, namely

$$q \equiv \tau/D, \quad r \equiv S^2/D^2, \quad s \equiv B^2/D, \quad t \equiv B_i S^i/D^{3/2}. \quad (217)$$

1D solver: Using data from the previous time step to calculate an initial guess for x , we iteratively solve these equations for x within the bounds

$$1 + q - s < x < 2 + 2q - s, \quad (218)$$

so that the final procedure can be written as

1. From the equation for $S^i S_i$, calculate an approximate Lorentz factor \hat{W}

$$\hat{W}^{-2} = 1 - \frac{x^2 r + (2x + s)t^2}{x^2 (x + s)^2}.$$

2. From the definition of D , calculate

$$\hat{\rho} = \frac{D}{\hat{W}}.$$

3. From the definition of τ and the total enthalpy, calculate

$$\hat{\epsilon} = -1 + \frac{x}{\hat{W}} \left(1 - \hat{W}^2\right) + \hat{W} \left[1 + q - s + \frac{t^2}{2x^2} + \frac{s}{2\hat{W}^2}\right].$$

4. Use the EoS table, invert $\hat{\epsilon}$ to find the corresponding temperature and then the pressure $P(\hat{\rho}, \hat{\epsilon}, Y_e)$.

5. Update the guess for x by solving the equation $f(x) = 0$ using the Brent method, being $f(x)$ just the definition of the unknown x ,

$$f(x) = x - \left(1 + \hat{\epsilon} + \frac{P(\hat{\rho}, \hat{\epsilon}, Y_e)}{\hat{\rho}}\right) \hat{W}$$

The root of $f(x) = 0$ from Step 5 becomes the new guess for x , and this process is repeated iteratively until the solution for x converges to a specified tolerance, which is ensured if there is a physical solution within the bounds. One advantage of this algorithm is that $f(x)$ is a function of a single variable, and, in contrast to a multiple variable search for a root, robust methods can be used to find any root that can be bracketed.

3D solver: Solvers for 2 or 3 variables can be faster in general than solving for only one, since there are less implicit calls $T = T(\rho, T, Y_e)$. We use the 3D solver from Moesta for the field $z = \rho h W^2$

We can extend our solver to 3D easily by defining the following functions (i.e., the definition of τ , S^2 and x) to be satisfied for the variables $\{W, z, T\}$, namely

$$\left[\tau + D - z - B^2 + \frac{(B^i S_i)^2}{2z^2} + P\right] W^2 - \frac{B^2}{2} = 0 \quad (219)$$

$$\left[(z + B^2)^2 - S^2 - \frac{(2z + B^2)}{z^2} (B^i S_i)^2\right] W^2 - (z + B^2)^2 = 0 \quad (220)$$

$$\frac{z - DW - PW^2}{DW} - \epsilon(\rho, T, Y_e) = 0 \quad (221)$$

taking into account that $\rho = D/W$ and $Y_e = D_Y/D$. A multi-dimensional Newton-Raphson solver requires the Jacobian of these equations, which can be computed analytically or numerically. Since this scheme also employs the temperature directly as an unknown, it does not require any inversions with the EOS. Once the system has been solved with a 3D NR scheme, one recovers the final primitives

$$v^i = \frac{\gamma^{ij} S_j}{z + B^2} + \frac{(B^j S_j) B^i}{z(z + B^2)} \quad (222)$$

$$\epsilon = \epsilon(\rho, T, Y_e) \quad (223)$$

Because of numerical error, a solution to these equations may either fall outside the physical range for the primitive variables, or a real solution for x may not exist. The solutions for ρ , T , and Y_e are, at a minimum, restricted to values in the table, and they are reset to new values (the minimum allowed value plus ten percent) if necessary. If a real solution for the primitive variables does not exist, the primitive variables are interpolated from neighboring points, and the conserved variables are reset to be consistent. If a valid interpolation stencil can not be constructed because the solver also failed at the neighboring points, then the update fails, and the run is terminated. This failure occurs very rarely and may be remedied by slightly increasing the density floor D_{vac} .

Technical details :There are several files, associated to the EoS, necessary to perform evolutions both with HAD or with MHDuet.

In had, we used shen.h5 (from stellarcollapse.org), which contains the full information about the EoS as a function of the coordinates (ρ, T, Y_e) . From this one we created several ascii files to help on initializing the initial data, like eos_table.d and eos_table_had.dat.

In MHDuet we started by changing the name of the shen.h5 file to eostable.h5, which is more generic. It still contains the whole information from the table and it is the one that can be downloaded from stellarcollapse.org. This data does not have the right format to be used with the Simflowny routines. Therefore, we extract the quantities needed for the initial data, the con2prim and the leakage, and save them in three different files with the right format, as we will describe before. These files are created with the python script esotable2hdf5.py, in script/Simflowny/tabEoS and also in newleakage/ID. **When creating the new tables, we convert the primitive fields from the table from \log_{10} to natural \log units, and from cgs to geometrized units. This makes the code faster, as natural \log is faster and we do not have to perform the conversions each time.** If we denote $\hat{u} = (\hat{\rho}, \hat{T}, \hat{\epsilon})$ the fields in the table and $u = (\rho, T, \epsilon)$ the ones that we use to evolve the code, to convert from one to the other we perform

$$\hat{\rho} = \log \rho \rightarrow \rho = \exp(\hat{\rho}) \quad (224)$$

$$\hat{T} = \log T \rightarrow T = \exp(\hat{T}) \quad (225)$$

$$\hat{\epsilon} = \log(\epsilon + \epsilon_{shift}) \rightarrow \epsilon = \exp(\hat{\epsilon}) - \epsilon_{shift} \quad (226)$$

The values of eos_rhomin, eos_rhomin, eos_yemin,... are computed automatically by checking the bounds in the table. The most sensible one is epsmin ($\hat{\epsilon}_{min}$), for which we need to search in the whole table. Then, the mintableEnergy can be recovered as $\epsilon_{min} = \exp(\hat{\epsilon}_{min}) - \epsilon_{shift}$. Notice that ϵ_{shift} is a parameter printed by the script esotable2hdf5.py, which needs to be introduced as an input in the MHDUET parameter file, similar to vacuum_rho (i.e., the density in the atmosphere), initialYe (i.e., constant or negative if beta-equilibrium) and initialTemperature (i.e., from the Lorene table). For the gradient SGS model we also need derivatives of the pressure with respect to (ρ, ϵ, Y_e) . They can be computed analytically for hybrid EoS, but for tabulated ones the easiest way is to compute the numerical derivatives of $\hat{p} = \hat{p}(\hat{\rho}, \hat{\epsilon}, \hat{Y}_e)$. Since $(\hat{p}, \hat{\rho}, \hat{\epsilon})$ are in natural log (i.e.,

\hat{Y}_e, \hat{T} are not! CP: pero T si esta en log scale!!), some factors appear when trying to convert from one to the other, namely

$$\frac{d\hat{p}}{d\hat{\rho}} = \frac{d \log p}{d \log \rho} = \frac{\rho}{p} \frac{dp}{d\rho} \rightarrow \frac{dp}{d\rho} = \frac{p}{\rho} \frac{d\hat{p}}{d\hat{\rho}} \quad (227)$$

$$\frac{dp}{d\epsilon} = \frac{p}{\epsilon} \frac{d\hat{p}}{d\hat{\epsilon}} \quad , \quad \frac{dp}{dY_e} = p \frac{d\hat{p}}{d\hat{Y}_e} \quad (228)$$

The script can be modified easily to allow for more outputs, although it seems more convenient to have one reduced file for the con2prim and a more extended one for the neutrino interaction terms.

- initial data: The file `InitialData.h5` is called in `ExternalInitialData` to fill the missing information in the initial data (pressure, internal energy and electron-fraction, assuming beta-equilibrium). Then, the inputs are (ρ, T, Y_e) in \log geometrized units, while that the outputs are $(c_s^2, \epsilon, p, \mu_e, \mu_p, \mu_n)$. This table is discarded after reading the initial data (i.e., not saved during the full simulation). Notice that it prints the values of `Energysift` and `eps_min`, which are input parameters of the parameter file of MHDUET.
- primitive recovery (con2prim): The file `Data.h5` is called in the `con2prim` routine to calculate the pressure and the internal energy in the Newton-Raphson. It also provides the sound speed, necessary for the spectral radius of the HRSC method. Then, the inputs are (ρ, T, Y_e) in \log geometrized units, while that the outputs are (c_s^2, ϵ, p) . This table is loaded if the `EoS.type=1`. Notice that sound speed from the table (\hat{c}_s^2) needs to be corrected with a factor $h = 1 + \epsilon + P/\rho$ such that $c_s^2 = \hat{c}_s^2/h$;
- leakage: The file `DataLeakage.h5` is called only when the neutrinos are activated and need to be evolved (i.e., 1 or 2 times for RK step). Right now it contains the fields necessary to calculate the opacities and the optical depth.

Notice that in order to generate the initial data Lorene needs an ascii table `eos_table.d` with (n_B, ρ, p) in physical units. We can generate this ascii table by using the EoS and routines provided in stellarcollapse.org. In particular, Dave modified some of them to include new options. The most important one is `idtable.f90`, available at `had/src/nuc_eos/Lorene.tables`. An example of usage to create such table could be

```
./idtable -b -g 2.0 -s -T 0.1 HShenEOS_rho220_temp180_ye65_version_1.1.20120817.h5
```

B. Ejecta

In order to characterize the ejecta from these simulations, we have analyzed data in 3D and at certain radius

spheres ($r = 100, 200, 300$). For the 3D case, we have computed:

- Histograms of the bound/unbound velocity follow-

ing two criteria: Bernoulli (for fluid, with $h_{\min} = 0.9987$) [$hu_t > h_{\min}$] and Geodesic (for particles) [$u_t > 1.0$].

- Histograms of the bound/unbound temperature (following the same two criteria). As T doesn't come out from the simulation, we needed to do an estimate following Appendix C (arXiv:2202.12901). The detailed procedure would be:

1. Estimate the cold part of the specific internal energy ϵ_{cold} knowing a_i, κ_i, γ_i (from APR4) and ρ (output of the simulation).
2. Obtain the thermal part:

$$\epsilon_{\text{th}} = \epsilon - \epsilon_{\text{cold}}$$

3. Compute the thermal energy density as:

$$\mathcal{E}_{\text{th}} = \rho \epsilon_{\text{th}}$$

4. Knowing that the nucleon number density is $n = \rho/m_{\text{nuc}}$, we can solve the following equation employing a Newton-Raphson to obtain the temperature:

$$\mathcal{E}_{\text{th}} = \frac{3}{2}nk_B T + \frac{11}{4}aT^4$$

5. Note 1: a is the radiation constant ($a = 7.567e - 15 \text{ erg}/(\text{cm}^3 \text{K}^4)$).

Note 2: For the Newton-Raphson we take as the initial guess: $T = \frac{2}{3} \frac{\mathcal{E}_{\text{th}}}{nk_B}$.

- Compute the period of the ejected mass at each point of the domain. The detailed procedure would be:

1. Compute the mass in each cell/point, i.e. $m = DdV$.
2. Estimate the energy in each cell (flat approximation) [in geometric units] by

$$E = \frac{1}{2}mv^2 - \frac{Mm}{r}$$

where M is the mass of the system (obtained from `mass.dat`).

3. We can compute the semi-major axis by

$$a = \frac{-Mm}{2E}$$

4. Now, we can obtain the period as

$$P = \sqrt{\frac{4\pi^2 a^3}{M}}$$

5. As we expect to only have negative values of a for the unbound matter, we set in these cases $a = 0$ to exclude them from the histograms

C. Eikonal

details on how to calculate the optical depth by solving the eikonal equation

The optical depth is the natural logarithm of the ratio of incident to transmitted radiant power through a material. Let us consider an incident flux I_ν^0 (depending on the frequency ν) which crosses a slab of material of size S and absorption coefficient per unit length κ_ν (or opacity). The transmitted flux is I_ν can be calculated from the differential equation

$$\begin{aligned} dI_\nu &= -\kappa_\nu I_\nu ds \rightarrow \\ I_\nu &= I_\nu^0 \exp(-\kappa_\nu S) = I_\nu^0 \exp(-S/l) = I_\nu^0 \exp(-\tau_\nu) \end{aligned} \quad (229)$$

where the optical depth $\tau_\nu = \kappa_\nu S = S/l$, where l is the mean free path, namely the distance at which the intensity falls over $1/e$. The regime $\tau_\nu \gg 1$ is called optically thick, because the mean free path is very small and the radiation needs many scattering events to get out of the medium. The regime $\tau_\nu \leq 1$ is called optically thin because the radiation only scatters once or less to get out of the medium.

In many occasions we will be interested on what is the shortest path for the radiation to escape from the medium where it is being produced, let us say, the interior of a star. For spherical stars, a radial path would be probably the shortest, but in more complicated remnants the minimum path is not so clear. Alternatively, the shortest distance from any point to the zero distance curve can be computed by solving the eikonal equation describing the motion of wave-fronts in optics, namely

$$|\nabla \tau_\nu| = \kappa_\nu \quad (230)$$

where τ_ν is the optical depth for some species of neutrino and κ_ν its corresponding opacity.

The usual approach to calculating the optical depth at a given point is to consider some small number of possible directions in which to integrate the opacity of the fluid. In `GR1D` [6], the assumption of spherical symmetry simplifies the calculation so that there are only ingoing and outgoing directions. Refs. [16] and [5] integrate the depth along rays in the coordinate directions, although Ref. [5] adds certain diagonal rays. The depth at any given point is then the minimum depth among the considered rays. Ref. [15] instead argues for rays that match the geometry of the problem and they therefore interpolate onto a spherical grid and considers the minimum depth among a set of radial rays.

In general, the existent algorithms necessarily involve global integrations that bring with them complexities due to multiple resolutions (from the AMR) and patches (from the domain decomposition). Instead, a more local approach that is independent of the particular symmetries of the problem is desirable. A simple approach appears to work quite well. In this scheme, the optical depth at any given point is simply the sum of the

depth incurred to get to a neighboring point plus the minimum depth among its neighbors. One can justify such an approach by arguing that neutrinos will explore all pathways out of the star, not just straight paths. This approach is also iterative since changes elsewhere do not immediately affect other areas, as would happen with a global integration. Physically one expects changes at the surface to take some time to propagate throughout the star. However, as noted in [6], because the depth depends on the opacity which itself depends on the depth, one expects to iterate in any case.

The computation of the optical depth appears to be an example of the eikonal problem. The eikonal equation takes the form

$$|\nabla u(\vec{x})|_{flat} = \sqrt{(\partial_x u)^2 + (\partial_y u)^2 + (\partial_z u)^2} = f(\vec{x}) \quad (231)$$

for scalar functions $u(x)$ and $f(x)$, and basically gives the minimal path line integral from the point \vec{x} to the zero level set, which can be located at infinity, namely

$$u(\vec{x}) = \min \int_{-\infty}^x f(\vec{l}) dl \quad (232)$$

Our naive algorithm explained above can be expressed as

$$U^{n+1} = \min(dU_{i\pm 1, j\pm 1, k\pm 1}^n) + \Delta x F \quad (233)$$

where $U \approx u(\vec{x})$ and $F \approx f(\vec{x})$ at the grid point, being d the normalized distance from the point $x_{i,j,k}$ to the minimum neighbor $x_{i\pm 1, j\pm 1, k\pm 1}$ (i.e., d takes values among 1, $\sqrt{2}$, or $\sqrt{3}$, depending on whether the point is immediately adjacent, diagonally along a plane parallel to a coordinate axis, or diagonally along a plane at 45° from a coordinate axis, respectively).

A more formal way of solving the eikonal equation is by using a first-order scheme to approximate the partial derivatives it is possible to show that this equation, in n -dimensions, can be written as

$$\sum_{j=1}^n \left(\frac{U - U_j}{\Delta x} \right)^2 = F^2 \quad (234)$$

where $U \approx u(\vec{x})$ and $F \approx f(\vec{x})$ at the grid point, being U_i the minimum of the neighboring values of $u(\vec{x})$ along the x^i direction. The solution of this quadratic equation for U is given by

$$U = \frac{1}{n} \sum_{j=1}^n U_j + \frac{1}{n} \sqrt{\left(\sum_{j=1}^n U_j \right)^2 - n \left(\sum_{j=1}^n U_j^2 - \Delta x^2 F^2 \right)} \quad (235)$$

If the discriminant in the square root is negative, then a lower-dimensional update must be performed recurrently, until that a one-dimensional formula is used. Therefore, the procedure in $n = 3$ would be:

- calculate the minimum of the solution at the point, $U_{i,j,k}$, along the x,y,z axis, denoted as

$$U_x \equiv \min(U_{i+1,j,k}, U_{i-1,j,k}) \quad (236)$$

$$U_y \equiv \min(U_{i,j+1,k}, U_{i,j-1,k}) \quad (237)$$

$$U_z \equiv \min(U_{i,j,k+1}, U_{i,j,k-1}) \quad (238)$$

- calculate the discriminant

$$D_3 = (U_x + U_y + U_z)^2 - 3(U_x^2 + U_y^2 + U_z^2 - \Delta x^2 F^2) \quad (239)$$

- if $D_3 \geq 0$, then $U = (U_x + U_y + U_z)/3 + \sqrt{D_3}/3$.
- if $D_3 < 0$, one needs to perform an $n-1=2$ dimensional update for all the planes and choose the minimum. For instance, in the plane $x - y$ would be

$$\begin{aligned} D_2^z &= (U_x + U_y)^2 - 2(U_x^2 + U_y^2 - \Delta x^2 F^2) \\ i f |U_x - U_y| &\leq \Delta x F \rightarrow U_2^z = (U_x + U_y)/2 + \sqrt{D_2^z}/2 \\ i f |U_x - U_y| &> \Delta x F \rightarrow U_2^z = \min(U_x, U_y) + \Delta x F \end{aligned}$$

This procedure is repeated to calculate U_2^x, U_2^y, U_2^z and then the final value is $U = \min(U_2^x, U_2^y, U_2^z)$.

The generalization to a curved background can be performed easily considering the generalized eikonal equation

$$|\nabla u(\vec{x})| = \sqrt{\gamma^{ij} (\nabla_i u) (\nabla_j u)} = f(\vec{x}) \quad (241)$$

which can be solved by assuming a conformally flat metric $\gamma^{ij} = \chi \eta^{ij}$, namely

$$|\nabla u(\vec{x})|_{flat} = \chi^{-1/2} f(\vec{x}) = (\sqrt{\gamma})^{1/3} f(\vec{x}) \quad (242)$$

Notice that the same factor can be obtained when computing the minimal distance, by using the line element $ds^2 = \gamma_{ij} dx^i dx^j \approx \chi^{-1} dx^2$.

D. Numerical methods

the numerical schemes that we are using to evolve the previous equations: method of lines with RK for time integrator, centered finite-difference for the spacetime, finite-difference HRSC for the matter with MP5 and Lax-Friedrichs.

Here we will describe in detail the numerical schemes available in Simflowny and their implementation into the SAMRAI infrastructure. We will discuss explicitly the sub-cycling in time for the Adaptive Mesh Refinement algorithms, including several useful tests. First we introduce the Method of Lines and the time discretization. Then we describe the spatial discretization for smooth and for non-smooth solutions.

1. The Method of Lines

The prototype of (first-order in time) evolution equations systems can be written as

$$\partial_t u = \mathcal{L}(u) \quad (243)$$

where u is the set of evolution fields and $\mathcal{L}(u)$ is an operator containing first and second order derivatives of the fields. We will consider that the system is hyperbolic with a spectral radius c_h (ie, the absolute value of the maximum eigenvalue). This continuum problem can be transformed to a semidiscrete one by allowing only for discrete space positions $\{x_i = i \Delta x, y_j = j \Delta y, z_k = k \Delta z\}$. At each point the evolution is given by the ODE

$$\partial_t U = \mathcal{L}(u) + Q_d^m(U) \quad (244)$$

where Q_d is an artificial dissipation operator included to remove the high frequency modes of the solution along a particular direction m , which can not be accurately resolved and achieve stability. The system can be fully discretized by choosing discrete timesteps $t^n = n \Delta t$. Explicit schemes are those for which the future solution which can be written in terms of the current one, namely

$$U^{n+1} = T(U^n) \quad (245)$$

where $T(U^n)$ can be a complicated operator depending on the time integrator, the space discretization and the dissipation.

The discrete system is stable, consistent and convergent to the continuum solution if locally stable time integrator is employed for the time evolution, like a Runge-Kutta of at least third order. Notice that the numerical scheme will be stable as long as the CFL condition $\Delta t \leq \Delta x / c_h$ is fulfilled.

2. The Runge-Kutta time integrator

A explicit Runge-Kutta scheme, applied to system (243), takes the form

$$U^{(i)} = U^n + \sum_{j=1}^{i-1} b_{ij} k_j, \quad k_j = \Delta t \mathcal{L}(U^{(j)})$$

$$U^{n+1} = U^n + \sum_{i=1}^s c_i k_i$$

where $U^{(i)}$ are the auxiliary intermediate values of the Runge-Kutta with s stages. The matrices $B = (b_{ij})$, with $b_{ij} = 0$ for $j \geq i$ are $s \times s$ matrices such that the resulting scheme is explicit and of order p . A Runge-Kutta is characterized by this matrix and the coefficient vector c_i , which can be represented by a tableau in the usual Butcher notation ([17])

where the coefficients \tilde{c} used for the treatment of non-autonomous systems are given by the consistency relation $a_i = \sum_{j=1}^{i-1} b_{ij}$. These schemes can be denoted as

$$\frac{a}{c^T} \Big| \frac{B}{c^T}$$

RK(s, p), where the doblot (s, p) characterizes the number of s stages of the explicit scheme and the order p of the scheme.

One can find Runge-Kuttas of order $p = s$ up to $p \leq 4$, making this choice optimal. A very well known fourth order Runge-Kutta with an effective CFL of 2 is given in Table I.

TABLE I. Tableau for a very standard explicit RK(4,4)

0	0	0	0	0
1/2	1/2	0	0	0
1/2	0	1/2	0	0
1	0	0	1	0
1/6 2/6 2/6 1/6				

Solutions of conservation equations have some norm that decreases in time. It would be desirable, in order to avoid spurious numerical oscillations arising near discontinuities of the solution, to maintain such property at a discrete level by the numerical method. If U^n represents a vector of solution values at the time $t = n \Delta t$, then a numerical scheme is said to be strong stability preserving (SSP) if maintains $\|U^{n+1}\| \leq \|U^n\|$ for all $n \geq 0$ in a given norm $\|\cdot\|$.

There are different SSP-RK schemes available in the literature. A very popular third order one [18], with an optimal CFL of 1, is given in Table II.

TABLE II. Tableau for the explicit RK-SSP(3,3) scheme

0	0	0	0
1	1	0	0
1/2	1/4	1/4	0
1/6 1/6 4/6			

Notice that a dense output interpolator can be constructed by using the sub-steps of the RK. Its generic forms is

$$U^{n+\theta} = U^n + \sum_{j=1}^s b_j(\theta) k_j, \quad \theta = \frac{t - t^n}{t^{n+1} - t^n} \quad (246)$$

where $b_i(\theta)$ are the coefficients to build the interpolator for a given RK scheme. Notice that the m -derivative can also be computed from this dense output interpolator

$$\frac{d^m}{dt^m} U(t^n + \theta \Delta t) = \frac{1}{h^m} \sum_{j=1}^s k_j \frac{d^m}{d\theta^m} b_j(\theta) + O(h^{4-m}), \quad (247)$$

For the standard RK(4, 4), it can be shown that there is a unique third order interpolator that can be written

as

$$\begin{aligned} b_1(\theta) &= \theta - \frac{3}{2}\theta^2 + \frac{2}{3}\theta^3, \quad b_2(\theta) = b_3(\theta) = \theta^2 - \frac{2}{3}\theta^3, \\ b_4(\theta) &= \frac{-1}{2}\theta^2 - \frac{2}{3}\theta^3 \end{aligned} \quad (248)$$

For the SSP-RK(3, 3) there is a second order interpolator which also satisfies the SSP condition

$$b_1(\theta) = \theta - \frac{5}{6}\theta^2, \quad b_2(\theta) = \frac{1}{6}\theta^2, \quad b_3(\theta) = \frac{4}{6}\theta^2 \quad (249)$$

3. IMEX

PD-ARS3 [ARS3] (Table III) In this section we will try a third-order 4-stage IMEX scheme [19]. Note its explicit tableau is SSPRK3. The Butcher tableau would be:

TABLE III. Tableau for the explicit (left) implicit (right) PD-ARS3 scheme

0	0	0	0	0	0	0	0	0
1	1	0	0	0	1	0	1	0
1/2	1/4	1/4	0	0	1/2	0	1/4	1/4
1	1/6	1/6	2/3	0	1	0	1/6	1/6
	1/6	1/6	2/3	0		0	1/6	1/6

PD-ARS4 [ARS4] (Table IV)

In this section we will try a fourth-order 4-stage IMEX scheme. Note its explicit tableau is SSPRK4. The Butcher tableau would be:

TABLE IV. Tableau for the explicit (left) implicit (right) PD-ARS4 scheme

0	0	0	0	0	0	0	0	0	0	0
1/2	1/2	0	0	0	0	1/2	0	1/2	0	0
1/2	0	1/2	0	0	0	1/2	0	0	1/2	0
1	0	0	1	0	0	1	0	0	0	1
1	1/6	2/6	2/6	1/6	0	1	0	1/6	2/6	2/6
	1/6	2/6	2/6	1/6	0		0	1/6	2/6	2/6

4. Spatial Discretization for smooth solutions

As a default we will use standard fourth-order centered finite difference. The first order derivative operators have the form

$$\begin{aligned} \partial_x U_{i,j,k} &= \frac{1}{12\Delta x} (U_{i-2,j,k} - 8U_{i-1,j,k} \\ &+ 8U_{i+1,j,k} - U_{i+2,j,k}) + \mathcal{O}(\Delta x^4) \end{aligned} \quad (250)$$

The second order can be constructed as the first order derivate applied twice on $U_{i,j,k}$. This is an acceptable choice for the cross-derivatives. For the instance, for the xy -derivative,

$$\partial_{xy} U_{i,j,k} = \partial_x (\partial_y U_{i,j,k}) = \partial_y (\partial_x U_{i,j,k}) \quad (251)$$

However, the stencil of the second order xx -derivative is twice larger, so it is preferable to change a different operator that is still fourth order and keeps the original stencil

$$\begin{aligned} \partial_{xx} U_{i,j,k} &= \frac{1}{12\Delta x^2} (-U_{i-2,j,k} + 16U_{i-1,j,k} - 30U_{i,j,k} \\ &+ 16U_{i+1,j,k} - U_{i+2,j,k}) + \mathcal{O}(\Delta x^4) \end{aligned} \quad (252)$$

We use centered derivative operators for all the derivative terms except for the advection terms, which are generically proportional to a vector β^i . In those case, we will use one-side derivative schemes,

$$\begin{aligned} \partial_x U_{i,j,k} &= \frac{1}{12\Delta x} (-U_{i-3,j,k} + 6U_{i-2,j,k} - 18U_{i-1,j,k} \\ &+ 10U_{i,j,k} + 3U_{i+1,j,k}) + \mathcal{O}(\Delta x^4), \quad \text{if } \beta^x < 0 \\ \partial_x U_{i,j,k} &= \frac{1}{12\Delta x} (U_{i+3,j,k} - 6U_{i+2,j,k} + 18U_{i+1,j,k} \\ &- 10U_{i,j,k} - 3U_{i-1,j,k}) + \mathcal{O}(\Delta x^4), \quad \text{if } \beta^x > 0 \end{aligned}$$

The Dissipation: As it was mentioned before, we use artificial dissipation to remove the high frequency modes of the solution which are not truly represented in our numerical grid (i.e., their wavelength is smaller than the grid size δx). We will use the Kreiss-Oliger dissipation operator [20]

$$Q_d^x = \sigma(-1)^{r-1} \Delta x^{2r-1} (D_+^x)^r (D_-^x)^r \quad (253)$$

where

$$D_+^x U_{i,j,k} = \frac{U_{i+1,j,k} - U_{i,j,k}}{\Delta x}, \quad (254)$$

$$D_-^x U_{i,j,k} = \frac{U_{i,j,k} - U_{i-1,j,k}}{\Delta x} \quad (255)$$

where $\sigma \geq 0$ is dissipative parameter. If the accuracy of the scheme without artificial dissipation is q , choosing $2r - 1 \geq q$ does not affect the accuracy of the scheme.

For a fourth order scheme we must use $r = 3$, leading to an operator for the x-direction

$$\begin{aligned} Q_d^x U_{i,j,k} &= \sigma(\Delta x)^5 (D_+^x)^3 (D_-^x)^3 U_{i,j,k} \\ &= \frac{\sigma}{64\Delta x} (U_{i-3,j,k} - 6U_{i-2,j,k} + 15U_{i-1,j,k} \\ &- 20U_{i,j,k} + 15U_{i+1,j,k} - 6U_{i+2,j,k} + U_{i+3,j,k}) \end{aligned} \quad (256)$$

Extrapolation at the boundaries: the simulation domain is extended by few ghost zones, which will allow us to compute centered derivatives of the fields and impose Sommerfeld-type boundary conditions. Filling the points outside the domain region is achieved by extrapolating the solution from the interior. This spatial interpolation must be at least as accurate as the spatial derivative operators in order to prevent the spoiling of the scheme accuracy. One of the simplest and most efficient options is to use Lagrange interpolating functions. Given a solution U_i at the position

x_i , one can construct a Lagrangian polynomial function of order k passing through a $k + 1$ set of points $\{(x_1, U_1), (x_2, U_2), \dots, (x_k, U_k), (x_{k+1}, U_{k+1})\}$, namely

$$p(x) = \sum_{j=1}^{k+1} U_j l_j(x) \quad , \quad l_j(x) = \prod_{\substack{m=1 \\ m \neq j}}^{k+1} \frac{x - x_m}{x_j - x_m} \quad (257)$$

where x is the point position in which the value is interpolated. To construct a symmetric Lagrangian polynomial of 4th-order, suitable for our 4th-order spatial scheme, five points are required,

$$\begin{aligned} p(x) &= \frac{(x - x_2)(x - x_3)(x - x_4)(x - x_5)}{(x_1 - x_2)(x_1 - x_3)(x_1 - x_4)(x_1 - x_5)} U_1 \\ &+ \frac{(x - x_1)(x - x_3)(x - x_4)(x - x_5)}{(x_2 - x_1)(x_2 - x_3)(x_2 - x_4)(x_2 - x_5)} U_2 \\ &+ \frac{(x - x_1)(x - x_2)(x - x_4)(x - x_5)}{(x_3 - x_1)(x_3 - x_2)(x_3 - x_4)(x_3 - x_5)} U_3 \\ &+ \frac{(x - x_1)(x - x_2)(x - x_3)(x - x_5)}{(x_4 - x_1)(x_4 - x_2)(x_4 - x_3)(x_4 - x_5)} U_4 \\ &+ \frac{(x - x_1)(x - x_2)(x - x_3)(x - x_4)}{(x_5 - x_1)(x_5 - x_2)(x_5 - x_3)(x_5 - x_4)} U_5 \end{aligned}$$

5. Spatial Discretization for non-smooth solutions

(Mostly from Ref. [21]). Our system of equations can be written formally in conservation law form, namely

$$\partial_t \mathbf{U} + \partial_k \mathcal{F}^k(\mathbf{U}) = S(\mathbf{U}) \quad (258)$$

where \mathbf{U} is the list of evolved fields, $\mathcal{F}^k(\mathbf{U}), S(\mathbf{U})$ their corresponding fluxes and sources, which might be non-linear but depend only on the fields and not on their derivatives.

However, these simple finite difference operators are not the optimal choice for the spatial discretization of fluxes in intrinsically non-linear systems like the MHD. In that case, it is advisable to use High-Resolution-Shock-Capturing (HRSC) methods [22], to deal with the possible appearance of shocks and to take advantage of the existence of weak solutions in the equations.

Our task is now to provide a stable and accurate non-oscillatory numerical approximation to $\mathcal{L}(U) = -\partial_k \mathcal{F}^k(\mathbf{U}) + S(\mathbf{U})$. To this purpose, we begin by focusing our attention to the x-direction and set, for ease of notations, $F_i = F^x(U_{i,j,k})$. We then let point values of the flux F_i correspond to the volume averages of another function, say \hat{F} , and define

$$F_i = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \hat{F}(\xi) d\xi = \frac{1}{\Delta x} [H(x_{i+1/2}) - H(x_{i-1/2})] \quad (259)$$

where

$$H(x) = \int_{-\infty}^x \hat{F}(\xi) d\xi \quad . \quad (260)$$

In this formalism, point values of the flux F_i are identified as cell averages of $\hat{F}(x)$ and $H(x)$ may be regarded as the primitive function of \hat{F} . Straightforward differentiation of Eq. (259) yields the conservative approximation

$$\frac{\partial F}{\partial x} \Big|_{x_i} = \frac{1}{\Delta x} (\hat{F}_{i+1/2} - \hat{F}_{i-1/2}) \quad (261)$$

Stated in this form, the problem consists of finding a high-order approximation to the interface values of $\hat{F}_{i+1/2}$ knowing the undivided differences of the primitive function $H(x)$, a procedure entirely analogous to that used in the context of finite volume methods. Thus one can set

$$\hat{F}_{i+1/2} = R(F_{[s]}) \quad (262)$$

where $R()$ is a highly accurate reconstruction scheme providing a stable interface flux value from point-wise values and the index $[s]$ spans through the interpolation stencil. The crucial issue in HRSC methods is how to approximately solve the Riemann problem, by reconstructing the fluxes at the interfaces such that no spurious oscillations appear in the solutions. This calculation can be performed as follows:

We consider the following combination of the fluxes and the fields, at each node i :

$$F_i^\pm = \frac{1}{2} (F_i \pm \lambda U_i) \quad (263)$$

where λ is the maximum propagation speed of the system in the neighboring points. Then, from the neighboring nodes $\{x_{i-n}, \dots, x_{i+1+n}\}$, we reconstruct the fluxes at the left and right of each interface as

$$F_{i+1/2}^L = R(\{F^+\}) \quad (264)$$

$$F_{i+1/2}^R = R(\{F^-\}) \quad (265)$$

The number $2(n+1)$ of such neighbors used in the reconstruction procedure depends on the order of the method. *Simflowny* already incorporates some commonly used reconstructions, like PPM [23], the Weighted-Essentially-Non-Oscillatory (WENO) reconstructions [24, 25], and MP5 [26], as well as other implementations like the FDOC families [27] which are almost as fast as centered finite difference schemes at the cost of some bounded spikes near the shock regions.

We use a flux formula to compute the final flux at each interface, e.g.:

$$\hat{F}_{i+1/2} = F_{i+1/2}^L + F_{i+1/2}^R \quad (266)$$

This reconstruction method does not require the characteristic decomposition of the system of equations (i.e., the full spectrum of characteristic velocities).

It is interesting to notice that, at the lowest order reconstruction, $F_{i+1/2}^L = F_i^+$ and $F_{i+1/2}^R = F_{i+1}^-$, so that the flux formula (266) reduces to the popular and robust Local-Lax-Friedrichs flux [22],

$$\hat{F}^{LLF} = \frac{1}{2} [F_i + F_{i+1} - \lambda(U_{i+1} - U_i)] \quad (267)$$

where

$$\lambda = \max(|\lambda_i|, |\lambda_{i+1}|). \quad (268)$$

For higher order reconstruction the non-linearities produces some differences and there is no direct correspondence between the finite-volume and finite-difference HRSC flux-formulas.

E. Linear reconstruction for advection-diffusion equations

Following Radice, the simplest FV-reconstruction that is asymptotically preserving in the limit of the diffusion equation is the piecewise linear

$$\hat{F}_{i+1/2}^{AC} = F_{i+1/2}^{HO} - A_{i+1/2} \varphi_{i+1/2} (F_{i+1/2}^{HO} - F_{i+1/2}^{LO}) \quad (269)$$

where

$$F_{i+1/2}^{HO} = \frac{1}{2} (F_i + F_{i+1}) \quad (270)$$

$$F_{i+1/2}^{LO} = \frac{1}{2} (F_i + F_{i+1}) - \frac{\lambda_{i+1/2}}{2} (U_{i+1} - U_i) \quad (271)$$

We can compute $A_{i+1/2}$ as the average of the neighboring points $A_{i+1/2} = \frac{1}{2}[A_i + A_{i+1}]$, with

$$A_i \approx \min \left[1, \frac{1}{\kappa_i \Delta x} \right] \approx \tanh \left[\frac{1}{\kappa_i \Delta x} \right]$$

For the limiter $\varphi_{i+1/2}$ we choose the Monotonic Centered (MC) limiter, namely

$$\varphi_{i+1/2} = \min \left[1, 2 \frac{u_i - u_{i-1}}{u_{i+1} - u_i}, 2 \frac{u_{i+2} - u_{i+1}}{u_{i+1} - u_i} \right]$$

This method can be translated to FD-reconstruction by using the above combination

$$F_i^\pm = \frac{1}{2} (F_i \pm \bar{\lambda}_i U_i) \quad (272)$$

$$F_{i+1/2}^L = F_i^+ + \frac{1}{2} \Delta_i^+ \quad (273)$$

$$F_{i+1/2}^R = F_{i+1}^- - \frac{1}{2} \Delta_{i+1}^- \quad (274)$$

$$\Delta_i^\pm = MC[F_{i+1}^\pm - F_i^\pm, F_i^\pm - F_{i-1}^\pm] \quad (275)$$

where in order to avoid oscillations, these slopes must be limited. A popular choice, much less dissipative than the straightforward minmod, is the Monotonized Central (MC) limiter

$$MC(a, b) = \frac{1}{2} (\text{sgn}(a) + \text{sgn}(b)) \min \left(\frac{1}{2} |a + b|, 2|a|, 2|b| \right) \quad (276)$$

Here there is some ambiguity on the limiter choice and on the value of $\bar{\lambda}_i$, which can be either

$$\begin{aligned} \bar{\lambda}_i &= A_i \max[\lambda_{i-2}, \lambda_{i-1}, \lambda_i, \lambda_{i+1}, \lambda_{i+2}] \\ \bar{\lambda}_i &= \max[A_{i-2} \lambda_{i-2}, A_{i-1} \lambda_{i-1}, A_i \lambda_i, A_{i+1} \lambda_{i+1}, A_{i+2} \lambda_{i+2}] \end{aligned} \quad (277)$$

In some cases, there might appear odd-even oscillations due to a decoupling of consecutive grid points in the spatial discretization scheme. A necessary and sufficient condition for this problem to appear is that [28]

$$g_i g_{i-1} < 0 \quad \text{and} \quad g_i g_{i+1} < 0 \quad (278)$$

where $g_i \equiv u_{i+1} + u_{i-1} - 2u_i$ is the concaveness of the evolved field u at the grid point x_i . When the above conditions are satisfied simultaneously, we set $A_i = 1$, which is enough to cure the problem.

For the FV-version of the scheme, we set $A_{i+1/2} = 1$ if the following conditions are satisfied

$$(u_i - u_{i-1})(u_{i+1} - u_i) < 0 \quad \text{and} \quad (u_{i+1} - u_i)(u_{i+2} - u_{i+1}) < 0 \quad (279)$$

F. Monotonic Preserving MP5

Undivided difference will be frequently used and denoted with $\Delta_{i+1/2} = f_{i+1} - f_i$. Occasionally, we will also make use of the Minmod and Median functions defined, respectively as

$$\text{Minmod}(a, b) = \frac{\text{sgn}(a) + \text{sgn}(b)}{2} \min(|a|, |b|) \quad (280)$$

$$\text{Median}(a, b, c) = a + \text{Minmod}(b - a, c - a) \quad (281)$$

The monotonicity preserving (MP) schemes of Suresh & Huynh achieve high-order interface reconstruction by first providing an accurate polynomial interpolation and then by limiting the resulting value so as to preserve monotonicity near discontinuities and accuracy in smooth regions. The MP algorithm is better sought on stencils with five or more points in order to distinguish between local extrema and a genuine $O(1)$ discontinuities. Here we employ the fifth-order accurate scheme based on the (unlimited) interface value given by

$$f_{i+1/2} = \frac{1}{60} (2f_{i-2} - 13f_{i-1} + 47f_i + 27f_{i+1} - 3f_{i+2}) \quad (281)$$

based on the five point values f_{i-2}, \dots, f_{i+2} . Together with this equation, we also define the monotonicity-preserving bound

$$f^{MP} = f_i + \text{Minmod}(\Delta_{i+1/2}, \alpha \Delta_{i-1/2}) \quad (281)$$

resulting from the median between f_i, f_{i+1} and the left-sided extrapolated upper limit $f^{UL} = f_i + \alpha \Delta_{i-1/2}$. The parameter $\alpha \geq 2$ controls the maximum steepness of the left sided slope and preserves monotonicity during a single Runge-Kutta stage, provided the CFL number satisfies $C_a \leq 1/(1 + \alpha)$. In practice, setting $\alpha = 4$ still allows larger values of C_a to be used. The interface value given by Eq. III F is not altered when the data is sufficiently smooth or monotone that $f_{i+1/2}$ lies inside the interval defined by $[f_i, f^{MP}]$. Otherwise limiting takes

place by bringing the original value back into a new interval $I[f^{min}, f^{max}]$ specifically designed to preserve accuracy near smooth extrema and provide monotone profile close to discontinuous data. The final reconstruction can be written as

$$R(f_{[s]}) = \begin{cases} f_{i+1/2} & \text{if } (f_{i+1/2} - f_i)(f_{i+1/2} - f^{MP}) < 0, \\ \text{Median}(f^{min}, f_{i+1/2}, f^{max}) & \text{otherwise} \end{cases} \quad (281)$$

where

$$\begin{aligned} f^{min} &= \max[\min(f_i, f_{i+1}, f^{MD}), \min(f_i, f^{UL}, f^{LC})], \\ f^{max} &= \min[\max(f_i, f_{i+1}, f^{MD}), \max(f_i, f^{UL}, f^{LC})]. \end{aligned}$$

These bounds provide accuracy-preserving constraints by allowing the original interface value $f_{i+1/2}$ to lie in a somewhat larger interval than $I[f_i, f_{i+1}]$ or $I[f_i, f^{UL}]$. This is accomplished by considering the intersection of the two extended intervals $I[f_i, f_{i+1}, f^{MD}]$ and $I[f_i, f^{UL}, f^{LC}]$ that leave enough room to accommodate smooth extrema based on a measure of the local curvature defined by

$$d_{i+1/2}^{M4} = \text{Minmod}(4d_i - d_{i+1}, 4d_{i+1} - d_i, d_i, d_{i+1}), \quad (279)$$

where $d_i = \Delta_{i+1/2} - \Delta_{i-1/2}$. Using Eq.III F, one defines the median f^{MD} and the large curvature f^{LC} values as

$$f_{i+1/2}^{MD} = \frac{f_i + f_{i+1}}{2} - \frac{1}{2}d_{i+1/2}^{M4} \quad (280)$$

$$f_{i+1/2}^{LC} = f_i + \frac{1}{2}\Delta_{i-1/2} + \frac{4}{3}d_{i-1/2}^{M4} \quad (281)$$

respectively. The curvature measure provided by Eq.III F is somewhat heuristic and chosen to reduce the amount of room for local extrema to develop. The reconstruction illustrated preserves monotonicity and does not degenerate to first-order in proximity of smooth extrema.

Finally, the finite-difference HRSC scheme with MP5 reconstruction could be summarized as follows

$$F_i^\pm = \frac{1}{2}(F_i \pm \lambda U_i) \quad (282)$$

$$F_{i+1/2}^L = R_{MP5}(F_{i-2}^+, F_{i-1}^+, F_i^+, F_{i+1}^+, F_{i+2}^+) \quad (283)$$

$$F_{i+1/2}^R = R_{MP5}(F_{i+3}^-, F_{i+2}^-, F_{i+1}^-, F_i^-, F_{i-1}^-) \quad (284)$$

$$\hat{F}_{i+1/2} = F_{i+1/2}^L + F_{i+1/2}^R \quad (285)$$

G. AMR

reminder on parallelization and AMR, in particular the sub-cycling in time with high-order accuracy and short ghost zones. Details on the parallelization of the eikonal? SAMRAI.

The code presented here has been generated by using *Simflowny* [29, 30] together with the infrastructure *SAMRAI* [31, 32]. *Simflowny* is an open-source and user-friendly platform developed by the IAC3 group since

2008 to facilitate the use of HPC infrastructures to non-specialist scientists. It allows to easily implement scientific dynamical models, by means of a Domain Specific Language, and a web-based integrated development environment, which automatically generates efficient parallel code for simulation frameworks. *Simflowny* splits the physical models and problems from the numerical techniques. The automatic generation of the simulating code allows to properly include the parallelization features, which in this case rely on the *SAMRAI* infrastructure [31]². *SAMRAI* is a patch-based structured AMR developed over more than 15 years by the Center for Applied Scientific Computing at the Lawrence Livermore National Laboratory. The latest upgrades on the AMR algorithms allow to improve the performance and reach a good scaling on up to 1.5M cores and 2M MPI tasks [32], at least for some specific problems. The combination of these two platforms provides a final code with a good balance of speed, accuracy, scalability, ability to switch physical models (flexibility), and the capacity to run in different infrastructures (portability).

One way to use efficiently the computational resources is increasing the grid resolution only on the localized regions of the simulation domain where the dynamics is more demanding and higher resolution is required to improve the accuracy of the solution. A mature and well-established strategy is the AMR, which introduces new additional grid levels with higher resolution on specific regions which might change dynamically with the solution. The AMR algorithm specifies how the solution on multi-processor and multi-levels is evolved, and in particular, how the information on the different domain boundaries is shared among the multiple processors. In our approach, a generic AMR algorithm is constructed by using the basic blocks (i.e., routines and functions) provided by SAMRAI. The algorithm skeleton for a problem with L refinement levels, for the specific case of RK integrators with S sub-steps, could be written as follows:

The algorithm calls **refinement criteria** to decide which regions need additional levels with smaller grid sizes to obtain an accurate solution. Once the solution is defined in all levels the simulation can start. The procedure to integrate a time-step is repeated over and over until reaching the final simulation time. The fields must be evolved in all grids each timestep, starting from the coarsest level $l = 0$ to the finest one $l = L$. Each time integration is performed by using a RK with S sub-steps. Therefore, the intermediate auxiliary states $U^{(i)}$ and the final one U^{n+1} must be computed at each level. The right-hand-side of the evolution equations, which involves spatial derivatives, need to be computed at each of these sub-steps, by using the discrete spatial operators described in the previous section. Notice also that

² See also the website

<https://computation.llnl.gov/project/SAMRAI/>

the nearby zones outside the boundary of the fine levels must be filled with points of the same resolution in order to accurately evolve the solution. This procedure is called **prolongation** and it usually involves interpolation from the coarse grid level into the fine one. After computing each intermediate RK-step the fields need to be synchronized among the different processors on level l in order to fill the boundaries of the domains splatted in each processor with the correct updated data. Similarly, after finishing all the steps of the RK, we need to inject the solution of the fine level l into the coarse one $l - 1$, a procedure known as **restriction**. After the values on the coarse grids have been updated, the information on the level $l - 1$ must be again synchronized among processors.

H. Refinement criteria, restriction and prolongation

There are several strategies to decide which regions need more resolution to be accurately resolved by including additional grid levels with higher resolution. These strategies rely on going through the points of the coarsest level and evaluating some refinement criteria, such that an additional level can afterwards be added in the tagged regions. This process can be repeated in the new refined levels until some condition is fulfilled, either on the refinement criteria or on a maximum allowed number of levels. There are two refinement tagging strategies provided by SAMRAI integrated in Simflowny.

- **Fixed Mesh Refinement (FMR)**. The user specifies statically a set of boxes where the refinement is located. Every level allows different boxes as long as they are nested in coarser level boxes.
- **Adaptive Mesh Refinement (AMR)**. The user sets a criteria (i.e., a measurement of the error or a function of the fields surpassing certain threshold) used to dynamically calculate the cells to be refined.

Notice that fixed and dynamical tagging strategies (i.e., FMR and AMR) can be combined in the same simulation. As the simulation evolves, the AMR tagging criteria will likely change, implying that new regions will be refined and old ones will be disposed of. This re-meshing procedure is performed periodically.

If a new refinement level is added dynamically during the simulation (i.e., or the region of a given level increases due to the dynamical AMR criteria), the domain of that grid increases with respect to the coarser level. The new grid points on the fine level are set by the prolongation procedure, interpolating the solution from the coarse grid into the fine one. This spatial interpolation must be more accurate than the spatial derivative operators in order to prevent the spoiling of the scheme accuracy. One of the simplest and most efficient options is to use Lagrange interpolating functions. Given a solution U_i at the position x_i , one can construct a Lagrangian

polynomial function of order k passing through a $k + 1$ set of points $\{(x_1, U_1), (x_2, U_2), \dots, (x_k, U_k), (x_{k+1}, U_{k+1})\}$, namely

$$p(x) = \sum_{j=1}^{k+1} U_j l_j(x) \quad , \quad l_j(x) = \prod_{\substack{m=1 \\ m \neq j}}^{k+1} \frac{x - x_m}{x_j - x_m} \quad (285)$$

where x is the point position in which the value is interpolated. To construct a symmetric Lagrangian polynomial of 5th-order, suitable for our 4th-order spatial scheme, six points are required (i.e, three at each side of the point to be interpolated). Such Lagrangian polynomial interpolation can be simplified for the centered point $x = x_0$, namely

$$p(x_0) = \frac{1}{256} \left[150(U_{x-1} + U_{x+1}) - 25(U_{x-2} + U_{x+2}) + 3(U_{x-3} + U_{x+3}) \right] \quad (285)$$

In structured grids it is common to choose refined grids such that the points of the coarse grid also exist in the fine grid (i.e., the ratio between their resolutions is 2^p), so this interpolation is the only one required.

Since we are interested on MHD problems involving non-smooth solutions, it is relevant to study if this interpolation is suitable when shocks and discontinuities are present in our simulation. Indeed, this simple Lagrange interpolation has been compared to a WENO interpolation for systems of equations with non-smooth solutions [33]. The comparison indicates that the simple and efficient Lagrange interpolation, combined with a WENO finite difference method to discretize the derivatives during the evolution, suffices for the domain interface treatment to retain high-order of accuracy and essentially non-oscillatory properties even for strong shocks [33].

The restriction procedure is complementary to the prolongation. In the restriction, on the regions with overlapping grids, the data from a fine level is injected into a coarse one. If the points of the coarse grid also exist in the fine grid (i.e., like when the ratio between the two resolutions is 2^p), the restriction is quite straightforward and only implies copying directly data from the fine level to the coarse one.

I. Sub-cycling in time

A necessary condition for the stability of explicit numerical schemes of hyperbolic systems is that the time step must satisfy the CFL condition $\Delta t \leq \lambda_{\text{CFL}} \Delta x$, with λ_{CFL} a factor depending on the dimensionality of the problem and the specific time integrator. When there are multi-levels $l = 0..L$, the solution on each refinement level can be evolved in a stable way by using the time-step corresponding to the finest grid resolution Δx_L , ensuring that all the grids satisfy the CFL condition. This

is however a very inefficient choice, since coarser grids are evolved with a time-step much smaller than the one allowed by their local CFL condition.

A common way to avoid such a restriction is by evolving the solution with sub-cycling in time, meaning that each grid uses the largest Δt as set by its local CFL condition. This means that the finer grids must perform two or more time-steps for each one of the coarse grid. In this case, it is not clear how to evolve the interior points of the fine grid at the refinement boundary, since the solution is not evaluated at the same time on the coarser grid. There have been several well motivated strategies to fill in this missing information:

- **Tapering.** The fine grid is extended by a number of points given by $N_{\text{ext}} = f_{\text{res}} N_{\text{st}} N_{\text{RK}}$, for a resolution ratio of f_{res} , stencil points N_{st} and RK time sub-steps N_{RK} , on each direction perpendicular to the refinement boundary. This way, points at the boundary can be evolved without any intermediate prolongation [34]. The boundary points at the end of the time step of the fine grid (i.e., when it reaches U^{n+1}) are inside the numerical domain of dependence of the extended initial fine grid. This algorithm is computationally expensive and it is difficult to achieve a good scalability because involves extending each refinement grid by a large number of points in each direction. For instance, with a 4th-order RK and 4th-order space discretization it would be around 16 points on each side of the fine grid. However, it is very accurate, since it minimizes boundary reflections at the interfaces between levels.
- **Berger-Oliger algorithm (BO1).** The solution of the coarser grid is evolved first up to $n+1$. Then, with the information from $\{U^{n+1}, U^n\}$, we can interpolate in time to calculate the solution at the required times of the RK schema of the finer grids. Spatial interpolation (prolongation) is also required to fill the points in the positions needed by the spatial discretization scheme. This algorithm is cheap, fast and efficient, since it requires to interpolate only in a number of points equal to the stencil of the spatial discretization scheme. For instance, it just requires 3 points in the ghost-zone for a 4th-order centered derivatives with 6th-order dissipation. The drawback of this simplest original version is that it is only 1st-order accurate in time.
- **Berger-Oliger with dense output interpolator (BO).** The original Berger-Oliger algorithm can be improved by using additional information to increase the accuracy of the interpolation scheme by either (i) including other time-levels $\{U^{n-1}, U^{n-2}, \dots\}$, or (ii) including the intermediate RK solutions $U^{(i)}$. This last option, that we will consider here, is commonly known as dense output interpolator, and its implementation for some RK schemes is discussed in detail in Appendix 5.

- **Berger-Oliger without order reduction (BOR).** The first step of the algorithm is similar to the BO one, using information from all the sub-steps of the RK (i.e., $\{U^n, U^{(i)}, U^{n+1}\}$) to build an internal dense output interpolator of order $q = p - 1$. However, in the second step this interpolator is used for computing all the time derivatives of the fine grid [35, 36]. By using the standard RK formula with these time derivatives it is possible to calculate the solution at each RK sub-step and achieve a final scheme at least order q in time. This algorithm, which is discussed thoroughly in Appendix 2, is fast, efficient and very accurate. Moreover, we have extended the algorithm to allow arbitrary resolution ratios between consecutive AMR grids.

The prolongation, restriction and number of executions depend on whether sub-cycling is active and which option from the previous ones is being considered. Currently, there are four available AMR time integrations in Simflowny: no sub-cycling in time, tapering, standard BO and BOR. The latter will be our preferred choice.

APPENDIX A. CONVERSION TO PHYSICAL UNITS

It is customary in general relativity to adopt geometrized units $G = c = 1$, such that all quantities, including mass (M) and time (T), have units of length (L). Vacuum solutions are invariant under changes in this fundamental length scale L . A quantity X that scales as $L^l M^m T^t$ can be converted into geometrized units by multiplying with the factor $c^t (G/c^2)^m$. After the conversion to geometrized units, X scales as L^{l+m+t} .

Most equations of state break this intrinsic scale-invariance, and the fundamental length-scale must be fixed by additional choices. Once the new scale is chosen, transformations between geometrized and physical units can be easily made. In the following, we summarize the basic procedure detailed in [37] to account for the proper scaling of quantities.

There are two common approaches in the literature to set this additional length scale. The first one is obtained by fixing a constant physical quantity, e.g., the solar mass $M_\odot = 1$, and from it deduce the appropriate conversion factors. That is, if a quantity \hat{X} has dimensions of $L^l M^m T^t$, its dimensionless counterpart, \bar{X} , is obtained from the following equation:

$$\hat{X} = \left(\frac{G M_\odot}{c^2} \right)^{l+t} \frac{M_\odot^m}{c^t} \bar{X}. \quad (285)$$

where the constant in the MKS (meter-kg-second) and the cgs (centimeter-g-second) systems are

$$\begin{aligned} G &= 6.67 \times 10^{-11} m^3 kg^{-1} s^{-2} = 6.67 \times 10^{-8} cm^3 g^{-1} s^{-2} \\ c &= 3.0 \times 10^8 ms^{-1} = 3.0 \times 10^{10} cms^{-1} \\ M_\odot &= 1.989 \times 10^{30} kg = 1.989 \times 10^{33} g \end{aligned}$$

Remember that in the MKS system the magnetic field is in Teslas and the energy in Jules, while that in the cgs the magnetic field is in Gauss and the energy in ergs.

There is still the freedom to choose κ , and all dimensions are scaled with this parameter. Usually the choice $\kappa = 100$ is preferred because it leads to physical units which are close to the current observations. For instance, TOV stars constructed with these parameters have a maximum stable mass of $\hat{M}_{\max} = 1.64M_{\odot}$ with a radius of $\hat{R}_{\max} = 14.11$ km.

For a polytropic EoS $p = K\rho^{\Gamma}$ with $\Gamma = 1 + 1/n$, the adimensional quantities \bar{X} are related to the physical ones through the polytropic constant K , namely

$$\begin{aligned} \bar{M} &= K^{-n/2}M, \quad \bar{R} = K^{-n/2}R, \\ \bar{T} &= K^{-n/2}T, \quad \bar{\Omega} = K^{n/2}\Omega, \quad \bar{B} = K^{n/2}B \end{aligned} \quad (282)$$

where the last relation can be obtained easily from the energy $B^2R^3 = M$, the luminosity $L = M/T = B^2R^6\Omega^4$ or from the magnetic pressure. This way, we can convert easily the values from any K to a different one. For instance, if the mass of a star with $K = 372$ and $\Gamma = 2$ is $M = 3.15$, then $\bar{M} = 372^{-1/2}M = 0.1633$. We can again go back to physical units (with $M_{\odot}=1$) with an arbitrary K . A common choice which leads to a masses and sizes similar to the observed for neutron stars is $K = 100$, so that $M = 100^{1/2}\bar{M} = 1.633M_{\odot}$.

The second method for choosing the length scale is explained in detail in [37], and is more involved. It is based on fixing the maximum stable mass for a family of solutions (with given $\{\kappa = 1, \Gamma\}$) to a physically motivated value. Thus, a quantity \hat{X} with dimensions $L^l M^m T^t$ is

obtained by using the relation:

$$\hat{X} = \hat{\kappa}^x c^y G^z X, \quad (282)$$

where

$$\begin{aligned} x &= \frac{l+m+t}{2(\Gamma-1)}, \quad y = \frac{(\Gamma-2)l + (3\Gamma-4)m - t}{\Gamma-1}, \\ z &= -\frac{l+3m+t}{2}. \end{aligned} \quad (282)$$

In this method $\hat{\kappa}$ has dimensions. We now identify the maximum stable mass for the given polytrope to some physical maximum mass. Although this second method for fixing the fundamental length scale generally leads to different results from the first, it can be checked that for $\Gamma = 2$ both methods (the first one with $\kappa = 100$, while the second one always has $\kappa = 1$) provide the same scaling factors when the physical maximum stable mass is set to $\hat{M} = 1.64M_{\odot}$. Since the dimensionless maximum stable mass is $M = 0.164$, Eq. (III I) can be solved for $\hat{\kappa}$ with $\{l = 0, m = 1, t = 0\}$, giving $\hat{\kappa} = 1.456 \times 10^5 \text{cm}^5 / (\text{g s}^2)$. With this value, (III I) can again be used to recover the dimensions of any quantity.

For the Lorene initial data for a single star (Magstar), the easiest thing to do is to calculate the corresponding $\hat{\kappa}$, considering that the maximum mass of the non-rotating star computed with numerically has a mass $M_{max} = 3.16$ and making it correspond in physical units to $M_{max} = 1.64M_{\odot}$. This way we obtain $\hat{\kappa} = 390.36 \text{cm}^5 / (\text{g s}^2)$. Then, we can convert all the unitless quantities to unit by using (III I), namely

$$\hat{T}(l=0, m=0, t=1) \Rightarrow (x=1/2, y=-1, z=-1/2) \Rightarrow \hat{T} = (2.55 \times 10^{-6}T)s = (2.55 \times 10^{-3}T)ms \quad (283)$$

$$\hat{M}(l=0, m=1, t=0) \Rightarrow (x=1/2, y=2, z=-3/2) \Rightarrow \hat{M} = (1.031 \times 10^{30}M)kg = (0.519M)M_{\odot} \quad (284)$$

$$\hat{L}(l=1, m=0, t=0) \Rightarrow (x=1/2, y=0, z=-1/2) \Rightarrow \hat{L} = (764L)m = (0.764L)km \quad (285)$$

$$\hat{\rho}(l=-3, m=1, t=0) \Rightarrow (x=-1, y=2, z=0) \Rightarrow \hat{\rho} = (2.3 \times 10^{21}\rho)kgm^{-3} = (2.3 \times 10^{18}\rho)g/cm^3 \quad (286)$$

$$\hat{B}(l=-1/2, m=1/2, t=-1) \Rightarrow (x=-1/2, y=2, z=0) \Rightarrow \hat{B} = \sqrt{8\pi}(4.5 \times 10^{19}B)G = (2.3 \times 10^{20}B)G \quad (287)$$

where we have used that the magnetic field pressure is $B^2/(8\pi)$ and has units in the cgs system of $dyn/cm^2 = g/(cm s^2)$.

We could repeat the same calculation for the Lorene initial data for a binary system with $K = p^{\Gamma}/\rho = 123$, rescale the adimensional maximum mass of the non-rotating star of mass 0.164 to be $M_{max} = 1.81$, and making it correspond in physical units to $M_{max} = 1.64M_{\odot}$.

This way we obtain $\hat{\kappa} = 1191.4 \text{cm}^5 / (\text{g s}^2)$.

Another way of setting the scale is to assume $M_{\odot} = 1$, so the mass is given naturally in solar mass units. We will follow this convention in general. The maximum mass for a non-rotating star with $K = 123$ is $M = 1.82$ (for the rotating is $M = 2.09$), which implies that $\hat{\kappa} = 1465.4 \text{cm}^5 / (\text{g s}^2)$. Then, we can convert all the unitless quantities to unit by using (III I), namely

$$\hat{T}(l=0, m=0, t=1) \Rightarrow (x=1/2, y=-1, z=-1/2) \Rightarrow \hat{T} = (4.94 \times 10^{-3} T) ms \quad (288)$$

$$\hat{M}(l=0, m=1, t=0) \Rightarrow (x=1/2, y=2, z=-3/2) \Rightarrow \hat{M} = M M_{\odot} \quad (289)$$

$$\hat{L}(l=1, m=0, t=0) \Rightarrow (x=1/2, y=0, z=-1/2) \Rightarrow \hat{L} = (1.48L) km \quad (290)$$

$$\hat{\rho}(l=-3, m=1, t=0) \Rightarrow (x=-1, y=2, z=0) \Rightarrow \hat{\rho} = (6.14 \times 10^{17} \rho) g/cm^3 \quad (291)$$

$$\hat{B}(l=-1/2, m=1/2, t=-1) \Rightarrow (x=-1/2, y=2, z=0) \Rightarrow \hat{B} = (2.3 \sqrt{8\pi} \times 10^{19} B) G \quad (292)$$

$$\hat{\mathcal{E}}(l=2, m=1, t=-2) \Rightarrow (x=1/2, y=4, z=-3/2) \Rightarrow \hat{\mathcal{E}} = (1.8 \times 10^{54} \mathcal{E}) erg \quad (293)$$

where we have used that the energetic units comes from $\mathcal{E} = Mc^2$.

A simple way to estimate the emission of a optically thick source is by means of the effective temperature T_{eff} of the corresponding black body, which is defined as

$$\mathcal{L} = 4\pi R^2 \sigma T_{eff}^4, \quad \sigma = 5.67 \times 10^{-5} ergs cm^{-2} s^{-1} K^{-4} \quad (294)$$

where σ is the Stefan-Boltzmann constant and \mathcal{L} the total luminosity of a source of size R . This effective temperature allows to compute the peak frequency of the black body radiation $\nu_{peak}(Hz) = 5.88 \times 10^{10} T(K)$. In order to know if the magnetosphere is really optically thick we need to know its density and temperature. The charge density can be estimated roughly from the Julian-Goldreich one $\nabla \cdot (\Omega r B)$. Dividing by the electric charge of the electron $e = 4.8 \times 10^{-10} statc$ and multiplying by its mass $m_e = 9.1 \times 10^{-28} g$ we can calculate the density,

$$\rho_{JG} = \frac{\Omega B m_e}{2\pi c e} [g/cm^3]. \quad (295)$$

For a $\Omega = 1.5 rad/ms$ and $B = 10^{12} G$ the density is just $1.5 \times 10^{-14} g/cm^3$.

APPENDIX B. WENO SCHEMES

1. Third-order WENO

Let us write explicitly the procedure for $k=2$, leading to a 3rd-order WENO reconstruction:

- The reconstructed values ${}^L U_{i+1/2}^{(r)}$ and ${}^R U_{i-1/2}^{(r)}$ of k^{th} -order accuracy are

$$\begin{aligned} {}^L U_{i+1/2}^{(0)} &= \frac{1}{2} U_i + \frac{1}{2} U_{i+1} \\ {}^L U_{i+1/2}^{(1)} &= -\frac{1}{2} U_{i-1} + \frac{3}{2} U_i \\ {}^R U_{i-1/2}^{(0)} &= -\frac{1}{2} U_{i+1} + \frac{3}{2} U_i \\ {}^R U_{i-1/2}^{(1)} &= \frac{1}{2} U_i + \frac{1}{2} U_{i-1} \end{aligned} \quad (293)$$

Notice that ${}^L U_{i-1/2}^{(r)}$ and ${}^R U_{i+1/2}^{(r)}$ can be obtained by substituting i by $i \pm 1$ in the previous expressions.

- We find the smooth indicators ${}^L \beta_{i+1/2}^{(r)}$ and ${}^R \beta_{i+1/2}^{(r)}$

$$\begin{aligned} {}^L \beta_{i+1/2}^{(0)} &= (U_{i+1} - U_i)^2 \\ {}^L \beta_{i+1/2}^{(1)} &= (U_i - U_{i-1})^2 \\ {}^R \beta_{i-1/2}^{(0)} &= (U_i - U_{i+1})^2 \\ {}^R \beta_{i-1/2}^{(1)} &= (U_{i-1} - U_i)^2 \end{aligned} \quad (291)$$

Again, ${}^L \beta_{i-1/2}^{(r)}$ and ${}^R \beta_{i+1/2}^{(r)}$ can be obtained by substituting i by $i \pm 1$ in the previous expressions.

- We find the 3rd-order reconstruction

$$\begin{aligned} U_{i+1/2}^L &= \omega_{i+1/2}^{(0)} {}^L U_{i+1/2}^{(0)} + \omega_{i+1/2}^{(1)} {}^L U_{i+1/2}^{(1)} \\ U_{i-1/2}^R &= \tilde{\omega}_{i-1/2}^{(0)} {}^R U_{i-1/2}^{(0)} + \tilde{\omega}_{i-1/2}^{(1)} {}^R U_{i-1/2}^{(1)} \end{aligned} \quad (291)$$

with weights $\omega_{i+1/2}^{(r)}$ and $\tilde{\omega}_{i-1/2}^{(r)}$ constructed by using the generic formulas

$$\omega_{i+1/2}^{(r)} = \frac{\alpha_{i+1/2}^{(r)}}{\sum_{s=0}^{k-1} \alpha_{i+1/2}^{(s)}}, \quad \tilde{\omega}_{i-1/2}^{(r)} = \frac{\tilde{\alpha}_{i-1/2}^{(r)}}{\sum_{s=0}^{k-1} \tilde{\alpha}_{i-1/2}^{(s)}} \quad (292)$$

where

$$\begin{aligned} \alpha_{i+1/2}^{(0)} &= \frac{2/3}{(\epsilon + {}^L \beta_{i+1/2}^{(0)})^2}, \quad \alpha_{i+1/2}^{(1)} = \frac{1/3}{(\epsilon + {}^L \beta_{i+1/2}^{(1)})^2} \\ \tilde{\alpha}_{i-1/2}^{(0)} &= \frac{1/3}{(\epsilon + {}^R \beta_{i-1/2}^{(0)})^2}, \quad \tilde{\alpha}_{i-1/2}^{(1)} = \frac{2/3}{(\epsilon + {}^R \beta_{i-1/2}^{(1)})^2} \end{aligned}$$

and ϵ is usually set to a very small number. The nominal expected convergence rate is achieved when $\epsilon = \Delta x^2$.

As it was mentioned before, the reconstructed values from the other cells can be found by substituting i by $i \pm 1$ in the previous expressions, namely

$$\begin{aligned} U_{i-1/2}^L &= \omega_{i-1/2}^{(0)} {}^L U_{i-1/2}^{(0)} + \omega_{i-1/2}^{(1)} {}^L U_{i-1/2}^{(1)} \\ U_{i+1/2}^R &= \tilde{\omega}_{i+1/2}^{(0)} {}^R U_{i+1/2}^{(0)} + \tilde{\omega}_{i+1/2}^{(1)} {}^R U_{i+1/2}^{(1)} \end{aligned} \quad (290)$$

2. Fifth-order WENO

Let us write explicitly the procedure for the 5th-order WENO, obtained with $k=3$.

- The k reconstructed values ${}^L U_{i+1/2}^{(r)}$ and ${}^R U_{i-1/2}^{(r)}$ of k^{th} -order accuracy are,

$$\begin{aligned}
{}^L U_{i+1/2}^{(0)} &= \frac{2}{6}U_i + \frac{5}{6}U_{i+1} - \frac{1}{6}U_{i+2} \\
{}^L U_{i+1/2}^{(1)} &= -\frac{1}{6}U_{i-1} + \frac{5}{6}U_i + \frac{2}{6}U_{i+1} \\
{}^L U_{i+1/2}^{(2)} &= \frac{2}{6}U_{i-2} - \frac{7}{6}U_{i-1} + \frac{11}{6}U_i \\
{}^R U_{i-1/2}^{(0)} &= \frac{2}{6}U_{i+2} - \frac{7}{6}U_{i+1} + \frac{11}{6}U_i \\
{}^R U_{i-1/2}^{(1)} &= -\frac{1}{6}U_{i+1} + \frac{5}{6}U_i + \frac{2}{6}U_{i-1} \\
{}^R U_{i-1/2}^{(2)} &= \frac{2}{6}U_i + \frac{5}{6}U_{i-1} - \frac{1}{6}U_{i-2}
\end{aligned} \tag{286}$$

The ${}^L U_{i-1/2}^{(r)}$ and ${}^R U_{i+1/2}^{(r)}$ can be obtained by substituting i by $i \pm 1$ in the previous expressions.

- We find the smooth indicators ${}^L \beta_{i+1/2}^{(r)}$ and ${}^R \beta_{i+1/2}^{(r)}$

$$\begin{aligned}
{}^L \beta_{i+1/2}^{(0)} &= \frac{13}{12}(U_i - 2U_{i+1} + U_{i+2})^2 \\
&\quad + \frac{1}{4}(3U_i - 4U_{i+1} + U_{i+2})^2 \\
{}^L \beta_{i+1/2}^{(1)} &= \frac{13}{12}(U_{i-1} - 2U_i + U_{i+1})^2 \\
&\quad + \frac{1}{4}(U_{i-1} - U_{i+1})^2 \\
{}^L \beta_{i+1/2}^{(2)} &= \frac{13}{12}(U_{i-2} - 2U_{i-1} + U_i)^2 \\
&\quad + \frac{1}{4}(U_{i-2} - 4U_{i-1} + 3U_i)^2 \\
{}^R \beta_{i-1/2}^{(0)} &= \frac{13}{12}(U_{i+2} - 2U_{i+1} + U_i)^2 \\
&\quad + \frac{1}{4}(U_{i+2} - 4U_{i+1} + 3U_i)^2 \\
{}^R \beta_{i-1/2}^{(1)} &= \frac{13}{12}(U_{i+1} - 2U_i + U_{i-1})^2 \\
&\quad + \frac{1}{4}(U_{i+1} - U_{i-1})^2 \\
{}^R \beta_{i-1/2}^{(2)} &= \frac{13}{12}(U_i - 2U_{i-1} + U_{i-2})^2 \\
&\quad + \frac{1}{4}(3U_i - 4U_{i-1} + U_{i-2})^2
\end{aligned} \tag{276}$$

Again, ${}^L \beta_{i-1/2}^{(r)}$ and ${}^R \beta_{i+1/2}^{(r)}$ can be obtained by substituting i by $i \pm 1$ in the previous expressions.

- We find the 5th-order reconstruction

$$\begin{aligned}
U_{i+1/2}^L &= \omega_{i+1/2}^{(0)} {}^L U_{i+1/2}^{(0)} + \omega_{i+1/2}^{(1)} {}^L U_{i+1/2}^{(1)} + \omega_{i+1/2}^{(2)} {}^L U_{i+1/2}^{(2)} \\
U_{i-1/2}^R &= \tilde{\omega}_{i-1/2}^{(0)} {}^R U_{i-1/2}^{(0)} + \tilde{\omega}_{i-1/2}^{(1)} {}^R U_{i-1/2}^{(1)} + \tilde{\omega}_{i-1/2}^{(2)} {}^R U_{i-1/2}^{(2)}
\end{aligned}$$

with weights $\omega_{i+1/2}^{(r)}$ and $\tilde{\omega}_{i+1/2}^{(r)}$ constructed by us-

ing the generic formulas

$$\omega_{i+1/2}^{(r)} = \frac{\alpha_{i+1/2}^{(r)}}{\sum_{s=0}^{k-1} \alpha_{i+1/2}^{(s)}}, \quad \tilde{\omega}_{i-1/2}^{(r)} = \frac{\tilde{\alpha}_{i-1/2}^{(r)}}{\sum_{s=0}^{k-1} \tilde{\alpha}_{i-1/2}^{(s)}} \tag{275}$$

by using

$$\begin{aligned}
\alpha_{i+1/2}^{(0)} &= \frac{3/10}{(\epsilon + {}^L \beta_{i+1/2}^{(0)})^2}, \quad \tilde{\alpha}_{i-1/2}^{(0)} = \frac{1/10}{(\epsilon + {}^R \beta_{i-1/2}^{(0)})^2} \\
\alpha_{i+1/2}^{(1)} &= \frac{6/10}{(\epsilon + {}^L \beta_{i+1/2}^{(1)})^2}, \quad \tilde{\alpha}_{i-1/2}^{(1)} = \frac{6/10}{(\epsilon + {}^R \beta_{i-1/2}^{(1)})^2} \\
\alpha_{i+1/2}^{(2)} &= \frac{1/10}{(\epsilon + {}^L \beta_{i+1/2}^{(2)})^2}, \quad \tilde{\alpha}_{i-1/2}^{(2)} = \frac{3/10}{(\epsilon + {}^R \beta_{i-1/2}^{(2)})^2}
\end{aligned}$$

where ϵ is usually set to a very small number.

Notice that we need also the reconstructed values from the other cells

$$\begin{aligned}
U_{i-1/2}^L &= \omega_{i-1/2}^{(0)} {}^L U_{i-1/2}^{(0)} + \omega_{i-1/2}^{(1)} {}^L U_{i-1/2}^{(1)} + \omega_{i-1/2}^{(2)} {}^L U_{i-1/2}^{(2)} \\
U_{i+1/2}^R &= \tilde{\omega}_{i+1/2}^{(0)} {}^R U_{i+1/2}^{(0)} + \tilde{\omega}_{i+1/2}^{(1)} {}^R U_{i+1/2}^{(1)} + \tilde{\omega}_{i+1/2}^{(2)} {}^R U_{i+1/2}^{(2)}
\end{aligned}$$

by substituting i by $i \pm 1$ in the previous expressions.

More recently there have been some improvements on the standard (or JS) WENO. One of them is the so-called WENO-Z [38], where the weights are changed by using

$$\begin{aligned}
\alpha_{i+1/2}^{(r)} &= d_r \left(1 + \left[\frac{{}^L \tau_{i+1/2}}{\epsilon + {}^L \beta_{i+1/2}^{(r)}} \right]^q \right) \\
\tilde{\alpha}_{i-1/2}^{(r)} &= \tilde{d}_r \left(1 + \left[\frac{{}^R \tau_{i-1/2}}{\epsilon + {}^R \beta_{i-1/2}^{(r)}} \right]^q \right)
\end{aligned} \tag{270}$$

where q is a coefficient between $[1, 2]$ and with ${}^L \tau_{i+1/2} = |{}^L \beta_{i+1/2}^{(0)} - {}^L \beta_{i+1/2}^{(2)}|$ and ${}^R \tau_{i-1/2} = |{}^R \beta_{i-1/2}^{(0)} - {}^R \beta_{i-1/2}^{(2)}|$. The scheme becomes more dissipative when the parameter q is increased. WENO-Z is 4th-order near simple smooth critical points (i.e., where $u'_j = 0$) for $q = 1$ and attains the designed 5th-order for $q = 2$, at the price of being more dissipative. For all these variants of fifth-order WENOs, the parameter ϵ is usually set to a very small number and the expected convergence rate is achieved if $\epsilon = \Delta x^4$.

APPENDIX C. BERGER-OLIGER WITHOUT ORDER REDUCTION (BOR)

Let us explain in detail the different steps of the BOR algorithm [35, 36], which is the most efficient but not common yet in the area. A direct Taylor expansion of the solution at $t = t^n$ leads to

$$U_{n+1} = U_n + \Delta t U'_n + \frac{1}{2} \Delta t^2 U''_n + \frac{1}{6} \Delta t^3 U'''_n + O(\Delta t^4) \tag{270}$$

By performing a similar expansion on the k_i of the RK we obtain

$$\begin{aligned}
k_1 &= \Delta t U'_n & (271) \\
k_2 &= \Delta t U'_n + c_2 \Delta t^2 U''_n + \frac{1}{2} c_2^2 \Delta t^3 [U'''_n - f_U U''_n] \\
k_3 &= \Delta t U'_n + c_3 \Delta t^2 U''_n \\
&\quad + \frac{1}{2} \Delta t^3 \left[c_3^2 U'''_n - \left(c_3^2 - 2 \sum_{j=1}^3 a_{3j} c_j \right) f_U U''_n \right] \\
k_4 &= \Delta t U'_n + c_4 \Delta t^2 U''_n \\
&\quad + \Delta t^3 \left[\frac{1}{2} c_4^2 U'''_n - \left(\frac{1}{2} c_4^2 - \sum_{j=1}^4 a_{4j} c_j \right) f_U U''_n \right]
\end{aligned}$$

where f_U is the Jacobian of f . Notice that one could solve now the derivatives of U in terms of k_i . However, the equations are not linearly independent and it is impossible to solve them. Instead, we will compute the derivatives here from the dense output interpolator Eqs. (5). Once we have these derivatives, we can calculate the k_i corresponding to the RK steps of the fine grid, that is, by doing $\Delta t \rightarrow \Delta t/2$ in Eqs. (271). From there we can calculate the solution at the different RK sub-steps required for the evolution of the boundary points of the fine grid. Next we will describe in detail the implementation for the two commonly-used RK scheme.

3. Standard fourth-order RK

Let us be more explicit and write down the steps for the standard 4th-order RK for an arbitrary space resolution ratio $R \equiv \Delta x / \Delta x_F$. First we define the time-step on the fine grid $\Delta t_F = \Delta t / R$. Then we start a loop over the steps on the fine grid, going from $r = 0, R - 1$:

1. define $t^{n+r/R} \equiv t + r \Delta t_F$ and evaluate the solution $U_{n+r/R}(t = t^{n+r/R})$ by using the dense output interpolator.
2. compute $\{U'_{n+r/R}, U''_{n+r/R}, U'''_{n+r/R}, f_U U''_{n+r/R}\}$ from the dense output interpolator as a function of $\{k_1, k_2, k_3, k_4\}$, that is, at $t = t^{n+r/R}$ or $\theta = r/R$. The Jacobian can be obtained directly from the k_i of the coarser grid by computing $f_U U''_{n+r/R} = 4(k_3 - k_2) / \Delta t^3$.
3. compute $\{k_1, k_2, k_3, k_4\}$ of the fine grid by using its

time-step Δt_F , namely

$$\begin{aligned}
k_1 &= \Delta t_F U'_{n+r/R} & (267) \\
k_2 &= \Delta t_F U'_{n+r/R} + \frac{1}{2} \Delta t_F^2 U''_{n+r/R} \\
&\quad + \frac{1}{8} \Delta t_F^3 [U'''_{n+r/R} - f_U U''_{n+r/R}] \\
k_3 &= \Delta t_F U'_{n+r/R} + \frac{1}{2} \Delta t_F^2 U''_{n+r/R} \\
&\quad + \frac{1}{8} \Delta t_F^3 [U'''_{n+r/R} + f_U U''_{n+r/R}] \\
k_4 &= \Delta t_F U'_{n+r/R} + \Delta t_F^2 U''_{n+r/R} + \frac{1}{2} \Delta t_F^3 U'''_{n+r/R}
\end{aligned}$$

4. use in each sub-step of the first RK step its intermediate value, that for our RK4 is

$$\begin{aligned}
U^{(1)} &= U_{n+r/R} & (263) \\
U^{(2)} &= U_{n+r/R} + \frac{1}{2} k_1 \\
U^{(3)} &= U_{n+r/R} + \frac{1}{2} k_2 \\
U^{(4)} &= U_{n+r/R} + k_3 \\
U^{n+\frac{r+1}{R}} &= U_{n+r/R} + \frac{1}{6} (k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}$$

The final RK step finalizes at t^{n+1} .

4. Strong Stability Preserving third-order RK

Let us write down now the procedure for the SSP 3rd-order RK for an arbitrary ratio R . First we define the time-step on the fine grid $\Delta t_F = \Delta t / R$. Then we start a loop over the steps on the fine grid, going from $r = 0, R - 1$:

1. define $t^{n+r/R} \equiv t + r \Delta t_F$ and evaluate the solution $U_{n+r/R}(t = t^{n+r/R})$ by using the dense output interpolator.
2. compute $\{U'_{n+r/R}, U''_{n+r/R}\}$ from the dense output interpolator as a function of $\{k_1, k_2, k_3\}$, that is, at $t = t^{n+r/R}$ or $\theta = r/R$.
3. compute $\{k_1, k_2, k_3\}$ of the fine grid by using its time-step Δt_F , namely

$$\begin{aligned}
k_1 &= \Delta t_F U'_{n+r/R} & (260) \\
k_2 &= \Delta t_F U'_{n+r/R} + \Delta t_F^2 U''_{n+r/R} \\
k_3 &= \Delta t_F U'_{n+r/R} + \frac{1}{2} \Delta t_F^2 U''_{n+r/R}
\end{aligned}$$

4. use in each sub-step of the first RK step its intermediate value, that for our RK3 is

$$\begin{aligned} U^{(1)} &= U_{n+r/R} & (259) \\ U^{(2)} &= U_{n+r/R} + k_1 \\ U^{(3)} &= U_{n+r/R} + \frac{1}{4}k_2 + \frac{1}{4}k_3 \\ U^{n+\frac{r+1}{R}} &= U_{n+r/R} + \frac{1}{6}(k_1 + 2k_2 + 4k_3) \end{aligned}$$

The final RK step finalizes at t^{n+1} .

5. Dense output interpolator

Notice that a dense output interpolator can be constructed by using the sub-steps of the RK [39]. Its generic form is

$$U^{n+\theta} = U^n + \sum_{j=1}^s b_j(\theta)k_j \quad , \quad \theta = \frac{t - t^n}{t^{n+1} - t^n} \quad (256)$$

where $b_i(\theta)$ are the coefficients to build the interpolator for a given RK scheme. Notice that the m -derivative can also be computed from this dense output interpolator as

$$\frac{d^m}{dt^m} U(t^n + \theta\Delta t) = \frac{1}{h^m} \sum_{j=1}^s k_j \frac{d^m}{d\theta^m} b_j(\theta) + O(h^{4-m}) \quad , \quad (256)$$

For the standard RK(4,4), it can be shown that there is a unique 3rd-order interpolator that can be written as

$$\begin{aligned} b_1(\theta) &= \theta - \frac{3}{2}\theta^2 + \frac{2}{3}\theta^3 \quad , \quad b_2(\theta) = b_3(\theta) = \theta^2 - \frac{2}{3}\theta^3 \quad , \\ b_4(\theta) &= \frac{-1}{2}\theta^2 - \frac{2}{3}\theta^3 \end{aligned} \quad (256)$$

There is a 2nd-order interpolator which also satisfies the Strong Stability Preserving (SSP) condition for the SSP-RK(3,3):

$$b_1(\theta) = \theta - \frac{5}{6}\theta^2 \quad , \quad b_2(\theta) = \frac{1}{6}\theta^2 \quad , \quad b_3(\theta) = \frac{4}{6}\theta^2 \quad (256)$$

Appendix A: Tabulated EOSs

1. Table format

Our EOS driver accepts tabulated EOS in **HDF5** format. Table V lists the required fields and a short description.

2. Shen EOS

a. Table Construction

Our Shen EOS is constructed on the basis of the Shen et al. 1998 relativistic-mean field nuclear EOS table.

Electrons (fully general, based on TimmesEOS) and Photons are added.

Original Shen EOS table extent:
Table extent of current table
[myshen_test_220r_180t_50y_extT_20090312.h5]:

This bigger table is realized by extending the original Shen table in multiple ways in multiple directions:

(a) density:

Match of pure ideal gas of Ni⁵⁶ + electrons/positrons + photons at densities below 10⁷g/cm³ – at this density pressures, energies and entropies match okayish with the values in the Shen table. The compositions (Abar,Zbar,Xh,Xa,Xp,Xn) are kept constant in the low-density region and mu_n and mu_p are set to 0 – ideally, at low densities, a full NSE EOS with nuclear reaction network (at low T) should be stitched onto the Shen; working on that, but not yet ready.

(b) temperature (extrapolation):

At high density: linear extrapolation of everything in T to lower temperatures and higher temperatures. At low densities (below 10⁷g/cm³), ideal gas of Ni⁵⁶ + electrons/positrons + photons.

b. Chemical Potentials

The nucleon chemical potentials are fully relativistic in the Shen EOS. They include the rest mass but are given with respect to a mass of $M = 938$ MeV, i.e. $\mu_n = \tilde{\mu}_n - M$. Therefore $\hat{\mu} = \mu_n - \mu_p$ includes the neutron-proton mass difference.

c. Energy Shift

In some regions the negative nuclear binding energy is larger in magnitude than the thermal/excitation energy. In this case the specific internal energy (ϵ) becomes negative. To allow for storage and interpolation of ϵ in logarithmic fashion, the energy is shifted up by an energy shift specified in the variable **energy_shift**. This energy shift is handled internally in the EOS routines.

3. LS EOSs

a. Chemical Potentials

The nucleon chemical potentials are fully relativistic in the LS EOSs in the sense that they include the rest mass of the particles. The chemical potentials are given with

Variable	Units	Description
<code>pointsrho</code>	dimensionless	number of table points in $\log_{10}(\rho)$
<code>pointstemp</code>	dimensionless	number of table points in $\log_{10}(T)$
<code>pointsye</code>	dimensionless	number of table points in Y_e
<code>logrho</code>	$\log_{10}(\rho[\text{g}/\text{cm}^3])$	index variable ρ
<code>logtemp</code>	$\log_{10}(T[\text{MeV}])$	index variable T
<code>ye</code>	number fraction	index variable Y_e
<code>Abar</code>	A	average heavy nucleus mass number
<code>Zbar</code>	Z	average heavy nucleus atomic number
<code>Xa</code>	mass fraction	α particle mass fraction
<code>Xh</code>	mass fraction	average heavy nucleus mass fraction
<code>Xn</code>	mass fraction	neutron mass fraction
<code>Xp</code>	mass fraction	proton mass fraction
<code>cs2</code>	cm^2/s^2	speed of sound squared
<code>dedt</code>	$\text{erg}/\text{g}/\text{MeV}$	C_v
<code>dpderho</code>	$\text{dynes g}/\text{cm}^2/\text{erg}$	$dP/d\epsilon$ at constant ρ
<code>dpdrhoe</code>	$\text{dynes cm}^3/\text{cm}^2/\text{g}$	$dP/d\rho$ at constant ϵ
<code>energy_shift</code>	erg/g	energy shift for table storage ^a
<code>entropy</code>	k_B/baryon	specific entropy
<code>gamma</code>	dimensionless	$d\log[P]/d\log[\rho]$
<code>logenergy</code>	$\log_{10}(\epsilon[\text{erg}/\text{g}])$	specific internal energy
<code>logpress</code>	$\log_{10}(P[\text{dynes}/\text{cm}^2])$	pressure
<code>mu_e</code>	MeV/baryon	electron chemical potential ^b
<code>mu_p</code>	MeV/baryon	proton chemical potential ^c
<code>mu_n</code>	MeV/baryon	neutron chemical potential ^d
<code>muhat</code>	Mev/baryon	$\mu_n - \mu_p$
<code>munu</code>	Mev/baryon	$\mu_e - \muhat$

^a see below

^b includes rest mass

^c includes rest mass, see specific EOS for details

^d includes rest mass, see specific EOS for details

TABLE V. EOS driver HDF5 variables

Density	$10^{5.1} - 10^{15.4} \text{ g}/\text{cm}^3$	includes the neutron-proton mass difference.
Temperature	0.1 - 100 MeV	
Y_e	0.01 - 0.56	

b. Energy Shift

Density	$10^3 - 10^{15.36} \text{ g}/\text{cm}^3$
Temperature	0.01 - 250 MeV
Y_e	0.015 - 0.56

In some regions the negative nuclear binding energy is larger in magnitude than the thermal/excitation energy. In this case the specific internal energy (ϵ) becomes negative. To allow for storage and interpolation of ϵ in logarithmic fashion, the energy is shifted up by an energy shift specified in the variable `energy_shift`. This energy shift is handled internally in the EOS routines.

respect to the neutron rest mass. Therefore $\hat{\mu} = \mu_n - \mu_p$

- [1] Daniela Alic, Carles Bona-Casas, Carles Bona, Luciano Rezzolla, and Carlos Palenzuela. Conformal and covariant formulation of the Z4 system with constraint-violation damping. *Phys. Rev. D*, 85(6):064040, Mar 2012.
- [2] Miguel Bezares, Carlos Palenzuela, and Carles Bona. Final fate of compact boson star mergers. *Phys. Rev. D*, 95:124005, Jun 2017.

- [3] C. Palenzuela, B. Miñano, D. Viganò, A. Arbona, C. Bona-Casas, A. Rigo, M. Bezares, C. Bona, and J. Massó. A Simflowny-based finite-difference code for high-performance computing in numerical relativity. *Classical and Quantum Gravity*, 35(18):185007, September 2018.
- [4] C. Palenzuela, S. L. Liebling, D. Neilsen, L. Lehner, O. L. Caballero, E. O'Connor, and M. Anderson. Effects of the microphysical equation of state in the mergers of magne-

- tized neutron stars with neutrino cooling. *Phys. Rev. D*, 92(4):044045, August 2015.
- [5] M. Brett Deaton, Matthew D. Duez, Francois Foucart, Evan O'Connor, Christian D. Ott, et al. Black Hole-Neutron Star Mergers with a Hot Nuclear Equation of State: Outflow and Neutrino-Cooled Disk for a Low-Mass, High-Spin Case. *Astrophys.J.*, 776:47, 2013.
- [6] Evan O'Connor and Christian D. Ott. A New Open-Source Code for Spherically-Symmetric Stellar Collapse to Neutron Stars and Black Holes. *Class.Quant.Grav.*, 27:114103, 2010.
- [7] G. Shen, C. J. Horowitz, and S. Teige. New equation of state for astrophysical simulations. *Phys. Rev. C*, 83(3):035802, March 2011.
- [8] J. M. Lattimer and F. Douglas Swesty. A generalized equation of state for hot, dense matter. *Nuclear Physics A*, 535:331–376, December 1991.
- [9] H. Shen, H. Toki, K. Oyamatsu, and K. Sumiyoshi. Relativistic Equation of State for Core-collapse Supernova Simulations. *ApJS*, 197:20, December 2011.
- [10] M. H. Ruffert, H. T. Janka, and Gerhard Schaefer. Coalescing neutron stars: A step towards physical models. I: Hydrodynamic evolution and gravitational-wave emission. *Astron. Astrophys.*, 311:532–566, 1996.
- [11] Stephan Rosswog and M. Liebendoerfer. High resolution calculations of merging neutron stars. 2: Neutrino emission. *Mon.Not.Roy.Astron.Soc.*, 342:673, 2003.
- [12] Federico Carrasco, Daniele Viganò, and Carlos Palenzuela. Gradient subgrid-scale model for relativistic mhd large-eddy simulations. *Physical Review D*, 101(6):063003, 2020.
- [13] Daniele Viganò, Ricard Aguilera-Miret, Federico Carrasco, Borja Miñano, and Carlos Palenzuela. General relativistic MHD large eddy simulations with gradient subgrid-scale model. *Phys. Rev. D*, 101(12):123019, June 2020.
- [14] Ricard Aguilera-Miret, Daniele Viganò, Federico Carrasco, Borja Miñano, and Carlos Palenzuela. Turbulent magnetic-field amplification in the first 10 milliseconds after a binary neutron star merger: Comparing high-resolution and large-eddy simulations. *Physical Review D*, 102(10):103006, 2020.
- [15] F. Galeazzi, W. Kastaun, L. Rezzolla, and J. A. Font. Implementation of a simplified approach to radiative transfer in general relativity. 88(6):064009, September 2013.
- [16] Yuichiro Sekiguchi, Kenta Kiuchi, Koutarou Kyutoku, and Masaru Shibata. Current Status of Numerical-Relativity Simulations in Kyoto. *PTEP*, 2012:01A304, 2012.
- [17] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. John Wiley and Sons, Ltd, 2008.
- [18] L. Pareschi and G. Russo. Implicit-explicit runge-kutta schemes and application to hyperbolic systems with relaxation. *J. Sci. Comput.*, 25:112, 2005.
- [19] Ran Chu, Eirik Endeve, Cory D. Hauck, Anthony Mezzacappa, and Bronson Messer. thornado-transport: IMEX schemes for two-moment neutrino transport respecting fermi-dirac statistics. *Journal of Physics: Conference Series*, 1225(1):012013, may 2019.
- [20] Gioel Calabrese. Finite differencing second order systems describing black hole spacetimes. *Phys. Rev.*, D71:027501, 2005.
- [21] Andrea Mignone, Petros Tzeferacos, and Gianluigi Bodo. High-order conservative finite difference GLM-MHD schemes for cell-centered MHD. *Journal of Computational Physics*, 229(17):5896–5920, August 2010.
- [22] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, 1997.
- [23] P. Colella and P. R. Woodward. The Piecewise Parabolic Method (PPM) for Gas-Dynamical Simulations. *Journal of Computational Physics*, 54:174–201, September 1984.
- [24] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.
- [25] Chi-Wang Shu. *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, pages 325–432. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998.
- [26] A. Suresh and H.T. Huynh. Accurate monotonicity-preserving schemes with runge-kutta time stepping. *Journal of Computational Physics*, 136(1):83 – 99, 1997.
- [27] C. Bona, C. Bona-Casas, and J. Terradas. Linear high-resolution schemes for hyperbolic conservation laws: TVB numerical evidence. *Journal of Computational Physics*, 228:2266–2281, April 2009.
- [28] H.S. Tang, W.B. Dong, and A. Agrawal. A phenomenon of artificial odd-even grid oscillation and its presence in domain decomposition computation: Algebraic analysis and numerical illustration. *Journal of Computational and Applied Mathematics*, 333:404–427, 2018.
- [29] A. Arbona, A. Artigues, C. Bona-Casas, J. Massó, B. Miñano, A. Rigo, M. Trias, and C. Bona. Simflowny: A general-purpose platform for the management of physical models and simulation problems. *Computer Physics Communications*, 184:2321–2331, October 2013.
- [30] A. Arbona, B. Miñano, A. Rigo, C. Bona, C. Palenzuela, A. Artigues, C. Bona-Casas, and J. Massó. Simflowny 2: An upgraded platform for scientific modelling and simulation. *Computer Physics Communications*, 229:170–181, August 2018.
- [31] Richard D. Hornung and Scott R. Kohn. Managing application complexity in the samrai object-oriented framework. *Concurrency and Computation: Practice and Experience*, 14(5):347–368, 2002.
- [32] Brian T.N. Gunney and Robert W. Anderson. Advances in patch-based adaptive mesh refinement scalability. *Journal of Parallel and Distributed Computing*, 89:65 – 84, 2016.
- [33] Kurt Sebastian and Chi-Wang Shu. Multidomain weno finite difference method with interpolation at subdomain interfaces. *J. Sci. Comput.*, 19:405–438, 12 2003.
- [34] Luis Lehner, Steven L. Liebling, and Oscar Reula. AMR, stability and higher accuracy. *Classical and Quantum Gravity*, 23(16):S421–S445, August 2006.
- [35] Peter McCorquodale and Phillip Colella. A high-order finite-volume method for conservation laws on locally refined grids. *Commun. Appl. Math. Comput. Sci.*, 6(1):1–25, 2011.
- [36] B. Mongwane. Toward a consistent framework for high order mesh refinement schemes in numerical relativity. *General Relativity and Gravitation*, 47:60, May 2015.
- [37] Scott Noble. *A numerical study of relativistic fluid collapse*. PhD thesis, The University of Texas at Austin, 2003. Appendix 1.
- [38] Rafael Borges, Monique Carmona, Bruno Costa, and Wai Sun Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws.

Journal of Computational Physics, 227(6):3191 – 3211, 2008.

[39] Ernst Hairer, Syvert P. Norsett, and Gerhard Wanner. *Solving Ordinary Differential Equations I*. Springer-Verlag Berlin Heidelberg, 1987.