Hybrid DG/FV schemes for magnetohydrodynamics and relativistic hydrodynamics

Jonatan Núñez-de La Rosa, Claus-Dieter Munz

Hybrid DG/FV schemes for magnetohydrodynamics and relativistic hydrodynamics

Jonatan Núñez-de la Rosa[a,b,*], Claus-Dieter Munz[b]

[a]*Departamento de Ingeniería Mecánica, Universidad del Norte*
Km 5 Vía Puerto Colombia, Barranquilla, Colombia
[b]*Institut für Aerodynamik und Gasdynamik, Universität Stuttgart*
Pfaffenwaldring 21, D-70569 Stuttgart, Germany

Abstract

This paper presents a high order hybrid discontinuous Galerkin/finite volume scheme for solving the equations of the magnetohydrodynamics (MHD) and of the relativistic hydrodynamics (SRHD) on quadrilateral meshes. In this approach, for the spatial discretization, an arbitrary high order discontinuous Galerkin spectral element (DG) method is combined with a finite volume (FV) scheme in order to simulate complex flow problems involving strong shocks. Regarding the time discretization, a fourth order strong stability preserving Runge–Kutta method is used. In the proposed hybrid scheme, a shock indicator is computed at the beginning of each Runge–Kutta stage in order to flag those elements containing shock waves or discontinuities. Subsequently, the DG solution in these troubled elements and in the current time step is projected onto a subdomain composed of finite volume subcells. Right after, the DG operator is applied to those unflagged elements, which, in principle, are oscillation-free, meanwhile the troubled elements are evolved with a robust second/third order FV operator. With this approach we are able to numerically simulate very challenging problems in the context of MHD and SRHD in one, and two space dimensions and with very high order polynomials. We make convergence tests and show a comprehensive one- and two dimensional testbench for both equation systems, focusing in problems with strong shocks. The presented hybrid approach shows that numerical schemes of very high order of accuracy are able to simulate these complex flow problems in an efficient and robust manner.

Keywords: discontinuous Galerkin spectral element methods; magnetohydrodynamics; shock capturing; divergence cleaning; high-order methods

1. Introduction

High order numerical methods for solving partial differential equations are becoming the new standard in computational science and engineering because of the advantages provided by this family of numerical schemes. Among the high-order methods, the discontinuous Galerkin schemes are the most renowned candidates and are playing a central role because of all the attractive properties for solving the equations of fluid dynamics on complex geometries, and in general, nonlinear hyperbolic systems of conservation laws [1–3]. In these schemes, high-order accuracy is obtained through the approximation of the solution with a higher degree polynomial. The method has the capacity to handle complicated geometries, and due to its locality, it is highly parallelizable. In 1971, the discontinuous Galerkin method was originally introduced by Nitsche [4] for the approximation of elliptic equations. In 1973, Reed and Hill [5] proposed the discontinuous Galerkin method for solving the steady-state neutron transport equation. The method was recovered by Cockburn and Shu [6] to solve time dependent non-linear conservation laws. In the subsequent years, in a series of papers, the general framework of the discontinuous Galerkin for nonlinear systems of conservation laws, even in multiple space dimensions was established [7–9].

*Corresponding author
Email addresses: nunez@iag.uni-stuttgart.de (Jonatan Núñez-de la Rosa), munz@iag.uni-stuttgart.de (Claus-Dieter Munz)

High-order methods are not as robust as low-order methods. Because of hyperbolic conservation laws allow discontinuities in their solution, these will produce spurious oscillations in the case the solution is computed with a high-order scheme. This phenomenon is called the Gibbs phenomenon [10, 11]. As a consequence, the oscillations will create unphysical states, and destroy conservation. Shock capturing methods have to be constructed and adapted to a high order numerical scheme in order to overcome this difficulty. In the case of finite difference and finite volume methods, the most used shock capturing strategies are those that employ slope/flux limiters [12, 13], and an adaptive reconstruction operator [14, 15]. For discontinuous Galerkin schemes, several approaches have been developed, for instance the *generalized slope limiter* [6], the *moment limiter* [16, 17], and the *artificial diffusion* [18, 19]. The family of WENO reconstruction methods used as limiters in the discontinuous Galerkin methods has been well studied in the last years [20–22]. The main drawback of the like-WENO limiters is that they have a huge computational overhead when high-order approximations are used.

A very promising approach that allows to solve conservation laws even when shocks/discontinuities take place consists in hybridizing high order numerical schemes with robust low order methods, and in this way we take advantage of the best features of each of them. For instance, a hybrid discontinuous Galerkin/finite volume scheme is an excellent approach to tackle complex flows with shocks [23–26]. The main idea behind these schemes is to use a high order discontinuous Galerkin scheme in flow regions where shocks and discontinuities are absent, and to employ a robust finite volume method in flow regions with such shocks/discontinuities. A more complete strategy consists in efficiently detecting troubled regions (with shocks, discontinuities, violation of positivity of some quantities of interest like pressure and density) and then applying an adaptive mesh refinement algorithm in those regions along with a robust finite volume scheme. Furthermore, this finite volume scheme can also have an adaptive reconstruction operator that lower its order in the extreme case it cannot obtain a physically correct solution. As last option in this adaptation, the algorithm will use a first order Godunov scheme. In this work we follow a simple, efficient and robust strategy based on the works of [24, 26, 27], and it is essentially a combination of a high-order DG method with a robust FV-WENO scheme on quadrilateral meshes. In the proposed hybrid scheme, an oscillations indicator is used to mark those elements containing discontinuities. The DG solution in these troubled elements and in the current time step is projected in to a subdomain made of finite-volume subcells, which are then evolved with a robust FV method. The cells without oscillations are evolved with a high-order DG method.

Several high resolution shock-capturing and high order numerical formulations have been constructed in the last years to solve the MHD and SRHD equations. For the MHD equations the finite difference methods [28, 29], and the finite volume methods [30–33] have been the most widely used. For the SRHD equations several high resolution shock capturing schemes based on finite difference and finite volume schemes have been successfully developed [34–36], and also high order schemes [37–40]. The hybrid DG/FV scheme presented in this work will be used to solve these equations in an efficient and robust manner, and using very high order polynomials. In the case of the MHD equations, besides the shock capturing strategy, and additional stabilization procedure has to be developed in order to maintain the solenoidal constraint [41, 42]. The constraint $\nabla \cdot \boldsymbol{B} = 0$ is not satisfied by the numerical scheme, even for high order methods. The most used approaches to keep it are the *divergence cleaning* and the *constrained transport* algorithms. The main idea of the divergence cleaning is to remove the numerical errors produced in the computation of $\nabla \cdot \boldsymbol{B}$. The Hodge projection [41], the 8-wave formulation [42], and the generalized Lagrange multiplier (GLM) approach [43] belong to the family of divergence cleaning. The constrained transport schemes preserve the solenoidal constraint of the magnetic field with machine accuracy, but introducing a staggered mesh [44]. In order to keep as simple as possible the numerical method, in this work we will use the GLM approach for controlling the solenoidal constraint of the magnetic field. In this approach a new scalar field ψ is introduced and couples the solenoidal constraint with the evolution equation of the magnetic field. The divergence errors produced during the numerical simulation are propagated to the boundary of the computational domain. This scheme is rather simple to implement, and the produced errors in the computation are small enough to guarantee conservation of all physical variables. In our algorithm we use the mixed hyperbolic/parabolic divergence cleaning as it is reported in [45].

The structure of this paper is as follows: in section 2 we briefly describe the MHD and SRHD equations written in conservation form. The GLM divergence cleaning procedure is also taken into consideration in this section. In section 3 the high order hybrid DG/FV method for conservation laws is discussed in detail. In section 4 several test problems are shown and analyzed. This includes convergence tests, shock tube problems and two-dimensional flow configurations with shocks for the MHD and SRHD equations. In section 5 the conclusions of this work are presented.

2. Governing equations

2.1. MHD equations

Magnetohydrodynamics is a macroscopic theory concerned with the study of magnetic fields interacting with electrically conducting fluids, ranging from laboratory to space plasmas. The ideal MHD equations are usually derived as a combination of the Euler equations of the hydrodynamics and the Ampère's law, but neglecting the displacement current. As a system of conservation laws, the ideal MHD equations consist of the conservation of mass, the conservation of momentum, the conservation of energy, and the induction equations:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \boldsymbol{S} = 0, \tag{1a}$$

$$\frac{\partial \boldsymbol{S}}{\partial t} + \nabla \cdot (\boldsymbol{S} \otimes \boldsymbol{v} + \mathbf{P}) = \boldsymbol{0}, \tag{1b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (E\boldsymbol{v} + \mathbf{P} \cdot \boldsymbol{v}) = 0, \tag{1c}$$

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \cdot (\boldsymbol{B} \otimes \boldsymbol{v} - \boldsymbol{v} \otimes \boldsymbol{B}) = \boldsymbol{0}. \tag{1d}$$

The conservative variables are the mass density ρ, the momentum density $\boldsymbol{S}$, the energy density E and the magnetic field $\boldsymbol{B}$. The magnetic field is subject to the solenoidal constraint, that is

$$\nabla \cdot \boldsymbol{B} = 0. \tag{2}$$

The pressure tensor in equations (1b) and (1c) includes the hydrodynamic and the magnetic pressure

$$\mathbf{P} = \left(p + \frac{1}{2}|\boldsymbol{B}|^2\right)\mathbf{I} - \boldsymbol{B} \otimes \boldsymbol{B}. \tag{3}$$

Finally, an equation of state (EOS) is used to close the system. In this work we only consider the ideal gas EOS with adiabatic exponent γ

$$p = \rho\epsilon(\gamma - 1), \tag{4}$$

where ϵ is the specific internal energy. This EOS can be written as a function of the total energy E, and the primitive variables, ρ, $\boldsymbol{v}$, and $\boldsymbol{B}$

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho|\boldsymbol{v}|^2 - \frac{1}{2}|\boldsymbol{B}|^2\right). \tag{5}$$

2.1.1. Divergence cleaning

The solenoidal constraint given in equation (2) is a condition that the numerical method has to maintain in order to obtain physically correct solutions. In this work we apply the GLM divergence cleaning because this is a simple and reliable method [43, 46]. The mixed hyperbolic/parabolic divergence cleaning is used in this work [43]. This scheme introduces a scalar field ψ in order to couple the solenoidal constraint (2) with the induction equation (1d). As result, the following equations are obtained

$$\frac{\partial \boldsymbol{B}}{\partial t} + \nabla \cdot (\boldsymbol{B} \otimes \boldsymbol{v} - \boldsymbol{v} \otimes \boldsymbol{B} + \psi\mathbf{I}) = \boldsymbol{0}, \tag{6a}$$

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \left(c_h^2 \boldsymbol{B}\right) = -\frac{c_h^2}{c_p^2}\psi. \tag{6b}$$

The MHD-GLM system consists in equations (1a), (1b), (1c), (6a), and (6b). Observe that this system is conservative (except for the equation for the unphysical scalar field ψ).

2.1.2. Eigenvalues of the MHD-GLM system

We solve the MHD equations with an explicit numerical scheme. Therefore, the computation of the eigenvalues is required in order to compute the current time step (see section 3.3). From the Jacobian matrices for the MHD-GLM equations we can get the eigenvalues of the MHD-GLM system. This matrices, denoted by $\boldsymbol{A}_\alpha$, are defined by

$$\boldsymbol{A}_\alpha = \frac{\partial \mathbf{f}_\alpha(\boldsymbol{u})}{\partial \boldsymbol{u}}, \quad (\alpha = x, y, z), \tag{7}$$

with $\boldsymbol{u}$ the state vector of conservative variables (along the scalar field ψ) and $\mathbf{f} \equiv [\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}]$ the tensor of physical fluxes. The fluxes $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$ are respectively the flux in the x-direction, in the y-direction, and in the z-direction. The eigenvalues of the Jacobian matrices in the x-direction, $\boldsymbol{A}_x$, in nondecreasing order, are

$$\begin{aligned} &\lambda_1 = -c_h, \quad \lambda_2 = v_x - c_f, \quad \lambda_3 = v_x - c_a, \quad \lambda_4 = v_x - c_s, \\ &\lambda_5 = v_x, \quad \lambda_6 = v_x + c_s, \quad \lambda_7 = v_x + c_a, \quad \lambda_8 = v_x + c_f, \quad \lambda_9 = c_h, \end{aligned} \tag{8}$$

where c_f, c_a, and c_s are the fast, Alfvén, and slow characteristic speeds. The cases for the y and z directions can be easily derived from symmetry. In equation (8) we can see that there are three MHD waves families associated with the characteristic speeds and an entropy mode. The three characteristic speeds are given by

$$c_a = |b_x|, \quad c_s = \sqrt{\frac{1}{2}\left(a^2 + b^2 - \sqrt{\left(a^2 + b^2\right)^2 - 4a^2 b_x^2}\right)}, \quad c_f = \sqrt{\frac{1}{2}\left(a^2 + b^2 + \sqrt{\left(a^2 + b^2\right)^2 - 4a^2 b_x^2}\right)}, \tag{9}$$

where the following abbreviations were used

$$a^2 = \frac{\gamma p}{\rho}, \quad b^2 = \frac{|\boldsymbol{B}|^2}{\rho}, \quad b_x^2 = \frac{B_x^2}{\rho}. \tag{10}$$

Two new eigenvalues appear in the MHD-GLM equations, namely $\lambda_{1,9} = \pm c_h$. These eigenmodes transport the divergence errors to the boundaries at speed c_h, and are also damped at a rate c_h^2/c_p^2. Following [43], the value of the constant c_p is chosen after setting $c_r \equiv c_p^2/c_h = 0.18$, and the magnitude of the wave speeds c_h is set to the maximum allowed speed in the ideal MHD equations (without divergence cleaning)

$$c_h = \max_\alpha(\max\left(|\lambda_{2,\alpha}|, |\lambda_{8,\alpha}|\right)), \quad \alpha = x, y, z. \tag{11}$$

2.2. SRHD equations

Taking into account the effects of the special relativity, the SRHD equations are a generalization of the Euler equations. A relativistic fluid may have as main features at least one of the following properties: (1) the fluid velocity is close to the speed of light, and (2) the internal energy of the fluid is greater or approximate to its rest mass density [47]. The SRHD equations, written as a system of conservation laws [48, 49], and taking the speed of light $c = 1$, are given by

$$\frac{\partial D}{\partial t} + \nabla \cdot (D\boldsymbol{v}) = 0, \tag{12a}$$

$$\frac{\partial \boldsymbol{S}}{\partial t} + \nabla \cdot (\boldsymbol{S} \otimes \boldsymbol{v} + p\mathbf{I}) = \boldsymbol{0}, \tag{12b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \boldsymbol{S} = 0. \tag{12c}$$

The conserved quantities D, $\boldsymbol{S}$ and E are the mass density, the momentum density, and the total energy density, respectively. They are all measured in the laboratory frame, and are related to quantities in the local rest frame of the fluid, the so-called primitive variables (the rest-mass density ρ, the gas pressure p, and the specific enthalpy h), through

$$D = \rho\Gamma, \tag{13a}$$

$$\boldsymbol{S} = \rho h \Gamma^2 \boldsymbol{v}, \tag{13b}$$

$$E = \rho h \Gamma^2 - p, \tag{13c}$$

where the Γ is the Lorentz factor defined by

$$\Gamma = \frac{1}{\sqrt{1-v^2}}. \tag{14}$$

The SRHD system (12) is closed with an EOS: $h = h(p,\rho)$, also written as $p = p(\rho,\varepsilon)$. The SRHD system is hyperbolic for causal EOS [47, 50], i.e., for those where the local sound speed satisfies $c_s < 1$, where c_s is defined by

$$hc_s^2 = -\rho\frac{\partial h}{\partial \rho}\left(\rho\frac{\partial h}{\partial p} - 1\right)^{-1}. \tag{15}$$

In this work, we employ an ideal gas EOS,

$$p = (\gamma-1)(\varepsilon-\rho), \quad \text{or} \quad h = 1 + \frac{\gamma}{\gamma-1}\frac{p}{\rho}. \tag{16}$$

Here $\gamma = c_p/c_v$ is the ratio of specific heats and ε is the sum of the internal and rest-mass energy densities in the local frame and is related to the specific enthalpy as

$$h = \frac{\varepsilon + p}{\rho}. \tag{17}$$

With this choice of EOS, the speed of sound is given by

$$c_s^2 = \frac{\gamma p}{\rho h}. \tag{18}$$

2.2.1. Eigenvalues of the SRHD system

The eigenvalues of the SRHD equations are also required for the explicit numerical scheme for computing the time step. The Jacobian matrices $\boldsymbol{A}_\alpha$ for the SRHD equations are defined by

$$\boldsymbol{A}_\alpha = \frac{\partial \mathbf{f}_\alpha(\boldsymbol{u})}{\partial \boldsymbol{u}}, \quad (\alpha = x, y, z), \tag{19}$$

where $\boldsymbol{u}$ is the state vector of conservative variables, and $\mathbf{f} = [\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}]$ the tensor of physical fluxes. Assuming an ideal gas EOS, the eigenvalues of the matrix $\boldsymbol{A}_x$ are given by

$$\lambda_1 = \frac{1}{1-\boldsymbol{v}^2 c_s^2}\left\{v_x(1-c_s^2) - c_s\sqrt{(1-\boldsymbol{v}^2)\left[1 - v_x v_x - (\boldsymbol{v}^2 - v_x v_x)c_s^2\right]}\right\}, \tag{20}$$

$$\lambda_2 = v_x, \quad \lambda_3 = v_x, \quad \lambda_4 = v_x, \tag{21}$$

$$\lambda_5 = \frac{1}{1-\boldsymbol{v}^2 c_s^2}\left\{v_x(1-c_s^2) + c_s\sqrt{(1-\boldsymbol{v}^2)\left[1 - v_x v_x - (\boldsymbol{v}^2 - v_x v_x)c_s^2\right]}\right\}. \tag{22}$$

The cases for the matrices $\boldsymbol{A}_y$ and $\boldsymbol{A}_z$ easily follows from symmetry. The eigenvalues of $\boldsymbol{A}_x$ represent the five characteristic speeds associated with two sound wave modes (λ_1 and λ_5) and three entropy modes ($\lambda_2, \lambda_3, \lambda_4$). The eigenvalues satisfy the following relation: $\lambda_1 < \lambda_2 = \lambda_3 = \lambda_4 < \lambda_5$.

2.2.2. Conservative to primitive variables conversion

In the SRHD system, the relation between the primitive and conservative variables is quite complex. In this work, two strategies for expressing the primitive variables in terms of the conservative ones are followed: the first one, by solving iteratively an equation for the pressure and then determining the other variables from this [48]. The second one, which is more robust, consists in solving a quartic equation on the velocity v (as it is outlined in [51, 52], and [49]). We combine both strategies in our algorithm, using the iterative scheme as base scheme, and when this fails, the robust approach is used. The details of both algorithms can be found in [48, 49].

3. Numerical methods

3.1. Discontinuous Galerkin methods

Let us consider the following system of conservation laws

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} + \frac{\partial \boldsymbol{g}(\boldsymbol{u})}{\partial y} + \frac{\partial \boldsymbol{h}(\boldsymbol{u})}{\partial z} = \boldsymbol{0}, \tag{23}$$

also written as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \mathbf{f}(\boldsymbol{u}) = \boldsymbol{0}, \tag{24}$$

where $\boldsymbol{u} \in \mathbb{R}^m$ is the vector of *conserved* quantities, and $\mathbf{f} = [\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}]$ is the tensor of physical fluxes, with $\boldsymbol{f}$, $\boldsymbol{g}$, $\boldsymbol{h} \in \mathbb{R}^m$ being the *flux functions* in the directions x, y, and z, respectively. Next let us consider the test function $\phi = \phi(\boldsymbol{x}) = \phi(x, y, z)$, which is an arbitrary differentiable function defined in a subset $\Omega_k \subset \Omega \subset \mathbb{R}^3$. We assume that the subset Ω_k corresponds to a cell in the discontinuous Galerkin scheme. The variational formulation consists in projecting the residual of the partial differential equation (24) on the test function $\phi(\boldsymbol{x})$ and demanding that this projection is zero

$$\int_{\Omega_k} \left(\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_{\boldsymbol{x}} \cdot \mathbf{f}(\boldsymbol{u}) \right) \phi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \boldsymbol{0}. \tag{25}$$

The weak formulation of the equation (24) is obtained by integrating by parts the flux term in the equation (25)

$$\int_{\Omega_k} \frac{\partial \boldsymbol{u}}{\partial t} \phi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} + \oint_{\partial \Omega_k} \mathbf{f} \cdot \boldsymbol{n} \, \phi(\boldsymbol{x}) \, \mathrm{d}s - \int_{\Omega_k} \mathbf{f} \cdot \nabla \phi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x} = \boldsymbol{0}. \tag{26}$$

So far, no approximation has been done on the vector $\boldsymbol{u}$ or the fluxes $\boldsymbol{f}, \boldsymbol{g}, \boldsymbol{h}$. Depending in the discretization ansatz on these functions, we have different *flavors* of the discontinuous Galerkin method. In this work we will focus on the very efficient discontinuous Galerkin spectral element method (DGSEM) developed by Kopriva and Kolias [53], Kopriva [54]. For an outstanding description and derivation of the method and how it is implemented, we recommend the book by Kopriva [3]. In a recent work the DGSEM has been applied to solve the Navier–Stokes equations [55]. In the DGSEM framework, a mapping from the *physical* domain, where we want the solution, to the *reference* domain, where actually the *equation* is solved, is constructed. With the help of this transformation, we map the equation to the reference domain and finally we solve the transformed equation in the reference domain. We stress that in the general formulation of the discontinuous Galerkin, it is usual to solve the original equation in the physical space, where the involved integrals are mapped to the reference element. In our case we map first the equation to the reference element and then we solve it there. Therefore, we apply the discontinuous Galerkin formulation (26) not to the original equation (24), but onto the mapped equation.

3.1.1. Coordinate transformation

Let us start by considering the domain $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$. We decompose Ω in K non-overlapping hexahedral elements Ω_k, which are further mapped onto the reference element $[-1, 1]^3$ (see figure 1). A point in the physical space is described by $\boldsymbol{x} = (x, y, z) = (x_1, x_2, x_3)$, and a point in the reference element by $\xi = (\xi, \eta, \zeta) = (\xi^1, \xi^2, \xi^3)$. The mapping between the physical and the reference element is defined as $\boldsymbol{x} = \boldsymbol{X}(\xi)$. Let us consider a function $f(\boldsymbol{x}) = (x, y, z)$. Under a coordinate transformation, the derivative on the reference space is $\nabla_\xi f = \boldsymbol{J} \nabla_x f$, where $\boldsymbol{J}$ is the Jacobi matrix of the transformation $\boldsymbol{x} = \boldsymbol{X}(\xi)$. Because we want to map the equations from the physical space to the reference space, in practice, the coordinate transformation we need is the inverse, namely, $\xi = \boldsymbol{X}^{-1}(\boldsymbol{x})$, which is represented in matrix form by $\boldsymbol{J}^{-1}$.

Any point in the physical space can be represented by using two different basis vectors: The *covariant* basis vectors $\boldsymbol{e}_i$, for $i = 1, 2, 3$, and the *contravariant* basis vectors $\boldsymbol{e}^i$, for $i = 1, 2, 3$. By definition, the covariant vectors are tangent to the coordinate lines, while the contravariant are normal the coordinate lines. That is, they are defined by the following expressions

$$\boldsymbol{e}_i = \frac{\partial \boldsymbol{x}}{\partial \xi^i}, \quad i = 1, 2, 3, \quad \text{(covariant vectors)}, \tag{27a}$$

$$\boldsymbol{e}^i = \nabla \xi^i, \quad i = 1, 2, 3, \quad \text{(contravariant vectors)}. \tag{27b}$$

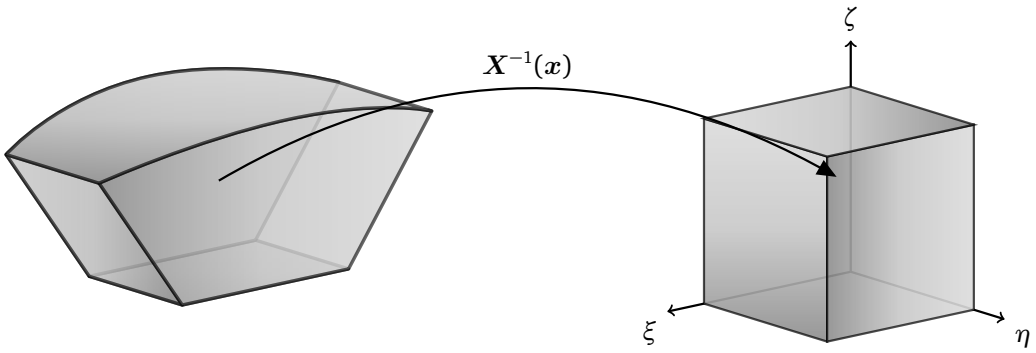

Figure 1: Mapping from the physical element Ω_i to the reference element $[-1, 1]^3$.

Vectors can be written either in covariant basis or in contravariant basis. That is possible because these bases form a base of the three-dimensional space. Now, coming back to the inverse of the Jacobi matrix, we use the definition of covariant basis vectors and we then introduce them in the expression for $\boldsymbol{J}$, yielding

$$\boldsymbol{J} = (\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3)^\intercal. \tag{28}$$

Using the general formula for the inverse of a 3×3 matrix expressed in terms of the cross product and triple product, we get

$$\boldsymbol{J}^{-1} = \frac{1}{\det(\boldsymbol{J})} \begin{pmatrix} (\boldsymbol{e}_2 \times \boldsymbol{e}_3)^\intercal \\ (\boldsymbol{e}_3 \times \boldsymbol{e}_1)^\intercal \\ (\boldsymbol{e}_1 \times \boldsymbol{e}_2)^\intercal \end{pmatrix}, \tag{29}$$

where the Jacobian $\mathcal{J} = \det(\boldsymbol{J})$ is given by

$$\mathcal{J} = \det(\boldsymbol{J}) = \boldsymbol{e}_i \cdot (\boldsymbol{e}_j \times \boldsymbol{e}_k). \tag{30}$$

From this last equation, we can easily see that the contravariant vectors $\boldsymbol{e}^i = \nabla\xi^i$ are related with the covariant vectors $\boldsymbol{e}_i = \dfrac{\partial \boldsymbol{x}}{\partial \xi^i}$ through

$$\mathcal{J}\boldsymbol{e}^i = \boldsymbol{e}_j \times \boldsymbol{e}_k, \quad i, j, k \text{ are cyclic.} \tag{31}$$

3.1.2. Conservation laws under mappings

The divergence operator in the conservation law (24) has been applied in the physical space. This operator under the coordinate transformation $\boldsymbol{x} = \boldsymbol{X}(\xi)$ is given by

$$\nabla_{\boldsymbol{x}} \cdot \mathbf{f} = \frac{1}{\mathcal{J}} \sum_{i=1}^{3} \frac{\partial}{\partial \xi^i} (\mathcal{J}\boldsymbol{e}^i \cdot \mathbf{f}) := \frac{1}{\mathcal{J}} \nabla_\xi \cdot \tilde{\mathbf{f}}, \tag{32}$$

with $\tilde{\mathbf{f}} = [\boldsymbol{f}^1, \boldsymbol{f}^2, \boldsymbol{f}^3]$ the tensor of contravariant fluxes and it is defined through its components by $\boldsymbol{f}^i \equiv (\mathcal{J}\boldsymbol{e}^i \cdot \mathbf{f})$. The conservation law in the reference element takes the form

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{\mathcal{J}} \nabla_\xi \cdot \tilde{\mathbf{f}} = \boldsymbol{0}. \tag{33}$$

Observe that $\boldsymbol{u} = \boldsymbol{u}(\xi, t)$, $\tilde{\mathbf{f}} = \tilde{\mathbf{f}}(\boldsymbol{u}(\xi, t))$, and $\mathcal{J} = \mathcal{J}(\xi)$. The variational formulation of the transformed conservation law is, after multiplying with a test function $\phi = \phi(\xi)$,

$$\int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + \frac{1}{\mathcal{J}} \nabla_\xi \cdot \tilde{\mathbf{f}} \right) \phi \, \mathrm{d}\boldsymbol{x} = \boldsymbol{0}. \tag{34}$$

Transforming the integral to the reference element $E = [-1,1]^3$, and reminding that $\mathrm{d}\boldsymbol{x} = \mathcal{J}(\xi)\,\mathrm{d}\xi$, we obtain the variational formulation of the transformed equation on the reference element

$$\int_E \left(\mathcal{J}(\xi)\frac{\partial \boldsymbol{u}}{\partial t} + \nabla_\xi \cdot \tilde{\mathbf{f}}\right)\phi(\xi)\,\mathrm{d}\xi = \boldsymbol{0}. \tag{35}$$

Finally, the weak formulation of the equation (33) is obtained by integrating by parts the flux term in the equation (35)

$$\int_E \mathcal{J}(\xi)\frac{\partial \boldsymbol{u}}{\partial t}\phi(\xi)\;\mathrm{d}\xi - \int_E \tilde{\mathbf{f}}\cdot\nabla_\xi\phi(\xi)\;\mathrm{d}\xi + \oint_{\partial E}\left[\tilde{\mathbf{f}}\cdot\tilde{\boldsymbol{n}}\right]^*\phi(\xi)\;\mathrm{d}\sigma = \boldsymbol{0}, \tag{36}$$

where $\tilde{\boldsymbol{n}}$ is the unit normal vector to the surface at the faces of the reference element. Observe that for a given direction in the reference space (e.g. $\tilde{\boldsymbol{n}}$ can be the normal unit vector at the face of the reference element in the direction $+\xi^i$) we have that

$$\tilde{\mathbf{f}}\cdot\tilde{\boldsymbol{n}} = \mathbf{f}\cdot\boldsymbol{n} = (\mathbf{f}\cdot\hat{\boldsymbol{n}})\|\boldsymbol{n}\|, \tag{37}$$

where $\boldsymbol{n}$ is the normal vector at the surface in the physical space, which is computed from the metric terms evaluated at the surfaces, and $\hat{\boldsymbol{n}}$ is the unit normal vector $\hat{\boldsymbol{n}} = \boldsymbol{n}/\|\boldsymbol{n}\|$; the components of the vector $\boldsymbol{n}$ are calculated from

$$n_l = \sum_{i=1}^{3}\mathcal{J}e_l^i\tilde{n}^i, \quad l = 1,2,3. \tag{38}$$

Its norm represents the area of the face element in the physical space and it is given by

$$\sigma = \|\boldsymbol{n}\| = \sqrt{\sum_{l=1}^{3}(\mathcal{J}e_l^i)^2}. \tag{39}$$

The superindex i means that $\boldsymbol{n}$ is associated with $\tilde{\boldsymbol{n}}$ in the direction of the coordinate ξ^i as we pointed out before. We will not use indices for this denomination and henceforth it will be understood from the context which direction it corresponds. Therefore

$$\left[\tilde{\mathbf{f}}\cdot\tilde{\boldsymbol{n}}\right]^* = \left[\mathbf{f}\cdot\boldsymbol{n}\right]^* = \left[\mathbf{f}\cdot\hat{\boldsymbol{n}}\right]^*\sigma. \tag{40}$$

At cell interfaces the solution is double-valued (one value is obtained from the data inside the current cell and the other value corresponds to the value calculated by the neighboring cell), just like the in the finite volume method. This discontinuity at cell interfaces implies that the flux has to be calculated by introducing the so-called *numerical flux*, and we write that in the variational formulation as $\left[\mathbf{f}\cdot\hat{\boldsymbol{n}}\right]^*$. In the finite volume community these numerical fluxes are computed usually with a Riemann solver. There is a large variety of Riemann solvers available for different systems of conservation laws, see for example the book by Toro [56] for an exhaustive treatment of this topic. We have to mention that the numerical flux in the normal direction is defined as a function of the left and right states at the element interface: $\boldsymbol{f}^* = \boldsymbol{f}^*(\boldsymbol{u}_L, \boldsymbol{u}_R; \hat{\boldsymbol{n}}) = \left[\mathbf{f}\cdot\hat{\boldsymbol{n}}\right]^*$, and it is computed by solving a *one-dimensional* Riemann problem for the states in the Gauss–Legendre quadrature points on the surface of the element. The Riemann solver used in our computations is the very well known Rusanov numerical flux [57]. The Rusanov numerical flux is given by

$$\boldsymbol{f}^*(\boldsymbol{u}_L, \boldsymbol{u}_R) = \frac{1}{2}(\boldsymbol{f}(\boldsymbol{u}_L) + \boldsymbol{f}(\boldsymbol{u}_L) - |\lambda_{\max}|(\boldsymbol{u}_L - \boldsymbol{u}_R)), \tag{41}$$

where $\lambda_{\max}$ is the largest local wave speed, which guarantees the stability of the scheme. Finally, the weak form of the equation (33) reads

$$\underbrace{\int_E \mathcal{J}(\xi)\frac{\partial \boldsymbol{u}}{\partial t}\phi(\xi)\;\mathrm{d}\xi}_{\text{Time Derivative Integral}} - \underbrace{\int_E \tilde{\mathbf{f}}\cdot\nabla_\xi\phi(\xi)\;\mathrm{d}\xi}_{\text{Volume Integral}} + \underbrace{\oint_{\partial E}\left[\mathbf{f}\cdot\hat{\boldsymbol{n}}\right]^*\sigma\phi(\xi)\;\mathrm{d}\sigma}_{\text{Surface Integral}} = \boldsymbol{0}. \tag{42}$$

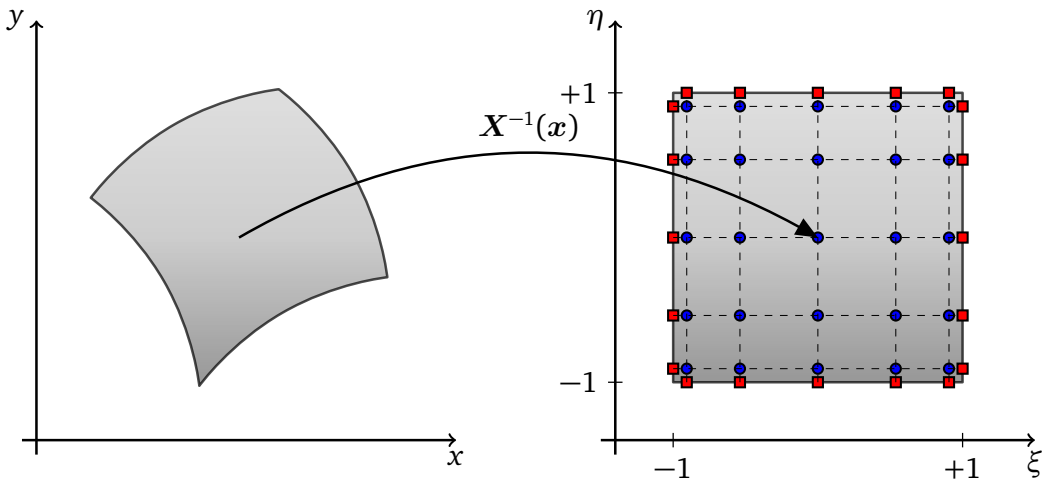

Figure 2: Mapping from the physical element Ω_i to the reference element $[-1,1]^2$ in a two-dimensional domain. The blue-filled circles depict the two-dimensional Gauss–Legendre quadrature points in the reference element, where the solution is computed, and the red-filled squares represent the nodes where the boundary fluxes are calculated, and they are located at Gauss–Legendre quadrature points along the boundary.

3.1.3. Discretization on hexahedral elements

The discontinuous Galerkin spectral element method is a discretization method specially designed for the approximate solution of conservation laws on quadrilateral/hexahedral computational domains. The partial differential equation is expressed in weak form and the solution and fluxes are approximated by using high-order Lagrangian interpolants. We stress that the polynomials are represented in a nodal form. The integrals are approximated by quadrature (in our case, Gauss quadrature), and the nodal points, where the solution is computed, are the Gauss–Legendre quadrature points. Along the boundaries, the nodes are chosen to be also Gauss–Legendre quadrature points in order to compute the boundary integrals. In the Figure 2 these nodal points are depicted for the two-dimensional case. The ansatz discussed before will be used in the following for the purpose of discretizing each integral in the equation (42).

In every hexahedral element, we approximate the vector of conserved variables and the contravariant fluxes by polynomials, which basis functions are obtained from the tensor-product of one-dimensional Lagrange polynomials of degree N. The state is given by

$$\boldsymbol{u} = \sum_{i,j,k=0}^{N} \hat{\boldsymbol{u}}_{ijk}\psi_{ijk}(\xi), \tag{43}$$

with $\hat{\boldsymbol{u}}_{ijk}$ the nodal degrees of freedom. The components of the contravariant fluxes are written as

$$\boldsymbol{f}^m(\boldsymbol{u}(\xi)) = \sum_{i,j,k=0}^{N} \hat{\boldsymbol{f}}^m_{ijk}\psi_{ijk}(\xi), \quad m = 1,2,3. \tag{44}$$

The coefficients $\hat{\boldsymbol{f}}^m_{ijk}$ are computed in terms of the covariant fluxes, as it was pointed out before

$$\hat{\boldsymbol{f}}^m_{ijk} = \hat{\boldsymbol{f}}^m_{ijk}(\boldsymbol{u}(\xi_{ijk})) = \sum_{l=1}^{3} \mathcal{J} e^m_l(\xi_{ijk})\boldsymbol{f}_l(\boldsymbol{u}(\xi_{ijk})). \tag{45}$$

The basis functions $\psi_{ijk}(\xi)$ used before have a tensor-product form

$$\psi_{ijk}(\xi) = \ell_i(\xi^1)\ell_j(\xi^2)\ell_k(\xi^3), \tag{46}$$

with ℓ_i the Lagrange polynomials

$$\ell_j(\xi) = \prod_{\substack{i=0 \\ i\neq j}}^{N} \frac{(\xi - \xi_i)}{(\xi_j - \xi_i)}, \tag{47}$$

with the property

$$\ell_j(\xi_i) = \delta_{ij}, \quad i, j = 0, \ldots, N. \tag{48}$$

Time derivative integral. In the time derivative integral in the equation (42), we insert the approximation for $\boldsymbol{u}$ given in (43). Additionally, we assume that the test function is the same as the basis functions, $\phi = \psi_{ijk}$ (this is the Galerkin ansatz), and finally we approximate the integrals by Gaussian quadrature

$$\begin{aligned}\frac{\partial}{\partial t}\int_E \mathcal{J}(\xi)\,\boldsymbol{u}(\xi)\,\phi(\xi)\,\mathrm{d}\xi &= \frac{\partial}{\partial t}\int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1}\mathcal{J}(\xi)\left(\sum_{l,m,n=0}^{N}\hat{\boldsymbol{u}}_{lmn}\psi_{lmn}(\xi)\right)\psi_{ijk}(\xi)\,\mathrm{d}\xi \\ &= \omega_i\omega_j\omega_k\mathcal{J}(\xi_{ijk})\frac{\mathrm{d}\hat{\boldsymbol{u}}_{ijk}}{\mathrm{d}t},\end{aligned}$$

where we have made use of the property (48) of the Lagrange interpolating polynomials. The ω_i are the weights in the Gaussian quadrature.

Volume integral. The volume integral in the equation (42) involves the tensor $\tilde{\mathbf{f}}$. This integral can be written as

$$\begin{aligned}\int_E \tilde{\mathbf{f}}\cdot\nabla_\xi\phi(\xi)\,\mathrm{d}\xi &= \sum_{d=1}^{3}\int_E \boldsymbol{f}^d(\boldsymbol{u}(\xi))\frac{\partial\phi(\xi)}{\partial\xi^d}\,\mathrm{d}\xi \\ &= \omega_j\omega_k\sum_{\lambda=0}^{N}D_{\lambda i}\hat{\boldsymbol{f}}^1_{\lambda jk}\,\omega_\lambda + \omega_i\omega_k\sum_{\mu=0}^{N}D_{\mu j}\hat{\boldsymbol{f}}^2_{i\mu k}\,\omega_\mu + \omega_i\omega_j\sum_{\nu=0}^{N}D_{\nu k}\hat{\boldsymbol{f}}^3_{ij\nu}\,\omega_\nu,\end{aligned}$$

where we have introduced the differentiation matrix $D_{\lambda i}$, which is defined as

$$D_{ij} = \left.\frac{\mathrm{d}\ell_j(\xi)}{\mathrm{d}\xi}\right|_{\xi=\xi_i}, \quad i, j = 0, \ldots, N. \tag{49}$$

Surface integral. The last integral we will compute is the surface integral in the equation (42). Observe that this surface integral is the sum of the surface integrals over each face of the reference element

$$\begin{aligned}\oint_{\partial E}[\mathbf{f}\cdot\hat{\boldsymbol{n}}]^*\sigma\phi(\xi)\,\mathrm{d}\sigma &= \sum_{l=1}^{6}\int_{\partial E_l}[\mathbf{f}\cdot\hat{\boldsymbol{n}}_l]^*\sigma_l\,\phi(\xi_l)\,\mathrm{d}\sigma_l \\ &= \omega_j\omega_k\Big([\boldsymbol{f}^*\sigma]_{jk}^{+\xi^1}\ell_i(+1) + [\boldsymbol{f}^*\sigma]_{jk}^{-\xi^1}\ell_i(-1)\Big) \\ &+ \omega_i\omega_k\Big([\boldsymbol{f}^*\sigma]_{ik}^{+\xi^2}\ell_j(+1) + [\boldsymbol{f}^*\sigma]_{ik}^{-\xi^2}\ell_j(-1)\Big) \\ &+ \omega_i\omega_j\Big([\boldsymbol{f}^*\sigma]_{ij}^{+\xi^3}\ell_k(+1) + [\boldsymbol{f}^*\sigma]_{ij}^{-\xi^3}\ell_k(-1)\Big).\end{aligned}$$

The vector ξ_l means that the points are on the surface ∂E_l, and σ_l is the area of the element face l. We will compute only the integrals in the direction $\pm\xi^1$. Remember that the sign in the directions is included in the normal vector. The numerical flux $\boldsymbol{f}^* = [\mathbf{f}\cdot\hat{\boldsymbol{n}}]^*$ on the faces $\pm\xi^1$ of the reference element E, as we pointed out before, are calculated by using the states $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$. The state $\boldsymbol{u}_L$ is obtained from the extrapolation of the inner state to the element boundaries and the state $\boldsymbol{u}_R$ from the extrapolation in the neighboring element (or in the case that the face to be evaluated is located on the physical boundary, from the boundary conditions imposed in the problem). This extrapolation is carried out on the Gauss–Legendre quadrature points on the face of the reference element. Therefore, the numerical fluxes on the faces $\pm\xi^1$ (i.e. the faces at $\xi^1 = \pm1$) are approximated by

$$\left.[\mathbf{f}\cdot\hat{\boldsymbol{n}}_l]^*\sigma_l\right|_{l=\pm\xi^1} = \boldsymbol{f}^*(\pm1,\xi^2,\xi^3)\,\sigma_{\pm\xi^1} = \sum_{j,k=0}^{N}[\boldsymbol{f}^*\sigma]_{jk}^{\pm\xi^1}\ell_j(\xi^2)\ell_k(\xi^3). \tag{50}$$

Semi-discrete formulation. Once discretized all integrals appearing in the equation (42), we are able to write down the semi-discrete formulation of the DGSEM on hexahedral elements. Before we write the final equation, we define some useful quantities, which can be precomputed and stored in memory. The operators to be defined are

$$\hat{D}_{ij} = -D_{ji}\frac{\omega_j}{\omega_i}, \quad \hat{\ell}_i(\pm 1) = \frac{\ell_i(\pm 1)}{\omega_i}, \quad i, j = 0, \dots, N. \tag{51}$$

The semi-discrete formulation has the following form

$$\begin{aligned}
\frac{\mathrm{d}\hat{\boldsymbol{u}}_{ijk}}{\mathrm{d}t} = &- \frac{1}{\mathcal{J}_{ijk}}\left[\sum_{\lambda=0}^{N} \hat{D}_{i\lambda}\hat{\boldsymbol{f}}^1_{\lambda jk} + \left([\boldsymbol{f}^*\sigma]^{+\xi^1}_{jk}\hat{\ell}_i(+1) + [\boldsymbol{f}^*\sigma]^{-\xi^1}_{jk}\hat{\ell}_i(-1)\right)\right] \\
&- \frac{1}{\mathcal{J}_{ijk}}\left[\sum_{\mu=0}^{N} \hat{D}_{j\mu}\hat{\boldsymbol{f}}^2_{i\mu k} + \left([\boldsymbol{f}^*\sigma]^{+\xi^2}_{ik}\hat{\ell}_j(+1) + [\boldsymbol{f}^*\sigma]^{-\xi^2}_{ik}\hat{\ell}_j(-1)\right)\right] \\
&- \frac{1}{\mathcal{J}_{ijk}}\left[\sum_{\nu=0}^{N} \hat{D}_{k\nu}\hat{\boldsymbol{f}}^3_{ij\nu} + \left([\boldsymbol{f}^*\sigma]^{+\xi^3}_{ij}\hat{\ell}_k(+1) + [\boldsymbol{f}^*\sigma]^{-\xi^3}_{ij}\hat{\ell}_k(-1)\right)\right].
\end{aligned} \tag{52}$$

The semi-discrete formulation given by equation (52) consists of one-dimensional DGSEM operators (the quantities in square brackets), each of them applied along a direction.

3.2. Discretization of the GLM divergence cleaning

The equations for B_x and ψ are decoupled from the rest of the MHD-GLM system. In fact, cleaning a vector quantity $\boldsymbol{B}$ with divergence errors results in the linear system

$$\frac{\partial}{\partial t}\begin{pmatrix} B_x \\ \psi \end{pmatrix} + \begin{pmatrix} 0 & 1 \\ c_h^2 & 0 \end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix} B_x \\ \psi \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{c_h^2}{c_p^2}\psi \end{pmatrix}. \tag{53}$$

The numerical flux is then derived as the solution of the local Riemann problem with left-hand state $(B_{x,l}, \psi_l)^T$ and right hand state $(B_{x,r}, \psi_r)^T$ as

$$B_{x,m} = \frac{1}{2}(B_{x,r} + B_{x,l}) - \frac{1}{2c_h}(\psi_r - \psi_l), \tag{54}$$

$$\psi_m = \frac{1}{2}(\psi_r + \psi_l) - \frac{c_h}{2}(B_{x,r} - B_{x,l}). \tag{55}$$

It is possible to employ for the quantities B_x and ψ the Riemann solver used in the full MHD system, but in this work we use the exact solution (54) of the linear Riemann problem (53). Dedner et al. [43] suggest to use the solution (54) as input for the Riemann solver used in the solution of the other conserved quantities.

There are basically two possible ways to deal with the source term in equation (6b). The first one consists in adding this source term to the hyperbolic update, that is to the semi-discrete scheme (52) (see [58]). The second one is based on an operator-splitting approach. Following the idea presented in [43], we first solve the homogeneous MHD-GLM system in a so-called *hyperbolic step*, and then we consider the source term in the *source step*. The scalar field ψ is then

$$\psi^{(\Delta t)} = \psi^{(0)} \exp\left(-\alpha_p \frac{c_h}{\Delta h/\Delta t}\right), \quad \text{with } \alpha_p = \Delta h \frac{c_h}{c_p^2}, \tag{56}$$

where $\psi^{(0)}$ has been computed in the hyperbolic step, and $\Delta h = \min(\Delta x, \Delta y, \Delta z)$ is the minimum mesh size. This approach is very simple to implement and is unconditionally stable [43].

3.3. Time discretization

The semi-discrete scheme (52) is solved explicitly using the so-called *method of lines*. The method of lines acquires the accuracy order of the integrator used to solve the system of ordinary differential equations, under the condition that the spatial discretization is of the same order of accuracy or higher [59]. The system of ordinary differential equations (52) is solved with the family of strong-stability preserving Runge–Kutta Methods (SSPRK) [60–62]. The main idea of these methods resides in assuming that the first-order forward Euler method is strongly stable under the total variation norm (and in general, any given norm) and a suitable time step restriction. From this, the aim is to construct a higher-order time discretization that preserves strong stability under the TVD norm, and maybe, with a different time step restriction [60]. The SSPRK methods are desirable in problems with discontinuities and strong shocks, because they guarantee that, as part of the time-integration process, no additional oscillations are introduced.

The Courant–Friedrichs–Lewy (CFL) condition is a necessary condition for stability of any explicit one-level numerical scheme [63]. It establishes that the domain of dependence of the solution is contained in the numerical domain of dependence of the numerical method. This condition is applied to constraint the time step. For the discontinuous Galerkin method, Cockburn and Shu [1] gave an approximate CFL condition for linear stability, and it has the following form

$$\Delta t = \frac{\mathrm{C_{CFL}}}{2N+1} \min_k \left(\frac{\Delta x}{|\lambda_{k,x}|}, \frac{\Delta y}{|\lambda_{k,y}|}, \frac{\Delta z}{|\lambda_{k,z}|} \right). \tag{57}$$

3.4. Shock capturing

In the discontinuous Galerkin method is common that some unphysical states are produced in the intermediate stages of the simulation (e.g., during the interpolation step, or the conservative to primitive variables conversion), especially when the flow involves shocks waves or strong rarefactions, and very high-order are used (interpolation order larger than second order). These unphysical states come from the loss of positivity of the density or the pressure, or the generation of super-luminal speeds. Several shock capturing schemes have been considered in the past, like the moment limiter [16, 17], the artificial viscosity [18, 19], and the WENO limiter [20–22]. It is very well known that in the vicinity of a discontinuity, shock-capturing schemes will reduce the solution accuracy to first-order [64]. This is because the location of the shock is proportional to the mesh size [65]. Taking this into account, it is natural to construct a robust shock-capturing by considering only two stages: the shock detection and a further order reduction of the scheme in the vicinity of the shock. This kind of schemes are called the *fallback approaches* [32, 38, 66]. For robustness and efficiency reasons, the hybrid DG/FV approach [26, 27], which is based on the fallback approach (see table 1), will be used as the main building block for the nonlinear stabilization of the solution when shocks take place. The hybrid DG/FV is constructed in such a way that, in regions of smooth flows, the DGSEM method is employed, and those parts of the flow having shocks, the DGSEM elements are interpreted as quadrilateral/hexahedral subdomains. In each of these subdomains, the nodal DG solution values are used to build a new local domain composed now of finite volume subcells, which are evolved with a robust finite volume method with third order WENO reconstruction. In the following we will discuss with more detail the hybrid DG/FV algorithm.

3.4.1. Hybrid DG/FV shock capturing

The hybrid DG/FV shock capturing method used in this work consists basically in two steps: the first step is detecting the elements in the DG computational domain that contain oscillating polynomials, which cause instabilities in the computation and make the code to blow up. The second step is replacing the troubled elements with subdomains made of $(N+1)^d$ finite volume subcells each, with d the space dimension. Then the subdomains are evolved with a robust finite volume method which uses a second or third order reconstruction operator, typically MUSCL (second order) or WENO3 (third order).

Let us represent the solution in the DG reference element E at time level t^n by $\boldsymbol{u}^n = \sum_{i,j,k=0}^{N} \hat{\boldsymbol{u}}_{ijk}^n \psi_{ijk}(\xi)$. In the context of DGSEM schemes, the reference element E has $(N+1)^3$ degrees of freedom. The DG solution is computed in the $(N+1)^3$ quadrature points. The reference element E is then decomposed into $(N+1)^3$ equidistant FV subcells E_{ijk}, in such a way that $E = \bigcup_{ijk} E_{ijk}$, see figure 3. While in the DG reference element E the solution is represented by

Table 1: Shock capturing strategy for the higher-order discontinuous Galerkin methods discussed in this work [27, 32, 38, 39, 66].

Checking of physical quantities with restrictions	Positivity of density Positivity of pressure Maximum propagation speed (only in the relativistic case)
Detection of regions with strong shocks	Shocks indicators Oscillations indicators Marking of troubled points/cells and direct neighbors
Special treatment in troubled regions	Employment of robust Riemann solvers Employment of robust first-order Godunov scheme Employment of robust second/third order numerical scheme

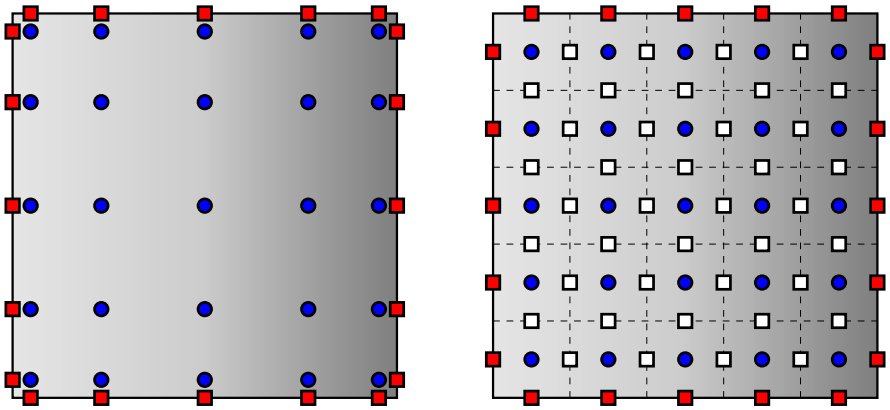

Figure 3: Hybrid DG/FV. DG element with $(N+1)^2$ inner nodal points (the blue filled circles) where the solution is computed, and which is approximated by a tensor-product of Lagrange polynomials of degree N (left). Corresponding FV subdomain made of $(N+1)^2$ equidistant subcells (right), with the barycenter of every subcell shown (the blue filled circles), and the points where the numerical fluxes are computed with the finite volume scheme (the white filled squares), and the borders of the subcells (dashed lines).

a polynomial, in the FV subdomain the solution at time level t^n is represented by $(N+1)^3$ piecewise constant subcell averages $\boldsymbol{v}_{ijk}^n$. These values are obtained from the DG polynomial itself by using L_2 projection, that is

$$\boldsymbol{v}_{ijk}^n = \widehat{\mathbf{P}}_{\mathrm{FV}}\left(\boldsymbol{u}^n\right) = \frac{1}{\left|E_{ijk}\right|}\int_{E_{ijk}} \boldsymbol{u}(\boldsymbol{x}, t^n)\,\mathrm{d}\boldsymbol{x}. \tag{58}$$

The inverse operator $\widehat{\mathbf{P}}_{\mathrm{DG}} = \widehat{\mathbf{P}}_{\mathrm{FV}}^{-1}$ has to satisfy the following constraint: The integral conservation of the cell average over the DG element

$$\int_E \boldsymbol{u}(\boldsymbol{x}, t^n)\,\mathrm{d}\boldsymbol{x} = \int_E \boldsymbol{v}(\boldsymbol{x}, t^n)\,\mathrm{d}\boldsymbol{x}. \tag{59}$$

We solve then the conservation law for the subcell cell averages

$$\frac{\partial \boldsymbol{v}}{\partial t} + \frac{1}{\mathcal{J}}\nabla_\xi \cdot \tilde{\mathbf{f}} = \boldsymbol{0}, \tag{60}$$

by using a second/third order FV method, see [26, 33]. Summarizing, given the DG state $\boldsymbol{u}_i^n$ at time level n, the oscillation indicator operator $\widehat{\mathbf{OI}}$ is applied to $\boldsymbol{u}_i^n$, $\forall i$, and only the troubled elements are flagged for shock capturing. For those not flagged elements, the state $\boldsymbol{u}^n$ is evolved with the DG solver operator $\widehat{\mathbf{DG}}$, but if the element is marked

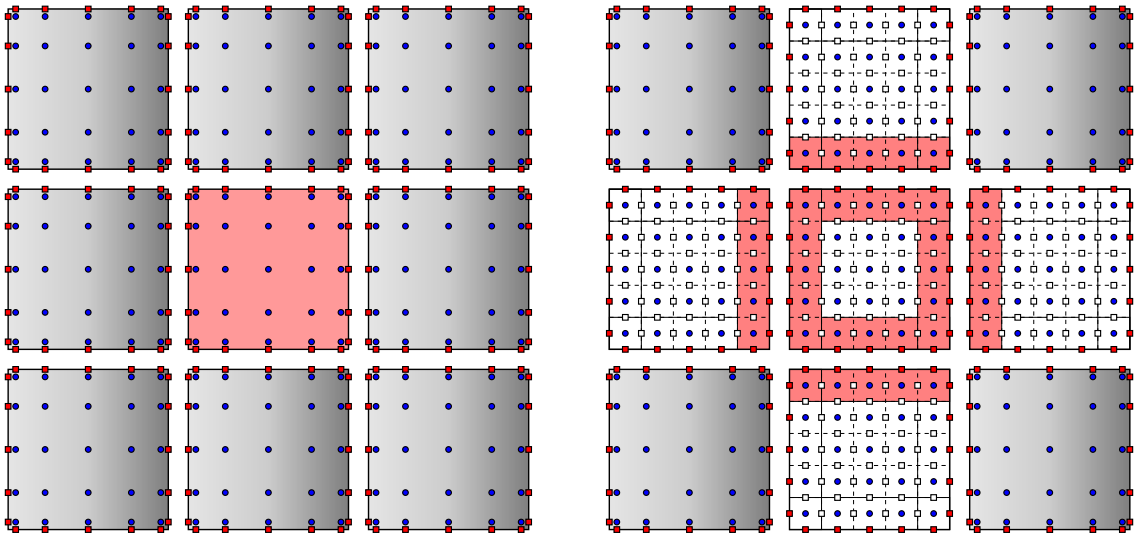

Figure 4: Hybrid DG/FV. Flagged element surrounded by DG elements (left). The DG polynomial in the troubled cell is then projected onto a FV subdomain made of $(N+1)^2$ subcells. For those subcells lying at the FV subdomain edges, neighboring elements are also projected onto a FV subdomain in order to use the cell averages at edges for computing the reconstructed values at subcell interfaces (right).

as a troubled cell, then we use the operator $\widehat{\mathbf{P}}_{\mathrm{FV}}$ to project the state $\boldsymbol{u}^n$ onto a FV subcells subdomain of piecewise cell averages, yielding $\boldsymbol{v}^n$; then we evolve the states $\boldsymbol{v}^n$ in the FV subdomain with the FV solver operator $\widehat{\mathbf{FV}}$, to get $\boldsymbol{v}^{n+1}$. The solution $\boldsymbol{v}^{n+1}$ is then projected back onto the DG element with the projection operator $\widehat{\mathbf{P}}_{\mathrm{DG}} = \widehat{\mathbf{P}}_{\mathrm{FV}}^{-1}$. Since the stencil of the reconstruction operator used in the FV subdomain needs at least two neighboring cell data (one left and one right to the subcell of interest), those subcells at edges of the FV subdomain require boundary values, which are obtained from the neighboring elements (see figure 4). A flow diagram for the DGSEM is depicted in the figure 5.

3.4.2. Finite volume schemes

For the sake of completeness, we briefly describe the foundations of the finite volume method with third-order WENO reconstruction. Let us consider the following system of conservation laws on structured Cartesian meshes

$$\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} + \frac{\partial \boldsymbol{g}(\boldsymbol{u})}{\partial y} + \frac{\partial \boldsymbol{h}(\boldsymbol{u})}{\partial z} = \boldsymbol{0}. \tag{61}$$

After the integration of the equation (61) over the computational cell $\Omega_{ijk} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$, we get the semi-discrete scheme

$$\frac{\mathrm{d}\boldsymbol{u}_{ijk}}{\mathrm{d}t} = -\frac{\hat{\boldsymbol{f}}_{i+\frac{1}{2},jk} - \hat{\boldsymbol{f}}_{i-\frac{1}{2},jk}}{\Delta x} - \frac{\hat{\boldsymbol{g}}_{i,j+\frac{1}{2},k} - \hat{\boldsymbol{g}}_{i,j-\frac{1}{2},k}}{\Delta y} - \frac{\hat{\boldsymbol{h}}_{ij,k+\frac{1}{2}} - \hat{\boldsymbol{h}}_{ij,k-\frac{1}{2}}}{\Delta z}. \tag{62}$$

The quantity $\boldsymbol{u}_{ijk}$ is the spatial average of $\boldsymbol{u}$ in the cell Ω_{ijk} at time t

$$\boldsymbol{u}_{ijk} = \frac{1}{|\Omega_{ijk}|} \int_{\Omega_{ijk}} \boldsymbol{u}(x,y,z)\,\mathrm{d}z\,\mathrm{d}y\,\mathrm{d}x, \tag{63}$$

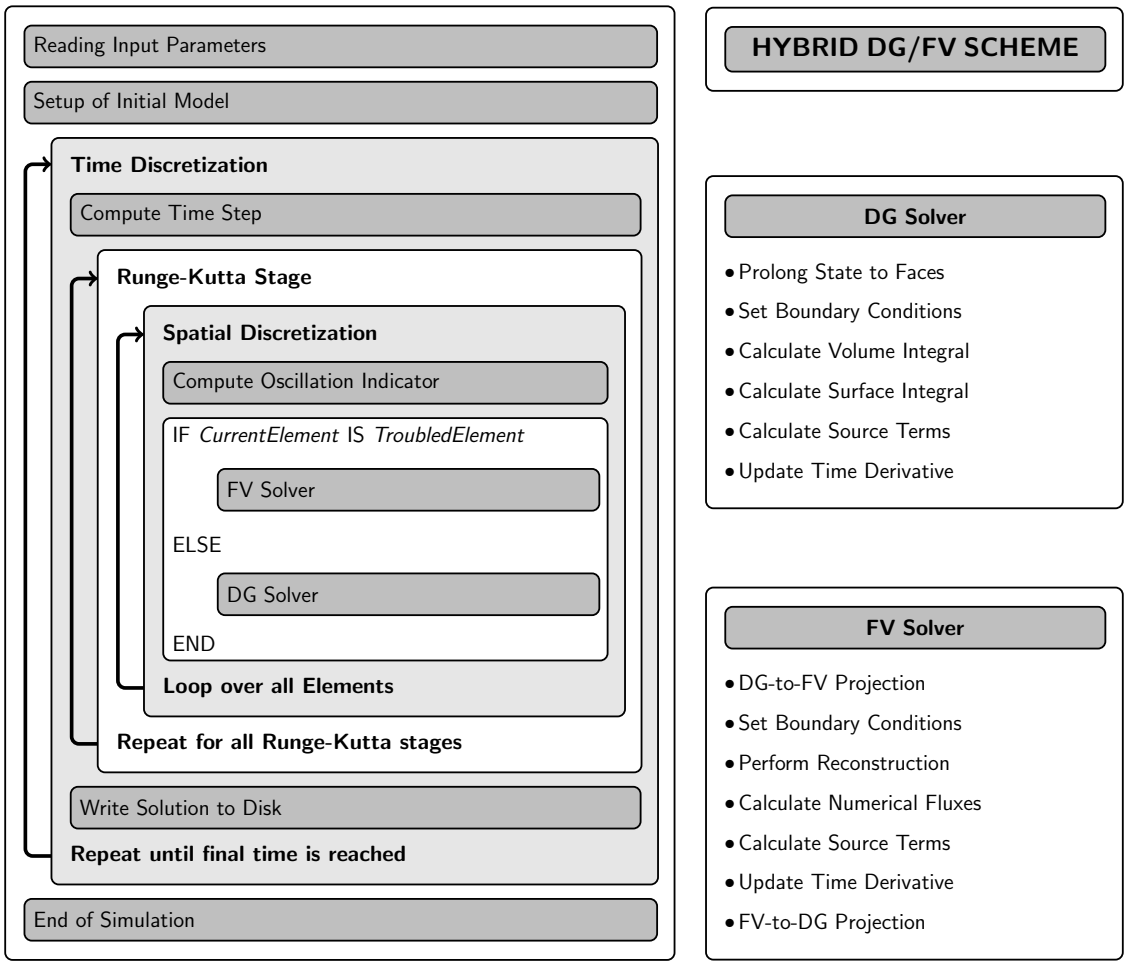

Figure 5: Flow diagram for the hybrid DG/FV method.

with $\left|\Omega_{ijk}\right| = \Delta x \Delta y \Delta z$, and $\hat{\boldsymbol{f}}_{i\pm\frac{1}{2},jk}$, $\hat{\boldsymbol{g}}_{i,j\pm\frac{1}{2},k}$, and $\hat{\boldsymbol{h}}_{ij,k\pm\frac{1}{2}}$ are spatial averages of the physical fluxes over the cell faces $x_{i\pm\frac{1}{2}}$, $y_{j\pm\frac{1}{2}}$, and $z_{k\pm\frac{1}{2}}$, respectively, at time t

$$
\begin{aligned}
\hat{\boldsymbol{f}}_{i\pm\frac{1}{2},jk} &= \frac{1}{\left|\sigma_{jk}\right|} \int_{\sigma_{jk}} \boldsymbol{f}(\boldsymbol{u}(x_{i\pm\frac{1}{2}}, y, z))\, \mathrm{d}z\, \mathrm{d}y, \\
\hat{\boldsymbol{g}}_{i,j\pm\frac{1}{2},k} &= \frac{1}{\left|\sigma_{ik}\right|} \int_{\sigma_{ik}} \boldsymbol{g}(\boldsymbol{u}(x, y_{j\pm\frac{1}{2}}, z))\, \mathrm{d}z\, \mathrm{d}x, \\
\hat{\boldsymbol{h}}_{ij,k\pm\frac{1}{2}} &= \frac{1}{\left|\sigma_{ij}\right|} \int_{\sigma_{ij}} \boldsymbol{h}(\boldsymbol{u}(x, y, z_{k\pm\frac{1}{2}}))\, \mathrm{d}y\, \mathrm{d}x,
\end{aligned}
\tag{64}
$$

with the surfaces elements defined by $\sigma_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, $\sigma_{jk} = [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$, and $\sigma_{ik} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [z_{k-\frac{1}{2}}, z_{k+\frac{1}{2}}]$. The area of the faces are then $\left|\sigma_{ij}\right| = \Delta x \Delta y$, $\left|\sigma_{jk}\right| = \Delta y \Delta z$, and $\left|\sigma_{ik}\right| = \Delta x \Delta z$. The semi-discrete scheme (62) is an exact relation as well as the averaged quantities (63) and (64). For the purpose of this work, we approximate these integral as follows

$$
\hat{\boldsymbol{f}}_{i\pm\frac{1}{2},jk} = \frac{\boldsymbol{f}(\boldsymbol{u}(x_{i\pm\frac{1}{2}}, y_j, z_k))}{\Delta y \Delta z}, \quad \hat{\boldsymbol{g}}_{i,j\pm\frac{1}{2},k} = \frac{\boldsymbol{g}(\boldsymbol{u}(x_i, y_{j\pm\frac{1}{2}}, z_k))}{\Delta x \Delta z}, \quad \hat{\boldsymbol{h}}_{ij,k\pm\frac{1}{2}} = \frac{\boldsymbol{h}(\boldsymbol{u}(x_i, y_j, z_{k\pm\frac{1}{2}}))}{\Delta x \Delta y}. \tag{65}
$$

The point-wise values $\boldsymbol{u}(x_{i\pm\frac{1}{2}}, y_j, z_k)$, $\boldsymbol{u}(x_i, y_{j\pm\frac{1}{2}}, z_k)$, $\boldsymbol{u}(x_i, y_j, z_{k\pm\frac{1}{2}})$ are reconstructed from the cell averages $\boldsymbol{u}_{ijk}$ with a third-order WENO scheme. Observe that two sets of reconstructed values at a given face are present: those obtained through the use of the cell Ω_{ijk} as the main cell in the reconstruction procedure and those associated with the neighboring cell. These are known as the left and right values at the face interface: $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$. Finally the fluxes are evaluated by replacing $\boldsymbol{f}$, $\boldsymbol{g}$, and $\boldsymbol{h}$ by a monotone flux. We employ the same Riemann solver used in the DGSEM scheme, that is the Rusanov numerical flux [57].

3.4.3. WENO reconstruction

The basic idea of the WENO schemes is based on an adaptive reconstruction procedure to obtain a higher-order approximation on smooth regions while the scheme remains non-oscillatory near discontinuities [15]. In each stencil, a polynomial is reconstructed from the cell averages $\bar{u}_i$ of the solution and later a weighted combination of all these polynomials is constructed. The nonlinear weights are computed taking into account the smoothness of every polynomial in its respective stencil.

Let us approximate the value of the function $u(x)$ at the points $x_{i-\frac{1}{2}}$, and $x_{i+\frac{1}{2}}$, by using polynomials of degree $N = 1$. For the stencil S_k we get the corresponding values $u^{(k)}_{i-\frac{1}{2}}$, and $u^{(k)}_{i+\frac{1}{2}}$. Let us write down the stencils

$$S_1 = \{I_{i-1}, I_i\}, \quad S_2 = \{I_i, I_{i+1}\}, \tag{66}$$

where $I_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$. The corresponding interpolated values for the point $x_{i-\frac{1}{2}}$ are

$$u^{(1)}_{i-\frac{1}{2}} = \frac{1}{2}(\bar{u}_{i-1} + \bar{u}_i), \quad u^{(2)}_{i-\frac{1}{2}} = \frac{1}{2}(3\bar{u}_i - \bar{u}_{i+1}), \tag{67}$$

and the corresponding interpolated values for the point $x_{i+\frac{1}{2}}$ are

$$u^{(1)}_{i+\frac{1}{2}} = \frac{1}{2}(-\bar{u}_{i-1} + 3\bar{u}_i), \quad u^{(2)}_{i+\frac{1}{2}} = \frac{1}{2}(\bar{u}_i + \bar{u}_{i+1}). \tag{68}$$

If we choose the large stencil

$$S = \{I_{i-1}, I_i, I_{i+1}\}, \tag{69}$$

which is the union of all 2 stencils S_k, then we are able to find an interpolation polynomial $p(x)$ of degree at most 2, satisfying $p(x_j) = u_j$ for $i - N \leq j \leq i + N$ and giving the approximations $u_{i-\frac{1}{2}} \equiv p(x_{i-\frac{1}{2}})$, $u_{i+\frac{1}{2}} \equiv p(x_{i+\frac{1}{2}})$,

$$u_{i-\frac{1}{2}} = \frac{1}{6}(2\bar{u}_{i-1} + 5\bar{u}_i - \bar{u}_{i+1}), \quad u_{i+\frac{1}{2}} = \frac{1}{6}(-\bar{u}_{i-1} + 5\bar{u}_i + 2\bar{u}_{i+1}), \tag{70}$$

provided that the function is smooth in the large stencil S.

The WENO idea consists in choosing the final approximation as a convex combination of the approximations $u^{(k)}_{i+\frac{1}{2}}$

$$u_{i+\frac{1}{2}} = \sum_{k=1}^{N+1} \omega_k u^{(k)}_{i+\frac{1}{2}}, \tag{71}$$

where $\omega_k \geq 0$, $\sum_{k=1}^{N+1} \omega_k = 1$. The approximations calculated by using the large stencils can be written as a *linear convex combination* of the approximations $u^{(k)}_{i+\frac{1}{2}}$ based on the small stencils S_i

$$u_{i+\frac{1}{2}} = \sum_{k=1}^{N+1} \gamma_k u^{(k)}_{i+\frac{1}{2}}, \tag{72}$$

where the constants γ_j satisfy $\sum_{j=1}^{N+1} \gamma_j = 1$, are usually referred to as the *linear weights* in the WENO literature. For the third-order WENO scheme, the linear weights for the points $x_{i-\frac{1}{2}}$, and $x_{i+\frac{1}{2}}$, are

$$x_{i-\frac{1}{2}}: \quad \gamma_1 = \frac{2}{3}, \quad \gamma_2 = \frac{1}{2}, \qquad x_{i+\frac{1}{2}}: \quad \gamma_1 = \frac{1}{2}, \quad \gamma_2 = \frac{2}{3}. \tag{73}$$

The choice of the nonlinear weights ω_k relies on the *smoothness indicator* β_k, which measures the relative smoothness of the function $u(x)$ in the stencil S_k. The larger this smoothness indicator β_k, the less smooth the function $u(x)$ is in the stencil S_k. The smoothness indicators for the third-order WENO scheme are

$$\beta_1 = (\bar{u}_{i-1} - \bar{u}_i)^2 , \quad \beta_2 = (\bar{u}_i - \bar{u}_{i+1})^2 . \tag{74}$$

Notice that these smoothness indicators are quadratic functions of the values of $u(x)$ in the relevant stencils. Equipped with these smoothness indicators, we can now define the nonlinear weights as

$$\omega_k = \frac{\tilde{\omega}_k}{\sum_{i=1}^{N} \tilde{\omega}_i}, \quad \text{with} \quad \tilde{\omega}_k = \frac{\gamma_k}{(\varepsilon + \beta_k)^2}. \tag{75}$$

Here ε is a small positive number used to avoid the denominator becoming zero and is typically chosen to be $\varepsilon = 10^{-6}$, but in our calculations we have used $\varepsilon = 10^{-24}$.

3.4.4. Oscillations indicators

In our *a priori* shock capturing strategy for discontinuous Galerkin methods, the first step consists in determining which elements contain oscillating polynomials, that is, those elements where shock waves or discontinuities are producing these kind of behavior in the approximating polynomials. The oscillation indicator is applied to every element at each stage of the explicit Runge–Kutta time discretization. Once the troubled element is flagged, the next step is to apply the hybrid DG/FV approach, where the DG element is used to build a new block made of finite volume subcells, and which is evolved using a robust second/third order finite volume scheme. In this work we mainly use the Persson indicator.

Persson indicator. The Persson indicator was developed by Persson and Peraire [18], and it measures cell-wise the maximum energy decay in the highest and second highest degrees of freedom of the DG polynomial for the state variable u:

$$\eta(u) = \log_{10} \left\{ \max \left[\left(\frac{\sum_{j=N_{\mathrm{DOF}}(N-1)+1}^{N_{\mathrm{DOF}}(N)} \left(\hat{u}_j\right)^2}{\sum_{j=1}^{N_{\mathrm{DOF}}(N)} \left(\hat{u}_j\right)^2} \right), \left(\frac{\sum_{j=N_{\mathrm{DOF}}(N-2)+1}^{N_{\mathrm{DOF}}(N-1)} \left(\hat{u}_j\right)^2}{\sum_{j=1}^{N_{\mathrm{DOF}}(N-1)} \left(\hat{u}_j\right)^2} \right) \right] \right\}, \tag{76}$$

where $N_{\mathrm{DOF}}(N)$ denotes the number of degrees of freedom for a DG polynomial of degree N and $\hat{u}_j$ the j^{th} degree of freedom of the polynomial representation of u. Observe that this indicator takes into account the influence of odd/even order effects in the DG polynomial. The typical variable used by the Persson indicator in order to determine if a cell suffers of spurious oscillations is the pressure, since shock waves imply a jump in the pressure and not in the density. With this is distinguished shock waves and contact discontinuities.

4. Numerical computations

In this section we will present several numerical calculations of the MHD and SRHD equations with the hybrid DG/FV scheme. Convergence tests, and one- and two-dimensional flow problems involving shock waves are discussed. We emphasize on the employment of very high order polynomials for the discretization of the vector of conservative quantities as well as the fluxes. In the two-dimensional computations the solution has been plotted in a hybrid manner, that is we plot the DG polynomial solution along the FV subdomains with their corresponding subcell distribution.

Table 2: Convergence rates for the MHD equations with initial condition given by the *smooth circularly polarized Alfvén wave*. Results for L_2 norm error of the x-component of magnetic field are given. The computations were performed with the pure DGSEM, and the hybrid DG/FV with all elements flagged as troubled. The schemes use polynomial degrees $N = 3, 5, 7$, that is for orders of accuracy $O(4)$, $O(6)$ and $O(8)$.

		DGSEM		Hybrid DG/FV	
Method	Elements	L_2 error	L_2 order	L_2 error	L_2 order
DG-$\mathcal{P}_3$	32×16	4.73×10^{-4}	—	3.68×10^{-4}	—
	64×32	2.95×10^{-5}	4.01	5.19×10^{-5}	2.83
	128×64	1.77×10^{-6}	4.06	6.37×10^{-6}	3.03
	256×128	9.55×10^{-8}	4.21	7.76×10^{-7}	3.04
DG-$\mathcal{P}_5$	16×8	2.18×10^{-7}	—	6.72×10^{-4}	—
	32×16	3.35×10^{-9}	6.02	8.38×10^{-5}	3.00
	64×32	4.70×10^{-11}	6.16	9.83×10^{-6}	3.09
	128×64	6.44×10^{-13}	6.19	1.13×10^{-6}	3.12
DG-$\mathcal{P}_7$	8×4	9.29×10^{-8}	—	7.19×10^{-4}	—
	16×8	4.82×10^{-10}	7.59	9.52×10^{-5}	2.92
	32×16	2.33×10^{-12}	7.69	1.21×10^{-5}	2.98
	64×32	1.10×10^{-14}	7.72	1.46×10^{-6}	3.05

4.1. Magnetohydrodynamics

4.2. Propagation of a smooth circularly polarized Alfvén wave

In order to check the order of accuracy of the algorithm with high order polynomials, we need to know a smooth exact solution of the MHD equations. The propagation of a smooth circularly polarized Alfvén wave is an example of such exact solution. Therefore it is suitable for performing a convergence test. Following Mignone et al. [45], the simulation parameters are set as follows: the two-dimensional computational domain has dimensions $L_x = 1.0$ and $L_y = 0.5$, and $n_x = 2n_y$, where n_x is the number of elements in x-direction, and n_y is the number of elements in y-direction. The Alfvén wave travels along the diagonal of the domain, and forms an angle $\alpha = \arctan(0.5) \approx 26.6°$ with respect to the x-axis. The primitive variables are set in the following way: the density is set to $\rho = 1$, and the pressure $p = 0.1$. Let us define the quantities $v_0 = 0.1$, $B_0 = 0.1$, and $x_\parallel = (x \cos\alpha + y \sin\alpha)$. The components of the velocity and magnetic field perpendicular to the wave vector are given by $v_\perp = v_0 \sin(2\pi x_\parallel)$ and $B_\perp = B_0 \sin(2\pi x_\parallel)$, and the velocity and magnetic field in z-direction are $v_z = v_0 \cos(2\pi x_\parallel)$ and $B_z = B_0 \cos(2\pi x_\parallel)$. The perpendicular components $v_\perp$ and $B_\perp$ can be written as linear combination of B_x and B_y through $B_\perp = B_y \cos\alpha - B_x \sin\alpha$, and $B_\parallel = B_x \cos\alpha + B_y \sin\alpha$. An ideal gas EOS with adiabatic exponent $\gamma = 5/3$ is used. Periodic boundary conditions are prescribed everywhere. The errors and the rate of convergence are reported in the table 2, for computations at simulation time $t = 1$, and for polynomials of degree $N = 3, 5, 7$. From this table we can observe that the experimental order of convergence (EOC) is achieved by the pure DGSEM, but the hybrid DG/FV scheme with all elements flagged as troubled is third order accurate, the order of the WENO3 reconstruction.

4.3. One-dimensional Riemann problems

Here, we run a series of one-dimensional Riemann problems, which have been extensively used to check the robustness of numerical schemes for conservation laws. We analyze the test problems reported in [28, 31, 67]. These problems are characterized by the appearance of different shock patterns after the breaking of the membrane. In order to clarify, a Riemann problem is an initial-value problem for a conservation law defined as

$$\begin{aligned} &\frac{\partial \boldsymbol{u}}{\partial t} + \frac{\partial \boldsymbol{f}(\boldsymbol{u})}{\partial x} = \boldsymbol{0}, \\ &\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_L(x) & \text{for } x < x_\mathrm{m}, \\ \boldsymbol{u}_R(x) & \text{for } x > x_\mathrm{m}, \end{cases} \end{aligned} \tag{77}$$

Table 3: Riemann problems for the magnetohydrodynamics equations. The initial states have been taken from Ryu and Jones [28]. The adiabatic index is set to $\gamma = 5/3$ and the computational domain is the interval $[0, 1]$. For the problems 1 and 2, the magnetic field components have to be divided by $\sqrt{4\pi}$.

Test	State	ρ	v_x	v_y	v_z	p	B_x	B_y	B_z
RP-1	$\boldsymbol{w}_L$	1.00	10.00	0.00	0.00	20.00	5.00	5.00	0.00
	$\boldsymbol{w}_R$	1.00	−10.00	0.00	0.00	1.00	5.00	5.00	0.00
RP-2	$\boldsymbol{w}_L$	1.08	1.20	0.01	0.50	0.95	2.00	3.60	2.00
	$\boldsymbol{w}_R$	1.00	0.00	0.00	0.00	1.00	2.00	4.00	2.00
RP-3	$\boldsymbol{w}_L$	1.00	0.00	0.00	0.00	1.00	0.70	0.00	0.00
	$\boldsymbol{w}_R$	0.30	0.00	0.00	1.00	0.20	0.70	1.00	0.00
RP-4	$\boldsymbol{w}_L$	1.00	0.00	0.00	0.00	1.00	0.75	1.00	0.00
	$\boldsymbol{w}_R$	0.125	0.00	0.00	0.00	0.10	0.75	−1.00	0.00

where x_{m} is the position of the initial discontinuity, and $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$ are two constant states written in conservative variables. Typically, the initial condition in a hyperbolic boundary-value problem is provided in terms of the primitive variables [56]. Therefore, in this work, the left state will be represented by $\boldsymbol{w}_L = (\rho, v_x, v_y, v_z, p, B_x, B_y, B_z)_L$ and the right state by $\boldsymbol{w}_R = (\rho, v_x, v_y, v_z, p, B_x, B_y, B_z)_R$. These vectors are written in primitive variables, in contrast with the vectors $\boldsymbol{u}_L$ and $\boldsymbol{u}_R$. For the four Riemann problems discussed, the one-dimensional domain is the closed interval $[0, 1]$. The membrane is localized in the point $x_{\mathrm{m}} = 0.5$. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is assumed for all Riemann problems. Dirichlet boundary conditions are imposed at domain boundaries. The computational domain is decomposed into 500 elements, for all polynomials degrees. The hybrid DG/FV scheme used for solving these problems employs polynomials of degree $N = 2, 4, 6$. The four initial conditions are given in table 3, and the exact solutions have been obtained with the exact Riemann solver developed by Torrilhon [68], and they are plotted as red lines (see figure 6). The description of the flow patterns is presented below together with some comments regarding the capabilities of the hybrid DG/FV scheme with WENO3 reconstruction in the FV solver.

Riemann problem 1 (RP-1). The flow pattern features, after the breaking of the membrane, two fast shocks, one weak slow shock, one slow rarefaction and one contact discontinuity. In the figure 6 (top) are depicted plots of the density, thermal pressure, and the y-component of the magnetic field at simulation time $t = 0.08$. The discontinuities are very well captured. Very small oscillations are observed for all polynomials degrees.

Riemann problem 2 (RP-2). This problem comprises multiple weak discontinuities, namely, two fast shocks, two slow shocks, two rotational discontinuities and one contact discontinuity. The shocks travel from each side of the contact discontinuity. In the figure 6 (second row, from top) are depicted plots of the density, thermal pressure, and the y-component of the magnetic field at simulation time $t = 0.2$. Thanks to the subcell resolution of the hybrid DG/FV scheme with WENO3 reconstruction, we observe a very good agreement with the exact solution.

Riemann problem 3 (RP-3). In this problem are present a switch-on slow rarefaction wave. The following flow structures are developed: a hydrodynamic rarefaction, a switch-on slow rarefaction, a contact discontinuity, a slow shock, a rotational discontinuity, and a fast rarefaction. In the figure 6 (second row, from bottom) are depicted plots of the density, thermal pressure, and the y-component of the magnetic field at time $t = 0.16$. All structures mentioned before are present in the figure, from left to right. The discontinuities have been very well captured and they are oscillations-free.

Riemann problem 4 (RP-4). This problem is characterized by the formation of the so-called compound structures. These structures involve a shock and a rarefaction wave traveling together. Besides, the test involves a left-going slow compound wave. In the figure 6 (bottom) are depicted plots of the density, thermal pressure, and the y-component of the magnetic field at simulation time $t = 0.1$. The flow pattern in the figures, from left to right, are a left-going fast

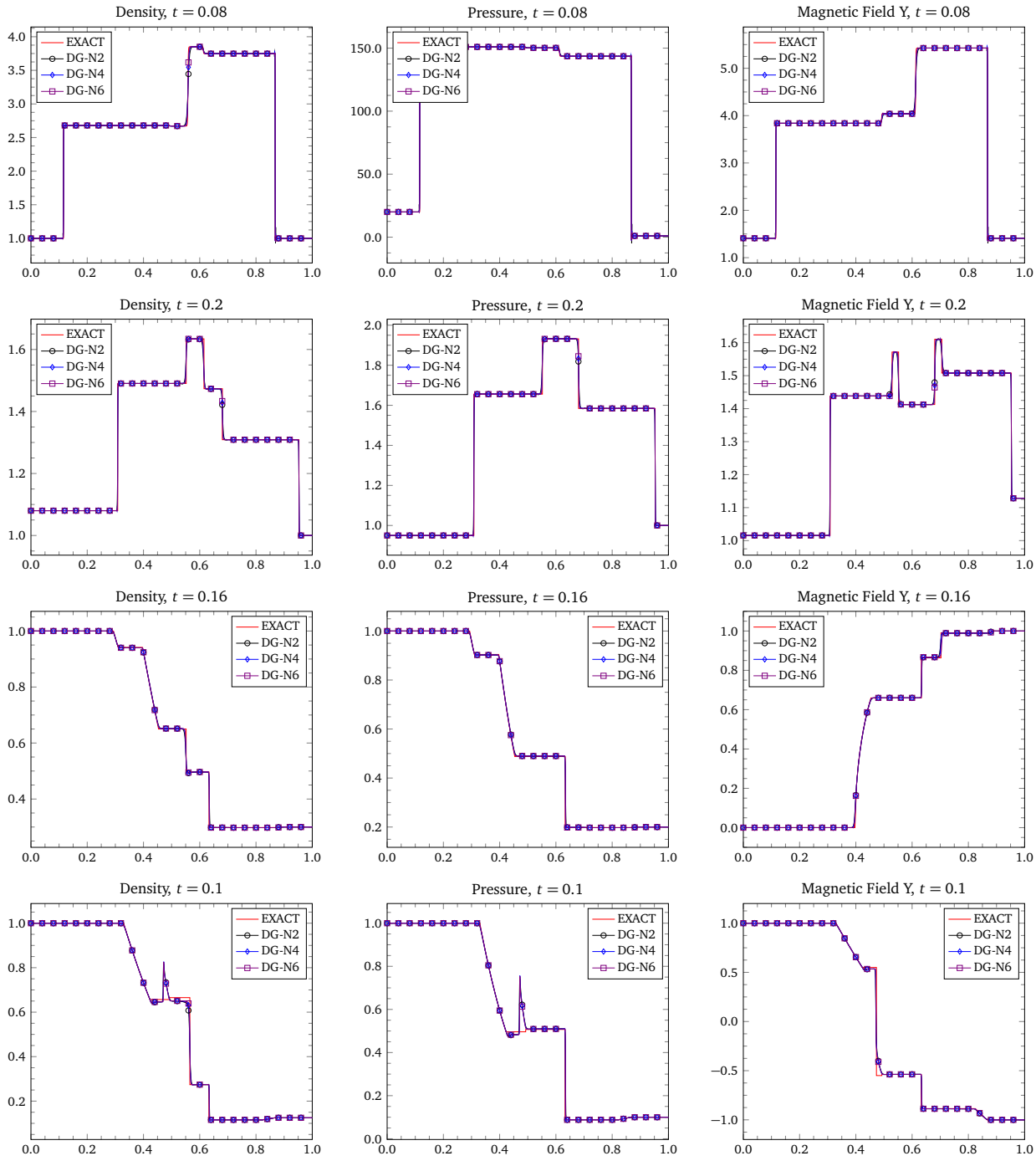

Figure 6: Plots for the Riemann problems RP-1, RP-2, RP-3, and RP-4. The initial states have been provided in the table 3. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is used. The hybrid DG/FV scheme used for solving these problems employs polynomials of degree $N = 2, 4, 6$.

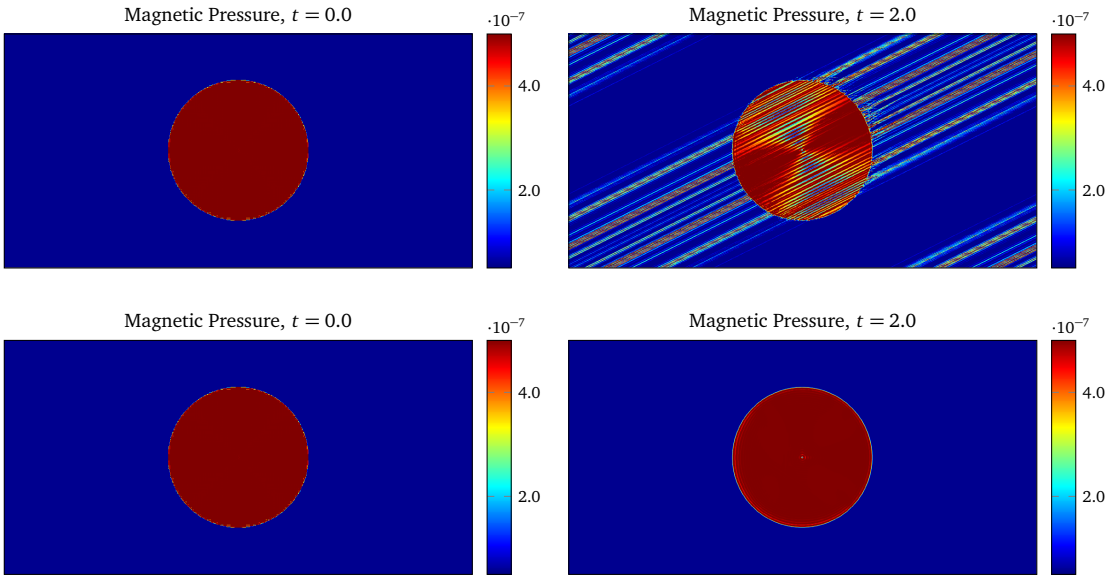

Figure 7: Magnetic field loop advection. Contour plots of the magnetic pressure without divergence cleaning (top) and with divergence cleaning (bottom) at time $t = 0$ (left) and at time $t = 2$ (right). The solution was computed with a hybrid DG/FV scheme with polynomial degree $N = 5$. The computational domain is $\Omega = [-1.0, +1.0] \times [-0.5, +0.5]$, and it has been discretized by using a mesh of 180×90 elements.

rarefaction, a left-going slow compound wave, a contact discontinuity, a right-going slow shock wave, and a right-going fast rarefaction wave. The hybrid DG/FV scheme with WENO3 reconstruction produces a solution that is in good agreement with the exact solution, except in the compound wave and contact discontinuity.

4.4. Multidimensional test problems

4.4.1. Magnetic field loop advection

The high-order DGSEM is not able by itself to maintain the solenoidal constraint $\nabla \cdot \boldsymbol{B} = 0$. It requires an additional procedure to tackle this issue. Here we employ the divergence cleaning of Dedner et al. [43], with $c_r = 0.18$, and c_h determined by the maximum propagation speed in the system. The magnetic field loop advection is designed to assess the capability of the numerical scheme to keep the solenoidal constraint. The problem consists in a cylindrical current distribution, which is advected along some direction of the computational domain. The main feature of the loop is that it remains in magnetostatic balance, and therefore, after some periods, its profile should be kept. Without a divergence cleaning algorithm, the magnetic loop will smear over the time [69].

The problem setup is as follows: the computational domain is $\Omega = [-1.0, +1.0] \times [-0.5, +0.5]$. The density and the pressure are initially set to $\rho = 1$ and $p = 1$ in the whole domain. The initial velocity profile is given by

$$v_x = v_0 n_x, \quad v_y = v_0 n_y, \quad v_z = 0, \tag{78}$$

where $v_0 = \sqrt{5}$ is the magnitude of the velocity, and $n_x = 2/\sqrt{5}$ and $n_y = 1/\sqrt{5}$ are the components of the unit vector in the loop travel direction. The magnetic field is uniform everywhere, except for the loop structure of radius $R = 0.3$. For $r \leq R$ we have

$$B_x = -B_0 y/r, \quad B_y = +B_0 x/r, \quad B_z = 0, \tag{79}$$

where $r = \sqrt{x^2 + y^2}$, and $B_0 = 10^{-3}$ is the magnitude of the magnetic field, and it is chosen in such a way that the magnetic pressure is smaller than the gas pressure. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is used. Regarding the discretization, the computational domain is made of 180×90 elements. A polynomial degree $N = 5$ is employed in the hybrid DG/FV scheme. We set periodic boundary conditions at the boundaries of the computational domain. The simulation time is $t = 2$, and in the figure 7 is depicted the magnetic pressure at times $t \in \{0.0, 2.0\}$. Observe that the loop profile is only kept when the divergence cleaning is used in the algorithm.

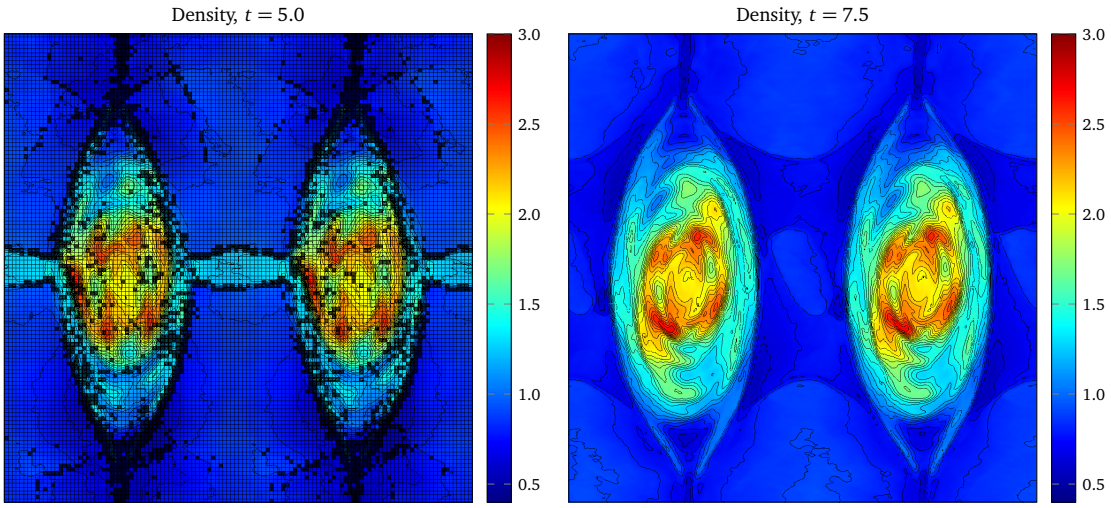

Figure 8: Current sheet problem. Contour plots of the density at times $t = 5.0$ (left) and $t = 7.5$ (right). The calculations were performed with a hybrid DG/FV scheme, with polynomial degree $N = 4$. The computational domain is $\Omega = [-0.5, +0.5] \times [-0.5, +0.5]$, and it has been discretized by using a mesh of 120×120 elements.

4.4.2. Current sheet

In this test problem, a region is uniformly filled with a gas at rest. The initial setup of the magnetic field is such that this switches signs at the slices $x = +0.25$ and $x = -0.25$. This configuration is perturbed with a sinusoidal velocity function in y, which generates nonlinear, linearly polarized Alfvén waves. These Alfvén waves turn into magnetosonic waves due to the magnetic pressure does not remain constant. The two current sheets at $x = \pm 0.25$ originate magnetic reconnection. The magnetic reconnection drives highly over-pressurized regions, which launch magnetosonic waves transverse to the field, causing magnetic energy to be transformed into thermal energy [70]. Large magnetic field gradients are produced in the proximity of the points where the magnetic reconnection takes place. This problem is useful to check if the numerical scheme is capable of handling magnetic reconnection and shock waves.

The computational domain is $\Omega = [-0.5, +0.5] \times [-0.5, +0.5]$. The density and the pressure are uniform in the whole computational domain, with $\rho = 1.0$, and $p = 0.5\beta$. The parameter $\beta = 0.1$ is the ratio of gas pressure to magnetic energy density. The components of the velocity are $v_y = v_z = 0$, and $v_x = A \sin(2\pi y)$, where $A = 0.1$ is a parameter that is typically used to test the robustness of the algorithm. The magnetic field components are $B_x = B_z = 0$, and $B_y = 1$ for $|x| > 0.25$ and $B_y = -1$ otherwise. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is assumed. Periodic boundary conditions are set at all four boundaries of the domain. The problem runs up to final time $t = 10$. In the figure 8 are depicted the density at times $t = 5.0$ and $t = 7.5$. At time $t = 5$ the solution is plotted along with the mesh. The FV subcells are located in those regions with strong shocks. Magnetic islands emerge and merge each other. The numerical results are similar to those computed with a high-order finite volume scheme [33]. Due to the complex structures present in this problem along with the large gradients in the flow variables, it is usual that a high-order method without an appropriate shock-capturing strategy crashes. We stress the robustness of the hybrid DG/FV scheme for this kind of problems, even for long time runs.

4.4.3. Orszag–Tang vortex

The Orszag–Tang vortex was first analyzed by Orszag and Tang [71] for the incompressible MHD equations. Nowadays this test is widely used to assess the capacity of the numerical scheme for handling the formation and the interactions of MHD shocks in the context of the compressible MHD equations [69, 72–74]. Initially, the flow profile consists in the superposition of a velocity vortex with a magnetic vortex. This configuration is rather unstable, which originates a broad range of MHD waves, interacting with each other, making a transition towards turbulence.

The density and the pressure are uniform in the whole domain, with $\rho = \gamma^2$ and $p = \gamma$. With this choice of the

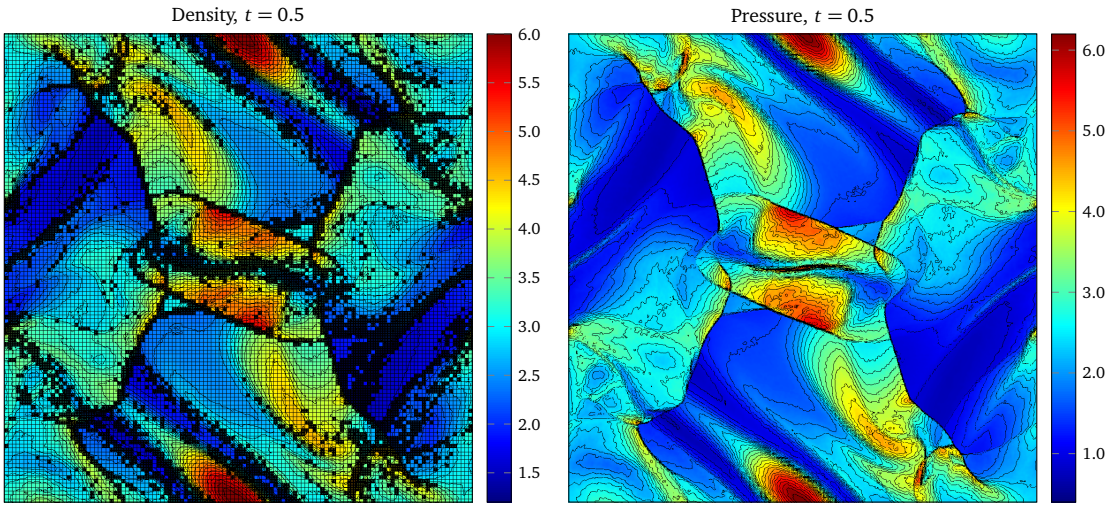

Figure 9: Orszag–Tang vortex problem. Contour plots of the density (left) and the pressure (right) at time $t = 0.5$. The calculations were performed with a hybrid DG/FV scheme, with polynomial degree $N = 5$. The computational domain is $\Omega = [0, 1] \times [0, 1]$, and it has been discretized by using a mesh of 120×120 elements.

density and pressure, the sound speed is $c_s = 1$. The initial velocity profile is given by

$$v_x = -\sin(2\pi y), \quad v_y = +\sin(2\pi x), \quad v_z = 0, \tag{80}$$

and the magnetic field is

$$B_x = -\sin(2\pi y), \quad B_y = +\sin(4\pi x), \quad B_z = 0. \tag{81}$$

An ideal gas EOS with adiabatic index $\gamma = 5/3$ is assumed. The computational domain is $\Omega = [0, 1] \times [0, 1]$, and it has been decomposed into 120×120 elements. Periodic boundary conditions are set at the boundaries of the domain. The simulation time is $t = 1.0$, and in the figure 9 are depicted the density and gas pressure at time $t = 0.5$. For the plot of the density, the mesh is also presented. The FV subdomains are evolved with a third-order finite volume WENO scheme. The hybrid DG/FV method is quite stable and robust even for this kind of supersonic flow configurations. Observe that the contours are not smooth. This is because we have chosen a small threshold for deciding where to use a FV solver and where a DG solver. We wanted to use as much DG cells as possible for this problem. The polynomial degree we employed is $N = 5$. The divergence cleaning of Dedner et al. [43] was used, with $c_r = 0.18$, and c_h determined by the maximum propagation speed in the system. Additionally, in the figure 10 is plotted the pressure along the slices $y = 0.4277$ and $y = 0.3125$ at time $t = 0.5$. Computations were made with a very high-order finite difference WENO5 scheme (mesh of 600^2 cells) and the hybrid DG/FV method with $N = 5$ (mesh of 120^2 elements). The profiles match very well, except at boundaries, where a slight shift is observed. It is also noticeable the lack of oscillations in these slices for the hybrid DG/FV scheme, considering the finite difference as reference solution.

4.4.4. Cylindrical blast wave

This problem features a cylindrical region located in the center of a domain. The cylindrical region is filled with a magnetized overpressured gas. After the system is released, strong shock waves move outwards [72, 75, 76]. The computational domain is $\Omega = [0, 1] \times [0, 1]$. The density and pressure are uniform in the whole domain, with $\rho = 1$ and $p = 0.1$. In the cylindrical region $(x - x_c)^2 + (y - y_c)^2 < R$, with $(x_c, y_c) = (0.5, 0.5)$, and $R = 0.1$, the pressure is $p = 10.0$. The initial velocity is zero, $v_x = v_y = v_z = 0$, and the magnetic field is $B_x = B_y = B_z = 1/\sqrt{2}$. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is employed. Periodic boundaries are also assumed. The simulation time is $t = 0.5$, and in the figure 11 are depicted the density and pressure at time $t = 0.1$. The computational domain is decomposed into 120×120 elements. The spatial discretization is the hybrid DG/FV scheme with WENO3 reconstruction operator. The polynomial used for the simulation has degree $N = 5$. The divergence cleaning of Dedner

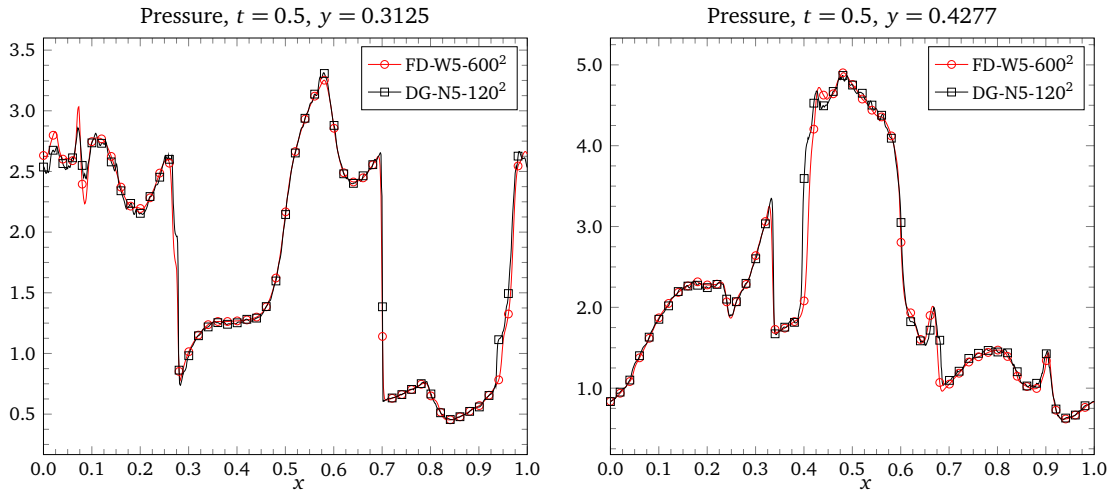

Figure 10: Orszag–Tang vortex problem. Plots of the pressure along the slices $y = 0.3125$ (left) and $y = 0.4277$ (right) at time $t = 0.5$. The computations were done with a very high-order finite difference WENO5 scheme and the hybrid DG/FV method with $N = 5$.

et al. [43] was employed, with $c_r = 0.18$. The FV solver is able to capture the shocks arising after the release of the cylindrical region. The solution of this very difficult problem with the hybrid scheme confirms the robustness of the shock-capturing approach. Very small oscillations can be observed in the contour lines of the pressure plot.

4.4.5. Rotor problem

This problem consists in a high-density, rapidly spinning fluid in a low-density fluid. Initially, the rotor and the ambient medium are subject to an uniform magnetic field. The rotating fluid launches torsional Alfvén waves into the fluid at rest [77]. The problem setup is as follows: in the ambient medium, the initial density and pressure are uniform, with $\rho = 1.0$ and $p = 1.0$, and the velocity is zero, $v_x = v_y = v_z = 0$. In the interior of the cylindrical rotor ($0.0 \leq (x - x_c)^2 + (y - y_c)^2 \leq 0.1$, with $(x_c, y_c) = (0.5, 0.5)$), the fluid density is $\rho = 10.0$, and the pressure is the same as in the ambient fluid. The fluid inside rotor has an angular velocity ω such that $v = \omega r = 1$ at $r = 0.1$. We applied a linear taper to the velocity and the density. This taper is applied only in the region $0 \leq r \leq 0.105$ in such a way that the density and the velocity have the same values that those of the ambient fluid. Finally, the components of the magnetic field are $B_x = 5.0$, $B_y = 0$, and $B_z = 0$ in the whole computational domain.

The computational domain is $\Omega = [0, 1] \times [0, 1]$. An ideal gas EOS with $\gamma = 7/5$ is employed. Dirichlet boundary conditions are assumed. The computational domain is discretized using 120×120 elements. The polynomial used in the calculations has degree $N = 5$. For the treatment of the solenoidal constraint, the divergence cleaning of Dedner et al. [43] has been employed. The simulation time is $t = 0.5$, and in the figure 12 are depicted the density (along the mesh), pressure, Mach number, and the magnitude of the magnetic field at time $t = 0.25$. In the figure 13 is plotted the pressure along the slices $x = 0.5$ and $y = 0.5$ at time $t = 0.25$. The computations were done with a very high-order finite difference WENO5 scheme and the hybrid DG/FV method with $N = 5$. Observe the good agreement with the solution obtained with the high-order finite difference scheme. A lack of oscillations is also noticeable. We point out the difference at boundaries in this plot. This is an effect of the boundary conditions implemented in the hybrid DG/FV scheme. A similar behavior also occurs for other computations with open boundaries in this work.

4.5. Relativistic hydrodynamics

4.5.1. Convergence test

We start by performing a convergence test of the hybrid DG/FV scheme for the SRHD equations. We consider a exact solution of the SRHD equations proposed by He and Tang [78]. The problem consists in a wave propagating in the physical domain $[0, 2/\sqrt{2}] \times [0, 2]$ at an angle $\theta = 30°$ relative to the horizontal axis. The initial profile is

$$\rho = 1 + A \sin(\phi(0)), \quad v_x = v_0, \quad v_y = 0, \quad p = 1, \tag{82}$$

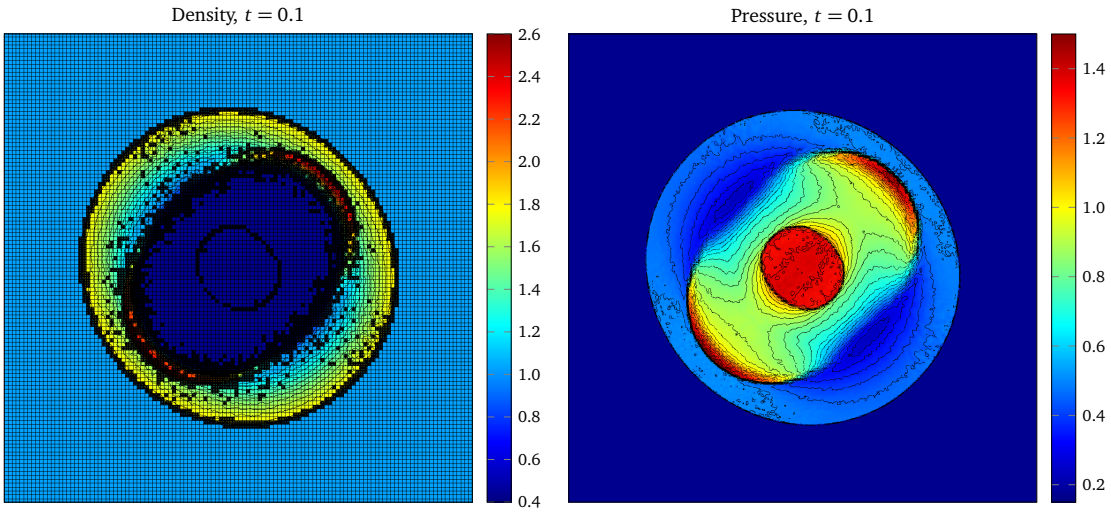

Figure 11: Cylindrical blast wave. Contour plots of the density (left) and pressure (right) at time $t = 0.1$ obtained with a hybrid DG/FV scheme, with polynomial degree $N = 5$. The computational domain is $\Omega = [0, 1] \times [0, 1]$, and it is discretized by using a mesh of 120×120 elements.

with $A = 0.2$, and $v_0 = 0.2$. This problem has the exact solution

$$\rho = 1 + A \sin(\phi(t)), \quad v_x = v_0, \quad v_y = 0, \quad p = 1. \tag{83}$$

The function $\phi(t)$ is defined as follows

$$\phi(t) = 2\pi \Big((x \cos\theta + y \sin\theta) - (v_x \cos\theta + v_y \sin\theta) t \Big). \tag{84}$$

The discretization of the computational domain satisfies $N_y = 2N_x$, where N_x, and N_y are, respectively, the number of elements in x- and y-direction. Periodic boundary conditions are imposed. In the table 4 are shown the convergence rates of the rest-mass density for the SRHD equations solved with the DGSEM and the hybrid DG/FV methods. The simulation time is $t = 1$, and for polynomials of degree $N = 3, 5, 7$. Observe that the EOC is achieved by the pure DGSEM, but the hybrid DG/FV scheme with all elements flagged as troubled is third order accurate, the order of the WENO3 reconstruction.

4.5.2. One-dimensional Riemann problems

In this section we will present four one-dimensional Riemann problems for the SRHD equations. These problems comprise the standard testbench for SRHD [79, 80]. Like for the MHD equations, for all test cases in this section we will assume that the one-dimensional domain is the interval $[0, 1]$. The membrane is localized at $x_{\rm m} = 0.5$. The numerical solution computed with the hybrid DG/FV scheme with polynomials of degree $N = 2, 4, 6$ is compared with the exact solution obtained from the exact Riemann solver by Rezzolla and Zanotti [81]. The left and right vector states in primitive variables are written as $\boldsymbol{w}_L = (\rho, v_x, v_y, p)_L$ and $\boldsymbol{w}_R = (\rho, v_x, v_y, p)_R$, respectively.

Riemann problem 1 (RP-1). The initial condition is given by the following left and right states

$$\boldsymbol{w}_L = (10, 0, 0, 40/3), \quad \boldsymbol{w}_R = (1, 0, 0, 0). \tag{85}$$

The EOS corresponds to the ideal gas EOS with adiabatic index $\gamma = 5/3$. This test problem is also known as the *mildly relativistic blast wave* [35, 82, 83]. The zero pressure in the right state is approximated by $p \approx 2/3 \times 10^{-6}$ for numerical reasons. Due to this very small value of the pressure, any oscillation generated in the approximate solution of the pressure will break up the simulation. A very robust shock capturing scheme is then required for handling such kind of problems. Plots of the rest-mass density, pressure, and the x-component of the velocity are depicted in the figure 14 (top). The domain has been discretized with 500 elements, and a polynomial of degree $N = 5$ is used in the

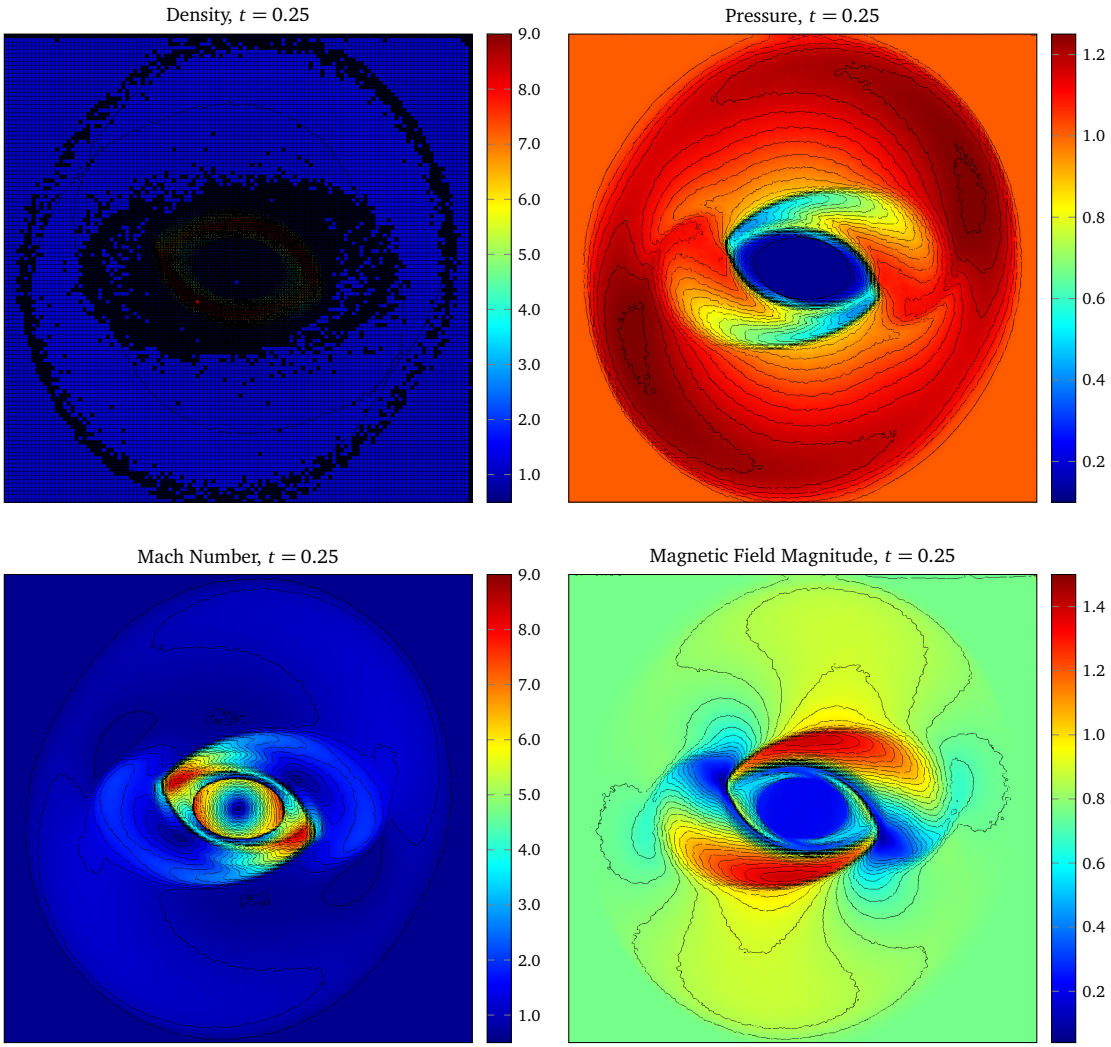

Figure 12: Rotor problem. Contour plots of the density, pressure, Mach number, and the magnitude of the magnetic field at time $t = 0.25$. These calculations were performed with a hybrid DG/FV scheme, with polynomial degree $N = 5$, and with WENO3 reconstruction in the FV solver. The computational domain is $\Omega = [0, 1] \times [0, 1]$, and it has been discretized by using a mesh of 120×120 elements.

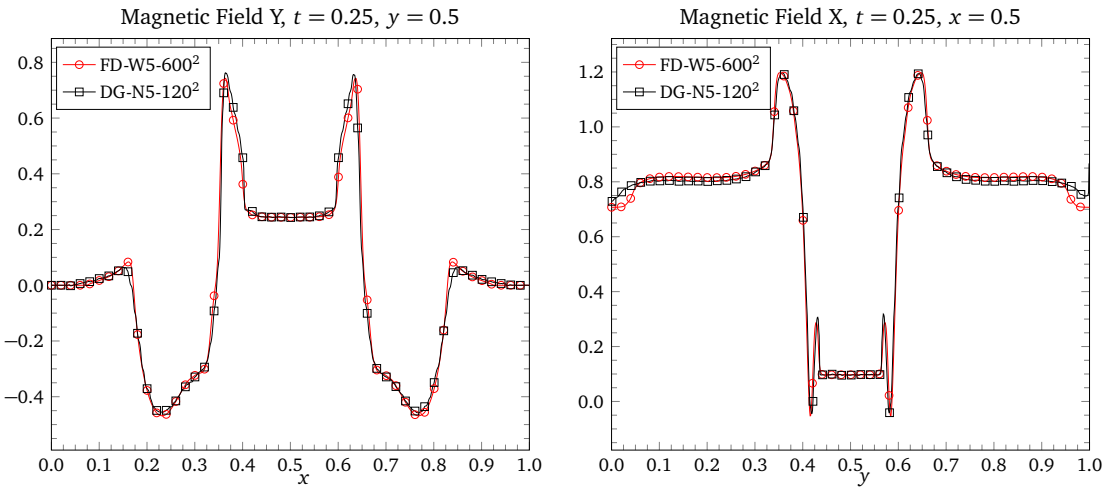

Figure 13: Rotor problem. Plots of the y-component of the magnetic field along the slice $y = 0.5$ (left) and plots of the x-component of the magnetic field along the slice $x = 0.5$ (right) at time $t = 0.25$. The computations were done with a very high-order finite difference WENO5 scheme and the hybrid DG/FV method with $N = 5$.

Table 4: Convergence rates for the SRHD equations with initial condition given by the *smooth flow problem*. Results for L_2 norm error of the rest-mass density are given. The computations were done with the pure DGSEM, and the hybrid DG/FV with all elements flagged as troubled. The schemes use polynomial degrees $N = 3, 5, 7$, that is for orders of accuracy $O(4)$, $O(6)$ and $O(8)$.

		DGSEM		Hybrid DG/FV	
Method	Elements	L_2 error	L_2 order	L_2 error	L_2 order
DG-$\mathcal{P}_3$	32×16	3.18×10^{-4}	—	4.76×10^{-4}	—
	64×32	1.87×10^{-5}	4.08	5.85×10^{-5}	3.03
	128×64	1.05×10^{-6}	4.16	7.08×10^{-6}	3.05
	256×128	5.75×10^{-8}	4.19	8.56×10^{-7}	3.05
DG-$\mathcal{P}_5$	16×8	4.38×10^{-7}	—	5.13×10^{-4}	—
	32×16	7.21×10^{-9}	5.93	6.52×10^{-5}	2.98
	64×32	1.13×10^{-10}	6.00	7.23×10^{-6}	3.17
	128×64	1.63×10^{-12}	6.11	7.98×10^{-7}	3.18
DG-$\mathcal{P}_7$	8×4	1.47×10^{-7}	—	1.37×10^{-3}	—
	16×8	6.98×10^{-10}	7.72	1.98×10^{-4}	2.79
	32×16	3.26×10^{-12}	7.74	2.79×10^{-5}	2.83
	64×32	1.62×10^{-14}	7.65	3.28×10^{-6}	3.09

hybrid DG/FV scheme. Dirichlet boundary conditions are set at both sides of the interval. The final simulation time is $t = 0.4$. The waves present in this problem are a left-going transonic rarefaction wave, a contact discontinuity and a right-going shock wave. The fluid behind the shock wave is traveling with a lightly relativistic speed $v = 0.72c$ to the right. In this dense shell behind the shock, the fluid is compressed. The fluid is thermodynamically relativistic because it is heated in such a way that the internal energy is much larger than the rest-mass energy, but mildly relativistic dynamically. The calculations are oscillations-free. This shows the robustness of the shock-capturing method.

Riemann problem 2 (RP-2). This problem, also called the *highly relativistic blast wave*, has the following initial state

$$\boldsymbol{w}_L = (1, 0, 0, 10^3), \quad \boldsymbol{w}_R = (1, 0, 0, 10^{-2}). \tag{86}$$

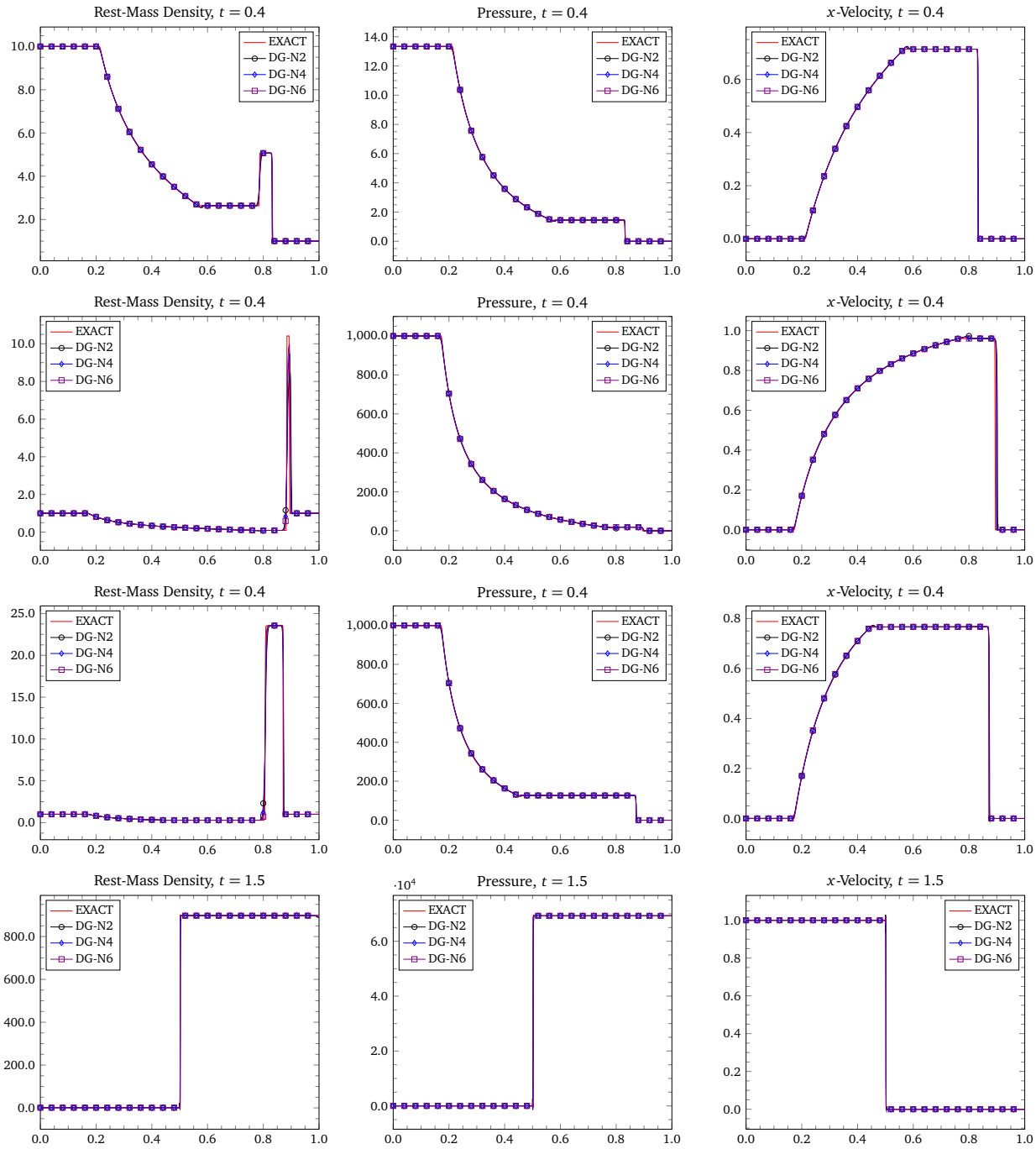

Figure 14: One-dimensional Riemann problems. Plots for the Riemann problems RP-1, RP-2, RP-3, and RP-4. The computations were performed with the hybrid DG/FV scheme with polynomials of degree $N = 2, 4, 6$, and with WENO3 reconstruction operator in the FV solver.

The ideal gas EOS with adiabatic index $\gamma = 5/3$ has been employed. The interval is discretized with 500 elements. Dirichlet boundary conditions have been utilized at domain edges. The final simulation time is $t = 0.4$. Plots of the rest-mass density, pressure, and the x-component of the velocity are depicted in the figure 14 (second row, from top to bottom). After the breakup of the membrane, a dense intermediate state is formed, and it is located between the right-traveling shock wave and the rarefaction wave moving to the left. Observe that the shock wave and the contact discontinuity are very close to each other. The very thin and dense shell is rather well resolved thanks to the subcell resolution of the hybrid DG/FV method. It is noticeable from the plots the lack of oscillations.

Riemann problem 3 (RP-3). The *transverse blast wave* problem has initial conditions similar to the RP-2, except that the transverse velocity in the right state is non-zero. The initial left and right states are

$$\boldsymbol{w}_L = (1, 0, 0, 10^3), \quad \boldsymbol{w}_R = (1, 0, 0.99, 10^{-2}). \tag{87}$$

We have employed the ideal gas EOS with adiabatic index $\gamma = 5/3$. The domain has 500 elements, and the final simulation time is $t = 0.4$. Plots of the rest-mass density, pressure, and the x-component of the velocity are depicted in the figure 14 (second row, from bottom to top). It worths to mention that in Newtonian hydrodynamics the transverse momentum is not coupled with the longitudinal one and it thus is simply advected. The momentum equations in SRHD are coupled to each other through the Lorentz factor. This introduces new physical effects, especially with non-zero transverse velocities [84, 85]. Observe that the solution is in very good agreement with the exact solution, and the computed solution is oscillation-free.

Riemann problem 4 (RP-4). In the *planar shock reflection* an ideal cold fluid collides a wall. Afterwards a shock wave propagates backwards, leaving the gas behind at rest. The initial state is given by

$$\boldsymbol{w} = (1, 0.99999, 0, 0.01) \tag{88}$$

The reflecting wall is located in $x = 1$, and at $x = 0$ transmissive boundary conditions are imposed. An ideal gas EOS with adiabatic index $\gamma = 4/3$ is used. The domain is discretized with 500 elements. The final simulation time is $t = 1.5$. The exact solution of this problem was first obtained by Blandford and McKee [86]. Plots of the rest-mass density, pressure, and the x-component of the velocity computed with the hybrid DG/FV scheme are depicted in the figure 14 (bottom). Close to the wall ($x = 1$), the numerical solution of the rest-mass density does not show an undershooting, as it occurs in the solution computed with finite-volume or finite difference methods [36]. The undershooting is due to the wall heating phenomenon [87]. The shock has been very well captured with the hybrid scheme, but some small oscillations are still visible around it.

4.5.3. Two-dimensional Riemann problems

Two-dimensional Riemann problems are now considered. The initial states have been taken from [38, 88]. The computational domain is $\Omega = [-1, 1]\times[-1, 1]$, and it has been divided into four quadrants. The separating membranes are located along the axes x and y. In these Riemann problems are present all essential features typically found in a two-dimensional flow, for instance, shock reflections, shock interactions, vortices, etc. We write the initial states for every quadrant ($Q1$, $Q2$, $Q3$, and $Q4$)

$$\begin{aligned}
Q1 &:= \left\{(x, y) \in [-1, 1]^2 \,\middle|\, x \geq 0,\ y \geq 0\right\}, \\
Q2 &:= \left\{(x, y) \in [-1, 1]^2 \,\middle|\, x < 0,\ y \geq 0\right\}, \\
Q3 &:= \left\{(x, y) \in [-1, 1]^2 \,\middle|\, x < 0,\ y < 0\right\}, \\
Q4 &:= \left\{(x, y) \in [-1, 1]^2 \,\middle|\, x \geq 0,\ y < 0\right\}.
\end{aligned} \tag{89}$$

For all Riemann problems, an ideal gas EOS with adiabatic index $\gamma = 5/3$ is used, and transmissive boundary conditions are used in all faces of the computational domain.

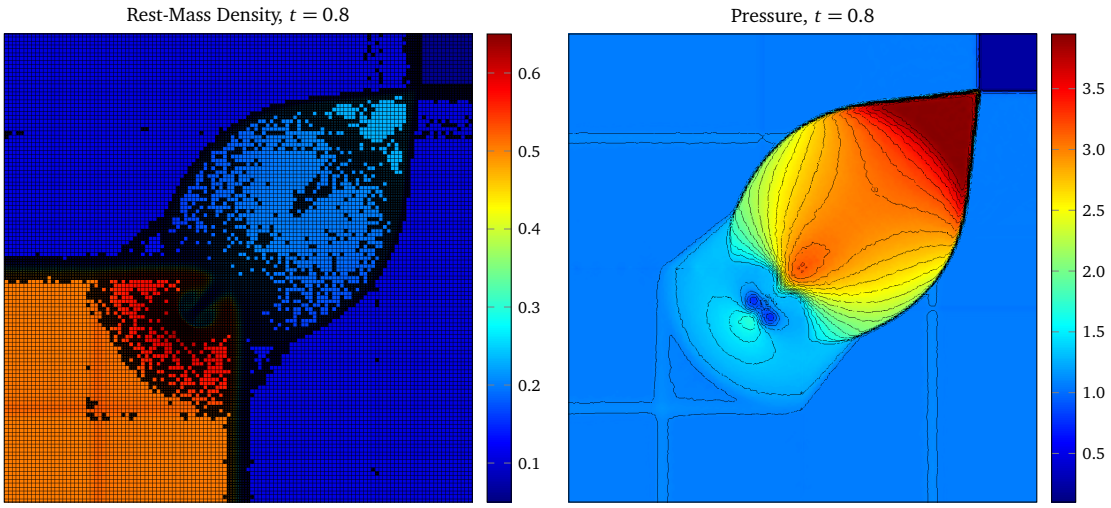

Figure 15: Two-dimensional Riemann problem RP-5. Contour plots of the rest-mass density (left) and pressure (right) at $t = 0.8$. The solution was obtained with a hybrid DG/FV scheme with polynomials of degree $N = 5$. The domain $\Omega = [-1, 1] \times [-1, 1]$ was decomposed into 120×120 elements.

Riemann problem 5 (RP-5). The domain is subdivided into four quadrants, where the states in primitive variables are given by

$$\begin{aligned} \boldsymbol{w}_{Q1} &= (0.035145216124503, 0, 0, 0.162931056509027), \\ \boldsymbol{w}_{Q2} &= (0.1, 0.7, 0, 1), \\ \boldsymbol{w}_{Q3} &= (0.5, 0, 0, 1), \\ \boldsymbol{w}_{Q4} &= (0.1, 0, 0.7, 1). \end{aligned} \tag{90}$$

The final simulation time is $t = 0.8$. In the figure 15 are depicted contour plots of the rest-mass density (along the mesh) and pressure. This problem features contact discontinuities and shock waves. After the breakup of the membranes, two contact discontinuities are originated on the left and bottom of the domain, and in the first quadrant two curved front shocks appear. In the third quadrant a like-jet structure moving in south-west direction is present [88]. The hybrid DG/FV scheme is quite robust and allows to compute in a multidimensional fashion such complex flow patterns. The approximated solution computed with the hybrid DG/FV scheme using polynomials of degree $N = 5$ on a mesh of 120×120 elements in good agreement with the reported in the literature [40, 88].

Riemann problem 6 (RP-6). The initial condition is given by the following states in primitive variables

$$\begin{aligned} \boldsymbol{w}_{Q1} &= (0.5, 0.5, -0.5, 5), \\ \boldsymbol{w}_{Q2} &= (1, 0.5, 0.5, 5), \\ \boldsymbol{w}_{Q3} &= (3, -0.5, 0.5, 5), \\ \boldsymbol{w}_{Q4} &= (1.5, -0.5, -0.5, 5). \end{aligned} \tag{91}$$

The final simulation time is $t = 0.8$. In the figure 16 are shown contour plots of the rest-mass density and the Lorentz factor. The mesh with the FV subcells is also shown. From the plots can be seen the interaction of four vortex sheets, which form a spiral with very low rest-mass density in the center of the domain. The hybrid DG/FV scheme using polynomials of degree $N = 5$ on a mesh of 120×120 elements has an excellent behavior in the very low density region, which means that the scheme does not produce negative densities or pressures even in regions with cavitation.

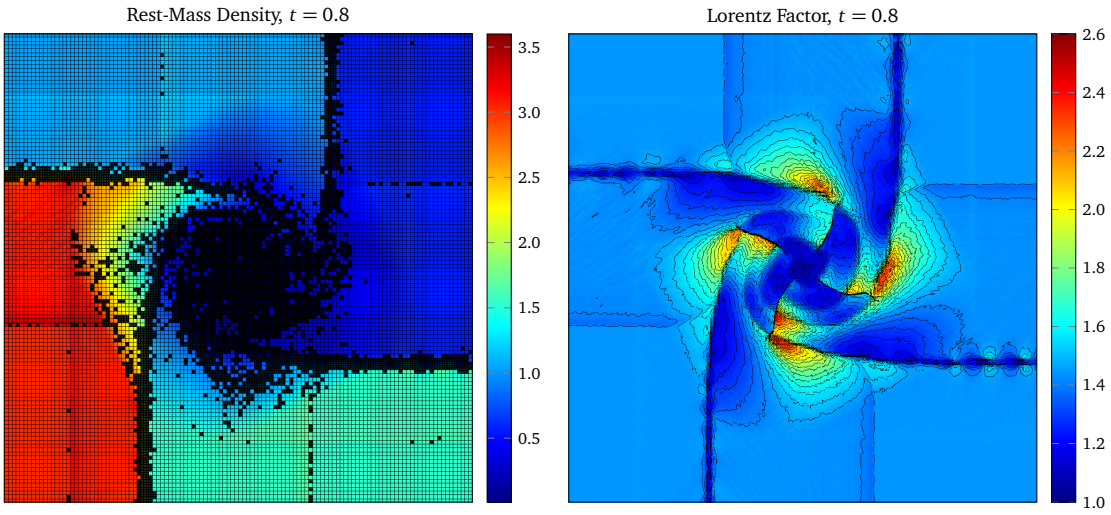

Figure 16: Two-dimensional Riemann problem RP-6. Contour plots of the rest-mass density (left) and Lorentz factor (right) at $t = 0.8$. The solution was obtained with a hybrid DG/FV scheme with polynomials of degree $N = 5$. The domain $\Omega = [-1, 1] \times [-1, 1]$ was decomposed into 120×120 elements.

Riemann problem 7 (RP-7). In this two-dimensional Riemann problem, the initial condition is given as follows

$$\begin{aligned} \boldsymbol{w}_{Q1} &= (1, 0, 0, 1), \\ \boldsymbol{w}_{Q2} &= (0.5771, -0.3529, 0, 0.4), \\ \boldsymbol{w}_{Q3} &= (1, -0.3529, -0.3529, 1), \\ \boldsymbol{w}_{Q4} &= (0.5771, 0, -0.3529, 0.4). \end{aligned} \tag{92}$$

The final simulation time is $t = 0.8$. In the figure 17 are shown contour plots of the rest-mass density and the Mach number computed with a hybrid DG/FV scheme using polynomials of degree $N = 5$ on a mesh of 120×120 elements. In this problem, the planar rarefaction waves originated after the breakup of the membranes interact each other. This interaction generates two symmetric shock waves in the zone where the rarefaction waves have interplayed. Observe the FV subdomains in and around the regions where the rarefaction waves are present.

4.5.4. Cloud-shock interaction

In this problem, we follow the initial setup given by He and Tang [78]. An ideal gas EOS with adiabatic index $\gamma = 5/3$ is employed in the simulation. The computational domain is $\Omega = [0, 2] \times [0, 1]$. Dirichlet boundary conditions are imposed. The center of the cylindrical cloud is located in the point $(1.4, 0.5)$ and its radius is $r = 0.15$. The initial state for the cloud in primitive variables is $\boldsymbol{w}_c = (3.1538, 0, 0, 0.05)$. The left traveling shock wave at time $t = 0$ is located at $x = 1.6$, with left and right states given by

$$\begin{aligned} \boldsymbol{w}_L &= (1, 0, 0, 0.05), \\ \boldsymbol{w}_R &= (1.86522508063118, -0.19678110737829, 0, 0.15). \end{aligned} \tag{93}$$

We discretize the computational domain with 160×80 elements. In the figure 18 are depicted contour plots of the rest-mass density and Mach number at simulation time $t = 3.0$. The mesh and the FV subdomains are also plotted. The computations were done with the hybrid DG/FV scheme with polynomials of degree $N = 5$. The profile of the bubble after the interaction with the shock wave is in good agreement with the reported in the literature [36, 78]. The computation was stable and negative densities or pressures were not generated.

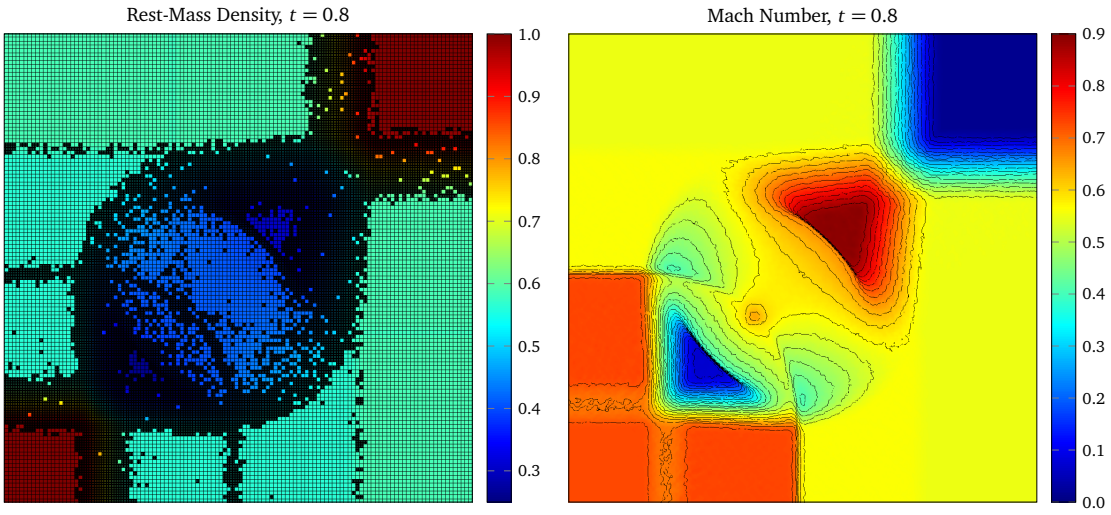

Figure 17: Two-dimensional Riemann problem RP-7. Contour plots of the rest-mass density (left) and Mach number (right) at $t = 0.8$. The solution was obtained with a hybrid DG/FV scheme with polynomials of degree $N = 5$. The domain $\Omega = [-1, 1] \times [-1, 1]$ was decomposed into 120×120 elements.

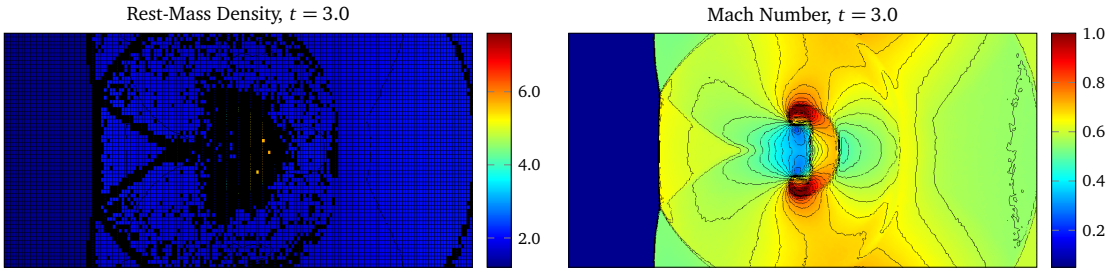

Figure 18: Relativistic cloud-shock interaction. Contour plots of the rest-mass density (left), and Mach number (right) at time $t = 3.0$. The computations were done with a hybrid DG/FV scheme with polynomials of degree $N = 5$. The computational domain is the $\Omega = [0, 2] \times [0, 1]$, and it is discretized by using a mesh of 160×80 elements.

5. Conclusions

In this work we have presented the buildings blocks of a high-order hybrid DG/FV scheme for MHD and SRHD. The hybrid scheme is based on a discontinuous Galerkin spectral element on quadrilateral/hexahedral meshes as main spatial discretization operator, and it is employed basically in regions with smooth flows. For those regions containing discontinuities, a robust second/third order finite volume scheme is used as shock capturing approach. In order to handle discontinuities, an oscillations indicator is used to detect and to mark the elements containing them. These troubled elements are projected onto a subdomain made of FV subcells, and then evolved with the robust FV method. In this way the very high-order hybrid method is capable of solving problems with shock waves. Time discretization is performed with a fourth-order SSPRK method. The solenoidal constraint is maintained with the GLM divergence cleaning. From our experience in the area of very high-order FV-WENO schemes for computational astrophysics [33, 36], the DGSEM domains are evolved using the conservative variables, while in the FV subdomains the solution is computed by employing the primitive variables. This choice provides more stability to the computations.

A comprehensive testbench for MHD and SRHD was discussed. One- and two-dimensional test problems were presented. Convergence tests have been also discussed in order to check the order of accuracy of the hybrid scheme for both equations systems. The computations show that the hybrid scheme converges to the theoretical order of accuracy. For problems with shocks, the hybrid scheme captures very well the discontinuities, even in very cases with extreme density or pressure conditions. Polynomials up to sixth order degree were used. These results demonstrated that the

hybrid DG/FV scheme is an excellent candidate as a numerical solver for equations systems typical in computational astrophysics. Among other outstanding properties of the hybrid scheme we can mention: the method is highly parallelizable [26], allows unstructured quadrilateral/hexahedral meshes, and it is very robust when strong shocks take place in the simulation. Further research is focused on hybrid DG/FV schemes with quad-/octree based adaptive mesh refinement in order to simulate in a more efficient manner small structures in magnetohydrodynamic and relativistic turbulence simulations. Furthermore, the proposed method can also be used in the context of polygon/polyhedron computational meshes. Some similar work has been done in that direction by other researchers [89]. This will be useful for some CFD codes for astrophysics which make use of DG schemes on moving unstructured grids defined as the Voronoi tessellation of a set of mesh-generating points, which will require such efficient shock-capturing schemes.

Acknowledgments

This research was funded by the Deutscher Akademischer Austauschdienst under grant A0873943. Further support was provided by the Deutsche Forschungsgemeinschaft, grant MU 1319/12-2. We thank Michael Dumbser for the very careful reading of the manuscript and their valuable suggestions, which improved the quality of this paper. Many thanks to Luciano Rezzolla, and Olindo Zanotti for providing the exact solution of the one-dimensional SRHD Riemann problems discussed in this work.

References

[1] B. Cockburn, C.-W. Shu, J. Sci. Comput. 16 (2001) 173–261. doi:10.1023/A:1012873910884.
[2] C. Canuto, M. Hussaini, A. Quarteroni, T. Zang, Spectral Methods: Fundamentals in Single Domains, Springer, Berlin, 2006.
[3] D. Kopriva, Implementing Spectral Methods for Partial Differential Equations, Springer, Berlin, 2009.
[4] J. Nitsche, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 36 (1971) 9–15. doi:10.1007/BF02995904.
[5] W. Reed, T. Hill, Triangular mesh methods for the neutron transport equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[6] B. Cockburn, C.-W. Shu, Math. Comp. 52 (1989) 411–435. doi:10.1090/S0025-5718-1989-0983311-4.
[7] B. Cockburn, S.-Y. Lin, C.-W. Shu, J. Comput. Phys. 84 (1989) 90–113. doi:10.1016/0021-9991(89)90183-6.
[8] B. Cockburn, S. Hou, C.-W. Shu, Math. Comp. 54 (1990) pp.545–581. doi:10.1090/S0025-5718-1990-1010597-0.
[9] B. Cockburn, C.-W. Shu, J. Comput. Phys. 141 (1998) 199–224. doi:10.1006/jcph.1998.5892.
[10] J. W. Gibbs, Nature 59 (1898) 200. doi:10.1038/059200b0.
[11] J. W. Gibbs, Nature 59 (1899) 606. doi:10.1038/059606a0.
[12] A. Harten, P. Lax, B. van Leer, SIAM Rev. 25 (1983) 35–61. doi:10.1137/1025002.
[13] A. Harten, J. Comput. Phys. 49 (1983) 357–393. doi:10.1016/0021-9991(83)90136-5.
[14] G.-S. Jiang, C.-W. Shu, J. Comput. Phys. 126 (1996) 202–228. doi:10.1006/jcph.1996.0130.
[15] C.-W. Shu, SIAM Rev. 51 (2009) 82–126. doi:10.1137/070679065.
[16] R. Biswas, K. Devine, J. Flaherty, Appl. Numer. Math. 14 (1994) 255–283. doi:10.1016/0168-9274(94)90029-9.
[17] L. Krivodonova, J. Comput. Phys. 226 (2007) 879–896. doi:10.1016/j.jcp.2007.05.011.
[18] P.-O. Persson, J. Peraire, in: Proceedings of the AIAA 44th Aerospace Sciences Meeting and Exhibit, AIAA-2006-0112, Reno, Nevada.
[19] E. Casoni, J. Peraire, A. Huerta, Int. J. Numer. Meth. Fluids 71 (2013) 737–755. doi:10.1002/fld.3682.
[20] J. Qiu, C.-W. Shu, SIAM J. Sci. Comput. 26 (2005) 907–929. doi:10.1137/S1064827503425298.
[21] J. Zhu, J. Qiu, C.-W. Shu, M. Dumbser, J. Comput. Phys. 227 (2008) 4330–4353. doi:10.1016/j.jcp.2007.12.024.
[22] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, J. Comput. Phys. 248 (2013) 200–220. doi:10.1016/j.jcp.2013.04.012.
[23] M. Sonntag, C.-D. Munz, in: J. Fuhrmann, M. Ohlberger, C. Rohde (Eds.), Finite volumes for complex applications VII-Elliptic, parabolic and hyperbolic problems, volume 78 of *Springer Proceedings in Mathematics & Statistics*, Springer International Publishing, 2014, pp. 945–953. doi:10.1007/978-3-319-05591-6_96.
[24] M. Dumbser, O. Zanotti, R. Loubère, S. Diot, J. Comput. Phys. 278 (2014) 47–75. doi:10.1016/j.jcp.2014.08.009.
[25] J. Núñez-de la Rosa, High-Order Methods for Computational Astrophysics, Dr. Hut Verlag, Munich, Germany, 2015. doi:10.18419/opus-3980.
[26] M. Sonntag, C.-D. Munz, J. Sci. Comput. (2016) 1–28. doi:10.1007/s10915-016-0287-5.
[27] D. Radice, L. Rezzolla, Phys. Rev. D 84 (2011) 024010. doi:10.1103/PhysRevD.84.024010.
[28] D. Ryu, T. Jones, Astrophys. J. 442 (1995) 228–258. doi:10.1086/175437.
[29] D. Ryu, T. Jones, Astrophys. J. 452 (1995) 785–796. doi:10.1086/176347.
[30] W. Dai, P. Woodward, J. Comput. Phys. 111 (1994) 354–372. doi:10.1006/jcph.1994.1069.
[31] W. Dai, P. Woodward, J. Comput. Phys. 115 (1994) 485–514. doi:10.1006/jcph.1994.1212.
[32] A. Mignone, G. Bodo, S. Massaglia, T. Matsakos, O. Tesileanu, C. Zanni, A. Ferrari, Astrophys. J. Suppl. Ser. 170 (2007) 228–242. doi:10.1086/513316.
[33] J. Núñez-de la Rosa, C.-D. Munz, Mon. Not. R. Astron. Soc. 455 (2016) 3458–3479. doi:10.1093/mnras/stv2531.
[34] P. Woodward, P. Colella, J. Comput. Phys. 54 (1984) 115–173. doi:10.1016/0021-9991(84)90142-6.
[35] J. M. Martí, E. Müller, J. Comput. Phys. 123 (1996) 1–14. doi:10.1006/jcph.1996.0001.

[36] J. Núñez-de la Rosa, C.-D. Munz, Mon. Not. R. Astron. Soc. 460 (2016) 535–559. doi:10.1093/mnras/stw999.
[37] W. Zhang, A. MacFadyen, Astrophys. J. Suppl. Ser. 164 (2006) 255. doi:10.1086/500792.
[38] A. Tchekhovskoy, J. McKinney, R. Narayan, Mon. Not. R. Astron. Soc. 379 (2007) 469–497. doi:10.1111/j.1365-2966.2007.11876.x.
[39] D. Radice, L. Rezzolla, Astron. Astrophys. 547 (2012) A26. doi:10.1051/0004-6361/201219735.
[40] O. Zanotti, M. Dumbser, Comput. Phys. Commun. 188 (2015) 110–127. doi:10.1016/j.cpc.2014.11.015.
[41] J. Brackbill, D. Barnes, J. Comput. Phys. 35 (1980) 426–430. doi:10.1016/0021-9991(80)90079-0.
[42] K. Powell, An approximate Riemann solver for magnetohydrodynamics, Technical Report, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, 1994.
[43] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, J. Comput. Phys. 175 (2002) 645–673. doi:10.1006/jcph.2001.6961.
[44] C. Evans, J. Hawley, Astrophys. J. 332 (1988) 659–677. doi:10.1086/166684.
[45] A. Mignone, P. Tzeferacos, G. Bodo, J. Comput. Phys. 229 (2010) 5896–5920. doi:10.1016/j.jcp.2010.04.013.
[46] C.-D. Munz, R. Schneider, E. Sonnendrücker, U. Voss, C. R. Acad. Sci. Math. 328 (1999) 431–436. doi:10.1016/S0764-4442(99)80185-2.
[47] L. Rezzolla, O. Zanotti, Relativistic Hydrodynamics, Oxford University Press, Oxford, 2013.
[48] A. Mignone, T. Plewa, G. Bodo, Astrophys. J. Suppl. Ser. 160 (2005) 199–219. doi:10.1086/430905.
[49] D. Ryu, I. Chattopadhyay, E. Choi, Astrophys. J. Suppl. Ser. 166 (2006) 410–420. doi:10.1086/505937.
[50] A. Anile, Relativistic Fluids and Magnetofluids, Cambridge University Press, Cambridge, 1989.
[51] V. Schneider, U. Katscher, D. Rischke, B. Waldhauser, J. Maruhn, C.-D. Munz, J. Comput. Phys. 105 (1993) 92–107. doi:10.1006/jcph.1993.1056.
[52] C. Duncan, P. Hughes, Astrophys. J. 436 (1994) L119–L122. doi:10.1086/187647.
[53] D. Kopriva, J. Kolias, J. Comput. Phys. 125 (1996) 244–261. doi:10.1006/jcph.1996.0091.
[54] D. Kopriva, J. Comput. Phys. 128 (1996) 475–488. doi:10.1006/jcph.1996.0225.
[55] F. Hindenlang, G. Gassner, C. Altmann, A. Beck, M. Staudenmaier, C.-D. Munz, Comput. Fluids 61 (2012) 86–93. doi:10.1016/j.compfluid.2012.03.006.
[56] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, 3 ed., Springer, Berlin, 2009.
[57] V. Rusanov, J. Comput. Math. Phys. (USSR) 1 (1961) 267–279.
[58] A. Susanto, L. Ivan, H. D. Sterck, C. Groth, J. Comput. Phys. 250 (2013) 141–164. doi:10.1016/j.jcp.2013.04.040.
[59] W. Schiesser, The Numerical Method of Lines: Integration of Partial Differential Equations, Academic Press, 1991.
[60] C.-W. Shu, S. Osher, J. Comput. Phys. 77 (1988) 439–471. doi:10.1016/0021-9991(88)90177-5.
[61] C.-W. Shu, SIAM J. Sci. Comput. 9 (1988) 1073–1084. doi:10.1137/0909073.
[62] R. Spiteri, S. Ruuth, SIAM J. Numer. Anal. 40 (2002) 469–491. doi:10.1137/S0036142901389025.
[63] R. Courant, K. Friedrichs, H. Lewy, Math. Ann. 100 (1928) 32–74. doi:10.1007/BF01448839.
[64] Z.-J. Wang, Prog. Aerosp. Sci. 43 (2007) 1–41. doi:10.1016/j.paerosci.2007.05.001.
[65] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, J. Comput. Phys. 71 (1987) 231–303. doi:10.1016/0021-9991(87)90031-3.
[66] K. Beckwith, J. Stone, Astrophys. J. Suppl. Ser. 193 (2011) 6–35. doi:10.1088/0067-0049/193/1/6.
[67] M. Brio, C.-C. Wu, J. Comput. Phys. 75 (1988) 400–422. doi:10.1016/0021-9991(88)90120-9.
[68] M. Torrilhon, J. Plasma Phys. 69 (2003) 253–276. doi:10.1017/S0022377803002186.
[69] G. Tóth, J. Comput. Phys. 161 (2000) 605–652. doi:10.1006/jcph.2000.6519.
[70] J. Hawley, J. Stone, Comput. Phys. Commun. 89 (1995) 127–148. doi:10.1016/0010-4655(95)00190-Q.
[71] S. Orszag, C.-M. Tang, J. Fluid Mech. 90 (1979) 129–143. doi:10.1017/S002211207900210X.
[72] A. Zachary, A. Malagoli, P. Colella, SIAM J. Sci. Comput. 15 (1994) 263–284. doi:10.1137/0915019.
[73] W. Dai, P. Woodward, Astrophys. J. 494 (1998) 317–335. doi:10.1086/305176.
[74] G.-S. Jiang, C.-C. Wu, J. Comput. Phys. 150 (1999) 561–594. doi:10.1006/jcph.1999.6207.
[75] P. Londrillo, L. del Zanna, Astrophys. J. 530 (2000) 508–524. doi:10.1086/308344.
[76] J. Stone, T. Gardiner, P. Teuben, J. Hawley, J. Simon, Astrophys. J. Suppl. Ser. 178 (2008) 137. doi:10.1086/588755.
[77] D. Balsara, D. Spicer, J. Comput. Phys. 149 (1999) 270–292. doi:10.1006/jcph.1998.6153.
[78] P. He, H. Tang, Comm. Comput. Phys. 11 (2012) 114–146. doi:10.4208/cicp.291010.180311a.
[79] J. M. Martí, E. Müller, Living Rev. Relat. 6 (2003) 1–100. doi:10.12942/lrr-2003-7.
[80] J. M. Martí, E. Müller, Living Rev. Comput. Astrophys. 1 (2015) 1–182. doi:10.1007/lrca-2015-3.
[81] L. Rezzolla, O. Zanotti, J. Fluid Mech. 449 (2001) 395–. doi:10.1017/S0022112001006450.
[82] J. M. Martí, E. Müller, J. Fluid Mech. 258 (1994) 317–333. doi:10.1017/S0022112094003344.
[83] R. Donat, J. A. Font, J. M. Ibáñez, A. Marquina, J. Comput. Phys. 146 (1998) 58–81. doi:10.1006/jcph.1998.5955.
[84] J. A. Pons, J. M. Martí, E. Müller, J. Fluid Mech. 422 (2000) 125–139. doi:10.1017/S0022112000001439.
[85] L. Rezzolla, O. Zanotti, J. A. Pons, J. Fluid Mech. 479 (2003) 199–219. doi:10.1017/S0022112002003506.
[86] R. Blandford, C. McKee, Phys. Fluids 19 (1976) 1130–1138. doi:10.1063/1.861619.
[87] W. F. Noh, J. Comput. Phys. 72 (1987) 78–120. doi:10.1016/0021-9991(87)90074-X.
[88] L. del Zanna, N. Bucciantini, Astron. Astrophys. 390 (2002) 1177–1186. doi:10.1051/0004-6361:20020776.
[89] M. Dumbser, R. Loubère, J. Comput. Phys. 319 (2016) 163–199. doi:10.1016/j.jcp.2016.05.002.