

A unified framework for the solution of hyperbolic PDE systems using high order direct Arbitrary-Lagrangian-Eulerian schemes on moving unstructured meshes with topology change

Elena Gaburro

Received: date / Accepted: date

Abstract In this work, we review the family of *direct* Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume (FV) and Discontinuous Galerkin (DG) schemes on *moving* meshes that at each time step are *rearranged* by explicitly allowing *topology changes*, in order to guarantee a robust mesh evolution even for high shear flow and very long evolution times. Two different techniques are presented: a local non-conforming approach for dealing with sliding lines, and a global regeneration of Voronoi tessellations for treating general unpredicted movements.

Corresponding elements at consecutive times are connected in space-time to construct closed *space-time* control volumes, whose bottom and top faces may be polygons with a different number of nodes, with different neighbors, and even degenerate *space-time sliver elements*. Our final ALE FV-DG scheme is obtained by integrating, over these arbitrary shaped space-time control volumes, the space-time conservation formulation of the governing hyperbolic PDE system: so, we directly evolve the solution in time avoiding any remapping stage, being conservative and satisfying the GCL by construction. *Arbitrary high order* of accuracy in space and time is achieved through a fully discrete one-step predictor-corrector ADER approach, also integrated with *well balancing* techniques to further improve the accuracy and to maintain *exactly* even at discrete level many physical *invariants* of the studied system.

A large set of different numerical tests has been carried out in order to check the accuracy and the robustness of our methods for both smooth and discontinuous problems, in particular in the case of vortical flows.

Keywords Hyperbolic Partial Differential Equations (PDE) · Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume (FV) and Discontinuous Galerkin (DG) schemes · Arbitrary High Order in space and time · Nonconforming meshes · Sliding lines · Moving Voronoi Tessellations with Topology Change · *A posteriori* sub-cell Finite Volume Limiter · Fully-discrete one-step ADER approach · Compressible Euler equations · MHD equations

1 Introduction

Lagrangian-type Finite Volume (FV) and Discontinuous Galerkin (DG) schemes are characterized by a moving computational mesh: at each time step the new position of all the nodes is recomputed according to a prescribed mesh velocity, which generally is chosen as close as possible to the local fluid velocity. In the Lagrangian description of the fluid the nonlinear convective terms disappear and Lagrangian schemes exhibit virtually no numerical dissipation at contact waves and material interfaces. So the aim of these methods is to reduce the numerical dissipation errors due to the convective terms, hence to capture contact discontinuities sharply and to precisely identify and track material interfaces. Among the different variants, that will be briefly recalled below, we would like to underline the role of the *direct* Arbitrary-Lagrangian-Eulerian (ALE) schemes. In this case the mesh velocity can be chosen in an arbitrary way, so usually it is chosen close to the fluid velocity, but the fact that it can be slightly modified allows first, some flexibility in the mesh motion reducing the mesh distortion, second let us perform the rezoning before the computation of the numerical flux, so that the remapping stage is no more needed, and finally it naturally extends to complex unstructured meshes, to slide lines treatment and to topology changes.

E. Gaburro
Department of Civil, Environmental and Mechanical Engineering, University of Trento,
Via Mesiano 77, 38123 Trento, Italy
E-mail: elena.gaburro@unitn.it

The *key point* of this work on *direct* ALE schemes lies in the use of closed, non-overlapping, *space-time* control volumes of *arbitrary shapes*, constructed via connecting the two meshes at time t^n and t^{n+1} ; then, on these closed space-time control volumes, the *space-time conservation formulation* of the governing PDE system is *directly* integrated by means of a high order fully discrete one-step ADER method, without needing of any further remapping steps. This allows to reach easily both arbitrary high order of accuracy in space and time and robust mesh motion.

1.1 State of the art

Lagrangian schemes were already of interest in the fifties when John von Neumann and Richtmyer were working on the one-dimensional case [124] and Wilkins proposed a two-dimensional extension in 1964 [154]. So, here we will only briefly recall the major contributions to the topic and for a complete review on general Lagrangian-type methods we refer the reader to [6, 123, 50, 76].

Examples of low order Lagrangian cell centered Finite Volume schemes, based on a conservation form of the equations involving mass, momentum and total energy, can be found in [123, 30, 43, 138, 116, 119, 118, 117]. Higher order Lagrangian-type schemes have been introduced in [38, 110, 39], where high order of accuracy in space is achieved with the aid of a ENO/WENO reconstruction and Runge-Kutta time stepping guarantees high order time discretization as well. Then, also *staggered* Lagrangian schemes [112, 113, 114] have been introduced to define the velocity at the grid vertexes and the other variables at the cell center, hence avoiding the need of a nodal solver [38, 110, 30, 116] to compute the mesh velocity of the grid nodes.

Another option for the numerical solution of hyperbolic conservation laws is given by Discontinuous Galerkin [136] and Finite Element (FE) schemes [45, 46, 47], where the numerical solution is approximated by piecewise polynomials within each control volume. Lagrangian DG schemes up to third order have been proposed for the first time in [75, 73, 74, 108], while high order FE methods applied to Lagrangian hydrodynamics and elasto-plasticity can be found in [127, 140, 53, 55, 54].

Although all these different schemes are widely used, a common problem that affects all Lagrangian methods is the severe mesh *distortion* or mesh tangling that happens in the presence of shear flows and that may even destroy the computation. Hence, all Lagrangian methods must be in general combined with an algorithm to (locally) *rezone* the mesh at least from time to time and to remap the solution from the old mesh to the new mesh in a conservative manner. This also led to the development of the Arbitrary-Lagrangian-Eulerian (ALE) methods [138, 9, 102, 109, 100, 7, 3], where the mesh velocity can be chosen *independently*

of the local fluid velocity and thus the grid nodes can be moved at an arbitrary velocity. The ALE schemes are generally divided into two families: the *indirect* ALE schemes [155, 99, 82] characterized by a purely Lagrangian phase followed by a remesh and a subsequent remap phase, and the *direct* ALE schemes [13, 14, 15], where the local rezoning is performed before the computation of the numerical fluxes, hence changing directly the chosen mesh velocity of the ALE approach. The *direct* approach naturally extends to unstructured meshes in multiple space dimensions [10] and to arbitrary high order both in the FV and DG case [18, 14, 15, 17, 12], and can also be combined with time-accurate local time stepping (LTS), see [56, 21, 96].

However, all the works characterized by a *fixed* mesh topology make it impossible to study phenomena affected by strong shear motion and vortex flows for very long simulation times, since mesh tangling would *inevitably* occur and lead to a breakdown of the simulation before the final time is reached. To overcome this problem some solutions, alternative to the remesh/remap approach, have been proposed in literature: i) special treatment of sliding lines [28, 133, 101, 41, 42, 29, 133, 8], which deal with moving *interfaces* so that elements on the two sides can slide in a *nonconforming* way in order to accommodate the distortion induced by shear flows; ii) a very original solution represented by the *AREPO* code of Springel and collaborators [143, 144, 129, 130], i.e. a second order direct ALE FV scheme where the connectivity of the moving mesh is dynamically regenerated via a moving unstructured but conforming Voronoi tessellation of the domain; iii) the well-known particle finite element method of Oñate and Idelsohn *et al.*, see [94, 132, 126, 103, 93, 125], where again the mesh is completely regenerated at each time step.

1.2 Structure of the paper

This paper is organized as follow. In Section 2 we will derive the *direct* ALE schemes on unstructured polygonal grids in a *unified framework* that already involves both the Finite Volume approach and the Discontinuous Galerkin approach at arbitrary high order of accuracy in space and time through the predictor-corrector ADER paradigm.

Then in what follows we will concentrate on the two dimensional case and on two novel techniques for a robust mesh motion based on the *dynamical changes* at each time step of the grid *topology*, i.e. of the elements shapes and neighbors. The topology changes are essential to preserve a good quality mesh in presence of high shear flow or vortical flows for very long computational times, but also imply the appearance of degenerate space-time control volumes that must be treated with special care.

In particular, in Section 3 we will propose a new and effective *local* technique to move the nodes at the interfaces

between regions with different characteristics in a *nonconforming* way. This approach will be presented in the case of a second order FV scheme, for the sake of simplicity, reproducibility and to facilitate the coupling with the well balancing. Indeed in Section 6 we will couple our nonconforming ALE algorithm with some new well balanced path-conservative schemes, i.e. methods appropriate to treat non-conservative products and able to maintain up to machine precision families of equilibria of the studied system (see Section 4). Since our nonconforming ALE scheme is especially well suited for modeling in polar coordinates vortical flows affected by strong differential rotation, and thanks to the novel combination with the well balancing, in Section 6.2 we will be able to show great results for challenging astronomical phenomena as the rotating Keplerian disk.

Then, in Section 7 we will describe a global technique that starting from a set of generator points rapidly rebuilds a new Voronoi tessellation at each time step, connects different grids in space–time and extends the *high* order scheme presented in Section 2 even to *sliver* degenerate elements.

Each Section is closed by a large set of numerical results proving the robustness and the accuracy of the proposed approach and the enhancements with respect to literature. The paper is closed by some conclusive remarks and an outlook to future work in Section 8.

2 Arbitrary high order direct ALE FV-DG schemes

2.1 System of equations

We consider here a very general formulation of the governing equations in order to model a wide class of physical phenomena, namely all the ones which are governed by equations that can be cast into the following form,

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) + \mathbf{B}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}), \mathbf{x} \in \Omega(t) \subset \mathbb{R}^d, t \in \mathbb{R}_0^+. \quad (1)$$

In this system, \mathbf{x} is the spatial position vector, $d = [1, 2, 3]$ denotes the number of space dimensions, t represents the time, and $\Omega(t)$ is the computational domain at time t . $\mathbf{Q}(\mathbf{x}, t) = (q_1(\mathbf{x}, t), q_2(\mathbf{x}, t), \dots, q_v(\mathbf{x}, t))$ is the vector of the conserved variables defined in the space of the admissible states $\Omega_{\mathbf{Q}} \subset \mathbb{R}^v$, $\mathbf{F}(\mathbf{Q}) = (\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q}), \mathbf{h}(\mathbf{Q}))$ is the nonlinear flux tensor, $\mathbf{B}(\mathbf{Q}) = (\mathbf{B}_1(\mathbf{Q}), \mathbf{B}_2(\mathbf{Q}), \mathbf{B}_3(\mathbf{Q}))$ is a matrix collecting the non-conservative terms, and $\mathbf{S}(\mathbf{Q})$ represents a nonlinear algebraic source term. The system (1) can also be written in the following quasi-linear form

$$\frac{\partial \mathbf{Q}}{\partial t} + \mathbf{A}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \mathbf{S}(\mathbf{Q}), \quad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^d, t \in \mathbb{R}_0^+, \quad (2)$$

with the system matrix $\mathbf{A}(\mathbf{Q}) = \partial \mathbf{F} / \partial \mathbf{Q} + \mathbf{B}(\mathbf{Q})$. The system is *hyperbolic* if for any normal direction $\mathbf{n} \neq \mathbf{0}$ the matrix

$\mathbf{A}(\mathbf{Q}) \cdot \mathbf{n}$ has v real eigenvalues and a full set of v linearly independent eigenvectors for all $\mathbf{Q} \in \Omega_{\mathbf{Q}}$. PDE systems like (1) include as particular cases systems of *conservation laws* ($\mathbf{B} = \mathbf{0}$, $\mathbf{S} = \mathbf{0}$), systems of conservation laws with source terms or *balance laws* ($\mathbf{B} = \mathbf{0}$), and even *non-conservative* hyperbolic systems ($\mathbf{B} \neq \mathbf{0}$).

They appear in many different physical models: in particular, in this work we will take into account the shallow water equations, the Euler equations of gas dynamics with and without gravity, and the magnetohydrodynamics equations.

2.2 Domain discretization

To discretize the moving two-dimensional domain $\Omega(t)$ we employ an unstructured mesh made of N_p non overlapping polygons $P_i, i = 1, \dots, N_p$. The tessellation is firstly built at time $t = 0$ and then it is rearranged at each time step t^n : elements and nodes are moved following the local fluid velocity and when necessary, in order to prevent mesh distortions, also the mesh topology is changed according to the procedures described in Sections 3 or 7.

Given a polygon P_i^n we denote by $\mathcal{V}(P_i^n) = \{v_{i_1}^n, \dots, v_{i_j}^n, \dots, v_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ Voronoi neighbors (the neighbors that share with P_i^n a vertex), by $\mathcal{E}(P_i^n) = \{e_{i_1}^n, \dots, e_{i_j}^n, \dots, e_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ edges, and by $\mathcal{D}(P_i^n) = \{d_{i_1}^n, \dots, d_{i_j}^n, \dots, d_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ vertexes, consistently *ordered counterclockwise*. Finally, the barycenter of P_i^n is noted as $\mathbf{x}_{b_i}^n = (x_{b_i}^n, y_{b_i}^n)$. When necessary, by connecting $\mathbf{x}_{b_i}^n$ with each vertex of $\mathcal{D}(P_i^n)$ we can subdivide a polygon P_i^n in $N_{V_i}^n$ subtriangles denoted as $\mathcal{T}(P_i^n) = \{T_{i_1}^n, \dots, T_{i_j}^n, \dots, T_{i_{N_{V_i}^n}}^n\}$.

To lighten the notation, when there is no confusion, with an index i we refer to an element P_i^n , with an index j we refer to one of its neighbors $P_j^n \in \mathcal{V}(P_i^n)$, and with an index k we refer to a node. The coordinate of each node at time t^n are denoted by \mathbf{x}_k^n , and $\bar{\mathbf{V}}_k^n$ represents the velocity at which it is supposed to move, so that its new coordinates at time t^{n+1} are given from the following relation

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \bar{\mathbf{V}}_k^n, \quad (3)$$

More details on how to obtain $\bar{\mathbf{V}}_k^n$ will be given in Section 3.2 for the nonconforming case and in 7.2 for the regenerating Voronoi case.

2.3 Spatial representation of the numerical solution

Data are represented via high order polynomials in each polygon, which are either given by a MUSCL or (C)WENO reconstruction of the cell average values for FV schemes, or

directly available from the numerical solution when a DG method is considered.

Indeed, the conserved quantities \mathbf{Q} in (1) are represented via a cell-centered approach inside each polygon P_i^n at the current time t^n by piecewise polynomials of degree $N \geq 0$ denoted by $\mathbf{u}_h^n(\mathbf{x}, t^n)$ and defined in the space \mathcal{U}_h ,

$$\mathbf{u}_h^n(\mathbf{x}, t^n) = \sum_{\ell=0}^{\mathcal{N}-1} \varphi_\ell(\mathbf{x}, t^n) \hat{\mathbf{u}}_{\ell,i}^n := \varphi_\ell(\mathbf{x}, t^n) \hat{\mathbf{u}}_{\ell,i}^n, \quad \mathbf{x} \in P_i^n, \quad (4)$$

where $\varphi_\ell(\mathbf{x}, t^n)$ are *modal* spatial basis functions used to span the space of polynomials \mathcal{U}_h up to degree N . In the rest of the paper we will use classical tensor index notation based on the Einstein summation convention, which implies summation over two equal indices. The total number \mathcal{N} of expansion coefficients (degrees of freedom) $\hat{\mathbf{u}}_i^n$ for the basis functions depends on the polynomial degree N and is given by $\mathcal{N} = \mathcal{L}(N, d)$, with

$$\mathcal{L}(N, d) = \frac{1}{d!} \prod_{m=1}^d (N+m), \quad (5)$$

where $d = 2$ for the two-dimensional case. As basis functions φ_ℓ in (4) we employ a Taylor series of degree N in the variables $\mathbf{x} = (x, y)$ directly defined on the *physical element* P_i^n , expanded about its current barycenter $\mathbf{x}_{b_i}^n$ and normalized by its current characteristic length h_i

$$\varphi_\ell(\mathbf{x}, t^n)|_{P_i^n} = \frac{(x - x_{b_i}^n)^{p_\ell} (y - y_{b_i}^n)^{q_\ell}}{p_\ell! h_i^{p_\ell} q_\ell! h_i^{q_\ell}}, \quad (6)$$

$$\ell = 0, \dots, \mathcal{N} - 1, \quad 0 \leq p_\ell + q_\ell \leq N,$$

h_i being the radius of the circumcircle of P_i^n . The unknown expansion coefficients $\hat{\mathbf{u}}_{\ell,i}^n$ in (4) are the rescaled derivatives $h_i^{p_\ell} h_i^{q_\ell} \frac{\partial^{p_\ell+q_\ell}}{\partial x^{p_\ell} \partial y^{q_\ell}} \mathbf{Q}(\mathbf{x}_{b_i}^n)$ of the Taylor expansion about $\mathbf{x}_{b_i}^n$.

The discontinuous finite element data representation (4) leads naturally to both a Discontinuous Galerkin (DG) scheme if $N > 0$, but also to a Finite Volume (FV) scheme in the case $N = 0$. This indeed means that for $N = 0$ we have $\varphi_\ell(\mathbf{x}) = 1$, with $\ell = 0$ and (4) reduces to the classical piecewise constant data representation that is typical of finite volume schemes:

$$\begin{aligned} \mathbf{u}_h^n(\mathbf{x}, t^n) &= 1 \cdot \hat{\mathbf{u}}_{0,i}^n = \hat{\mathbf{u}}_{0,i}^n, \quad \mathbf{x} \in P_i^n, \\ \hat{\mathbf{u}}_{0,i}^n &= \frac{1}{|P_i^n|} \int_{P_i^n} \mathbf{Q}(\mathbf{x}, t^n) d\mathbf{x}. \end{aligned} \quad (7)$$

Here, the only degree of freedom per element P_i^n is the usual cell average $\hat{\mathbf{u}}_{0,i}^n$. Note also that in the case $N > 0$ the representation given by (4) already provides a spatially high order accurate data representation with accuracy $N + 1$, which is not the case when $N = 0$. If we are interested in increasing the spatial order of accuracy of a finite volume scheme, up to $M + 1$ for example, we need to perform a spatial *reconstruction* that generates a spatially high order accurate

reconstruction polynomial $\mathbf{w}_h^n(\mathbf{x}, t^n)$ of degree $M > N$ (see the MUSCL procedure described in 3.4 or the CWENO procedure presented in 7.1) that reads

$$\mathbf{w}_h^n(\mathbf{x}, t^n) = \sum_{\ell=0}^{\mathcal{M}-1} \psi_\ell(\mathbf{x}, t^n) \hat{\mathbf{w}}_{\ell,i}^n := \psi_\ell(\mathbf{x}, t^n) \hat{\mathbf{w}}_{\ell,i}^n, \quad (8)$$

$$\mathbf{x} \in P_i^n, \quad \mathcal{M} = \mathcal{L}(M, d),$$

where we simply employ the same basis functions $\psi_\ell(\mathbf{x}, t^n) = \varphi_\ell(\mathbf{x}, t^n)$ for the reconstruction according to (6), but with $0 \leq \ell \leq \mathcal{M} - 1$ rather than $0 \leq \ell \leq \mathcal{N} - 1$, see also [58].

With this notation, our method falls within the more general class of $P_N P_M$ schemes introduced in [58] for fixed unstructured simplex meshes in two and three space dimensions. In this paper, in Section 3 and 4 we will consider the simple case of $M = 0$ and $M = 1$, i.e. a FV scheme of order one or two; instead in Section 7 we consider the two main situations: (i) $N = 0$, with arbitrary high order reconstruction of degree $M > N$, which indeed corresponds to a FV scheme of order $M + 1$, and (ii) $N = M$, which corresponds to a DG scheme of accuracy $N + 1$.

For the sake of uniform notation, in the DG case, i.e. when $N > 0$ and $M = N$, we trivially impose that the reconstruction polynomial is given by the DG polynomial, i.e. $\mathbf{w}_h^n(\mathbf{x}, t^n) = \mathbf{u}_h^n(\mathbf{x}, t^n)$, which automatically implies that in the case $N = M$ the reconstruction operator is simply the identity.

2.4 Space–time control volumes - general framework

As already said in the Introduction, the family of direct ALE schemes proposed in this work are based on integrating the governing equation directly over *space–time* control volumes; they are constructed as follow.

For each element P_i^n the new vertex coordinates \mathbf{x}_k^{n+1} , $k = 1, \dots, N_{V_i}^n$, are connected to the old coordinates \mathbf{x}_k^{n+1} via straight line segments, yielding the multidimensional space–time control volume C_i^n , that involves overall $N_{V_i}^{n,st} + 2$ space–time sub-surfaces. Specifically, the space–time volume C_i^n is bounded on the bottom and on the top by the element configuration at the current time level P_i^n and at the new time level P_i^{n+1} , respectively, while it is closed with a total number of $N_{V_i}^{n,st}$ lateral space–time surfaces $\partial C_{i,j}^n$, $j = 1, \dots, N_{V_i}^{n,st}$ that are given by the evolution of each edge $e_{i,j}^n$ of element P_i^n within the time step $\Delta t = t^{n+1} - t^n$. Therefore the space–time volume C_i^n is bounded by its surface ∂C_i^n which is given by

$$\partial C_i^n = \left(\bigcup_j \partial C_{i,j}^n \right) \cup P_i^n \cup P_i^{n+1}. \quad (9)$$

For a graphical interpretation one can refer to Figure 1, where we have reported an example of a control volume and of the parametrization of the lateral space–time surfaces.

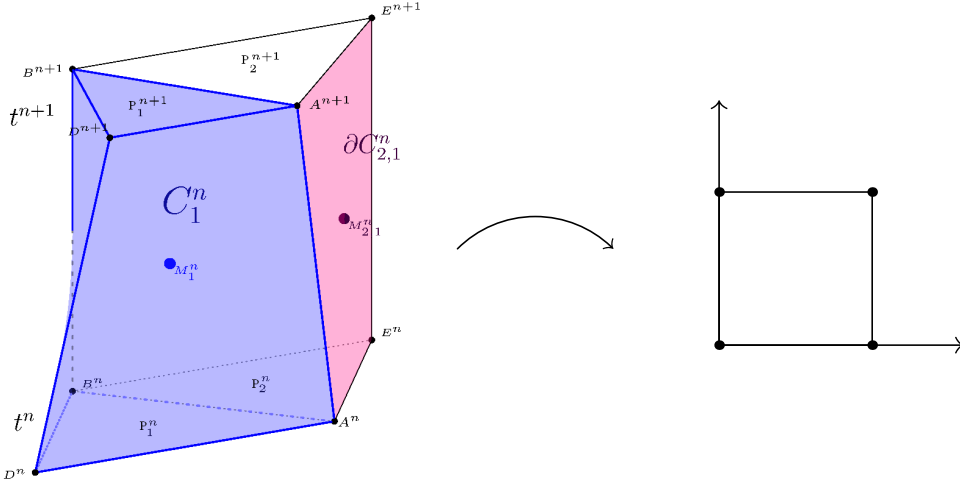


Fig. 1: Left. In blue we show the physical space–time control volume C_1^n obtained by connecting via straight line segments each vertex of P_1^n with the corresponding vertex of P_1^{n+1} , and its space–time midpoint M_1^n . In pink we show one of the lateral surfaces of C_2^n , $\partial C_{2,1}^n$, together with its space–time midpoint $M_{2,1}^n$. Right. The reference system (χ, τ) adopted for the bilinear parametrization of the lateral surfaces ∂C_{ij}^n .

2.5 Fully-discrete predictor-corrector ADER approach

Now we can solve (1) using a high order fully-discrete *one-step predictor-corrector* ADER FV-DG method obtained by generalizing the scheme first presented in [58] to our moving geometry. ADER finite were firstly introduced by Toro and Titarev in [147, 150, 148, 139, 151] and have been successfully developed in the fixed grids framework in [61, 27]; subsequently they have been extended to moving meshes in the ALE context [12, 15, 10, 16].

The *predictor* step consists in a *local* solution of the governing PDE (1) *in the small*, see [89], inside each space–time element C_i^n . We underline that the predictor step is the key for obtaining the high order of accuracy in time in a *one single step* procedure and that this is achieved acting at a completely *local* level: indeed it is obtained by only considering cell C_i^n with initial data \mathbf{w}_h^n on P_i^n , the governing equations (1) and the geometry of C_i^n , without taking into account any interaction between C_i^n and its neighbors. It provides, for each space–time control volume C_i^n , a polynomial data representation \mathbf{q}_h^n (see below for the details) of high order both in space and time, which serves as a predictor solution, only valid inside C_i^n , to be used for evaluating the numerical fluxes and sources when integrating the PDE in the final corrector step.

Then, the *corrector* step integrates the weak form of the PDE over the space–time control volumes C_i^n , making use of the predictor solution \mathbf{q}_h^n , and returns \mathbf{u}_h^{n+1} by taking care of the coupling with neighbors through the numerical flux computations across ∂C_i^n . It ensures high order of accuracy

in space and time, provided the high order of accuracy of \mathbf{q}_h^n . The scheme is by construction conservative since it takes into account all the flux contributions over ∂C_i^n .

Finally, the time step size Δt is given by

$$\Delta t < \text{CFL} \left(\frac{|P_i^n|}{(2N+1) |\lambda_{\max,i}| \sum_{\partial P_{ij}^n} |\ell_{ij}|} \right), \quad \forall P_i^n \in \Omega^n, \quad (10)$$

where, ℓ_{ij} is the length of the edge j of P_i^n and $|\lambda_{\max,i}|$ is the spectral radius of the Jacobian of the flux \mathbf{F} . Stability on unstructured meshes is guaranteed by the satisfaction of the inequality $\text{CFL} < \frac{1}{d}$, see [58].

2.6 High order space–time predictor

Our ADER space–time predictor is given by a high order piecewise space–time polynomial $\mathbf{q}_h^n(\mathbf{x}, t)$ of degree M of the form

$$\mathbf{q}_h^n(\mathbf{x}, t) = \sum_{\ell=0}^{\mathcal{Q}-1} \theta_\ell(\mathbf{x}, t) \hat{\mathbf{q}}_\ell^n, \quad (\mathbf{x}, t) \in C_i^n, \quad \mathcal{Q} = \mathcal{L}(M, d+1). \quad (11)$$

with $\theta_\ell(\mathbf{x}, t)$ being a *modal space–time* basis of the polynomials of degree M in $d+1$ dimensions (d space dimensions plus time), which read

$$\theta_\ell(x, y, t)|_{C_i^n} = \frac{(x - x_{b_i}^n)^{p_\ell}}{p_\ell! h_i^{p_\ell}} \frac{(y - y_{b_i}^n)^{q_\ell}}{q_\ell! h_i^{q_\ell}} \frac{(t - t^n)^{r_\ell}}{q_\ell! h_i^{r_\ell}}, \quad (12)$$

$$\ell = 0, \dots, \mathcal{L}(M, d+1), \quad 0 \leq p_\ell + q_\ell + r_\ell \leq M.$$

It will serve as a predictor of the solution valid *locally* inside C_i^n .

It is computed through an iterative procedure that looks for the polynomial satisfying a weak form of (1) obtained for any control volume C_i^n as follows. We multiply the governing PDE (1), evaluated on \mathbf{q}_h^n , by a test function θ_k and we integrate over C_i^n , hence

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} d\mathbf{x} dt + \int_{C_i^n} \theta_k(\mathbf{x}, t) \nabla \cdot \mathbf{F}(\mathbf{q}_h^n) d\mathbf{x} dt = \int_{C_i^n} \theta_k(\mathbf{x}, t) \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x} dt. \quad (13)$$

Differently from what has been proposed in [58, 61, 14, 15], here we do *not* integrate the first term in (13) by parts in time. Instead, we take into account potential jumps of \mathbf{q}_h on the boundaries of C_i^n in the sense of distributions, combined with *upwinding* of the fluxes in time. This approach is similar to the path-conservative schemes proposed in [131, 33, 31], but much simpler, since the test functions are only taken from within C_i^n and there is no need to define a non-conservative product on ∂C_i^n . Therefore, the integral containing the time derivative in (13) is rewritten as

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} d\mathbf{x} dt = \int_{C_i^n \setminus \partial C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} d\mathbf{x} dt + \int_{\partial C_i^n} \theta_k(\mathbf{x}, t) \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right) \tilde{\mathbf{n}}_t^- dS. \quad (14)$$

Here, $\mathbf{q}_h^{n,-}$ and $\mathbf{q}_h^{n,+}$ denote the boundary-extrapolated inner and outer states across the jump on ∂C_i^n . Furthermore, $\tilde{\mathbf{n}}^-$ are only those outward pointing unit-normal vectors on ∂C_i^n that point *back* in time and $\tilde{\mathbf{n}}_t^-$ is their time component, i.e. $\tilde{\mathbf{n}}_t^- = \min(0, \tilde{\mathbf{n}} \cdot (0, 0, 1)) \leq 0$. Upwinding in time is therefore automatically guaranteed, since we only consider the contributions coming from the *past*, according to the causality principle. In other words, only time fluxes that *enter* the space–time control volume C_i^n contribute to the jump term in (14), and they are easily identified by checking the sign of the time component of the space–time normal vector $\tilde{\mathbf{n}}$.

In the classical case of control volume with the shape of an oblique prism, we apply the jump term only on the bottom surface P_i^n of the space–time element C_i^n under consideration: in this manner all space–time predictors are *decoupled* from each other, since they only require the initial data \mathbf{w}_h^n and no information from the neighbor elements and the procedure that gives us high order in time results to be completely local.

In particular, the jump term simplifies

$$\left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right) \tilde{\mathbf{n}}_t^- \Big|_{P_i^n} = - \left(\mathbf{w}_h^n(\mathbf{x}, t^n) - \mathbf{q}_h^n(\mathbf{x}, t^n) \right) = \mathbf{q}_h^n(\mathbf{x}, t^n) - \mathbf{w}_h^n(\mathbf{x}, t^n), \quad (15)$$

with $\mathbf{q}_h^{n,+} = \mathbf{w}_h(\mathbf{x}, t^n)$ being simply given by the reconstruction polynomial at time t^n , $\tilde{\mathbf{n}}^- = (0, 0, -1)$ on P_i^n and thus $\tilde{\mathbf{n}}_t^- = -1$. In this case, (14) reduces to

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} d\mathbf{x} dt = \int_{C_i^n \setminus P_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} d\mathbf{x} dt + \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \left(\mathbf{q}_h^n(\mathbf{x}, t^n) - \mathbf{w}_h(\mathbf{x}, t^n) \right) d\mathbf{x}. \quad (16)$$

Finally, with the following definitions

$$\begin{aligned} \mathbf{K}_1 &= \int_{C_i^n \setminus P_i^n} \theta_k \frac{\partial \theta_\ell}{\partial t} d\mathbf{x} dt, & \mathbf{K}_x &= \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial x} d\mathbf{x} dt, \\ \mathbf{K}_y &= \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial y} d\mathbf{x} dt, & \mathbf{M} &= \int_{C_i^n} \theta_k \theta_\ell d\mathbf{x} dt, \\ \mathbf{F}_0 &= \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \psi_\ell(\mathbf{x}, t^n) d\mathbf{x}, & \mathbf{F}_1 &= \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \theta_\ell(\mathbf{x}, t^n) d\mathbf{x}, \end{aligned} \quad (17)$$

the weak form (13)-(14) can be compactly rewritten as

$$(\mathbf{K}_1 + \mathbf{F}_1) \hat{\mathbf{q}}_i^n = \mathbf{F}_0 \hat{\mathbf{w}}_i^n - \mathbf{K}_x \mathbf{f}(\hat{\mathbf{q}}_i^n) - \mathbf{K}_y \mathbf{g}(\hat{\mathbf{q}}_i^n) + \mathbf{M} \mathbf{S}(\hat{\mathbf{q}}_i^n), \quad (18)$$

where $\hat{\mathbf{q}}_i^n$ and $\hat{\mathbf{w}}_i^n$ contain all the expansion coefficients of $\hat{\mathbf{q}}_{\ell,i}^n$ in (11) and $\hat{\mathbf{w}}_{\ell,i}^n$ in (8), respectively. The solution of (18) can be found via a simple and fast converging fixed point iteration (a discrete Picard iteration), as detailed in [58, 90]. Here, as initial guess we simply impose $\hat{\mathbf{q}}_{\ell,i}^n = \hat{\mathbf{w}}_{\ell,i}^n$ for the common spatial degrees of freedom (with $\ell \leq \mathcal{M}$) and zero for the other ones. For linear homogeneous systems, the discrete Picard iteration converges in a finite number of at most $M + 1$ steps, since the involved iteration matrix is nilpotent, see [95]. In the nonlinear case we allow a maximum of 10 iterations if convergence is not reached before, being $M + 1$ iterations enough for obtaining the correct order M of convergence.

The integrals above are evaluated using multidimensional Gaussian quadrature rules of suitable order of accuracy, see [145] and Figure 2 for details. In order to carry out the integration, we split the space–time volume C_i^n into a set of sub–space–time volume $sC_{i_j}^n$ of C_i^n , whose shape is an oblique triangular prism.

2.7 Corrector step: flux computation and update of the solution

This Section contains the core of our direct ALE FV-DG scheme used to solve (1) on regenerating moving meshes.

Following [14, 15, 16], the PDE system (1) is rewritten in a space-time divergence form as

$$\tilde{\mathbf{V}} \cdot \tilde{\mathbf{F}} = \mathbf{S}, \quad (19)$$

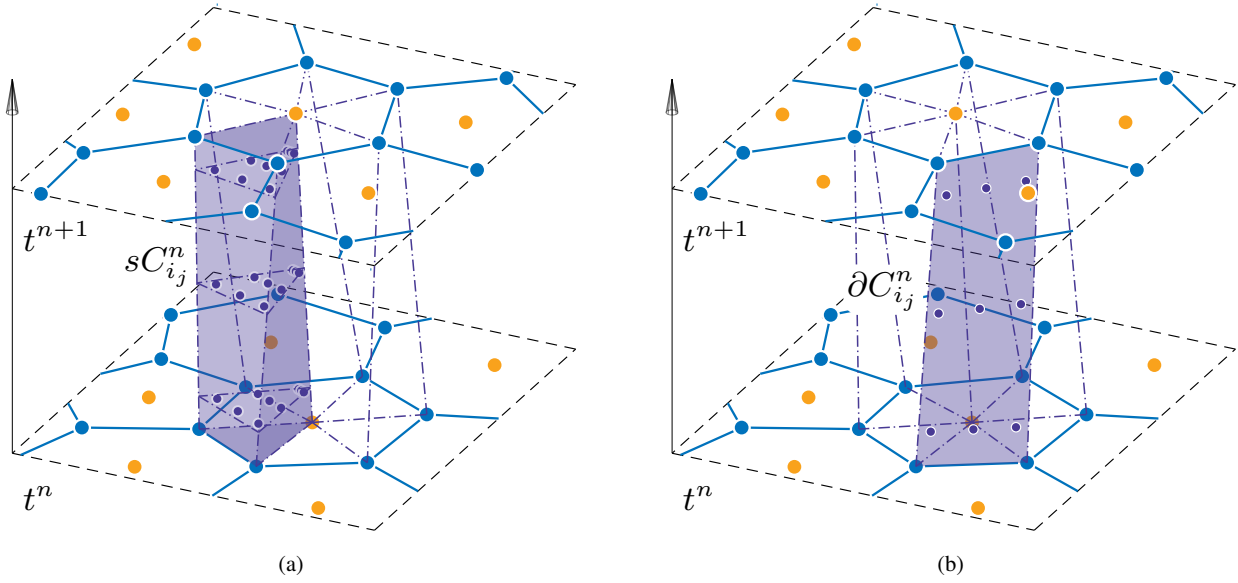


Fig. 2: Space–time quadrature points for third order methods, i.e. $M = 2$. (a) Quadrature points for the volume integrals and the space–time predictor. (b) Quadrature points for the surface integrals, i.e. for flux computation.

with $\tilde{\nabla} = (\partial_x, \partial_y, \partial_t)$ denoting the space-time divergence operator and $\tilde{\mathbf{F}} = (\mathbf{f}, \mathbf{g}, \mathbf{Q})$ being the corresponding space-time flux tensor. Then, we multiply (19) by a set of *moving* spatial modal test functions $\tilde{\varphi}_k(\mathbf{x}, t)$, which coincide with (6) at $t = t^n$ and at $t = t^{n+1}$, i.e. $\tilde{\varphi}_k(\mathbf{x}, t^n) = \varphi_k(\mathbf{x}, t^n)$ and $\tilde{\varphi}_k(\mathbf{x}, t^{n+1}) = \varphi_k(\mathbf{x}, t^{n+1})$. The test functions are tied to the motion of the barycenter $\mathbf{x}_{b_i}(t)$ and move together with $P_i(t)$ in such a way that at time $t = t^{n+1}$ they refer to the new barycenter $\mathbf{x}_{b_i}^{n+1}$. Thus, the test functions explicitly read as follows:

$$\tilde{\varphi}_\ell(x, y, t)|_{C_i^n} = \frac{(x - x_{b_i}(t))^{p_\ell} (y - y_{b_i}(t))^{q_\ell}}{p_\ell! h_i^{p_\ell} q_\ell! h_i^{q_\ell}}, \quad (20)$$

$$\text{with } \mathbf{x}_{b_i}(t) = \frac{t - t^n}{\Delta t} \mathbf{x}_{b_i}^n + \left(1 - \frac{t - t^n}{\Delta t}\right) \mathbf{x}_{b_i}^{n+1}, \quad (21)$$

$$\ell = 0, \dots, \mathcal{N}, \quad 0 \leq p + q \leq N.$$

These *moving modal basis functions* are essential for the approach presented in this paper. They *naturally* allow for topology changes, without the need of any remapping steps, which we want to avoid in a direct ALE formulation.

Next, integration over the closed space-time control volume C_i^n yields

$$\int_{C_i^n} \tilde{\varphi}_k \tilde{\nabla} \cdot \tilde{\mathbf{F}}(\mathbf{Q}) \, d\mathbf{x}dt = \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{Q}) \, d\mathbf{x}dt. \quad (22)$$

Application of the Gauss theorem leads to the following weak form that is the basis of our fully-discrete ALE scheme

$$\int_{\partial C_i^n} \tilde{\varphi}_k \tilde{\mathbf{F}}(\mathbf{Q}) \cdot \tilde{\mathbf{n}} \, dS - \int_{C_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{Q}) \, d\mathbf{x}dt = \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{Q}) \, d\mathbf{x}dt, \quad (23)$$

where $\tilde{\mathbf{n}} = (\tilde{n}_x, \tilde{n}_y, \tilde{n}_t)$ denotes the outward pointing space-time unit normal vector on the space-time faces composing the boundary ∂C_i^n of the space-time control volume. Moreover, the surface integral can be decomposed over the faces of ∂C_i^n given by (9).

After introducing the discrete solution \mathbf{u}_h , the space-time predictor \mathbf{q}_h and a two-point numerical flux function on the element boundaries of the type

$$\tilde{\mathbf{F}}(\mathbf{Q}) \cdot \tilde{\mathbf{n}} := \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}}, \quad (24)$$

into (23), where $\mathbf{q}_h^{n,-}$ and $\mathbf{q}_h^{n,+}$ are the inner and outer boundary-extrapolated data respectively, (i.e. the values assumed by the predictors of two elements at a point on the shared space-time lateral surface), we obtain the final direct ALE scheme:

$$\begin{aligned} \int_{P_i^{n+1}} \tilde{\varphi}_k \mathbf{u}_h(\mathbf{x}, t^{n+1}) \, d\mathbf{x} &= \int_{P_i^n} \tilde{\varphi}_k \mathbf{u}_h(\mathbf{x}, t^n) \, d\mathbf{x} \\ &- \sum_{j=1}^{N_{V_i}^{n,st}} \int_{\partial C_{ij}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS \\ &+ \int_{C_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h) \, d\mathbf{x}dt + \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h) \, d\mathbf{x}dt, \end{aligned} \quad (25)$$

where the unknown solution at the new time step $\mathbf{u}_h(\mathbf{x}, t^{n+1})$ can be computed *directly* from the solution at the previous time step $\mathbf{u}_h(\mathbf{x}, t^n)$ through the integration of the fluxes and source terms over C_i^n , without needing any further remapping/remeshing steps.

The boundary fluxes are obtained by a Riemann solver, thus providing the coupling between neighbors, which was

neglected in the predictor step. The ALE Jacobian matrix w.r.t. the normal direction in space reads

$$\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{Q}) = \left(\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2} \right) \left[\frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right], \quad \mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \quad (26)$$

with \mathbf{I} representing the identity matrix and $\mathbf{V} \cdot \mathbf{n}$ denoting the local normal mesh velocity. Furthermore, \mathbf{n} is the spatial normalized normal vector, which is different from the space-time normal vector $\tilde{\mathbf{n}}$. We adopt either a simple and robust Rusanov-type [137] ALE scheme,

$$\mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} = \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}_h^{n,+}) + \tilde{\mathbf{F}}(\mathbf{q}_h^{n,-}) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} s_{\max} \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \quad (27)$$

where s_{\max} is the maximum eigenvalue of $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{q}_h^{n,+})$ and $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\mathbf{q}_h^{n,-})$, or a less dissipative Osher-type [128,67] ALE flux

$$\mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} = \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}_h^{n,+}) + \tilde{\mathbf{F}}(\mathbf{q}_h^{n,-}) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} \left(\int_0^1 |\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}(\Psi(s))| ds \right) \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \quad (28)$$

where we choose to connect the left and the right state across the discontinuity using a simple straight-line segment path

$$\Psi(s) = \mathbf{q}_h^{n,-} + s \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \quad 0 \leq s \leq 1. \quad (29)$$

The absolute value of $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}$ is evaluated as usual as $\mathbf{R}|\Lambda|\mathbf{R}^{-1}$, where \mathbf{R} , \mathbf{R}^{-1} and Λ denote, respectively, the right eigenvector matrix, its inverse and the eigenvalues matrix of $\mathbf{A}_{\mathbf{n}}^{\mathbf{V}}$.

Finally, using the definitions (4) and (8), our arbitrary high order one-step direct ALE FV-DG scheme becomes

$$\begin{aligned} \left(\int_{P_i^{n+1}} \tilde{\phi}_k \phi_\ell d\mathbf{x} \right) \hat{\mathbf{u}}_\ell^{n+1} &= \left(\int_{P_i^n} \tilde{\phi}_k \psi_\ell d\mathbf{x} \right) \hat{\mathbf{w}}_\ell^n \\ &- \sum_{j=1}^{N_{V_i}^{st}} \int_{\partial C_{ij}^n} \tilde{\phi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} dS \\ &+ \int_{C_i^n} \tilde{\nabla} \tilde{\phi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) d\mathbf{x} dt + \int_{C_i^n} \tilde{\phi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x} dt. \end{aligned} \quad (30)$$

The volume integrals in the above expression (30) can be easily computed directly on the physical space-time element C_i^n by summing up the contributions on each sub-volume sC_{ij}^n and employing Gaussian quadrature rules of sufficient precision, see [145]. The lateral space-time surfaces of ∂C_{ij}^n instead are parameterized using a set of bilinear basis functions [14], that is

$$\partial C_{ij}^n = \tilde{\mathbf{x}}(\chi, \tau) = \sum_{k=1}^4 \beta_k(\chi, \tau) \tilde{\mathbf{X}}_{ij,k}^n, \quad (31)$$

$$0 \leq \chi \leq 1, \quad 0 \leq \tau \leq 1,$$

where the $\tilde{\mathbf{X}}_{ij,k}^n$ represent the physical space-time coordinates of the four vertexes of ∂C_{ij}^n , and the functions $\beta_k(\chi, \tau)$ are defined as follows

$$\begin{aligned} \beta_1(\chi, \tau) &= (1-\chi)(1-\tau), & \beta_2(\chi, \tau) &= \chi(1-\tau), \\ \beta_3(\chi, \tau) &= \chi\tau, & \beta_4(\chi, \tau) &= (1-\chi)\tau. \end{aligned} \quad (32)$$

The mapping in time is given by the transformation

$$t = t_n + \tau \Delta t, \quad \tau = \frac{t - t^n}{\Delta t}. \quad (33)$$

In this way, every ∂C_{ij}^n can be mapped to a reference square $[0, 1] \times [0, 1]$ and surface integrals can be computed.

2.7.1 Properties

We want to emphasize that the integration over a closed space-time control volume, as done above, automatically satisfies the so-called geometric conservation law (GCL), for all test functions $\tilde{\phi}_k$. This simply follows from Gauss theorem

$$\int_{\partial \mathcal{C}_i^n} \tilde{\mathbf{n}} dS = 0, \quad (34)$$

applied to *closed* space-time control volumes. The relation between (34) and the usual form of the *GCL* that is typically employed in the community working on Lagrangian schemes has been established in the appendix of [15]. The satisfaction of the GCL property up to machine precision has been numerically verified in *each* simulation presented in this paper and a series of test cases aiming in demonstrating its validity is presented in Section 7.6.2.

Moreover the scheme is locally and globally *conservative* for mass and inertial momentum.

Finally, we underline that the direct ALE scheme presented here does in general *not* lead to a vanishing mass flux across element boundaries, similar to previous work on direct ALE schemes presented in [14, 15]. The mass flux is exactly zero only for isolated contact discontinuities moving in uniform flow when using appropriate Riemann solvers that resolve contact waves, like the Godunov method, or the Roe, HLLC, HLLEM and Osher flux.

2.8 A posteriori sub-cell finite volume limiter

The direct ALE scheme presented so far is arbitrarily high order accurate in space and time and incorporates the FV case, if $N = 0$ and if a high order spatial reconstruction is adopted, and the DG case, when $N = M$. But actually there is one major difference between FV and DG: indeed the reconstruction operators (CWENO or MUSCL) provide a nonlinear stabilization of the FV scheme, while the DG scheme presented so far is unlimited and, as such, it is affected by

the so-called Gibbs phenomenon, i.e. oscillations are likely to appear in presence of shock waves or other discontinuities, which typically occur while solving nonlinear hyperbolic systems. These oscillations could be explained also by the Godunov theorem [83], because the presented high order DG scheme is linear in the sense of Godunov.

As a consequence, a limiting technique is required. Our strategy is based on the MOOD approach [40, 51, 52], which has already been successfully introduced in the framework of ADER finite volume schemes [111, 24, 22]. The idea behind this kind of schemes consists in checking the numerical solution *a posteriori* for nonphysical values and spurious oscillations: we consider as troubled elements those that do not pass a set of admissibility detection criteria, given by both physical and numerical requirements. If the solution in a cell is discarded, instead of applying a limiter to the already computed solution, the solution is *locally recomputed* relying on a first order finite volume method applied to a fine sub-grid generated within each troubled cell.

We refer in particular to [69, 156, 66, 17, 26] for an exhaustive description of the *a posteriori* finite volume subcell limiter. Here, for the sake of clarity, we briefly recall the main concepts and we underline the differences introduced for dealing with moving polygonal elements.

Firstly, using the notation adopted in [17], the numerical solution computed so far is assumed to be a *candidate* solution and denoted with $\mathbf{u}_h^{n+1,*}(\mathbf{x}, t^{n+1})$. Then, we define a sub-triangulation of P_i^n made of a set of non-overlapping so called *small sub-triangles*. Consequently, each control volume C_i^n is split into sub-triangular prisms, called *small sub-volumes*, as follows.

- For $N = 1$ we consider a total number of small sub-triangles \mathcal{S}_i which is equal to $N_{C_i}^n$, i.e. $\mathcal{S}_i = N_{C_i}^n$. The small sub-triangles are given by $T_{i_j}^n$ and the associated small sub-volumes are $sC_{i_j}^n$.
- For $N \geq 2$ we further subdivide each $T_{i_j}^n$ into N^2 small sub-triangles, which are defined through the sub-nodes provided by standard nodes of classical high order conforming finite elements on triangular meshes. In this way, a total number of $\mathcal{S}_i = N_{C_i}^n \cdot N^2$ small sub-triangles is taken into account. The splitting of $sC_{i_j}^n$ is consequently defined.

We denote each small sub-triangle of P_i^n with $s_{i,\alpha}^n$, where $\alpha \in [1, \mathcal{S}_i]$. Next, we define the corresponding subcell average of the numerical solution at time t^n

$$\begin{aligned} \mathbf{v}_{i,\alpha}^n(\mathbf{x}, t^n) &= \frac{1}{|s_{i,\alpha}^n|} \int_{s_{i,\alpha}^n} \mathbf{u}_h^n(\mathbf{x}, t^n) d\mathbf{x} \\ &= \frac{1}{|s_{i,\alpha}^n|} \int_{s_{i,\alpha}^n} \varphi_\ell(\mathbf{x}) d\mathbf{x} \hat{\mathbf{u}}_i^n := \mathcal{P}(\mathbf{u}_h^n) \quad \forall \alpha \in [1, \mathcal{S}_i], \end{aligned} \quad (35)$$

where $|s_{i,\alpha}^n|$ denotes the volume of subcell $s_{i,\alpha}^n$ of element P_i^n and the definition $\mathcal{P}(\mathbf{u}_h)$ is the L_2 projection operator. We fix also the *candidate* subcell average of the numerical solution at time t^{n+1} as $\mathbf{v}_{i,\alpha}^{n+1,*}(\mathbf{x}, t^{n+1}) = \mathcal{P}(\mathbf{u}_h^{n+1,*})$.

Now, we mark the troubled cells. The candidate solution $\mathbf{v}_h^{n+1,*}(\mathbf{x}, t^{n+1})$ is checked against a set of detection criteria. According to [17], the first criterion is the requirement that the computed solution is physically acceptable, i.e. belongs to the phase space of the conservation law being solved. For instance, if the compressible Euler equations for gas dynamics are considered, density and pressure should be positive and in practice we require that they are greater than a prescribed tolerance $\varepsilon = 10^{-12}$. Then, a relaxed discrete maximum principle (DMP) is applied, hence we verify

$$\begin{aligned} \min_{m \in \mathcal{V}(C_i^n)} \left(\min_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) - \delta &\leq \mathbf{v}_{i,\alpha}^{n+1,*} \\ &\leq \max_{m \in \mathcal{V}(C_i^n)} \left(\max_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) + \delta \quad \forall \alpha \in [1, \mathcal{S}_i], \end{aligned} \quad (36)$$

where δ is a parameter which, according to [17, 69, 156], reads

$$\delta = \max \left(\delta_0, \varepsilon \cdot \left[\max_{m \in \mathcal{V}(C_i^n)} \left(\max_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) - \min_{m \in \mathcal{V}(C_i^n)} \left(\min_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) \right] \right), \quad (37)$$

with $\delta_0 = 10^{-4}$ and $\varepsilon = 10^{-3}$.

If a cell fulfils the detection criteria in *all* its subcells, then the cell is marked as *good*, otherwise the cell is *troubled*. Note that, this step is performed independently in each element and thus the projection $\mathbf{v}_h^*(\mathbf{x}, t^{n+1})$ does not need to be retained after the cell is assigned its mark.

Then we need to recompute the solution *only* in the troubled cells with a first order FV scheme, applied in each small sub-triangle/sub-volume, that evolves the cell averages $\mathbf{v}_{i,\alpha}^n$ in order to obtain $\mathbf{v}_{i,\alpha}^{n+1}$.

We do not report the details on the first order ALE-FV scheme (assumed to be well-known), but we add some remarks on flux computation at the space–time lateral surfaces of each $s_{i,\alpha}^n$. i) The same numerical flux function, i.e. (27) or (28), used in the rest of the scheme is adopted here as well. ii) The employed quadrature rule is a simple midpoint rule that makes use of the space–time barycenters of the space–time lateral faces of the sub-volume. iii) The normal vectors are also computed at the space–time barycenters. iv) Referring to (24), when computing the flux between the sub-volume α of C_i^n and the neighboring sub-volume β (of C_i^n or of any other $C_{i_j}^n$), boundary data are simply given by $\mathbf{q}_h^{n,-} = \mathbf{v}_{i,\alpha}^n$ and $\mathbf{q}_h^{n,+} = \mathbf{v}_{i_j,\beta}^n$. v) If instead the neighbor is *not* troubled (which thus has not been sub-triangulated), then $\mathbf{q}_h^{n,-} = \mathbf{v}_{i,\alpha}^n$ and $\mathbf{q}_h^{n,+} = \mathbf{q}_h^n|_{C_{i_j}^n}(\mathbf{g}_{i_j}^n)$.

A first order finite volume scheme always provides a valid solution, hence $\mathbf{v}_{i,\alpha}^{n+1}$ is acceptable. Moreover, since the FV scheme is not directly applied to the element but to each of its sub-triangles, the sub-mesh resolution does not completely spoil the solution of the DG scheme. Nevertheless, the method does not maintain the formal order of accuracy of the $P_N P_M$ scheme, but it is only used and activated across shock waves and strong discontinuities. Note also that for a troubled cell the mesh motion is not recomputed because it has been fixed using only information coming from space at time t^n , which are, as such, not affected by any problem.

Finally, the DG polynomial for the cell P_i^{n+1} is recovered from the robust and stable solution on the sub-grid level $\mathbf{v}_{i,\alpha}^{n+1}$ by applying the reconstruction operator $\mathcal{R}(\mathbf{v}_{i,\alpha}^{n+1}(\mathbf{x}, t^n))$, that is

$$\int_{S_{i,\alpha}^n} \mathbf{u}_h^{n+1}(\mathbf{x}, t^{n+1}) d\mathbf{x} = \int_{S_{i,\alpha}^n} \mathbf{v}_{i,\alpha}^{n+1}(\mathbf{x}, t^n) d\mathbf{x} := \mathcal{R}(\mathbf{v}_{i,\alpha}^{n+1}(\mathbf{x}, t^n))$$

$$\forall \alpha \in [1, \mathcal{S}_i]. \quad (38)$$

The reconstruction is imposed to be *conservative* on the main cell P_i^n , hence yielding the additional linear constraint

$$\int_{P_i^n} \mathbf{u}_h(\mathbf{x}, t^{n+1}) d\mathbf{x} = \int_{P_i^n} \mathbf{v}_h(\mathbf{x}, t^{n+1}) d\mathbf{x}. \quad (39)$$

As a consequence, the projection operator \mathcal{P} in (35) and the reconstruction operator \mathcal{R} in (38) satisfy the property $\mathcal{P} \cdot \mathcal{R} = \mathcal{I}$, with \mathcal{I} being the identity operator.

If a cell C_i^n is good but has at least one bad neighbor cell $C_{i_j}^n$ in its $\mathcal{V}(C_i^n)$, we cannot accept its candidate solution $\mathbf{u}_h^{n+1,*}(\mathbf{x}, t^{n+1})$ because the scheme would become nonconservative. Indeed, at the common space–time lateral surface $\partial C_{i_j}^n$, the flux computed from C_i^n would be obtained through the DG scheme (i.e. high order predictor and high order corrector), while the one coming from the troubled neighbor $C_{i_j}^n$ would be updated using the first order FV scheme. Thus, the DG solution in these cells is recomputed in a *mixed way*: the volume integral and the surface integrals on good faces are kept, while the numerical flux across the troubled faces is always provided by the first order limiter.

The numerical method given in this Section provides an exhaustive description for our arbitrary high order accurate both in space and time *direct* ALE FV-DG scheme on *moving* meshes. The high order of accuracy in space is obtained as in any standard FV scheme through a reconstruction operator, and is available for free in the DG case; in both the cases it relies only on information coming from time t^n , thus independent of the mesh motion. The high order in time, provided by the predictor-corrector ADER paradigm, naturally fits into our moving framework, being already constructed on space–time control volumes. Finally, in order to

stabilize the DG scheme we have proposed a limiter, both robust because based on a FV strategy and accurate because applied on a sub-triangulation.

In the following, we will concentrate on new techniques to move the mesh in such a way to maintain its high quality for long times and we will extend the numerical scheme presented so far to degenerate space–time control volume.

3 Second order ALE FV scheme on nonconforming moving unstructured meshes

In this Section we propose a new and effective technique to move the nodes at the interfaces between regions with different characteristics, in order to avoid the typical mesh distortion of Lagrangian-type methods [80, 81].

In particular, in our new approach the interfaces could be both prescribed a priori by the user or automatically detected by the algorithm, if the tangential velocity difference across an element interface is sufficiently large. The new scheme then subdivides the neighbors of an interface edge into two groups, each of which has similar properties, namely the *left* and the *right* neighbors. The nodes far away from the shear discontinuity are moved with a standard node solver, while for those at the interface a *new node* is inserted in such a way that the old one moves according to an averaged velocity over the left neighbors, and the new one slides along the edge according to the average velocity prescribed by the right group.

We refer to the resulting mesh as *nonconforming* because each edge can be shared between *more* than two elements and a node can lie on an edge not only at its extremities, i.e. we explicitly allow so-called *hanging nodes*, see Figure 3 for an example of a nonconforming mesh that can be treated by our algorithm. This gives us more flexibility in the grid motion and helps to maintain a high quality mesh: indeed, the elements on both sides of the shear wave can move with a different velocity, without producing highly distorted elements. The core of the proposed method is the *local* update of all the necessary connectivity tables, as a consequence of the insertion or the deletion of nodes and edges, and the computation of the numerical fluxes between the space-time control volumes, taking in particular into account that the lateral faces can be shared between more than two elements.

In this case, we restrict ourselves to a FV scheme for which *second* order of accuracy in space and time is obtained through a MUSCL-Hancock strategy, together with a Barth and Jespersen slope limiter, which can be seen as an ADER predictor-corrector approach but is simpler to be reproduced. Moreover, in consider *conservative* systems of hyperbolic equations; the non-conservative case will be treated in Sections 4.

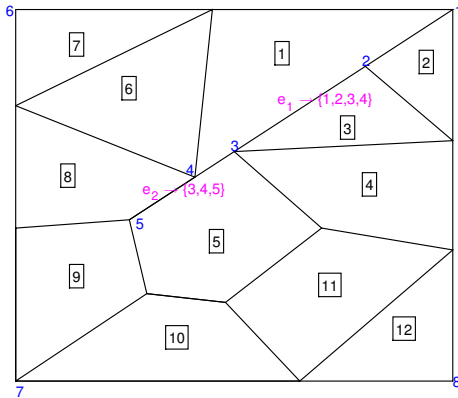


Fig. 3: Example of a nonconforming mesh that can be treated by our algorithm. The mesh contains $N_E = 12$ elements: triangles, quadrilaterals and five-sided polygons. The mesh is nonconforming: note for example edge e_1 . It is shared between the elements P_1, P_2, P_3, P_4 and on it we can find two intermediate nodes 2,3 called hanging nodes. A similar situation can be noted for edge e_2 .

Finally, in Section 3.6 some numerical test problems are presented in order to check the efficiency and the robustness of the proposed approach in maintaining a high quality mesh, local and global volume conservation, and in satisfying the GCL condition. The numerical results presented in this Section concern the shallow water equations both in Cartesian coordinates (see Section 3.5.1), and in polar coordinates (see Section 3.5.2). In particular we have compared the results for a steady vortex in equilibrium solved with a standard *conforming* ALE method (without any rezoning technique) and with our new *nonconforming* ALE scheme, to show that the new nonconforming scheme is able to avoid mesh distortion even after very long simulation times, refer to Figure 15. Other interesting applications will be shown in Section 6 where the nonconforming ALE framework will be coupled together with a well balancing technique both for studying the shallow water equations and the Euler equations with gravity.

Note that, the algorithm presented here is not necessarily limited to logically straight slipe lines. In Section 3.7 we therefore show first preliminary results for general, logically non-straight slide lines.

3.1 Nonconforming mesh motion

In this Section we focus on the detailed description of the procedure needed to determine how the computational mesh moves: a crucial point in any moving mesh algorithm, because following the fluid exactly with its own velocity may

become sometimes very complex, leading to highly deformed or degenerate control volumes. Moreover these damaged elements may drastically reduce the admissible time step, which is computed under a classical CFL stability condition, see (10), causing a slowdown or even an interruption of the algorithm.

In particular, at each time step our algorithm computes the new node positions through the following intermediate stages. First, the edges along relevant shear flows are detected (see Section 3.1.2) and the nodes on these edges are marked as problematic. Then the new node positions are computed according to the type of the considered node, in particular

- Regular non-hanging nodes that are not in regions of relevant shear flow (i.e. they have not been marked as problematic) are evolved using a mass-weighted Cheng and Shu node solver, see (43).
- Regular non-hanging nodes that are in regions of relevant shear flow (i.e. they have been marked as problematic) are *doubled*; their new position is projected along the nearest interface edge, and they subsequently change their type from regular non-hanging nodes to *hanging* nodes (refer to Section 3.2.1).
- Hanging nodes on an edge are allowed to slide only along that edge (see Section 3.2.2), and if they get too close to other nodes, they are *merged* together (deleted), eventually changing back their type from hanging nodes to regular non-hanging nodes (refer to Section 3.2.3).

Associated to *b)* and *c)* there is a procedure for updating all the connectivities of the unstructured mesh: we would like to underline that all these procedures are done at a *local* level, so affecting only the neighbors of the considered node and nothing else.

3.1.1 Connectivity matrices

Since the core of the proposed method is the motion and the changing of the nonconforming mesh topology in time, we have to know all the connectivities of the mesh and to maintain them updated. In this way we will have enough information both to rearrange the mesh after the insertion of a new node, or the fusion of two existing nodes, and to know all the neighbors of each space-time lateral surface during the numerical flux computation.

As in the standard conforming case for each element P_i^n we save the global numbering of its vertexes $d_{i1}^n, \dots, d_{iN_i^n}^n$ in row i of a matrix called `tri` in counterclockwise order, and in matrix `Elem2Edge` we store the global numbering of its edges $e_{i1}^n, \dots, e_{iN_i^n}^n$. However, in the nonconforming case, additional connectivity tables are needed, since more than two elements can share the same edge and more than two

points can belong to it. For each edge e_{ij}^n , we store the elements that share it in row j of matrix `Edge2Elem`, and all the nodes that belong to e_{ij}^n in row j of matrix `Edge2Vertex` in such a way that the first two entries of each row contain the endpoints of the corresponding edge. Then, for each node we memorize the edge to which it belongs in `Vertex2Edge` (both if this node is an endpoint of the edge or an intermediate point) and the elements for which it is a vertex in `Vertex2Elem`. Moreover, each node has a label that indicates whether the node is free to move everywhere, if it has been doubled, or if it is constrained to slide along a particular edge, i.e. if it is a hanging node.

Besides, we allow our data structures to be completely dynamic in such a way that nodes and edges can appear and disappear in time: so rows can be added to our matrices or be nullified, and the information regarding which global numbering of nodes and edges is currently used is always available.

3.1.2 Shear interface detector

Since the sliding interfaces are not defined *a priori* by the user, at the beginning of each time step the algorithm has first to identify along which *edges* the shear interfaces lie, and mark the corresponding edges and nodes. Basically an edge e will be considered at the interface if the tangential velocity difference ΔV_e across e exceeds a certain threshold value κ_e . So for each edge we need to compute ΔV_e and κ_e .

Given the set of nodes S_e^n over the edge e , and the set of neighbors S_j^n of each node j , the threshold value κ_e is computed as

$$\kappa_e = \min_{j \in S_e^n} \kappa_j, \quad \text{with} \quad \kappa_j = \max_{i \in S_j^n} \left(\frac{\alpha d_i}{\|J_i\|} \right), \quad (40)$$

where d_i is the encircle diameter of element P_i^n , J_i is the Jacobian of the transformation that maps element P_i^n in the corresponding reference element, the norm is the two-norm of Frobenius divided by $\sqrt{2}$ (other matrix norms could also be used), and α is chosen in $[0, 1]$ according to the desired sensitivity of the detector. If the velocity jump at the interface is very large, the value of α does not matter. Instead, where the velocity field changes smoothly, the number of interfaces, and as a consequence the number of new nodes, will be dependent on α . Moreover, in the limit $\alpha \rightarrow +\infty$ we recover the standard conforming algorithm.

Once the threshold value has been fixed we loop over all the edges of the mesh: for each edge e we consider all its neighbors and we compute their tangential velocity with respect to e . Say, for example, that two elements $A = P_a^n$ and $B = P_b^n$ with area $|P_a^n|$ and $|P_b^n|$ share the same edge e and their tangential velocities are v_A^n and v_B^n . If the quantity ΔV_e

exceeds κ_e

$$\Delta V_e = \frac{|v_A^n |P_a^n| - v_B^n |P_b^n|}{(|v_A^n| |P_a^n| + |v_B^n| |P_b^n| + \varepsilon)} \geq \kappa_e, \quad (41)$$

with $\varepsilon = 10^{-14}$ to avoid division by zero, then edge e is marked as an edge at a shear interface, and the elements A and B are divided into two different groups: the elements on the left and the ones on the right with respect to this particular edge e . Afterwards, we also need to find the *nodes* that have *to be doubled* and to separate their Voronoi neighbors (the elements stored in `Vertex2Elem`) into two groups. So we loop over the nodes considering the ones which belong to an interface edge. If in their list of Voronoi neighbors there are elements from both the sides of the interface, according to the previous subdivision, we mark them and we separate their Voronoi neighbors into two groups which are stored in two matrices. Note that the two cycles, the one over the edges and the other over the nodes, are not nested one into the other, but are run one after the other.

3.2 Node motion

At this point we are able to distinguish between nodes far away from the interfaces, hanging nodes and nodes which lie at the interface. So we loop over the nodes and according to their labels we choose what to do. First, consider a *regular* non-hanging node k located in a smooth region. We compute its coordinates at the new time level t^{n+1} simply by

$$\mathbf{x}_k^{n+1} = \mathbf{x}_k^n + \Delta t \bar{\mathbf{v}}_k^n, \quad (42)$$

where $\bar{\mathbf{v}}_k^n$ is obtained using the node solver of Cheng and Shu. Cheng and Shu introduced a very simple and general formulation for obtaining the final node velocity, which is chosen to be the arithmetic average velocity among all the contributions coming from the Voronoi neighbor elements \mathcal{V}_k^n . Moreover, following the ideas presented in [19] we take a mass weighted average velocity among the neighborhood \mathcal{V}_k^n , that is,

$$\bar{\mathbf{v}}_k^n = \frac{1}{\mu_k} \sum_{P_j^n \in \mathcal{V}_k^n} \mu_{k,j} \bar{\mathbf{v}}_{k,j} \quad (43)$$

with

$$\mu_k = \sum_{P_j^n \in \mathcal{V}_k^n} \mu_{k,j}, \quad \mu_{k,j} = \rho_j^n |P_j^n|. \quad (44)$$

The local weights $\mu_{k,j}$, which are the masses of the elements P_j^n , are defined by multiplying the cell averaged value of density ρ_j^n (or water depth h_j^n for shallow water flows) with the cell area $|P_j^n|$. The local contributions $\bar{\mathbf{v}}_{k,j}$ in a pure Lagrangian context represent the fluid velocity in the j^{th} neighbor of vertex k , while in the ALE framework they can be

obtained either according to an arbitrary, prescribed mesh velocity function or by the local fluid velocity.

Now let us consider the *nodes at the interfaces*. The following considerations are carried out by supposing for the moment that each interface is separated from the others and lies on a straight line. Even if this is a rigid constraint, already with this configuration interesting test cases can be studied; a proof of concept that the extension to the general case of piece-wise linear interfaces is feasible will be presented in Section 3.7.

3.2.1 Insertion of a new node

The first situation we encounter is a node k that has some of its Voronoi neighbors on the left of the interface, call them *left neighbors*, $\mathcal{V}_{k,\text{left}}$, and the others on the right of the same interface, call them *right neighbors*, $\mathcal{V}_{k,\text{right}}$; these two sets of neighbors have been provided by the *interface detector* described above. We apply the node solver of Cheng and Shu at the two sets of neighbors obtaining two different new coordinates

$$\tilde{\mathbf{x}}_{k,\text{left}}^{n+1} = \mathbf{x}_k^n + \Delta t \sum_{P_j^n \in \mathcal{V}_{k,\text{left}}} \frac{\mu_{k,j}}{\mu_k} \bar{\mathbf{v}}_{k,j}, \quad \text{and} \quad (45)$$

$$\tilde{\mathbf{x}}_{k,\text{right}}^{n+1} = \mathbf{x}_k^n + \Delta t \sum_{P_j^n \in \mathcal{V}_{k,\text{right}}} \frac{\mu_{k,j}}{\mu_k} \bar{\mathbf{v}}_{k,j}.$$

We allow this kind of nodes to move only along the interface, so basically according to their averaged tangential velocity with respect to the interface: for this reason we need to find the nearest interface edges and to project onto them the coordinates in (45) obtaining thus $\mathbf{x}_{k,\text{left}}^{n+1}$ and $\mathbf{x}_{k,\text{right}}^{n+1}$. Call the nearest interface edges belonging to the left elements e_1^ℓ and e_2^ℓ , and the nearest interface edges belonging to the right elements e_1^r and e_2^r (suppose also that $e_1^{\ell,r}$ are closer to $\tilde{\mathbf{x}}_{k,\text{left}}^{n+1}$ than to $\tilde{\mathbf{x}}_{k,\text{right}}^{n+1}$, so that $\mathbf{x}_{k,\text{left}}^{n+1}$ is obtained by projecting $\tilde{\mathbf{x}}_{k,\text{left}}^{n+1}$ onto e_1^ℓ , and $\mathbf{x}_{k,\text{right}}^{n+1}$ by projecting $\tilde{\mathbf{x}}_{k,\text{right}}^{n+1}$ onto e_1^r). We decide to assign $\mathbf{x}_{k,\text{left}}^{n+1}$ as new coordinate to the old node k

$$\mathbf{x}_k^{n+1} = \mathbf{x}_{k,\text{left}}^{n+1} \quad (46)$$

and to create a new node with global number k_{new} and coordinates (at time n and $n+1$)

$$\mathbf{x}_{k_{\text{new}}}^n = \mathbf{x}_k^n \quad \text{and} \quad \mathbf{x}_{k_{\text{new}}}^{n+1} = \mathbf{x}_{k,\text{right}}^{n+1}. \quad (47)$$

The global number k_{new} can be larger than the total number of nodes if all the numbers below are currently used, otherwise we choose the first of the unused numbers (indeed if two nodes have been merged together one of their global numbers is no more used, see Section 3.2.3).

Now we have to update the connectivity tables taking into account the insertion of this new node. See also Figure 4 to follow our construction. First, in matrix `tri` we

substitute k with k_{new} in all the right elements; moreover, we consider matrix `Vertex2Elem` and in row k we leave only the left elements and we put the others in row k_{new} (because now k is no more a vertex for the right neighbors). Then we have to deal with the edges: if $e_1^\ell = e_1^r$ we need to substitute e_1^r with a new edge $e_{1_{\text{new}}}^r$. In matrix `Elem2Edge` all the right neighbors change e_1^r with $e_{1_{\text{new}}}^r$, and in matrix `Edge2Elem` we insert a new row $e_{1_{\text{new}}}^r$ equal to row e_1^r (the new edge inherits all the characteristics from the old one). The same has to be done if $e_2^\ell = e_2^r$. The endpoints of these new edges are the endpoints of the substituted edges seen from the right (so basically there is k_{new} instead of k). The endpoints of the left edges do not change. Besides we add k as intermediate point in e_1^r and k_{new} as intermediate point of e_2^ℓ , (note that an edge is allowed to have more than one intermediate point). In this way also matrix `Edge2Vertex` has been updated. Matrix `Vertex2Edge` is easily modified at the same time. Finally, we have to revise the list of neighbors: in particular the edges that gained an intermediate point (e_1^r and e_2^ℓ) gain also neighbors. In particular the new neighbors of e_1^r are the left neighbors of e_2^ℓ and the new neighbors of e_2^ℓ are the right neighbors of e_1^r . This allows us to update `Edge2Elem` and `Elem2Edge`.

At the end we mark with a label the nodes which are intermediate for an edge: we call them *hanging nodes* and they are constrained to move along that edge. Note that in the case of straight slip-lines no distinction between master and slave nodes is required, since both will move along the same straight interface. To extend the algorithm to the case of piece-wise linear interfaces, this distinction is introduced in such a way that only slave nodes will be constrained to slide along edges, while the master nodes can move freely. For some first preliminary results concerning the extension to completely general slide lines, see Section 3.7.

3.2.2 Hanging nodes

Consider a hanging node k which lies on the edge e : it is at the interface and it is a vertex only of elements lying on one side of the interface, so its Voronoi neighbors are in the same smooth region. However it is not free to move everywhere but it must slide along that edge, to avoid creation of holes or superposition of elements in the mesh.

We apply the averaged node solver of Cheng and Shu among its Voronoi neighbors, we find its new coordinates $\tilde{\mathbf{x}}_k^{n+1}$ and we project them over edge e , obtaining \mathbf{x}_k^{n+1} . Now, we compute also the new coordinates of the other points over edge e . If two of them are sufficiently close, we decide to merge them (see Section 3.2.3), otherwise the computed coordinates \mathbf{x}_k^{n+1} are the new coordinates of such a node and no update of the connectivity matrices is required.

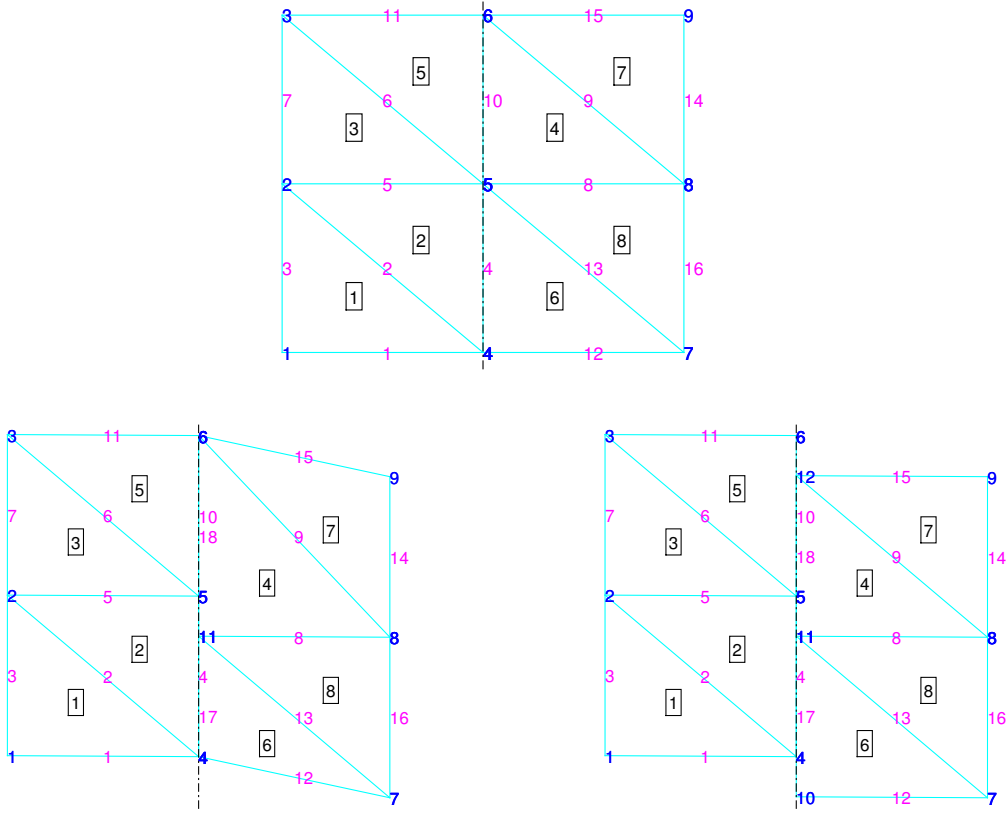


Fig. 4: Example of how to double a node. At the top we show the situation before a nonconforming motion, and at the bottom after the motion and the corresponding update of the connectivity matrices. Precisely at the bottom on the left we have supposed to move in a nonconforming way only one of the nodes at the interface, while the realistic motion of all the nodes at the interface is shown on the right. The black vertical dotted line represents the interface: suppose that the elements on the left $\{1, 2, 3, 5\}$ move with velocity $\mathbf{v} = (0, 2)$ and the elements on the right $\{4, 6, 7, 8\}$ move with velocity $\mathbf{v} = (0, -2)$. We want to double vertex number $k = 5$, so we insert a new node $k_{\text{new}} = 11$. The nearest interface edges on which we project the new coordinates of node 5 are $e_1^l = e_1^r = 10$ and $e_2^l = e_2^r = 4$. Note that edges $e_1^{l,r}$ are closer to k than to k_{new} . Since the edges from the left and from the right are equal we create two new edges $e_{1_{\text{new}}}^r = 18$ and $e_{2_{\text{new}}}^l = 17$. The endpoints of edges 10 and 4 remain untouched. Edge 4 gains an intermediate point, the node 11, and edge 18 gains the node 5. To better understand we list now the vertexes of each edge at the end of the updating process (first we write the endpoints and then, if existing, the intermediate points): $e_1^l = 10 \rightarrow \{5, 8\}$, $e_2^l = 4 \rightarrow \{4, 5, 11\}$, $e_1^r = 18 \rightarrow \{11, 6, 5\}$ and $e_2^r = 17 \rightarrow \{4, 11\}$. Finally, elements $\{1, 3, 5, 6, 7, 8\}$ maintain the same edge neighbors, while the neighbors of elements 2 and 4 are augmented: indeed edge 4 has neighbors $\{2, 6, 4\}$ and edge 18 has neighbors $\{4, 5, 2\}$. Note that the situation on the right appears to be more complicated only because also nodes 4 and 6 have been doubled and so the corresponding update of the connectivity matrices has been done.

3.2.3 Fusion of two existing nodes

Suppose we computed the new coordinates at time t^{n+1} of all the nodes k_i over the same edge e denoted by $\mathbf{x}_{k_i}^{n+1}$, which are assumed to be already projected onto the straight line spanned by edge e . If the new coordinates of two of them, say k_1 and k_2 , are too close, we decide to merge them. Moreover, if one intermediate node of edge e falls outside the edge, we decide to merge it with the closest endpoint of the edge. Since the loop over the nodes is carried out according to the increasing global numbering of the nodes, we decide

to remove the node with the largest global number (we call it *dead node*, k_{dn}) because we have not worked with it yet, and to maintain the one with the smallest global numbering (call it *fusion node*, k_{fn}) assigning to it as new coordinates the average between $\mathbf{x}_{k_1}^{n+1}$ and $\mathbf{x}_{k_2}^{n+1}$

$$\mathbf{x}_{k_{\text{fn}}}^{n+1} = \frac{\mathbf{x}_{k_1}^{n+1} + \mathbf{x}_{k_2}^{n+1}}{2}. \quad (48)$$

We assign the same coordinates also to the dead node

$$\mathbf{x}_{k_{\text{dn}}}^{n+1} = \frac{\mathbf{x}_{k_1}^{n+1} + \mathbf{x}_{k_2}^{n+1}}{2}. \quad (49)$$

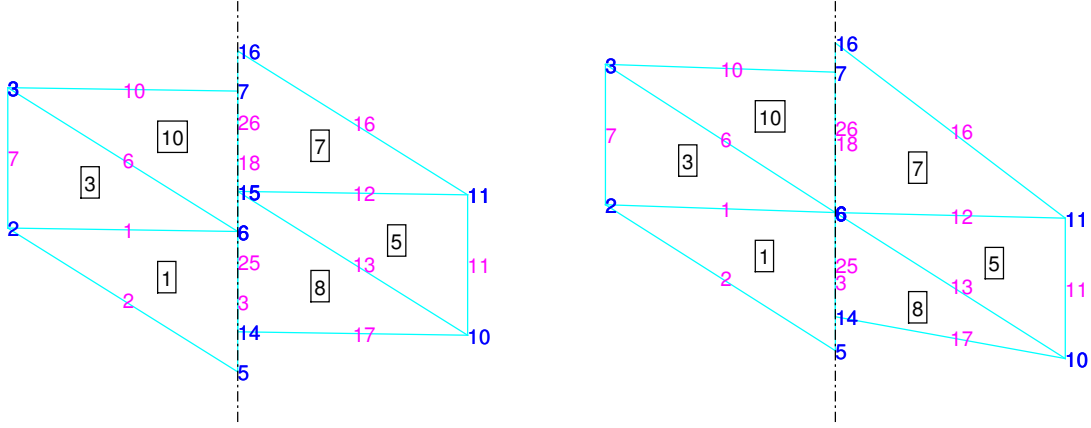


Fig. 5: Example of how to merge two existing nodes. The black dotted line represents the interface: suppose that the elements on the left $\{1, 3, 10\}$ move with positive velocity and the elements on the right $\{5, 7, 8\}$ move with negative velocity. On the left we show the situation at time t^n and on the right at time t^{n+1} . Nodes 6 and 15 at t^{n+1} will be so close that we decide to merge them (as in the previous example, for the sake of clarity, we present on the right the situation after the fusion of only two nodes). We maintain the smallest global number so $k_{\text{fn}} = 6$ and we remove $k_{\text{dn}} = 15$. In `triNew` elements $\{5, 7, 8\}$ substitute $k_{\text{dn}} = 15$ with $k_{\text{fn}} = 6$. Note that in `tri` nothing changes, so some elements refer to node 6 and some other to node 15, but everything works because at time t^{n+1} they have the same new coordinates $\mathbf{x}_{k_{\text{dn}}}^{n+1} = \mathbf{x}_{k_{\text{fn}}}^{n+1}$ and at the successive time step t^{n+2} `tri` will no longer exist because it will be overwritten by `triNew`. In row k_{fn} of matrix `Vertex2ElemNew` there are listed elements $\{1, 3, 5, 7, 8, 10\}$, while row k_{dn} is empty. In row k_{fn} of matrix `Vertex2EdgeNew` there are edges $\{1, 3, 6, 12, 13, 18, 25, 26\}$, while row k_{dn} is empty. List `Edge[dn-fn]` contains edges $\{18, 25\}$ and list `Elem[dn-fn]` contains elements $\{8, 10\}$. Knowing these lists we can update matrices `Edge2ElemNew` because we remove element 8 from the neighbor of edge 18 and element 10 from the neighbors of edge 25. In this case even if we removed the segment $\bar{6}, 15$ no edge becomes equal so we do not need to merge edges neither to update `Elem2EdgeNew`.

Now, we need to update the connectivity tables. See also Figure 5 to follow our construction. This process is somehow more complicated than the nodes splitting. Indeed when we insert a new node at time t^{n+1} we only add information without losing anything about the previous time step, and even if it is true that the right neighbors of a doubled node k change their node k with a new one k_{new} , we can dispose of all its reference simply by giving to k_{new} at time t^n the same coordinates of k , see also (45). On the contrary, when we remove a node we lose all the reference to it, reference that, only for time t^{n+1} , we still need during the computation of the interface fluxes in the Finite Volume scheme (it is for this reason that in (49) we have assigned the coordinates $\mathbf{x}_{k_{\text{dn}}}^{n+1}$ even to the dead node). So we decide to duplicate some of the connectivity tables, creating `triNew`, `Elem2EdgeNew`, `Edge2ElemNew`, `Edge2VertexNew`, and `Vertex2ElemNew`. During the insertion procedure we modify in the same way both the old and the new matrices, while during the fusion we modify only the new matrices. Hence we can use the old ones in the Finite Volume scheme, because they store all the needed information (for example they refer both to the dead and the fusion node which have the same coordinates at the new time t^{n+1}), while when we advance in time, to t^{n+2} , we maintain updated only the new ones because the information

about two previous time steps are no longer necessary and so we can overwrite the old connectivity matrices with the new ones.

First, in matrix `triNew` we substitute k_{dn} with k_{fn} in all the neighbors of the dead node; moreover, we consider matrix `Vertex2ElemNew`, in row k_{fn} we put both the neighbors of the dead and the fusion node and we nullify row k_{dn} . We do the same with matrix `Vertex2EdgeNew`: we nullify row k_{dn} and we put in row k_{fn} all the edges that contain k_{fn} or k_{dn} . Then all the edges that contain k_{dn} substitute it with k_{fn} (in matrix `Edge2VertexNew`), whereas the edges with both k_{dn} and k_{fn} (that we memorize in a list `Edge[dn-fn]`) remove k_{dn} . We note that merging k_{dn} and k_{fn} we are removing the segment in between, so we look for the edges that contain it (listed in `Edge[dn-fn]`) and its neighbor elements that we list in `Elem[dn-fn]`. We update now matrix `Edge2ElemNew` because the edges in `Edge[dn-fn]` have no more one of the neighbors in `Elem[dn-fn]`. Afterward we check if the absence of this segment makes some edges in `Edge[dn-fn]` equal: in this case we remove one of them (the one with the largest global number) and we update correspondingly the new connectivity matrices. Besides we modify the labels telling us if a node is hung to some edges and which nodes and edges are currently existing. This last step

prevents us to work again with disappeared nodes and allows us to reuse their global numbering when we want to insert a new node or a new edge.

3.3 Space–time control volumes - nonconforming case

As introduced in Section 2.4, our direct ALE schemes are based on integrating the governing equation directly in space and time over the *space–time control volumes* C_i^n .

Let us consider now the peculiarity connected with the nonconforming case, i.e. the case when on the same edge we can find more than the two extreme nodes that means that more than two control volumes share the same edge. In this case the surface can be subdivided in sub–surfaces. The treatment of the nonconforming lateral space–time surfaces basically requires only to repeat the computation of the necessary geometric information over each sub–surface (the same will hold for the flux computation, which will be simply split in several parts).

For example, consider the case of $\partial C_{i,j}^n$ with the four standard vertexes and two more hanging nodes on the edges orthogonal to the time coordinate (as in the middle of Figure 6). Then the lateral surface is shared between three (and not two, as usual) control volumes. However it can be subdivided into two pieces, each one shared between only two control volumes, which are still trapezoidal, so each of them can be mapped into the reference element using the standard map in (31), just taking care to select in a correct way the vertexes of each piece. Hence areas, normal vectors, and space–time midpoints can be computed exactly as in the conforming case but on each part. Next, we have to analyze the two extreme situations that happens due to our dynamical insertion and deletion of nodes. On the left and on the right of Figure 6 we have reported these two limiting cases: first, at time t^{n+1} a new node has been inserted, which at the previous time t^n did not yet exist; or vice-versa, at time t^{n+1} a hanging node is merged together with one of the other vertexes and hence it disappears. In these cases the lateral surfaces can still be divided into two parts, and even if one of them is triangular it can still be treated as a degenerate quadrilateral face, so all the computations can be performed, once again, as above. The coordinates of a hanging node at the moment of its creation or destruction will be set equal to those of the vertex from which the hanging node was born, or those of the vertex to whom it was merged, respectively.

3.4 MUSCL type space–time reconstruction

Here, we restrict ourselves to a simple second order FV scheme. Thus, the first step consists in reconstructing a second order spatial polynomial $\mathbf{w}_h(\mathbf{x}, t^n)$ for each P_i^n exploiting the cell averages of the cell $\hat{\mathbf{u}}_{0,i}^n$ and of its neighbors $\hat{\mathbf{u}}_{0,i_j}^n$,

through a MUSCL approach. Then, in order to obtain the high order both in space and in time predictor $\mathbf{q}_h(\mathbf{x}, t^n)$ of (11), we employ a time-evolution procedure, the so-called MUSCL–Hancock procedure, that was for the first time proposed by van Leer in [152] and which is very well explained in [149],

For the *spatial* reconstruction, let us define a polynomial $\mathbf{w}_h(\mathbf{x}, t^n)$ of the form

$$\begin{aligned} \mathbf{w}_h(\mathbf{x}, t^n) |_{P_i^n} = \mathbf{w}_i(\mathbf{x}, t^n) &= \hat{\mathbf{u}}_{0,i}^n + \hat{\mathbf{u}}_{1,i}^n(x - x_{b_i}^n) + \hat{\mathbf{u}}_{2,i}^n(y - y_{b_i}^n), \\ \mathbf{x} = (x, y) &\in P_i^n, \end{aligned} \quad (50)$$

where $\mathbf{x}_{b_i}^n = (x_{b_i}^n, y_{b_i}^n)$ is the barycenter of cell P_i^n . We denote by \mathcal{S}_i^n the set of neighbors of P_i^n that share a common edge with P_i^n (the set \mathcal{S}_i^n may change at each time step). To compute $\hat{\mathbf{u}}_{1/2,i}^n$, integral conservation is imposed on each element of \mathcal{S}_i^n

$$\frac{1}{|P_{i,j}^n|} \int_{P_{i,j}^n} \mathbf{w}_h(\mathbf{x}, t^n) d\mathbf{x} = \hat{\mathbf{u}}_{0,i_j}^n \quad \forall P_{i,j}^n \in \mathcal{S}_i^n. \quad (51)$$

The above system is in general over-determined, so we add the constraint that equation (51) holds exactly at least for P_i^n . This is easily satisfied by rewriting the equations as

$$\begin{aligned} \frac{1}{|P_{i,j}^n|} \int_{P_{i,j}^n} \hat{\mathbf{u}}_{1,i}^n(x - x_{b_i}^n) + \hat{\mathbf{u}}_{2,i}^n(y - y_{b_i}^n) d\mathbf{x} &= \hat{\mathbf{u}}_{0,i_j}^n - \hat{\mathbf{u}}_{0,i}^n \\ \forall P_{i,j}^n &\in \mathcal{S}_i^n, \end{aligned} \quad (52)$$

then we solve (52) via a classical least-squares approach using the normal equation of (52), and we thus obtain the *non-limited* slope $\hat{\mathbf{u}}_{1/2,i}^n$.

To ensure that new extrema are not created in the reconstruction process, we employ the classical *slope limiter* function Φ_i presented by Barth and Jespersen in [5]. The idea is to find the largest admissible Φ_i in such a way that

$$\tilde{\mathbf{w}}_h(\mathbf{x}, t^n) = \hat{\mathbf{u}}_{0,i}^n + \Phi_i(\hat{\mathbf{u}}_{1,i}^n(x - x_{b_i}^n) + \hat{\mathbf{u}}_{2,i}^n(y - y_{b_i}^n)) \quad (53)$$

satisfies

$$\min_{j \in \mathcal{V}_i^n} \hat{\mathbf{u}}_{0,i_j}^n = \mathbf{Q}_i^{min} \leq \tilde{\mathbf{w}}_h(\mathbf{x}, t^n) \leq \mathbf{Q}_i^{max} = \max_{j \in \mathcal{V}_i^n} \hat{\mathbf{u}}_{0,i_j}^n, \quad (54)$$

where \mathbf{Q}_i^{max} and \mathbf{Q}_i^{min} are the componentwise maximum and minimum among the cell-averages of the set \mathcal{V}_i^n , respectively. The set \mathcal{V}_i^n contains *all* the vertex neighbors of P_i^n and the element P_i^n itself. Since \mathbf{w}_h is obtained as a piecewise linear reconstruction of the data, its extrema occur at the vertices of P_i^n . Hence, to compute the limiter for each conserved variable, it suffices to find for all vertices d_{i_j} of P_i^n

$$\Phi_{i,j} = \begin{cases} \min\left(1, \frac{\mathbf{Q}_i^{max} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \hat{\mathbf{u}}_{0,i}^n}\right), & \text{if } \mathbf{w}_{h,j} - \hat{\mathbf{u}}_{0,i}^n > 0 \\ \min\left(1, \frac{\mathbf{Q}_i^{min} - \mathbf{Q}_i^n}{\mathbf{w}_{h,j} - \hat{\mathbf{u}}_{0,i}^n}\right), & \text{if } \mathbf{w}_{h,j} - \hat{\mathbf{u}}_{0,i}^n < 0 \\ 1 & \text{if } \mathbf{w}_{h,j} - \hat{\mathbf{u}}_{0,i}^n = 0. \end{cases} \quad (55)$$

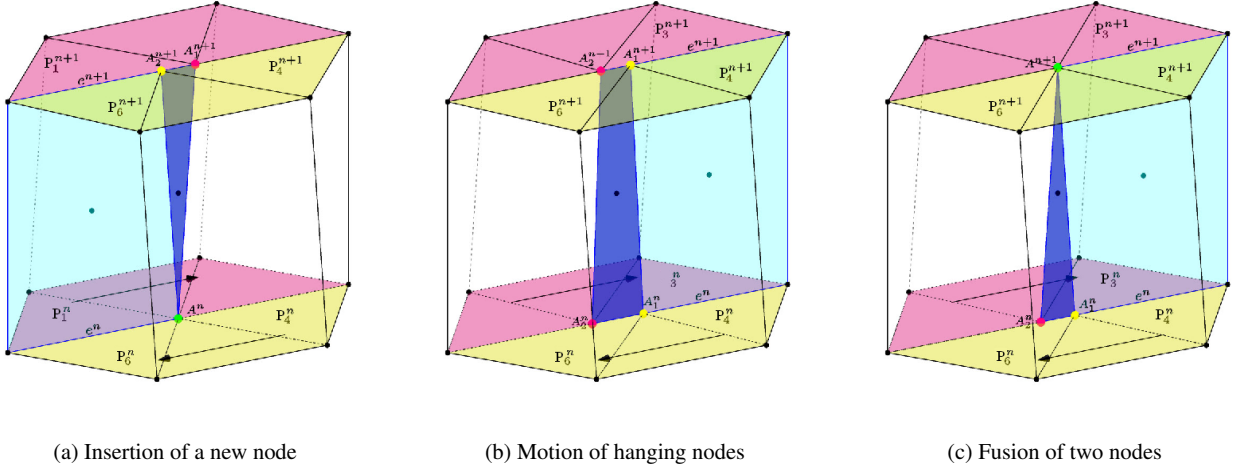


Fig. 6: Suppose that at time t^n across the pink and the yellow elements the tangential fluid velocity changes sharply, as suggested by the arrows; at t^{n+1} the pink elements will move in one direction and the yellow ones will move in the opposite direction. In (a) at time t^n we have a conforming mesh, but in order to avoid a severe mesh distortion in the following time steps we decide to double the green node A^n . So at time t^{n+1} there are both A_1^{n+1} and A_2^{n+1} : A_1^{n+1} is a vertex for the pink elements and A_2^{n+1} is a vertex for the yellow elements. Moreover A_2^{n+1} is hung to edge e^{n+1} . So the blue lateral face of P_i^n , which has e^n and e^{n+1} as bases, is composed by two pieces: the one in light blue which is trapezoidal and touches elements P_1 and P_6 , and the one in dark blue which is triangular and touches elements P_1 and P_4 . Note in particular that we need to compute the flux between P_1 and P_4 during the interval $[t^n, t^{n+1}]$ even if at time t^n they were not in contact. In (b) we show the intermediate situation where a hanging node slides along an edge. In this case the blue surface is still divided into two parts and it is shared between three elements P_3, P_4 and P_6 , so the computation of two fluxes will be required. In order to compute the fluxes and to maintain the second order of accuracy of the entire method the reconstruction polynomial $\mathbf{q}_h(\mathbf{x}, t)$ will be evaluated at the midpoints of each of the part of the lateral surface. Finally, in (c) we report the last limiting case: A_1^n and A_2^n are close and at t^{n+1} will be even closer since they are moving one towards the other, so we decide to merge them and to restore the conforming and simpler situation, in particular to avoid that A_1^n will leave edge e^{n+1} at time t^{n+1} . Eventually A^{n+1} could be doubled again at t^{n+2} if the tangential velocity difference across the interface is sufficiently large.

with $\mathbf{w}_{h,j} = \mathbf{w}_h(\mathbf{x}_j, t^n)$ (ratios and inequalities are to be understood component-wise). Then, the slope limiter is defined as

$$\Phi_i = \min_j (\Phi_{i,j}). \quad (56)$$

Finally, second order of accuracy in *time* is achieved by an element-local predictor stage that evolves the reconstructed polynomials $\mathbf{w}_i(\mathbf{x}, t^n)$ within each element $P_i^n(t)$ during the time interval $[t^n, t^{n+1}]$. The piecewise space-time polynomials are denoted by $\mathbf{q}_h(\mathbf{x}, t)$, and are of the form

$$\begin{aligned} \mathbf{q}_h(\mathbf{x}, t)|_{P_i^n} &= \mathbf{q}_i(\mathbf{x}, t^n) \\ &= \hat{\mathbf{u}}_{0,i}^n + \Phi_i(\hat{\mathbf{u}}_{1,i}^n(x - x_{b_i}^n) + \hat{\mathbf{u}}_{2,i}^n(y - y_{b_i}^n)) + \partial_t \mathbf{Q}_i(t - t^n), \quad (57) \\ \mathbf{x} &\in P_i(t), \quad t \in [t^n, t^{n+1}]. \end{aligned}$$

The value of $\partial_t \mathbf{Q}_i$ can be easily recovered through the strong form of the PDE

$$\partial_t \mathbf{Q} = -\mathbf{f}_x(\mathbf{Q}) - \mathbf{g}_y(\mathbf{Q}) + \mathbf{S}(\mathbf{Q}), \quad (58)$$

where the r.h.s of (58) can be easily computed. Indeed the fields \mathbf{f} and \mathbf{g} over P_i^n are approximated as linear fields

$$\begin{aligned} \mathbf{f}(x, y) &= \mathbf{f}_0 + \mathbf{f}_1(x - x_i) + \mathbf{f}_2(y - y_i), \\ \mathbf{g}(x, y) &= \mathbf{g}_0 + \mathbf{g}_1(x - x_i) + \mathbf{g}_2(y - y_i), \end{aligned} \quad (59)$$

whose coefficients \mathbf{f}_i and \mathbf{g}_i are determined interpolating the values of the fields computed at the vertices j of P_i^n , i.e.

$$\mathbf{f}(\tilde{\mathbf{w}}_h(\mathbf{x}_j, t^n)) \quad \text{and} \quad \mathbf{g}(\tilde{\mathbf{w}}_h(\mathbf{x}_j, t^n)), \quad \forall j = 1, \dots, N_{V_i}^n. \quad (60)$$

Then $\mathbf{f}_1 = \mathbf{f}_x(\mathbf{Q})$ and $\mathbf{g}_2 = \mathbf{g}_y(\mathbf{Q})$. Besides, the source $\mathbf{S}(\mathbf{Q})$ is computed at the barycenter $\mathbf{x}_{b_i}^n$ of P_i^n , i.e. $\mathbf{S}(\tilde{\mathbf{w}}_h(\mathbf{x}_{b_i}^n, t^n))$.

Now, \mathbf{q}_h can be employed in the corrector step, exactly as described in Section 2.7.

3.5 Shallow water equations

In the next Section, we will solve a large set of numerical tests in order to validate the presented nonconforming direct

ALE scheme. The robustness of the method is checked both on smooth and discontinuous problems related to the shallow water equations written both in Cartesian and in polar coordinates. These equations are briefly recalled here.

3.5.1 Cartesian coordinates

We consider the shallow water equations with flat bottom topography, which can be cast into form (1) with

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \end{pmatrix}, \quad (61)$$

where the variables h , u and v are respectively the water depth, the velocity in x -direction, and the velocity in y -direction. The term $\bar{p} = \frac{1}{2}gh^2$ denotes the averaged pressure force along the water depth (normalized with the fluid density ρ), where g is the gravity acceleration along the vertical direction. Since the shallow water equations are derived from depth-integrating the incompressible Navier-Stokes equations assuming that the horizontal length scale is much larger than the vertical one, the vertical pressure is assumed to be hydrostatic.

The Jacobian matrices, necessary for the computation of the ALE Jacobian matrix in (26), are

$$\frac{\partial \mathbf{f}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 1 & 0 \\ a^2 - u^2 & 2u & 0 \\ -uv & v & u \end{pmatrix}, \quad \frac{\partial \mathbf{g}}{\partial \mathbf{Q}} = \begin{pmatrix} 0 & 0 & 1 \\ -uv & v & u \\ a^2 - v^2 & 0 & 2v \end{pmatrix}. \quad (62)$$

where $a = \sqrt{gh}$.

3.5.2 Polar coordinates

We recover here the formulation of the shallow water equations in polar coordinates. Consider the usual relation between polar (r, φ) and Cartesian (x, y) coordinates

$$x = r \cos \varphi, \quad \text{and} \quad y = r \sin \varphi, \quad (63)$$

and the corresponding relations for the derivatives

$$\begin{aligned} \frac{\partial}{\partial x} &= \cos \varphi \frac{\partial}{\partial r} - \frac{\sin \varphi}{r} \frac{\partial}{\partial \varphi}, \quad \text{and} \\ \frac{\partial}{\partial y} &= \sin \varphi \frac{\partial}{\partial r} + \frac{\cos \varphi}{r} \frac{\partial}{\partial \varphi} \end{aligned} \quad (64)$$

and let u_ρ and u_φ be respectively the radial and the tangential component of the velocity, linked to u and v by

$$u = \cos \varphi u_\rho - \sin \varphi u_\varphi, \quad v = \sin \varphi u_\rho + \cos \varphi u_\varphi. \quad (65)$$

Now by substituting into (61) the expressions given in (65) and (64), after some calculations, we derive a new set of hyperbolic equations

$$\begin{cases} \frac{\partial rh}{\partial t} + \frac{\partial rhu_\rho}{\partial r} + \frac{\partial hu_\varphi}{\partial \varphi} = 0, \\ \frac{\partial rhu_\rho}{\partial t} + \frac{\partial}{\partial r} \left(rhu_\rho^2 + \frac{1}{2}grh^2 \right) + \frac{\partial hu_\rho u_\varphi}{\partial \varphi} = hu_\varphi^2 + \frac{1}{2}gh^2, \\ \frac{\partial rhu_\varphi}{\partial t} + \frac{\partial rhu_\rho u_\varphi}{\partial r} + \frac{\partial}{\partial \varphi} \left(hu_\varphi^2 + \frac{1}{2}gh^2 \right) = -hu_\rho u_\varphi, \end{cases} \quad (66)$$

which, however, does not yet fit into the form (1), since the fluxes in the above system depend explicitly on the spatial coordinate r (i.e. the system is not autonomous). Thus, we add to the system the trivial equation,

$$\frac{\partial r}{\partial t} = 0, \quad (67)$$

obtaining finally

$$\begin{cases} \frac{\partial rh}{\partial t} + \frac{\partial rhu_\rho}{\partial r} + \frac{\partial hu_\varphi}{\partial \varphi} = 0, \\ \frac{\partial rhu_\rho}{\partial t} + \frac{\partial}{\partial r} \left(rhu_\rho^2 + \frac{1}{2}grh^2 \right) + \frac{\partial hu_\rho u_\varphi}{\partial \varphi} = hu_\varphi^2 + \frac{1}{2}gh^2, \\ \frac{\partial rhu_\varphi}{\partial t} + \frac{\partial rhu_\rho u_\varphi}{\partial r} + \frac{\partial}{\partial \varphi} \left(hu_\varphi^2 + \frac{1}{2}gh^2 \right) = -hu_\rho u_\varphi, \\ \frac{\partial r}{\partial t} = 0. \end{cases} \quad (68)$$

The vector of the conserved variables, the nonlinear flux, and the source can now be written as

$$\mathbf{Q} = \begin{pmatrix} rh \\ rhu_\rho \\ rhu_\varphi \\ r \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} rhu_\rho \\ rhu_\rho^2 + \frac{1}{2}grh^2 \\ rhu_\rho u_\varphi \\ 0 \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} hu_\varphi \\ hu_\rho u_\varphi \\ hu_\varphi^2 + \frac{1}{2}gh^2 \\ 0 \end{pmatrix},$$

$$\mathbf{S} = \begin{pmatrix} 0 \\ vhu_\varphi^2 + \frac{1}{2}gh^2 \\ -hu_\rho u_\varphi \\ 0 \end{pmatrix}. \quad (69)$$

and the Jacobian matrices, necessary for the computation of the ALE Jacobian matrix in (26), are

$$\begin{aligned} \frac{\partial \mathbf{f}}{\partial \mathbf{Q}} &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ -u_\rho^2 + gh & 2u_\rho & 0 & -\frac{1}{2}gh^2 \\ -u_\rho u_\varphi & u_\varphi & u_\rho & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \\ \frac{\partial \mathbf{g}}{\partial \mathbf{Q}} &= \begin{pmatrix} 0 & 0 & \frac{1}{r} & -\frac{hu_\varphi}{r} \\ -\frac{u_\rho u_\varphi}{r} & \frac{u_\varphi}{r} & \frac{u_\rho}{r} & -\frac{hu_\rho u_\varphi}{r} \\ -\frac{u_\varphi^2}{r} + g\frac{h}{r} & 0 & \frac{2u_\varphi}{r} & -\frac{hu_\varphi^2}{r} - g\frac{h^2}{r} \\ 0 & 0 & 0 & 0 \end{pmatrix}. \end{aligned} \quad (70)$$

3.6 Numerical results

The test cases presented in this Section are carried out by supposing straight slip lines and using either the Rusanov or the Osher type flux; the value of α in (40) is always taken equal to $\alpha = 1$ unless otherwise specified, and the CFL number is chosen as $\text{CFL} = 0.3$. Furthermore, the order of convergence is verified first fixing for the mesh motion an arbitrary velocity, then in the case of a steady vortex in equilibrium using the local fluid velocity.

3.6.1 Sanity checks: pure sliding

The numerical examples reported in this Section are sanity checks testing the ability of the method to detect and maintain straight slip-line interfaces.

First, we consider the shallow water equations (61), with initial computational domain given by $\Omega(t_0) = [-2, 2] \times [0, 4]$ and the following initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, 0, -2) & \text{if } x \leq 0, \\ (1, 0, 2) & \text{if } x > 0, \end{cases} \quad (71)$$

which also coincides with the exact solution at any time. We impose *wall* boundary conditions on the left and on the right side of the domain, respectively, whereas at the top and at the bottom of the domain we impose *transmissive* boundary conditions. In Figure 7 we show the numerical results over a triangular mesh and then over a mixed mesh composed of both, triangular and quadrilateral elements. The chosen mesh velocity coincides exactly with the fluid velocity, as in a pure Lagrangian context. At each time step we have verified that the total water volume is conserved up to machine precision both locally and globally and that relation (34), the GCL, is verified also up to machine precision.

Next, we consider as initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = (1, 0, 0.5 \text{ floor}(2x)), \quad -2 \leq x \leq 2, \quad (72)$$

with $\text{floor}(x) = \lfloor x \rfloor$ denoting the lower Gauss bracket, and we run our algorithm until a final time $t = 0.7$ with different *threshold* values, see (40), in such a way that there will be a different number of interfaces along which nodes have to be doubled and merged in time. The discretization of the computational domain is reported in Figure 8. Also in this case we reach the exact solution (that is the initial condition translated in the motion direction), the total volume of water is conserved and relation (34) is verified up to machine precision at each time step and on each element.

Finally, we want to show that the interface can be along any straight line (provided that edges lie over this line): we take as initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, -1, 1) & \text{if } x + y - 2 \leq 0, \\ (1, 1, -1) & \text{if } x + y - 2 > 0, \end{cases} \quad (73)$$

and in Figure 9 we report the computational domain at different times. Again, the numerical solution matches the exact one and as expected, the total volume is conserved and equation (34) is satisfied up to machine precision.

3.6.2 Periodic boundary conditions

The tests reported in the previous Section can be run also by imposing periodic boundary conditions on the top and at the bottom of the computational domain. In Figure 10 we show the discretization of the computational domain at time $t = 100.2$ for the initial conditions in (71) and in (72). We would like to underline that no distortion of the mesh elements appears even after a very long computational time, and as a direct consequence the time steps remain almost constant during the computation. As always in this type of test cases the volume conservation holds and the numerical solution is equal to the exact one up to machine precision.

3.6.3 Riemann problem

Let us now consider as initial condition a Riemann problem with a discontinuity in the water level

$$\mathbf{Q}(\mathbf{x}, 0) = \begin{cases} (1, 0, 0) & \text{if } x \leq 0, \\ (0.5, 0, 0) & \text{if } x > 0, \end{cases} \quad (74)$$

that originates a left-traveling rarefaction fan and a right-moving shock wave. We decided to move the mesh with an arbitrary mesh velocity function

$$\mathbf{V} = (0, 0.5 \text{ floor}(2x)) \quad -2 \leq x \leq 2, \quad (75)$$

in order to check the robustness of the algorithm also in the presence of discontinuities. We impose periodic boundary conditions on the top and on the bottom of the square, and wall boundary conditions on the left and on the right. The final discretization of the computational domain together with the comparison between the numerical and the exact solution are depicted in Figure 11 both for the first order accurate scheme (i.e. without the MUSCL-Hancock strategy for the reconstruction) and the second order accurate scheme.

3.6.4 Convergence test

To verify the order of convergence of the proposed method we study the passive transport of a quantity c , that at time $t = 0$ is taken equal to a Gaussian profile and then will be passively transported in the direction of the fluid flow without changing its shape. The PDE system describing this situation is obtained from the standard shallow water equations

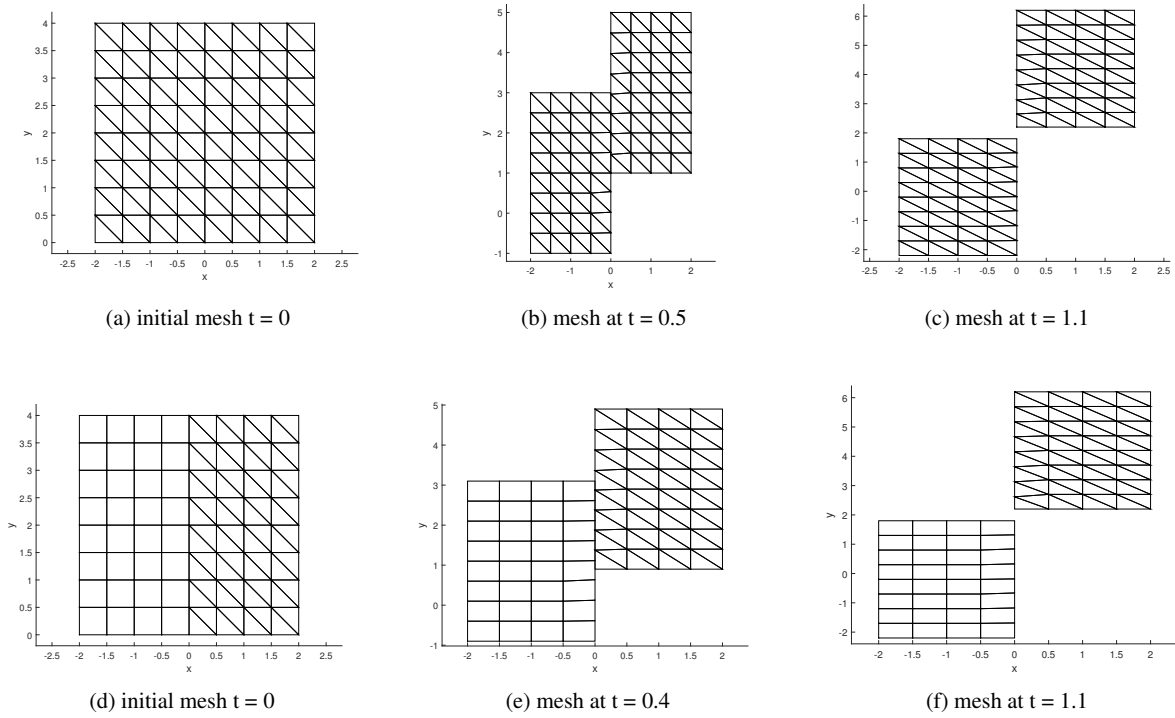


Fig. 7: Slide lines test case with initial condition as in equation (71). The mesh is moved with the local fluid velocity, which at $x = 0$ is discontinuous: so nodes over there are handled in a nonconforming way. At the top we show the results obtained employing a triangular mesh and at the bottom using a mesh made of both triangular and quadrilateral elements. We report the mesh at three different computational times: note that the computational domain can also be split in two non connected parts. The level of the water, the total area and the total volume are conserved at any time step, and the solution coincides with the exact one up to machine precision.

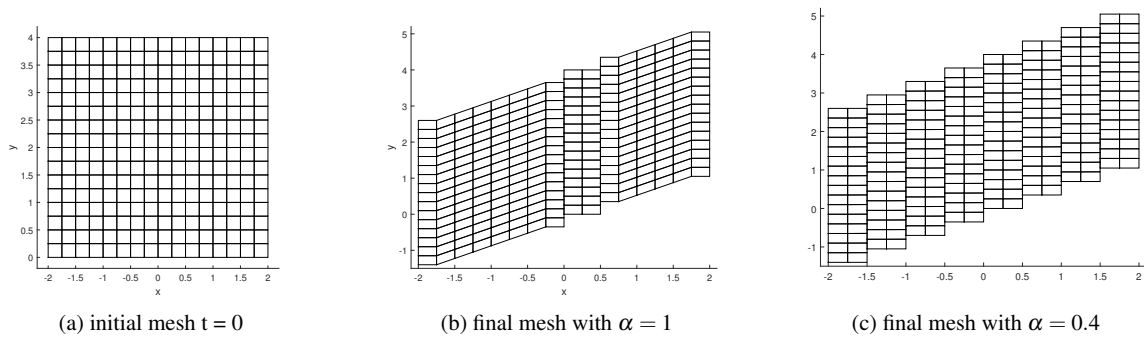


Fig. 8: Slide lines test case with initial condition as in equation (72). We start with a conforming quadrilateral mesh; using a value of $\alpha = 1$ in (40) we obtain only two slip-lines (at $x = 0$ and $x = 0.5$), whereas using $\alpha = 0.4$, which makes the detector more strict, the mesh slides along each straight line where the fluid velocity changes.

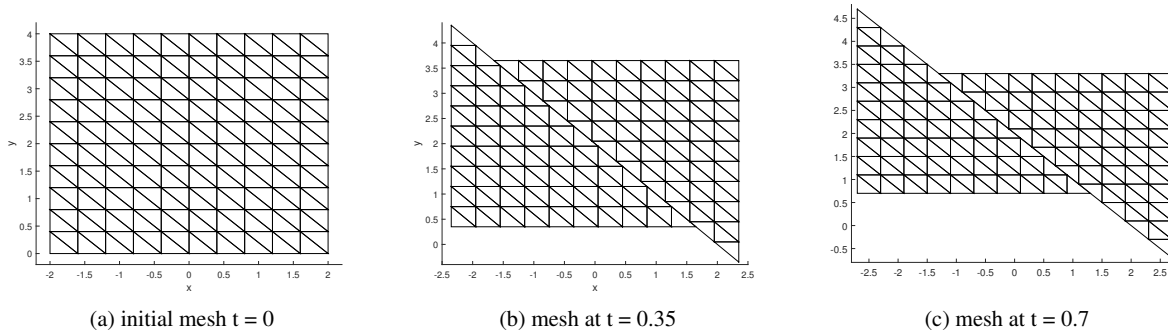


Fig. 9: Oblique slide line. We show the discretization of the computational domain at three different times. The corresponding numerical solution matches the exact one.

Table 1: Numerical convergence results for the passive transport of a Gaussian profile on moving nonconforming meshes. The error norms refer to the variable c at time $t = 0.5$. On the left we report the result for the first order method (i.e. without using the MUSCL-Hancock reconstruction procedure) and on the right using the second order accurate scheme.

$\mathcal{O}1$				$\mathcal{O}2$			
mesh points	$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$	mesh points	$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$
12×12	1.95E-01	1.44E-01	-	12×12	1.95E-01	4.96E-02	-
24×24	9.78E-02	7.58E-02	0.93	24×24	9.78E-02	1.23E-02	2.02
40×40	5.88E-02	4.69E-02	0.94	40×40	5.88E-02	4.24E-03	2.10
80×80	2.95E-02	2.41E-02	0.97	80×80	2.95E-02	1.01E-03	2.09
120×120	1.98E-02	1.62E-02	0.99	120×120	1.98E-02	4.51E-04	2.01

(61) with the addition of the concentration c of a passive tracer,

$$\mathbf{Q} = \begin{pmatrix} h \\ hu \\ hv \\ hc \end{pmatrix}, \quad \mathbf{f} = \begin{pmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \\ huc \end{pmatrix}, \quad \mathbf{g} = \begin{pmatrix} hv \\ huv \\ hv^2 + \frac{1}{2}gh^2 \\ hvc \end{pmatrix}. \quad (76)$$

We fix the following initial condition

$$\mathbf{Q}(\mathbf{x}, 0) = \begin{pmatrix} 1 \\ u \\ 0 \\ 1 + e^{\frac{-0.5(x^2 + (y-0.5p)^2)}{0.5^2}} \end{pmatrix}, \quad (77)$$

$$-2 \leq x \leq 2, \quad 0 \leq y \leq p,$$

where we use a fluid velocity of $u = 1$ and where we have taken the period $p = 4$. The mesh is moved with the velocity

$$\mathbf{V} = (0, 0.5 \text{floor}(x)) \quad -2 \leq x \leq 2, \quad (78)$$

according to the ALE framework, where the mesh velocity can be chosen arbitrarily. We prescribed periodic boundary conditions on the upper and lower side of the rectangular domain, and wall boundary conditions on the left and right sides. Since the exact solution is known ($\mathbf{Q}(\mathbf{x}, t) = \mathbf{Q}(\mathbf{x} - ut, 0)$) and it is smooth, we can verify the order of convergence of our method. In Table 1 we report the order of convergence of the basic first order Finite Volume method,

and of its second order extension that uses the MUSCL-Hancock strategy for the reconstruction procedure in space and time. Moreover, in Figure 12 we compare the numerical solution for the variable c with the profile of the exact solution and we show the mesh at the final time.

3.6.5 Steady vortex in equilibrium

To show that our method is also robust enough for vortex flows, we simulate the case of a steady vortex in equilibrium and we will compare the results obtained with our non-conforming method with a standard conforming algorithm (without any rezoning technique) looking at the differences after long simulation times.

Consider the shallow water in polar coordinates (69) and the following initial condition

$$h(r, \varphi, 0) = 1 - \frac{1}{2g} e^{-(r^2-1)}, \quad (79)$$

$$u_\rho(r, \varphi, 0) = 0, \quad u_\varphi(r, \varphi, 0) = r e^{-\frac{1}{2}(r^2-1)},$$

which is a stationary solution, and so coincides with the exact solution at any time. We performe our test both with the Osher-type and the Rusanov-type fluxes and with a mesh made of triangles, quadrilaterals or both. The considered computational domain is $\Omega(r, \varphi) = [0.2, 2] \times [0, 2\pi]$ which

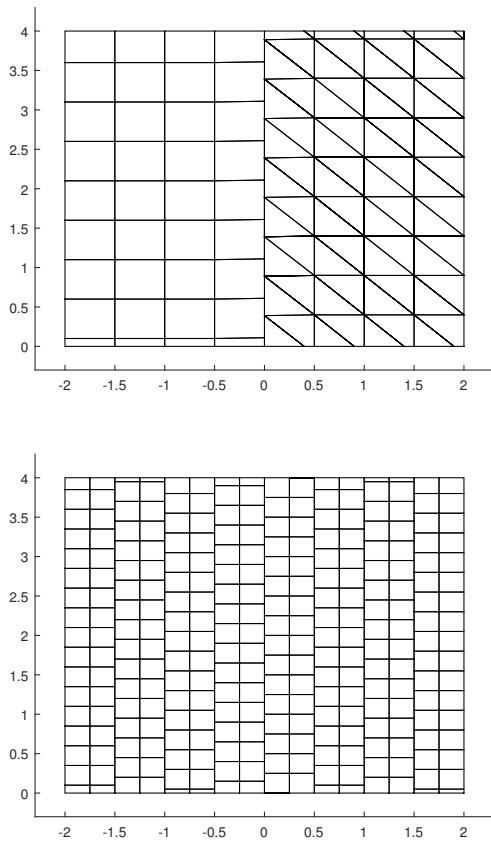


Fig. 10: Slide lines with periodic boundary conditions. We report the final computational domain at time $t = 100.2$ corresponding to the initial condition in (71) on the left, and the one corresponding to the initial condition in (72) on the right. No distortion of the computational domain appears neither at the interfaces, and the numerical solution coincides with the exact one.

is easily mapped into the annulus with radius $[0.2, 2]$. Indeed the choice of considering the shallow water equations in polar coordinates allows us to study the vortex over a rectangular domain with periodic boundary conditions (at $\varphi = 0$ and $\varphi = 2\pi$) instead of dealing with circles. At $r = 0.2$ and $r = 2$ we impose reflective boundary conditions. In particular using the polar coordinates the detected shear interfaces lie over straight lines and so they are perfectly handled by our algorithm. The images presented in this Section are then obtained by mapping back our results to Cartesian coordinates, as shown in Figure 13.

First, Table 2 confirms the designed order of convergence of our algorithm in multiple situations: so primarily we can say that the mesh motion does not affect the standard order of convergence of the MUSCL-Hancock strategy and moreover this shows once again that the numerical flux computation, even at the nonconforming interfaces, is car-

ried out correctly. The numerical solution at $t = 15$ is compared with the analytical one in Figure 14.

Then we compare the results with a standard conforming method. First, let us underline that when the velocity changes even within the same element the only way to overcome the mesh distortion would be to split the element itself. For this reason, where the velocity field changes smoothly and as a consequence the shear flow affects all the vertexes of the same element, at a certain time the mesh will become invalid even in the nonconforming case. This would not happen if the velocity field were uniform within each element, i.e. if each element moved all its vertexes with the same velocity, e.g. the velocity of the barycenter. The main difference between the new nonconforming algorithm and a conventional conforming method is the final time at which the computation stops due to an invalid mesh, and the time step restriction that depends on the smallest encircle diameter of the elements.

In Table 3 we report the employed number of time steps and their dimension for different kinds of meshes and at different times. We remark that a larger value of Δt decreases the required number of time steps and in this way also the total amount of computational time. The last results of each group refer to the moment at which the algorithm breaks due to an invalid mesh: one can easily see that the nonconforming method is able to run almost eight times longer than a conventional ALE method on conforming grids. Finally, looking at Figure 15 one can appreciate that the conforming method destroys the mesh immediately and then breaks, whereas the new nonconforming algorithm maintains a high quality mesh for a very long time, even with a very coarse mesh.

3.7 Proof of concepts: general slide lines

All test problems shown before were limited to logically straight slide lines. However, there is no intrinsic limitation to logically straight slide lines in our algorithm, since the integral space-time conservation form of the conservation law is valid for *arbitrary* closed space-time control volumes. This simple, elegant but at the same time very powerful formulation allows also to dynamically *add* and *remove* elements or to change their type during the simulation in a consistent manner that respects the GCL as well as local and global conservation. All these features are trivially built in *by construction*, due to the integral formulation on closed space-time control volumes. In Figure 16 we show examples of space-time control volumes that result when elements change type or when elements are dynamically added and removed during a simulation. For logically non-straight slide lines, it is necessary to divide elements and nodes into masters and slaves, where the master elements maintain their number of nodes, while the slave elements

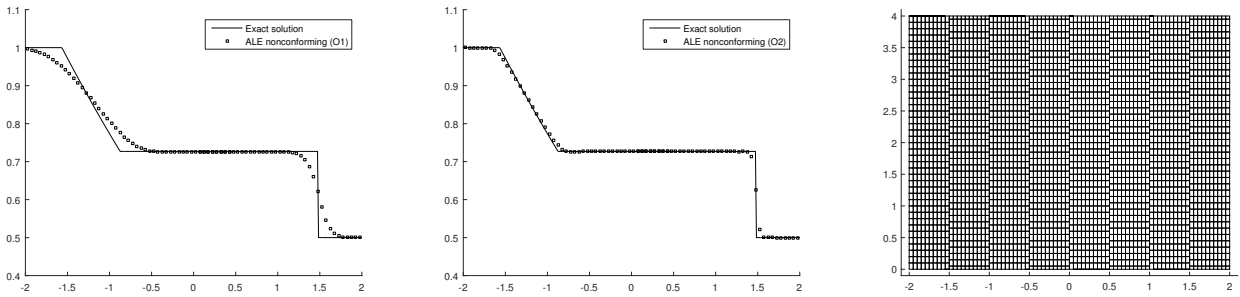


Fig. 11: Riemann problem with an arbitrary mesh velocity. Taking $\alpha = 0.4$ in (40) the algorithm identifies 7 interfaces which are then handled in a nonconforming way. In the figure we report the final discretization of the computational domain, and the comparison between the exact solution and the numerical solutions obtained with our nonconforming method showing first order results (left), second order results (center) and the mesh at the final time (right).

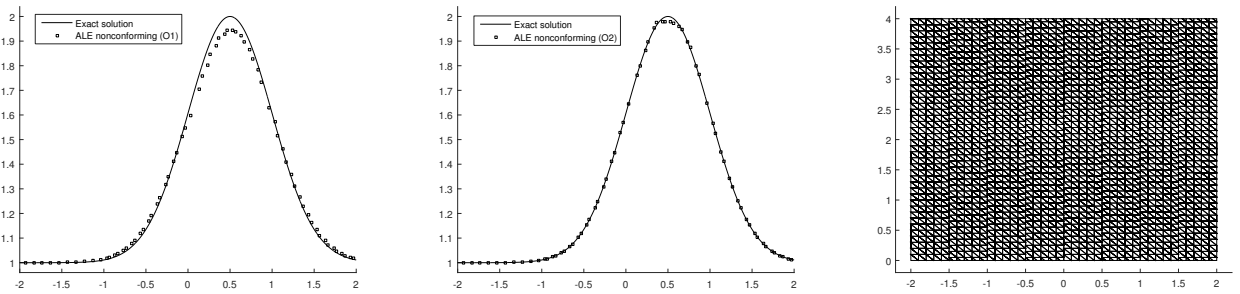


Fig. 12: Comparison of the exact solution for the quantity c with the numerical solution obtained on moving nonconforming meshes. The results obtained with the first order algorithm are shown on the left, while those obtained with the second order MUSCL-Hancock method are presented in the center. The comparison is done at time $t = 0.5$ taking a cut of the profile of c corresponding to $y = 2$. On the right we show the discretization of the computational domain at time $t = 0.5$.

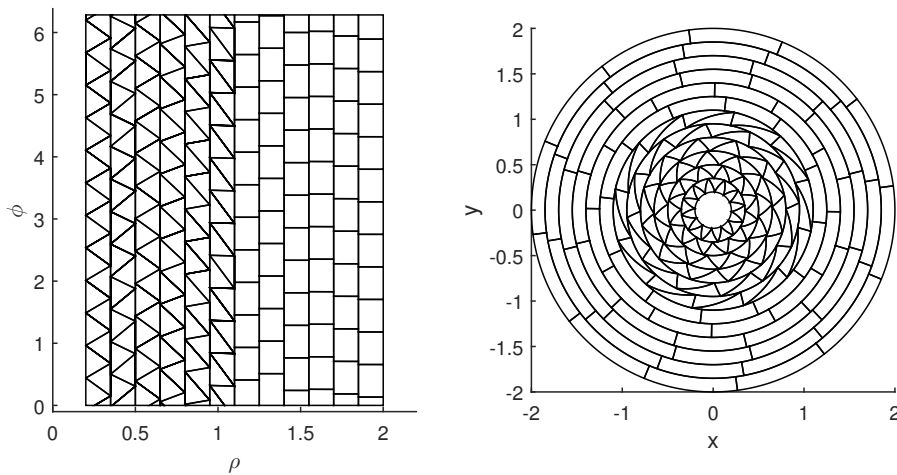


Fig. 13: Domain discretization at time $t = 15$. On the left we report the grid in polar coordinates where the shear discontinuities lie over straight lines. On the right the corresponding grid in Cartesian coordinates.

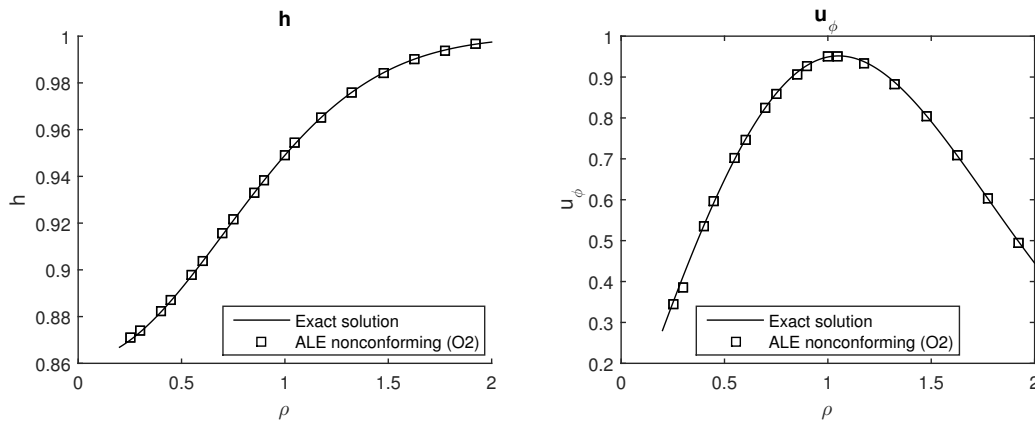


Fig. 14: Comparison between analytical solution and second-order accurate numerical results for the water level h (left) and the tangential component of the velocity u_ϕ (right), with $\phi = 2\pi$ and $r \in [0.2, 2]$.

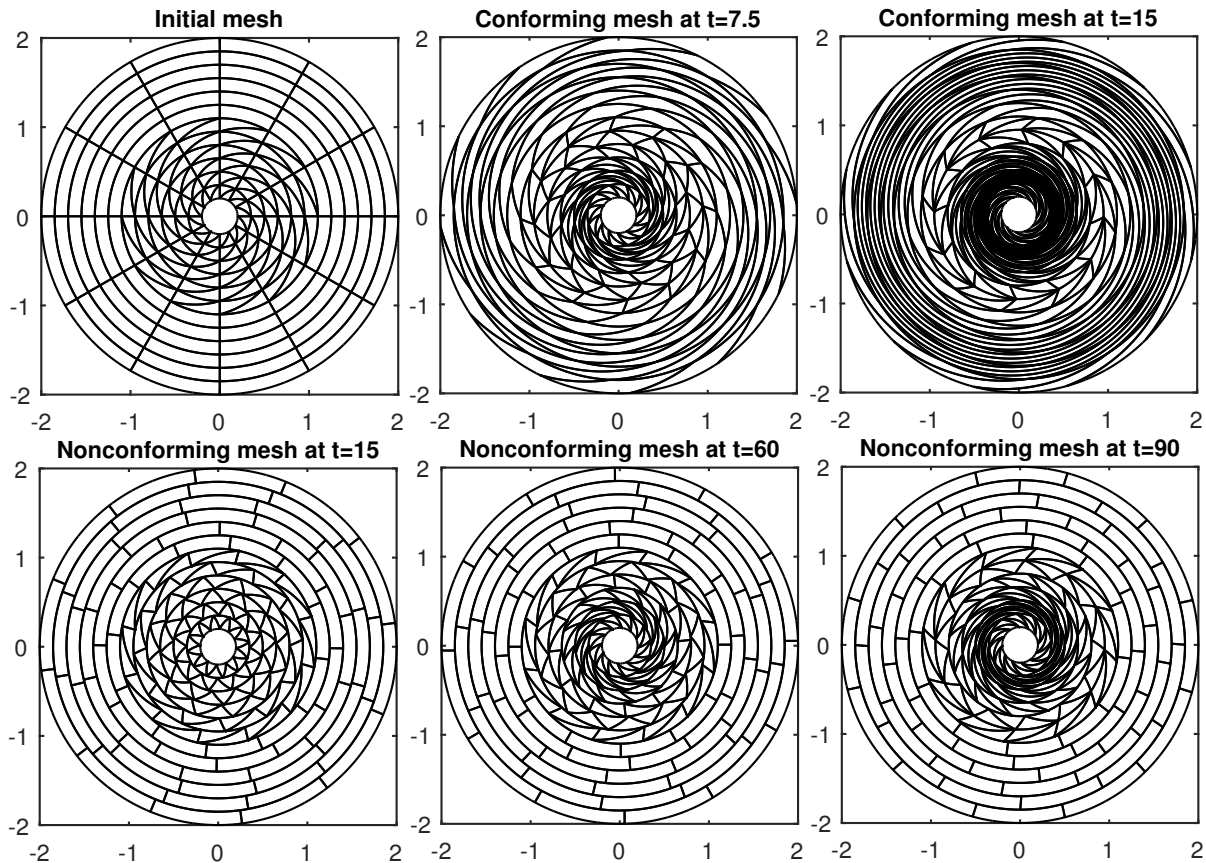


Fig. 15: Steady vortex in equilibrium. We compared the behavior of a standard conforming algorithm (without any rezoning technique) and of our new nonconforming method. Using the conforming algorithm the elements are deformed in a very short time, the time step is heavily reduced and hence the computation is slower. On the contrary, the nonconforming slide lines introduced by our scheme are able to maintain a good shape of each element and an almost constant time step for a long computational time. Indeed only at time $t = 90$ some elements with $r \rightarrow 0$ are deformed because of the presence of shear *inside* the elements, which could be remedied only by subdividing the elements themselves or by removing them.

Table 2: Numerical convergence results for the steady vortex in equilibrium using nonconforming meshes. In the left table we report the results obtained on a quadrilateral mesh using the Osher type flux. For the results on the right we have employed a triangular mesh and the Rusanov type flux. The error refers to the difference between the computed water level h and the exact one at time $t_f = 0.5$.

$\mathcal{O}2$, Osher flux, quadrilateral elements				$\mathcal{O}2$, Rusanov flux, triangular elements			
mesh points	$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$	mesh points	$h(\Omega(t_f))$	ϵ_{L_2}	$\mathcal{O}(L_2)$
12×12	2.33E-01	1.36E-03	-	20×20	7.18E-02	5.97E-04	-
24×24	1.17E-01	3.42E-04	1.99	30×30	5.21E-02	2.54E-04	2.11
32×32	8.74E-02	1.94E-04	1.97	40×40	3.91E-02	1.43E-04	2.01
44×44	6.36E-02	1.03E-04	1.98	55×55	2.84E-02	7.76E-05	1.91
60×60	4.66E-02	5.57E-05	1.99	60×60	2.60E-02	6.58E-05	1.91

Table 3: In this table we report the number of time steps n necessary to reach the time t and the dimension of the time step Δt at that time. We used three different meshes with N_E total number of elements (triangles or quadrilaterals). The results are obtained by applying a standard conforming method and our new nonconforming algorithm. Looking at the bold data one can see that with almost the same number of time steps one reaches a simulation time that is twice as large with the nonconforming algorithm compared to a classical conforming one. Besides the final simulation time that can be reached before obtaining an invalid mesh is almost 8 times larger.

$N_E \rightarrow$	216			264			300		
	conforming algorithm								
	t	n	Δt	t	n	Δt	t	n	Δt
	1	110	9.58E-03	1	180	5.40E-03	1	180	5.71E-03
	8	1163	4.13E-03	8	2180	2.52E-03	10	2071	3.11E-03
	12	2370	2.70E-03	12	4035	1.89E-03	15	4098	2.04E-03
stop at \rightarrow	15.3	3773	2.06E-03	15.5	6072	1.54E-03	17	5190	1.78E-03
	nonconforming algorithm								
	1	110	9.58E-03	1	180	5.82E-03	1	175	5.68E-03
	8	851	9.50E-03	8	1410	5.52E-03	10	1720	5.92E-03
	30	3175	9.30E-03	30	6033	4.06E-03	15	2565	5.94E-03
	60	7757	4.90E-03	60	15010	2.84E-03	80	15979	3.34E-03
stop at \rightarrow	119	26430	2.24E-03	129	35791	1.94E-03	132	36275	2.13E-03

must in general change their element type during the sliding process. Also note that master nodes are free to move anywhere, while slave nodes must slide along the master edges. Furthermore, small elements need to be removed if they lead to excessively small time steps due to the CFL condition.

We now repeat the same shallow water vortex test problem as described in the previous Section, but using the PDE in Cartesian coordinates. This leads to logically non-straight slide lines. The comparison between the classical conforming moving mesh algorithm and our new nonconforming approach is shown in Fig. 17 and Table 4. We observe the improved mesh quality and time step size compared to the classical conforming approach, in particular when the moving nonconforming mesh is combined with the removal of small elements. The obtained results look promising and justify further research in this direction in the future.

Table 4: Time step size for three different moving mesh algorithms. The main improvement is achieved when using a nonconforming algorithm combined with small element removal. This allows to maintain reasonable time steps also for longer simulation times.

time	Time step size		
	conforming	nonconforming	nonconforming + element removal
0.3	3.8E-3	3.2E-3	3.2E-2
0.6	3.6E-3	2.1E-3	2.1E-3
1.0	1.9E-3	9.0E-4	1.2E-3
1.3	5.8E-4	1.2E-4	1.4E-3
1.7	-	-	1.4E-3

4 Well balanced path-conservative methods - outline

This central part of the article is dedicated to the presentation of a new family of Well Balanced (WB) path-conservative schemes, able to maintain up to machine precision the equilibrium solutions of the studied system and to drastically increase the power of resolutions on small perturbations that may appear around those equilibria. In particular we show

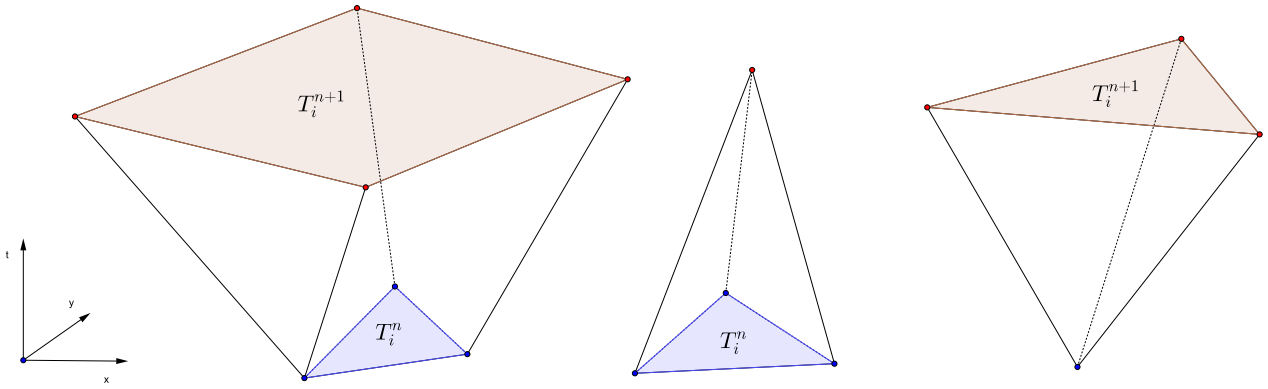


Fig. 16: Dynamic change of element type (left), element removal (center) and element insertion (right) between time t^n and time t^{n+1} . Nodes and element T_i^n at time t^n are highlighted in blue, nodes and element T_i^{n+1} at time t^{n+1} are colored in red.

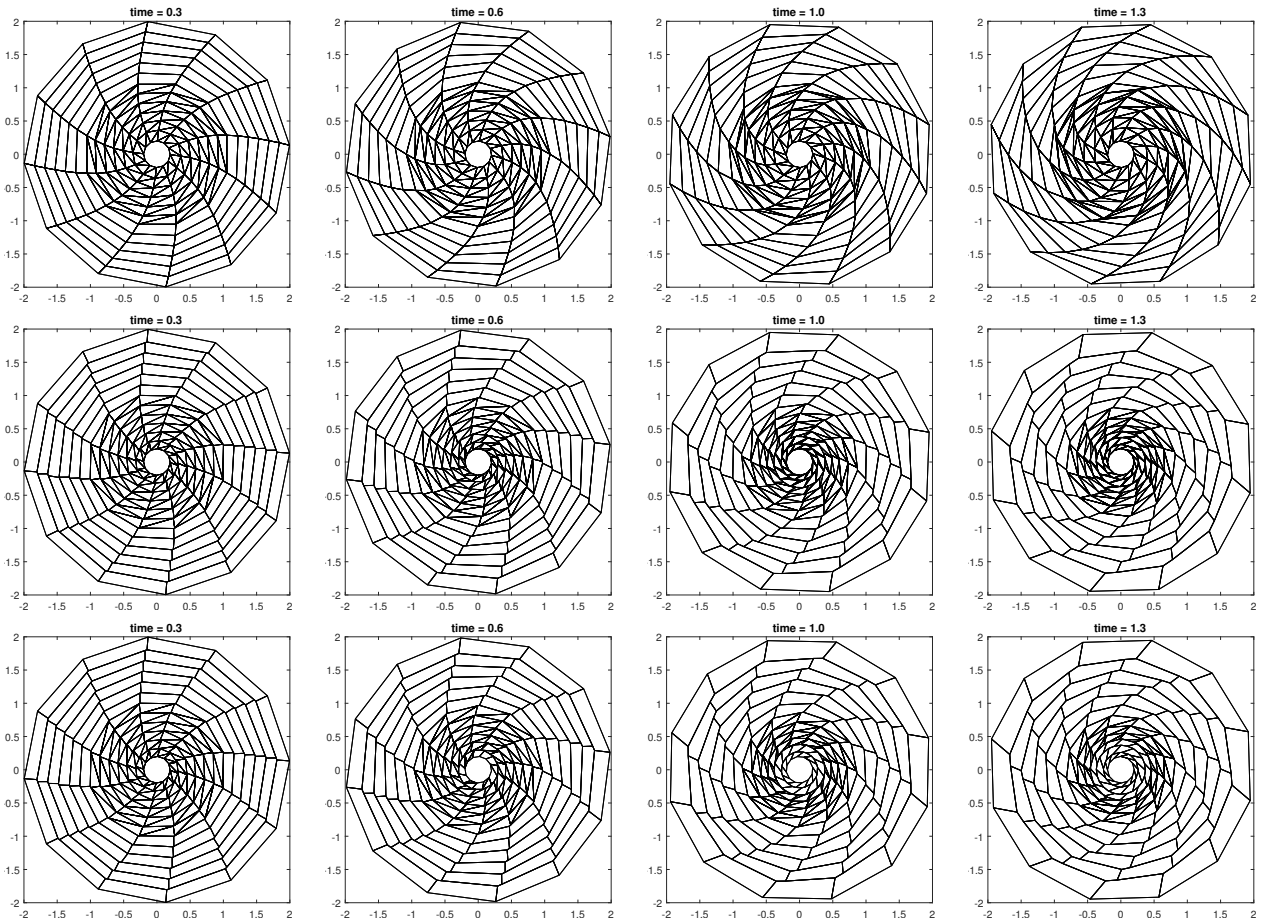


Fig. 17: Isolated vortex in Cartesian coordinates. Classical conforming algorithm without any rezoning technique (top). Moving mesh obtained with the new *nonconforming* algorithm at different times (center) without small element removal. Moving *nonconforming* mesh with small element removal (bottom), which allows to control the time step size and to maintain a better mesh quality. The nonconforming algorithms used here use logically *non-straight* slide lines. The sliding edges are automatically detected based on the tangential velocity difference.

that the little dissipative Osher scheme is modified in order to be well balanced for nontrivial equilibria on moving meshes, and we couple the WB technique with ALE for the Euler equations with gravity; moreover thanks to the use of a well balanced Osher scheme joint with the Lagrangian framework allows we are able to maintain exactly even *moving* equilibria. This topic is covered in the next two Sections.

In Section 5, we concentrate on the one-dimensional case. In Section 5.1 we recall the basic notions on well balanced path-conservative schemes introduced by Dal Maso et al. [120] and Parés et al. [131,31]. Then we have decided to detail our novel well balanced scheme in the particular case of the Euler equations of gas dynamics with gravity in polar coordinates, which are recovered in Section 5.2. Indeed, the method is based on very general and powerful funding concepts, but some computations differ from system to system and depend on the family of equilibria that has to be maintained, so in order to present all its specific features we have to introduce a concrete case. However, it is easy to reproduce the same reasoning for other systems following the presented line, as it is done for shallow water equations in Section 6.3.

Moreover, the Euler equations with gravity represent a very interesting and highly challenging case of study. They allow to explore complex *astrophysical* phenomena as the rotating *Keplerian disk*: it is characterized by a family of steady equilibria between pressure gradient, centrifugal force and gravity force and by a high shear flow due to the differential rotation in the disk. In order to model with high accuracy and minimal dissipation small perturbations around those equilibria for very long period of times we propose to use a well balanced method coupled with our nonconforming ALE framework, [78]. This coupling is presented in Section 6.1, together with a large set of numerical test cases in Section 6.2 that witnesses the power of resolution of our method both close and far away from the equilibria. Besides, a comparison with *not* well balanced schemes or *not* Lagrangian-type methods show the superiority of the proposed scheme with respect to the state of the art, demonstrated also by the comparison with the PLUTO code (a well established software targeting astrophysical simulations); we would like to emphasize the excellent quality of the results obtained in Section 6.2. Then we conclude by showing that the same coupling between ALE and WB can be applied to the shallow water equations, refer to Section 6.3.

To conclude, we would like to focus the attention on the key idea for the construction of our well balanced scheme: the introduction of a path which directly exploits the known stationary solution (and so it is exact on it), and treats in a approximate way only the fluctuations around the equilibrium, see equation (208). The same idea of using an approximate technique only on the fluctuations appears even in the re-

construction process, see equation (145)-(147). This simple idea can guide to the understanding of the method.

5 Well balancing: one dimensional case

5.1 General framework

For the numerical approximation of the a one dimensional system, the spatial domain is discretized by N fixed intervals $P_i^n = I_i = [x_{i-1/2}, x_{i+1/2}]$ of regular size $\Delta x = x_{i+1/2} - x_{i-1/2}$, $i = 1, \dots, N$. As a standard Finite Volume scheme, a path-conservative scheme is obtained first by integrating the governing PDE (1) in space and time. After that, instead of introducing the time-averaged flux, following [131] and [36] we write the scheme as follows

$$\begin{aligned} \frac{d\mathbf{Q}_i}{dt}(t) = & -\frac{\Delta t}{\Delta x} \left(\mathcal{D}_{i-\frac{1}{2}}^+ \left(\mathbf{q}_{i-\frac{1}{2}}^-(t), \mathbf{q}_{i-\frac{1}{2}}^+(t) \right) + \mathcal{D}_{i+\frac{1}{2}}^- \left(\mathbf{q}_{i+\frac{1}{2}}^-(t), \mathbf{q}_{i+\frac{1}{2}}^+(t) \right) \right) \\ & - \frac{\Delta t}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial}{\partial x} \mathbf{f}(\mathbf{q}_i(x,t)) dr \\ & - \frac{\Delta t}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{B}_1(\mathbf{q}_i(x,t)) \frac{\partial}{\partial x} (\mathbf{q}_i(x,t)) dr. \end{aligned} \quad (80)$$

In the scheme, $\mathbf{q}_i(x,t)$ is the approximation of the conserved variables inside cell I_i at time t , computed via a reconstruction operator from the conserved variables in I_i and its neighbors, while $\mathbf{q}_{i-\frac{1}{2}}^+(t) = \mathbf{q}_i(x_{i-1/2}, t)$ and $\mathbf{q}_{i+\frac{1}{2}}^-(t) = \mathbf{q}_i(x_{i+1/2}, t)$ denote the evaluation of $\mathbf{q}_i(x,t)$ at the left and right boundaries of cell I_i . According to the above reference, $\mathcal{D}_{i+\frac{1}{2}}^\pm$ is defined as follows

$$\begin{aligned} \mathcal{D}_{i+\frac{1}{2}}^\pm \left(\mathbf{q}_{i+\frac{1}{2}}^-, \mathbf{q}_{i+\frac{1}{2}}^+ \right) = & \frac{1}{2} \left(\mathbf{f}(\mathbf{q}_{i+\frac{1}{2}}^+) - \mathbf{f}(\mathbf{q}_{i+\frac{1}{2}}^-) + \right. \\ & \left. \mathcal{B}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right) \pm \mathcal{V}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right) \right), \end{aligned} \quad (81)$$

where $\mathbf{f}(\mathbf{q})$ is the physical flux, $\mathcal{B}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right)$ is the discretization of the non-conservative terms and $\mathcal{V}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right)$ is the viscosity term, that characterizes the method. In (81), the dependency on t has been dropped for simplicity.

$\mathcal{B}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right)$ and $\mathcal{V}_{i+\frac{1}{2}} \left(\mathbf{q}_{i+\frac{1}{2}}^+ - \mathbf{q}_{i+\frac{1}{2}}^- \right)$ have to be defined in terms of a family of paths $\Phi(s; \mathbf{q}_{i+\frac{1}{2}}^-, \mathbf{q}_{i+\frac{1}{2}}^+)$, $s \in [0, 1]$. In this work the family of paths will be chosen so that a class of stationary solutions could be preserved.

The scheme as written in (80) is similar to the wave-propagation formulation proposed by LeVeque in [104] for

standard conservative systems. We refer the reader interested in the complete theoretical framework of path-conservative schemes to the very detailed paper of Parés [131], especially to Chapter 3 where the classical Finite Volume scheme is first rewritten in terms of distributions and then generalized to non-conservative products. In particular it is shown that the terms $\mathcal{D}_{i+\frac{1}{2}}^\pm$ allow to decompose the punctual masses placed at the interface in such a way that $\mathcal{D}_{i+\frac{1}{2}}^-$ contributes to cell I_i and $\mathcal{D}_{i+\frac{1}{2}}^+$ to cell I_{i+1} , leading to the key definition (see equation (86)) for resolving the discontinuities at the interface. Moreover we refer to [48] for the definition of the non-conservative products as Borel measure.

For the sake of clarity let us now recall some definitions, taken from the above references.

Definition 5.11 (Family of paths) *A family of paths in $\Omega \subset \mathbb{R}^d$ is a locally Lipschitz map*

$$\Phi : [0, 1] \times \Omega \times \Omega \mapsto \Omega, \quad (82)$$

such that

- i. $\Phi(0; \mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{Q}_L$, $\Phi(1; \mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{Q}_R$, $\Phi(s; \mathbf{Q}, \mathbf{Q}) = \mathbf{Q}$, for any $\mathbf{Q}_L, \mathbf{Q}_R, \mathbf{Q} \in \Omega$;
- ii. for every arbitrary bounded set $\mathcal{O} \subset \Omega$, there exists a constant k such that

$$\left| \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L, \mathbf{Q}_R) \right| \leq k |\mathbf{Q}_R - \mathbf{Q}_L|, \quad (83)$$

for any $\mathbf{Q}_L, \mathbf{Q}_R \in \mathcal{O}$ and almost every $s \in [0, 1]$;

- iii. for every bounded set $\mathcal{O} \subset \Omega$, there exists a constant K such that

$$\left| \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L^1, \mathbf{Q}_R^1) - \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L^2, \mathbf{Q}_R^2) \right| \leq K (|\mathbf{Q}_L^1 - \mathbf{Q}_L^2| + |\mathbf{Q}_R^1 - \mathbf{Q}_R^2|), \quad (84)$$

for any $\mathbf{Q}_L^1, \mathbf{Q}_R^1, \mathbf{Q}_L^2, \mathbf{Q}_R^2 \in \mathcal{O}$ and almost every $s \in [0, 1]$.

Definition 5.12 (Path-conservative) *Given a family of path Φ , a numerical scheme is said to be Φ -conservative or path-conservative if it can be written under the form (80) with*

$$\mathcal{D}^\pm(\mathbf{Q}, \mathbf{Q}) = \mathbf{0} \quad \forall \mathbf{Q} \in \Omega, \quad (85)$$

and

$$\mathcal{D}^-(\mathbf{Q}_L, \mathbf{Q}_R) + \mathcal{D}^+(\mathbf{Q}_L, \mathbf{Q}_R) = \int_0^1 \mathbf{A}(\Phi(s; \mathbf{Q}_L, \mathbf{Q}_R)) \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L, \mathbf{Q}_R) ds, \quad (86)$$

for every $\mathbf{Q}_L, \mathbf{Q}_R \in \Omega$.

Note that, in the case of a one dimensional system i.e. (1) with $\mathbf{g}, \mathbf{h} = \mathbf{0}$ and $\mathbf{B}_{2,3} = \mathbf{0}$, we can rewrite equation (86) as

$$\mathcal{D}^-(\mathbf{Q}_L, \mathbf{Q}_R) + \mathcal{D}^+(\mathbf{Q}_L, \mathbf{Q}_R) = \mathbf{f}(\mathbf{Q}_R) - \mathbf{f}(\mathbf{Q}_L) + \mathcal{B}_{LR}(\mathbf{Q}_R - \mathbf{Q}_L), \quad (87)$$

where

$$\mathcal{B}_{LR}(\mathbf{Q}_R - \mathbf{Q}_L) = \int_0^1 \mathbf{B}_1(\Phi(s; \mathbf{Q}_L, \mathbf{Q}_R)) \frac{\partial \Phi}{\partial s}(s; \mathbf{Q}_L, \mathbf{Q}_R) ds. \quad (88)$$

Definition 5.13 (Well balance) *Let Γ be the set of all the integral curves gamma of a linearly degenerate field of $\mathbf{A}(\mathbf{Q})$ such that the corresponding eigenvalues vanishes on Γ . Given a curve $\gamma \in \Gamma$, a numerical scheme*

$$\mathbf{Q}_i^{n+1} = \mathbf{Q}_i^n + \frac{\Delta t}{\Delta x} \mathbf{H}(\mathbf{Q}_{i-q}^n, \dots, \mathbf{Q}_{i+p}^n) \quad (89)$$

is said to be exactly well balanced for γ if, given any \mathcal{C}^1 function $x \in (\alpha, \beta) \subset \mathbb{R} \mapsto \mathbf{Q}(x) \in \Omega$ such that

$$\mathbf{Q}(x) \in \gamma, \quad \forall x \in (\alpha, \beta), \quad (90)$$

and $p+q+1$ points in (α, β) such that $x_{-q} \leq \dots \leq x_p$, then

$$H(\mathbf{Q}(x_{-q}), \dots, \mathbf{Q}(x_p)) = 0. \quad (91)$$

Proposition 5.14 *A first order path-conservative scheme is exactly well balanced for a curve $\gamma \in \Gamma$ if and only if*

$$D^\pm(\mathbf{Q}_L, \mathbf{Q}_R) = 0, \quad \forall \mathbf{Q}_L, \mathbf{Q}_R \in \gamma. \quad (92)$$

Proposition 5.15 *A reconstruction operator \mathcal{P} is said to be exactly well balanced for a curve $\gamma \in \Gamma$ if, given a sequence $\{\mathbf{Q}_i\} \in \gamma$, it satisfies*

$$\mathcal{P}(x, \mathbf{Q}_{i-l}, \dots, \mathbf{Q}_{i+r}) \in \gamma \quad \forall x \in [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}], \quad (93)$$

for every i .

Proposition 5.16 *The numerical scheme in (80) is exactly well balanced for a curve $\gamma \in \Gamma$ if both the underlying first order scheme and the reconstruction operator are exactly well-balanced for γ .*

This part was to introduce the basic notions on well balanced path-conservative schemes for general systems. From now on, we will focus on a specific set of equations and on a family of equilibria. However, we would like to underline once again that all the methods we will propose, for example in Section 5.3 in the particular case of the Euler equations with gravity in polar coordinates, are based on very general and powerful concepts so that they can be easily extended to other systems.

5.2 Euler equations with gravity

The Euler equations of compressible gas dynamics with an externally given gravitational field allow to study problems in computational astrophysics connected with the rotation of gas clouds around a central object. In particular, we are interested in situations close to equilibrium solutions and affected by strong shear flows. What we propose is to use the nonconforming ALE technique presented in the previous Section 3 in order to maintain a good quality mesh despite the differential rotation, and a well balanced method in order to preserve exactly the steady states and to reduce the numerical errors close to them. But in order to apply these two techniques *coupled together* we have first to recover a suitable formulation of the equations.

In this Section we present the classical Cartesian form of the Euler equations with gravity (Section 5.2.1), then we rewrite them in polar coordinates (Section 5.2.2) so that the nonconforming ALE can be used in the more simple case of straight slip lines. And finally, in Section 5.2.3 we eliminate the algebraic source terms, which would make the equilibria preservation very difficult, by introducing some non-conservative products. This last non-conservative formulation (102) represents a very useful framework for applying our *new* well balanced techniques.

5.2.1 Cartesian coordinates

The Euler equations with gravity in two space dimensions represent a strongly hyperbolic system of equations that can be cast in the form of balance laws, by taking in (1)

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u_x \\ \rho u_y \\ \rho E \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} \rho u_x \\ \rho u_x^2 + p \\ \rho u_x u_y \\ u_x(\rho E + p) \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho u_y \\ \rho u_x u_y \\ \rho u_y^2 + p \\ u_y(\rho E + p) \end{pmatrix},$$

$$\mathbf{B}(\mathbf{Q}) = 0, \quad \mathbf{S}(\mathbf{Q}) = \begin{pmatrix} 0 \\ -\cos \varphi \rho \frac{Gm_s}{r^2} \\ -\sin \varphi \rho \frac{Gm_s}{r^2} \\ -(u_x \cos \varphi + u_y \sin \varphi) \rho \frac{Gm_s}{r^2} \end{pmatrix}. \quad (94)$$

Here ρ is the density, u_x and u_y are respectively the velocities along the x and y directions, $r = \sqrt{x^2 + y^2}$, $\varphi = \arctan(y/x)$, E is the specific total energy (excluding the gravitational energy), m_s is the mass of the central object, G is the gravitational constant and the pressure p is given by

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \rho (u_x^2 + u_y^2) \right), \quad \gamma = \frac{c_p}{c_v} > 1, \quad (95)$$

where γ is the ratio of the specific heats at constant pressure and at constant volume, and which is supposed to be constant.

5.2.2 Polar coordinates

Now we are interested in studying *rotational phenomena* affected by sheared vortex flows, so we decide to rewrite the Euler equations in polar coordinates (r, φ) . We follow the same procedure of Section 3.5.2. To shorten the notation, from now on when referring to the Euler equations in polar coordinates, we denote the radial velocity u_r by u , and the angular velocity u_φ by v . The resulting hyperbolic system still takes the form (1) with

$$\mathbf{Q} = \begin{pmatrix} r\rho \\ r\rho u \\ r\rho v \\ r\rho E \\ r \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} r\rho u \\ r\rho u^2 + rp \\ r\rho uv \\ ru(\rho E + p) \\ 0 \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ 0 \end{pmatrix}, \quad (96)$$

$$\mathbf{B}(\mathbf{Q}) = 0, \quad \mathbf{S}(\mathbf{Q}) = \begin{pmatrix} 0 \\ -\rho \frac{Gm_s}{r} + p + \rho v^2 \\ -\rho uv \\ -\rho u \frac{Gm_s}{r} \\ 0 \end{pmatrix}.$$

Note that the system is written in terms of conserved variables, which is made possible by the insertion of an additional trivial equation as in (67)

$$\frac{\partial r}{\partial t} = 0, \quad (97)$$

which implies that the radius r is both a coordinate and a conserved variable.

5.2.3 Non-conservative formulation and equilibrium solutions

The aim of our work is to construct a Finite Volume scheme that is second order accurate in general situations, and, at the same time, can solve exactly (i.e. up to machine precision) a class of stationary solutions given by

$$\rho = \rho(r), \quad u = 0, \quad \frac{\partial v}{\partial \varphi} = 0. \quad (98)$$

Looking at the second equation in (96) and at the equilibrium constraints in (98), we notice that equilibria should balance the pressure and gravitational forces. More precisely

$$\frac{\partial rp}{\partial r} = -\rho \left(\frac{Gm_s}{r} - v^2 \right) + p. \quad (99)$$

This relation has to be achieved also at the discrete level in order to preserve these stationary solutions. In standard Finite Volume schemes, fluxes and sources are typically discretized in different ways and therefore, the balancing between them is usually lost. In order to construct a numerical scheme that exactly preserves those stationary solutions, here we first rewrite the equations in the following way, where both, pressure and gravitational forces (99) are treated as non-conservative terms. Thus, by exploiting some trivial equalities as

$$\frac{\partial r p}{\partial r} = p + r \frac{\partial p}{\partial r} \quad \text{and} \quad \frac{\partial r}{\partial r} = 1, \quad (100)$$

the forces in (99) can be rearranged as

$$r \frac{\partial p}{\partial r} + \left(\rho \frac{Gm_s}{r} - \rho v^2 \right) \frac{\partial r}{\partial r} = 0, \quad (101)$$

and finally the Euler equations with gravity in polar coordinates can be cast in form (1) with non trivial non-conservative terms and with zero algebraic source term as

$$\mathbf{Q} = \begin{pmatrix} r\rho \\ r\rho u \\ r\rho v \\ r\rho E \\ r \end{pmatrix}, \quad \mathbf{f}(\mathbf{Q}) = \begin{pmatrix} r\rho u \\ r\rho u^2 \\ r\rho uv \\ ru(\rho E + p) \\ 0 \end{pmatrix}, \quad \mathbf{g}(\mathbf{Q}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(\rho E + p) \\ 0 \end{pmatrix},$$

$$\mathbf{S}(\mathbf{Q}) = \mathbf{0}, \quad \mathbf{B}(\mathbf{Q}) \cdot \nabla \mathbf{Q} = \begin{pmatrix} 0 \\ r \frac{\partial p}{\partial r} + \left(\rho \frac{Gm_s}{r} - \rho v^2 \right) \frac{\partial r}{\partial r} \\ (\rho uv) \frac{\partial r}{\partial r} \\ \rho u \frac{Gm_s}{r} \frac{\partial r}{\partial r} \\ 0 \end{pmatrix}, \quad (102)$$

i.e.

$$\mathbf{B}_1 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ r \frac{\partial p}{\partial q_1} & r \frac{\partial p}{\partial q_2} & r \frac{\partial p}{\partial q_3} & r \frac{\partial p}{\partial q_4} & r \frac{\partial p}{\partial q_5} + \rho \frac{Gm_s}{r} - \rho v^2 \\ 0 & 0 & 0 & 0 & \rho uv \\ 0 & 0 & 0 & 0 & \rho u \frac{Gm_s}{r} \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad \mathbf{B}_2 = \mathbf{0}, \quad (103)$$

where q_i , $i = 1, \dots, 5$ denotes the i -th component of vector \mathbf{Q} . Notice that it is possible to write the source terms as non-conservative products thanks to the introduction of the coordinate r also as conserved variables (see the added equation in (97)), which is the typical strategy adopted in [87, 86, 84, 85, 34].

5.3 WB for the Euler equations with gravity

In this Section we focus on the one dimensional version of the previous system (102), i.e. \mathbf{g} and \mathbf{B}_2 are not considered, and we achieve an exact balancing in the radial direction r (we use r instead of x to indicate the spatial domain). Then in Section 6.1, we will extend the method to two space dimensions and moving nonconforming meshes. In both cases the key point of our new numerical method is the discretization of the terms in (101).

The rest of this Section is organized as follows: we start by proposing two different first order well balanced schemes, the first one is named as Osher-Romberg scheme, and the second one is a well balanced HLL scheme. Next we propose a second order scheme constructed using the previous first order schemes in combination with a second order well balanced reconstruction operator.

5.4 First order well balanced schemes

Let us remark first, that the scheme (80) reduces to

$$\frac{d\mathbf{Q}_i}{dt}(t) = -\frac{\Delta t}{\Delta r} \left(\mathcal{D}_{i-\frac{1}{2}}^+ \left(\mathbf{q}_{i-\frac{1}{2}}^-(t), \mathbf{q}_{i-\frac{1}{2}}^+(t) \right) + \mathcal{D}_{i+\frac{1}{2}}^- \left(\mathbf{q}_{i+\frac{1}{2}}^-(t), \mathbf{q}_{i+\frac{1}{2}}^+(t) \right) \right), \quad (104)$$

if $\mathbf{q}_i(r, t) = \mathbf{Q}_i(t)$ is constant within each cell, for every time t and coincides with the cell average $\mathbf{Q}_i(t)$. The time derivative is discretized by the first order explicit Euler method. Thus, the resulting scheme will be first order accurate in space and time. Moreover, $\mathbf{q}_{i+\frac{1}{2}}^- = \mathbf{q}_i = \mathbf{Q}_i$ and $\mathbf{q}_{i+\frac{1}{2}}^+ = \mathbf{q}_{i+1} = \mathbf{Q}_{i+1}$.

Therefore, to determine the numerical scheme we should define $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ and $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$. In order to define $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$, a family of paths should be prescribed, so that the resulting scheme is well balanced for (98)-(99). Note that if the standard segment path is prescribed, that is

$$\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1}) = \mathbf{q}_i + s(\mathbf{q}_{i+1} - \mathbf{q}_i), \quad (105)$$

then, the resulting scheme is not well balanced for our set of stationary solutions.

Here we propose the following *family of paths*. Let $\Phi^E(s; \mathbf{Q}_i^E, \mathbf{Q}_{i+1}^E)$ be a reparametrization of a stationary solution given by (98)-(99) that connects the state \mathbf{Q}_i^E with \mathbf{Q}_{i+1}^E , where \mathbf{Q}_i^E is the cell average of the given stationary solution in the cell I_i . Note that in the case of first and second order schemes \mathbf{Q}_i^E could be approximated by the evaluation of the stationary solution at the center of the cell. Then we define $\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1})$ as follows

$$\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1}) = \Phi^E(s; \mathbf{Q}_i^E, \mathbf{Q}_{i+1}^E) + \Phi^f(s; \mathbf{q}_i^f, \mathbf{q}_{i+1}^f), \quad (106)$$

where $\mathbf{q}_i^f = \mathbf{q}_i - \mathbf{Q}_i^E$ and $\mathbf{q}_{i+1}^f = \mathbf{q}_{i+1} - \mathbf{Q}_{i+1}^E$ and

$$\Phi^f(s; \mathbf{q}_i^f, \mathbf{q}_{i+1}^f) = \mathbf{q}_i^f + s(\mathbf{q}_{i+1}^f - \mathbf{q}_i^f). \quad (107)$$

That is, Φ^f is a segment path on the *fluctuations* with respect to a given stationary solution. With this choice, it is clear that if \mathbf{q}_i and \mathbf{q}_{i+1} lie on the same stationary solution satisfying (98)-(99), then $\mathbf{q}_i^f = \mathbf{q}_{i+1}^f = \mathbf{0}$ and Φ reduces to Φ^E . In such situations we have that $\mathbf{f}(\mathbf{q}_{i+1}) = \mathbf{f}(\mathbf{q}_i) = \mathbf{0}$ and

$$\begin{aligned} \mathcal{B}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) \\ = \int_0^1 \mathbf{B}_1(\Phi^E(s, \mathbf{q}_i, \mathbf{q}_{i+1})) \frac{\partial \Phi^E}{\partial s}(s; \mathbf{q}_i, \mathbf{q}_{i+1}) ds = \mathbf{0}. \end{aligned} \quad (108)$$

Therefore

$$\mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \mathbf{0}. \quad (109)$$

For the sake of simplicity, in the following we will use the notation $\Phi(s)$ instead of $\Phi(s; \mathbf{q}_i, \mathbf{q}_{i+1})$ when there is no confusion.

Let us now define $\mathcal{B}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ in the general case, where \mathbf{q}_{i+1} and \mathbf{q}_i do not lie on a stationary solution. In this case we have that

$$\mathcal{B}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \left(b_1^{i+\frac{1}{2}} b_2^{i+\frac{1}{2}} b_3^{i+\frac{1}{2}} b_4^{i+\frac{1}{2}} b_5^{i+\frac{1}{2}} \right)^T. \quad (110)$$

It is clear from the definition of \mathbf{B}_1 that

$$b_1^{i+\frac{1}{2}} = b_5^{i+\frac{1}{2}} = 0, \quad (111)$$

$$b_2^{i+\frac{1}{2}} = \int_0^1 \Phi_r(s) \frac{\partial \Phi_p}{\partial s}(s) + \Phi_{(rp)}(s) \Phi_{\zeta_r}(s) \frac{\partial \Phi_r}{\partial s}(s) ds, \quad (112)$$

where $\Phi_r(s) = \Phi_r(s; r_i, r_{i+1}) = r_i + s(r_{i+1} - r_i)$, $\Phi_p(s) = \Phi_p^E(s) + \Phi_p^f(s)$, $\Phi_{(rp)}(s) = \Phi_{(rp)}^E(s) + \Phi_{(rp)}^f(s)$ and, finally, $\Phi_{\zeta_r}(s) = \Phi_{\zeta_r}^E(s) + \Phi_{\zeta_r}^f(s)$ where

$$\zeta_r(r) = \left(\frac{Gm_s}{r^2} - \frac{v^2}{r} \right), \quad \text{with } \zeta(r) = \int \zeta_r(r) dr. \quad (113)$$

Taking into account that

$$\int_0^1 \Phi_r(s) \frac{\partial \Phi_p^E}{\partial s}(s) + \Phi_{(rp)}^E(s) \Phi_{\zeta_r}^E(s) \frac{\partial \Phi_r}{\partial s}(s) ds = 0, \quad (114)$$

$b_2^{i+\frac{1}{2}}$ can be rewritten as follows:

$$\begin{aligned} b_2^{i+\frac{1}{2}} &= \int_0^1 \Phi_r(s) \frac{\partial \Phi_p^f}{\partial s}(s) ds \\ &+ \int_0^1 \left(\Phi_{(rp)}^E(s) \Phi_{\zeta_r}^f(s) + \Phi_{(rp)}^f(s) \Phi_{\zeta_r}(s) \right) \frac{\partial \Phi_r}{\partial s}(s) ds. \end{aligned} \quad (115)$$

Note that, $\frac{\partial \Phi_p^f}{\partial s}(s) = p_{i+1}^f - p_i^f$ and $\frac{\partial \Phi_r}{\partial s}(s) = r_{i+1} - r_i = \Delta r_{i+\frac{1}{2}}$. Observe that in uniform meshes $\Delta r_{i+\frac{1}{2}} = \Delta r$. With the previous notation $b_2^{i+\frac{1}{2}}$ reduces to

$$\begin{aligned} b_2^{i+\frac{1}{2}} &= r_{i+\frac{1}{2}} \Delta p_{i+\frac{1}{2}}^f \\ &+ \left(\int_0^1 \left(\Phi_{(rp)}^E(s) \Phi_{\zeta_r}^f(s) + \Phi_{(rp)}^f(s) \Phi_{\zeta_r}(s) \right) ds \right) \Delta r_{i+\frac{1}{2}}, \end{aligned} \quad (116)$$

where $r_{i+\frac{1}{2}} = \frac{r_i + r_{i+1}}{2}$ and $\Delta p_{i+\frac{1}{2}}^f = p_{i+1}^f - p_i^f$. In general, the integral term could be difficult to compute, therefore we propose to use a numerical quadrature formula. Here the mid-point rule is used. In this case, we define $b_2^{i+\frac{1}{2}}$ as follows:

$$\begin{aligned} b_2^{i+\frac{1}{2}} &= \left((r\rho)_{i+\frac{1}{2}}^E (\zeta_r)_{i+\frac{1}{2}}^f + (r\rho)_{i+\frac{1}{2}}^f (\zeta_r)_{i+\frac{1}{2}} \right) \Delta r_{i+\frac{1}{2}} \\ &+ r_{i+\frac{1}{2}} \Delta p_{i+\frac{1}{2}}^f, \end{aligned} \quad (117)$$

where

$$(r\rho)_{i+\frac{1}{2}}^E = \Phi_{(rp)}^E(1/2), \quad (\zeta_r)_{i+\frac{1}{2}}^f = \frac{(\zeta_r^f)_i + (\zeta_r^f)_{i+1}}{2}, \quad (118)$$

$$(r\rho)_{i+\frac{1}{2}}^f = \frac{(r\rho)_i^f + (r\rho)_{i+1}^f}{2}, \quad \text{and } (\zeta_r)_{i+\frac{1}{2}} = \Phi_{\zeta_r}(1/2). \quad (119)$$

It is clear from the definition that $b_2^{i+\frac{1}{2}} = 0$ if \mathbf{q}_i and \mathbf{q}_{i+1} lie on the same stationary solution as $\Delta p_{i+\frac{1}{2}}^f = 0$, $(r\rho)_{i+\frac{1}{2}}^f = 0$ and $(\zeta_r)_{i+\frac{1}{2}}^f = 0$. Finally, terms $b_3^{i+\frac{1}{2}}$ and $b_4^{i+\frac{1}{2}}$ could be approximated in the same way. Nevertheless, as those terms explicitly depend on u and we are interested in preserving equilibria with $u = 0$, a simpler approach can be used. Thus, $b_3^{i+\frac{1}{2}}$ is defined as

$$b_3^{i+\frac{1}{2}} = \frac{(r\rho u)_{i+\frac{1}{2}}}{r_{i+\frac{1}{2}}} v_{i+\frac{1}{2}} \Delta r_{i+\frac{1}{2}}, \quad (120)$$

where

$$(r\rho u)_{i+\frac{1}{2}} = \frac{(r\rho u)_i + (r\rho u)_{i+1}}{2}, \quad v_{i+\frac{1}{2}} = \frac{v_i + v_{i+1}}{2}, \quad (121)$$

and $b_4^{i+\frac{1}{2}}$ as

$$b_4^{i+\frac{1}{2}} = (r\rho u)_{i+\frac{1}{2}} \frac{Gm_s}{r_{i+\frac{1}{2}}^2} \Delta r_{i+\frac{1}{2}}. \quad (122)$$

Note that both terms vanish when $u = 0$.

As pointed in [131] and in Proposition 5.14, a sufficient condition for a first order path-conservative scheme to be well balanced is that $\mathcal{D}_{i+\frac{1}{2}}^\pm(\mathbf{q}_i, \mathbf{q}_{i+1}) = \mathbf{0}$, if \mathbf{q}_i and \mathbf{q}_{i+1} lie on the same stationary solution. Therefore, with the previous choice of paths, $\mathcal{D}_{i+\frac{1}{2}}^\pm = 0$ if $\mathcal{V}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \mathbf{0}$. In the next paragraph we are going to present two different schemes defined in terms of two different viscosity, both of them verifying that $\mathcal{V}_{i+\frac{1}{2}}(\mathbf{q}_{i+1} - \mathbf{q}_i) = 0$ for stationary solutions (98)-(99).

Osher-Romberg scheme. A path-conservative Osher-type scheme following [67,68,32] can be cast in form (81) with $\mathcal{V}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ being defined as follows

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \int_0^1 |\mathbf{A}(\Phi(s))| \partial_s \Phi(s) ds, \quad 0 \leq s \leq 1, \quad (123)$$

with $|\mathbf{A}| = \mathbf{R}|\Lambda|\mathbf{R}^{-1}$ being the usual definition of the matrix absolute value operator given in terms of the right eigenvector matrix \mathbf{R} , its inverse \mathbf{R}^{-1} and the diagonal matrix of the absolute values of the eigenvalues $|\Lambda| = \text{diag}(|\lambda_1|, |\lambda_2|, \dots, |\lambda_\nu|)$. For the numerical approximation of the viscosity matrix, first we notice that it can be written as

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \int_0^1 \text{sign}(\mathbf{A}(\Phi(s))) \mathbf{A}(\Phi(s)) \partial_s \Phi(s) ds, \quad (124)$$

with $\text{sign}(\mathbf{A}) = \mathbf{R} \text{sign}(\Lambda) \mathbf{R}^{-1}$ and $\text{sign}(\Lambda)$ the diagonal matrix containing the signs of all eigenvalues of \mathbf{A} . Then, we approximate the previous expression by a quadrature formula as follows:

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \sum_{j=1}^l \omega_j \text{sign}(\mathbf{A}(\Phi(s_j))) \mathbf{A}(\Phi(s_j)) \partial_s \Phi(s_j). \quad (125)$$

Now, we propose to approximate $\mathbf{A}(\Phi(s_j)) \partial_s \Phi(s_j)$ by the following expression:

$$\mathbf{A}(\Phi(s_j)) \partial_s \Phi(s_j) \approx \frac{\mathbf{A}_{\Phi_j}}{2\varepsilon_j} (\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)), \quad (126)$$

where $\mathbf{A}_{\Phi_j} = \mathbf{A}(\Phi(s_j - \varepsilon_j), \Phi(s_j + \varepsilon_j))$ is a Roe-matrix associated to the system (see [131] for details), that is a matrix satisfying

$$\begin{aligned} \mathbf{A}_{\Phi_j} (\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)) &= \mathbf{f}(\Phi(s_j + \varepsilon_j)) - \mathbf{f}(\Phi(s_j - \varepsilon_j)) \\ &+ \mathcal{B}_{\Phi_j} (\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)), \end{aligned} \quad (127)$$

where $\mathcal{B}_{\Phi_j} (\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j))$ is defined as in the previous Section using the states $\Phi(s_j - \varepsilon)$ and $\Phi(s_j + \varepsilon)$. Therefore, the viscosity term reads as follows

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \sum_{j=1}^l \omega_j \text{sign}(\mathbf{A}(\Phi(s_j))) \frac{\mathcal{R}_j}{2\varepsilon_j}, \quad (128)$$

where

$$\begin{aligned} \mathcal{R}_j &= \mathbf{f}(\Phi(s_j + \varepsilon_j)) - \mathbf{f}(\Phi(s_j - \varepsilon_j)) \\ &+ \mathcal{B}_{\Phi_j} (\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)). \end{aligned} \quad (129)$$

Note that if \mathbf{q}_i and \mathbf{q}_{i+1} lie on the same stationary solution we have $\Phi(s) = \Phi^E(s)$ and $\mathcal{R}_j = \mathbf{0}$, $j = 1, \dots, l$ and

$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ vanishes. Therefore, the numerical scheme (104) with (81), where $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is defined as (110), (111), (117), (120) and (122) and $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is defined by (128) is exactly well balanced for stationary solutions given by (98)-(99).

Here we propose the Romberg method with $l = 3$ and

$$\begin{aligned} s_1 &= 1/4, \quad s_2 = 3/4, \quad s_3 = 1/2, \\ \omega_1 &= 2/3, \quad \omega_2 = 2/3, \quad \omega_3 = -1/3, \\ \varepsilon_1 &= 1/4, \quad \varepsilon_2 = 1/4, \quad \varepsilon_3 = 1/2. \end{aligned} \quad (130)$$

With this choice, the viscosity term $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ of the Osher-Romberg method reads as follows

$$\begin{aligned} \mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) &= \\ &\frac{4}{3} \text{sign}(\mathbf{A}(\Phi(1/4))) (\mathbf{f}(\Phi(1/2)) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+1/4}(\Phi(1/2) - \mathbf{q}_i)) \\ &\frac{4}{3} \text{sign}(\mathbf{A}(\Phi(3/4))) (\mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\Phi(1/2)) + \mathcal{B}_{i+3/4}(\mathbf{q}_{i+1} - \Phi(1/2))) \\ &- \frac{1}{3} \text{sign}(\mathbf{A}(\Phi(1/2))) (\mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)). \end{aligned} \quad (131)$$

Note that the major drawback in the previous expression is that the complete eigenstructure of the matrix $\mathbf{A}(\mathbf{Q})$ is required since $\text{sign}(\mathbf{A}) = \mathbf{R} \text{sign}(\Lambda) \mathbf{R}^{-1}$. However, on the other hand, the Osher-Romberg method is very little dissipative and is stable under the standard CFL condition.

We would like to underline that, without a well balanced way of treating the viscosity, the entire scheme will lose the property of preserving equilibria up to machine precision; instead this choice of the viscosity term guarantees the well balancing and moreover does it in an automatic way, so that it can be easily applied to other systems of equations and family of equilibria. For example it has been applied to a diffuse interface model deduced from the Baer-Nunziato system in [79] allowing to sharply follow the water-air interface.

HLL scheme. Following [37], the standard HLL scheme can be written in the form (104) with (81), where the numerical viscosity term is given by

$$\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \alpha_{i+1/2}^0 I_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) + \alpha_{i+1/2}^1 \mathcal{R}_{i+1/2}, \quad (132)$$

where $I_{i+1/2}$ is the identity matrix,

$$\mathcal{R}_{i+1/2} = \mathbf{f}(\mathbf{q}_{i+1}) - \mathbf{f}(\mathbf{q}_i) + \mathcal{B}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) \quad (133)$$

and

$$\begin{aligned} \alpha_{i+1/2}^0 &= \frac{S_{i+1/2}^R |S_{i+1/2}^L| - S_{i+1/2}^L |S_{i+1/2}^R|}{S_{i+1/2}^R - S_{i+1/2}^L}, \\ \alpha_{i+1/2}^1 &= \frac{|S_{i+1/2}^R| - |S_{i+1/2}^L|}{S_{i+1/2}^R - S_{i+1/2}^L} \end{aligned} \quad (134)$$

being $S_{i+1/2}^L$ and $S_{i+1/2}^R$ respectively the minimum and the maximum of the wave speeds of the Riemann problem associated to states \mathbf{q}_i and \mathbf{q}_{i+1} . In particular to compute $S_{i+1/2}^L$ we recover the eigenvalues associated to \mathbf{q}_i and $\frac{\mathbf{q}_i + \mathbf{q}_{i+1}}{2}$ and we consider the minimum value, similarly to compute $S_{i+1/2}^R$ we take the maximum of the eigenvalues associated to $\frac{\mathbf{q}_i + \mathbf{q}_{i+1}}{2}$ and \mathbf{q}_{i+1} . It is clear that $\mathcal{V}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ does not vanish if \mathbf{q}_{i+1} and \mathbf{q}_i lie on a stationary solution: $\mathcal{R}_{i+1/2}$ vanishes, but it is not the case for the term $\alpha_{i+1/2}^0 I_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$.

Here, we follow the ideas described in [35] and [37] to modify the viscosity term such that the resulting scheme is exactly well balanced for the stationary solutions (98)-(99). In particular $I_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$, will be replaced by $\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ that vanishes when a stationary solution is considered. Here we consider the following expression for $\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$

$$\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \begin{pmatrix} b_2^{i+1/2} \left(\frac{\rho}{\gamma p}\right)_{i+1/2} \\ \Delta(r\rho u)_{i+1/2} \\ b_2^{i+1/2} \left(\frac{\rho}{\gamma p}\right)_{i+1/2} (v)_{i+1/2} \\ b_2^{i+1/2} \left(\frac{\rho}{\gamma p}\right)_{i+1/2} (z)_{i+1/2} \\ 0 \end{pmatrix} \quad (135)$$

where $b_2^{i+1/2}$ is given in (117), $\left(\frac{\rho}{\gamma p}\right)_{i+1/2} = \frac{\rho_{i+1} + \rho_i}{\gamma(p_{i+1} + p_i)}$,

$\Delta(r\rho u)_{i+1/2} = (r\rho u)_{i+1} - (r\rho u)_i$, $(v)_{i+1/2} = \frac{v_{i+1} + v_i}{2}$,

$(z)_{i+1/2} = \frac{z_{i+1} + z_i}{2}$, being $z = \frac{\partial(ru(\rho E + p))}{\partial q_2}$. Following [35] and

[37] $\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is obtained as follows: we start by computing the eigenstructure of the extended Jacobian matrix \mathbf{A} at the equilibrium

$$\mathbf{A}(\mathbf{Q}) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ r \frac{\partial p}{\partial q_1} & 0 & r \frac{\partial p}{\partial q_3} & r \frac{\partial p}{\partial q_4} & \rho \frac{Gm_s}{r} + \rho v^2 \\ 0 & v & 0 & 0 & 0 \\ 0 & \frac{\partial(ru(\rho E + p))}{\partial q_2} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (136)$$

In this situation the eigenstructure is easy to be computed: let \mathbf{R} the matrix of the right-eigenvectors and $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_5)$ the diagonal matrix of the eigenvalues of (136). In particular we have

$$\Lambda = \text{diag} \left(\frac{\rho u + \sqrt{\gamma p \rho}}{\rho}, \frac{\rho u - \sqrt{\gamma p \rho}}{\rho}, u, u, 0 \right), \quad \text{with } u = 0. \quad (137)$$

Then $\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i)$ is given by

$$\tilde{I}_{i+1/2}(\mathbf{q}_{i+1} - \mathbf{q}_i) = \mathbf{R}_{i+1/2} \tilde{\Lambda} (\mathbf{R}_{i+1/2})^{-1} (\mathbf{q}_{i+1} - \mathbf{q}_i), \quad (138)$$

where

$$\tilde{\Lambda} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (139)$$

Note that $\tilde{\Lambda}$ is a diagonal matrix composed with 0 and 1, where the 0 elements correspond to the zero eigenvalues at the stationary solution. The final expression (135) is obtained considering the following relation that it is derived from (88)

$$\begin{aligned} r_{i+1/2} & \left(\left(\frac{\partial p}{\partial q_1} \right)_{i+1/2} \Delta q_{1,i+1/2} + \left(\frac{\partial p}{\partial q_3} \right)_{i+1/2} \Delta q_{3,i+1/2} \right. \\ & \left. + \left(\frac{\partial p}{\partial q_4} \right)_{i+1/2} \Delta q_{4,i+1/2} + \left(\rho \frac{Gm_s}{r^2} - \frac{\rho v^2}{r} \right)_{i+1/2} \Delta r_{i+1/2} \right) \\ & = b_2^{i+1/2}. \end{aligned} \quad (140)$$

We underline once again that this *identity modification* allows the viscosity to vanish at the equilibria, so that the scheme is exactly well balanced. Finally, we would like to note that a similar HLL scheme could also be obtained within the framework of path-conservative HLLM methods recently proposed by [57], in which according to [70] the intermediate HLL state is assumed to be linear rather than constant.

5.4.1 2nd order well balanced reconstruction

Let us recall the numerical scheme presented in (80) considering the space-time conservation form of the PDE

$$\begin{aligned} \mathbf{Q}_i^{n+1} &= \mathbf{Q}_i^n - \frac{\Delta t}{\Delta r} \left(\mathcal{D}_{i-\frac{1}{2}}^+ \left(\mathbf{q}_{i-\frac{1}{2}}^{n+,-}, \mathbf{q}_{i-\frac{1}{2}}^{n+,+} \right) + \mathcal{D}_{i+\frac{1}{2}}^- \left(\mathbf{q}_{i+\frac{1}{2}}^{n+,-}, \mathbf{q}_{i+\frac{1}{2}}^{n+,+} \right) \right) \\ & \quad - \frac{\Delta t}{\Delta r} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \frac{\partial}{\partial r} \mathbf{f} \left(\mathbf{q}_i^{n+}(r) \right) dr \\ & \quad - \frac{\Delta t}{\Delta r} \int_{r_{i-\frac{1}{2}}}^{r_{i+\frac{1}{2}}} \mathbf{B}_1 \left(\mathbf{q}_i^{n+}(r) \right) \frac{\partial}{\partial r} \left(\mathbf{q}_i^{n+}(r) \right) dr, \end{aligned} \quad (141)$$

where $\mathbf{q}_i^n(r, t)$ is the approximation of the conserved variables inside cell I_i at time t^n , $\mathbf{q}_{i-\frac{1}{2}}^{n+,-}(t) = \mathbf{q}_i^n(r_{i-1/2}, t^{n+1/2})$ and $\mathbf{q}_{i+\frac{1}{2}}^{n+,-}(t) = \mathbf{q}_i^n(r_{i+1/2}, t^{n+1/2})$, that is the evaluation of $\mathbf{q}_i^n(r, t)$ at the two boundaries of cell I_i at the time-midpoint of $[t^n, t^{n+1}]$. We would like to underline that in order to obtain a second order scheme \mathbf{q}_i^n should be a second order reconstruction of the cell averages $\mathbf{Q}_{i-1}^n, \mathbf{Q}_i^n, \mathbf{Q}_{i+1}^n$.

According to [131], [33] and Proposition 5.16, scheme (141) is well balanced if both, the underlying first order scheme and the reconstruction operator are well balanced, and all the integrals that appear in (141) are computed exactly. Therefore, in order to define a second order scheme, a second order well balanced reconstruction operator should be defined.

The most popular way to define a second order reconstruction operator is based on the MUSCL method introduced by van Leer in [153] joint with the *minmod* limiter. He proposed to reconstruct \mathbf{q}_i^n using a linear polynomial in space and time as follows

$$\mathcal{P}_i^n(r, t) = \mathbf{Q}_i^n + \frac{\Delta \mathbf{Q}_i^n}{\Delta r} (r - r_i) + \partial_t \mathbf{Q}_i^n (t - t^n), \quad (142)$$

where

$$\Delta \mathbf{Q}_i^n = \text{minmod} \left(\Delta \mathbf{Q}_{i-1/2}^n, \Delta \mathbf{Q}_{i+1/2}^n \right) \quad (143)$$

with $\Delta \mathbf{Q}_{i-1/2}^n = \mathbf{Q}_i^n - \mathbf{Q}_{i-1}^n$, $\Delta \mathbf{Q}_{i+1/2}^n = \mathbf{Q}_{i+1}^n - \mathbf{Q}_i^n$ and

$$\text{minmod}(a, b) = \begin{cases} 0, & \text{if } ab \leq 0 \\ a, & \text{if } |a| < |b| \\ b, & \text{if } |a| \geq |b|. \end{cases} \quad (144)$$

It is clear that the standard MUSCL method is only well balanced for linear stationary solutions, which is not the case here. In this work we therefore follow the strategy proposed in [31], where the reconstruction operator is defined as a combination of a smooth stationary solution together with a standard reconstruction operator to reconstruct the *fluctuations* with respect to the given stationary solution, that is

$$\mathbf{q}_i^n(r, t) = \mathbf{Q}_i^E(r) + \mathcal{P}_i^f(r, t), \quad r \in I_i, t \in [t^n, t^{n+1}], \quad (145)$$

where $\mathcal{P}_i^f(r, t)$ is the standard MUSCL reconstruction operator applied to the fluctuations around the stationary solution at every cell of the stencil. Thus, if we define

$$\mathbf{Q}_i^{f,n} = \mathbf{Q}_i^n - \mathbf{Q}_i^E, \quad \mathbf{Q}_{i-1}^{f,n} = \mathbf{Q}_{i-1}^n - \mathbf{Q}_{i-1}^E, \quad \mathbf{Q}_{i+1}^{f,n} = \mathbf{Q}_{i+1}^n - \mathbf{Q}_{i+1}^E, \quad (146)$$

then, $\mathcal{P}_i^f(r, t)$ is defined as follows:

$$\mathcal{P}_i^{f,n}(r, t) = \mathbf{Q}_i^{f,n} + \frac{\Delta \mathbf{Q}_i^{f,n}}{\Delta r} (r - r_i) + \partial_t \mathbf{Q}_i^{f,n} (t - t^n), \quad (147)$$

where

$$\Delta \mathbf{Q}_i^{f,n} = \text{minmod} \left(\Delta \mathbf{Q}_{i-1/2}^{f,n}, \Delta \mathbf{Q}_{i+1/2}^{f,n} \right) \quad (148)$$

with

$$\Delta \mathbf{Q}_{i-1/2}^{f,n} = \mathbf{Q}_i^{f,n} - \mathbf{Q}_{i-1}^{f,n}, \quad \Delta \mathbf{Q}_{i+1/2}^{f,n} = \mathbf{Q}_{i+1}^{f,n} - \mathbf{Q}_i^{f,n}. \quad (149)$$

Note that we have replaced $\partial_t \mathbf{Q}_i^{f,n}(t - t^n)$ by $\partial_t \mathbf{Q}_i^n(t - t^n)$ in (147) as $\partial_t \mathbf{Q}_i^E = 0$. It is clear from its construction that the

reconstruction operator is exactly well balanced, and it is second order accurate for non-stationary solutions as $\mathbf{Q}^E(r)$ is a smooth stationary solution. The term $\partial_t \mathbf{Q}_i^n$ indicates the time derivative of \mathbf{Q} and it can be computed using a discrete version of the governing equation

$$\begin{aligned} \partial_t \mathbf{Q}_i^n &= - \frac{\mathbf{f}(\mathbf{q}_{i+1/2}^{n,-}) - \mathbf{f}(\mathbf{q}_{i-1/2}^{n,+})}{\Delta r} - \frac{\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})}{\Delta r}, \\ \mathbf{q}_{i\pm 1/2}^{n,\mp} &= \mathbf{q}_i(x_{i\pm 1/2}^\mp, t^n), \end{aligned} \quad (150)$$

where the fluxes have been approximated by a central finite difference with respect to the cell center r_i , and $\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})$ is obtained in the same way of (110),(117),(120),(122), where by replacing for example \mathbf{q}_i and \mathbf{q}_{i+1} by $\mathbf{q}_{i+1/2}^{n,-}$ and $\mathbf{q}_{i-1/2}^{n,+}$ respectively, and using as central value the cell average one obtains

$$\begin{aligned} \frac{\mathcal{B}_i(\mathbf{q}_{i+1/2}^{n,-} - \mathbf{q}_{i-1/2}^{n,+})}{\Delta r} &= (b_1^i \ b_2^i \ b_3^i \ b_4^i \ b_5^i)^T, \quad \text{with} \\ b_1^i &= b_5^i = 0, \quad b_3^i = \rho_i u_i v_i, \quad b_4^i = \rho_i u_i \frac{G m_s}{r_i}, \\ b_2^i &= r_i \left(p_{i+1/2}^{f,n,-} - p_{i-1/2}^{f,n,+} \right) + \left(r_i \rho_i^E (\zeta_r)_i^f + r_i \rho_i^f (\zeta_r)_i \right). \end{aligned} \quad (151)$$

The last ingredient for a second order scheme is the computation of the integrals in (141): the first one can be computed exactly

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \frac{\partial}{\partial r} \mathbf{f}(\mathbf{q}_i(r, t)) dr = \mathbf{f}(\mathbf{q}_{i+1/2}^{n,-}) - \mathbf{f}(\mathbf{q}_{i-1/2}^{n,+}). \quad (152)$$

Note that this first integral vanishes for stationary solutions with $u = 0$. The second integral is more sophisticated, and it is not easy to compute it exactly, except in some particular situations. Therefore we will use a quadrature formula to approximate this integral, but this must be done carefully to maintain the well balanced property of the scheme: effectively, a wrong choice in the quadrature formula will destroy all the work we have done up to now in order to define a well balanced scheme. Here we proceed as follows: first we express the particular form of the reconstruction operator: $\mathbf{q}_i^n(x, t) = \mathbf{Q}_i^E(x) + \mathcal{P}_i^f(x, t)$ and we use the fact that

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \mathbf{B}_1(\mathbf{Q}_i^E(r)) \frac{\partial \mathbf{Q}_i^E(r)}{\partial r} dr = \mathbf{0}. \quad (153)$$

Here, we only show the details for the second component of

$$\int_{r_{i-1/2}}^{r_{i+1/2}} \mathbf{B}_1(\mathbf{q}_i^n(r)) \frac{\partial \mathbf{q}_i^n(r)}{\partial r} dr, \quad (154)$$

i.e.

$$\begin{aligned} & \int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial p}{\partial r} + \rho \left(\frac{Gm_s}{r^2} - \frac{v^2}{r} \right) \right] dr \\ &= \int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial(p^E + p^f)}{\partial r} + (\rho^E + \rho^f) (\zeta^E + \zeta^f)_r \right] dr \quad (155) \\ &= \int_{r_{i-1/2}}^{r_{i+1/2}} r \frac{\partial p^f}{\partial r} + r \rho^E \zeta_r^f + r \rho^f \zeta_r^f dr. \end{aligned}$$

Now, the mid-point quadrature formula is used to ensure second order accuracy obtaining that

$$\begin{aligned} & \int_{r_{i-1/2}}^{r_{i+1/2}} r \left[\frac{\partial p}{\partial r} + \rho \left(\frac{Gm_s}{r^2} - \frac{v^2}{r} \right) \right] dr \\ & \approx \Delta r \left[r_i (\Delta p^f)_i + (r \rho^E)_i (\zeta_r^f)_i + (r \rho^f)_i (\zeta_r)_i \right], \quad (156) \end{aligned}$$

$$\text{where } (\Delta p^f)_i = \frac{p_{i+1/2}^{f,-} - p_{i-1/2}^{f,+}}{\Delta r},$$

$$(\zeta_r)_i = \frac{Gm_s}{r_i^2} - \frac{v_i^2}{r_i}, \quad (\zeta_r^f)_i = \frac{v_i^{E^2}}{r_i} - \frac{v_i^2}{r_i}.$$

It is clear that this approximation is second order accurate and, moreover, will vanish for stationary solutions (98)-(99). For the third and fourth component we could perform the same procedure, but, again, as both trivially vanish when $u = 0$, we could use directly the mid-point rule.

Finally, note that $r_{i+1/2}^\pm = r_{i+1/2}$ and therefore $\Delta r_{i+1/2} = 0$. Therefore $\mathcal{B}_{i+1/2}(\mathbf{q}_{i+\frac{1}{2}}^{n^+,+} - \mathbf{q}_{i+\frac{1}{2}}^{n^+,-})$ reduces to

$$\mathcal{B}_{i+1/2}(\mathbf{q}_{i+\frac{1}{2}}^{n^+,+} - \mathbf{q}_{i+\frac{1}{2}}^{n^+,-}) = (0, b_2^{i+1/2}, 0, 0, 0)^T \quad (157)$$

where

$$b_2^{i+1/2} = r_{i+1/2} \Delta p_{i+1/2}^f = r_{i+1/2} (p_{i+1/2}^{f,+} - p_{i+1/2}^{f,-}). \quad (158)$$

5.5 Numerical results

First of all, we show the ability of both schemes to preserve a wide class of stationary solutions of the Euler equations with gravity of the form (98)-(99); we also report the convergence tables for some smooth solutions. Then, we test both methods with a classical Riemann problem: this show that our methods are able to deal with situations far from the equilibrium, hence they do not fall into the case of perturbation methods. And finally, we study their behavior in capturing small perturbations around the equilibrium.

Table 5: Constant pressure equilibrium. The following results show the capability of the schemes to preserve equilibria both for a hierarchy of meshes for a fixed time $t = 1$ (on the left) and for a fixed mesh ($N = 64$ cells) and for increasing computational times. The table on the left refers to the L_1 -norm error between the continuous ρ_1 profile and the table on the right refers to the discontinuous ρ_2 profile. Data have been obtained using either the Osher or HLL flux (and no significant differences have been noticed).

tend = 1		N = 64	
N	E_ρ - Osher	time	E_ρ - Osher
64	9.54E-17	1	9.54E-17
128	9.54E-17	2	2.36E-16
256	6.49E-16	5	8.85E-16
512	6.23E-16	10	1.67E-15
1024	1.21E-15	50	6.24E-17

tend = 1		N = 64	
N	E_ρ - HLL	time	E_ρ - HLL
64	8.45E-18	1	8.45E-18
128	1.38E-16	2	1.19E-17
256	5.54E-16	5	6.71E-16
512	2.64E-15	10	2.42E-15
1024	5.05E-16	50	1.13E-13

5.5.1 Stationary solutions with constant pressure

Simple, but non trivial, stationary solutions of the Euler equations can be obtained by considering velocities as in (98) and a constant pressure p . It is easy to verify that under these conditions for any density profile the velocity in the angular direction v must satisfy

$$v = \sqrt{\frac{Gm_s}{r}}, \quad (159)$$

while $u = 0$. For the numerical simulations we consider a spatial domain $r \in [1, 2]$, $G = 1$, $m_s = 1$, $\gamma = 1.4$, $p = 1$ and two density profiles:

$$\rho_1 = r, \quad (160)$$

$$\rho_2 = \begin{cases} 1, & \text{if } r < 1.5 \\ 0.1, & \text{if } r \geq 1.5. \end{cases} \quad (161)$$

In Table 5 we report the errors between the exact equilibrium and the numerical solution obtained with both schemes using a hierarchy of meshes and long term time integration. We can notice that all the errors are of the order of machine precision and no significant differences can be noticed between the two fluxes. Moreover the method is perfectly well balanced both with continuous and discontinuous density profiles, as expected.

5.5.2 General equilibrium

Using the equilibrium relation between the pressure and the gravitational forces in (99) and ζ given by (113), we obtain

Table 6: General equilibrium. L_1 -norm error for the density between the exact and the numerical solution. On the left we have the error for different meshes at $t = 1$ and on the right we show the error for a given mesh ($N = 64$ cells) for different computational times.

tend = 1		$N = 64$	
N	E_ρ - OSHER	time	E_ρ - HLL
64	6.28E-15	1	5.03E-15
128	1.17E-14	2	1.01E-14
256	1.70E-14	5	2.65E-14
512	2.15E-14	10	7.21E-14
1024	3.19E-14	50	3.07E-12

another class of stationary solutions of the Euler equations

$$\rho = \rho_0 e^{-\zeta(r)}, \quad p = \rho + p_0, \quad v = \sqrt{r \left(\frac{Gm_s}{r^2} - \zeta_r \right)}. \quad (162)$$

We have applied both schemes to two different choices of ζ obtaining always a well balanced result. Table 6 shows the L_1 -norm error for the density between the equilibrium and the numerical solution in the case

$$\zeta = kr, \quad k = -1, \quad \rho = \rho_0 e^{-kr}, \quad \rho_0 = 1, \quad p = \rho + p_0, \quad p_0 = 1. \quad (163)$$

Again, both methods are able to exactly preserve these non-trivial equilibria.

5.5.3 Order of convergence

To study numerically the order of convergence of both schemes we have considered the following equilibrium situation

$$\rho = 1, \quad u = 0, \quad p = 1, \quad v = \sqrt{r \left(\frac{Gm_s}{r^2} - \zeta_r \right)}, \quad (164)$$

and at the initial time, we have added a small perturbation (with a Gaussian profile) to the velocity field

$$\begin{aligned} \tilde{u} &= u + 10^{-5} \exp\left(\frac{-0.5(r-1.5)^2}{0.01}\right), \\ \tilde{v} &= v + 10^{-5} \exp\left(\frac{-0.5(r-1.5)^2}{0.01}\right). \end{aligned} \quad (165)$$

We have computed a reference solution using our method with the Osher-type flux over a fine mesh ($N = 2^{13} = 8192$). In Table 7 we report the L_1 norm errors for the density ρ with respect to our reference solution and both numerical schemes achieve second order of convergence.

Table 7: Perturbation around a stationary solution. The reference solution has been obtained with the second order Osher-type scheme over 2^{13} cells. L_1 -norm errors for ρ at time $t = 0.1$ are shown: on the left we report the result obtained using the Osher-type flux and on the right using the HLL-type flux.

Osher $\mathcal{O}2$			HLL $\mathcal{O}2$		
N	$\varepsilon(\rho)$	$\mathcal{O}(L_1)$	N	$\varepsilon(\rho)$	$\mathcal{O}(L_1)$
16	1.59E-07	-	16	1.16E-07	-
32	3.82E-08	2.06	32	2.90E-08	2.01
64	9.50E-09	2.00	64	7.22E-09	2.00
128	2.31E-09	2.04	128	1.77E-09	2.03
256	5.72E-10	2.01	256	4.44E-10	1.99
512	1.45E-10	1.97	512	1.14E-10	1.96

5.5.4 Riemann Problem

To show that our method is accurate even far away from an equilibrium, we consider as initial condition a classical Riemann problem with non-vanishing angular velocity

$$\begin{aligned} \rho_L = 1.0, u_L = 0, v_L = \sqrt{\frac{Gm_s}{r}}, p_L = 1.0, r = r, 1 \leq r \leq 4.5, \\ \rho_R = 0.1, u_R = 0, v_R = \sqrt{\frac{Gm_s}{r}}, p_R = 0.1, r = r, 4.5 < r \leq 8, \end{aligned}$$

and we compute the solution by employing the schemes set up to preserve the equilibrium in (161). We report the results obtained with the first and second order scheme and with the HLL and Osher-type flux in Figure (18). Note that both schemes produce quite similar results.

5.5.5 Evolution of perturbations

Following the idea presented in [97] we have tried to study small perturbations around the equilibrium. We have considered a density profile as in (160) and we have imposed a periodic perturbation on the velocity u through the left boundary conditions, by imposing

$$u_0 = A \sin\left(6 \frac{2\pi t^n}{t_f}\right), \quad t_f = 1. \quad (166)$$

Two situations are analyzed. First we consider a big perturbation, with $A = 10^{-2}$ and we simulate the evolution using the second order well balanced HLL scheme and a standard second order HLL scheme using a hierarchy of grids with increasing number of cells. A reference solution computed with the second order well balanced HLL method is also considered using a fine grid composed of $N = 2048$ cells. Figure (19) shows the errors for the different meshes. Note that in this case no big differences are visible between the well balanced and not well balanced schemes as the perturbations are so large so that shocks are quickly generated

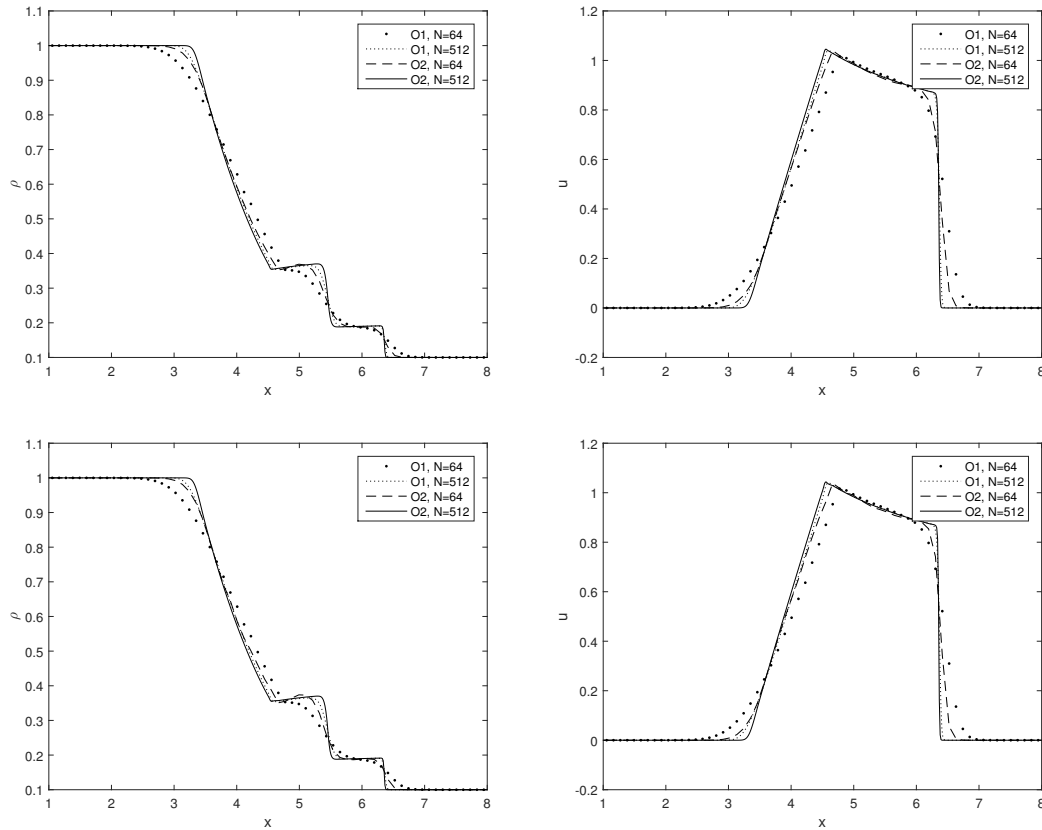


Fig. 18: Riemann problem at final time $t_f = 1$. On the top the density and velocity profiles obtained using the HLL scheme and on the bottom the profiles obtained with the Osher-type flux. We have employed two meshes: a coarse one with $N = 64$ elements and a fine one with $N = 512$ elements. Moreover, we have compared the first and second order schemes.

and the solution is far away from the stationary profile. The situation changed significantly when a small perturbation is considered ($A = 10^{-5}$). In that case the well balanced method performs much better than the non well balanced scheme on the finer grid, as shown in Figure (19).

6 Well balancing: coupling with nonconforming ALE

Now, we extend our well balanced method to the two dimensional ALE context on moving nonconforming meshes. In particular, the scheme will *inherit* the well balanced property of the previous one-dimensional scheme in the radial direction and the addition of some constraints on the mesh structure will guarantee the well balancing in the angular direction. The extension concerns first the Euler equations of gas dynamics with gravity, see Section 6.1; a large number of numerical test are presented in Section 6.2. Then we consider the shallow water equations (in Section 6.3) and in particular the steady vortex test case already introduced in Section 3.6.5. One can appreciate how the same basic ideas

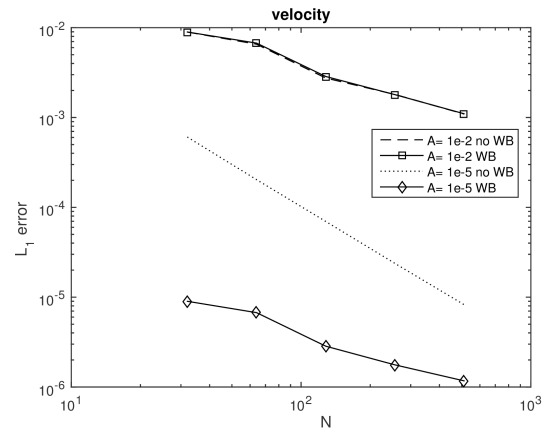


Fig. 19: Error in the L_1 norm between a reference solution and the numerical solutions computed with the well balanced HLL method and a second order non well balanced scheme. Well balanced and non well balanced methods perform equally well for large perturbations, while well balanced schemes perform significantly better for the small perturbation problem.

can be applied to a different set of equations and of equilibrium solutions.

6.1 WB ALE for the Euler equations with gravity

Consider the Euler equations of gas dynamics with gravity in the two dimensional non-conservative form stated in (102). This Section is organized as follows: first, we revisit the moving domain discretization introduced in Section 3.1 by specifying the required shape of the elements and the mesh velocity. Next, we derive the one-step path-conservative ALE scheme, and we explain where the 1D well balanced techniques are employed in order to guarantee the well balancing of the scheme even in a two dimensional moving mesh framework.

6.1.1 Domain discretization and mesh constraints

To discretize the moving domain, we consider a nonconforming mesh \mathcal{T}_Ω^n and we fix a total number $N_E = N \times M$ of elements that we suppose to be all quadrilaterals.

These elements should satisfy the following conditions:

- i. their barycenters should be aligned along straight lines with $r = r_i = \text{const}$, $i = 1, \dots, N$,
- ii. the two bounding edges of each element in radial direction must be aligned with $r = r_{i \pm 1/2} = \text{const}$, $i = 1, \dots, N+1$,
- iii. the other two bounding edges must be parallel between them.

For example a Cartesian grid satisfies these conditions, but we could accept even something more general (which allows us to move the computational domain). See Figure 20 for a general mesh that satisfies the above constraints. In Section 6.1.2 these choices will be justified. We emphasize that our numerical scheme works for completely general unstructured and nonconforming moving meshes, but it will be well balanced only if the mesh satisfies these special conditions. Indeed a general well balanced scheme for completely unstructured meshes is more difficult to be achieved, since the considered equilibria have $v \neq 0$, which implies that $\mathbf{g} \neq \mathbf{0}$ (recall that instead $\mathbf{f} = \mathbf{0}$). This does not mean that it is impossible, but it will be the object of another work, because the above conditions do not appear to us as restrictive for our scopes.

We have decided to couple the well balanced techniques with a Lagrangian-type method in order to reduce the errors due to the advection. And our nonconforming ALE algorithm results to be particularly *well suited* for this situation, where the gas at the equilibrium is advected with a *known* equilibrium velocity field $\mathbf{V}(\mathbf{x}) = (u^E(\mathbf{x}), v^E(\mathbf{x}))$

Fig. 20: Example of a mesh that allows a well balanced treatment of the fluxes. Each element has two vertical edges and the other two are parallel between them. Besides the vertical edges lie on straight lines and the barycenters are aligned along $r = r_i$. Moreover the domain is periodic so that $\varphi = 0$ coincides with $\varphi = 2\pi$.

which reads

$$u^E(\mathbf{x}) = 0, \quad v^E(\mathbf{x}) = \sqrt{r \left(\frac{Gm_s}{r^2} - \zeta_r \right)}. \quad (167)$$

Note that the *a priori* knowledge of the velocity field significantly simplifies the application of an ALE scheme: indeed, we can move the nodes following directly the exact equilibrium velocity, which is not affected by any physical or numerical perturbation. The node velocity $\bar{\mathbf{V}}_k^n$ of (43) can be obtained by choosing

$$\bar{\mathbf{V}}_{k,j} = \mathbf{V}(\mathbf{x}_j^n), \quad (168)$$

being \mathbf{x}_j the barycenter of element P_j^n . This allows us to control the movement of the mesh avoiding the violation of the above conditions: indeed the radial component of $\bar{\mathbf{V}}_k^n$ will be always zero, hence nodes will slide along straight lines with $r = \text{const}$ where the edges lie. Besides, since the barycenters are placed on the straight lines with $r = r_i$, all nodes lying on the same edge will move with the same velocity maintaining the parallelism constraint between the edges.

Furthermore, the presence of slide lines clearly requires a special approach to preserve a high level of grid quality during the mesh motion, and this is automatically provided by our nonconforming treatment of the interfaces, in addition simplified by the fact that their position is *a priori* known: $r = \text{const}$.

6.1.2 Well balanced direct ALE scheme

In order to obtain a space-time formulation of a direct path-conservative ALE scheme, as proposed in [59], the governing PDE (1) is first reformulated in a space-time divergence form as

$$\tilde{\mathbf{V}} \cdot \tilde{\mathbf{F}}(\mathbf{Q}) + \tilde{\mathbf{B}}(\mathbf{Q}) \cdot \tilde{\mathbf{V}}\mathbf{Q} = \mathbf{S}(\mathbf{Q}), \quad \tilde{\mathbf{V}} = (\partial_r, \partial_\varphi, \partial_t)^T \quad (169)$$

with

$$\tilde{\mathbf{F}} = (\mathbf{F}, \mathbf{Q})^T = (\mathbf{f}, \mathbf{g}, \mathbf{Q})^T, \quad \tilde{\mathbf{B}} = (\mathbf{B}, \mathbf{0})^T = (\mathbf{B}_1, \mathbf{0}, \mathbf{0})^T, \quad \text{and } \mathbf{S} = \mathbf{0},$$

and it is then integrated over the space-time control volume C_i^n

$$\int_{C_i^n} (\tilde{\mathbf{V}} \cdot \tilde{\mathbf{F}}(\mathbf{Q}) + \tilde{\mathbf{B}}(\mathbf{Q}) \cdot \tilde{\mathbf{V}}\mathbf{Q}) \, dxdt = \mathbf{0}. \quad (170)$$

Now, the space-time volume integral of $\tilde{\mathbf{V}} \cdot \tilde{\mathbf{F}}(\mathbf{Q})$ can be rewritten using the Gauss theorem as

$$\int_{\partial C_i^n} \tilde{\mathbf{F}} \cdot \tilde{\mathbf{n}} + \int_{C_i^n} \tilde{\mathbf{B}} \cdot \tilde{\nabla} \mathbf{Q} = \mathbf{0}, \quad (171)$$

where $\tilde{\mathbf{n}} = (\tilde{n}_r, \tilde{n}_\phi, \tilde{n}_t)$ is the outward pointing space-time unit normal vector on the space-time surface ∂C_i^n . Taking into account the jump of $\tilde{\mathbf{B}}$ at the interfaces, the final high order ALE one-step Finite Volume scheme is then obtained from equation (171) as

$$\begin{aligned} |P_i^{n+1}| \mathbf{Q}_i^{n+1} &= |P_i^n| \mathbf{Q}_i^n - \sum_j \int_0^1 \int_0^1 |\partial C_{ij}^n| \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} d\chi d\tau \\ &\quad - \int_{C_i^n} \tilde{\mathbf{B}}(\mathbf{q}_i^n) \cdot \tilde{\nabla} \mathbf{q}_i^n dx dt \end{aligned} \quad (172)$$

where $\mathbf{q}_i^n(\mathbf{x}, t)$ is a well balanced second order reconstruction of the conserved variables \mathbf{Q} inside cell P_i^n at time t^n , and the discontinuity of the solution at the space-time sub-surface ∂C_{ij}^n is resolved by a well balanced path-conservative ALE flux $\tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$, which accounts for the jump in the discrete solution between two neighbors across the intermediate space-time lateral surface. This *generalizes* the ALE scheme introduced in (30) (only in the FV case) for conservative systems to the non-conservative case. Recall that when the lateral surface is shared between more than two control volumes we have to compute the flux across each sub-piece and sum each contribution.

Well balanced ALE numerical flux function. The core of the well balanced method in (172) is the design of the well balanced space-time flux function. Its final expression will be

$$\begin{aligned} \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} &= \frac{1}{2} (\tilde{\mathbf{F}}(\mathbf{q}^+) + \tilde{\mathbf{F}}(\mathbf{q}^-) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)) \cdot \tilde{\mathbf{n}}_{ij} \\ &\quad - \frac{1}{2} \mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-), \end{aligned} \quad (173)$$

where \mathbf{q}^- is the value of the reconstructed numerical solution inside the element C_i^n evaluated at the space-time midpoint $M_{i,j}^n$ of the lateral surface ∂C_{ij}^n , and \mathbf{q}^+ is the evaluation at the same point of the reconstructed numerical solution inside the neighbor C_j^n at ∂C_{ij}^n . Besides, generalizing the notation introduced in Section 5.1, $\tilde{\mathbf{F}}$ is the physical flux, the term $\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)$ represents a well balanced way to write the non-conservative products, and $\mathcal{V}_{i+\frac{1}{2}}(\mathbf{q}^+ - \mathbf{q}^-)$ is the viscosity term.

As already pointed out, according to [131] (summarized in Definition 5.12 and Proposition 5.14), the numerical flux should satisfy the following properties

$$\tilde{\mathbf{D}}_{ij}(\mathbf{Q}, \mathbf{Q}) \cdot \tilde{\mathbf{n}}_{ij} = \mathbf{0} \quad \forall \mathbf{Q} \in \Omega, \text{ and} \quad (174)$$

$$\tilde{\mathbf{D}}_{ij}(\mathbf{q}^-, \mathbf{q}^+) \cdot \tilde{\mathbf{n}}_{ij} = \int_0^1 \mathbf{A}_n^V(\Phi(s; \mathbf{q}^-, \mathbf{q}^+)) \frac{\partial \Phi}{\partial s}(s; \mathbf{q}^-, \mathbf{q}^+) ds, \quad (175)$$

where, due to the ALE framework,

$$\begin{aligned} \mathbf{A}_n^V(\mathbf{Q}) &= \sqrt{\tilde{n}_r^2 + \tilde{n}_\phi^2} \left(\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}} + \mathbf{B} \right) \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right), \\ \mathbf{n} = (n_r, n_\phi) &= \frac{(\tilde{n}_r, \tilde{n}_\phi)^T}{\sqrt{\tilde{n}_r^2 + \tilde{n}_\phi^2}}, \end{aligned} \quad (176)$$

with \mathbf{I} representing the identity matrix and $\mathbf{V} \cdot \mathbf{n}$ denoting the local normal mesh velocity.

We explain now how to discretize \mathcal{B}_{ij} and \mathcal{V}_{ij} in (173) in a well balanced way. Here we perform our reasoning edge-by-edge and we distinguish two situations: the first one across the vertical edges, which evolving in time originate a surface orthogonal to the radial direction, easier to be treated, and the second one across the other two parallel edges (see the constraints stated at the beginning of Section 6.1.1).

First of all, it is easy to see that the flux across the lateral surfaces evolved from the vertical edges coincides with the one dimensional flux. Indeed, in this case, $\mathbf{n} = (n_r, 0)$, $\mathbf{V} = (0, v)$ and so $\mathbf{V} \cdot \mathbf{n} = 0$. Hence $\mathbf{A}_n^V(\mathbf{Q}) = \mathbf{J}_r(\mathbf{Q}) + \mathbf{B}_1(\mathbf{Q})$ which coincides with the one dimensional Jacobian. So we can discretize \mathcal{B}_{ij} as stated in (110)-(117)-(120)-(122), and \mathcal{V}_{ij} by using the Osher-Romberg method (131) or the modified HLL scheme as described in Section 5.4. Therefore the scheme is well balanced in the radial direction and second order accurate provided that the reconstruction \mathbf{q}_i^n and the integrals in (172) are computed in a well balanced manner and with second order of accuracy (see Section 6.1.2).

For what concerns the flux through the other two surfaces (see Point (iii) of Section 6.1.1) let us first state the following remark.

Remark 6.11 *Given an element P_i^n consider its two edges which are parallel between them but not vertical. Their evolution in time originates two parallel surfaces with the same areas and opposite normal vectors. Moreover call P_{j_1} and P_{j_2} the two neighbors of P_i through these edges. Since the barycenters of P_i , P_{j_1} and P_{j_2} are aligned on the same vertical line, i.e. their r -coordinate is the same, the equilibrium values \mathbf{Q}_i^E , $\mathbf{Q}_{j_1}^E$ and $\mathbf{Q}_{j_2}^E$ coincide.*

Now let us rewrite (176) as

$$\mathbf{A}_n^V(\mathbf{Q}) = \sqrt{\tilde{n}_r^2 + \tilde{n}_\phi^2} \left((\mathbf{J}_f + \mathbf{B}_1) n_r + \mathbf{J}_g n_\phi - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right). \quad (177)$$

and (175) as

$$\begin{aligned} \tilde{\mathbf{D}}_{ij}(\mathbf{q}^-, \mathbf{q}^+) \cdot \tilde{\mathbf{n}}_{ij} &= \sqrt{\tilde{n}_r^2 + \tilde{n}_\phi^2} \int_0^1 \left((\mathbf{J}_f + \mathbf{B}_1) n_r + \mathbf{J}_g n_\phi - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right) \Phi(s) \frac{\partial \Phi}{\partial s}(s) ds. \end{aligned}$$

(178)

Thus, by exploiting the linearity of the integral, we can give the discretization of $\tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$ in (173) as the sum of the following contributions

$$\begin{aligned} \tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = & \frac{1}{2} (\mathbf{f}(\mathbf{q}^+) + \mathbf{f}(\mathbf{q}^-) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)) \tilde{n}_r \\ & + \frac{1}{2} (\mathbf{g}(\mathbf{q}^+) + \mathbf{g}(\mathbf{q}^-)) \tilde{n}_\varphi \\ & + \frac{1}{2} (\mathbf{q}^+ + \mathbf{q}^-) \tilde{n}_t - \frac{1}{2} \mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-). \end{aligned} \quad (179)$$

Note that, whereas the discretization of $\tilde{\mathbf{F}}$ and of \mathcal{B}_{ij} can be splitted, the same cannot be done automatically for the viscosity \mathcal{V}_{ij} , whose expression depends on the chosen method (Osher-Romberg, HLL or others).

The expression in (179) results to be well balanced, provided that a well balanced expression for \mathcal{V}_{ij} is given. Indeed the first row coincides with the one dimensional flux along the radial direction for which \mathcal{B}_{ij} is given by (110)-(117)-(120)-(122) that are well balanced. With regards to the second line we know that in general it is not zero evaluated at the equilibrium because, as already pointed out at the beginning of the Section, \mathbf{g} is not zero evaluated at the equilibrium. But, if we consider, together with the flux between P_i and P_{j_1} , also the flux between P_1 and P_{j_2} and we sum them up, we can see that all the values at the equilibrium cancel exactly, thanks to the properties stated in Remark 6.11, that follows from the geometrical constraints we have imposed on our mesh. Finally, the same argument shows that also the third line goes to zero when $\mathbf{q}^- = \mathbf{Q}_i^E$ and $\mathbf{q}^+ = \mathbf{Q}_{j_1:j_2}^E$.

Viscosity term

To end with the formulation of the well balanced ALE flux (173) across this second kind of surfaces, we have to provide an expression for the viscosity $\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)$ which vanishes on stationary solutions (98)-(99).

First of all, it is easy to generalize the Osher-Romberg scheme introduced in Section 5.4. Indeed in the two dimensional ALE context the viscosity matrix introduced in (123) can be written as

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \int_0^1 |\mathbf{A}_n^V(\mathbf{Q})(\Phi(s))| \partial_s \Phi(s), \quad 0 \leq s \leq 1. \quad (180)$$

Following the same reasoning of Section 5.4 we get the following expression

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \sum_{j=1}^l \omega_j \text{sign}(\mathbf{A}_n^V(\Phi(s_j))) \frac{\mathcal{R}_j}{2\varepsilon_j}, \quad (181)$$

where

$$\begin{aligned} \mathcal{R}_j = & \tilde{\mathbf{F}}(\Phi(s_j + \varepsilon_j)) - \tilde{\mathbf{F}}(\Phi(s_j - \varepsilon_j)) \\ & + \tilde{\mathcal{B}}_{\Phi_j}(\Phi(s_j + \varepsilon_j) - \Phi(s_j - \varepsilon_j)) \end{aligned} \quad (182)$$

is discretized as explained in the 1D case above and the Romberg quadrature formula with $l = 3$ is still used. Hence, if \mathbf{q}_i^n and \mathbf{q}_{i+1}^n lie on the same stationary solution $\Phi(s) = \Phi^E(s)$ and $\mathcal{R}_j = \mathbf{0}$, $j = 1, \dots, l$. Thus, the extension to two dimensions of the Osher-Romberg scheme results to be straightforward. The only drawback is that the complete eigenstructure of the extended Jacobian matrix \mathbf{A}_n^V should be computed, which could be costly in particular when edges are not parallel to the axis (we underline that \mathbf{A}_n^V does not enjoy the property of rotational invariance that characterizes the Euler equations in Cartesian coordinates). As counter part, the method is very little dissipative and allows us to obtain very good results in convective transport problems.

The generalization of the HLL scheme is simpler. equation (132) can be rewritten in two dimensions as

$$\mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \alpha_{ij}^0 I_{ij}(\mathbf{q}^+ - \mathbf{q}^-) + \alpha_{ij}^1 \mathcal{R}_{ij}, \quad (183)$$

where I_{ij} is the identity matrix,

$$\mathcal{R}_{ij} = \mathbf{F}(\mathbf{q}_{i+1}) - \mathbf{F}(\mathbf{q}_i) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) \quad (184)$$

(which can be discretized as described in Section 6.1.2 to maintain the well balanced properties), and $\alpha_{ij}^{0;1}$ can be computed as in (134) being S^L and S^R the minimum and the maximum eigenvalues of $\mathbf{A}_n^V(\mathbf{q}_{i,i+1}^n)$.

For the same reasons stated in Section 5.4, I_{ij} must be replaced by a matrix that vanishes when a stationary solution is considered. In particular we choose the following identity modification

$$\tilde{I}_{ij} = \tilde{I}_{i+1/2} n_r + I n_\varphi, \quad (185)$$

where $\tilde{I}_{i+1/2}$ is given by (135), which we already know to vanish for stationary solutions. Moreover it follows from Remark 6.11 that when $n_\varphi \neq 0$ the term $I n_\varphi$ cancels at the equilibrium (by considering the two contributions of the neighbors P_{j_1} and P_{j_2} of P_i).

2nd order well balanced reconstruction. The missing ingredient for (172) to be well balanced up to second order is the definition of a second order well balanced reconstruction operator. As in the one dimensional case we are going to employ a combination of a smooth stationary solution together with the standard MUSCL method, hence our reconstruction will be of the form

$$\mathbf{q}_i^n(\mathbf{x}, t) = \mathbf{Q}_i^E(\mathbf{x}, t) + \mathcal{P}_i^f(\mathbf{x}, t), \quad \mathbf{x} \in C_i^n, \quad (186)$$

where, as in Section 5.4.1, $\mathcal{P}_i^f(\mathbf{x}, t)$ is the standard MUSCL method applied in order to reconstruct the fluctuations with respect to the given stationary solution computed for all the neighbors P_j^n of P_i^n as

$$\mathbf{Q}_j^{f;n} = \mathbf{Q}_j^n - \mathbf{Q}_j^{E;n}. \quad (187)$$

The expression of the reconstruction operator is

$$\mathcal{P}_i^{f,n}(\mathbf{x}, t) = \mathbf{Q}_i^{f,n} + \Phi_i \nabla \mathbf{Q}_i^{f,n}(\mathbf{x} - \mathbf{x}_{\mathbf{b}_i}^n) + \partial_t \mathbf{Q}_i^n(t - t^n), \quad (188)$$

where $\mathbf{x}_{\mathbf{b}_i}^n$ is the barycenter of cell P_i^n . To compute $\nabla \mathbf{Q}_i^{f,n}$ we use the standard MUSCL method (see [153]) together with the Barth and Jespersen limiter (see [5]), as described in Section 3.4. We would like to remark that the employed methods are standard, the novelty is in the fact that both are applied only to the fluctuations.

Finally, the term $\partial_t \mathbf{Q}_i^n$ indicates the time derivative of \mathbf{Q} and it can be computed using a discrete version of the governing equation

$$\partial_t \mathbf{Q}_i^n = (\mathbf{J}_f + \mathbf{B}_1)|_{\mathbf{x}_{\mathbf{b}_i}^n} \partial_r \mathbf{Q}(\mathbf{x}_{\mathbf{b}_i}^n) + \mathbf{J}_g|_{\mathbf{x}_{\mathbf{b}_i}^n} \partial_\varphi \mathbf{Q}(\mathbf{x}_{\mathbf{b}_i}^n), \quad (189)$$

evaluated at the barycenter $\mathbf{x}_{\mathbf{b}_i}^n$ of P_i^n . In particular the gradient of the conserved variables must be expressed as the gradient of the equilibrium plus the previously computed gradient of the fluctuation, i.e.

$$\nabla \mathbf{Q} = \nabla \mathbf{Q}_i^E + \nabla \mathbf{Q}_i^f = \nabla \mathbf{Q}_i^E + \nabla \mathbf{Q}_i^{f,n}, \quad (190)$$

in order to preserve the well balancing.

The same idea of (190) can be exploited in order to rewrite

$$\int_{C_i^n} \tilde{\mathbf{B}}(\mathbf{q}_i^n) \cdot \tilde{\nabla} \mathbf{q}_i^n \, dx dt, \quad (191)$$

where, as in Section 5.4.1, the equilibrium terms cancel and the remaining terms all contain fluctuations. So the integral can be computed through the mid-point quadrature rule which is second order accurate on the fluctuations without affecting the equilibrium.

6.2 Numerical results

6.2.1 Equilibrium preservation

First of all we want to show the accuracy of our scheme in preserving some equilibrium of interest. We consider a discontinuous equilibrium

$$\rho = 1, \text{ if } r < r_m, \quad \rho = 0.1, \text{ if } r \geq r_m, \quad (192)$$

$$u = 0, \quad v = \sqrt{\frac{Gm_s}{r}}, \quad p = 1,$$

with $r_m = 1.5$, $G = 1$, $m_s = 1$, over the computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$. In Figure 21 we depict the density profile at the equilibrium and in Table 8 we report the maximum error, committed using the HLL flux, with respect to the exact solution after long computational times over a coarse mesh, both for order 1 and 2. The equilibrium results to be perfectly preserved.

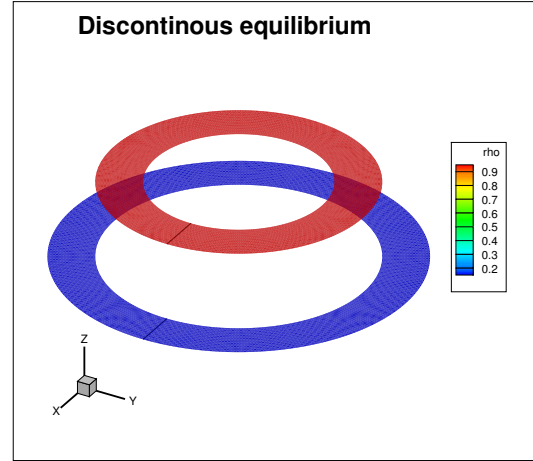


Fig. 21: Discontinuous density profile for the equilibrium solution considered in the test case of Section 6.2.1.

Table 8: Maximum error between the exact and the numerical density obtained with the first and the second order well balanced methods (using the HLL flux). We underline that similar results have been obtained using the Osher-Romberg flux and that the same precision is achieved for the velocities.

time	points 20×40	
	$\mathcal{O}1$	$\mathcal{O}2$
10	7.32E-13	4.20E-13
40	2.83E-12	8.18E-12
80	3.92E-12	1.72E-11
100	2.25E-12	1.99E-11

Then we consider a hydrostatic equilibrium without tangential velocity, so that the gravity force is perfectly balanced with the pressure gradient. The initial data is given by

$$\rho = 1, \quad \mathbf{u} = (u, v) = \mathbf{0}, \quad P = 1/r, \quad G = m_s = 1. \quad (193)$$

We consider a computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$ covered by a coarse mesh of 20×40 elements. In Table 9 we show the error between the analytical solution and our numerical solution obtained with the second order Osher-Romberg scheme. Since the scheme is exactly well balanced the errors are maintained at the order of machine precision for very long computational times. Similar results are also achieved with our well balanced HLL-type flux.

6.2.2 Order of convergence

To study numerically the order of convergence of our method we consider a smooth isentropic vortex, similar to the one

Table 9: Hydrostatic equilibrium. Maximum error in L_∞ norm between the exact solution and the numerical results for density, velocity and pressure at different times. The values refer to the second order Osher-Romberg ALE scheme, but similar results have been obtained at first order and with the HLL-type flux.

time	E_ρ	E_u	E_v	E_P
1	7.77E-15	3.29E-16	3.95E-16	3.33E-16
10	1.60E-14	3.16E-16	1.05E-15	3.33E-16
40	2.66E-14	3.58E-16	1.37E-15	3.33E-16
80	3.02E-13	1.30E-13	4.98E-14	3.87E-14

proposed in [92]. The initial condition in polar coordinates is given by

$$\rho = 1 + \delta\rho, \quad u = 0, \quad v = \delta v, \quad P = 1 + \delta P,$$

$$\delta v = r \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}}, \quad \delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi} e^{1-r^2}, \quad (194)$$

$$\delta P = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta\rho = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1,$$

with $\varepsilon = 5$, $G = 0$, $m_s = 0$ and $\gamma = 1.4$ and the computational domain defined as $[r, \varphi] = [1, 2] \times [0, 2\pi]$. The final time is $t_f = 1$. Our new scheme is able to preserve this equilibrium up to machine precision if we impose the above initial data (194) also as the equilibrium profile to be preserved. However, it is also possible to impose a different equilibrium profile to be maintained, e.g. the one given by (192). In this way, equilibrium and initial condition are not close one to the other so the method comes back to its standard order of convergence, i.e. second order. Refer to Table 10 and Figure 22 for the numerical results, which confirm that our scheme is indeed second order accurate away from the prescribed equilibrium profile. Finally, we would like to remark that we are working with a moving nonconforming grid. In Figure 23 we report an example of the final mesh configuration obtained with our Osher-Romberg scheme.

6.2.3 Riemann problem

To show the correctness of our method also in the presence of shock waves we solve a classical Riemann problem with non-vanishing angular velocity using both the well balanced HLL and Osher-Romberg ALE schemes. We consider the computational domain $[r, \varphi] = [1, 4] \times [0, 2\pi]$ and we impose the following initial conditions

$$\rho = 1, \text{ if } r < r_m, \quad \rho = 0.1, \text{ if } r \geq r_m,$$

$$u = 0, \quad v = \sqrt{G m_s / r}, \quad (195)$$

$$p = 1, \text{ if } r < r_m, \quad p = 0.1, \text{ if } r \geq r_m,$$

with $r_m = 2.5$. The results at the final computational time $t_f = 0.5$ are shown in Figure 24 where we report a cut along

Table 10: Order of convergence, isentropic vortex. We report the results obtained with our second order accurate well-balanced Osher-Romberg ALE scheme. The mesh size h is computed as the maximum incircle diameter of the elements of the final mesh. The errors refer to the L_1 norm of the difference between our numerical solution and the exact one. The last column refers to the setting where the initial data (194) are also imposed as the smooth known equilibrium to be maintained, hence in this case the scheme is accurate up to machine precision. The other results are for the setting where the code is used to evolve a different equilibrium profile (192) that does not coincide with the initial data (194), so that we can show its formal order of accuracy.

mesh size h	E_ρ , eq. (192)	$\mathcal{O}(L_1)$	E_ρ , eq. (194)
5.59E-2	1.48E-4	-	1.86E-14
2.80E-2	3.60E-5	2.04	1.45E-13
1.86E-2	1.58E-5	2.03	4.78E-13
1.40E-2	8.85E-6	2.02	5.36E-13

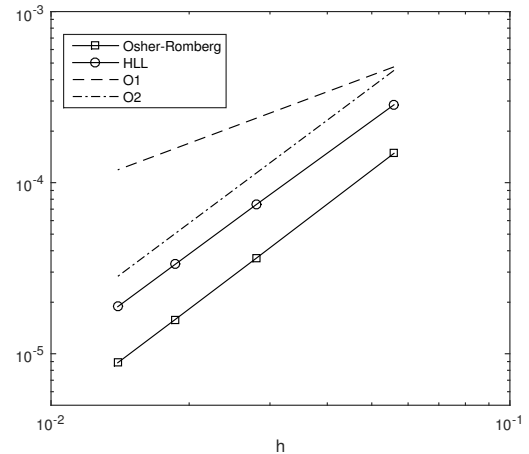


Fig. 22: Order of convergence, isentropic vortex for imposed equilibrium (eq.) given by (192), i.e. different from the initial data of the isentropic vortex (194). We report the L_1 error norm of the density obtained with our well-balanced Osher-Romberg and HLL ALE schemes. The dashed lines represent the theoretical slopes of order one and two, respectively.

$\varphi = \pi/2$ and a comparison with a one-dimensional reference solution computed on a fine grid using 1024 elements. We note a good agreement between the numerical solution obtained with the well balanced ALE scheme on moving non-conforming meshes and the reference solution also in this case where the solution is far from any equilibrium. Moreover we show the order of convergence of our method with respect to the reference solution in Figure 25: obviously it cannot reach order two because of the presence of shocks. However, the observed convergence order is higher than one.

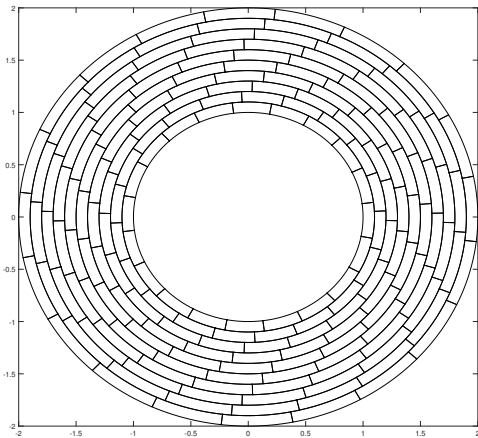


Fig. 23: Isentropic vortex, final mesh. We report the final mesh configuration at time $t_f = 1$ obtained with our Osher-Romberg scheme in the case of a very coarse mesh of 10×20 elements so that the nonconforming motion is clearly visible.

6.2.4 Noh shock test

The Noh shock test consists of a circular infinite strength shock propagating out from the origin. We have chosen this test case to prove that our method can deal also with highly supersonic flows, low pressure atmospheres and shocks of infinite strength. Consider a gas with $\gamma = 5/3$ initialized with density $\rho = 1$, radial velocity $u = -1$, tangential velocity $v = 0$, and pressure $P = 10^{-6}$ as an approximation to zero pressure. The shock wave propagates with speed $1/3$. The exact solution inside the shock region, i.e. $r \leq \frac{t}{3}$, is given by the following relations

$$\rho = 16, \quad P = 16/3, \quad u = 0, \quad v = 0, \quad (196)$$

and outside the shock region, i.e. $r > \frac{t}{3}$, by

$$\rho = 1 + \frac{t}{r}, \quad P = 0, \quad u = -1, \quad v = 0. \quad (197)$$

We consider an initial domain $[r, \varphi] \in [0, 1] \times [0, \pi/2]$. We impose periodic boundary conditions on $\varphi = 0 = \pi/2$, and we exploit the exact solution to impose the boundary conditions at $r = 0$ and on the moving outer boundary.

The presented results have been obtained with the HLL-type scheme. First we have considered the Eulerian case, hence we have imposed a zero mesh velocity. The results at time $t_f = 1.2$ obtained with the second order scheme are shown in Figure 26. Then we have employed the ALE framework moving the mesh with the local fluid velocity. Due to the absence of shear flow, the mesh remains conforming. The results obtained with the moving mesh are shown in Figure 27, where the well-known wall heating problem is

visible. Apart from the wall heating, in both the cases the method shows a good agreement with the exact solution. For what concerns the convergence performances of our code refer to Figure 25.

6.2.5 Comparison with the PLUTO code

For the following test cases that concern Keplerian discs, we compare the results obtained with our new second order well balanced Osher-Romberg scheme with the results one can obtain with the PLUTO code. PLUTO is a freely-distributed software for the numerical solution of mixed hyperbolic/parabolic systems of partial differential equations (conservation laws) targeting high Mach number flows in astrophysical fluid dynamics. The code has been systematically checked against several benchmarks available in the literature in the papers [121] and [122], using fixed uniform and AMR grids. It provides a multi-physics and multi-algorithm modular environment, where one can choose the Newtonian description for the fluid motion (HD option) and add a potential $\Phi = -\frac{Gm}{r}$ to the right hand side by setting the option BODY_FORCE equal to POTENTIAL. In this way one can study (94) within this code. Then we select POLAR GEOMETRY and we do not activate any other options.

The modular structure allows us to choose between different numerical fluxes, limiters, spatial reconstructions and time integrators. In particular, we have selected a little dissipative setting by imposing LIMITER equal to MC_LIM (the monotized central difference limiter), or sometimes equal to MINMOD_LIM (the classical minmod limiter), and using the Roe solver as numerical flux. Then we have compared our second order scheme with both a *second order* configuration of PLUTO (with LINEAR reconstruction in space and RK2 in time) and a *third order* configuration (with WENO3 reconstruction in space and RK3 in time). Finally, for the comparison we set the number of elements in PLUTO either equal to the number of elements used for our scheme, or we double it in each dimension.

We remark that within PLUTO special care is taken for the treatment of source and pressure terms when a polar (cylindrical or spherical) geometry is chosen, because in those cases the equations are discretized in angular momentum conserving form and pressure terms are treated separately. For this reason the results are more accurate than those obtained with standard finite volume techniques.

6.2.6 Mass transport in a Keplerian disc

Let us consider a steady state solution of the Euler equations with gravity which satisfies the constraints in (98)-(99) and with a constant density profile,

$$\rho_E = 1, \quad u_E = 0, \quad v_E = \sqrt{\frac{Gm_s}{r}}, \quad p = 1, \quad (198)$$

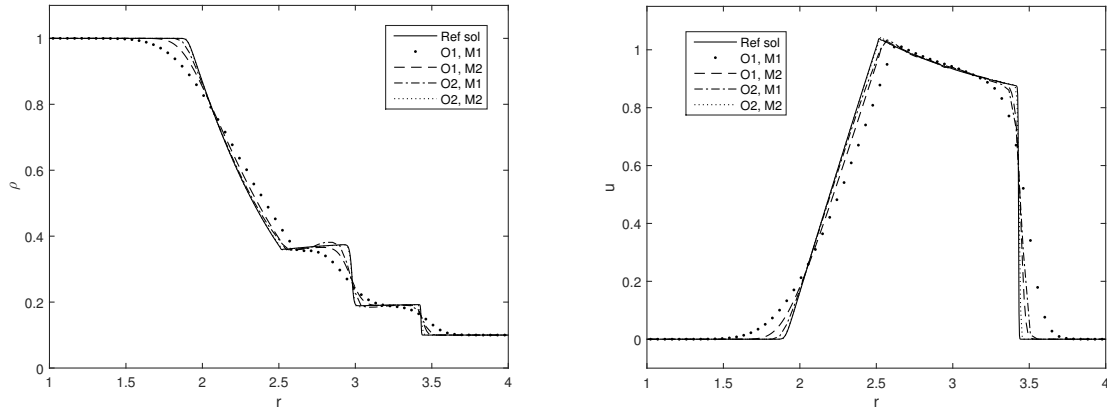


Fig. 24: Riemann problem in a 2D domain. The test have been carried out over two meshes: the first one, M_1 , with 64×20 control volumes and the second one, M_2 , with 256×40 control volumes. The reported results have been obtained using the well balanced HLL scheme with first and second order of accuracy. On the left we report the results for the density and on the right for the velocity at the final time $t_f = 0.5$. The graphs have been obtained as a 1D cut along $\varphi = \pi/2$. One can observe that the second order scheme captures the discontinuities sharply. The results are compared against a reference solution obtained with our second order well balanced HLL scheme in one space dimension with $N = 1024$.

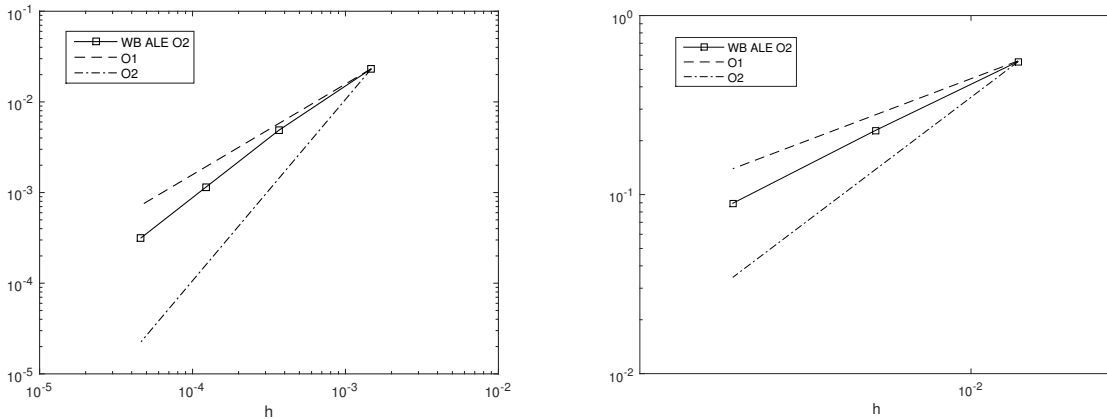


Fig. 25: Convergence results. Left: we refer to the Riemann problem (195) and compare the results obtained with our WB ALE HLL code with a fine grid reference solution. Right: we refer to the Noh shock test of Section 6.2.4 and we compare our results with the exact solution. Note that the L_1 norm of our numerical errors are depicted with squares and is compared with the theoretical slopes of order one and two (dotted lines), respectively. It is evident that the method is better than first order accurate even in presence of shocks.

over the computational domain $[r, \varphi] \in [1, 2] \times [0, 2\pi]$. At the initial time, we perturb this equilibrium solution by imposing a higher density $\rho = 2$ within the disc defined in Cartesian coordinates as $(x - 1.5)^2 + y^2 \leq (0.15)^2$.

The expected result is the transport of this density fluctuation (contact discontinuity) at different velocities which are bigger at the interior and smaller at the exterior, without any dissipation. The velocity and the pressure field should remain constant in time, according to the equilibrium solution. In Figure 28 we compare the results obtained with different numerical methods with the exact solution: Eulerian

and ALE schemes coupled or not with the well balanced Osher Romberg scheme. As expected, the Eulerian scheme is very dissipative, even when coupled with our new well balanced technique. The dissipation is evident in the angular direction, since the radial velocity in this problem is zero and the Osher scheme is a complete Riemann solver that is able to resolve steady contact waves exactly. The ALE scheme, without well balancing does not dissipate too much in the angular direction, but if it is not coupled with a proper well balanced technique, some spurious velocity oscillations appear which lead to unphysical dissipation in the radial direc-

Table 11: The results shown in this table testify that our code is able to maintain up to machine precision even moving equilibria. Indeed for the test cases presented both in Section 6.2.6 and Section 6.2.7 the L_1 norm of the difference between the numerical solution computed with our WB ALE Osher Romberg scheme and the exact stationary profiles of angular velocity v and pressure P , at the respective final times ($t = 30$ and $t = 15$), is of the order of machine precision. The other two lines refer to the results obtained with PLUTO both with second and third order of accuracy.

Method	Elem	Test Section 6.2.6		Test Section 6.2.7	
		$\ v-v_E\ _{L_1}$	$\ P-P_E\ _{L_1}$	$\ v-v_E\ _{L_1}$	$\ P-P_E\ _{L_1}$
WB ALE Osher-R.	100×200	2.17E-12	7.19E-14	2.13E-12	6.36E-14
PLUTO O2 minmod	100×200	5.56E-7	2.36E-6	5.44E-7	9.89E-6
PLUTO O3 mc_lim	200×400	1.30E-7	5.28E-7	1.49E-7	2.44E-6

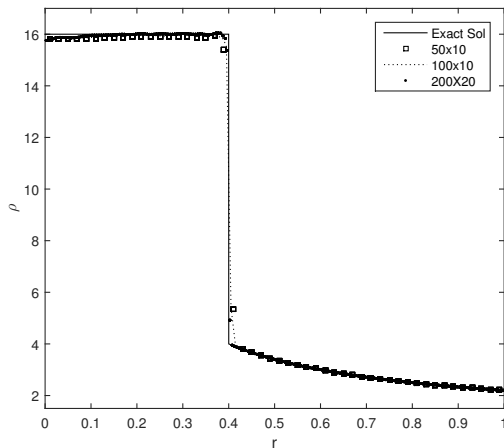


Fig. 26: Noh shock test. We show the numerical results obtained with our second order HLL-type flux at time $t_f = 1.2$ on three fixed grids with respectively 50×10 , 100×10 and 200×20 elements. In the figure the density profile ρ has been depicted along the radial direction r , compared with the exact solution.

tion and which also produce some oscillations on the density profiles, which are evident even for short computational times.

The coupling between the two techniques reduces the dissipation both in the radial and in the angular directions. In our computations we have observed that for this test problem the error in the pressure and in the velocity field was always of the order of *machine precision*, since the advection of a contact discontinuity does *not* affect the equilibrium of pressure and velocity. We emphasize that this property of conserving even moving equilibria (density is not constant in time here) is anything else than trivial to achieve and to the best knowledge of the authors, the scheme presented here is the first Finite Volume method to achieve it. Referring to Table 11, one can notice that indeed the precision achieved by our code on angular velocity and pressure is of the order of machine precision (even at time $t = 30$), where instead this is not the case for various PLUTO configurations.

Finally, we report the results obtained with PLUTO by selecting the configuration setting described in Section 6.2.5 with the MC_LIM. First, in Figure 29 we use the described second order method and 30×350 elements. Then, in Figure 30 we use the third order method and 60×700 elements. In both the cases the density is dissipated faster than with our method: this shows that it is not a finer grid or a higher order of accuracy that can solve this type of problem, but a very specific treatment of the equilibrium together with the Lagrangian framework proposed in this paper.

6.2.7 Keplerian disc with density perturbations

For this test we have considered the equilibrium profile in (198) and we have added a periodic perturbation over the density profile as follows

$$\rho = \rho_E + A \sin(k_1 \varphi)(0.25 - |r_m - r|), \quad r \in [r_1, r_2] \quad (199)$$

with $A = 0.5$, $k_1 = 12$, $r_1 = 1.25$, $r_2 = 1.75$, $r_m = 1.5$. The goal of this test is to show that our well balanced ALE scheme is able to maintain the equilibrium pressure and velocity exactly and that the numerical method does *not* generate any spurious numerical perturbations of pressure and velocity that would usually lead to Kelvin-Helmholtz type flow instabilities for density fluctuations combined with shear flow as in the above setup. In Figure 31 we show the evolution of the perturbations at different times. They are properly transported with different velocities with only very little numerical dissipation. As in the previous case we stress that the velocity and pressure remain at the *equilibrium solution* up to *machine accuracy* throughout the entire simulation. No spurious Kelvin-Helmholtz instabilities are generated, since the equilibrium pressure and velocity are exactly maintained for arbitrary long simulation times.

Finally, we compare our result at time $t = 15$ with the results obtained with PLUTO, refer to Figure 32. For the visualization we have always used the software Tecplot and the same colormap; even if the results look similar, one can notice that to obtain the same resolution of our code (left image of the panel) we need the third order version of PLUTO and a finer mesh (last image of the panel). We stress that our

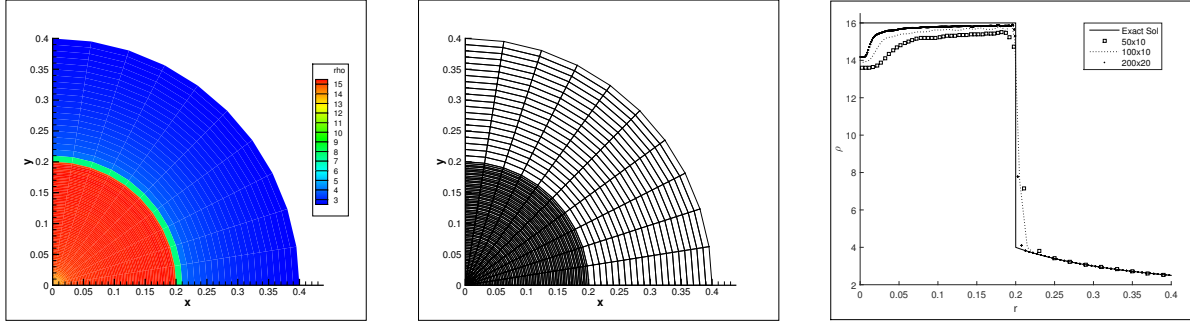


Fig. 27: Noh shock test. We show the density profile (left) and the final mesh (center) obtained with the second order ALE HLL-type scheme at time $t_f = 0.6$, using a moving grid of 100×10 elements. On the right we compare the density profile along the radial direction r with the exact solution for three different meshes with respectively 50×10 , 100×10 and 200×20 elements.

code maintains u and P up to machine precision, whereas PLUTO produces standard numerical errors, see Table 11.

6.2.8 Keplerian disc with Kelvin-Helmholtz instabilities I

Let us consider an equilibrium solution which satisfies the equilibrium constraints in (98)-(99) so that

$$\rho_E = \rho_0 + \rho_1 \tanh\left(\frac{r-r_m}{\sigma}\right), \quad u_E = 0, \quad v_E = \sqrt{\frac{Gm_s}{r}}, \quad p_E = 1, \quad (200)$$

with $G = 1$, $m_s = 1$, $\rho_0 = 1$, $\rho_1 = 0.25$, $r_m = 1.5$ and $\sigma = 0.01$. It shows a steep gradient in the density for $r \rightarrow 1.5$. We consider as computational domain a ring sector with radius $r \in [1, 2]$ and $\varphi \in [0, \pi/2]$. For the boundary conditions we exploit the exact solution when $r = 1, 2$, and we impose periodic boundary conditions for $\varphi = 0, \pi/2$.

As confirmed by the previous tests, our well balanced ALE scheme is able to maintain the equilibrium up to machine precision for very long computational times. So we can study with high accuracy the evolution of the perturbations over the density, the radial velocities and the pressure prescribed by the following initial condition

$$\begin{cases} \rho = \rho_E + A \rho_0 \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \\ u = u_E + A \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \quad v = v_E, \\ p = p_E + A \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \end{cases} \quad (201)$$

with $A = 0.1$, $k = 8$, $s = 0.005$. The computational results are depicted in Figure 33. In particular, for this flow configuration with physical perturbations in all flow quantities we observe the appearance of Kelvin-Helmholtz instabilities and a very good resolution of the developing vortices, which is achieved thanks to the ALE technique and despite the rather coarse mesh of 100×200 elements used here.

Moreover we have compared our well balanced ALE scheme with a well balanced *Eulerian* method on a fixed grid, which appears to be quite diffusive, and a *not* well balanced ALE scheme, which produces visible spurious oscillations in the density profile. The results are presented in Figure 34 and, once again, they show that it is indeed the coupling between the well balanced techniques and the moving mesh framework that allows to achieve a high resolution on small perturbations around an equilibrium solution for very long computational times.

We also compare our numerical results at time $t = 37.5$ with those obtained by PLUTO, see Figure 35. In order to obtain the same accuracy of our new second order well balanced Osher Romberg ALE scheme (left image of the panel) one needs the third order version of PLUTO on a finer mesh (last image of the panel).

6.2.9 Keplerian disc with Kelvin-Helmholtz instabilities II

We finally consider another equilibrium solution which satisfies the equilibrium constraints in (98)-(99) and which reads

$$\rho_E = r, \quad u_E = 0, \quad v_E = \sqrt{\frac{Gm_s}{r}}, \quad p_E = 1, \quad (202)$$

with $G = 1$, $m_s = 1$ and $r_m = 1.5$. With respect to the previous example, here the density profile is linear. However, also in this example we expect the Kelvin-Helmholtz instabilities to arise if some perturbations are added to the stationary profile. The computational domain and the boundary conditions are chosen as before. The initial condition used in this test problem reads

$$\begin{cases} \rho = \rho_E + A \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \\ u = u_E + A \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \quad v = v_E, \\ p = p_E + A \sin(k\varphi) \exp\left(-\frac{(r-r_m)^2}{s}\right), \end{cases} \quad (203)$$

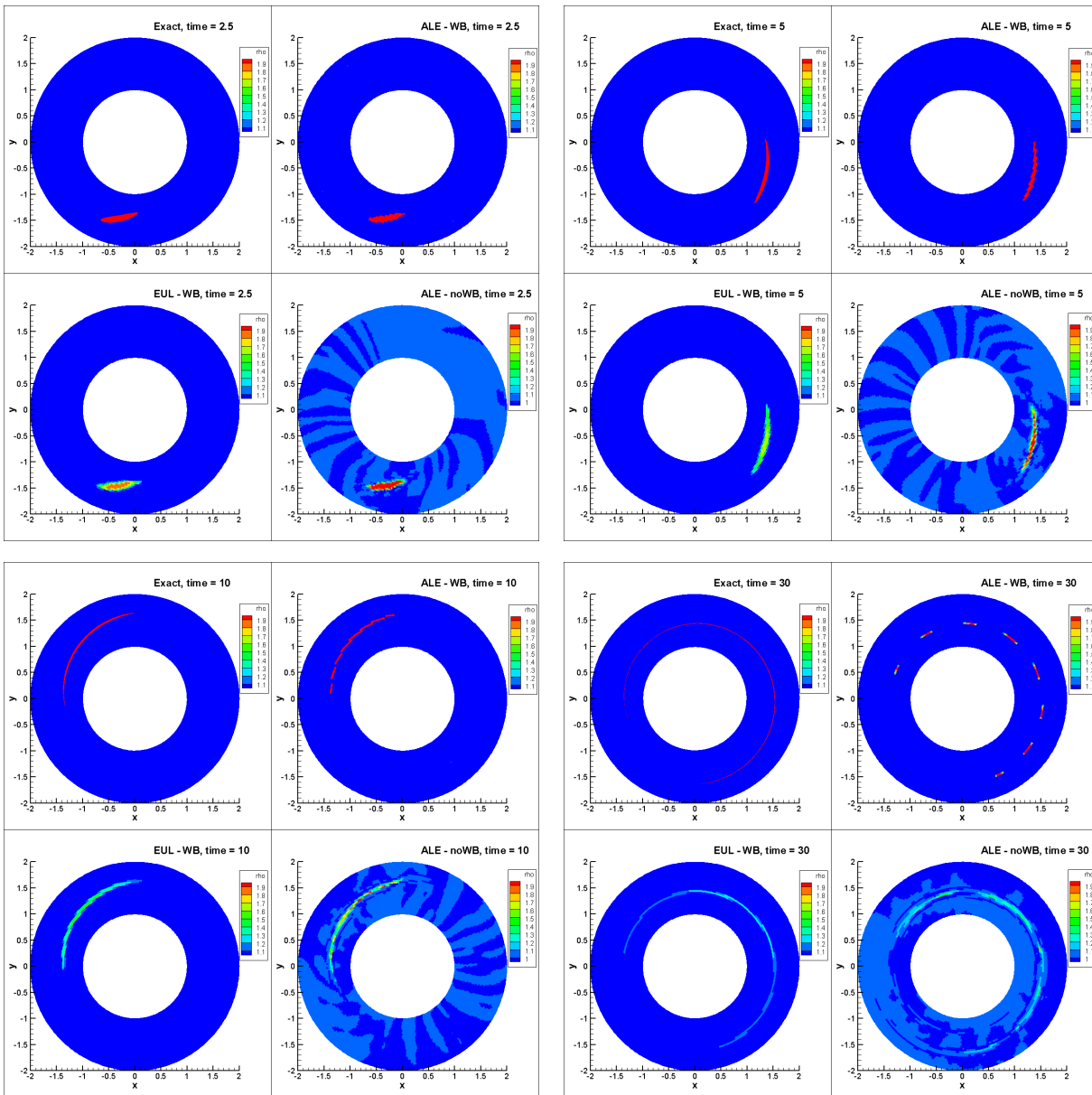


Fig. 28: We compare the exact solution with the numerical solutions obtained with different methods at times $t = 2.5$ (top-left), $t = 5$ (top-right), $t = 10$ (bottom-left), $t = 30$ (bottom-right). For all the cases the employed numerical flux is an Osher-type flux. The Lagrangian algorithms show their ability in reducing the viscosity along the angular direction. The well balanced methods do not diffuse the quantities in the radial direction. When coupled together (top-right of each square) we obtain a result very close to the exact solution (top-left of each square). We want to remark that in the well balanced ALE case (top-right of each square), the quantity with higher density remains in the same cells in which it is confined at the initial time since the method is very little diffusive in any direction and the differential rotation is treated in a nonconforming way. Moreover, only the well balanced ALE scheme is able to maintain the concentration of the higher density gas.

with $A = 0.1$, $k = 8$, $s = 0.005$, i.e. we are again solving a problem that is close to an equilibrium and therefore difficult to solve with standard numerical techniques that are not well balanced. The computational results are depicted in Figure 36. Again we observe the appearance of Kelvin-Helmholtz instabilities that are well resolved also on a rather

coarse mesh, without any visible spurious numerical oscillations. Finally, we compare once again our code with results obtained with PLUTO, refer to Figure 37. A similar resolution of the vortices is obtained with our second order code and the third order version of PLUTO with a finer mesh (refer to Section 6.2.5 for the details on the PLUTO configu-

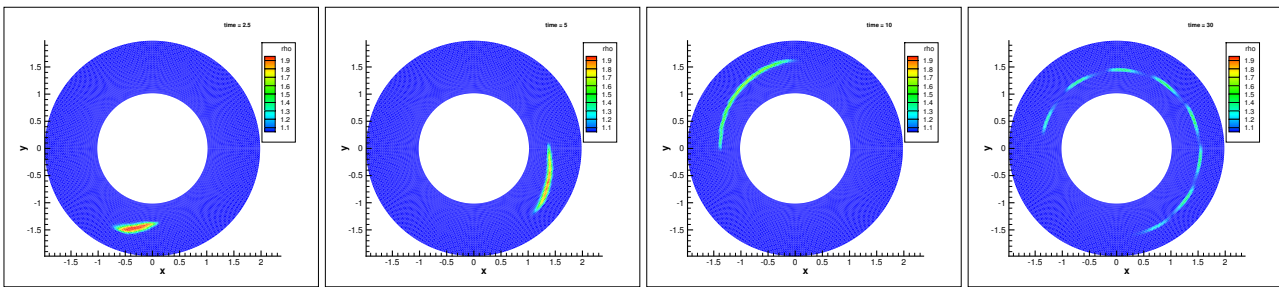


Fig. 29: Results obtained with PLUTO, using the Roe solver combined with the `mc_lim` limiter, linear reconstruction in space and RK2 in time on a grid of 30×350 elements. One can observe that the results are more dissipative compared to those shown in Fig. 28.

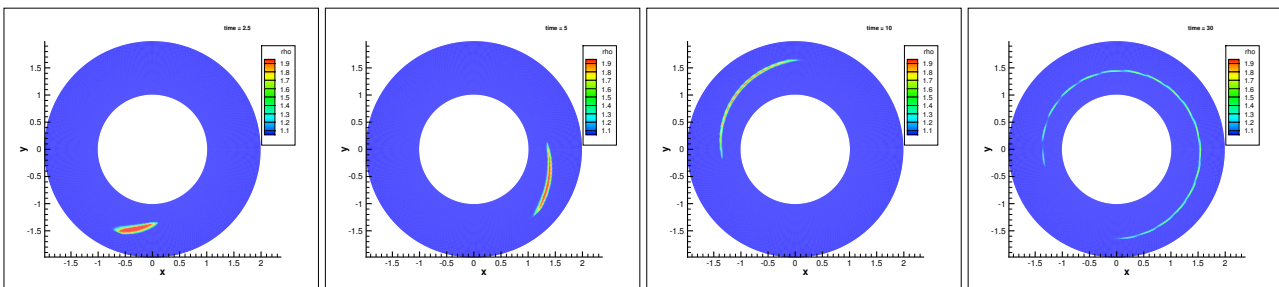


Fig. 30: Results obtained with PLUTO, using the Roe solver, a third order WENO reconstruction in space combined with the `mc_lim` limiter and a third order RK3 time integrator on a grid of 60×700 elements.

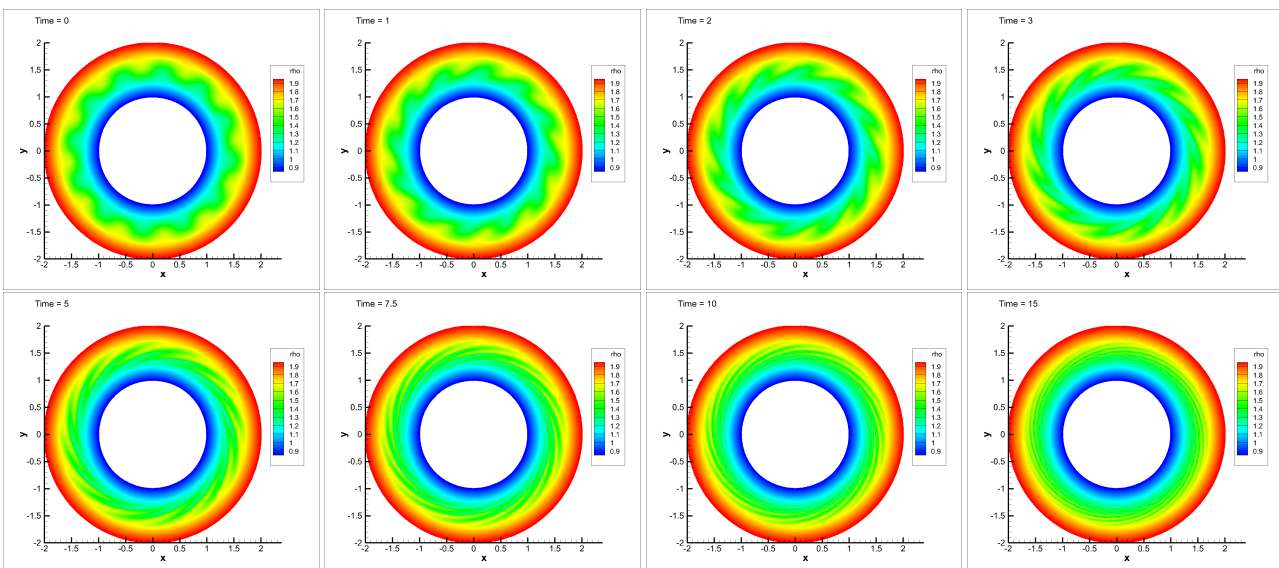


Fig. 31: Evolution of periodic density perturbations in an equilibrium disc obtained with the well balanced ALE scheme with Osher-Romberg flux. The perturbations are perfectly convected (with an inner velocity bigger than the outer one), and no spurious Kelvin-Helmholtz instabilities are generated, even after long computational times.

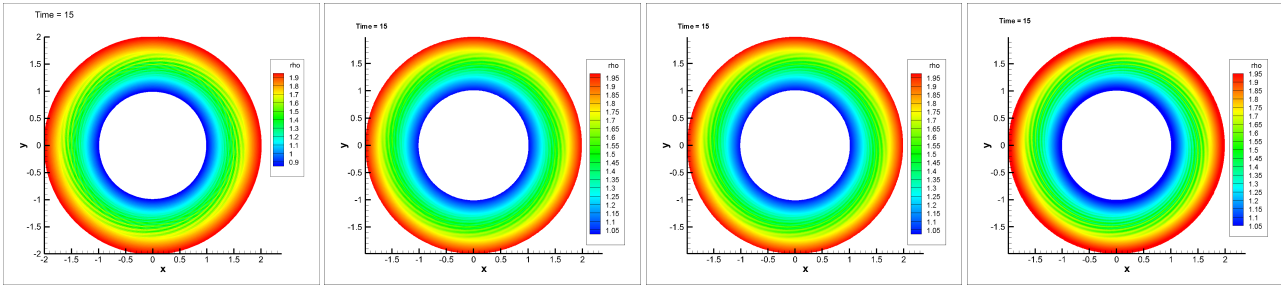


Fig. 32: Method comparison at time $t = 15$. The first image is obtained with our code and 50×500 elements. The second and the third one with PLUTO using 50×500 elements and respectively a second order scheme with `mc_lim` limiter and a third order scheme with `minmod_lim` limiter. The last image is obtained with the third order version of PLUTO using `mc_lim` and 100×1000 elements. All images are drawn with the same color map. Even if the results are similar, one can notice that to obtain the same resolution of our code we need the third order version of PLUTO and a finer mesh.

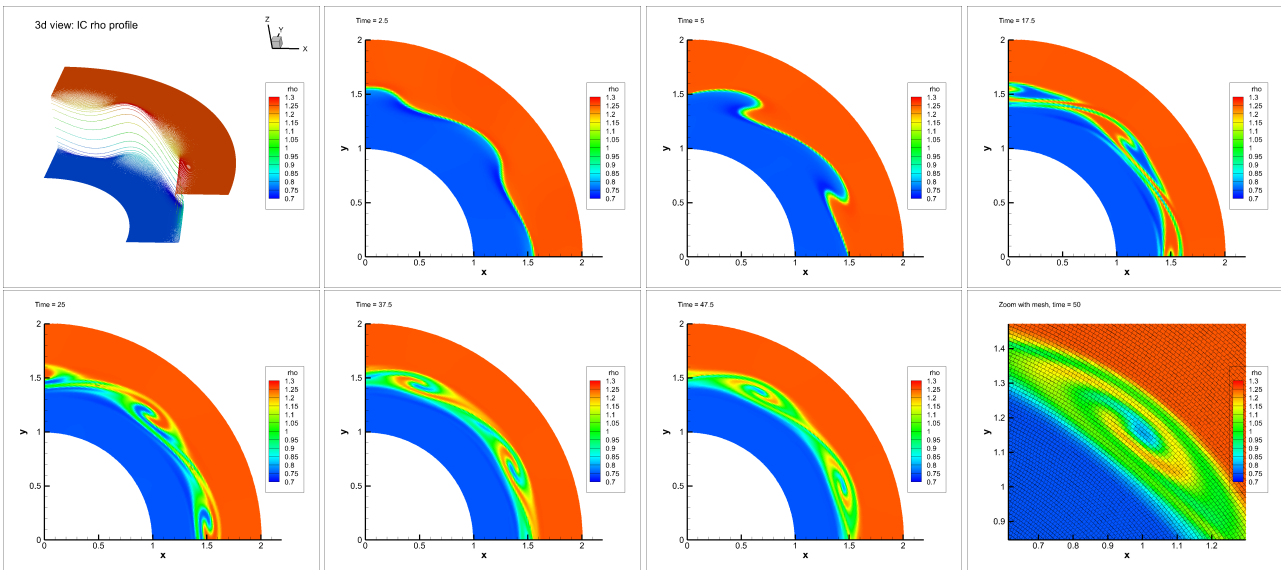


Fig. 33: Kelvin-Helmholtz instabilities I. In the panel we show the evolution of the imposed periodic perturbations at different times. The results have been obtained with our second order well balanced ALE Osher-Romberg scheme over a grid with 100×200 control volumes.

ration we have chosen). In this case we want to underline also that our code avoids other oscillations that instead can be noticed in the images obtained with PLUTO.

6.3 WB ALE for the shallow water equations

Coming back to the steady vortex test case presented in Section 6.3 we know that our method is able to maintain a high quality mesh even in the case of strong shear flows, to preserve the physical properties of the system (mass, momentum, energy) for very long computational times, and that it is very little dissipative for contact discontinuities. In this Section we want to extend the algorithm in such a way that in addition it can preserve also exactly certain relevant and non-trivial equilibrium solutions. In particular, our interest

is focused on the shallow water equations in polar coordinates given by (69), and to stationary solutions characterized by the equilibrium between the centrifugal and the gravitational forces. Since the two forces appear one in the flux and the other one in the source term, it is difficult to maintain the exact balancing until they are discretized with different techniques. So once again (as in Section 5.2.3), we rewrite the source terms via non-conservative products so that all the terms connected with the equilibrium could be treated together. By adding as auxiliary variables the radius r and the bottom topography b such that the free surface is $\eta(r, \varphi) = b + h(r, \varphi)$, our equations can be cast in form (1)

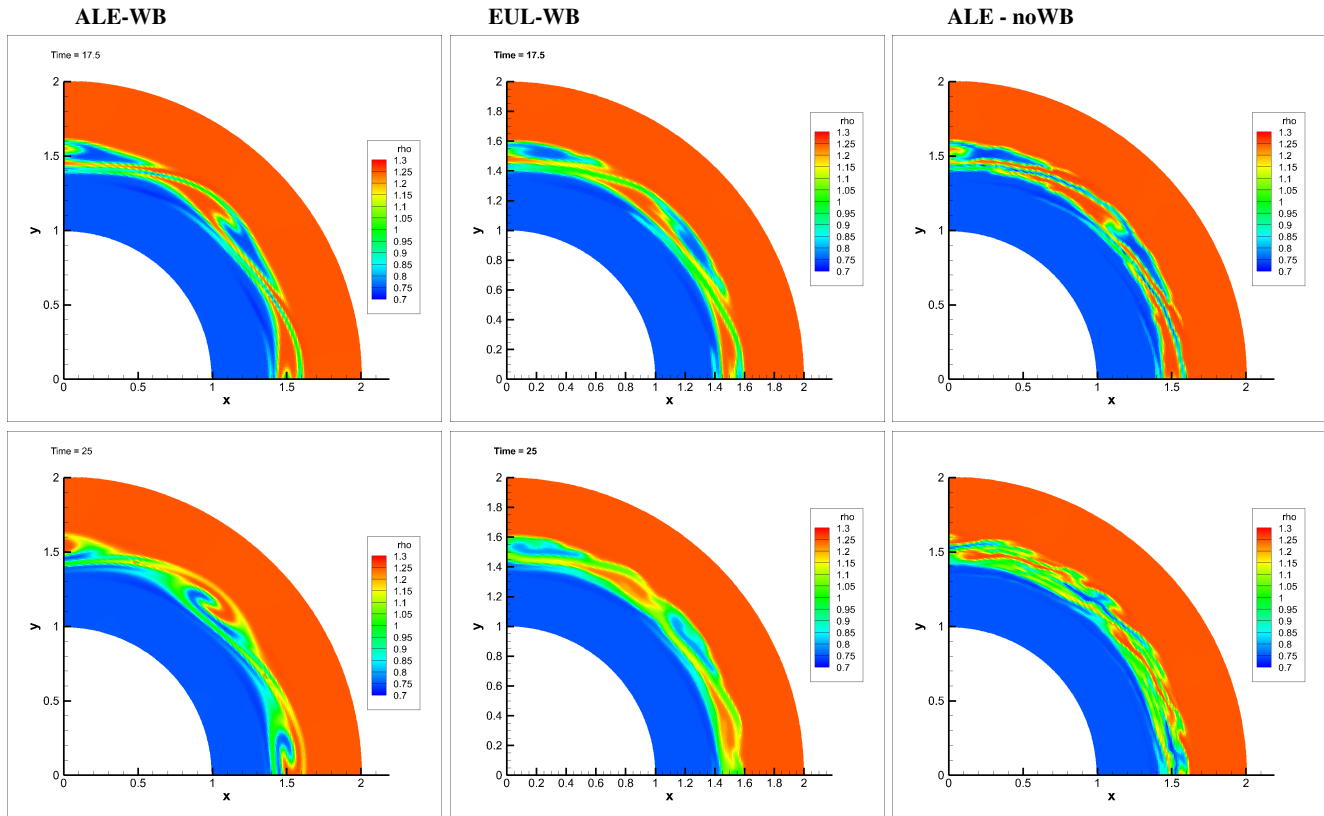


Fig. 34: Kelvin-Helmholtz instabilities I. In the panel we show the obtained solution for the density profile at time $t = 17.5$ (first row) and time $t = 25$ (second row). The results presented in the first column have been obtained using the Osher-Romberg well balanced ALE scheme. The ones in the second column have been obtained using a zero velocity mesh (Eulerian case) and the well balanced Osher-Romberg scheme. The third column is obtained with a standard nonconforming ALE scheme (i.e. using the ALE Osher type flux without well balancing). One can appreciate that it is really the coupling between the ALE and the well balancing that allows to achieve this high resolution.

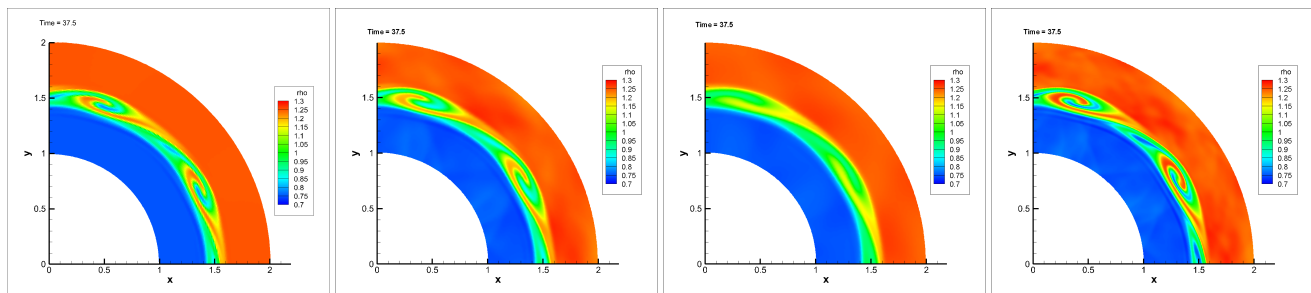


Fig. 35: Kelvin-Helmholtz instabilities I. Method comparison at time $t = 37.5$. The first image is obtained with our code and 100×200 elements. The second and the third one with PLUTO, 100×200 elements and respectively a second order scheme with `mc_lim` limiter and a third order scheme and `minmod_lim` limiter. The last image is obtained with the third order version of PLUTO using `mc_lim` and 200×400 elements. All images are drawn with the same color map.

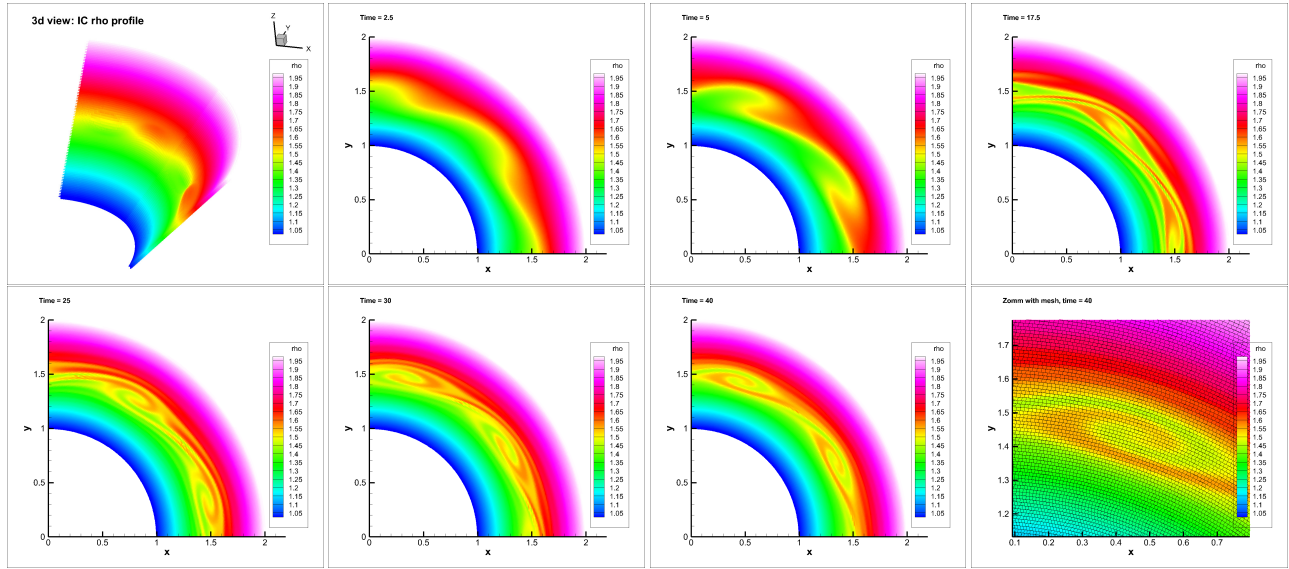


Fig. 36: Kelvin-Helmholtz instabilities II. In the panel we show the evolution of the imposed periodic perturbations at different times. The results have been obtained with our second order Osher-Romberg scheme over a grid with 100×200 control volumes.

with

$$\mathbf{Q} = \begin{pmatrix} rh \\ rhu_\rho \\ rhu_\varphi \\ rb \\ r \end{pmatrix}, \mathbf{f} = \begin{pmatrix} rhu_\rho \\ rhu_\rho^2 \\ rhu_\rho u_\varphi \\ 0 \\ 0 \end{pmatrix}, \mathbf{g} = \begin{pmatrix} hu_\varphi \\ hu_\rho u_\varphi \\ hu_\varphi^2 + \frac{1}{2}gh^2 \\ 0 \\ 0 \end{pmatrix}, \quad (204)$$

$$\mathbf{S} = \mathbf{0}, \mathbf{B}_1 \cdot \nabla \mathbf{Q} = \begin{pmatrix} 0 \\ grh \frac{\partial \eta}{\partial r} - hu_\varphi^2 \frac{\partial r}{\partial r} \\ hu_\rho u_\varphi \frac{\partial r}{\partial r} \\ 0 \\ 0 \end{pmatrix}, \mathbf{B}_2 \cdot \nabla \mathbf{Q} = \mathbf{0} \quad (205)$$

6.3.1 First order WB ALE scheme

In what follows we explain how to adapt the first order Osher-Romberg scheme proposed for the Euler equations to (204), in particular with the aim of preserving equilibrium solutions such that

$$u_\rho = 0, \quad \frac{\partial u_\rho}{\partial \varphi} = \frac{\partial u_\varphi}{\partial \varphi} = \frac{\partial \eta}{\partial \varphi} = 0, \quad \text{and} \quad \frac{\partial \eta}{\partial r} = \frac{u_\varphi^2}{gr}. \quad (206)$$

For the domain discretization let us consider the same setting introduced in Section 6.1.1, and for the basic form of a path-conservative ALE scheme refer to (172). The core of the scheme is the design of the well balanced space-time flux function, whose final expression is

$$\tilde{\mathbf{D}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} = \frac{1}{2} (\tilde{\mathbf{F}}(\mathbf{q}^+) + \tilde{\mathbf{F}}(\mathbf{q}^-) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} \mathcal{V}_{ij}(\mathbf{q}^+ - \mathbf{q}^-), \quad (207)$$

with the same notation of Section 6.1.2. We can also follow the same reasoning of Section 6.1.2 regarding the well balancing in the angular direction and Sections 5.4 - 6.1.2 for the Osher-Romberg viscosity term. So the missing ingredients for the first order scheme are the well balanced path and the definition of $\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)$.

Here we propose a family of paths that is connected to the known equilibrium profiles for the free surface η and the angular velocity u_φ , whereas for the other variables the segment path is sufficient (because they assume simple profiles at the equilibrium). Let $\Phi^E(s; \mathbf{q}_E^-, \mathbf{q}_E^+)$ be a reparametrization of a stationary solution given by (206) that connects the two equilibrium states \mathbf{q}_E^- with \mathbf{q}_E^+ , then we define $\Phi(s; \mathbf{q}^-, \mathbf{q}^+)$ as follows

$$\Phi(s; \mathbf{q}^-, \mathbf{q}^+) = \Phi^E(s; \mathbf{q}_E^-, \mathbf{q}_E^+) + \Phi^f(s; \mathbf{q}_f^-, \mathbf{q}_f^+), \quad (208)$$

where $\mathbf{q}_f^- = \mathbf{q}^- - \mathbf{q}_E^-$ and $\mathbf{q}_f^+ = \mathbf{q}^+ - \mathbf{q}_E^+$ and

$$\Phi^f(s; \mathbf{q}_f^-, \mathbf{q}_f^+) = \mathbf{q}_f^- + s(\mathbf{q}_f^+ - \mathbf{q}_f^-). \quad (209)$$

That is Φ^f is a segment path on the *fluctuations* with respect to a stationary solution. With this choice, it is clear that if \mathbf{q}^- and \mathbf{q}^+ lie on the same stationary solution satisfying (206), then $\mathbf{q}_f^- = \mathbf{q}_f^+ = \mathbf{0}$ and Φ , reduces to Φ^E . In such situation we have that $\mathbf{f}(\mathbf{q}^+) = \mathbf{f}(\mathbf{q}^-) = \mathbf{0}$ and

$$\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \int_0^1 \mathbf{B}_1(\Phi(s; \mathbf{q}^-, \mathbf{q}^+)) \tilde{n}_r \frac{\partial \Phi^E}{\partial s}(s; \mathbf{q}^-, \mathbf{q}^+) ds \quad (210)$$

$$= \mathbf{0}.$$

Therefore $\mathbf{f}(\mathbf{q}^+) - \mathbf{f}(\mathbf{q}^-) + \mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = \mathbf{0}$.

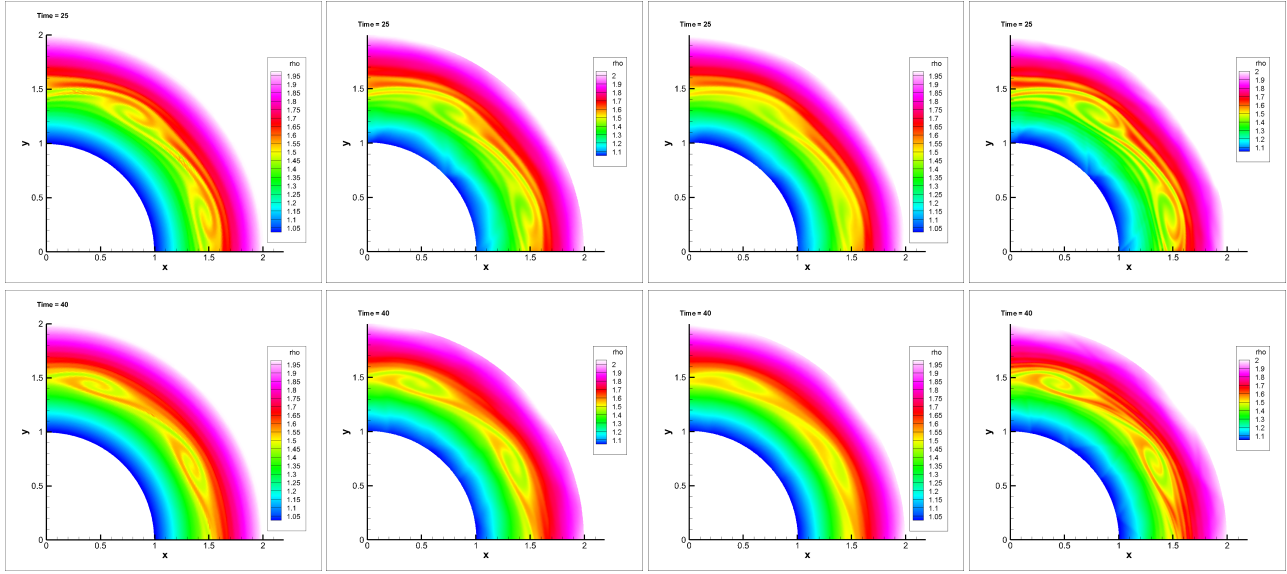


Fig. 37: Kelvin-Helmholtz instabilities II. Method comparison at time $t = 25$ (first row) and at time $t = 40$ (second row). The first images are obtained with our code and 100×200 elements. The second and the third ones with PLUTO, 100×200 elements and using respectively a second order (mc_lim) and a third order (minmod_lim) scheme. The last images are obtained with PLUTO using a third order scheme (mc_lim) and 200×400 elements. All images are drawn with the same color map. The vortices have a similar resolution in the leftmost and rightmost images.

Let us now define $\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-)$ in the general case, where \mathbf{q}^+ and \mathbf{q}^- do not lie on a stationary solution. In such case we have that

$$\mathcal{B}_{ij}(\mathbf{q}^+ - \mathbf{q}^-) = (b_1^{ij}, b_2^{ij}, b_3^{ij}, b_4^{ij}, b_5^{ij})^T. \quad (211)$$

It is clear from the definition of \mathbf{B} that $b_1^{ij} = b_4^{ij} = b_5^{ij} = 0$. What is interesting is the discretization of the second term that can be rewritten as

$$\begin{aligned} & \left(grh \frac{\partial \eta}{\partial r} - hu_\varphi^2 \frac{\partial r}{\partial r} \right) \tilde{n}_r \\ &= \left(grh \frac{\partial \eta}{\partial r} - grh \left[\int \frac{u_\varphi^2}{rg} dr \pm \int \frac{u_{\varphi,E}^2}{rg} dr \right]_r \right) \tilde{n}_r, \end{aligned} \quad (212)$$

where $u_{\varphi,E}$ is any known profile for the angular velocity at the equilibrium; moreover call $\zeta(r)$ a primitive of $\frac{u_{\varphi,E}^2}{rg}$, i.e.

$\zeta(r) = \int \frac{u_{\varphi,E}^2}{rg} dr$. In this way we obtain that

$$b_2^{ij} = \int_0^1 \left(g\Phi_{rh}(s) \frac{\partial \Phi_\eta(s)}{\partial s} - g\Phi_{rh}(s) \frac{\Phi_A(s)}{rg} \frac{\partial \Phi_r(s)}{\partial s} \right) \tilde{n}_r ds, \quad (213)$$

where for variables r and rh we can employ a standard segment path to connect the left and the right states

$$\begin{aligned} \Phi_r(s) &= \Phi_r(s; r^-, r^+) = r^- + s(r^+ - r^-), \\ \Phi_{rh}(s) &= \Phi_{rh}(s; (rh)^-, (rh)^+) = (rh)^- + s((rh)^+ - (rh)^-). \end{aligned} \quad (214)$$

Instead, following the idea in (208) and considering the terms in (212) we define

$$\Phi_A(s) = \Phi_A(s; u_\varphi^-, u_\varphi^+) = \Phi_{\zeta_r}^E(s) + \frac{\Phi_{u_\varphi}^f(s)}{rg} \quad (215)$$

which exploits the reparametrization of $\zeta(r)$ at the equilibrium and approximates with a segment path the fluctuations of the angular velocity

$$\Phi_{u_\varphi}^f(s) = \Phi_{u_\varphi}^f(s; u_{\varphi,f}^-, u_{\varphi,f}^+) = \frac{1}{rg} (u_{\varphi,f}^- + s(u_{\varphi,f}^+ - u_{\varphi,f}^-)). \quad (216)$$

A similar approach is used for $\Phi_\eta(s)$ defined as

$$\Phi_\eta(s) = \Phi_\eta(s; \eta^-, \eta^+) = \Phi_\eta^E(s) + \Phi_\eta^f(s). \quad (217)$$

Taking into account that

$$\int_0^1 \left(g\Phi_{rh}(s) \frac{\partial \Phi_\eta^E(s)}{\partial s} - g\Phi_{rh}(s) \frac{\Phi_{\zeta_r}^E(s)}{rg} \frac{\partial \Phi_r(s)}{\partial s} \right) \tilde{n}_r ds = 0, \quad (218)$$

b_2^{ij} could be rewritten as follows

$$b_2^{ij} = \int_0^1 \left(g\Phi_{rh}(s) \frac{\partial \Phi_\eta^f(s)}{\partial s} - g\Phi_{rh}(s) \frac{\Phi_{u_\varphi}^f(s)}{rg} \frac{\partial \Phi_r(s)}{\partial s} \right) \tilde{n}_r ds. \quad (219)$$

Note that $\frac{\partial \Phi_\eta^f(s)}{\partial s} = \eta_f^+ - \eta_f^- = \Delta \eta_f$ and $\frac{\partial \Phi_r(s)}{\partial s} = r^+ - r^- =$

Δr , therefore b_2^{ij} reduces to

$$b_2^{ij} = \left(g(rh)_{ij} \Delta \eta_f - g(rh)_{ij} \left(\frac{u_\varphi^2 - (u_\varphi^E)^2}{rg} \right)_{ij} \Delta r \right) \tilde{n}_r \quad (220)$$

where we have employed the mid point rule to approximate the integrals and the following notation holds $(\cdot)_{ij} = (\cdot_i + \cdot_j)/2$. Finally, term b_3^{ij} could be approximated in the same way. Nevertheless, as this terms explicitly depends on u_ρ and we are interested to preserve equilibria with $u_\rho = 0$, a more simple approach could be used. Thus, b_3^{ij} could be defined as

$$b_3^{ij} = \left((rhu_\rho)_{ij} (u_\varphi)_{ij} \Delta r \right) \tilde{n}_r, \quad (221)$$

which vanishes when $u_\rho = 0$. For any other detail of the scheme one can refer to the previous Sections and adapt the method introduced for the Euler equations to this new context. Indeed the method is very general and easy to be generalized to different sets of equations and equilibria.

6.4 Numerical results

In this Section, first we want to show that the well balanced method works in general situations and not only close to the equilibria of the system. In this way, it will be clear that it can be applied in any context without corrupting the standard characteristics of the scheme, and it will perform better than classical schemes when near to a prescribed equilibrium. Then, we will see that the coupling between our nonconforming techniques with the well balanced strategy allow us to study the vortex flow of Section 3.6.5 even for longer periods of time.

6.4.1 Riemann problem

To show the correctness of our method in the presence of shocks we solve a classical Riemann problem with our well balanced Osher-Romberg ALE scheme. We consider the system of equation in (69), and as computational domain $[r, \varphi] = [1, 5] \times [0, 2\pi]$. We impose the following initial conditions

$$\begin{cases} h = 1, & \text{if } r < r_m, & h = 0.125, & \text{if } r \geq r_m, \\ u_\rho = u_\varphi = 0 \end{cases} \quad (222)$$

with $r_m = 3$. The results at the final computational time $t_f = 0.4$ are shown in Figure 38, where we report a cut along $\varphi = \pi/4$. The method, even if it is set up to preserve the smooth stationary profile described in Section 3.6.5, converges properly to the reference solution of this problem, despite the presence of discontinuities.

6.4.2 Steady vortex in equilibrium

Test A. Let us consider again the test case of Section 3.6.5, with the initial condition of (79). The coupling between our novel nonconforming ALE scheme together with the well balanced techniques gives us, even after a very long computational time, a good mesh quality (see Figure 39) and a numerical solution equal to the exact one up to machine precision (refer to Table 12).

Note that we have employed a mesh of squares with the constraints that interfaces lie over straight lines with constant radius. This automatically implies that each square of the mesh has two edges parallel to the φ -axis: over this kind of edges the \mathbf{g} component of the flux does not play any role, and so the method is well balanced simply because the \mathbf{f} component of the flux is zero for stationary vortex-type solutions and (211) has been proved to be discretized in the correct way. The other two edges are parallel between them, so at the equilibrium, fluxes through them cancel.

Test B. Moreover, to show that the method is able to preserve any known stationary solution that satisfies the constraint in (206), we have performed a similar test but starting from a different stationary condition

$$h(r, \varphi, 0) = \frac{r^2}{2g}, \quad u_\rho(r, \varphi, 0) = 0, \quad u_\varphi(r, \varphi, 0) = r, \quad (223)$$

over the same computational domain $\Omega(r, \varphi) = [0.2, 2] \times [0, 2\pi]$. Even in this case the numerical solution remains close to the exact one up to machine precision for very long times, as also shown in Table 12.

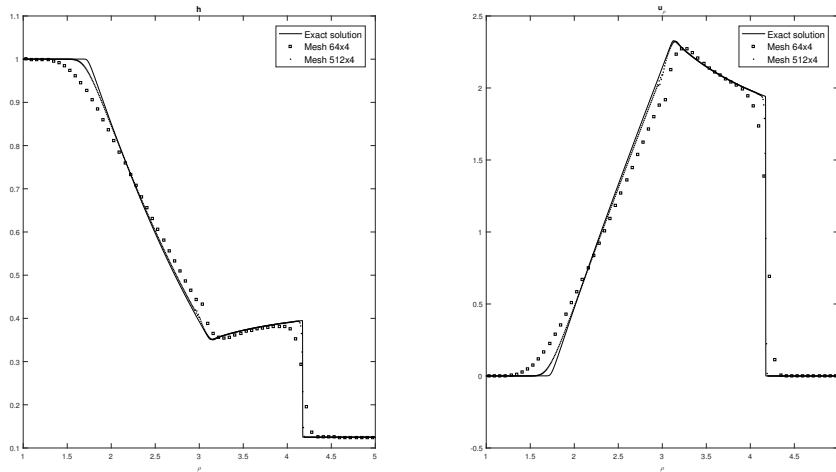


Fig. 38: Comparison between the exact and the numerical solution for the Riemann problem. The numerical solution is obtained with the well balanced scheme of order one with two different meshes (a coarser and a finer one). On the left we show the water level h and on the right the radial velocity u_r for $r \in [1, 5]$ at a fixed angle $\varphi = \frac{\pi}{4}$.

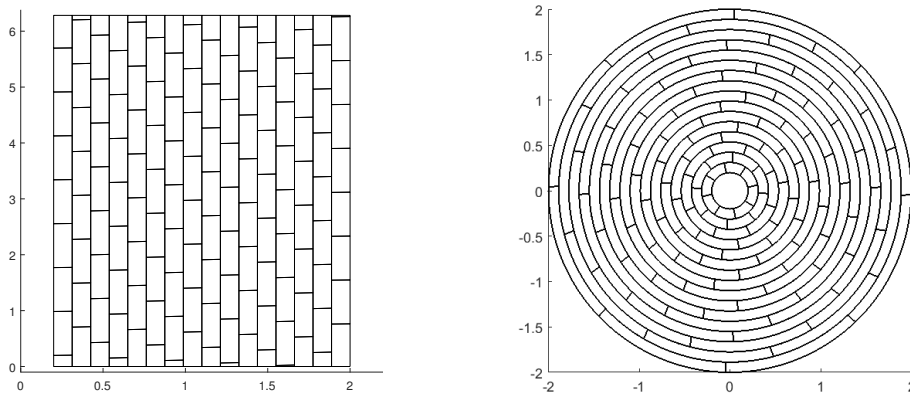


Fig. 39: Stationary vortex in equilibrium obtained with well-balanced ALE schemes on moving nonconforming meshes. The mesh is shown at time $t = 200$. On the left we report the grid in polar coordinates where the shear discontinuities lie over straight lines. On the right the corresponding grid is shown in Cartesian coordinates.

Table 12: Stationary vortex in equilibrium. Maximum error on the water level h between the exact and the numerical solution obtained with the first order well balanced nonconforming ALE method. In the left column we show the error for Test A with finer and finer meshes with a fixed final time, in the central column we choose a coarse mesh and show the error for longer and longer times. In the right column, the results for Test B are shown.

Test A		Test B			
$t_{end} = 10$		points 16×8		points 16×8	
points	error	time	error	time	error
12×6	1.42E-14	10	1.28E-14	10	2.11E-13
16×8	1.28E-14	50	3.74E-14	100	4.84E-13
24×12	3.04E-14	150	4.02E-14	150	3.25E-13
36×18	6.68E-14	200	4.88E-14	200	2.62E-13

7 High order of accuracy on Voronoi tessellations with topology changes

In this Section we come back to the complete *arbitrary high order* FV-DG framework introduced in Section 2. We first briefly explain in Section 7.1 the CWENO reconstruction procedure that we use to obtain the high order in space, and in Section 7.2 we clarify how we move the mesh. Then, we concentrate on the Voronoi tessellation that we *regenerate* at any time step and to which we confer the adjoin freedom of a mesh topology that can continuously change, i.e the shape of each element and its neighbors can be different at any time. Thus, we explain how to connect meshes with different typologies in space–time and the kind of degenerate space–time elements that this connection originates in Section 7.3; we then extend the predictor-corrector ADER paradigm even to these degenerate elements in Sections 7.4 and 7.5. Finally, in Section 7.6 we show some numerical results that prove the robustness of our new algorithm in the case of vortical problem and shock waves and its order of accuracy.

This completely new approach in the framework of direct ALE FV-DG schemes was first presented [77].

7.1 CWENO reconstruction

For finite volume schemes ($N = 0$) the reconstruction procedure allows us to compute a high order non-oscillatory polynomial representation $\mathbf{w}_h^n(\mathbf{x}, t^n)$ of the solution $\mathbf{Q}(\mathbf{x}, t^n)$ for each Voronoi polygon P_i^n , starting from the values of $\mathbf{u}_h^n(\mathbf{x}, t^n)$ in P_i^n and its neighbors.

In order to fulfill the requirement of non-oscillatory behavior and the necessity of using a large stencil centered in P_i^n for achieving high accuracy, in contrast with each other due to the well-known Godunov theorem [83], we adopt a nonlinear reconstruction operator. Any ENO [88], WENO [64,65], HWENO [134], strategy could in principle be used. Here we rely on the CWENO reconstruction strategy first introduced in [105,106,107], and which can be cast in the general framework described in [44]. We closely follow the work outlined in [60] for unstructured triangular and tetrahedral meshes.

The reconstruction starts from the computation of a so-called *central polynomial* \mathbf{P}_{opt} of degree M . In order to define \mathbf{P}_{opt} in a robust manner, following [60,4,98,142], we consider a stencil \mathcal{S}_i^0 which is filled with a total number of $n_e = f \cdot \mathcal{M} = f \cdot \mathcal{L}(M, d)$ elements, containing cell P_i^n and its neighbors

$$\mathcal{S}_i^0 = \bigcup_{k=1}^{n_e} P_{i_k}^n, \quad (224)$$

with the safety factor $f \geq 1.5$. Stencil \mathcal{S}_i^0 includes the current Voronoi polygon P_i^n , the first layer of Voronoi neigh-

bors (node neighbors of P_i^n) denoted by $\mathcal{V}(P_i^n)$, and is filled by recursively adding neighbors of elements that have been already selected, until the desired number n_e is reached. The polynomial $\mathbf{P}_{\text{opt}}(\mathbf{x}, t^n)$ is then defined by imposing that its average on each cell $P_{i_k}^n$ matches the known cell average $\hat{\mathbf{u}}_{0,i_k}^n$. Since $n_e > \mathcal{M}$, this of course leads to an overdetermined linear system, which is solved using a constrained least-squares technique (CLSQ) [62].

To stabilize the reconstruction operator, the CWENO algorithm makes use of *other polynomials* of lower degree. Given a Voronoi polygon P_i^n with $N_{V_i}^n$ Voronoi neighbors $\mathcal{V}(P_i^n)$, we construct $N_{V_i}^n$ interpolating polynomials of degree $M^s = 1$ referred to as *sectorial polynomials*. More precisely, we consider $N_{V_i}^n$ stencils S_i^s with $s \in [1, N_{V_i}^n]$, each of them containing exactly $\hat{n}_e = \mathcal{L}(M^s, d) = (d+1)$ cells. Each S_i^s includes always the central cell P_i^n and two consecutive neighbors belonging to $\mathcal{V}(P_i^n)$.

For each stencil S_i^s we compute a linear polynomial $\mathbf{P}_s(\mathbf{x}, t^n)$ by solving the reconstruction systems

$$\mathbf{P}_s(\mathbf{x}, t^n) \in \mathbb{P}_1 \text{ s.t. } \forall P_{i_k}^n \in S_i^s : \frac{1}{|P_{i_k}^n|} \int_{P_{i_k}^n} \mathbf{P}_s(\mathbf{x}, t^n) d\mathbf{x} = \hat{\mathbf{u}}_{0,i_k}^n, \quad (225)$$

which are not overdetermined and therefore have a unique solution for non-degenerate locations of the Voronoi barycenters. Following the general framework introduced in [44], we select a set of positive coefficients $\lambda_0, \dots, \lambda_{N_p}$ such that

$$\sum_{s=0}^{N_{V_i}^n} \lambda_s = 1 \quad (226)$$

and we define a new polynomial

$$\mathbf{P}_0(\mathbf{x}, t^n) = \frac{1}{\lambda_0} \left(\mathbf{P}_{\text{opt}}(\mathbf{x}, t^n) - \sum_{s=1}^{N_p} \lambda_s \mathbf{P}_s(\mathbf{x}, t^n) \right) \in \mathbb{P}_M, \quad (227)$$

so that the linear combination of the polynomials $\mathbf{P}_0, \dots, \mathbf{P}_{N_{V_i}^n}$ with the coefficients $\lambda_0, \dots, \lambda_{N_p}$ is equal to \mathbf{P}_{opt} and conservation is ensured. We consider the linear weights used in [63], namely $\lambda_0 = 10^5$ for \mathcal{S}_i^0 and $\lambda_s = 1$ for the sectorial stencils. These weights are then normalized in order to sum to unity, according to the requirement (226).

Finally, the sectorial polynomials \mathbf{P}_s with $s \in [1, N_{V_i}^n]$ are nonlinearly hybridized with \mathbf{P}_0 . We thus obtain $\mathbf{w}_h(\mathbf{x}, t^n)$ in P_i^n as

$$\mathbf{w}_h(\mathbf{x}, t^n) = \sum_{s=0}^{N_p} \omega_s \mathbf{P}_s(\mathbf{x}, t^n), \quad \mathbf{x} \in P_i^n, \quad (228)$$

where the normalized *nonlinear weights* ω_s are given by

$$\omega_s = \frac{\tilde{\omega}_s}{\sum_{m=0}^{N_{V_i}^n} \tilde{\omega}_m}, \quad \text{with} \quad \tilde{\omega}_s = \frac{\lambda_s}{(\sigma_s + \varepsilon)^r}. \quad (229)$$

In the above expression the weights are chosen according to [62].

7.2 Evolution of the Voronoi tessellation

At this point we have a high order spatial representation of the solution $\mathbf{Q}(\mathbf{x}, t^n)$ at the current time t^n given by the polynomial $\mathbf{w}_h^n = \mathbf{w}_h(\mathbf{x}, t^n)$ of degree M given or by the previous reconstruction procedure if $N = M > 0$ (FV case) or by $\mathbf{w}_h^n = \mathbf{u}_h^n$ if $N = M > 0$.

By evaluating \mathbf{w}_h^n at the generator points \mathbf{x}_c^n , i.e. $\mathbf{w}_h^n(\mathbf{x}_c^n, t^n)$ with (8), we recover the local fluid velocity $\mathbf{v}(\mathbf{x}_c^n)$, that can be used to compute the new coordinates of the generator points simply as

$$\mathbf{x}_c^{n+1} = \mathbf{x}_c^n + \Delta t \mathbf{v}(\mathbf{x}_c^n). \quad (230)$$

The Delaunay triangulation connecting the new coordinates of the generator points \mathbf{x}_c^{n+1} is now recomputed, as well as the corresponding updated Voronoi tessellation. Note that the *only connection* between the tessellations at time t^n and t^{n+1} is the number N_p of generator points (i.e. of Voronoi polygons) and their *global numbering*. Instead, the shape of each polygon is allowed to change, i.e. $N_{V_i}^n \neq N_{V_i}^{n+1}$, and consequently also the connectivities, i.e. for example $\mathcal{V}(P_i^n) \neq \mathcal{V}(P_i^{n+1})$.

This change of the grid topology is actually the *strength* of the present algorithm, since it allows us to maintain a high mesh quality without distorted elements, as depicted in Figures 43 and 44, where we show a comparison between the results obtained by allowing topology changes and by imposing a fixed connectivity, respectively.

Note that in our ALE formalism, the mesh can be moved with *any* velocity, hence it is not necessary to integrate the above relation (230) with high order of accuracy. Nevertheless, (230) can be solved at the aid of a high order Taylor method [20, 25, 146], leading to high order approximation of the Lagrangian trajectories of the generators points.

In particular, the Taylor expansion of a generator \mathbf{c}_i at time t^{n+1} with respect to its position at time t^n can be written as

$$\mathbf{x}_c^{n+1} = \mathbf{x}_c^n + \Delta t \frac{d\mathbf{x}}{dt} + \frac{\Delta t^2}{2} \frac{d^2\mathbf{x}}{dt^2} + \frac{\Delta t^3}{6} \frac{d^3\mathbf{x}}{dt^3} + \frac{\Delta t^4}{24} \frac{d^4\mathbf{x}}{dt^4} + \mathcal{O}(5), \quad (231)$$

which achieve *fourth* order of accuracy in time.

Now, the high order time derivatives in (231) are replaced by high order spatial derivatives, via the *Cauchy-Kovalevskaya* procedure, using the trajectory equation

$$\frac{d\mathbf{x}}{dt} = \mathbf{v}(\mathbf{x}(t)), \quad (232)$$

and assuming a stationary velocity field (i.e. $\frac{\partial \mathbf{v}}{\partial t} = \mathbf{0}$).

Now, the partial derivatives of \mathbf{v} are recovered from the local fluid velocities \mathbf{u} through the high order polynomials \mathbf{w}_h (8) which represent with high order of accuracy the conserved variables inside each cells.

7.3 Space–time control volumes - topology changes

As said in Section 2.4, if there are no topology changes between t^n and t^{n+1} , i.e. $N_{V_i}^n = N_{V_i}^{n+1}$ and $\mathcal{V}(P_i^n) = \mathcal{V}(P_i^{n+1})$, the space–time control volume C_i^n is easily obtained by connecting each node of the polygon P_i^n via *straight* line segments with the *corresponding* node of P_i^{n+1} (see Figure 40), and each sub–triangle $T_{ij}^n \in \mathcal{T}(P_i^n)$ is connected with the corresponding $T_{ij}^{n+1} \in \mathcal{T}(P_i^{n+1})$ obtaining a *sub–space–time* control volume, denoted by sC_{ij}^n .

We recall that the node numbering (i.e. the numbering of the blue points in Figure 40) could be in principle different at the two time levels so the *correspondence* between the nodes at time level t^n and t^{n+1} is not obvious. Nevertheless, it can be recovered from the numbering of the Voronoi neighbors $\mathcal{V}(P_i^{n/n+1})$ that on the contrary remains the same.

Let us now consider P_i^n and P_i^{n+1} in the case $N_{V_i}^n \neq N_{V_i}^{n+1}$, as in first panel of Figure 41). The space–time connection between them induces the appearance of degenerate elements of two types: (i) degenerate *sub–space–time* control volumes sC_{ij}^n , where either their top or bottom faces are degenerate triangles that are collapsed just to a line, see Figures 41b–41c; (ii) and also sliver space–time elements, see Figure 41d. Technical details on their construction can be found in [77].

The degenerate *sub–space–time* control volumes of type (i) do not pose particular problems because they are part of a standard control volume, so everything is naturally well defined on them (basis functions, quadrature points, values of the numerical solution \mathbf{u}_h^n , of the reconstruction polynomials \mathbf{w}_h^n , and of the space–time predictor \mathbf{q}_h^n defined below in (11)).

On the contrary, the space–time sliver element in Figure 41d is a completely new control volume which does neither exist at time t^n , nor at time t^{n+1} , since it coincides with an edge of the tessellation at the old and at the new time levels, and, as such, has zero area in space at t^n and t^{n+1} . We denote this kind of element with S_i^n , its total lateral surface with ∂S_i^n and each of the four lateral faces with $\partial S_{ij}^n, j = 1, \dots, 4$. Space–time sliver elements always have four neighbors, namely the two Voronoi polygons that share their degenerate bottom face (edge) and the two Voronoi polygons that share their degenerate top face (edge).

Two *problems* can arise while assembling the space–time connectivity: $\mathcal{V}(C_i^n)$ could be not sortable respecting both the order of $\mathcal{V}(P_i^n)$ and $\mathcal{V}(P_i^{n+1})$, or more than three sliver elements could be necessary to complete the connection path. In this case we adopt a MOOD type [23, 22] proce-

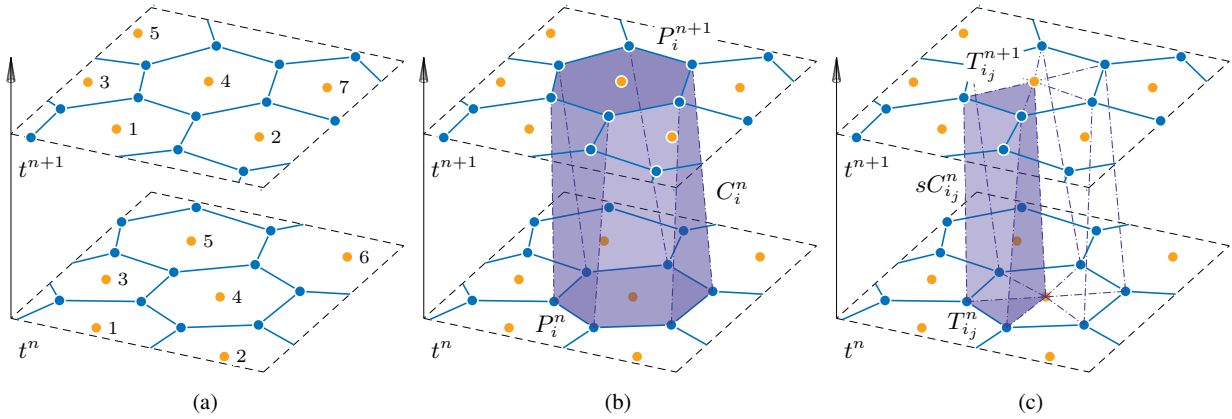


Fig. 40: Space time connectivity *without* topology changes. (a) The tessellation at time t^n and time t^{n+1} . (b) P_i^n is connected with P_i^{n+1} to construct the space–time control volume C_i^n . (c) The sub-triangle T_{ij}^n is connected with T_{ij}^{n+1} to construct the sub–space–time control volume sC_{ij}^n .

ture, i.e. we restart the current time step with a smaller time step size Δt (reduced by a factor of 2 for example). Eventually, more restarts are needed, until the connection between the two meshes is coherent. Since the mesh generation and the connectivity construction are not expensive, the performances of the algorithm are not negatively influenced by this additional MOOD-type procedure (which applies *before* the evolution in time). We emphasize that such problems are encountered very rarely, see Tables 16 for some statistics.

However, notice that the presence of degenerate elements is *strictly unavoidable* in order to connect meshes in space *and* time that include topology changes. They are also needed to collect enough geometrical information for ensuring high order of accuracy in a direct ALE framework. For comparison purposes, let us consider the work presented in [135], where the authors, in order to connect meshes with topology changes (within a different framework w.r.t. this work), have introduced some *pyramidal* degenerate elements instead of our sliver elements. The strategy proposed in the aforementioned reference is indeed interesting and could in principle be applied also to the framework of our *explicit* high order direct ALE schemes. However, besides the same complexities described for our sliver elements, an additional difficulty would arise, since a degeneracy would occur at the midpoint of the time step.

Finally, we remark here the difficulties related to this kind of elements: i) the fact that they coincide with an edge at time t^n makes it difficult to fix a valid initial condition in the predictor step necessary for the high order of accuracy in time, and ii) the fact that they coincide with an edge at time t^{n+1} could prevent conservation in an explicit scheme. Nevertheless, with the strategy that will be outlined in Sections 7.4 and 7.5, no space–time contributions are lost while advancing the numerical solution in time, i.e. our proposed

ADER ALE FV-DG schemes are fully conservative and keep their formal high order of accuracy even in the presence of space–time sliver elements.

7.4 Space–time predictor on sliver elements

The predictor procedure on space–time sliver elements, as those shown in Figures 41d needs particular care, because their bottom face is degenerate and consists only in a line segment, hence the spatial integral over P_i^n vanishes, i.e. there is no possibility to introduce the initial condition of the local Cauchy problem at time t^n into the predictor for space–time sliver elements.

Furthermore, the degenerate bottom faces are the edges of the Voronoi tessellation at t^n and are thus at the interface between two adjacent elements, which have in principle a discontinuous solution \mathbf{w}_h^n . Therefore, an initial value for a sliver element is in general not easy to define. Thus, in order to couple (13) with some known data from the past we have to slightly modify the algorithm detailed previously.

In particular, the upwinding in time approach is not only used for the surface P_i^n , as done in (15), but we actually use the jump terms on the *entire* part of the space–time surface ∂C_i^n that closes a sliver control volume. As already stated in the previous section, the information needed to feed the predictor is allowed to come only from the past, i.e. only from those space–time neighbors C_j^n whose common surface $\partial C_{ij}^n = C_i^n \cap C_j^n$ exhibits a *negative* time component of the outward pointing space–time normal vector ($\bar{\mathbf{n}}_t^- < 0$). In this way, we can introduce information from the past into the space–time sliver elements by considering also its neighbor elements, but respecting at the same time the causality principle in time, hence using again upwinding for the flux

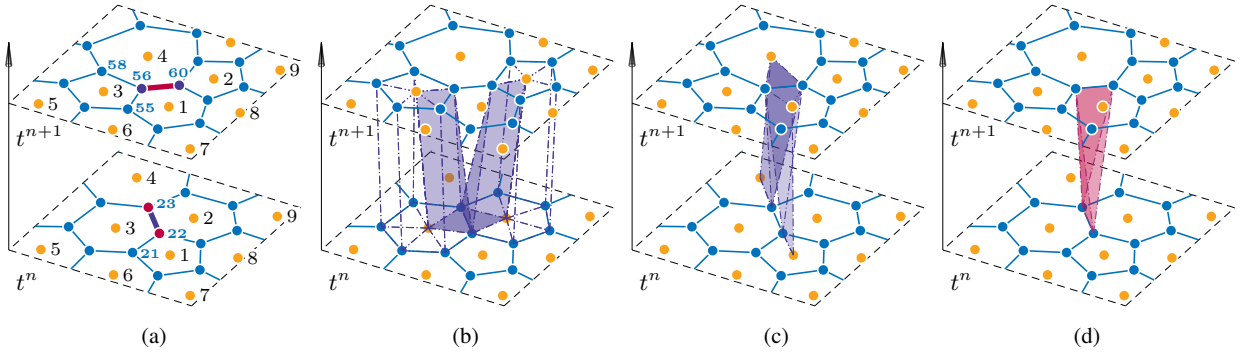


Fig. 41: Space time connectivity *with* topology changes, degenerate sub–space–time control volumes and sliver element. Panel (a): at time t^n the polygons P_2^n and P_3^n are neighbors and share the highlighted edge, instead at time t^{n+1} they do not touch each other; the opposite situation occurs for polygons P_1^n and P_4^n . This change of topology causes the appearance of degenerate elements of different types. The first type is given by degenerate sub–space–time control volumes colored in violet in Panels (b) and (c). The second type of degenerate elements are called *space–time sliver elements*, an example is colored in magenta in Panel (d). The sub–space–time control volumes of Panels (b) and (c) are triangular prisms with one of their faces collapsed to just a line: they do not pose particular problems because they are part of a standard control volume, so everything is naturally well defined on them (basis functions, quadrature points, values of \mathbf{u}_h^n , \mathbf{w}_h^n , \mathbf{q}_h^n). On the contrary, the sliver element in panel (d) is a completely new control volume which does neither exist at time t^n , nor at time t^{n+1} , since it coincides with an edge of the tessellation and, as such, has zero areas in space. However, it has a *non-negligible volume* in space–time. The difficulties associated to this kind of element are due to the fact that \mathbf{w}_h is not clearly defined for it at time t^n and that contributions across it should not be lost at time t^{n+1} in order to guarantee conservation.

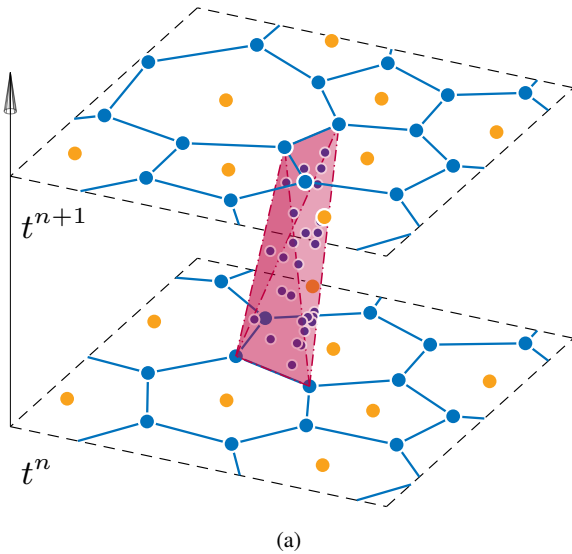


Fig. 42: Quadrature points for the computation of volume integrals and the space–time predictor on sliver elements (with $M = 2$).

evaluation of the jump term in (14). As a consequence, the predictor solution \mathbf{q}_h^n is again obtained by means of (13), but treating the *entire* space–time surface ∂C_i^n with the upwind

in time approach, hence leading to

$$\begin{aligned} (\mathbf{K}_1^* - \mathbf{F}_1^*) \hat{\mathbf{q}}_i^n &= - \sum_j \mathbf{F}_j^* \hat{\mathbf{q}}_j^n - \mathbf{K}_x^* \mathbf{f}(\hat{\mathbf{q}}_i^n) - \mathbf{K}_y^* \mathbf{g}(\hat{\mathbf{q}}_i^n) + \mathbf{M}^* \mathbf{S}(\hat{\mathbf{q}}_i^n), \end{aligned} \quad (233)$$

where the following definitions for the sliver element hold

$$\begin{aligned} \mathbf{K}_1^* &= \int_{C_i^n \setminus \partial C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial t} d\mathbf{x} dt, & \mathbf{K}_x^* &= \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial x} d\mathbf{x} dt, \\ \mathbf{K}_y^* &= \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial y} d\mathbf{x} dt, & \mathbf{M}^* &= \int_{S_i^n} \theta_k \theta_\ell d\mathbf{x} dt, \\ \mathbf{F}_1^* &= \int_{\partial C_i^n} \theta_k \theta_\ell \tilde{\mathbf{n}}_i^- dS, & \mathbf{F}_j^* &= \int_{\partial C_{ij}^n} \theta_k \theta_\ell \tilde{\mathbf{n}}_i^- dS. \end{aligned} \quad (234)$$

This is slightly different from what is done for standard elements in (18), where only the space–time surface at time t^n , i.e. P_i^n , is considered for introducing the initial condition \mathbf{w}_h^n . Here, the information from the past comes through the upwind fluxes contained in the term $\mathbf{F}_j^* \hat{\mathbf{q}}_j^n$ in (233) and thus requires the knowledge of the predictor solution $\hat{\mathbf{q}}_j^n$ in the neighbor C_j^n . This is the reason why the predictor step must *first* be performed over all the standard elements using (18), so that the predictor solution \mathbf{q}_h^n is always available to feed the temporal fluxes with the quantities $\hat{\mathbf{q}}_j^n$ that are needed for solving (233) in the case of the space–time sliver

elements. We underline again that a space–time sliver element has always four standard Voronoi elements as neighbors. This closes the description of the predictor step for the space–time sliver elements.

7.5 Corrector step on sliver elements

Let us now consider the numerical scheme given by (30) in the case of a sliver element $C_i^n = S_i^n$:

$$\begin{aligned} 0_\ell \hat{\mathbf{u}}_\ell^{n+1} = & 0_\ell \hat{\mathbf{w}}_\ell^n - \sum_{j=1}^4 \int_{\partial S_{i_j}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} dS \\ & + \int_{S_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) d\mathbf{x}dt + \int_{S_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt, \end{aligned} \quad (235)$$

Since for sliver elements $|P_i^n| = |P_i^{n+1}| = 0$, the first two terms vanish. However, since the method is explicit and \mathbf{q}_h^n only depends on information coming from the past, the remaining terms in (235) are in general not equal to zero, i.e.

$$\begin{aligned} - \sum_{j=1}^4 \int_{\partial S_{i_j}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} dS \\ + \int_{S_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) d\mathbf{x}dt + \int_{S_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt \neq 0. \end{aligned} \quad (236)$$

We underline that computing these quantities does not pose any problem, since \mathbf{q}_h^n on S_i^n is well defined, and the shape of a space–time sliver element is that of a tetrahedron in space–time, hence allowing standard quadrature rules to be used for integral evaluations (see also Figure 42).

The problem here arises from the fact that, using (235), the non-null quantity (236) will be lost at time t^{n+1} because it plays a role only in the evolution of S_i^n , which exists between t^n and t^{n+1} , but is null at t^{n+1} . In order to be conservative, we must avoid losing any contribution from the sliver elements. We therefore couple the weak formulation on S_i^n with the weak form of *one* of its standard space–time neighbors. Here, we always choose the one with the biggest space–time volume, referred to as C_{big} . The choice of the biggest volume is not mandatory, it only represents our way to uniquely fix the choice of a particular neighbor of the sliver element. The test function $\tilde{\varphi}_k$ of (235) is then referred to the barycenter of C_{big} . Conservation is guaranteed by adding the contribution (236) of the sliver element S_i^n to

the neighbor C_{big} , hence

$$\begin{aligned} \left(\int_{P_{\text{big}}^{n+1}} \tilde{\varphi}_k \varphi_\ell d\mathbf{x} \right) \hat{\mathbf{u}}_\ell^{n+1} = & \left(\int_{P_{\text{big}}^n} \tilde{\varphi}_k \psi_\ell d\mathbf{x} \right) \hat{\mathbf{w}}_\ell^n \\ & - \sum_{j=1}^{N_{C_{\text{big}}}^{n,sl}} \int_{\partial C_{\text{big}_j}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} dS \\ & + \int_{C_{\text{big}}^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) d\mathbf{x}dt + \int_{C_{\text{big}}^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt \\ & + \sum_{j=1}^4 \int_{\partial S_{i_j}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} dS \\ & + \int_{S_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) d\mathbf{x}dt + \int_{S_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt. \end{aligned} \quad (237)$$

7.6 Numerical results

The numerical results presented in this section comes from applications to the Euler equations of gas dynamics and to the magnetohydrodynamics (MHD) system.

With these results we want to demonstrate the capability of our scheme in *maintaining a high quality mesh* for very long computational times, even in the case of strong shear flows and vortices, and the achieved *order of convergence* for both Finite Volume and Discontinuous Galerkin schemes, see Tables 13, 14, 18 and 19.

Moreover, for *all* the presented test cases we have numerically verified that mass and volume conservation is respected up to *machine precision* at *any* time step, and that the same holds true for the GCL condition on *each* element, even when topology changes occur. In addition, to be more convincing on the fact that our schemes always satisfy the GCL condition by construction, we refer to the set of test cases of Section 7.6.2, where constant states are preserved up to machine precision for very long computational times over moving meshes where topology changes regularly occur.

Finally, a more complicated test problem (Section 7.6.3) aims at showing the *robustness* of our method, concerning both the mesh quality in presence of arbitrary and strong velocity fields as well as the consistency/stability of our high order schemes. In particular, we test the *a posteriori* sub–cell finite volume *limiter* used to stabilize the DG scheme that indeed avoids undesirable oscillations by activating only where needed (see Figure 47).

7.6.1 Isentropic vortex

To verify the order of convergence of the proposed ALE FV-DG scheme we consider a smooth isentropic vortex flow according to [91]. The initial computational domain is the square $\Omega = [0; 10] \times [0; 10]$ with wall boundary conditions

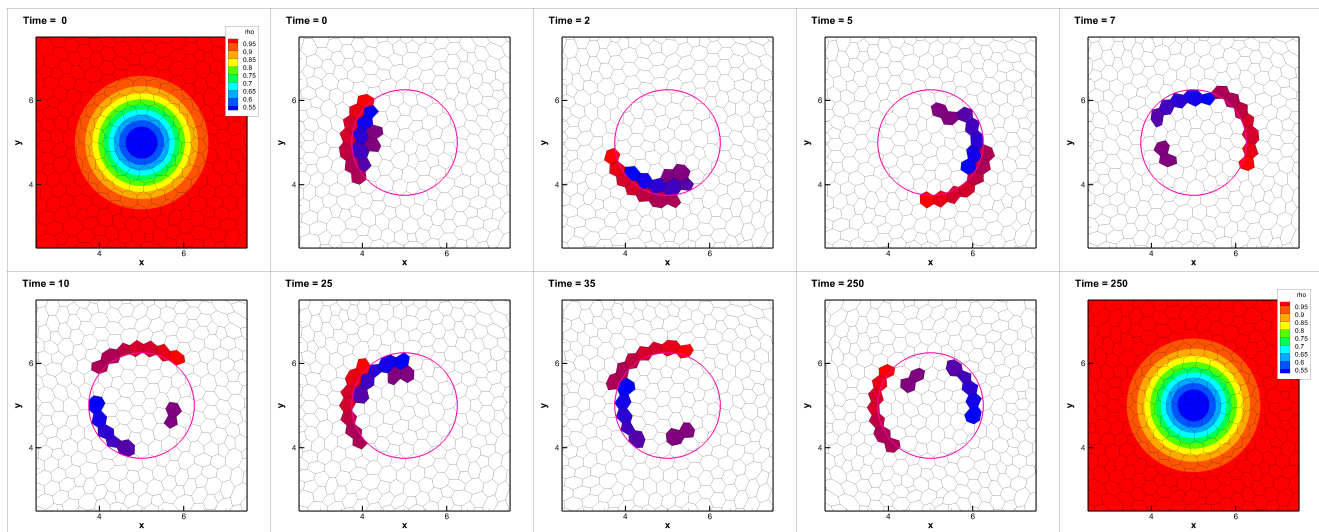


Fig. 43: Stationary rotating vortex solved with our fourth order P_3P_3 ALE-DG scheme on a moving Voronoi mesh of 2116 elements with dynamical change of connectivity. Density contours (first and last image) and the position of a bunch of highlighted elements are provided at different times. The mesh is regenerated at every time step and connected in space time to reach high order of accuracy on a moving domain: this makes it possible to *substantially improve the mesh quality* w.r.t. standard conforming ALE schemes without topology change, for which mesh tangling would occur leading to a stop of the simulation.

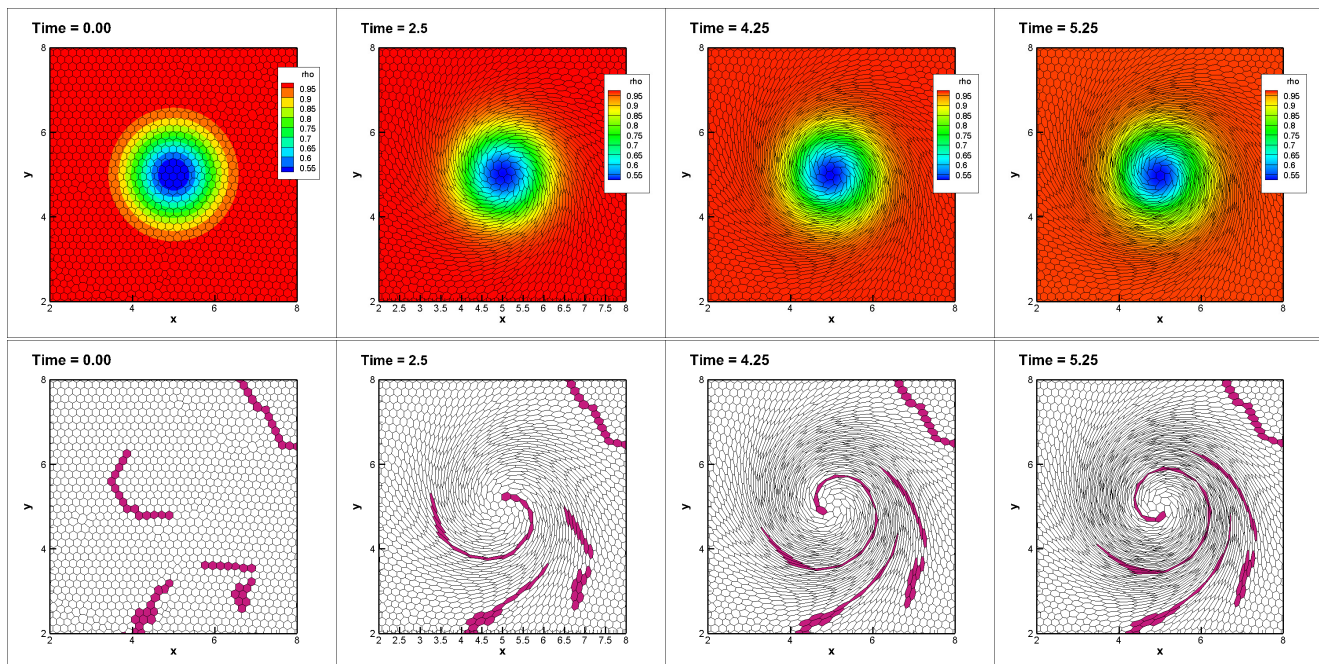


Fig. 44: Stationary rotating vortex solved with a fourth order P_3P_3 ALE-DG scheme on a moving Voronoi mesh of 2116 elements with *fixed* connectivity. Density contours (top) and position of a bunch of highlighted elements (bottom) are provided at different times. The mesh quality is deteriorating already at time $t \simeq 4$ and the simulation ultimately stops at $t \simeq 5.25$ due to tangling elements.

Table 13: Isentropic vortex. Numerical convergence results for the finite volume algorithm on moving meshes with topology changes. The error norms refer to the variable ρ at time $t = 0.5$ in L_1 norm.

$P_0P_1 \rightarrow \mathcal{O}2$			$P_0P_2 \rightarrow \mathcal{O}3$			$P_0P_3 \rightarrow \mathcal{O}4$			$P_0P_4 \rightarrow \mathcal{O}5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$
3.8E-01	3.1E-02	-	3.8E-01	2.9E-02	-	1.9E-01	1.6E-03	-	4.7E-01	4.0E-02	-
2.0E-01	6.2E-03	2.4	1.9E-01	4.6E-03	2.8	1.3E-01	4.1E-04	3.4	3.8E-01	1.4E-02	4.8
1.3E-01	2.4E-03	2.4	1.3E-01	1.4E-03	2.9	9.9E-02	1.4E-04	3.8	1.3E-01	2.5E-04	3.8
9.9E-02	1.3E-03	2.3	9.9E-02	6.1E-04	3.0	7.9E-02	6.0E-05	3.9	9.9E-02	6.7E-05	4.6
8.0E-02	7.8E-04	2.2	7.9E-02	3.1E-04	2.0	6.7E-03	3.0E-05	3.8	7.9E-02	2.4E-05	4.7

Table 14: Isentropic vortex. Numerical convergence results for the discontinuous Galerkin algorithm on moving meshes with topology changes. The error norms refer to the variable ρ at time $t = 0.5$ in L_1 norm.

$P_1P_1 \rightarrow \mathcal{O}2$			$P_2P_2 \rightarrow \mathcal{O}3$			$P_3P_3 \rightarrow \mathcal{O}4$			$P_4P_4 \rightarrow \mathcal{O}5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$
7.5E-01	6.3E-03	-	7.5E-01	1.4E-02	-	6.1E-01	1.4E-03	-	1.4E-00	1.1E-02	-
6.1E-01	4.2E-04	1.9	6.1E-01	7.2E-03	3.4	5.2E-01	7.4E-04	3.7	1.0E-00	2.0E-03	5.9
3.2E-01	9.9E-04	2.2	3.2E-01	9.3E-04	3.2	4.7E-01	4.1E-04	5.9	9.8E-01	1.6E-03	4.7
2.2E-01	4.4E-04	2.0	2.2E-01	2.8E-04	3.0	3.2E-01	7.7E-05	4.4	8.9E-01	9.0E-04	5.9
1.6E-01	2.5E-05	2.0	1.6E-01	1.2E-04	3.0	2.2E-01	1.6E-05	4.0	8.5E-01	7.0E-04	5.1

Table 15: Isentropic vortex. Numerical convergence results for the third order P_2P_2 discontinuous Galerkin algorithm on moving meshes with topology changes. The error norms refer to the variable ρ at time $t = 0.5$ in L_2 norm. The three groups of results refer to three different ways of ordering the space-time neighbors of each element. The fact that the errors are exactly the same up to machine precision proves that the algorithm is independent of the neighbor ordering used in the construction of the space-time elements.

ordering from 1 st common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$\mathcal{O}(L_2)$
0.319411631217116	9.2414523328907E-04	-
0.242212163540348	3.9353901580992E-04	3.1
0.194949032600822	2.0616099552666E-04	3.0
0.163155447483668	1.1964571728528E-04	3.1
0.122985013713313	5.1270456290057E-05	3.0
ordering from 2 nd common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$\mathcal{O}(L_2)$
0.319411631217114	9.2414523328982E-04	-
0.242212163540348	3.9353901581037E-04	3.1
0.194949032600822	2.0616099552752E-04	3.0
0.163155447483668	1.1964571728459E-04	3.1
0.122985013713313	5.1270456288495E-05	3.0
ordering from 3 rd common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$\mathcal{O}(L_2)$
0.319411631217116	9.2414523328907E-04	-
0.242212163540348	3.9353901580992E-04	3.1
0.194949032600822	2.0616099552666E-04	3.0
0.163155447483668	1.1964571728400E-04	3.1
0.122985013713313	5.1270456291299E-05	3.0

set everywhere. The initial condition is given by some perturbations δ that are superimposed onto a homogeneous background field $\mathbf{Q}_0 = (\rho, u, v, p) = (1, 0, 0, 1)$, assuming that the entropy perturbation is zero, i.e. $\delta S = 0$. The perturbations for density and pressure are

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1, \quad (238)$$

with the temperature fluctuation $\delta T = -\frac{(\gamma-1)\varepsilon^2}{8\gamma\pi^2}e^{1-r^2}$ and the vortex strength is $\varepsilon = 5$. The velocity field is affected by the following perturbations

$$\begin{pmatrix} \delta u \\ \delta v \end{pmatrix} = \frac{\varepsilon}{2\pi} e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ (x-5) \end{pmatrix}. \quad (239)$$

This is a stationary equilibrium of the system so the exact solution coincides with the initial condition at any time.

Convergence. Tables 13 and 14 report the convergence rates from second up to fifth order of accuracy for the vortex test problem run on a sequence of successively refined meshes. For each element, its characteristic size h_i^n at time t^n is given by the diameter of the circumscribed circle and we denote with $h(\Omega(t_f))$ the average of h_i^n at the final time of the simulation $t_f = 0.5$. Thus, $h(\Omega(t_f))$ represents the characteristic mesh size of our mesh. The optimal order of accuracy is achieved both in space and time for the FV schemes as well as for the DG schemes. We would like to underline that this is not trivial for moving Voronoi meshes, because the changing characteristic mesh sizes could affect the convergence results (the mesh is *not* stationary at all).

Quality. In Figure 43 we plot the density contours and the two-dimensional mesh configuration at various output times obtained with our fourth order ALE-DG scheme. We would like to attract the attention on the endurance of the simulation and on the high quality of the density profile obtained even after very long simulation times. The correct density profile and a high quality mesh are conserved for at least sixty times longer with respect to standard conforming ALE schemes, where mesh tangling would occur and stop the simulation much earlier (see Figure 44). The obtained results are also superior with respect to existing ReALE codes, which are usually of very low order of accuracy in space and time and are therefore affected by a much higher numerical dissipation.

The position of a bunch of elements is also highlighted at different times: this makes clear how strong the rotation is to which the mesh elements are subject. It also highlights the importance of allowing topology changes in the computational grid, which needs to provide enough topological flexibility in order to preserve a high quality mesh over long computational times. Indeed, if the preservation of the connectivity had been imposed, the elements would have been quite distorted after only rather short times (see Figure 44).

Independence of the neighbor numbering. To prove that our algorithm is also completely independent of the space–time neighbor numbering chosen when connecting the old mesh to the new one, we have carried out the following test. In the framework of a third order P_2P_2 DG scheme we have simulated the isentropic vortex up to a final time of $t = 0.5$ on a series of meshes, namely composed by 961, 1681, 2601, 3721 and 6561 Voronoi elements moving with the exact velocity computed at the generator point of each element. Then, we have run the algorithm for each mesh configuration by ordering the space–time neighbors in three different ways, namely starting first with the first common neighbor, next with the second common neighbor and last with the third common neighbor (if existing, otherwise we have used the first one again).

Table 15 shows that not only the order of the algorithm does not depend on the neighbor numbering, but also that the final errors are the same up to machine precision.

7.6.2 Numerical study of GCL property satisfaction

In order to verify that our schemes satisfy the GCL property up to machine precision we consider the following standard test case for the ReALE community, see for example [115].

The initial condition of this test are given by *constant states*, namely $\mathbf{Q}_0 = (\rho, u, v, p) = (1, 0, 0, 1)$ that should stay constant even with moving meshes if the GCL condition is satisfied. The initial computational domain is the square $\Omega = [0; 10] \times [0; 10]$ covered with a mesh of 1979 Voronoi elements and wall boundary conditions are set everywhere.

Then, in agreement with the ALE framework, the mesh is moved with a completely arbitrary velocity field (thus not with the fluid velocity which would be zero in this test case). In particular, we have chosen a vortical velocity field varying in a sinusoidal fashion, given in the form

$$\begin{aligned} \mathbf{v}(x, y) = & \left(-\sin\left(\frac{2\pi}{\ell}(y-y_0)\right) \cos\left(\frac{\pi}{\ell}(x-x_0)\right) \exp(-kr), \right. \\ & \left. \cos\left(\frac{\pi}{\ell}(y-y_0)\right) \sin\left(\frac{2\pi}{\ell}(x-x_0)\right) \exp(-kr) \right), \end{aligned} \quad (240)$$

where $\mathbf{x}_0 = (x_0, y_0) = (5, 5)$, $\ell = 10$, $k = 0.1$ and $r = \sqrt{(x-x_0)^2 + (y-y_0)^2}$.

In Figure 45 we show the error between our numerical results for density and velocity, with respect to the exact constant solution, obtained with our FV and DG schemes of order three and four, namely the P_0P_2 , P_0P_3 , P_2P_2 and P_3P_3 schemes. We emphasize that these results are obtained over long computational times (60 seconds, and three complete turns of a reference point), thus after thousands of time steps. Moreover topology changes regularly occurs during the mesh motion: the number of sliver elements generated

Table 16: In this Table we report the number of total time steps needed to reach the final computational time $t_f = 60$ with four different methods. We report also the number of sliver elements that have been originated along the simulations due to the occurred topology changes and the number of time steps that have been repeated through the MOOD loop.

Method	time steps	slivers	restarts
FV $\mathcal{O}(3)$	5524	21930	5
FV $\mathcal{O}(4)$	4896	19819	4
DG $\mathcal{O}(3)$	33019	19392	0
DG $\mathcal{O}(4)$	55496	18995	2

per simulation over the total number of time steps is reported in Table 16, and a bunch of initially neighbors elements is plotted in Figure 46 at different times, in order to make clear that elements really change their topology and connectivity during the simulation.

This test case proves numerically that GCL property is satisfied by our scheme even when topology changes occur and sliver elements appear, which is indeed a property that our schemes satisfy *by construction*, since PDE integration is always performed over close control volumes [15].

Finally, in Table 17 we report the percentage of computational times employed by i) all the the procedures necessary to rebuild a new mesh at each time step, namely the mesh regeneration, and the construction of the space-time connectivity; ii) the predictor-corrector step performed on standard elements; iii) the predictor-corrector step performed on sliver elements. We remark that step i) actually consists in a complete rebuild of a new configuration without exploiting the previous one, which would decrease the computational cost of the geometric part of the code. Nevertheless, we emphasize that the computational cost of the geometric part of our scheme is minimal and does not affect the final cost of the entire algorithm.

7.6.3 Sedov problem

This test problem is widespread in the literature [113] and it describes the evolution of a blast wave that is generated at the origin $\mathbf{O} = (x, y) = (0, 0)$ of the computational domain $\Omega(0) = [0; 1.2] \times [0; 1.2]$. An exact solution based on self-similarity arguments is available from [141] and the fluid is assumed to be an ideal gas with $\gamma = 1.4$, which is initially at rest and assigned with a uniform density $\rho_0 = 1$. The initial pressure is p_0 everywhere (with $p_0 = 10^{-6}$ or $p_0 = 10^{-10}$) except in the cell V_{or} containing the origin \mathbf{O} where it is given by

$$p_{or} = (\gamma - 1)\rho_0 \frac{E_{tot}}{|V_{or}|}, \quad \text{with } E_{tot} = 0.979264, \quad (241)$$

Table 17: In this Table we report some statistics on the computational cost of reaching the final computational time $t_f = 60$ with four different schemes. In the second column we report the percentage of computational time employed by mesh regeneration and space time connectivity generation, in the third column there is the percentage of time needed for the predictor-corrector step on standard elements, and in the last column the percentage of time spent on sliver elements. It is evident that the cost due to mesh rearrangement and sliver computations is minimal. (For what concern FV schemes, the time for spatial reconstruction is not included in the third column, in order to facilitate the comparison between the cost on standard elements and sliver elements, for which spatial reconstruction is not performed.)

Method	Mesh %	$P_N P_M$ standard %	$P_N P_M$ sliver %
FV $\mathcal{O}(3)$	0.31	56.03	0.01
FV $\mathcal{O}(4)$	0.31	56.00	0.01
DG $\mathcal{O}(3)$	1.29	91.07	0.004
DG $\mathcal{O}(4)$	0.25	96.31	0.001

being E_{tot} the total energy concentrated at $\mathbf{x} = \mathbf{0}$. We solve this numerical test with a second order $P_1 P_1$ DG scheme for $p_0 = 10^{-6}$ and a third order DG scheme for $p_0 = 10^{-10}$; we employ a mesh of 7234 Voronoi elements. The density profiles are shown in Figure 47 for various output times $t = 0, 0.2, 0.5, 0.8, 1.0$. The obtained results are in good agreement with the literature. Moreover, one can refer to Figure 48 for a comparison between our numerical solution (*scatter plot*) and the reference one: the position of the shock wave and the density high peak are perfectly captured. We remark that this is quite a challenging benchmark because of the low pressure and the strong shock.

Finally, we refer to the last panel of Figure 47 for the behavior of our *a posteriori* sub-cell finite volume limiter, which activates only and exactly where the shock wave is located.

7.6.4 Ideal MHD equations

We also consider the equations of ideal classical magnetohydrodynamics (MHD) that result in a more complicated system of hyperbolic conservation laws. The state vector \mathbf{Q} and the flux tensor \mathbf{F} for the MHD equations in the general form (1) are

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \\ \mathbf{B} \\ \psi \end{pmatrix}, \quad \mathbf{F}(\mathbf{Q}) = \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p_t \mathbf{I} - \frac{1}{4\pi} \mathbf{B} \otimes \mathbf{B} \\ \mathbf{v}(\rho E + p_t) - \frac{1}{4\pi} \mathbf{B}(\mathbf{v} \cdot \mathbf{B}) \\ \mathbf{v} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{v} + \psi \mathbf{I} \\ c_h^2 \mathbf{B} \end{pmatrix}. \quad (242)$$

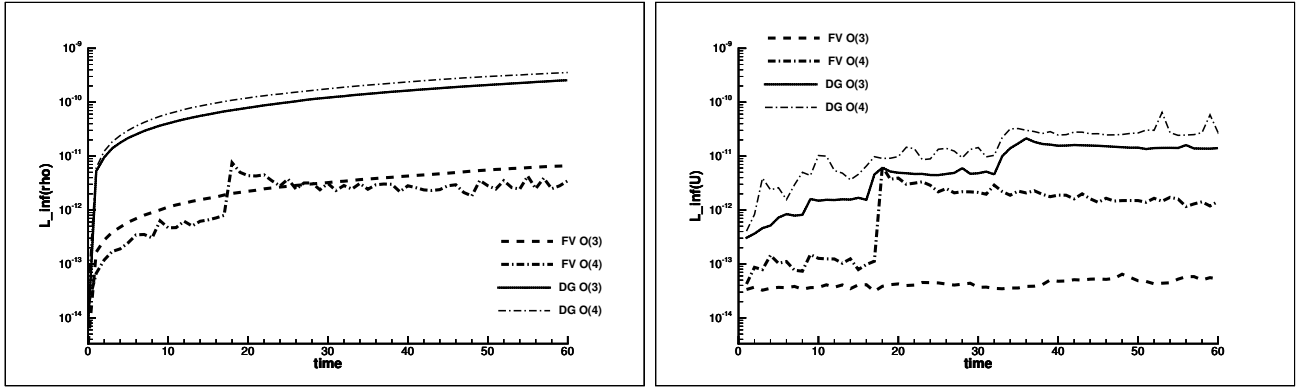


Fig. 45: In this Figure we report the error in L -infinity norm of the error between our numerical results and the exact constant solution for the density ρ (left) and the velocity $U = \sqrt{u^2 + v^2}$ (right) for four different methods, namely a third and fourth order Finite Volume scheme and a third and fourth order Discontinuous Galerkin scheme. We underline that the final computational time of $t_f = 60$ is reached after *thousands* of time steps and involves the generations of *thousands* of sliver elements, as reported in Table 16.

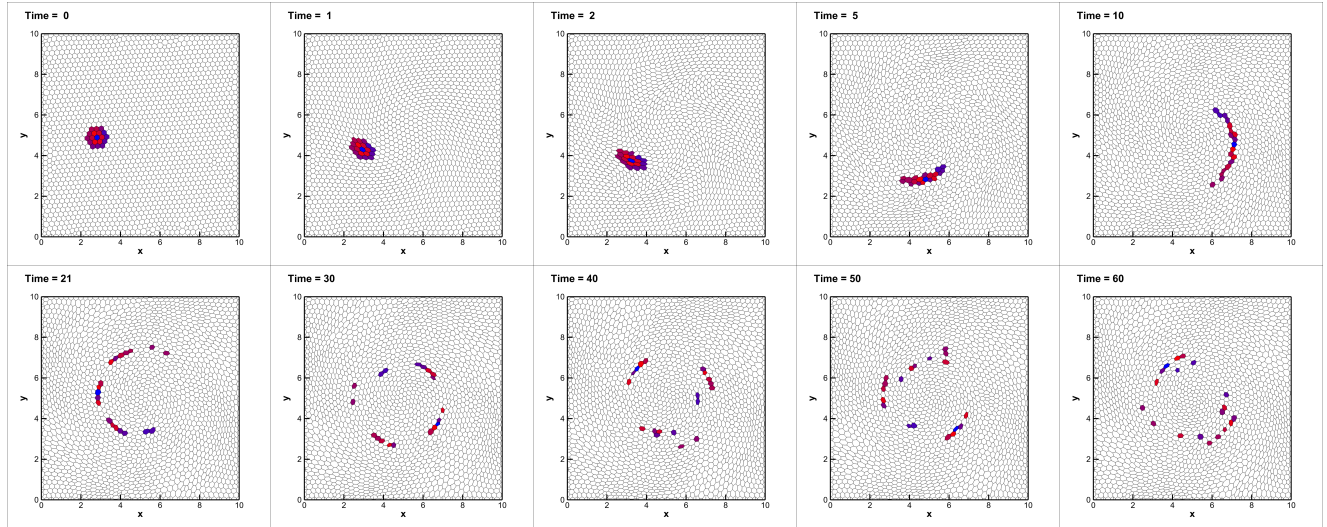


Fig. 46: In this Figure we report the position of a bunch of elements initially neighbors subjected to the velocity field (240) at different computational times. The Voronoi tassellation is regenerated at any of the 33019 timesteps of the employed third order P_2P_2 DG schemes. In total 19392 sliver element have been treated during this simulation (and the MOOD procedure has never been activated).

Here, $\mathbf{B} = (B_x, B_y, B_z)$ represents the magnetic field and $p_t = p + \frac{1}{8\pi} \mathbf{B}^2$ is the total pressure. The hydrodynamic pressure is given by the equation of state used to close the system, thus

$$p = (\gamma - 1) \left(\rho E - \frac{1}{2} \mathbf{v}^2 - \frac{\mathbf{B}^2}{8\pi} \right). \quad (243)$$

System (242) requires an additional constraint on the divergence of the magnetic field to be satisfied, that is

$$\nabla \cdot \mathbf{B} = 0. \quad (244)$$

Here, (242) includes one additional scalar PDE for the evolution of the variable ψ , which is needed to transport diver-

gence errors outside the computational domain with an artificial divergence cleaning speed c_h , see [49]. A more recent and more sophisticated methodology to fulfill this condition exactly on the discrete level also in the context of high order ADER WENO finite volume schemes on unstructured simplex meshes can be found in [2]. A similar approach is adopted in [72, 18, 11].

7.6.5 MHD vortex

For the numerical convergence studies, we solve the vortex test problem proposed by Balsara in [1]. The computational domain is given by the box $\Omega = [0; 10] \times [0; 10]$ with wall

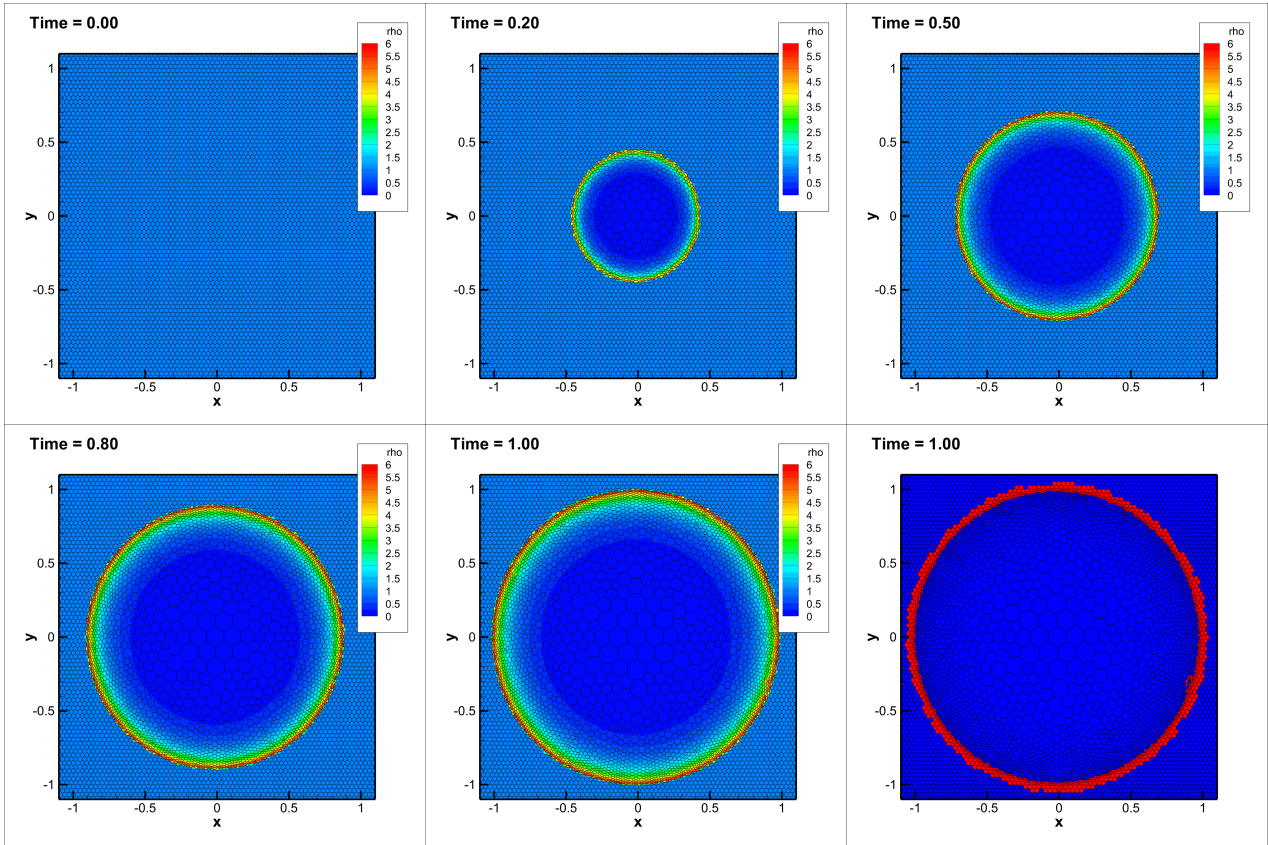


Fig. 47: Sedov problem solved with our P_2P_2 scheme on a moving Voronoi mesh of 7234 elements. We depict the density profile and the mesh configuration at times $t = 0, 0.2, 0.5, 0.8, 1$ and in the last images we show in red the cells on which the limiter is activated.

$P_0P_1 \rightarrow \mathcal{O}2$			$P_0P_2 \rightarrow \mathcal{O}3$			$P_0P_3 \rightarrow \mathcal{O}4$			$P_0P_4 \rightarrow \mathcal{O}5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$
4.6E-01	3.3E-02	-	3.2E-01	1.0E-02	-	4.7E-01	2.1E-02	-	6.0E-01	3.6e-02	-
3.9E-01	1.6E-02	1.8	2.4E-01	5.5E-03	2.3	3.2E-01	6.0E-03	3.2	5.8E-01	3.0e-02	5.8
2.4E-01	8.9E-03	2.3	1.9E-01	2.7E-03	3.3	2.4E-01	2.0E-03	3.9	5.6E-01	2.7e-02	3.6
1.9E-01	5.3E-03	2.4	1.6E-01	1.5E-03	3.1	2.2E-01	1.3E-03	3.6	5.5E-01	2.3e-02	5.9
1.6E-01	3.4E-03	2.5	1.4E-01	1.0E-03	2.9	1.9E-01	8.1E-04	4.8	5.2E-01	1.8e-02	4.8

Table 18: MHD vortex. Numerical convergence results for the finite volume algorithm on moving meshes with topology changes. The error norms refer to the variable ρ at time $t = 1.0$ in L_1 norm.

$P_1P_1 \rightarrow \mathcal{O}2$			$P_2P_2 \rightarrow \mathcal{O}3$			$P_3P_3 \rightarrow \mathcal{O}4$			$P_4P_4 \rightarrow \mathcal{O}5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$\mathcal{O}(L_1)$
4.7E-01	8.5E-03	-	6.1E-01	2.8E-03	-	8.8E-01	1.1E-03	-	1.6E-00	6.9e-03	-
3.2E-01	3.2E-04	2.5	4.7E-01	1.3E-03	2.8	7.5E-01	6.2E-04	3.5	6.1E-01	1.3e-04	4.1
2.8E-01	2.1E-04	2.9	3.8E-01	7.3E-04	2.7	6.1E-01	3.1E-04	3.4	5.2E-01	4.7e-05	5.8
2.4E-01	1.6E-04	2.0	3.5E-01	5.6E-04	3.6	5.5E-01	1.9E-04	4.3	4.9E-01	3.1e-05	8.1
1.9E-01	9.7E-05	2.4	3.2E-01	4.1E-04	3.0	3.2E-01	2.3E-05	3.9	4.7E-01	2.4e-05	5.3

Table 19: MHD vortex. Numerical convergence results for the discontinuous Galerkin algorithm on moving meshes with topology changes. The error norms refer to the variable ρ at time $t = 1.0$ in L_1 norm.

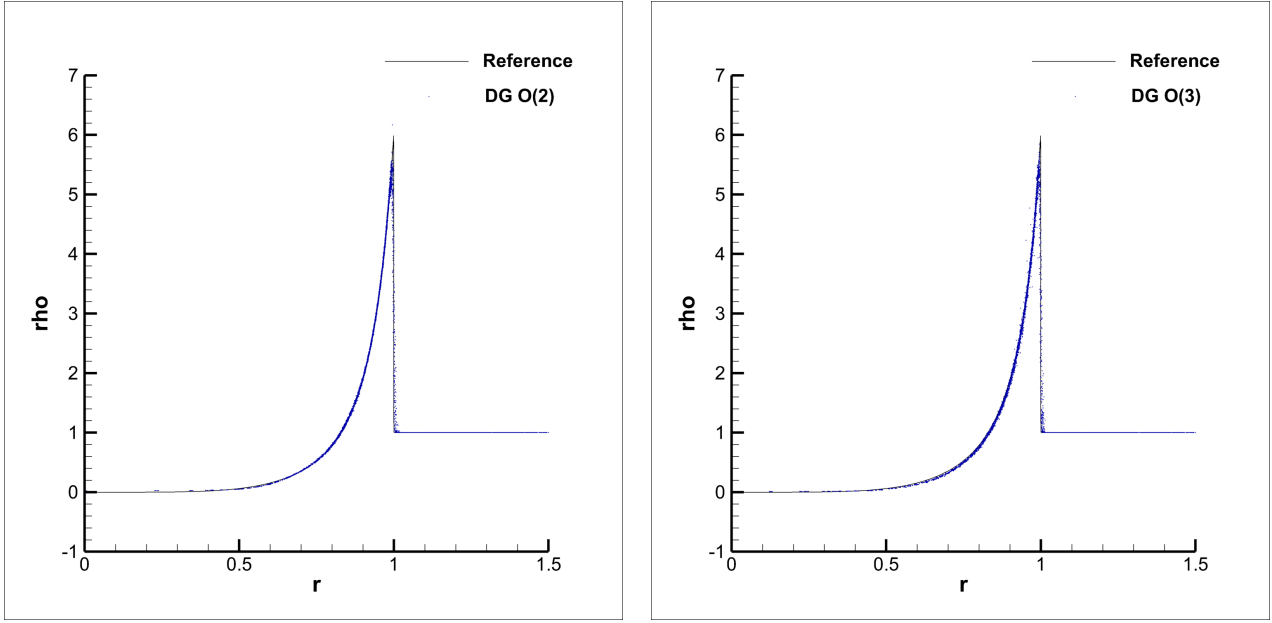


Fig. 48: Scatter plot of density values (blue dots) for the Sedov problem compared with the reference solution (line). Left: the initial setting is with outer pressure equal to $p_0 = 10^{-6}$ and it is solved with our DG scheme of order 2. Right: the initial setting is with outer pressure equal to $p_0 = 10^{-10}$ and it is solved with our DG scheme of order 3. In the two cases the mesh moves with the fluid flow and topology changes occur.

Table 20: In this Table we report the number of total time steps needed to reach the final computational time $t_f = 70$ with our P_3P_3 DG scheme, the number of sliver elements that have been originated along the simulations due to the occurred topology changes and the number of time steps that have been repeated through the MOOD loop. Moreover, we report the percentage of computational time employed by mesh regeneration and space time connectivity generation, by the predictor-corrector step on standard elements and on sliver elements. It is evident that the cost due to mesh rearrangement and sliver computations is minimal, that MOOD restart activates very rarely and that sliver elements are an essential ingredient of our scheme.

Method	time steps	slivers	restarts	Mesh %	$P_N P_M$ standard %	$P_N P_M$ sliver %
DG $\mathcal{O}(4)$	62741	21369	3	0.17	97.39	7.5E-4

boundary conditions imposed everywhere. The initial condition is given in terms of the vector of primitive variables $\mathbf{V} = (\rho, u, v, w, p, B_x, B_y, B_z, \Psi)^T$ as

$$\mathbf{V}(\mathbf{x}, 0) = (1, \delta u, \delta v, 0, 1 + \delta p, \delta B_x, \delta B_y, 0, 0)^T, \quad (245)$$

with $\delta \mathbf{v} = (\delta u, \delta v, 0)^T$, $\delta \mathbf{B} = (\delta B_x, \delta B_y, 0)^T$ and

$$\begin{aligned} \delta \mathbf{v} &= \frac{\kappa}{2\pi} e^{q(1-r^2)} \mathbf{e}_z \times \mathbf{r} \\ \delta \mathbf{B} &= \frac{\mu}{2\pi} e^{q(1-r^2)} \mathbf{e}_z \times \mathbf{r}, \\ \delta p &= \frac{1}{64q\pi^3} (\mu^2(1-2qr^2) - 4\kappa^2\pi) e^{2q(1-r^2)}. \end{aligned} \quad (246)$$

We have $\mathbf{e}_z = (0, 0, 1)$, $\mathbf{r} = (x-5, y-5, 0)$ and $r = \|\mathbf{r}\| = \sqrt{(x-5)^2 + (y-5)^2}$. The divergence cleaning speed is chosen as $c_h = 3$. The other parameters are $q = \frac{1}{2}$, $\kappa = 1$ and $\mu = \sqrt{4\pi}$, according to [1].

Convergence. Tables 18 and 19 report the convergence rates from second up to fifth order of accuracy for the MHD vortex test problem run on a sequence of successively refined meshes up to the final time $t = 1.0$. The optimal order of accuracy is achieved both in space and time for the FV schemes as well as for the DG schemes.

Quality. In Figure 49 we show the pressure profile and the magnetic field obtained with our fourth order P_3P_3 DG scheme at the initial time and after a long simulation with $t_f = 65$. Once again, the profile of the vortex is simulated and conserved for a longer computational time with respect to standard conforming ALE scheme, for which mesh tangling would occur and stop the simulation earlier. Moreover, a scatter plot of the constant density profile and of the pressure profile are reported in Figure 51: the fact that the ρ value is only

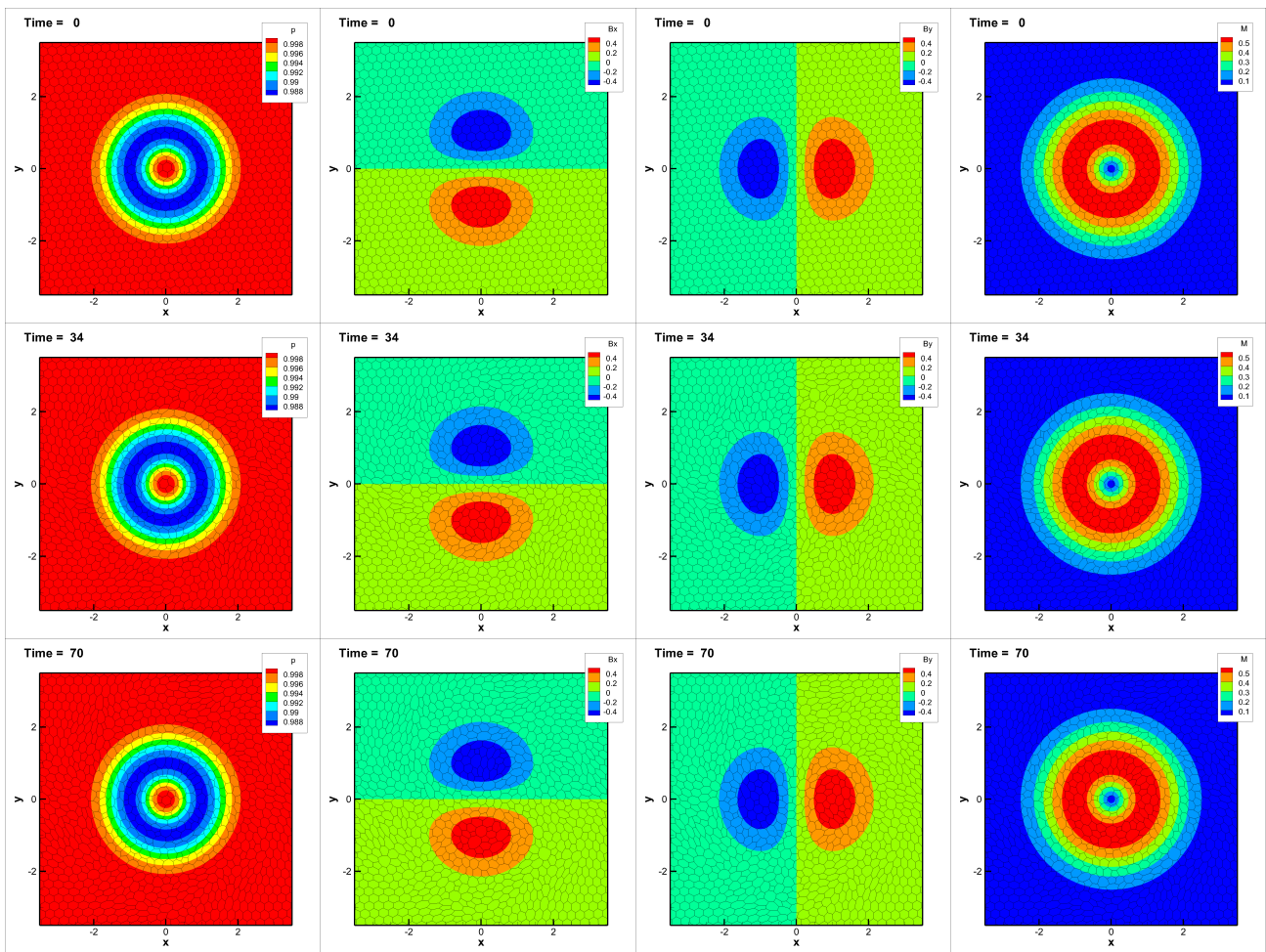


Fig. 49: MHD vortex solved with our P_3P_3 DG scheme on a moving Voronoi mesh of 1900 elements: we depict, from left to right, the pressure profile, the x - and y - components of the magnetic field, and the value of $M = \sqrt{B_x^2 + B_y^2}$ at the initial time $t = 0$, at an intermediate time $t = 34$ and at the final time of $t_f = 70$, corresponding to 2 complete loops of the elements located at $r = 1$. The connectivity changes (see also Figure 50) together with the high order methods allow to preserve the stationary MHD vortex for long times.

perturbed by standard numerical errors is another proof that our scheme satisfies the CGL condition.

In Figure 50 we report the position of a bunch of initially neighbors elements at different times: this makes it clear how strong the rotation is to which the mesh elements are subjected and the freedom that should be allowed to them in order to preserve a high quality mesh.

Finally in Table 20, we report some statistic on the number of sliver elements created over the total number of time steps, and on the percentage of computational times required both for the geometrical part of the code and for the $P_N P_M$ predictor-corrector algorithm.

8 Conclusion

To conclude, I would like to underline that the research presented in this article is based on three powerful founding concepts. First the space time conservation formulation of the governing PDEs which naturally applies to the novel moving ALE framework even with topology changes as it is based on the integration on *arbitrarily shaped* closed control volumes; second, the ADER predictor-corrector paradigm that allows to obtain high order of accuracy both in space and in time in one single step procedure; and third, the idea to treat the nonconservative products in a well balanced way by directly employing the steady equilibrium in the path construction. These original ideas have been applied in different contexts and coupled together obtaining promising results that justify further research in many directions.

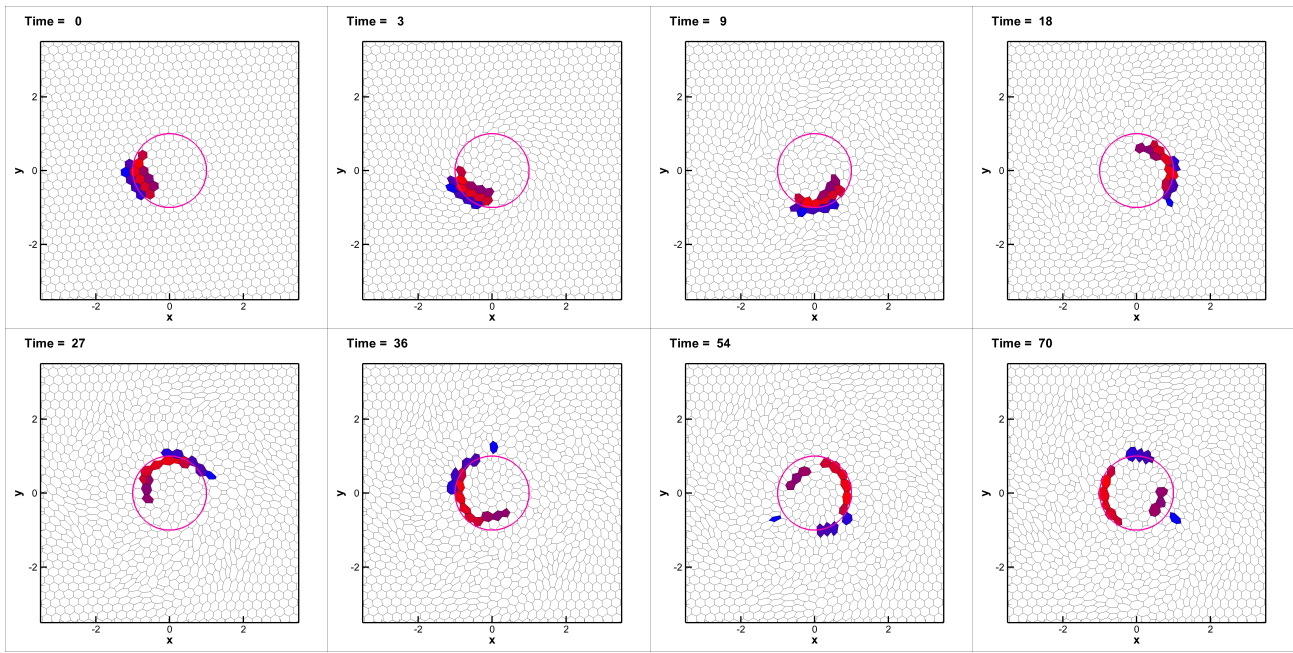


Fig. 50: MHD vortex solved with our P_3P_3 DG scheme on a moving Voronoi mesh of 1900 elements. In this Figure we show the position of a bunch of initially neighbors elements at different times, namely at $t = 70$. In this way one can notice the evolution of the grid topology during time and the necessity of allowing the mesh changing its topology in order to correctly follow the fluid velocities without distortion.

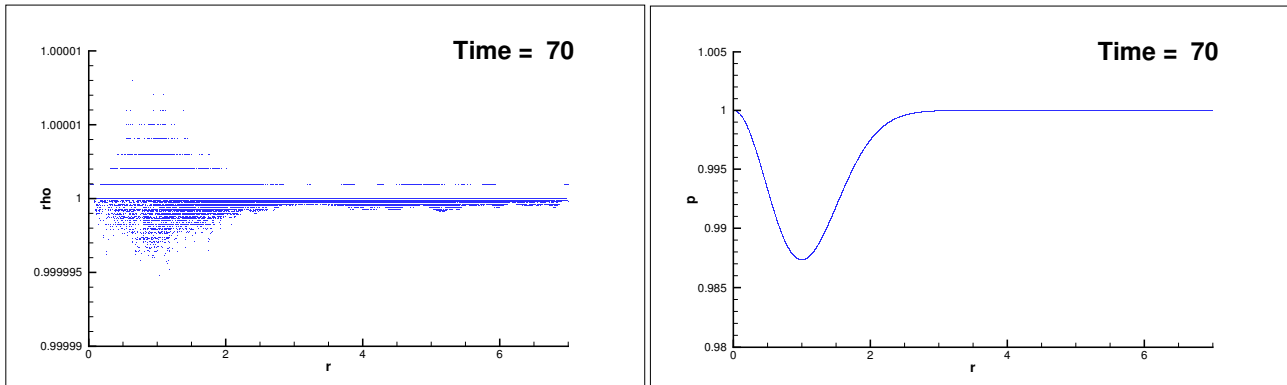


Fig. 51: MHD vortex solved with our P_3P_3 DG scheme on a moving Voronoi mesh of 1900 elements. We report the *scatter* plot of the constant density profile (left) and of the pressure profile (right) at the final time $t_f = 70$.

With this work we can benefit simultaneously of very high order methods, high quality grids and substantially reduced numerical dissipation, and we would like to emphasize that usually at least one of the previous ingredients is missing. Lagrangian methods, which almost cancel advection errors, are usually affected by dangerous mesh distortions, and available algorithms which are able to avoid it are only low order accurate; Eulerian methods are in general high order accurate, but limited by dissipation errors due to the advective terms; well balancing techniques are rarely coupled with unstructured moving meshes.

The results on vortical flows give evidence of the advantages conveyed by the proposed algorithms, and a large set of different numerical tests shows their robustness and efficiency. In particular, the introduced techniques enables the resolution even of complex shear flows with differential rotations in an effective and accurate way. It is noteworthy to stress again that standard conforming or topology-fixed Lagrangian schemes will *crash* after *finite times* for any vortex flow with differential rotation due to *mesh tangling*. Moreover the reduced dissipation characterizing the Lagrangian methods, together with the high mesh quality provided by the nonconforming treatment of sliding lines,

and the increased accuracy near the equilibria given by the well balanced techniques, allow us to obtain significant improvements compared to the existing state of the art. The major benefits are achieved with our new class of schemes when studying physical phenomena that arise close to stationary solutions, where standard discretizations would hide the flow physics with spurious oscillations and excessive numerical dissipation.

Future research will consider an extension to three space dimensions as well as to more general classes of stationary solutions and an automatic detector of the equilibrium profiles in order to extend our method to situations in which the equilibrium is not known exactly *a priori*.

Furthermore, based on the high order path-conservative methods introduced in [71] we intend to use the algorithms developed in this paper in order to design exactly well balanced schemes for gravity driven equilibrium flows in *general relativity*, where the use of well balanced methods appears to be still rather unknown.

Acknowledgements The research presented in this paper has been partially financed by a *national mobility grant for young researchers* in Italy, funded by GNCS-INdAM, and by the *UniTN Starting Grant*, funded by the University of Trento in Italy.

Conflict of interest

The author declares that she has no conflict of interest.

References

- Balsara, D.: Second-order accurate schemes for magnetohydrodynamics with divergence-free reconstruction. *The Astrophysical Journal Supplement Series* **151**, 149–184 (2004)
- Balsara, D., Dumbser, M.: Divergence-free MHD on unstructured meshes using high order finite volume schemes based on multidimensional Riemann solvers. *Journal of Computational Physics* pp. 687–715 (2015)
- Barlow, A., Maire, P., Rider, W., Rieben, R., Shashkov, M.: Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows. *Journal of Computational Physics* **322**, 603–665 (2016)
- Barth, T., Frederickson, P.: Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. AIAA paper no. 90-0013 (1990)
- Barth, T., Jespersen, D.: The design and application of upwind schemes on unstructured meshes. AIAA Paper 89-0366 pp. 1–12 (1989)
- Benson, D.J.: Computational methods in lagrangian and eulerian hydrocodes. *Computer Methods in Applied Mechanics and Engineering* **99**(2), 235 – 394 (1992). DOI [https://doi.org/10.1016/0045-7825\(92\)90042-I](https://doi.org/10.1016/0045-7825(92)90042-I). URL <http://www.sciencedirect.com/science/article/pii/004578259290042I>
- Berndt, M., Breil, J., Galera, S., Kucharik, M., Maire, P., Shashkov, M.: Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods. *Journal of Computational Physics* **230**, 6664–6687 (2011)
- Bertoluzza, S., Pino, S.D., Labourasse, E.: A conservative slide line method for cell-centered semi-Lagrangian and ALE schemes in 2D. *ESAIM: Mathematical Modelling and Numerical Analysis (M2AN)* **50**, 187–214 (2016)
- Bochev, P., Ridzal, D., Shashkov, M.: Fast optimization-based conservative remap of scalar fields through aggregate mass transfer. *Journal of Computational Physics* **246**, 37–57 (2013)
- Boscheri, W.: An efficient high order direct ALE ADER finite volume scheme with a posteriori limiting for hydrodynamics and magnetohydrodynamics. *International Journal for Numerical Methods in Fluids* **84**, 76–106 (2017)
- Boscheri, W.: An efficient high order direct ale ader finite volume scheme with a posteriori limiting for hydrodynamics and magnetohydrodynamics. *International Journal for Numerical Methods in Fluids* **84**(2), 76–106 (2017)
- Boscheri, W.: High order direct arbitrary-lagrangian–eulerian (ale) finite volume schemes for hyperbolic systems on unstructured meshes. *Archives of Computational Methods in Engineering* **24**(4), 751–801 (2017)
- Boscheri, W., Balsara, D., Dumbser, M.: Lagrangian ADER-WENO Finite Volume Schemes on Unstructured Triangular Meshes Based On Genuinely Multidimensional HLL Riemann Solvers. *Journal of Computational Physics* **267**, 112–138 (2014)
- Boscheri, W., Dumbser, M.: Arbitrary–Lagrangian–Eulerian One–Step WENO Finite Volume Schemes on Unstructured Triangular Meshes. *Communications in Computational Physics* **14**, 1174–1206 (2013)
- Boscheri, W., Dumbser, M.: A direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3d. *Journal of Computational Physics* **275**, 484 – 523 (2014)
- Boscheri, W., Dumbser, M.: High order accurate direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume schemes on moving curvilinear unstructured meshes. *Computers and Fluids* **136**, 48–66 (2016)
- Boscheri, W., Dumbser, M.: Arbitrary-Lagrangian-Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes. *Journal of Computational Physics* **346**, 449 – 479 (2017)
- Boscheri, W., Dumbser, M., Balsara, D.: High-order ader-weno ale schemes on unstructured triangular meshes—application of several node solvers to hydrodynamics and magnetohydrodynamics. *International Journal for Numerical Methods in Fluids* **76**(10), 737–778 (2014)
- Boscheri, W., Dumbser, M., Balsara, D.: High Order Lagrangian ADER-WENO Schemes on Unstructured Meshes – Application of Several Node Solvers to Hydrodynamics and Magnetohydrodynamics. *International Journal for Numerical Methods in Fluids* **76**, 737–778 (2014)
- Boscheri, W., Dumbser, M., Righetti, M.: A semi-implicit scheme for 3d free surface flows with high-order velocity reconstruction on unstructured voronoi meshes. *International journal for numerical methods in fluids* **72**(6), 607–631 (2013)
- Boscheri, W., Dumbser, M., Zanotti, O.: High Order Cell-Centered Lagrangian-Type Finite Volume Schemes with Time-Accurate Local Time Stepping on Unstructured Triangular Meshes. *Journal of Computational Physics* **291**, 120–150 (2015)
- Boscheri, W., Loubère, R.: High order accurate direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for non-conservative hyperbolic systems with stiff source terms. *Communications in Computational Physics* **21**, 271–312 (2017)
- Boscheri, W., Loubère, R., Dumbser, M.: Direct arbitrary-lagrangian–eulerian ader-mood finite volume schemes for multidimensional hyperbolic conservation laws. *Journal of Computational Physics* **292**, 56–87 (2015)

24. Boscheri, W., Loubère, R., Dumbser, M.: Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws. *Journal of Computational Physics* **292**, 56–87 (2015)
25. Boscheri, W., Pisaturo, G.R., Righetti, M.: High-order divergence-free velocity reconstruction for free surface flows on unstructured voronoi meshes. *International Journal for Numerical Methods in Fluids* **90**(6), 296–321 (2019)
26. Boscheri, W., Semplice, M., Dumbser, M.: Central WENO Sub-cell Finite Volume Limiters for ADER Discontinuous Galerkin Schemes on Fixed and Moving Unstructured Meshes. *Communications in Computational Physics* **25**, 311–346 (2019)
27. Busto, S., Ferrín, J., Toro, E.F., Vázquez-Cendón, M.E.: A projection hybrid high order finite volume/finite element method for incompressible turbulent flows. *Journal of Computational Physics* **353**, 169–192 (2018)
28. Caramana, E.: The implementation of slide lines as a combined force and velocity boundary condition. *Journal of Computational Physics* **228**, 3911–3916 (2009)
29. Caramana, E., Burton, D., Shashkov, M., Whalen, P.: The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *Journal of Computational Physics* **146**, 227–262 (1998)
30. Carré, G., Pino, S.D., Després, B., Labourasse, E.: A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics* **228**, 5160–5183 (2009)
31. Castro, M., Gallardo, J., López, J., Parés, C.: Well-balanced high order extensions of godunov’s method for semilinear balance laws. *SIAM Journal of Numerical Analysis* **46**, 1012–1039 (2008)
32. Castro, M., Gallardo, J., Marquina, A.: Approximate Osher–Solomon schemes for hyperbolic systems. *Applied Mathematics and Computation* **272**, 347–368 (2016)
33. Castro, M., Gallardo, J., Parés, C.: High-order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. Applications to shallow-water systems. *Mathematics of Computation* **75**, 1103–1134 (2006)
34. Castro, M., Pardo, A., Parés, C.: Well-balanced numerical schemes based on a generalized hydrostatic reconstruction technique. *Mathematical Models and Methods in Applied Sciences* **17**(12), 2055–2113 (2007)
35. Castro, M., Pardo, A., Parés, C., Toro, E.: On some fast well-balanced first order solvers for nonconservative systems. *Mathematics of computation* **79**(271), 1427–1472 (2010)
36. Castro, M.J., Fernández, E., Ferriero, A., García, J.A., Parés, C.: High order extensions of Roe schemes for two dimensional non-conservative hyperbolic systems. *Journal of Scientific Computing* **39**, 67–114 (2009)
37. Castro Díaz, M.J., Fernández-Nieto, E.D.: A class of computationally fast first order finite volume solvers: Pvm methods. *SIAM J. Scientific Computing* **34**(4) (2012)
38. Cheng, J., Shu, C.: A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics* **227**, 1567–1596 (2007)
39. Cheng, J., Shu, C.: A cell-centered Lagrangian scheme with the preservation of symmetry and conservation properties for compressible fluid flows in two-dimensional cylindrical geometry. *Journal of Computational Physics* **229**, 7191–7206 (2010)
40. Clain, S., Diot, S., Loubère, R.: A high-order finite volume method for systems of conservation laws—multi-dimensional optimal order detection (MOOD). *Journal of Computational Physics* **230**(10), 4028 – 4050 (2011). DOI 10.1016/j.jcp.2011.02.026. URL <http://www.sciencedirect.com/science/article/pii/S002199911100115X>
41. Clair, G., Després, B., Labourasse, E.: A new method to introduce constraints in cell-centered Lagrangian schemes. *Computer Methods in Applied Mechanics and Engineering* **261-262**, 56–65 (2013)
42. Clair, G., Després, B., Labourasse, E.: A one-mesh method for the cell-centered discretization of sliding. *Computer Methods in Applied Mechanics and Engineering* **269**, 315–333 (2014)
43. Claisse, A., Després, B., E.Labourasse, Ledoux, F.: A new exceptional points method with application to cell-centered Lagrangian schemes and curved meshes. *Journal of Computational Physics* **231**, 4324–4354 (2012)
44. Cravero, I., Puppo, G., Semplice, M., Visconti, G.: Cweno: uniformly accurate reconstructions for balance laws. *Mathematics of Computation* **87**(312), 1689–1719 (2018)
45. Cremonesi, M., Frangi, A., Perego, U.: A lagrangian finite element approach for the analysis of fluid–structure interaction problems. *International Journal for Numerical Methods in Engineering* **84**(5), 610–630 (2010)
46. Cremonesi, M., Frangi, A., Perego, U.: A lagrangian finite element approach for the simulation of water-waves induced by landslides. *Computers & Structures* **89**(11-12), 1086–1093 (2011)
47. Cremonesi, M., Meduri, S., Perego, U., Frangi, A.: An explicit lagrangian finite element method for free-surface weakly compressible flows. *Computational Particle Mechanics* **4**(3), 357–369 (2017)
48. Dal Maso, G., LeFloch, P., Murat, F.: Definition and weak stability of nonconservative products. *J. Math. Pures Appl.* **74**, 483–548 (1995)
49. Dedner, A., Kemm, F., Kröner, D., Munz, C.D., Schnitzer, T., Wesenberg, M.: Hyperbolic divergence cleaning for the MHD equations. *Journal of Computational Physics* **175**, 645–673 (2002)
50. Després, B.: *Numerical Methods for Eulerian and Lagrangian Conservation Laws*. Birkhäuser (2017)
51. Diot, S., Clain, S., Loubère, R.: Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Computers and Fluids* **64**, 43 – 63 (2012). DOI 10.1016/j.compfluid.2012.05.004. URL <http://www.sciencedirect.com/science/article/pii/S0045793012001909>
52. Diot, S., Loubère, R., Clain, S.: The MOOD method in the three-dimensional case: Very-high-order finite volume method for hyperbolic systems. *International Journal of Numerical Methods in Fluids* **73**, 362–392 (2013)
53. Dobrev, V., Ellis, T., Kolev, T., Rieben, R.: Curvilinear Finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids* **65**, 1295–1310 (2011)
54. Dobrev, V., Ellis, T., Kolev, T., Rieben, R.: High-order curvilinear finite elements for axisymmetric lagrangian hydrodynamics. *Computers & Fluids* **83**(0), 58 – 69 (2013)
55. Dobrev, V., Kolev, T., Rieben, R.: High-order curvilinear finite element methods for lagrangian hydrodynamics. *SIAM Journal on Scientific Computing* **34**(5), B606–B641 (2012). DOI 10.1137/120864672
56. Dumbser, M.: Arbitrary–Lagrangian–Eulerian ADER–WENO finite volume schemes with time–accurate local time stepping for hyperbolic conservation laws. *Computer Methods in Applied Mechanics and Engineering* **280**, 57–83 (2014)
57. Dumbser, M., Balsara, D.: A new, efficient formulation of the HLLEM Riemann solver for general conservative and non-conservative hyperbolic systems. *Journal of Computational Physics* **304**, 275–319 (2016)
58. Dumbser, M., Balsara, D., Toro, E., Munz, C.: A unified framework for the construction of one–step finite–volume and discontinuous Galerkin schemes. *Journal of Computational Physics* **227**, 8209–8253 (2008)

59. Dumbser, M., Boscheri, W.: High-order unstructured lagrangian one-step weno finite volume schemes for non-conservative hyperbolic systems: applications to compressible multi-phase flows. *Computers & Fluids* **86**, 405–432 (2013)
60. Dumbser, M., Boscheri, W., Semplice, M., Russo, G.: Central weighted eno schemes for hyperbolic conservation laws on fixed and moving unstructured meshes. *SIAM Journal on Scientific Computing* **39**(6), A2564–A2591 (2017)
61. Dumbser, M., Enaux, C., Toro, E.: Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics* **227**, 3971–4001 (2008)
62. Dumbser, M., Kaeser, M.: Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics* **221**, 693–723 (2007)
63. Dumbser, M., Kaeser, M.: Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics* **221**, 693–723 (2007)
64. Dumbser, M., Käser, M.: Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics* **221**, 693–723 (2007)
65. Dumbser, M., Käser, M., Titarev, V., Toro, E.: Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics* **226**, 204–243 (2007)
66. Dumbser, M., Loubère, R.: A simple robust and accurate a posteriori sub-cell finite volume limiter for the discontinuous Galerkin method on unstructured meshes. *Journal of Computational Physics* **319**, 163–199 (2016)
67. Dumbser, M., Toro, E.F.: On universal Osher–type schemes for general nonlinear hyperbolic conservation laws. *Communications in Computational Physics* **10**, 635–671 (2011)
68. Dumbser, M., Toro, E.F.: A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *Journal of Scientific Computing* **48**, 70–88 (2011)
69. Dumbser, M., Zanotti, O., Loubère, R., Diot, S.: A posteriori sub-cell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics* **278**, 47–75 (2014)
70. Einfeldt, B., Munz, C.D., Roe, P.L., Sjögreen, B.: On godunov-type methods near low densities. *Journal of Computational Physics* **92**, 273–295 (1991)
71. Fambri, F., Dumbser, M., Köppel, S., Rezzolla, L., Zanotti, O.: ADER discontinuous Galerkin schemes for general-relativistic ideal magnetohydrodynamics. *Monthly Notices of the Royal Astronomical Society (MNRAS)* (2018). Submitted. <https://arxiv.org/abs/1801.02839>
72. Fambri, F., Dumbser, M., Zanotti, O.: Space–time adaptive aderg schemes for dissipative flows: Compressible navier–stokes and resistive mhd equations. *Computer Physics Communications* **220**, 297–318 (2017)
73. F.Vilar: Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics. *Computers and Fluids* **64**, 64–73 (2012)
74. F.Vilar, Maire, P., Abgrall, R.: Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics. *Computers and Fluids* **46**(1), 498–604 (2010)
75. F.Vilar, Maire, P., Abgrall, R.: A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids. *Journal of Computational Physics* **276**, 188–234 (2014)
76. Gaburro, E.: Well balanced arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming meshes for non-conservative hyperbolic systems. Ph.D. thesis, University of Trento (2018)
77. Gaburro, E., Boscheri, W., Chiocchetti, S., Klingenberg, C., Springel, V., Dumbser, M.: High order direct arbitrary-lagrangian-eulerian schemes on moving voronoi meshes with topology changes. arXiv preprint arXiv:1905.00967 (2019)
78. Gaburro, E., Castro, M.J., Dumbser, M.: Well-balanced arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming meshes for the euler equations of gas dynamics with gravity. *Monthly Notices of the Royal Astronomical Society* **477**(2), 2251–2275 (2018)
79. Gaburro, E., Castro, M.J., Dumbser, M.: A well balanced diffuse interface method for complex nonhydrostatic free surface flows. *Computers & Fluids* **175**, 180–198 (2018)
80. Gaburro, E., Dumbser, M., Castro, M.J.: Direct arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming unstructured meshes. *Computers and Fluids* **159**, 254–275 (2017)
81. Gaburro, E., Dumbser, M., Castro, M.J.: Reprint of: Direct arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming unstructured meshes. *Computers & Fluids* (2018)
82. Galera, S., Maire, P., Breil, J.: A two-dimensional unstructured cell-centered multi-material ale scheme using vof interface reconstruction. *Journal of Computational Physics* **229**, 5755–5787 (2010)
83. Godunov, S.: Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics. *Mathematics of the USSR: Sbornik* **47**, 271–306 (1959)
84. Gosse, L.: A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Computers & Mathematics with Applications* **39**(9), 135–159 (2000)
85. Gosse, L.: A well-balanced scheme using non-conservative products designed for hyperbolic systems of conservation laws with source terms. *Mathematical Models and Methods in Applied Sciences* **11**(02), 339–365 (2001)
86. Greenberg, J., Leroux, A., Baraille, R., Noussair, A.: Analysis and approximation of conservation laws with source terms. *SIAM Journal on Numerical Analysis* **34**(5), 1980–2007 (1997)
87. Greenberg, J.M., Leroux, A.Y.: A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis* **33**(1), 1–16 (1996)
88. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.: Uniformly high order accurate essentially non-oscillatory schemes III. *Journal of Computational Physics* **71**, 231–303 (1987)
89. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.: Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics* **71**, 231–303 (1987)
90. Hidalgo, A., Dumbser, M.: Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations. *Journal of Scientific Computing* **48**(1-3), 173–189 (2011)
91. Hu, C., Shu, C.: A high-order weno finite difference scheme for the equations of ideal magnetohydrodynamics. *Journal of Computational Physics* **150**, 561–594 (1999)
92. Hu, C., Shu, C.: Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics* **150**(1), 97–127 (1999)
93. Idelsohn, S., Mier-Torrecilla, M., Oñate, E.: Multi-fluid flows with the Particle Finite Element Method. *Comput. Methods Appl. Mech. Engrg.* **198**, 2750–2767 (2009)
94. Idelsohn, S.R., Oñate, E., Pin, F.D.: The Particle Finite Element Method: a powerful tool to solve incompressible flows with free-surfaces and breaking waves. *International Journal for Numerical Methods in Engineering* **61**, 964–984 (2004)

95. Jackson, H.: On the eigenvalues of the ader-weno galerkin predictor. *Journal of Computational Physics* **333**, 409–413 (2017)
96. J.R. Cavalcanti M. Dumbser, D.d.M.M., Junior, C.F.: A Conservative Finite Volume Scheme with Time-Accurate Local Time Stepping for Scalar Transport on Unstructured Grids. *Advances in Water Resources* **86**, 217–230 (2015)
97. Käppeli, R., Mishra, S.: A well-balanced finite volume scheme for the euler equations with gravitation. *Astronomy and Astrophysics* **587**, A94 (2016)
98. Käser, M., Iske, A.: ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics* **205**, 486–508 (2005)
99. Knupp, P.: Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii – a framework for volume mesh optimization and the condition number of the jacobian matrix. *Int. J. Numer. Meth. Engng.* **48**, 1165 – 1185 (2000)
100. Kucharik, M., Breil, J., Galera, S., Maire, P., Berndt, M., Shashkov, M.: Hybrid remap for multi-material ALE. *Computers and Fluids* **46**, 293–297 (2011)
101. Kucharik, M., Loubère, R., Bednàrik, L., Liska, R.: Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition. *Computers & Fluids* **83**, 3–14 (2013)
102. Kucharik, M., Shashkov, M.: One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics* **231**, 2851–2864 (2012)
103. Larese, A., Rossi, R., Oñate, E., Idelsohn, S.: Validation of the Particle Finite Element Method (PFEM) for Simulation of the Free-Surface Flows. *Engineering Computations* **25**, 385–425 (2008)
104. LeVeque, R.J.: Balancing source terms and flux gradients in high-resolution godunov methods: the quasi-steady wave-propagation algorithm. *Journal of Computational Physics* **146**(1), 346–365 (1998)
105. Levy, D., Puppo, G., Russo, G.: Central WENO schemes for hyperbolic systems of conservation laws. *M2AN Math. Model. Numer. Anal.* **33**(3), 547–571 (1999)
106. Levy, D., Puppo, G., Russo, G.: A third order central WENO scheme for 2D conservation laws. *Applied Numerical Mathematics* **33**, 415–421 (2000)
107. Levy, D., Puppo, G., Russo, G.: A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws. *SIAM Journal on Scientific Computing* **24**, 480–506 (2002)
108. Li, Z., Yu, X., Jia, Z.: The cell-centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two dimensions. *Computers and Fluids* **96**, 152–164 (2014)
109. Liska, R., Váchal, M.S.P., Wendroff, B.: Synchronized flux corrected remapping for ALE methods. *Computers and Fluids* **46**, 312–317 (2011)
110. Liu, W., Cheng, J., Shu, C.: High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations. *Journal of Computational Physics* **228**, 8872–8891 (2009)
111. Loubere, R., Dumbser, M., Diot, S.: A new family of high order unstructured mood and ader finite volume schemes for multidimensional systems of hyperbolic conservation laws. *Communications in Computational Physics* **16**(3), 718–763 (2014)
112. Loubère, R., Maire, P., Váchal, P.: A second-order compatible staggered Lagrangian hydrodynamics scheme using a cell-centered multidimensional approximate Riemann solver. *Procedia Computer Science* **1**, 1931–1939 (2010)
113. Loubère, R., Maire, P., Váchal, P.: 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity. *International Journal for Numerical Methods in Fluids* **72**, 22 – 42 (2013)
114. Loubère, R., Maire, P.H., Váchal, P.: Staggered Lagrangian hydrodynamics based on cell-centered Riemann solver **10**(4), 940–978 (2010)
115. Ma, R., Chang, X., Zhang, L., He, X., Li, M.: On the geometric conservation law for unsteady flow simulations on moving mesh. *Procedia Engineering* **126**, 639–644 (2015)
116. Maire, P.: A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics* **228**, 2391–2425 (2009)
117. Maire, P.: A high-order one-step sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *Computers and Fluids* **46**(1), 341–347 (2011)
118. Maire, P.: A unified sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids. *International Journal for Numerical Methods in Fluids* **65**, 1281–1294 (2011)
119. Maire, P., Nkonga, B.: Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics* **228**, 799–821 (2009)
120. Maso, G.D., LeFloch, P., Murat, F.: Definition and weak stability of nonconservative products. *J. Math. Pures Appl.* **74**, 483–548 (1995)
121. Mignone, A., Bodo, G., Massaglia, S., Matsakos, T., Tesileanu, O., Zanni, C., Ferrari, A.: Pluto: A numerical code for computational astrophysics. *The Astrophysical Journal Supplement Series* **170**(1), 228 (2007)
122. Mignone, A., Zanni, C., Tzeferacos, P., Van Straalen, B., Colella, P., Bodo, G.: The pluto code for adaptive mesh computations in astrophysical fluid dynamics. *The Astrophysical Journal Supplement Series* **198**(1), 7 (2011)
123. Munz, C.: On Godunov-type schemes for Lagrangian gas dynamics. *SIAM Journal on Numerical Analysis* **31**, 17–42 (1994)
124. von Neumann, J., Richtmyer, R.: A method for the calculation of hydrodynamics shocks. *Journal of Applied Physics* **21**, 232–237 (1950)
125. Oñate, E., Celigueta, M., Idelsohn, S., Salazar, F., Suarez, B.: Possibilities of the Particle Finite Element Method for fluid–soil–structure interaction problems. *Journal of Computational Mechanics* **48**, 307–318 (2011)
126. Oñate, E., Idelsohn, S., Celigueta, M., Rossi, R.: Advances in the Particle Finite Element Method for the Analysis of Fluid-Multibody Interaction and Bed Erosion in Free-surface Flows. *Computer Methods in Applied Mechanics and Engineering* **197**, 1777–1800 (2008)
127. Ortega, A.L., Scovazzi, G.: A geometrically-conservative, synchronized, flux-corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements. *Journal of Computational Physics* **230**, 6709–6741 (2011)
128. Osher, S., Solomon, F.: Upwind difference schemes for hyperbolic conservation laws. *Math. Comput.* **38**, 339–374 (1982)
129. Pakmor, R., Marinacci, F., Springel, V.: Magnetic fields in cosmological simulations of disk galaxies. *The Astrophysical Journal Letters* **783**(1), L20 (2014)
130. Pakmor, R., Springel, V., Bauer, A., Mocz, P., Munoz, D.J., Ohlmann, S.T., Schaal, K., Zhu, C.: Improving the convergence properties of the moving-mesh code arepo. *Monthly Notices of the Royal Astronomical Society* **455**(1), 1134–1143 (2015)
131. Parés, C.: Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis* **44**, 300–321 (2006)
132. Pin, F.D., Idelsohn, S.R., Oñate, E., Aubry, R.: The ALE/Lagrangian Particle Finite Element Method: A new approach to computation of free-surface flows and fluid-object interactions. *Computers and Fluids* **36**, 27–38 (2007)
133. Pino, S.D.: A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates. *Comptes Rendus de l’Académie des Sciences - Series I - Mathematics* **348**, 1027–1032 (2010)
134. Qiu, J., Shu, C.W.: Hermite weno schemes and their application as limiters for runge–kutta discontinuous galerkin method ii: Two dimensional case. *Computers & Fluids* **34**(6), 642–663 (2005)

135. Re, B., Dobrzynski, C., Guardone, A.: An interpolation-free ale scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids. *Journal of Computational Physics* **340**, 26–54 (2017)
136. Reed, W., Hill, T.: Triangular mesh methods for neutron transport equation. Tech. Rep. LA-UR-73-479, Los Alamos Scientific Laboratory (1973)
137. Rusanov, V.V.: Calculation of Interaction of Non-Steady Shock Waves with Obstacles. *J. Comput. Math. Phys. USSR* **1**, 267–279 (1961)
138. Sambasivan, S., Shashkov, M., Burton, D.: A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids. *International Journal for Numerical Methods in Fluids* **72**, 770–810 (2013)
139. Schwartzkopff, T., Munz, C., Toro, E.: ADER: A high order approach for linear hyperbolic systems in 2d. *Journal of Scientific Computing* **17**(1-4), 231–240 (2002)
140. Scovazzi, G.: Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach. *Journal of Computational Physics* **231**, 8029–8069 (2012)
141. Sedov, L.: *Similarity and Dimensional Methods in Mechanics*. Academic Press, New York (1959)
142. Semplice, M., Coco, A., Russo, G.: Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction. *Journal of Scientific Computing* **66**(2), 692–724 (2016)
143. Springel, V.: E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh. *Monthly Notices of the Royal Astronomical Society (MNRAS)* **401**, 791–851 (2010)
144. Springel, V.: Moving-mesh hydrodynamics with the arepo code. *Proceedings of the International Astronomical Union* **6**(S270), 203–206 (2010)
145. Stroud, A.: *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey (1971)
146. Tavelli, M., Boscheri, W.: A high order parallel eulerian-lagrangian algorithm for advection-diffusion problems on unstructured meshes. *International Journal for Numerical Methods in Fluids*
147. Titarev, V., Toro, E.: ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing* **17**(1-4), 609–618 (2002)
148. Titarev, V., Toro, E.: ADER schemes for three-dimensional nonlinear hyperbolic systems. *Journal of Computational Physics* **204**, 715–736 (2005)
149. Toro, E.: *Riemann Solvers and Numerical Methods for Fluid Dynamics*, second edn. Springer (1999)
150. Toro, E., Titarev, V.: Solution of the generalized Riemann problem for advection-reaction equations. *Proc. Roy. Soc. London* pp. 271–281 (2002)
151. Toro, E.F., Titarev, V.A.: Derivative Riemann solvers for systems of conservation laws and ADER methods. *Journal of Computational Physics* **212**(1), 150–165 (2006)
152. van Leer, B.: Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme. *Journal of Computational Physics* **14**, 361–370 (1974)
153. van Leer, B.: Towards the ultimate conservative difference scheme V: A second order sequel to Godunov’s method. *Journal of Computational Physics* **32**, 101–136 (1979)
154. Wilkins, M.L.: Calculation of elastic-plastic flow. *Methods in Computational Physics* **3** (1964)
155. Winslow, A.M.: Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.* **135**(2), 128–138 (1997)
156. Zanutti, O., Fambri, F., Dumbser, M., Hidalgo, A.: Space-time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers and Fluids* **118**, 204–224 (2015)