High order direct Arbitrary-Lagrangian-Eulerian schemes on moving Voronoi meshes with topology changes

Elena Gaburro[*a], Walter Boscheri[b], Simone Chiocchetti[a], Christian Klingenberg[c], Volker Springel[d], Michael Dumbser[a]

[a]*Department of Civil, Environmental and Mechanical Engineering, University of Trento, Via Mesiano 77, 38123 Trento, Italy*
[b]*Department of Mathematics and Computer Science, University of Ferrara, via Machiavelli 30, 44121 Ferrara, Italy*
[c]*Department of Mathematics at Würzburg University, Emil Fischer Str. 40, Würzburg, 97074, Germany*
[d]*Max-Planck-Institut für Astrophysik, Karl-Schwarzschild-Str 1, D-85748 Garching, Germany*

Abstract

We present a new family of very high order accurate direct Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume (FV) and Discontinuous Galerkin (DG) schemes for the solution of nonlinear hyperbolic PDE systems on moving Voronoi meshes that are *regenerated* at each time step and which explicitly allow *topology changes* in time. The Voronoi tessellations are obtained from a set of generator points that move with the local fluid velocity. We employ an AREPO-type approach [1], which rapidly rebuilds a new high quality mesh exploiting the previous one, but *rearranging* the element shapes and neighbors in order to guarantee that the mesh evolution is robust even for vortex flows and for very long computational times. The old and new Voronoi elements associated to the same generator point 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 sides. We also need to incorporate some degenerate *space–time sliver elements*, which are needed in order to fill the space–time holes that arise because of the topology changes in the mesh between time t^n and time t^{n+1}. The final ALE FV-DG scheme is obtained by a novel redesign of the high order accurate fully discrete direct ALE schemes of Boscheri and Dumbser [2, 3], which have been extended here to general moving Voronoi meshes and space–time sliver elements. Our new numerical scheme is based on the integration over arbitrary shaped closed space–time control volumes combined with a fully-discrete space–time conservation formulation of the governing hyperbolic PDE system. In this way the discrete solution is *conservative* and satisfies the geometric conservation law (GCL) *by construction*. Numerical convergence studies as well as a large set of benchmark problems for hydrodynamics and magnetohydrodynamics (MHD) demonstrate the accuracy and robustness of the proposed method. Our numerical results clearly show that the new combination of very high order schemes with regenerated meshes that allow topology changes in each time step lead to substantial improvements over the existing state of the art in direct ALE methods.

Keywords: Arbitrary-Lagrangian-Eulerian (ALE) Finite Volume (FV) and Discontinuous Galerkin (DG) schemes, arbitrary high order in space and time, moving Voronoi tessellations with topology change, *a posteriori* sub-cell finite volume limiter, fully-discrete one-step ADER approach for hyperbolic PDE, compressible Euler and MHD equations

1. Introduction

The aim of this work is to present a novel family of *arbitrary high order accurate* direct ALE Finite Volume (FV) and Discontinuous Galerkin (DG) schemes on moving Voronoi meshes that are *regenerated* at each time-step and which consequently allow also *topology changes* of the computational grid during the time evolution of the PDE system. The *main novelty* lies in the use of a *space–time conservation formulation* of the governing PDE system over closed, non-overlapping *space-time* control volumes that are constructed from the moving, regenerated Voronoi meshes between time t^n and time t^{n+1}. On these closed space–time control volumes the governing equations are then

---

[*]Corresponding author
Email address: `elena.gaburro@unitn.it` (Elena Gaburro[*])

directly integrated by means of a high order fully discrete one-step ADER method. To the best knowledge of the authors, this is the first time that arbitrary high order accurate direct ALE FV and DG schemes are developed with an embedded mesh generator that builds a new mesh with a different topology at each time step.

1.1. State of the art

Lagrangian algorithms [4, 5, 6, 7, 8, 9, 10, 11, 12] are characterized by a moving computational mesh displaced with a velocity chosen as close as possible to the local fluid velocity. In the Lagrangian description of the fluid, the nonlinear convective terms disappear and, as a consequence, Lagrangian schemes exhibit virtually no numerical dissipation at contact discontinuities and material interfaces. Therefore, the aim of these methods is to *reduce the numerical dissipation errors* due to the convective terms, so that contact discontinuities are sharply captured and material interfaces can be properly identified and tracked.

Lagrangian finite volume schemes [9, 13, 14, 15, 16, 17, 18] have been developed for the solution of nonlinear hyperbolic systems of PDEs, using the conservation form of the equations based on the physically conserved quantities like mass, momentum and total energy. Higher order Lagrangian-type schemes have been introduced in [19, 20, 21], 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. Contrarily to the *cell-centered* methods listed so far, where all variables are located at the cell center of the primal mesh, *staggered* Lagrangian schemes [22, 23, 24] define the velocity at the grid vertexes and the other variables at the cell center, hence avoiding the need of a nodal solver to compute the mesh velocity of the grid nodes.

Another option for the numerical solution of hyperbolic conservation laws is given by Discontinuous Galerkin [25] and Finite Element (FE) schemes, 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 [26, 27, 28, 29], while high order FE methods applied to Lagrangian hydrodynamics and elasto-plasticity can be found in [30, 31, 32, 33, 34].

Although these schemes are widely used, a common problem that affects almost all Lagrangian methods is the severe mesh distortion or mesh tangling that happens in the presence of shear flows, which may even cause a breakdown of the computation. This is the reason which led the development of the so-called Arbitrary-Lagrangian-Eulerian (ALE) methods [14, 35, 36, 37, 38, 39, 40], 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. Cell-centered indirect ALE schemes aim at improving the mesh quality and the overall scheme robustness by performing a purely Lagrangian phase with subsequent rezoning (mesh optimization) [41, 42, 43] and remapping [44], where the numerical solution defined on the old mesh is transferred onto the new grid. To overcome the problem of mesh tangling, sliding line techniques have also been proposed [45, 46, 47], which deal with moving *nonconforming* meshes, whose element sides can slide in order to accommodate the distortion induced by shear flows. In the context of indirect ALE schemes, an interesting approach for handling the mesh motion has been introduced by the so-called Reconnection ALE (ReALE) algorithms [48, 49, 50, 51], where the rezoning phase allows for topology changes at each time step of the computation. There, moving Voronoi tessellations have been employed and the obtained numerical results demonstrate that the flow features that have been computed in the Lagrangian phase can be better preserved compared to standard indirect ALE methods.

Among the different approaches that have been presented in the literature (pure Lagrangian, indirect ALE based on rezoning and remapping, ReALE as well as a peculiar nonconforming slide line treatments), a novel family of methods has been proposed, so-called *direct* Arbitrary-Lagrangian-Eulerian (ALE) schemes. Also in the framework of direct ALE the mesh velocity can be chosen in an arbitrary way. Usually, it is chosen close to the local fluid velocity. However, the mesh quality can be optimized by a rezoning phase which takes place *before* the computation of the numerical fluxes, hence allowing the space-time control volumes to be defined for each computational cell by connecting the element configuration at the current time level t^n to the next time level t^{n+1}. Next, the mesh motion is taken into account *directly* in the numerical flux computation of the FV or DG scheme, without needing any remeshing plus remapping strategy. Furthermore, such approaches naturally extend to unstructured meshes in multiple space dimensions [52] and to slide line treatment with nonconforming meshes [53, 54]. Direct ALE schemes have been recently presented in [55, 56, 2, 3, 57] by employing either very high order FV and DG schemes, also in combination with time-accurate local time stepping (LTS), see [58, 59]. These works are characterized by a *fixed* mesh topology, which makes 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. Notice that direct ALE schemes, even when constrained to a fixed connectivity, already ameliorate standard Lagrangian results for complex flow patterns.

From what was observed so far, the idea of *allowing a change of topology* at each time step within the direct ALE framework arises. A seminal work along this direction is represented by the *AREPO* code of Springel and collaborators [1, 60, 61, 62]. *AREPO* is a massively parallel second order accurate two- and three-dimensional direct ALE finite volume scheme on moving Voronoi tessellations that are rebuilt at each time step from a set of generator points which are moving with the local fluid velocity. The documented results obtained with the *AREPO* technique clearly highlight the robustness and potential of that approach. Similar work in the context of finite element schemes can be found in the well-known particle finite element method of Oñate and Idelsohn *et al.*, see [63, 64, 65, 66, 67, 68]. In the above-mentioned references, the mesh is completely regenerated at each time step, thus naturally allowing for large deformations and strong shear flows without causing mesh tangling and highly distorted elements.

1.2. Challenges of this work

Up to now the *AREPO* algorithm [1, 60] is at most second order accurate in space and time. We therefore believe that its results can still be improved by (i) increasing the order of accuracy of the underlying FV scheme in both space and time and by (ii) introducing a higher order DG method into the AREPO framework. However, above all, the main difficulty arises from the fact that high order direct ALE schemes need a complete knowledge of the *space–time connectivity* between two consecutive time steps t^n and t^{n+1}, and not only of the *spatial* connectivity at each time level. Moreover, if a change of connectivity is allowed, the space-time connectivity does not coincide neither with the connectivity at time t^n, nor with the one at time t^{n+1}. Hence, an automatic way to construct the missing space-time connectivity from the available spatial connectivities at t^n and t^{n+1} must be found. In addition, the space–time control volumes should be allowed to have as bottom and top faces polygons with a different number of edges, and, moreover, even degenerate *space–time sliver elements* must be incorporated in order to fill the space-time holes that are caused by the changing topology. With sliver elements we refer to space–time elements whose areas at time t^n and t^{n+1} are null, but whose space–time volume is not zero, see Sections 2.5 and 2.6. In other words, sliver elements exist only in the space-time volume strictly bounded between two consecutive time levels, therefore they must be taken into account only if the numerical scheme requires the full space-time connectivity.

Finally, this kind of elements should be not only built, but also the one-step ADER finite volume and DG schemes must be substantially modified to handle the integration of the PDE over these new types of space-time control volumes. A proof of concept that direct ALE methods can work even on degenerate space-time elements was already given in [53] for second order FV schemes on moving nonconforming meshes, but a much greater effort is necessary for dealing with such a general situation as the one treated in this work.

1.3. Structure of the paper

The rest of the paper is organized as follows. In Section 2 we introduce our *moving computational mesh* and how to deal with the *topology changes* that are caused by the *regeneration* of the *Voronoi tessellation* at each time step. Then, we explain how to automatically construct the space–time connectivity and the space–time sliver elements.

Once this has been set up, in Section 3 we describe our *direct* ALE FV-DG scheme, namely an algorithm belonging to the class of direct ALE $P_N P_M$ schemes [69], which allows us to formulate a Finite Volume (FV) and a Discontinuous Galerkin (DG) scheme within a *unique* framework. The method is first presented for standard moving Voronoi elements, i.e. Voronoi elements that are displaced without modifying their shape, i.e. the number of their nodes remains the same at each time level. Then, the method is extended to Voronoi elements with different bottom and top faces and finally to sliver elements in Sections 3.1.2 and 3.2.2.

In Section 4 we show a large set of numerical result, including convergence rates up to fifth order of accuracy in space and time for smooth problems as well as a wide set of benchmark test cases solved with our ALE FV-DG scheme on regenerated Voronoi meshes for different systems of hyperbolic equations, namely the Euler equations of compressible gas dynamics, including the gravity source term, and the ideal MHD equations. The numerical results are compared with available reference solutions where possible and widely commented.

The paper is closed by some conclusive remarks and an outlook to future work in Section 5.

2. Numerical method I: handling a moving Voronoi tessellation with topology changes and data reconstruction

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

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q}), \qquad \mathbf{x} \in \Omega(t) \subset \mathbb{R}^2, \qquad \mathbf{Q} \in \Omega_{\mathbf{Q}} \subset \mathbb{R}^\nu, \tag{1}$$

where $\mathbf{x} = (x, y)$ is the spatial position vector, t represents the time, $\mathbf{Q} = (q_1, q_2, \ldots, q_\nu)$ is the vector of conserved variables defined in the space of the admissible states $\Omega_{\mathbf{Q}} \subset \mathbb{R}^\nu$, $\mathbf{F}(\mathbf{Q}) = (\mathbf{f}(\mathbf{Q}), \mathbf{g}(\mathbf{Q}))$ is the non linear flux tensor, and $\mathbf{S}(\mathbf{Q})$ represents a non linear algebraic source term.

To discretize the moving two-dimensional domain $\Omega(t)$ we employ a centroid based Voronoi-type tessellation made of N_P non overlapping polygons $P_i, i = 1, \ldots N_P$. The tessellation is firstly built at time $t = 0$ and then it is regenerated at each time step t^n. Data are represented via high order polynomials in each Voronoi polygon, which are either given by a (C)WENO reconstruction procedure for FV schemes, or directly available from the numerical solution when a DG method is considered.

2.1. Computational grid

At time $t^n = 0$ we fix the position of N_P points, called generator points: their coordinates are denoted as $\mathbf{x}_{\mathbf{c}_i}^n, i = 1, \ldots, N_P$ and they are uniformly distributed inside the rectangular domain $\Omega(0)$ as well as on its boundary. Next, we build a Delaunay triangulation having these generators $\mathbf{x}_{\mathbf{c}}^n$ as vertexes of the triangles. The defining property of the Delaunay triangulation is that the circumcircle of each triangle is not allowed to contain any of the other generator points in its interior. This *empty circumcircle property* distinguishes the Delaunay triangulation from the many other triangulations of the plane that are possible for the point set. Furthermore, this condition uniquely determines the triangulation for points in general position (except for circles with more than three generator points on them for which the Delaunay triangulation contains degenerate cases where it may flip by an infinitesimal motion of one of the points). For this step we follow the Delaunay algorithm presented in [70, 71], where the point location phase is efficiently performed by employing a *jump-and-walk* method [72].

Each generator point $\mathbf{x}_{\mathbf{c}_i}^n$ is then associated to a centroid based Voronoi element P_i^n by connecting counterclockwise the *barycenters* of all the Delaunay triangles having this generator point as a vertex. Note that the use of barycenters (instead of circumcenters) to construct these Voronoi-type elements avoids degenerate situations caused by the violation of the empty circumcircle property, thus it does not need to be resolved. We refer to Figure 1 for a graphical interpretation (generator points are always plotted in red and Voronoi vertexes in blue). In particular, given a Voronoi polygon P_i^n we denote by $\mathcal{V}(P_i^n) = \{v_{i_1}^n, \ldots, v_{i_j}^n, \ldots, v_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ Voronoi neighbors, by $\mathcal{E}(P_i^n) = \{e_{i_1}^n, \ldots, e_{i_j}^n, \ldots, e_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ edges, and by $\mathcal{D}(P_i^n) = \{d_{i_1}^n, \ldots, d_{i_j}^n, \ldots, d_{i_{N_{V_i}^n}}^n\}$ the set of its $N_{V_i}^n$ vertexes, consistently *ordered counterclockwise*. Finally, the barycenter of a Voronoi polygon P_i^n is noted as $\mathbf{x}_{\mathbf{b}_i}^n = (x_{b_i}^n, y_{b_i}^n)$ (note that usually it does not coincide with the generator point, and it is always plotted in orange). By connecting $\mathbf{x}_{\mathbf{b}_i}^n$ with each vertex of $\mathcal{D}(P_i)$ we subdivide the Voronoi polygon P_i^n in $N_{V_i}^n$ subtriangles denoted as $\mathcal{T}(P_i^n) = \{T_{i_1}^n, \ldots, T_{i_j}^n, \ldots, T_{i_{N_{V_i}^n}}^n\}$.

2.2. Spatial representation of the numerical solution

The numerical solution for the conserved quantities $\mathbf{Q}$ in (1) is represented via a cell-centered approach inside each Voronoi 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 $\mathcal{U}_h$,

$$\mathbf{u}_h^n(\mathbf{x}, t^n) = \sum_{\ell=0}^{N-1} \varphi_\ell(\mathbf{x}, t^n) \, \hat{\mathbf{u}}_{\ell,i}^n := \varphi_\ell(\mathbf{x}, t^n) \, \hat{\mathbf{u}}_{\ell,i}^n, \qquad \mathbf{x} \in P_i^n, \tag{2}$$

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
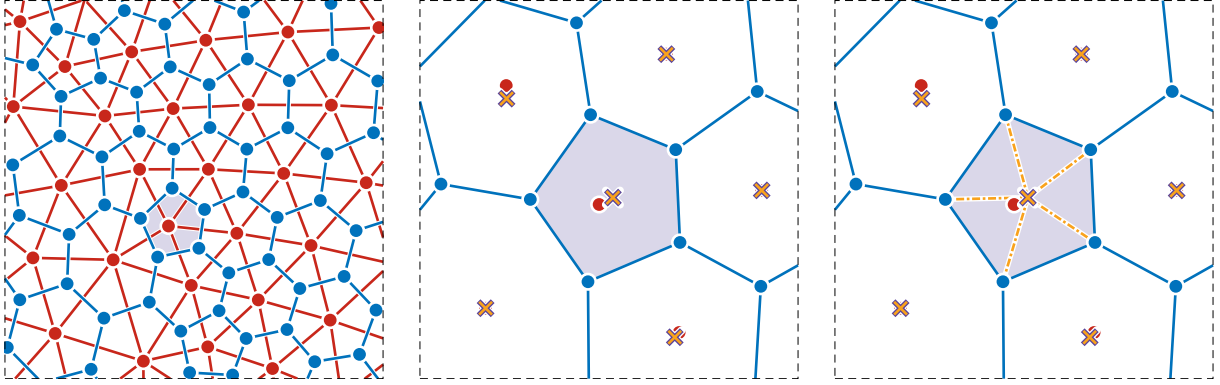
Figure 1: In these three panels we report the Delaunay triangulation and the generator points in red. The barycenters of the Delaunay triangles and the Voronoi tessellation are represented in blue. Finally, the barycenters of the Voronoi polygons are represented with orange crosses. Note that to each generator point corresponds a Voronoi polygon which is obtained by connecting the barycenters of the triangles having this generator point as a vertex. Note also that we employ its barycenter to construct the sub–triangulation of each Voronoi element (orange dotted line in the right panel).

summation over two equal indices. The total number $\mathcal{N}$ of expansion coefficients (degrees of freedom) $\hat{\mathbf{u}}_l^n$ for the basis functions depends on the polynomial degree N and is given by $\mathcal{N} = \mathcal{L}(N, d)$, with

$$\mathcal{L}(N, d) = \frac{1}{d!} \prod_{m=1}^{d} (N + m), \tag{3}$$

where $d = 2$ in this paper, since we are dealing only with two-dimensional domains. As basis functions φ_ℓ in (2) 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 $\mathbf{x}_{\mathbf{b}_i}^n$ and normalized by its current characteristic length h_i

$$\varphi_\ell(\mathbf{x}, t^n)|_{P_i^n} = \frac{(x - x_{b_i}^n)^{p_\ell}}{p_\ell! \, h_i^{p_\ell}} \frac{(y - y_{b_i}^n)^{q_\ell}}{q_\ell! \, h_i^{q_\ell}}, \qquad \ell = 0, \dots, \mathcal{N} - 1, \quad 0 \le p_\ell + q_\ell \le N, \tag{4}$$

h_i being the radius of the circumcircle of P_i^n. The unknown expansion coefficients $\hat{\mathbf{u}}_{\ell,i}^n$ in (2) are the rescaled derivatives $h_i^{p_\ell} h_i^{q_\ell} \frac{\partial^{p_\ell + q_\ell}}{\partial x^{p_\ell} \partial y^{p_\ell}} \mathbf{Q}\left(\mathbf{x}_{\mathbf{b}_i}^n\right)$ of the Taylor expansion about $\mathbf{x}_{\mathbf{b}_i}^n$. The time dependence of $\varphi(\mathbf{x}, t^n)$ derives from the time-dependence of the cell barycenter $\mathbf{x}_{\mathbf{b}_i}^n$.

The discontinuous finite element data representation (2) 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 (2) reduces to the classical piecewise constant data representation that is typical of finite volume schemes:

$$\mathbf{u}_h^n(\mathbf{x}, t^n) = 1 \cdot \hat{\mathbf{u}}_{0,i}^n, \qquad \mathbf{x} \in P_i^n, \qquad \hat{\mathbf{u}}_{0,i}^n = \frac{1}{|P_i^n|} \int_{P_i^n} \mathbf{Q}(\mathbf{x}, t^n) d\mathbf{x}. \tag{5}$$

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 (2) 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 CWENO procedure presented in 2.3) that reads

$$\mathbf{w}_h^n(\mathbf{x}, t^n) = \sum_{\ell=0}^{\mathcal{M}-1} \psi_\ell(\mathbf{x}, t^n) \, \hat{\mathbf{w}}_{\ell,i}^n := \psi_\ell(\mathbf{x}, t^n) \, \hat{\mathbf{w}}_{\ell,i}^n, \qquad \mathbf{x} \in P_i^n, \qquad \mathcal{M} = \mathcal{L}(M, d), \tag{6}$$

where we simply employ the same basis functions $\psi_l(\mathbf{x}, t^n) = \varphi_l(\mathbf{x}, t^n)$ for the reconstruction according to (4), but with $0 \le \ell \le \mathcal{M} - 1$ rather than $0 \le \ell \le \mathcal{N} - 1$, see also [69].

With this notation, our method falls within the more general class of $P_N P_M$ schemes introduced in [69] for fixed unstructured simplex meshes in two and three space dimensions. In [69, 73, 74, 75] a new family of hybrid, reconstructed discontinuous Galerkin methods was proposed, in which a Hermite-type reconstruction of degree $M \geq N$ is performed on cell data represented by polynomials of degree N. In this paper, however, we restrict ourselves to the two most common 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 $N + 1$. Within the general $P_N P_M$ formalism one can simultaneously deal with arbitrary high order FV and DG schemes inside a unified framework, with only very few differences between the two schemes.

For the sake of uniform notation, when $N > 0$ and hence $M = N$, 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.

2.3. 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. It should be employed in the case $N = 0, M > 0$. As already stated above, the total number of unknown degrees of freedom $\mathbf{w}_h^n(\mathbf{x}, t^n)$ is $\mathcal{M} = \mathcal{L}(M, d)$, with M denoting the degree of the reconstruction polynomial $\mathbf{w}_h$.

In order to achieve high accuracy, a large stencil centered in P_i^n is required, but this choice produces oscillations close to discontinuities, the well-known Gibbs phenomenon. Indeed, for linear reconstruction operators, the requirements of high order of accuracy and non-oscillatory behavior are in contrast with each other, due to the well-known Godunov theorem [76]. In order to fulfill also the requirement of non-oscillatory behavior, a nonlinear reconstruction operator has to be adopted. In this paper we rely on the CWENO reconstruction strategy first introduced in [77, 78, 79], and which can be cast in the general framework described in [80]. Here, we closely follow the work outlined in [81] for unstructured triangular and tetrahedral meshes. For the sake of completeness, we report here the entire algorithm: the differences with respect to [81] are highlighted in the last paragraph of this section.

The reconstruction starts from the computation of a so-called *central polynomial* $\mathbf{P}_{\text{opt}}$ of degree M. In order to define $\mathbf{P}_{\text{opt}}$ in a robust manner, following [81, 82, 83, 84], 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

$$\mathcal{S}_i^0 = \bigcup_{k=1}^{n_e} P_{i_k}^n, \tag{7}$$

with the safety factor $f \geq 1.5$. Stencil S_i^0 includes the current Voronoi polygon P_i^n, the first layer of Voronoi neighbors (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) [85] as

$$\mathbf{P}_{\text{opt}}(\mathbf{x}, t^n) = \underset{\mathbf{p} \in \mathcal{P}_i}{\arg\min} \sum_{P_{i_k}^n \in \mathcal{S}_i^0} \left(\hat{\mathbf{u}}_{0,i_k}^n - \frac{1}{|P_{i_k}^n|} \int_{P_{i_k}^n} \mathbf{p}(\mathbf{x}, t^n) d\mathbf{x} \right)^2, \quad \text{with} \quad \mathcal{P}_i = \left\{ \mathbf{p} \in \mathbb{P}_M : \frac{1}{|P_i^n|} \int_{P_i^n} \mathbf{p}(\mathbf{x}, t^n) d\mathbf{x} = \hat{\mathbf{u}}_{0,i}^n \right\}, \tag{8}$$

where $\mathbb{P}_M$ is the set of all polynomials of degree at most M. In other words, the polynomial $\mathbf{P}_{\text{opt}}$ has exactly the cell average $\hat{\mathbf{u}}_{0,i}^n$ on the polygon P_i^n and matches all the other cell averages of the remaining stencil elements in the least-square sense. The polynomial $\mathbf{P}_{\text{opt}}$ is expressed in terms of the basis functions (4) of degree M, hence

$$\mathbf{P}_{\text{opt}}(\mathbf{x}, t^n) = \sum_{\ell=0}^{\mathcal{M}-1} \psi_\ell(\mathbf{x}, t^n) \hat{\mathbf{p}}_{\ell,i}^n, \tag{9}$$

and the integrals appearing in (8) are computed in each Voronoi polygon $P_{i_k}^n$ by summing the contribution of each of its sub-triangles $T \in \mathcal{T}(P_{i_k}^n)$. On the sub-triangles we employ $(M + 1)^2$ quadrature points defined by the conical product of the one-dimensional Gauss-Jacobi formula, see [86].

6

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 $\mathcal{V}(P_i^n)$. An example of stencils S_i^0 and S_i^s for a polygon with $N_{V_i}^n = 5$ and $M = 2$ is reported in Figure 2.
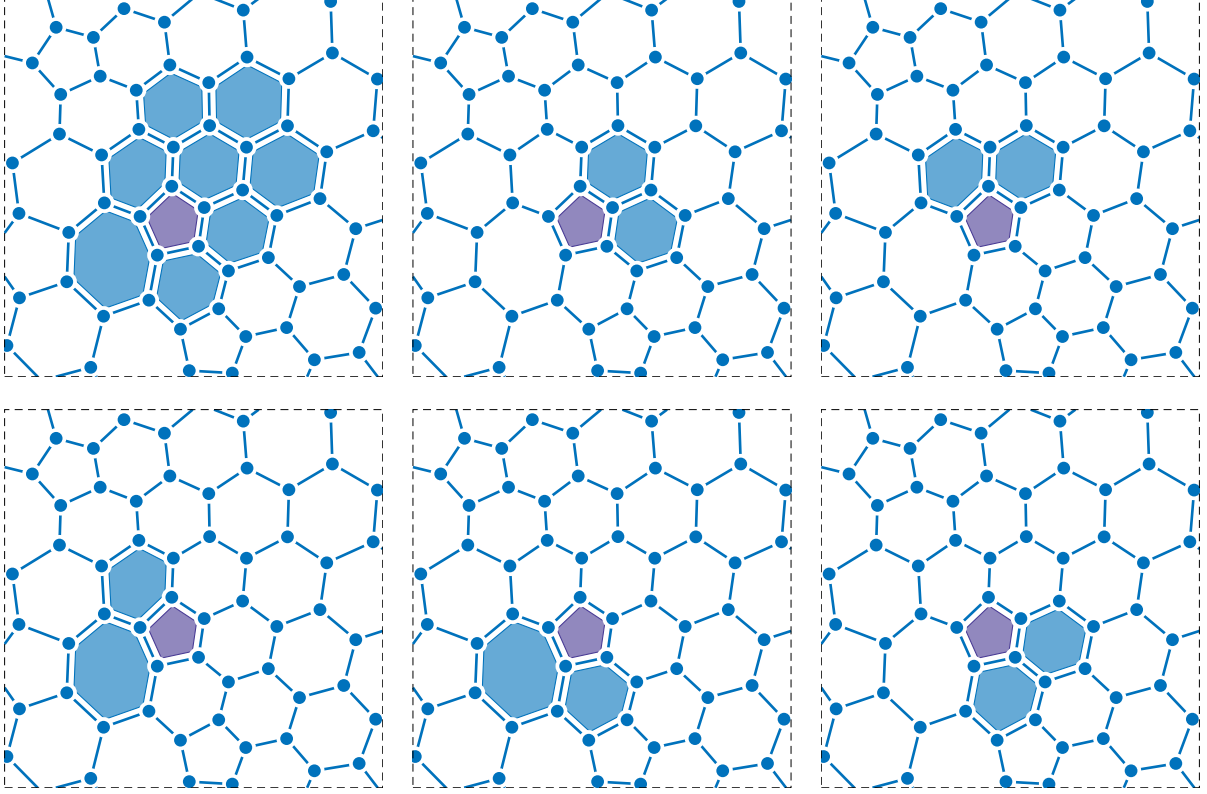

Figure 2: Stencils for the CWENO reconstruction of order three ($M = 2$) with $f = 1.5$ for a pentagonal element P_i^n. Top-left: central stencil made of the element itself P_i^n (in violet) and $n_e - 1 = 8$ of its neighbors (in blue). In the other panels we report the $N_{V_i}^n = 5$ sectorial stencils containing the element itself 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

$$\mathbf{P}_s(\mathbf{x}, t^n) \in \mathbb{P}_1 \text{ s.t. } \forall P_{i_k}^n \in S_i^s : \frac{1}{|P_{i_k}^n|} \int_{P_{i_k}^n} \mathbf{P}_s(\mathbf{x}, t^n)\, d\mathbf{x} = \hat{\mathbf{u}}_{0,i_k}^n, \tag{10}$$

which are not overdetermined and therefore have a unique solution for non-degenerate locations of the Voronoi barycenters. Following the general framework introduced in [80], we select a set of positive coefficients $\lambda_0, \ldots, \lambda_{N_p}$ such that

$$\sum_{s=0}^{N_{V_i}^n} \lambda_s = 1 \tag{11}$$

and we define a new polynomial

$$\mathbf{P}_0(\mathbf{x}, t^n) = \frac{1}{\lambda_0} \left(\mathbf{P}_{\text{opt}}(\mathbf{x}, t^n) - \sum_{s=1}^{N_p} \lambda_s \mathbf{P}_s(\mathbf{x}, t^n) \right) \in \mathbb{P}_M, \tag{12}$$

7

so that the linear combination of the polynomials $\mathbf{P}_0, \ldots, \mathbf{P}_{N_{V_i}^n}$ with the coefficients $\lambda_0, \ldots, \lambda_{N_p}$ is equal to $\mathbf{P}_{\text{opt}}$ and conservation is ensured. Specifically, we consider the linear weights used in [87], 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 (11). Finally, the sectorial polynomials $\mathbf{P}_s$ with $s \in [1, N_{V_i}^n]$ are nonlinearly hybridized with $\mathbf{P}_0$, as it is done also in other WENO schemes [88, 89, 90]. We thus obtain $\mathbf{w}_h(\mathbf{x}, t^n)$ in P_i^n as

$$\mathbf{w}_h(\mathbf{x}, t^n) = \sum_{s=0}^{N_p} \omega_s \mathbf{P}_s(\mathbf{x}, t^n), \qquad \mathbf{x} \in P_i^n, \tag{13}$$

where the normalized *nonlinear weights* ω_s are given by

$$\omega_s = \frac{\tilde{\omega}_s}{\sum\limits_{m=0}^{N_{V_i}^n} \tilde{\omega}_m}, \qquad \text{with} \qquad \tilde{\omega}_s = \frac{\lambda_s}{(\sigma_s + \varepsilon)^r}. \tag{14}$$

In the above expression the non-normalized weights $\tilde{\omega}_s$ depend on the linear weights λ_s and the oscillation indicators σ_s with the parameters $\varepsilon = 10^{-14}$ and $r = 4$ chosen according to [85]. Note that in smooth areas, $\omega_s \simeq \lambda_s$ and then $\mathbf{w}_{h_i} \simeq \mathbf{P}_{\text{opt}}$, so that we recover optimal accuracy. On the other hand, close to a discontinuity, $\mathbf{P}_0$ and some of the low degree polynomials $\mathbf{P}_s$ would be oscillatory and have high oscillation indicators, leading to $\omega_s \simeq 0$ and in these cases only lower order non-oscillatory data are employed in $\mathbf{w}_{h_i}$, guaranteeing the non-oscillatory property of the reconstruction. The oscillation indicators σ_s appearing in (14) are simply given by

$$\sigma_s = \sum_l \left(\hat{\mathbf{p}}_{l,i}^{n,s} \right)^2. \tag{15}$$

The CWENO procedure adopted in this work is similar to the one presented in [81] and it has been adapted to Voronoi polygons and their connectivity. The needed modifications concern the computation of integrals in (8), the number of *sectorial polynomials*, and the fact that basis functions are rescaled Taylor monomials referred to the physical element and not to the reference element, hence yielding a different and very simple evaluation of the oscillation indicators (15).

2.4. Evolution of the computational domain

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. We recall that if $N = M > 0$ then $\mathbf{w}_h^n = \mathbf{u}_h^n$; if instead $N = 0$ then $\mathbf{w}_h^n$ is obtained through the reconstruction procedure described in the previous Section 2.3.

By evaluating $\mathbf{w}_h^n$ at the generator points $\mathbf{x}_{\mathbf{c}}^n$, i.e. $\mathbf{w}_h^n(\mathbf{x}_{\mathbf{c}}^n, t^n)$ with (6), we recover the local fluid velocity $\mathbf{v}(\mathbf{x}_{\mathbf{c}}^n)$, that can be used to compute the new coordinates of the generator points simply as

$$\mathbf{x}_{\mathbf{c}_i}^{n+1} = \mathbf{x}_{\mathbf{c}_i}^n + \Delta t \, \mathbf{v}(\mathbf{x}_{\mathbf{c}_i}^n). \tag{16}$$

Note that in our ALE formalism, the mesh can be moved with *any* velocity, hence it is not necessary to integrate the above relation (16) with high order of accuracy. The Delaunay triangulation connecting the new coordinates of the generator points $\mathbf{x}_{\mathbf{c}}^{n+1}$ 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 to maintain a high mesh quality without distorted elements, as depicted in Figures 7 and 8, where we show a comparison between the results obtained by allowing topology changes and by imposing a fixed connectivity, respectively. However, more care is needed in order to update the solution from time t^n to t^{n+1}. In particular, to obtain a *high order direct* ALE scheme we need a complete knowledge of the space–time structure between the two time levels, i.e. we need to construct the so called space–time control volumes and their space–time connectivity. We would like to underline that up to Finite Volume schemes of order 2, one could avoid the procedure that we are going to introduce (see [1, 62]), but starting from order 3 it is essential.
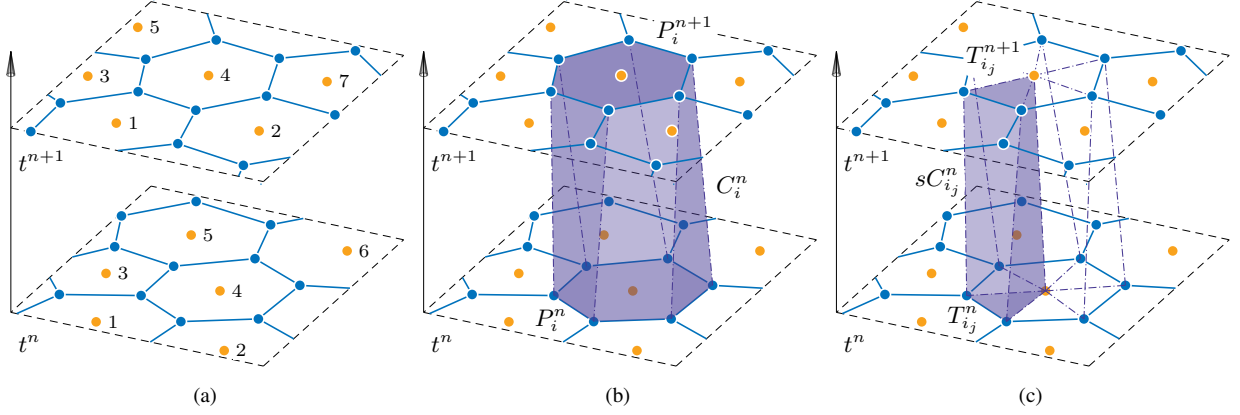
2.5. Space–time connectivity

Figure 3: 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 $T_{i_j}^n$ is connected with $T_{i_j}^{n+1}$ to construct the sub–space–time control volume $sC_{i_j}^n$.

For the sake of clarity, let us first consider the simple case in which no topology changes have occurred between t^n and t^{n+1}, i.e. $N_{V_i}^n = N_{V_i}^{n+1}$ and $\mathcal{V}(P_i^n) = \mathcal{V}(P_i^{n+1})$, as illustrated in Figure 3. Here, 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}. Moreover, each sub–triangle $T_{i_j}^n \in \mathcal{T}(P_i^n)$ is connected with the corresponding $T_{i_j}^{n+1} \in \mathcal{T}(P_i^{n+1})$ obtaining a *sub–space–time control volume*, denoted by $sC_{i_j}^n$ in the following, which has the form of an oblique prism in space–time, with triangular faces on the bottom (t^n) and the top (t^{n+1}).

We underline that each space–time element C_i^n is given by a volume that is closed by the polygon P_i^n at time t^n, the polygon P_i^{n+1} at t^{n+1} and by the lateral space-time faces $\partial C_{i_j}^n$ $j = 1, \ldots, N_{V_i}^{n,st}$ which are quadrilaterals in space–time and represent the time evolution of the edges $e_{i_j}^n \in \mathcal{E}(P_i^n)$. Here, $N_{V_i}^{n,st} = N_{V_i}^n = N_{V_i}^{n+1}$ denotes the number of space–time neighbors of C_i^n. The total surface of C_i^n is denoted with ∂C_i^n

$$\partial C_i^n = \bigcup_{j=1}^{N_{V_i}^{n,st}} \partial C_{i_j}^n \cup P_i^n \cup P_i^{n+1}. \tag{17}$$

Technical details 1. We recall that the node numbering (i.e. the numbering of the blue points in Figure 3) 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. Therefore, we loop over $\mathcal{V}(P_i^{n/n+1})$, we find the edges $e_{i_j}^{n/n+1}$ shared between $\mathcal{V}(P_i^{n/n+1})$ and $P_i^{n/n+1}$, and we put in correspondence their end points, so that the space–time control volume C_i^n can be defined. Besides, the surface obtained by connecting the end points of $e_{i_j}^n$ and $e_{i_j}^{n+1}$ is noted as $\partial C_{i_j}^n$, see Figure 6b. ∎

Let us now consider P_i^n and P_i^{n+1} in the case $N_{V_i}^n \neq N_{V_i}^{n+1}$. Now, the space–time connection between them induces the appearance of degenerate elements of two types: (i) degenerate *sub–space–time control volumes* $sC_{i_j}^n$, where either their top or bottom faces are degenerate triangles that are collapsed just to a line, see Figures 4b-4c; (ii) and also sliver space–time elements, see Figure 4d. Technical details on their construction (intended for the reader interested in reproducing the algorithm) are reported in the following paragraph. The main characteristics of this kind of elements are described in next Section 2.6.

Technical details 2. First, we order $\mathcal{V}(P_i^n)$ and $\mathcal{V}(P_i^{n+1})$ starting from the *first* common neighbor (evidences that this choice does *not* affect the results are shown in Table 3). Then, we merge the two set of neighbors to compute $\mathcal{V}(C_i^n)$ which, in this case, does not coincide neither with $\mathcal{V}(P_i^n)$ nor with $\mathcal{V}(P_i^{n+1})$. $\mathcal{V}(C_i^n)$ contains all the polygons of $\mathcal{V}(P_i^n)$

and $\mathcal{V}(P_i^{n+1})$ counted once (i.e. without multiple entries) and counterclockwise ordered respecting the order of both $\mathcal{V}(P_i^n)$ and $\mathcal{V}(P_i^{n+1})$. It represents the set of $N_{V_i}^{n,st}$ space–time neighbors of C_i^n.

Next, we have to find the node connections in order to build C_i^n, which are not obviously determined and are recovered from $\mathcal{V}(C_i^n)$. We loop on $P_{i_j} \in \mathcal{V}(C_i^n)$ and proceed as follows:

I. If P_{i_j} belongs both to $\mathcal{V}(P_i^n)$ and to $\mathcal{V}(P_i^{n+1})$, the node connection procedure falls into the previous one, and a standard $sC_{i_j}^n$ and $\partial C_{i_j}^n$ can be recovered by connecting the end points of the edges shared between $P_i^n - P_{i_j}^n$ and $P_i^{n+1} - P_{i_j}^{n+1}$. Referring to P_3^n depicted in Figure 4, we could fix as first common neighbor P_1^n because $P_1^n \in \mathcal{V}(P_3^n)$ and $P_1^n \in \mathcal{V}(P_3^{n+1})$: nodes $21 - 55$ and $22 - 56$ can be easily connected.

II. If $P_{i_j} \in \mathcal{V}(P_i^n)$ but $P_{i_j} \notin \mathcal{V}(P_i^{n+1})$, then the end points of the edge shared between $P_i^n - P_{i_j}^n$ will be connected to a unique node at time t^{n+1}, namely the *top* node which is *common* to $P_{i_{j-1}}$ and $P_{i_{j+1}}$ at time t^{n+1}. Referring to Figure 4, both nodes 22 and 23 will be connected with node 56. In this case, $\partial C_{i_j}^n$ is degenerate: it does not have a rectangular shape but a triangular one. Also $sC_{i_j}^n$ is degenerate because its top face is just given by a line connecting the barycenter of P_i^{n+1} with the *common top* node (node 56 in Figure 4).

III. If $P_{i_j} \in \mathcal{V}(P_i^{n+1})$ but $P_{i_j} \notin \mathcal{V}(P_i^n)$, then the end points of the edge shared between $P_i^{n+1} - P_{i_j}^{n+1}$ will be connected to a unique node ad time t^n, namely the *bottom* node which is *common* to $P_{i_{j-1}}$ and $P_{i_{j+1}}$ at time t^n. Referring to P_4^n shown in Figure 4, both nodes 56 and 60 will be connected with node 23. As in the previous case, $\partial C_{i_j}^n$ has a degenerate triangular shape and also $sC_{i_j}^n$ is degenerate because its bottom face is just given by a line connecting the barycenter of P_i^n with the *common bottom* node (node 23 in Figure 4). ∎

Note that when a change of topology occurs in a Voronoi polygon, the same happens to three of its neighbors and a total of four degenerate *sub*–space–time control volumes will be originated, two of type (II) and two of type (III), refer to Figures 4b-4c. Moreover, a void is left between them: to fill it and recover a fully conservative discretization, we insert a new element called *space–time sliver element*, depicted in Figure 4d, whose bottom and top faces just coincide with an edge of the tessellation at time t^n and t^{n+1}, respectively. 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 $\partial S_{i_j}^n$, $j = 1, \ldots, 4$.

Technical details 3. The nodes of a *sliver element* are given by the end points of those edges that *flip* between the two time steps and are ordered in such a way that the volume of S_i^n is positive. Let us consider case (II) in which $P_{i_j} \in \mathcal{V}(P_i^n)$ but $P_{i_j} \notin \mathcal{V}(P_i^{n+1})$: the edge between $P_i^n - P_{i_j}^n$ is taken as bottom face for the sliver. Then, we loop over the edges outgoing from the *common top* node: two of them belong to P_i^{n+1}, the third one will be taken as top face of the sliver element. If that edge connects $P_i^{n+1} \to P_{i_j}^{n+1}$ then *one* sliver element is enough to fill the space–time hole left from the topology change.

If this is not the case, as illustrated in Figures 5b-5d, *more consecutive sliver elements* will be necessary to fill the space–time holes. These consecutive sliver elements have the bottom face in common, given by the edge between $P_i^n - P_{i_j}^n$, and the top faces given respectively by the edges composing the path connecting $P_i^{n+1} \to P_{i_j}^{n+1}$. A similar procedure is employed for situations depicted in Figures 5a-5c, corresponding to case (III). We allow a maximum of three consecutive sliver elements. ∎

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 a MOOD [91, 92] procedure described in Section 3.4 will be adopted.

2.6. Degenerate sub–space–time control volumes and sliver space–time elements

The change of topology induces the appearance of degenerate elements in the space–time connectivity.

As is evident from Figures 4b-4c, some of the sub–space–time control volumes $sC_{i_j}^n$ of C_i^n, are triangular prisms with one of their top or bottom faces collapsed to just a line, and with the lateral space–time surface $\partial C_{i_j}^n$ being of triangular shape (instead of the standard quadrilateral shape). 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 $\mathbf{u}_h^n$, $\mathbf{w}_h^n$, $\mathbf{q}_h^n$).
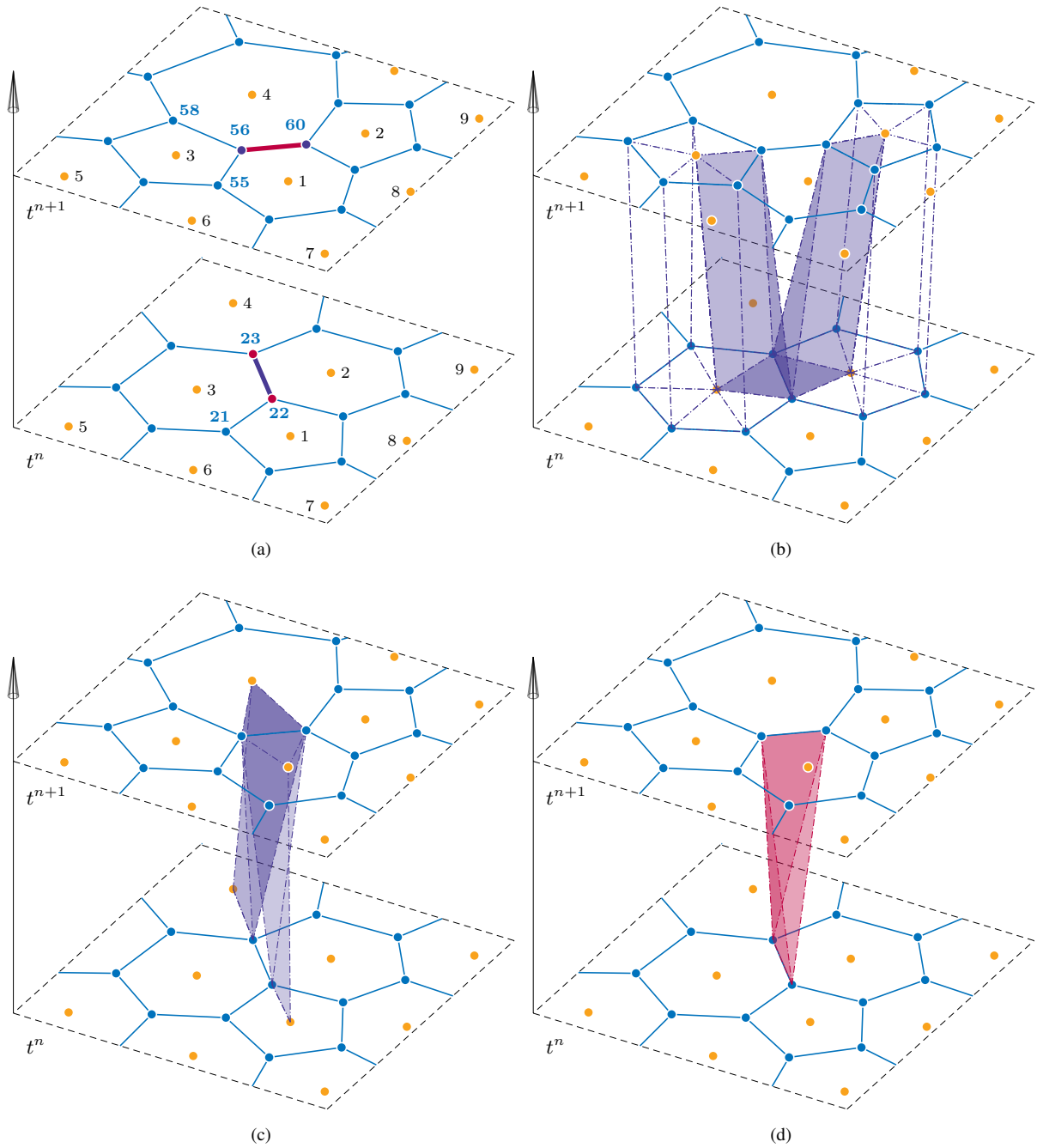
Figure 4: Space time connectivity *with* topology changes, degenerate sub–space–time control volumes and sliver element. Panel (a): at time t^n the polygons P_2^n and P_3^n are neighbors and share the highlighted edge, instead at time t^{n+1} they do not touch each other; the opposite situation occurs 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.
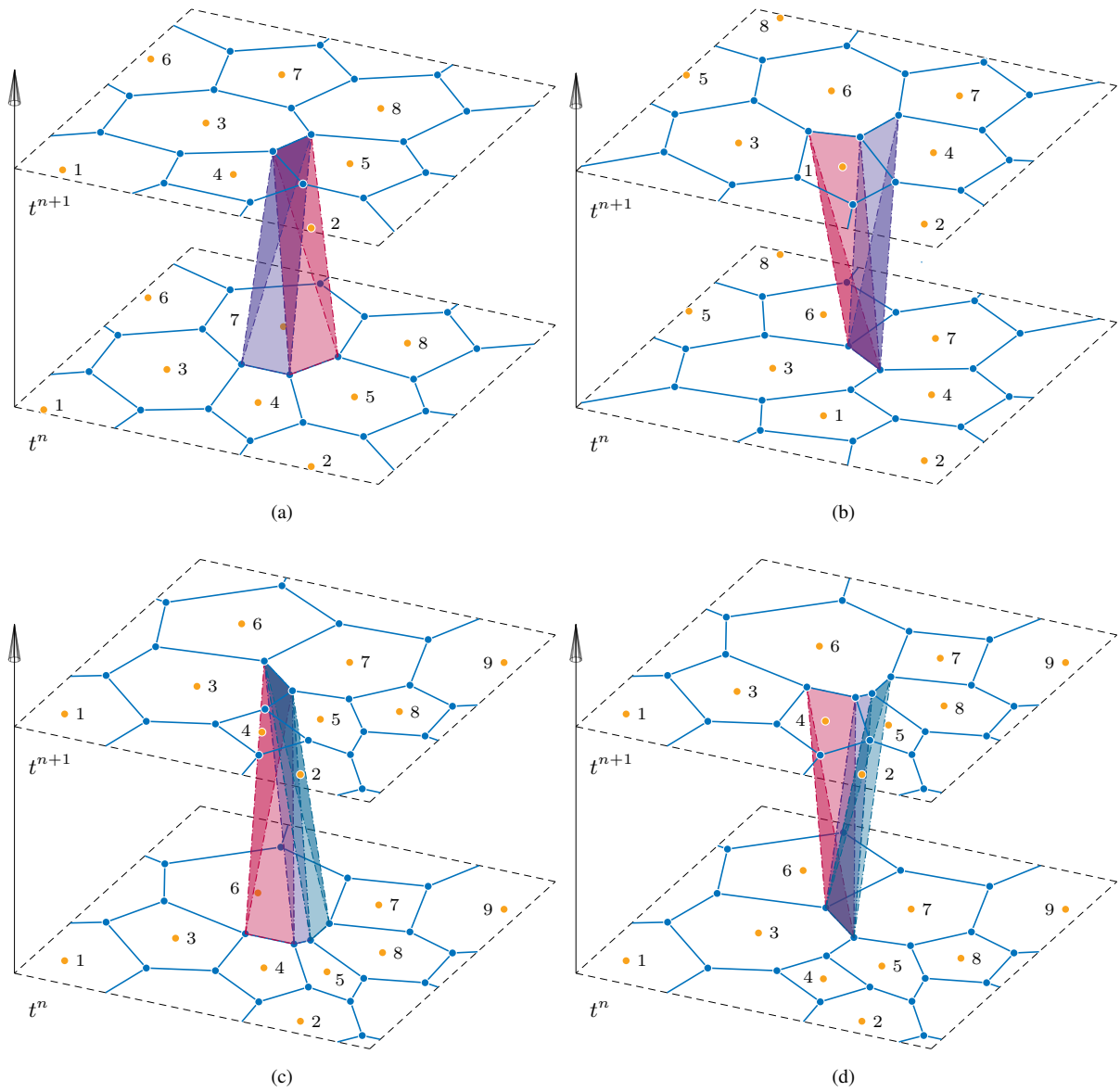
11

Figure 5: Consecutive space–time sliver elements. Refer for example to Panel (d): P_3^n and P_7^n are neighbors at time t^n but this is no longer the case at time t^{n+1} and moreover P_4^{n+1}, P_5^{n+1}, P_6^{n+1} and P_8^{n+1} are among them; this complex change of topology causes the appearance of 3 space–time sliver elements. A similar situation with 3 space–time sliver elements is depicted in Panel (c). In Panels (a) and (b) we show a change of topology with 2 space–time sliver elements.

12

On the contrary, the space–time sliver element in Figure 4d 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}. However, it has a *non-negligible volume* in space–time. The difficulties related to this kind of elements are due to the fact that $\mathbf{w}_h$ is not clearly defined for them at time t^n and that contributions across them should not be lost at time t^{n+1}, in order to ensure conservation. 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).

Note that the computation of numerical fluxes across degenerate triangular space–time faces has already been treated in [53]. In the same paper a proof of concept was given, that situations like those shown in Figures 4b-4c could be handled up to second order of accuracy. Instead, the treatment of sliver elements is a completely new topic.

3. Numerical method II: high order fully-discrete direct ALE FV-DG scheme

The governing equations (1) are now solved with the aid of a high order fully-discrete *one-step predictor-corrector* ADER FV-DG method obtained by generalizing the scheme first presented in [69] to our regenerating moving geometry. ADER finite volume schemes go back to the pioneering work of Toro and Titarev [93, 94, 95, 96, 97] on approximate solvers of the generalized Riemann problem (GPR) and have been successfully developed and applied to the Eulerian framework on fixed grids also in [98, 99] and subsequently extended to moving meshes in the ALE context [57, 2, 52, 100].

We recall that high order of accuracy in *space* is provided by the polynomial data representation $\mathbf{w}_h^n$, which for $N = M > 0$ coincides with the DG polynomial, i.e. $\mathbf{w}_h^n = \mathbf{u}_h^n$, while, in the Finite Volume case ($N = 0$), $\mathbf{w}_h^n$ is obtained through the reconstruction procedure described in Section 2.3. In any case, $\mathbf{w}_h^n$ only depends on the mesh configuration at time t^n, so that an eventual degeneracy of the space–time geometry does not affect this first step.

Then, the *predictor* step consists in a *local* solution of the governing PDE (1) *in the small*, see [101], inside each space-time element C_i^n, thus including the sliver elements S_i^n. It is called *local* because 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 of the ADER scheme.

Lastly, 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, including those across the sliver elements (see Section 3.2.2). Moreover, the method is stable if the time-step size Δt satisfies an explicit CFL stability condition, which reads

$$\Delta t < \text{CFL} \left(\frac{|P_i^n|}{(2N + 1) |\lambda_{\max,i}| \sum_{\partial P_{i_j}^n} |\ell_{i_j}|} \right), \qquad \forall P_i^n \in \Omega^n. \tag{18}$$

In the above formula, ℓ_{i_j} 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 $\mathbf{F}$. On unstructured meshes the CFL stability condition requires the inequality $\text{CFL} < \frac{1}{d}$ to be satisfied, see [69].

3.1. High order in time: space–time predictor

In what follows, a predictor of the solution is recovered, which is valid *locally* inside C_i^n and is given by high order piecewise space-time polynomials $\mathbf{q}_h^n(\mathbf{x}, t)$ of degree M that are expressed as

$$\mathbf{q}_h^n(\mathbf{x}, t) = \sum_{\ell=0}^{Q-1} \theta_\ell(\mathbf{x}, t) \hat{\mathbf{q}}_\ell^n, \qquad (\mathbf{x}, t) \in C_i^n, \qquad Q = \mathcal{L}(M, d + 1). \tag{19}$$

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

$$\theta_\ell(x, y, t)|_{C_i^n} = \frac{(x - x_{b_i}^n)^{p_\ell}}{p_\ell! \, h_i^{p_\ell}} \frac{(y - y_{b_i}^n)^{q_\ell}}{q_\ell! \, h_i^{q_\ell}} \frac{(t - t^n)^{q_\ell}}{q_\ell! \, h_i^{q_\ell}}, \qquad \ell = 0, \ldots, \mathcal{L}(M, d + 1), \quad 0 \leq p_\ell + q_\ell + r_\ell \leq M. \qquad (20)$$

The predictor $\mathbf{q}_h^n$ 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

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} \, d\mathbf{x}dt + \int_{C_i^n} \theta_k(\mathbf{x}, t) \nabla \cdot \mathbf{F}(\mathbf{q}_h^n) \, d\mathbf{x}dt = \int_{C_i^n} \theta_k(\mathbf{x}, t) \, \mathbf{S}(\mathbf{q}_h^n) \, d\mathbf{x}dt. \qquad (21)$$

Differently from what has been proposed in [69, 98, 56, 2], here we do *not* integrate the first term in (21) 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 [102, 103, 104], 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 (21) is rewritten as

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} \, d\mathbf{x}dt = \int_{C_i^n \setminus \partial C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} \, d\mathbf{x}dt + \int_{\partial C_i^n} \theta_k(\mathbf{x}, t) \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right) \tilde{\mathbf{n}}_t^- \, dS. \qquad (22)$$

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 (22), and they are easily identified by checking the sign of the time component of the space–time normal vector $\tilde{\mathbf{n}}$.

3.1.1. Space–time predictor on standard space–time elements

For standard elements, we apply the jump term only on the bottom surface P_i^n of the space–time element C_i^n under consideration, where it then simplifies to

$$\left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right) \tilde{\mathbf{n}}_t^- \Big|_{P_i^n} = - \left(\mathbf{w}_h^n(\mathbf{x}, t^n) - \mathbf{q}_h^n(\mathbf{x}, t^n) \right) = \mathbf{q}_h^n(\mathbf{x}, t^n) - \mathbf{w}_h^n(\mathbf{x}, t^n), \qquad (23)$$

with $\mathbf{q}_h^{n,+} = \mathbf{w}_h(\mathbf{x}, t^n)$ being simply given by the reconstruction polynomial at time t^n and obviously $\tilde{\mathbf{n}}^- = (0, 0, -1)$ on P_i^n and thus $\tilde{\mathbf{n}}_t^- = -1$. In this case, (22) reduces to

$$\int_{C_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} \, d\mathbf{x}dt = \int_{C_i^n \setminus P_i^n} \theta_k(\mathbf{x}, t) \frac{\partial \mathbf{q}_h^n}{\partial t} \, d\mathbf{x}dt + \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \left(\mathbf{q}_h^n(\mathbf{x}, t^n) - \mathbf{w}_h(\mathbf{x}, t^n) \right) d\mathbf{x} \qquad (24)$$

for standard space–time elements. The reason for this choice is that in this manner, all space–time predictors of the standard elements are *decoupled* from each other, since they only require the initial data $\mathbf{w}_h^n$ and no information from the neighbor elements. This will not be the case for sliver elements, for which we do not have any reconstruction polynomial available at t^n. If we considered the jump terms also on lateral surfaces of standard space–time elements, the space–time predictors would no longer be independent of each other, since our mesh is *moving* and there will be in general always a non–empty subset of ∂C_i^n with $\tilde{\mathbf{n}}_t^- < 0$. This would require a proper ordering of the execution sequence of the space–time predictors on the standard elements, but this is something we want to avoid. With the following definitions

$$\mathbf{K}_1 = \int_{C_i^n \setminus P_i^n} \theta_k \frac{\partial \theta_\ell}{\partial t} \, d\mathbf{x}dt, \quad \mathbf{K}_x = \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial x} \, d\mathbf{x}dt, \quad \mathbf{K}_y = \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial y} \, d\mathbf{x}dt,$$

$$\mathbf{M} = \int_{C_i^n} \theta_k \theta_\ell \, d\mathbf{x}dt, \quad \mathbf{F}_0 = \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \psi_\ell(\mathbf{x}, t^n) \, d\mathbf{x}, \quad \mathbf{F}_1 = \int_{P_i^n} \theta_k(\mathbf{x}, t^n) \theta_\ell(\mathbf{x}, t^n) \, d\mathbf{x}, \qquad (25)$$
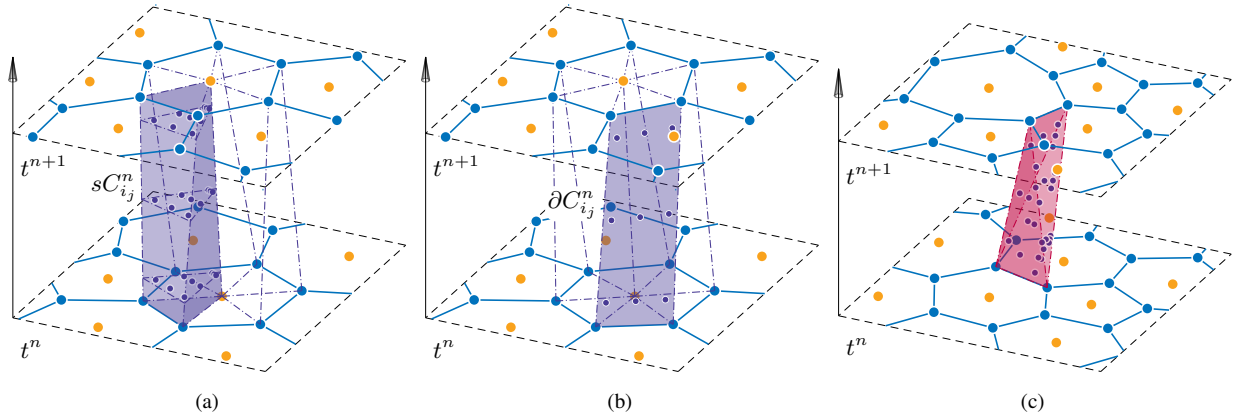
14

Figure 6: 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. (c) Quadrature points for the volume integrals and the space–time predictor for a sliver element.

the weak form (21)-(22) can be compactly rewritten as

$$(\mathbf{K}_1 + \mathbf{F}_1)\,\hat{\mathbf{q}}_i^n = \mathbf{F}_0 \hat{\mathbf{w}}_i^n - \mathbf{K}_x\,\mathbf{f}(\hat{\mathbf{q}}_i^n) - \mathbf{K}_y\,\mathbf{g}(\hat{\mathbf{q}}_i^n) + \mathbf{M}\,\mathbf{S}(\hat{\mathbf{q}}_i^n), \tag{26}$$

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 (19) and $\hat{\mathbf{w}}_{\ell,i}^n$ in (6), respectively. The solution of (26) can be found via a simple and fast converging fixed point iteration (a discrete Picard iteration), as detailed in [69, 105]. 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 [106]. 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.

Notice again that in (24) and therefore in (26) we have considered only one jump term, namely the contribution *coming from the past* through the bottom face P_i^n of C_i^n, where $\mathbf{w}_h^n = \mathbf{w}_h(\mathbf{x}, t^n)$ is *known* and *well defined*. This allows us to couple (21) with the initial condition $\mathbf{w}_h(\mathbf{x}, t^n)|_{P_i^n}$ via (24). No other information (as neighbors values) is taken into account in this local phase. Indeed, neighbor data will be considered later in the corrector step (Section 3.2).

The integrals above are evaluated using multidimensional Gaussian quadrature rules of suitable order of accuracy, see [86] and Figure 6 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. Note that for degenerate sub–space–time control volumes, as those of Figures 4b and 4c, the above quadrature formulae remain well defined, hence the predictor procedure over them does not pose any problem and does not need any adaptation.

We emphasize that we *first* carry out the space–time predictor for all standard elements, which can be computed independently of each other, and only subsequently process the remaining space–time sliver elements. The reason for this will become clear in the next section.

3.1.2. Space–time predictor on the space–time sliver elements

The predictor procedure on space–time sliver elements, as those shown in Figures 4d and 5, needs particular care. The main problem connected with the space–time sliver elements is the fact that 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 tesselation 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 (21) with some known data from the past we have to slightly modify the algorithm detailed previously.

15

In particular, the upwinding in time approach is not only used for the surface P_i^n, as done in (23), 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 ($\tilde{\mathbf{n}}_t^- < 0$). 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 evaluation of the jump term in (22). As a consequence, the predictor solution $\mathbf{q}_h^n$ is again obtained by means of (21), but treating the *entire* space–time surface ∂C_i^n with the upwind in time approach, hence leading to

$$(\mathbf{K}_1^* - \mathbf{F}_1^*)\,\hat{\mathbf{q}}_i^n = -\sum_j \mathbf{F}_j^*\,\hat{\mathbf{q}}_j^n - \mathbf{K}_x^*\,\mathbf{f}(\hat{\mathbf{q}}_i^n) - \mathbf{K}_y^*\,\mathbf{g}(\hat{\mathbf{q}}_i^n) + \mathbf{M}^*\,\mathbf{S}(\hat{\mathbf{q}}_i^n), \tag{27}$$

where the following definitions for the sliver element hold

$$\mathbf{K}_1^* = \int_{C_i^n \setminus \partial C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial t}\, d\mathbf{x}dt, \quad \mathbf{K}_x^* = \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial x}\, d\mathbf{x}dt, \quad \mathbf{K}_y^* = \int_{C_i^n} \theta_k \frac{\partial \theta_\ell}{\partial y}\, d\mathbf{x}dt,$$

$$\mathbf{M}^* = \int_{S_i^n} \theta_k \theta_\ell\, d\mathbf{x}dt, \quad \mathbf{F}_1^* = \int_{\partial C_i^n} \theta_k \theta_\ell\, \tilde{\mathbf{n}}_t^-\, dS, \quad \mathbf{F}_j^* = \int_{\partial C_{ij}^n} \theta_k \theta_\ell\, \tilde{\mathbf{n}}_t^-\, dS. \tag{28}$$

This is slightly different from what is done for standard elements in (26), 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 (27) 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 (26), 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 (27) 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.

3.2. Corrector step: direct ALE FV-DG scheme

This section contains the core of our direct ALE FV-DG scheme used to solve (1) on regenerating moving meshes. Following [56, 2, 100], the PDE system (1) is rewritten in a space-time divergence form as

$$\tilde{\nabla} \cdot \tilde{\mathbf{F}} = \mathbf{S}, \tag{29}$$

with $\tilde{\nabla} = \left(\partial_x,\, \partial_y,\, \partial_t\right)$ 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 (29) by a set of *moving* spatial modal test functions $\tilde{\varphi}_k(\mathbf{x}, t)$, which coincide with (4) 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 $\mathbf{x}_{\mathbf{b}_i}(t)$ and move together with $P_i(t)$ in such a way that at time $t = t^{n+1}$ they refer to the new barycenter $\mathbf{x}_{\mathbf{b}_i}^{n+1}$. Thus, the test functions explicitly read as follows:

$$\tilde{\varphi}_\ell(x, y, t)|_{C_i^n} = \frac{(x - x_{b_i}(t))^{p_\ell}}{p_\ell!\, h_i^{p_\ell}} \frac{(y - y_{b_i}(t))^{q_\ell}}{q_\ell!\, h_i^{q_\ell}}, \quad \text{with } \mathbf{x}_{\mathbf{b}_i}(t) = \frac{t - t^n}{\Delta t}\mathbf{x}_{\mathbf{b}_i}^n + \left(1 - \frac{t - t^n}{\Delta t}\right)\mathbf{x}_{\mathbf{b}_i}^{n+1}, \tag{30}$$

$$\ell = 0, \ldots, \mathcal{N}, \quad 0 \le p + q \le N.$$

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

$$\int_{C_i^n} \tilde{\varphi}_k \tilde{\nabla} \cdot \tilde{\mathbf{F}}(\mathbf{Q})\, d\mathbf{x}dt = \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{Q})\, d\mathbf{x}dt. \tag{31}$$

16

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

$$\int_{\partial C_i^n} \tilde{\varphi}_k \tilde{\mathbf{F}}(\mathbf{Q}) \cdot \tilde{\mathbf{n}} \, dS - \int_{C_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{Q}) \, d\mathbf{x} dt = \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{Q}) \, d\mathbf{x} dt, \tag{32}$$

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 (17).

3.2.1. Corrector step for standard space–time elements

We first describe the corrector step for standard space–time control volumes. 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

$$\tilde{\mathbf{F}}(\mathbf{Q}) \cdot \tilde{\mathbf{n}} := \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}}, \tag{33}$$

into (32), where $\mathbf{q}_h^{n,-}$ and $\mathbf{q}_h^{n,+}$ are the inner and outer boundary-extrapolated data, respectively, we obtain the final direct ALE scheme:

$$\int_{P_i^{n+1}} \tilde{\varphi}_k \mathbf{u}_h(\mathbf{x}, t^{n+1}) \, d\mathbf{x} = \int_{P_i^n} \tilde{\varphi}_k \mathbf{u}_h(\mathbf{x}, t^n) \, d\mathbf{x} - \sum_{j=1}^{N_{V_i}^{n,st}} \int_{\partial C_{ij}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS + \int_{C_i^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h) \, d\mathbf{x} dt + \int_{C_i^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h) \, d\mathbf{x} dt, \tag{34}$$

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.

Our scheme is high order accurate in space and time because the predictor solution $\mathbf{q}_h^n$, which is given by piecewise space–time polynomials of degree M, is employed for a high order accurate space-time integration of all remaining terms in (34), namely the numerical surface flux integral on ∂C_{ij}^n and the volume integrals on C_i^n for the fluxes and the source terms.

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

$$\mathbf{A_n^V}(\mathbf{Q}) = \left(\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2} \right) \left[\frac{\partial \mathbf{F}}{\partial \mathbf{Q}} \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right], \qquad \mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2}}, \tag{35}$$

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 [107] ALE scheme,

$$\mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} = \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}_h^{n,+}) + \tilde{\mathbf{F}}(\mathbf{q}_h^{n,-}) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} s_{\max} \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \tag{36}$$

where $s_{\max}$ is the maximum eigenvalue of $\mathbf{A_n^V}(\mathbf{q}_h^{n,+})$ and $\mathbf{A_n^V}(\mathbf{q}_h^{n,-})$, or a less dissipative Osher-type [108, 109] ALE flux

$$\mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} = \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}_h^{n,+}) + \tilde{\mathbf{F}}(\mathbf{q}_h^{n,-}) \right) \cdot \tilde{\mathbf{n}}_{ij} - \frac{1}{2} \left(\int_0^1 \left| \mathbf{A_n^V}(\mathbf{\Psi}(s)) \right| ds \right) \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \tag{37}$$

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

$$\mathbf{\Psi}(s) = \mathbf{q}_h^{n,-} + s \left(\mathbf{q}_h^{n,+} - \mathbf{q}_h^{n,-} \right), \qquad 0 \le s \le 1. \tag{38}$$

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

17

Finally, using the definitions (2) and (6), our arbitrary high order one-step direct ALE FV-DG scheme becomes

$$\left(\int_{P_i^{n+1}} \tilde\varphi_k \varphi_\ell \, d\mathbf{x}\right) \hat{\mathbf{u}}_\ell^{n+1} = \left(\int_{P_i^n} \tilde\varphi_k \psi_\ell \, d\mathbf{x}\right) \hat{\mathbf{w}}_\ell^n - \sum_{j=1}^{N_{V_i}^{n,st}} \int_{\partial C_{i_j}^n} \tilde\varphi_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS + \int_{C_i^n} \tilde\nabla \tilde\varphi_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) \, d\mathbf{x}dt + \int_{C_i^n} \tilde\varphi_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt. \quad (39)$$

The volume integrals in the above expression (39) can be easily computed directly on the physical space-time element C_i^n by summing up the contributions on each sub-volume $sC_{i_j}^n$ and employing Gaussian quadrature rules of sufficient precision, see [86]. The lateral space–time surfaces of ∂C_{ij}^n instead are parameterized using a set of bilinear basis functions [56], that is

$$\partial C_{ij}^n = \tilde{\mathbf{x}}(\chi, \tau) = \sum_{k=1}^{4} \beta_k(\chi, \tau) \, \tilde{\mathbf{X}}_{ij,k}^n, \qquad 0 \le \chi \le 1, \quad 0 \le \tau \le 1, \quad (40)$$

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

$$\beta_1(\chi, \tau) = (1-\chi)(1-\tau), \qquad \beta_2(\chi, \tau) = \chi(1-\tau), \qquad \beta_3(\chi, \tau) = \chi\tau, \qquad \beta_4(\chi, \tau) = (1-\chi)\tau. \quad (41)$$

The mapping in time is given by the transformation

$$t = t_n + \tau \Delta t, \qquad \tau = \frac{t - t^n}{\Delta t}. \quad (42)$$

In this way, every ∂C_{ij}^n (even if degenerate, i.e. with a triangular shape) can be mapped to a reference square $[0,1] \times [0,1]$ and surface integrals can be computed.

We close this section remarking that the integration of the governing PDE over the space-time volume C_i^n automatically satisfies the geometric conservation law (GCL) for all test functions $\tilde\varphi_k$. This simply follows from Gauss theorem applied to *closed* space–time control volumes and we refer to [2] for a complete proof.

3.2.2. Corrector step on sliver elements

Let us now consider the numerical scheme given by (39) in the case of a sliver element $C_i^n = S_i^n$:

$$0_\ell \hat{\mathbf{u}}_\ell^{n+1} = 0_\ell \hat{\mathbf{w}}_\ell^n - \sum_{j=1}^{4} \int_{\partial S_{ij}^n} \tilde\varphi_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS + \int_{S_i^n} \tilde\nabla \tilde\varphi_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) \, d\mathbf{x}dt + \int_{S_i^n} \tilde\varphi_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt, \quad (43)$$

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 (43) are in general not equal to zero, i.e.

$$-\sum_{j=1}^{4} \int_{\partial S_{ij}^n} \tilde\varphi_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS + \int_{S_i^n} \tilde\nabla \tilde\varphi_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) \, d\mathbf{x}dt + \int_{S_i^n} \tilde\varphi_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt \ne \mathbf{0}. \quad (44)$$

We underline that computing these quantities does not pose any problem, since $\mathbf{q}_h^n$ on S_i^n is well defined (refer to Section 3.1.2), 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.

The problem here arises from the fact that, using (43), the non-null quantity (44) 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 (43) is then referred

to the barycenter of C_{big}. Conservation is guaranteed by adding the contribution (44) of the sliver element S_i^n to the neighbor C_{big}, hence

$$\left(\int_{P_{\text{big}}^{n+1}} \tilde{\varphi}_k \varphi_\ell \, d\mathbf{x}\right) \hat{\mathbf{u}}_\ell^{n+1} = \left(\int_{P_{\text{big}}^n} \tilde{\varphi}_k \psi_\ell \, d\mathbf{x}\right) \hat{\mathbf{w}}_\ell^n - \sum_{j=1}^{N_{C_{\text{big}}}^{n,st}} \int_{\partial C_{\text{big}_j}^n} \tilde{\varphi}_k \mathcal{F}(\mathbf{q}_h^{n,-}, \mathbf{q}_h^{n,+}) \cdot \tilde{\mathbf{n}} \, dS + \int_{C_{\text{big}}^n} \tilde{\nabla} \tilde{\varphi}_k \cdot \tilde{\mathbf{F}}(\mathbf{q}_h^n) \, d\mathbf{x}dt + \int_{C_{\text{big}}^n} \tilde{\varphi}_k \mathbf{S}(\mathbf{q}_h^n) d\mathbf{x}dt$$

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

(45)

We would like to remark that sliver elements *only* exist in between two consecutive time levels and are degenerate both at t^n and t^{n+1}, hence they introduce some complexity in the algorithm. In particular, 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 outlined in Sections 3.1.2 and 3.2.2, 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.

Furthermore, 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 [110], 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.

3.3. A posteriori sub–cell finite volume limiter

Up to now, the presented $P_N P_M$ scheme is high order accurate in space and time and, formally, the differences between the FV case ($N = 0$) and the DG case ($N = M$) are only due to the procedure for achieving high order of accuracy in space, which is obtained through a CWENO reconstruction in the FV case and is instead automatic for DG. But there is actually one major difference, because the CWENO operator provides 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 [76], 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 [111, 112, 113], which has already been successfully applied in the framework of ADER finite volume schemes [114, 115, 92]. Specifically, the numerical solution is checked *a posteriori* for nonphysical values and spurious oscillations and, instead of applying a limiter to the already computed solution, the solution is *locally recomputed* with a more robust scheme in the so-called *troubled cells*. Troubled elements are those that do not pass the admissibility detection criteria, given by both physical and numerical requirements which mark the numerical solution as *acceptable* or *not acceptable*. If the solution in a cell is discarded, it is recomputed relying on a first order finite volume method applied to a fine sub-grid generated within each troubled cell. A second order TVD scheme has been used as limiter in [116, 3, 117], while higher order ADER-WENO subcell finite volume limiters are presented in [118, 119, 120, 121, 122].

We refer to the aforementioned references 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 Voronoi elements and topology changes.

Firstly, using the notation adopted in [3], 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.

19

- 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 $sC_{i_j}^n$, as defined in Section 2.5.

- If a topology change happens with $N = 1$, i.e. $\mathcal{V}(P_i^n) \neq \mathcal{V}(P_i^{n+1})$, degenerate small sub-triangles/sub-volumes are considered as well, thus including also sub-triangles which can be given by a line.

- 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.

- Even in the case $N \geq 2$, degenerate sub-triangles/sub-volumes are counted if a topology change happens, i.e. $\mathcal{V}(P_i^n) \neq \mathcal{V}(P_i^{n+1})$. This results in small sub-triangles which may be given by a portion of a line.

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

$$\mathbf{v}_{i,\alpha}^n(\mathbf{x}, t^n) = \frac{1}{|s_{i,\alpha}^n|} \int_{s_{i,\alpha}^n} \mathbf{u}_h^n(\mathbf{x}, t^n)\, d\mathbf{x} = \frac{1}{|s_{i,\alpha}^n|} \int_{s_{i,\alpha}^n} \varphi_\ell(\mathbf{x})\, d\mathbf{x}\, \hat{\mathbf{u}}_l^n := \mathcal{P}(\mathbf{u}_h^n) \qquad \forall \alpha \in [1, \mathcal{S}_i], \tag{46}$$

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 [3], 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

$$\min_{m \in \mathcal{V}(C_i^n)} \left(\min_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) - \delta \leq \mathbf{v}_{i,\alpha}^{n+1,*} \leq \max_{m \in \mathcal{V}(C_i^n)} \left(\max_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) + \delta \qquad \forall \alpha \in [1, \mathcal{S}_i], \tag{47}$$

where δ is a parameter which, according to [3, 118, 119], reads

$$\delta = \max \left(\delta_0\,, \ \varepsilon \cdot \left[\max_{m \in \mathcal{V}(C_i^n)} \left(\max_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) - \min_{m \in \mathcal{V}(C_i^n)} \left(\min_{\beta \in [1, \mathcal{S}_m]} (\mathbf{v}_{m,\beta}^n) \right) \right] \right), \tag{48}$$

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

If a cell fulfills the detection criteria in *all* its subcells, then the cell is marked as *good*, otherwise the cell is *troubled*. We emphasize 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.

The following step consists in re-computing 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 very well-known first order ALE-FV scheme, but we add some remarks on flux computation at the space–time lateral surfaces of each s_i^n. i) The same numerical flux function, i.e. (36) or (37), used in the rest of the scheme is adopted here as well. ii) The employed quadrature rule is a simple mid-point rule that makes use of the space–time barycenters g_i^n of the space–time lateral faces of the sub-volume. iii) The normal vectors are also computed at g_i^n. iv) Referring to (33), 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 $\mathbf{q}_h^{n,+} = \mathbf{v}_{i/i_j,\beta}^n$. v) If instead the neighbor is *not* a troubled Voronoi element $C_{i_j}^n$ (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 Voronoi 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

20

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 Voronoi 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

$$\int_{S_{i,\alpha}^n} \mathbf{u}_h^{n+1}(\mathbf{x}, t^{n+1})\, d\mathbf{x} = \int_{S_{i,\alpha}^n} \mathbf{v}_{i,\alpha}^{n+1}(\mathbf{x}, t^n)\, d\mathbf{x} := \mathcal{R}(\mathbf{v}_{i,\alpha}^{n+1}(\mathbf{x}, t^n)) \qquad \forall \alpha \in [1, \mathcal{S}_i]. \tag{49}$$

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

$$\int_{P_i^n} \mathbf{u}_h(\mathbf{x}, t^{n+1})\, d\mathbf{x} = \int_{P_i^n} \mathbf{v}_h(\mathbf{x}, t^{n+1})\, d\mathbf{x}. \tag{50}$$

As a consequence, the projection operator $\mathcal{P}$ in (46) and the reconstruction operator $\mathcal{R}$ in (49) 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.

Neighborhood of a sliver element.

At the subcell level, the difficulties associated with degenerate small sub-volumes are the same stated at the end of Section 3.2.2 for degenerate big elements: how to impose an initial condition for cells with zero area at t^n and how not to lose any contribution computed through elements with zero area at t^{n+1}. In order to activate and apply the limiter, the following strategy is proposed.

Firstly, the sliver elements are *not* sub-triangulated. If one neighbor of a sliver S_i^n is troubled, we mark as troubled also the remaining three neighbors. Among the four neighbors of S_i^n, we select the one with the biggest volume which we call C_{big}^n.

Next, we need to provide the values $\mathbf{q}_h^{+,-}$ when computing the fluxes (33).

- For a degenerate $s_{i,\alpha}^n$ with zero area at t^n we take the value obtained by evaluating $\mathbf{u}_h^n$ at the mid point of $s_{i,\alpha}^n|_{t^n}$ (this value is well defined because $s_{i,\alpha}^n \subset P_i^n$ and so $\mathbf{u}_h^n$ is continuous).

- For a sliver element S_i^n we take the value obtained by evaluating $\mathbf{u}_h^n$ of C_{big} at the mid point of $S_i^n|_{t^n}$; this arbitrary choice is justified by the fact that here we simply employ a first order method, which is stable even if the sliver elements are neglected (see [1]).

Finally, we need to redistribute the fluxes computed across the degenerate elements when they disappear at t^{n+1}.

- For a degenerate $s_{i,\alpha}^n$ with zero area at t^{n+1} we assign the sum of the fluxes computed through its space–time lateral surfaces to the closest $s_{i,\beta}^n$ that is not degenerate at t^{n+1} (the concept of *closest* is uniquely fixed through a specific numbering of the sub-volumes).

- For a sliver element S_i^n we assign its fluxes to C_{big}^n.

Besides, we remark that the space–time geometry definition in itself does not pose any problem: indeed, the configuration of big elements has already been fixed in Section 2.5 and the subdivision has been deduced just above. Therefore, quadrature formulae, normal vectors and bilinear mapping are always well defined.

3.4. MOOD approach to verify the space-time connectivity consistency

As already stated at the end of Section 2.5, it may not always be possible to connect two consecutive meshes in a consistent way if the associated topology changes are too strong. However, these situations are immediately detected at the beginning of the new time step, when the space–time connectivity is built. Indeed, if i) the set $\mathcal{V}(C_i^n)$ cannot be ordered consistently with both the order of $\mathcal{V}(P_i^n)$ and $\mathcal{V}(P_i^{n+1})$, or if ii) more than three sliver elements are necessary to complete a path between elements which are neighbors at one time level but not at the previous or at the next one, then the algorithm detects the problem. To overcome it, the current time step is simply restarted 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). However, future work will consider the possibility of remeshing only *locally*, in the neighborhood of a connectivity problem without reducing the time step size. We underline that such problems are encountered very rarely.

4. Numerical results

The numerical results presented in this section will demonstrate the following properties of our algorithms.

i. Our method naturally leads to *multi-physics* applications, namely it is designed in such a way that any kind of hyperbolic system cast in the general form (1) can be readily studied: for this reason we test it on several models, namely the standard Euler equations of gas dynamics (Section 4.1), the Euler equations with gravity source term (Section 4.2) and the magnetohydrodynamics (MHD) system (Section 4.3).

ii. Next, we show 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, thanks to its high *robustness and adaptability* to complex flow patterns, see Sections 4.1.1 and 4.3.1. We would like to underline that in this work we focus on the quality of the mesh evolution in space–time in the sense of avoiding *mesh tangling* or persistent small elements, without taking care of having an exceptional mesh quality in all time steps. Indeed, no optimization procedures, as for examples Lloyd-type algorithms or rounder cells, have been applied in our algorithm, and the initial discretization is never symmetric nor adapted to the initial flow condition; even without these measures, the method achieves results beyond the current state of the art.

iii. Then, we compute numerically the *order of convergence* of both Finite Volume and Discontinuous Galerkin schemes for two different test problems, see Tables 1, 2, 4 and 5.

iv. Finally, we study some more complicated test problems (see Sections 4.1.2, 4.1.3 and 4.3.2) to show 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 Figures 9 and 11).

The great variety of the presented tests is intended to show both the wide range of applicability of the proposed high order ALE scheme on regenerating Voronoi meshes and its level of novelty with respect to the state of the art. 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.

4.1. Euler equations of gasdynamics

A well-known example of a hyperbolic system of the form (1) is given by the homogeneous Euler equations of compressible gas dynamics with

$$\mathbf{Q} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} \rho u & \rho v \\ \rho u^2 + p & \rho uv \\ \rho uv & \rho v^2 + p \\ \rho uw & \rho vw \\ u(\rho E + p) & v(\rho E + p) \end{pmatrix}, \quad \mathbf{S} = 0. \tag{51}$$

The vector of conserved variables $\mathbf{Q}$ involves the fluid density ρ, the momentum density vector $\rho\mathbf{v} = (\rho u, \rho v)$ and the total energy density ρE. The fluid pressure p is related to conservative quantities $\mathbf{Q}$ using the equation of state for an ideal gas

$$p = (\gamma - 1)\left(\rho E - \frac{1}{2}\rho\mathbf{v}^2\right), \tag{52}$$

where γ is the ratio of specific heats so that the speed of sound takes the form $c = \sqrt{\frac{\gamma p}{\rho}}$. Where not otherwise specified we employ the Rusanov-type ALE flux (36) as numerical flux function and we move the generator points using the local fluid velocity obtained from $\mathbf{w}_h^n$ (see Section 2.4). Furthermore, we set $\gamma = 1.4$.

4.1.1. Isentropic vortex
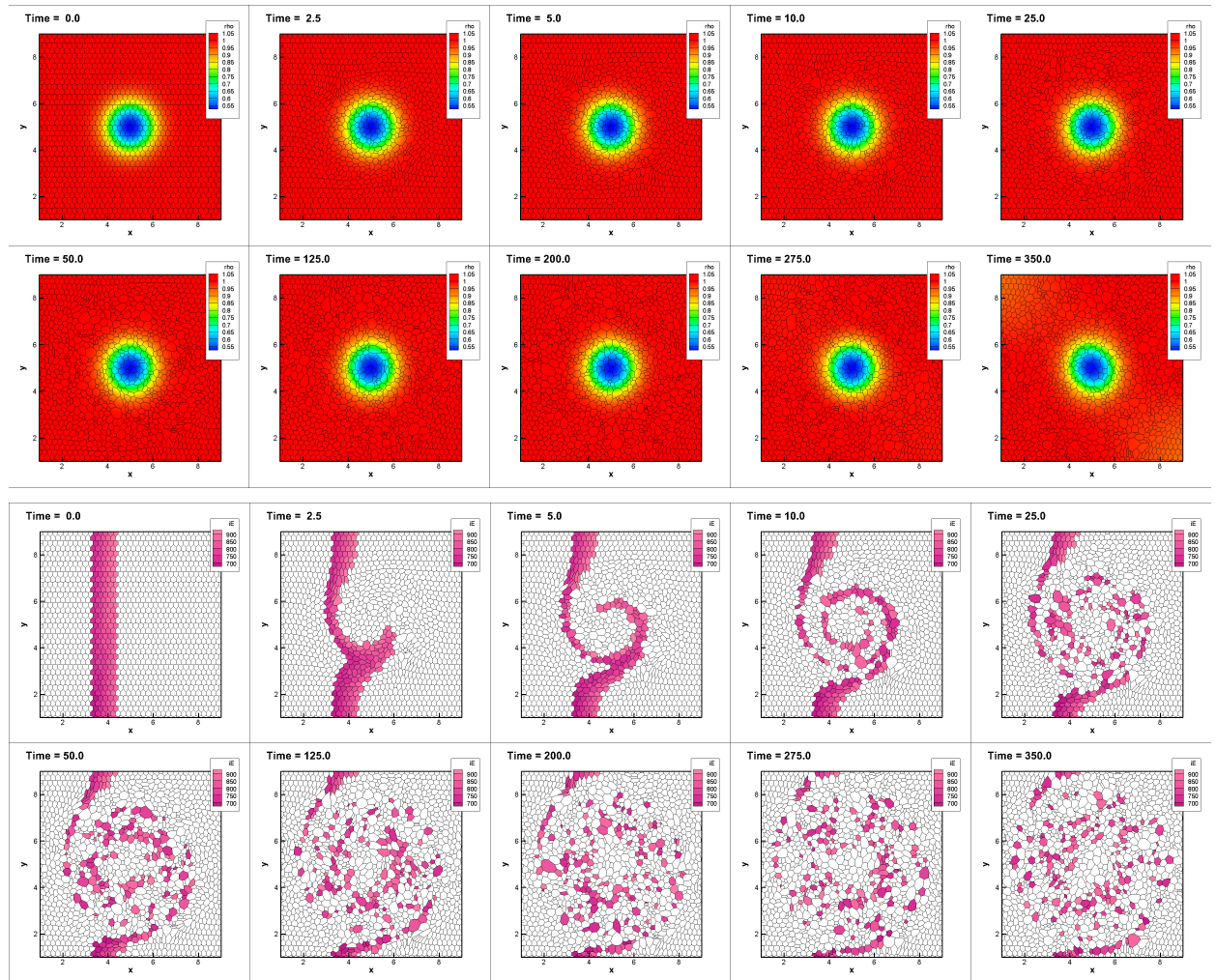
Figure 7: Stationary rotating vortex solved with our fourth order P_3P_3 ALE-DG scheme on a moving Voronoi mesh of 2116 elements with dynamical change of connectivity. Density contours (top) and the position of a bunch of highlighted elements (bottom) are provided at different times. The mesh is regenerated 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.

To verify the order of convergence of the proposed ALE FV-DG scheme we consider a smooth isentropic vortex flow according to [123]. The initial computational domain is the square $\Omega = [0; 10] \times [0; 10]$ with wall boundary
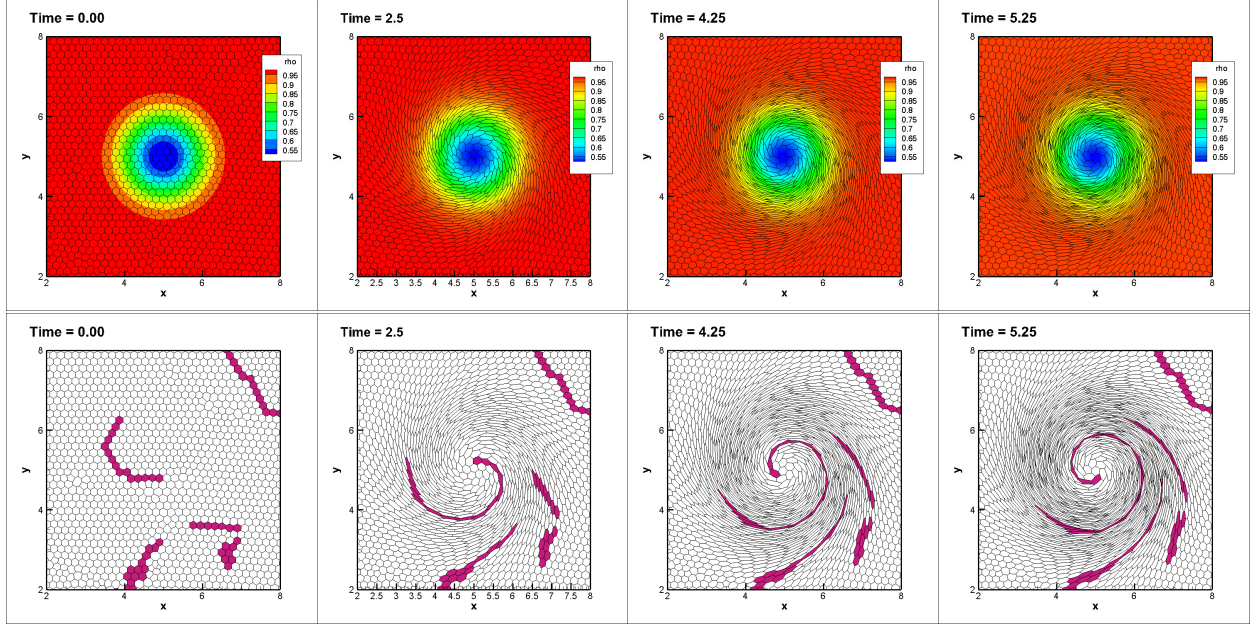
Figure 8: Stationary rotating vortex solved with a fourth order P_3P_3 ALE-DG scheme on a moving Voronoi mesh of 2116 elements with *fixed* connectivity. Density contours (top) and position of a bunch of highlighted elements (bottom) are provided at different times. The mesh quality is deteriorating already at time $t \simeq 4$ and the simulation ultimately stops at $t \simeq 5.25$ due to tangling elements.

$P_0P_1 \to O2$			$P_0P_2 \to O3$			$P_0P_3 \to O4$			$P_0P_4 \to O5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$
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 1: 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.

$P_1P_1 \to O2$			$P_2P_2 \to O3$			$P_3P_3 \to O4$			$P_4P_4 \to O5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$
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 2: 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.

conditions set everywhere. The initial condition is given by some perturbations δ that are superimposed onto a homogeneous background field $\mathbf{Q}_0 = (\rho, u, v, w, p) = (1, 0, 0, 0, 1)$, assuming that the entropy perturbation is zero, i.e. $\delta S = 0$. The perturbations for density and pressure are

$$\delta\rho = (1 + \delta T)^{\frac{1}{\gamma-1}} - 1, \quad \delta p = (1 + \delta T)^{\frac{\gamma}{\gamma-1}} - 1, \tag{53}$$

24

ordering from 1st common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$O(L_2)$
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 2nd common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$O(L_2)$
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 3rd common neighbor		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_2}$	$O(L_2)$
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

Table 3: 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.

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

$$\begin{pmatrix} \delta u \\ \delta v \\ \delta w \end{pmatrix} = \frac{\varepsilon}{2\pi}e^{\frac{1-r^2}{2}} \begin{pmatrix} -(y-5) \\ (x-5) \\ 0 \end{pmatrix}. \tag{54}$$

This is a stationary equilibrium of the system so the exact solution coincides with the initial condition at any time.

Convergence. Tables 1 and 2 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 circumcircle 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 7 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 8). 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.
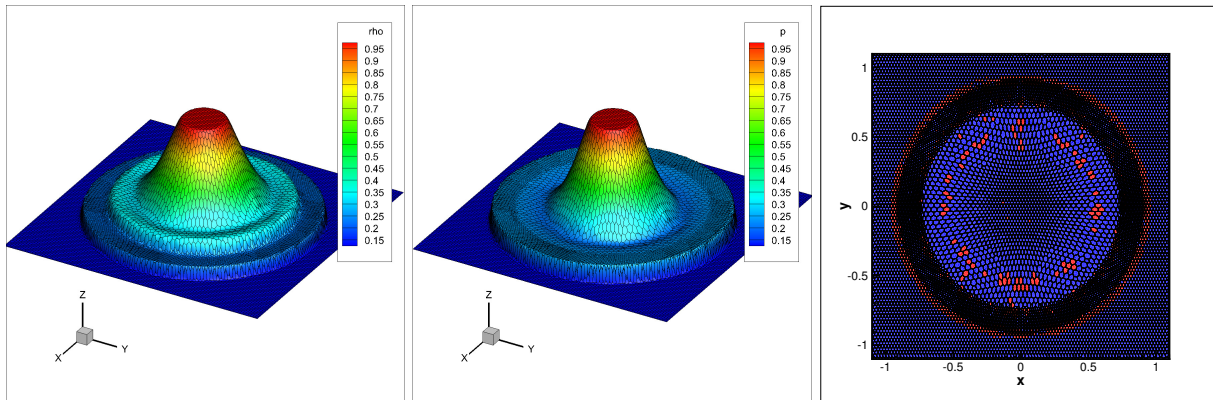
Figure 9: Explosion problem solved with our P_2P_2 method on a moving Voronoi mesh of 10201 elements. We show the density profile (left), the pressure profile (center) and we color in red the cells on which the limiter is active (right) at time $t = 0.25$. We underline that the limiter activates in particular on the external cylindrical shock wave.

The second row of Figure 7 shows the position of a bunch of highlighted elements 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 8).

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 (see Section 2.5), 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 3 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.

4.1.2. Explosion problem

The explosion problems can be seen as a multidimensional extension of the classical Sod test case. Here, we consider as computational domain a square of dimension $[-1.1; 1.1] \times [-1.1; 1.1]$, and the initial condition is composed of two different states, separated by a discontinuity at radius $r_d = 0.5$

$$\begin{cases} \rho_L = 1, & \mathbf{u}_L = 0, & p_L = 1, & \|\mathbf{x}\| \leq r_d \\ \rho_R = 0.125, & \mathbf{u}_R = 0, & p_R = 0.1, & \|\mathbf{x}\| > r_d. \end{cases} \tag{55}$$

The final time is chosen to be $t_f = 0.25$, so that the shock wave does not cross the external boundary of the domain, where a transmissive boundary condition is set. We run this problem with two different configurations.

(a) In the first case we use a third order P_2P_2 DG scheme on a mesh of 10201 Voronoi elements. The results are depicted in Figure 9. In particular, one can notice that the limiter activates in proximity of the shock waves where it is indeed essential, and only on a handful of other elements.

(b) Then, we test our FV algorithm by employing a fourth order P_0P_3 scheme on a finer mesh of 22801 Voronoi elements.
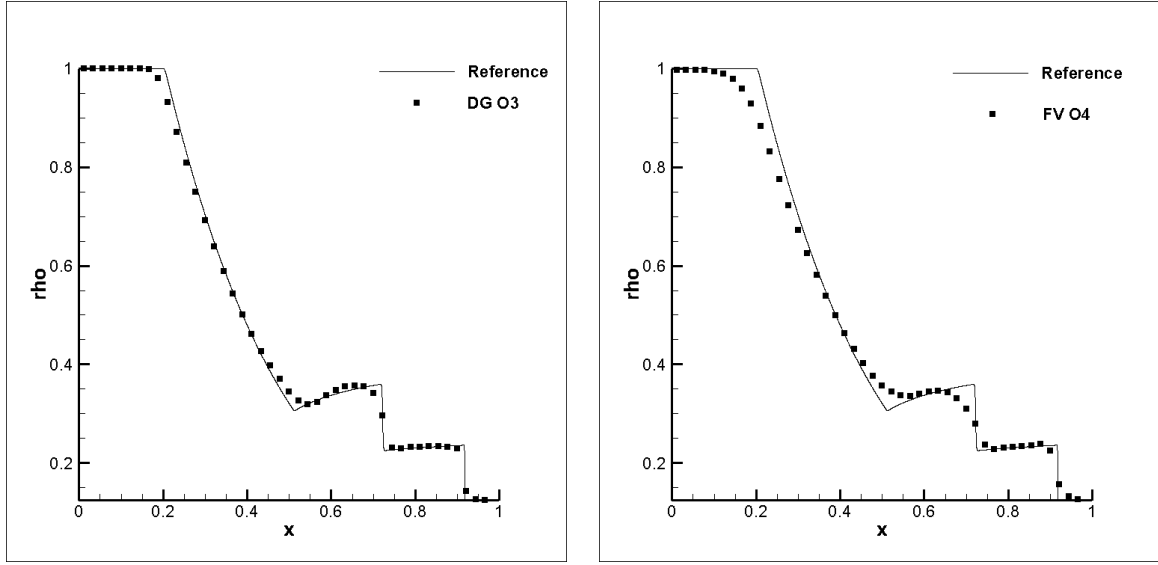
26

Figure 10: Explosion problem: we compare our numerical results (squares) with the reference solutions (line) at time $t_f = 0.25$. Left: results obtained with our P_2P_2 DG scheme on a moving Voronoi mesh of 10201 elements. Right: results obtained with our P_0P_3 FV scheme on a moving Voronoi mesh of 22801 elements. The represented values (squares) are obtained from a cut of our numerical solutions along $y = 0$.

In both cases, we can observe a good agreement between the numerical results and the reference solution. The non perfect symmetry is justified by the non symmetric initial meshes.

As in [2, 124], a reference solution can be obtained by making use of the rotational symmetry of the problem and by solving a reduced one-dimensional system with geometric source terms using a classical second order TVD scheme on a very fine one-dimensional mesh. The comparison between our numerical solutions and the reference solution is given in Figure 10. In order to obtain a similar resolution, the FV scheme needs one order more of accuracy w.r.t. the DG scheme and a finer mesh as well. We would like to underline that this test problem involves three different waves, therefore it allows each ingredient of our scheme to be properly checked. Indeed, we have

- one cylindrical shock wave that is running towards the external boundary: our scheme does not exhibit spurious oscillations thanks to the CWENO reconstruction, in the case (b), and to the *a posteriori* sub–cell finite volume limiter, in case (a);

- a rarefaction fan traveling in the opposite direction, which is well captured thanks to the high order of accuracy;

- an outward-moving contact wave in between, which is not dissipated thanks to the Lagrangian framework of our scheme, in which the mesh moves together with the fluid flow.

4.1.3. Sedov problem

This test problem is widespread in the literature [23] 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 $\Omega(0) = [0; 1.3] \times [0; 1.3]$. An exact solution based on self-similarity arguments is available from [125] 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 $p_0 = 10^{-6}$ everywhere except in the cell V_{or} containing the origin $\mathbf{O}$ where it is given by

$$p_{or} = (\gamma - 1)\rho_0 \frac{E_{tot}}{|V_{or}|}, \quad \text{with } E_{tot} = 0.979264, \tag{56}$$

being E_{tot} the total energy concentrated at $\mathbf{x} = \mathbf{0}$. We solve this numerical test with a second order P_1P_1 DG scheme on a mesh of 6399 Voronoi elements. The density profiles are shown in Figure 11 for various output times $t =$
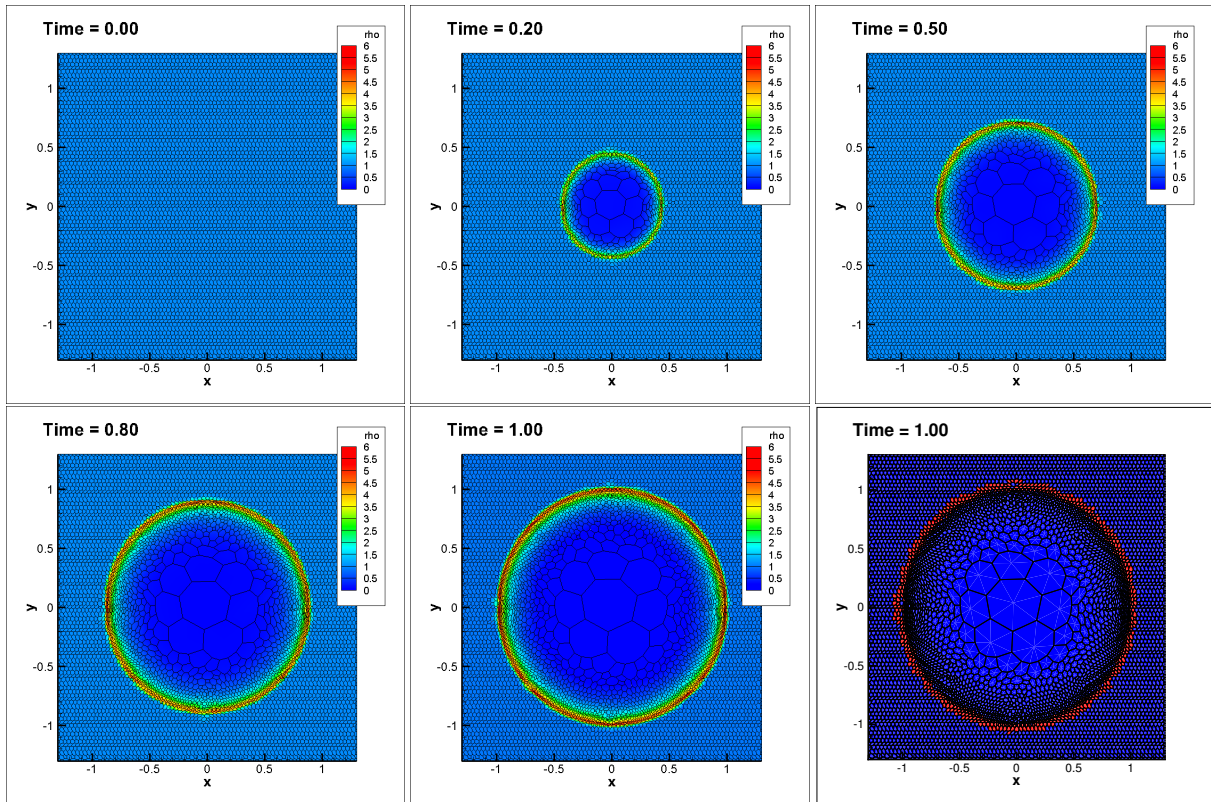
27

Figure 11: Sedov problem solved with our $P_1 P_1$ scheme on a moving Voronoi mesh of 6399 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.
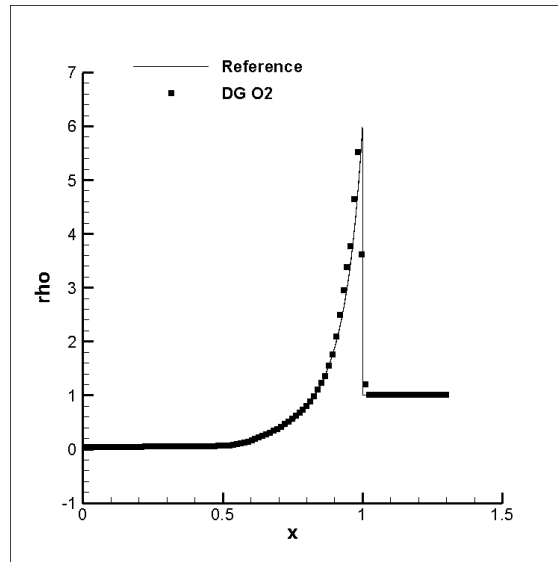

Figure 12: Sedov problem solved with our DG scheme of order 2 on moving Voronoi meshes. We compare the density profile of our numerical solution (square) with the analytic density profile (line).

$0, 0.2, 0.5, 0.8, 1.0$. The obtained results are in good agreement with the literature and the symmetry is quite good

despite a non symmetric initial mesh. Moreover, one can refer to Figure 12 for a comparison between our numerical solution 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 11 for the behavior of our *a posteriori* sub–cell finite volume limiter, which activates only and exactly where the shock wave is located.

4.2. Euler equations with source term

Next, we consider the Euler equations given in (51), but with a gravity source term of the form

$$\mathbf{S} = (0, 0, -g\rho, 0, -g\rho v)^T. \tag{57}$$

This kind of simple model is of interest not only in hydrodynamics [126, 127, 128, 129, 130], but also in the astrophysical community [1, 60, 131].

Rayleigh-Taylor instability.

With this test case we study an important type of fluid instability that arises in stratified atmospheres in approximate hydrostatic equilibrium if a denser fluid lies above a lighter phase. In such a Rayleigh-Taylor unstable state, energy can be gained if the lighter fluid rises in the gravitational field, triggering buoyancy-driven fluid motions. We consider here a simple test where we excite only one single Rayleigh-Taylor mode.

Our setup is a small variation of a similar test considered in [132] and in [1]. The computational domain is $[-0.15, 0.65] \times [0, 1.5]$, with wall boundary conditions everywhere. The imposed initial condition is given by the following hydrostatic equilibrium state

$$\begin{cases} \rho_B = 2, & p_B = P_0 + g(y - 0.75)\rho_B, & y \leq 0.75 \\ \rho_T = 1, & p_T = P_0 + g(y - 0.75)\rho_T, & y > 0.75, \end{cases} \tag{58}$$

with $P_0 = 2.5$ and $g = -0.1$. The initial velocities are zero everywhere, i.e. $\mathbf{u} = (u, v, w) = \mathbf{0}$, except for a small perturbation that is designed to excite one single mode for the Rayleigh-Taylor instability

$$v(x, y) = \omega_0 (1 - \cos(4\pi x)) (1 - cos(4\pi y/3)) \quad \text{if } 0 \leq x \leq 0.5, \tag{59}$$

where $\omega_0 = 0.0025$. Next, we smooth the initial discontinuity (in such a way that the limiter for the DG scheme will not be necessary) with a classical smoother [133]

$$\rho(\mathbf{x}) = \frac{1}{2}(\rho_B + \rho_T) + \frac{1}{2}(\rho_T - \rho_B)\operatorname{erf}\left(\frac{y - 0.75}{\varepsilon}\right). \tag{60}$$

We solve this problem deliberately on coarse meshes (M_1 made of 2 706 elements and M_2 made of 13 340 cells) and we compare the resolution of the instabilities obtained with our ALE FV-DG scheme with different order of accuracy, see Figure 13. Specifically, we compare second and third order FV and DG schemes, i.e. P_0P_1, P_1P_1, P_0P_2, P_2P_2 and we employ the Osher-type ALE flux as approximate Riemann solver (37); we note that secondary instability vortexes only appear within a high order DG method, being hidden by numerical dissipation in the other cases.

Comparing our results with those presented in [1], we underline the importance of coupling our new high order DG and FV algorithms, which provide an increased resolution on a given mesh, with a highly sophisticated software such as *AREPO*, which is able to maintain a high quality of the spatial mesh, to deal with periodic boundary conditions, and doing this in a very efficient parallel HPC environment.

4.3. 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
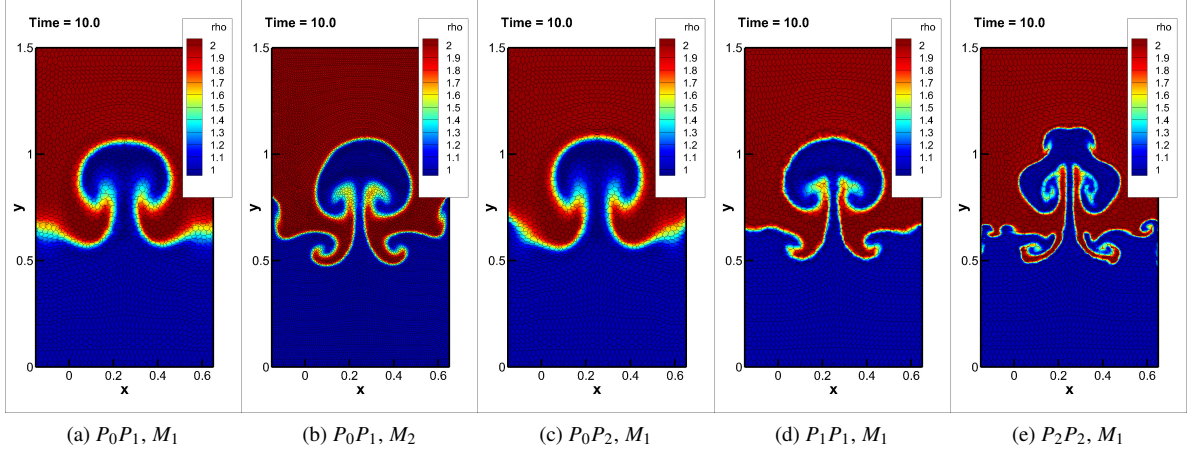
| (a) P_0P_1, M_1 | (b) P_0P_1, M_2 | (c) P_0P_2, M_1 | (d) P_1P_1, M_1 | (e) P_2P_2, M_1 |

Figure 13: Rayleigh-Taylor instabilities. The results in the panel are obtained by using two meshes: M_1 made of 2 706 elements and M_2 which is made of 13 340 elements and is 5 times finer than M_1. We have employed our FV scheme of order 2 (a,b) and 3 (c) and our DG scheme of order 2 (d) and 3 (e). We would like to underline that the use of a high order DG scheme makes secondary structures appear even on the coarse mesh $M1$ (e) which cannot be seen with standard second order FV schemes not even by refining 5 times the initial mesh (b).

form (1) are

$$
\mathbf{Q} = \begin{pmatrix} \rho \\ \rho\mathbf{v} \\ \rho E \\ \mathbf{B} \\ \psi \end{pmatrix}, \quad \mathbf{F}(\mathbf{Q}) = \begin{pmatrix} \rho\mathbf{v} \\ \rho\mathbf{v} \otimes \mathbf{v} + p_t\mathbf{I} - \frac{1}{4\pi}\mathbf{B} \otimes \mathbf{B} \\ \mathbf{v}(\rho E + p_t) - \frac{1}{4\pi}\mathbf{B}(\mathbf{v} \cdot \mathbf{B}) \\ \mathbf{v} \otimes \mathbf{B} - \mathbf{B} \otimes \mathbf{v} + \psi\mathbf{I} \\ c_h^2\mathbf{B} \end{pmatrix}. \tag{61}
$$

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

$$
p = (\gamma - 1)\left(\rho E - \frac{1}{2}\mathbf{v}^2 - \frac{\mathbf{B}^2}{8\pi}\right). \tag{62}
$$

System (61) requires an additional constraint on the divergence of the magnetic field to be satisfied, that is

$$
\nabla \cdot \mathbf{B} = 0. \tag{63}
$$

Here, (61) includes one additional scalar PDE for the evolution of the variable ψ, which is needed to transport divergence errors outside the computational domain with an artificial divergence cleaning speed c_h, see [134]. 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 [135]. A similar approach is adopted in [136, 55, 137].

4.3.1. MHD vortex

For the numerical convergence studies, we solve the vortex test problem proposed by Balsara in [138]. The computational domain is given by the box $\Omega = [0; 10] \times [0; 10]$ with wall 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

$$
\mathbf{V}(\mathbf{x}, 0) = (1, \delta u, \delta v, 0, 1 + \delta p, \delta B_x, \delta B_y, 0, 0)^T, \tag{64}
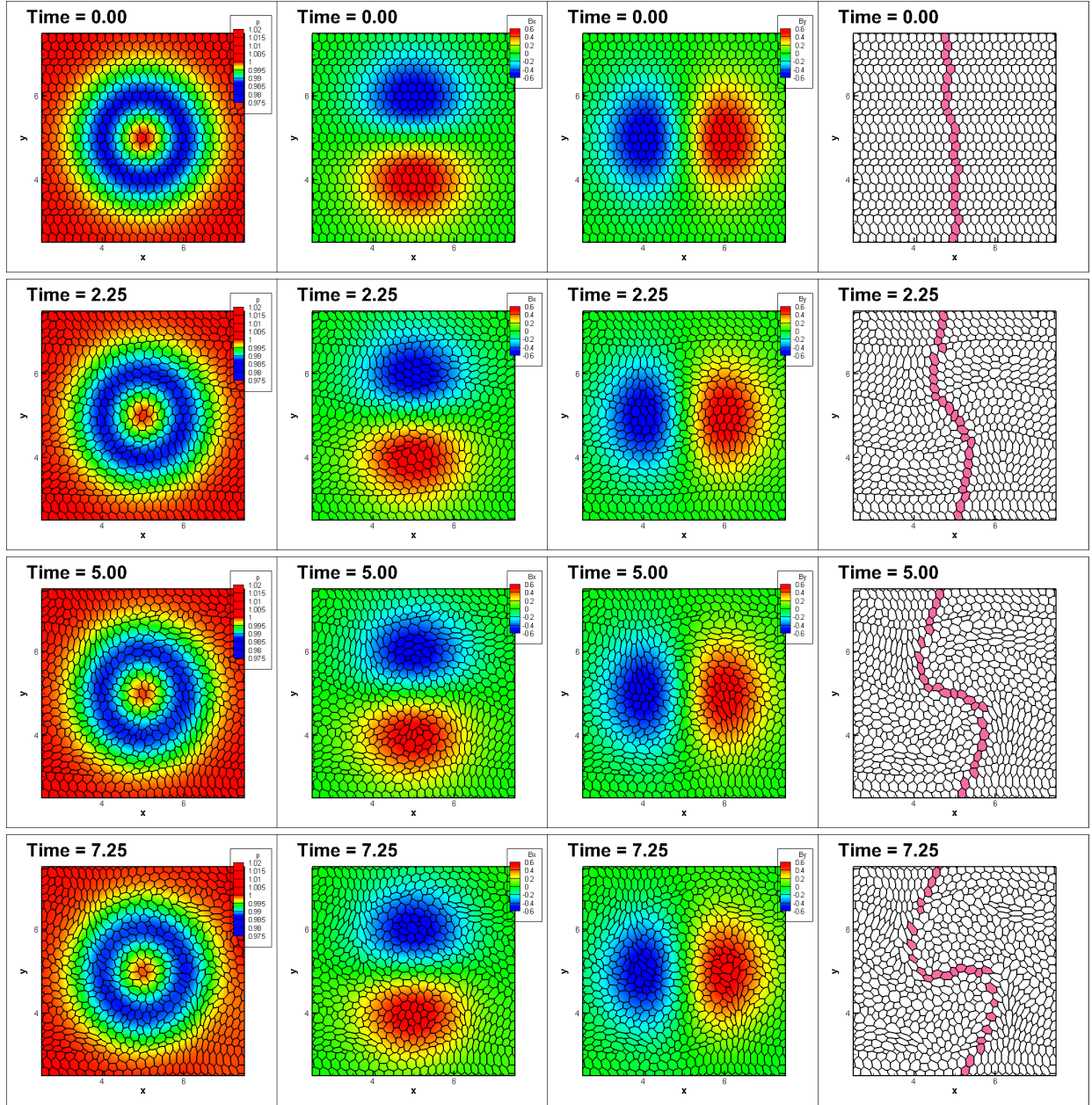$$

Figure 14: MHD vortex solved with our P_0P_3 FV scheme on a moving Voronoi mesh of 2601 elements: we depict the pressure profile (left) the $x-$ and $y-$ components of the magnetic field (middle) and the position of a bunch of highlighted elements (right) at four different times $t = 0.0, 2.25, 5.0$ and 7.25. The connectivity changes and the high order methods allow to preserve the stationary MHD vortex for long times.

with $\delta\mathbf{v} = (\delta u, \delta v, 0)^T$, $\delta\mathbf{B} = (\delta B_x, \delta B_y, 0)^T$ and

$$\delta\mathbf{v} = \frac{\kappa}{2\pi} e^{q(1-r^2)} \mathbf{e}_z \times \mathbf{r}$$
$$\delta\mathbf{B} = \frac{\mu}{2\pi} e^{q(1-r^2)} \mathbf{e}_z \times \mathbf{r},$$
$$\delta p = \frac{1}{64q\pi^3} \left(\mu^2(1 - 2qr^2) - 4\kappa^2\pi \right) e^{2q(1-r^2)}.$$

(65)

31

$P_0P_1 \to O2$			$P_0P_2 \to O3$			$P_0P_3 \to O4$			$P_0P_4 \to O5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$
4.6E-01	3.3E-02	-	3.2E-01	1.0E-02	-	4.7E-01	2.1E-02	-	6.0E-01	3.6e-0.2	-
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-0.2	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-0.2	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-0.2	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-0.2	4.8

Table 4: 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.

$P_1P_1 \to O2$			$P_2P_2 \to O3$			$P_3P_3 \to O4$			$P_4P_4 \to O5$		
$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$	$h(\Omega(t_f))$	$\varepsilon(\rho)_{L_1}$	$O(L_1)$
4.7E-01	8.5E-03	-	6.1E-01	2.8E-03	-	8.8E-01	1.1E-03	-	1.6E-00	6.9e-0.3	-
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-0.4	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-0.5	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-0.5	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-0.5	5.3

Table 5: 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.

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 [138].

Convergence. Tables 4 and 5 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 14 we show the pressure profile and the magnetic field obtained with our fourth order P_0P_3 FV scheme at different output times $t = 0, 2.25, 5.0, 7.25$. 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.

In the forth column of Figure 14 the position of a bunch of elements is highlighted 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.

4.3.2. MHD rotor problem

This last MHD test case is the classical MHD rotor problem proposed by Balsara and Spicer in [139]. It consists of a rapidly rotating fluid of high density embedded in a fluid at rest with low density. Both fluids are subject to an initially constant magnetic field. The rotor produces torsional Alfvén waves that are launched into the outer fluid at rest, resulting in a decrease of angular momentum of the spinning rotor. The computational domain is taken to be $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$. The density inside is $\rho = 10$ for $0 \le r \le 0.1$ while the density of the ambient fluid at rest is set to $\rho = 1$. The rotor has an angular velocity of $\omega = 10$. The pressure is $p = 1$ and the magnetic field vector is set to $\mathbf{B} = (2.5, 0, 0)^T$ in the entire domain. As proposed by Balsara and Spicer we apply a linear taper to the velocity and to the density in the range from $0.1 \le r \le 0.12$ so that density and velocity match those of the ambient fluid at rest at a radius of $r = 0.12$. The speed for the hyperbolic divergence cleaning is set to $c_h = 2$ and $\gamma = 1.4$ is used. Wall boundary conditions are applied everywhere. We run this problem with two different configurations: in all the cases a mesh of 22801 Voronoi elements has been employed.

(a) For the first test case we have applied our fourth order P_0P_3 Finite Volume scheme on a mesh of 22801 moving Voronoi elements, see the results in Figure 15.
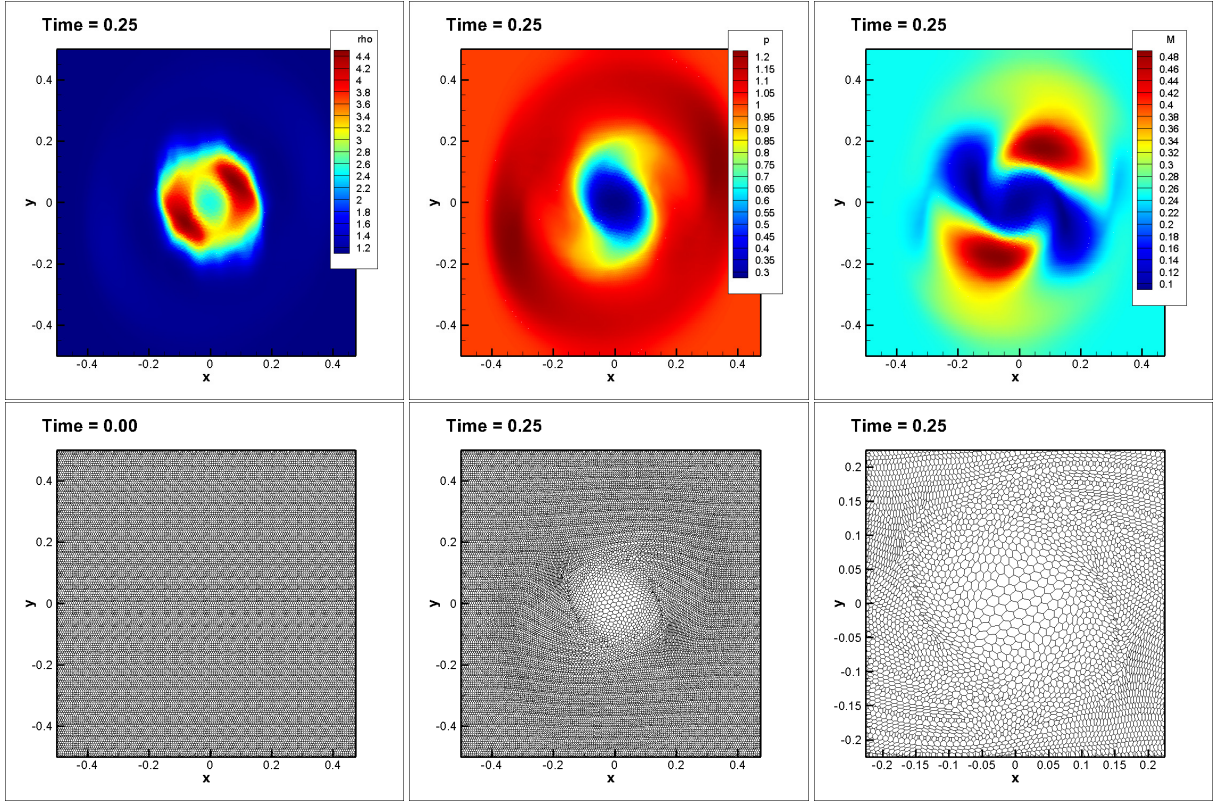
Figure 15: MHD rotor problem solved with our P_0P_3 FV scheme on a moving Voronoi mesh of 22801 elements. Top: we depict the density profile (left) the pressure profile (middle) and the magnetic density profile $M = \frac{(B_x^2 + B_y^2 + B_z^2)}{(8\pi)}$ (right). Bottom: we report the initial mesh (left), the final mesh (middle) and a zoom on the central part of the final mesh (right).

(b) Then we have employed our third order accurate P_2P_2 DG scheme on a mesh of 22801 moving Voronoi elements, see the results in Figure 16.

In all the cases, we can observe a good agreement between the obtained numerical results and those available in the literature. The comparison between our test and the literature allows also to conclude that the DG scheme, even though of one order of accuracy less w.r.t. the employed FV scheme, is more accurate. Future applications of our new algorithm will also concern the unified first order hyperbolic formulation of continuum physics recently proposed in [140, 141, 142].

5. Conclusion

In this work we have developed the *worldwide first* high order accurate direct Arbitrary-Lagrangian-Eulerian FV and DG schemes on moving unstructured Voronoi meshes with *topology change*, in order to benefit simultaneously from high order methods, high quality grids and substantially reduced numerical dissipation. Indeed, we would like to underline that in the current literature at least one of the previous ingredients is always 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. In particular, the results on vortical flows give evidence of the advantages conveyed by the proposed algorithm, and a large set of different numerical tests shows its robustness and efficiency.

We recall that the key ingredient of our novel algorithm is the generalization of the $P_N P_M$ scheme [69, 2] to Voronoi and sliver space–time elements, which has required the investigation of several intricate steps. First, the
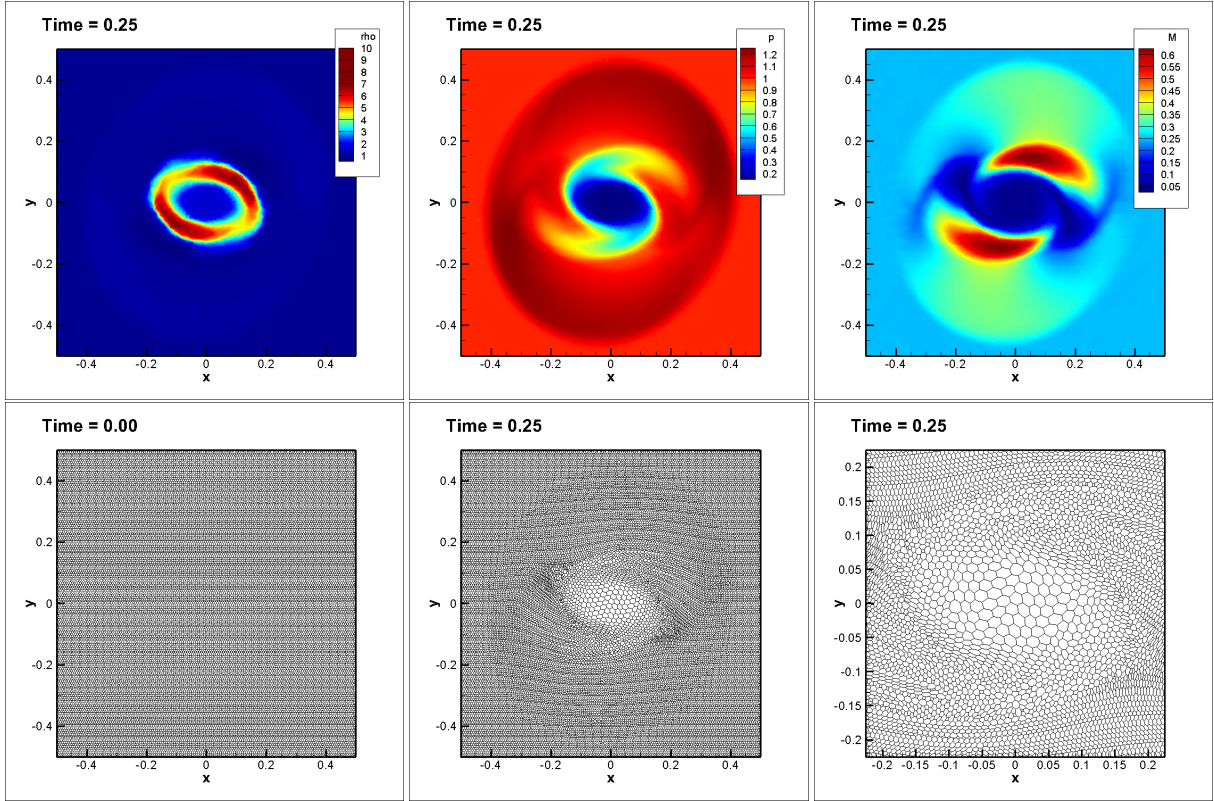
Figure 16: MHD rotor problem solved with our P_2P_2 DG scheme on a moving Voronoi mesh of 22801 elements. Top: we depict the density profile (left) the pressure profile (middle) and the magnetic density profile $M = \frac{(B_x^2 + B_y^2 + B_z^2)}{(8\pi)}$ (right). Bottom: we report the initial mesh (left), the final mesh (middle) and a zoom on the central part of the final mesh (right).

introduction of an automatic procedure to connect in space–time meshes with different topologies has never been proposed before. Next, computations on Voronoi elements have required their subdivision into triangular prisms, the adaptation of the basis functions, the neighbors search, the projection and reconstruction algorithms, and also a change in the notions of areas, volumes and characteristic mesh sizes. Finally, the presence of sliver elements forced us to revisit the core of the $P_N P_M$ scheme, i.e. the space–time predictor and the update of the solution through flux computations, in order to maintain the property of mass, momentum and energy conservation, essential for solving non linear hyperbolic equations. We would like to underline that these last points (treated in Sections 3.1.2 and 3.2.2) would represent a novelty already in one space dimension, since to the best knowledge of the authors, it is the first time that degenerate elements are taken into account in better than second order accurate FV and DG schemes.

Future work will enhance the present algorithm in *three* directions. First, we plan to incorporate a path-conservative method to treat non conservative products, so that also a well balanced treatment of sources and a proper well-balanced preservation of stationary equilibria of the PDE system will be possible, following the ideas outlined in [102, 104, 143, 54, 144, 145, 146]. Above all, we plan to incorporate the presented high order techniques inside the massively parallel second order accurate ALE-FV code *AREPO* [1], which currently includes one of the most advanced moving Voronoi mesh generators in 2D and 3D. In this way, we will ameliorate even more the quality of our moving mesh (in *AREPO* both a Lloyd algorithm [147] to make cells *rounder* and an algorithm to automatically maintain constant mass per cell are already implemented), and we will gain a very efficient parallel environment which also redistributes the moving elements among the CPU cores in a dynamic load balancing approach. At this point, even challenging astrophysical simulations will be feasible in a reasonable amount of time. Finally, the extension to three-dimensional domains is also envisaged. Although the *AREPO* code is already available in three space dimensions, it is currently still low order accurate and does not yet provide any information about the space–time connectivity of

34

the Voronoi meshes between two consecutive time levels, which is, however, needed by our high order DG and FV schemes. In our opinion, the realization of a coherent 4D space–time connection will be complex, but feasible (a first hint in this direction could be taken from [110]), and formally the $P_N P_M$ direct ALE scheme will require the same adaptations here introduced in order to deal with degenerate four dimensional space–time control volumes.

Acknowledgments

References

[1] V. Springel, E pur si muove: Galilean-invariant cosmological hydrodynamical simulations on a moving mesh, Monthly Notices of the Royal Astronomical Society (MNRAS) 401 (2010) 791–851.

[2] W. Boscheri, M. Dumbser, 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 (2014) 484 – 523.

[3] W. Boscheri, M. Dumbser, Arbitrary-Lagrangian-Eulerian discontinuous Galerkin schemes with a posteriori subcell finite volume limiting on moving unstructured meshes, Journal of Computational Physics 346 (2017) 449 – 479.

[4] J. von Neumann, R. Richtmyer, A method for the calculation of hydrodynamics shocks, Journal of Applied Physics 21 (1950) 232–237.

[5] D. J. Benson, Computational methods in lagrangian and eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (1992) 235 – 394.

[6] G. Carré, S. D. Pino, B. Després, E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension., Journal of Computational Physics 228 (2009) 5160–5183.

[7] F. De Vuyst, Lagrange-flux schemes and the entropy property, in: International Conference on Finite Volumes for Complex Applications, Springer, 2017, pp. 235–243.

[8] E. Gaburro, Well balanced Arbitrary-Lagrangian-Eulerian Finite Volume schemes on moving nonconforming meshes for non-conservative Hyperbolic systems, Ph.D. thesis, University of Trento, 2018.

[9] C. Munz, On Godunov–type schemes for Lagrangian gas dynamics, SIAM Journal on Numerical Analysis 31 (1994) 17–42.

[10] E. Caramana, D. Burton, M. Shashkov, P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, Journal of Computational Physics 146 (1998) 227–262.

[11] R. Smith, AUSM(ALE): a geometrically conservative arbitrary lagrangian–eulerian flux splitting scheme, Journal of Computational Physics 150 (1999) 268–286.

[12] P. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered lagrangian scheme for two-dimensional compressible flow problems, SIAM Journal on Scientific Computing 29 (2007) 1781–1824.

[13] A. Claisse, B. Després, E.Labourasse, F. Ledoux, A new exceptional points method with application to cell-centered Lagrangian schemes and curved meshes, Journal of Computational Physics 231 (2012) 4324–4354.

[14] S. Sambasivan, M. Shashkov, D. Burton, A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids, International Journal for Numerical Methods in Fluids 72 (2013) 770–810.

[15] P. Maire, A high-order cell-centered lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes., Journal of Computational Physics 228 (2009) 2391–2425.

[16] P. Maire, B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, Journal of Computational Physics 228 (2009) 799–821.

[17] P. Maire, A unified sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids, International Journal for Numerical Methods in Fluids 65 (2011) 1281–1294.

[18] P. Maire, A high-order one-step sub-cell force-based discretization for cell-centered lagrangian hydrodynamics on polygonal grids, Computers and Fluids 46(1) (2011) 341–347.

[19] J. Cheng, C. Shu, A high order ENO conservative Lagrangian type scheme for the compressible Euler equations, Journal of Computational Physics 227 (2007) 1567–1596.

[20] W. Liu, J. Cheng, C. Shu, High order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations, Journal of Computational Physics 228 (2009) 8872–8891.

[21] J. Cheng, C. Shu, 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 (2010) 7191–7206.

[22] R. Loubère, P. Maire, P. Váchal, A second–order compatible staggered Lagrangian hydrodynamics scheme using a cell–centered multidimensional approximate Riemann solver, Procedia Computer Science 1 (2010) 1931–1939.

[23] R. Loubère, P. Maire, P. Váchal, 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity., International Journal for Numerical Methods in Fluids 72 (2013) 22 – 42.

[24] R. Loubère, P.-H. Maire, P. Váchal, Staggered Lagrangian hydrodynamics based on cell-centered Riemann solver 10 (2010) 940–978.

[25] W. Reed, T. Hill, Triangular Mesh Methods for Neutron Transport Equation, Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.

[26] F.Vilar, P. Maire, R. Abgrall, 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 (2014) 188–234.

[27] F.Vilar, Cell-centered discontinuous Galerkin discretization for two-dimensional Lagrangian hydrodynamics, Computers and Fluids 64 (2012) 64–73.

[28] F.Vilar, P. Maire, R. Abgrall, 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) (2010) 498–604.

[29] Z. Li, X. Yu, Z. Jia, The cell–centered discontinuous Galerkin method for Lagrangian compressible Euler equations in two dimensions, Computers and Fluids 96 (2014) 152–164.

[30] A. L. Ortega, G. Scovazzi, A geometrically–conservative, synchronized, flux–corrected remap for arbitrary Lagrangian–Eulerian computations with nodal finite elements, Journal of Computational Physics 230 (2011) 6709–6741.

[31] G. Scovazzi, Lagrangian shock hydrodynamics on tetrahedral meshes: A stable and accurate variational multiscale approach, Journal of Computational Physics 231 (2012) 8029–8069.

[32] V. Dobrev, T. Ellis, T. Kolev, R. Rieben, Curvilinear Finite elements for Lagrangian hydrodynamics, International Journal for Numerical Methods in Fluids 65 (2011) 1295–1310.

[33] V. Dobrev, T. Kolev, R. Rieben, High-order curvilinear finite element methods for lagrangian hydrodynamics, SIAM Journal on Scientific Computing 34 (2012) B606–B641.

[34] V. Dobrev, T. Ellis, T. Kolev, R. Rieben, High-order curvilinear finite elements for axisymmetric lagrangian hydrodynamics, Computers & Fluids 83 (2013) 58 – 69.

[35] P. Bochev, D. Ridzal, M. Shashkov, Fast optimization-based conservative remap of scalar fields through aggregate mass transfer, Journal of Computational Physics 246 (2013) 37–57.

[36] M. Kucharik, M. Shashkov, One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods, Journal of Computational Physics 231 (2012) 2851–2864.

[37] R. Liska, M. S. P. Váchal, B. Wendroff, Synchronized flux corrected remapping for ALE methods, Computers and Fluids 46 (2011) 312–317.

[38] M. Kucharik, J. Breil, S. Galera, P. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Computers and Fluids 46 (2011) 293–297.

[39] M. Berndt, J. Breil, S. Galera, M. Kucharik, P. Maire, M. Shashkov, Two–step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods, Journal of Computational Physics 230 (2011) 6664–6687.

[40] A. Barlow, P. Maire, , W. Rider, R. Rieben, M. Shashkov, Arbitrary LagrangianEulerian methods for modeling high-speed compressible multimaterial flows, Journal of Computational Physics 322 (2016) 603–665.

[41] A. M. Winslow, Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh, J. Comput. Phys. 135 (1997) 128–138.

[42] P. Knupp, 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 (2000) 1165 – 1185.

[43] S. Galera, P. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ale scheme using vof interface reconstruction., Journal of Computational Physics 229 (2010) 5755–5787.

[44] G. Blanchard, R. Loubère, High order accurate conservative remapping scheme on polygonal meshes using a posteriori MOOD limiting, Computers and Fluids 136 (2016) 83–103.

[45] E. Caramana, The implementation of slide lines as a combined force and velocity boundary condition, Journal of Computational Physics 228 (2009) 3911–3916.

[46] S. D. Pino, 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 (2010) 1027–1032.

[47] M. Kucharik, R. Loubère, L. Bednàrik, R. Liska, Enhancement of Lagrangian slide lines as a combined force and velocity boundary condition, Computers & Fluids 83 (2013) 3–14.

[48] R. Loubère, P. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: A reconnection-based arbitrary-LagrangianEulerian method, Journal of Computational Physics 229 (2010) 4724–4761.

[49] R. Loubère, P. Maire, M. Shashkov, ReALE: A Reconnection Arbitrary-LagrangianEulerian method in cylindrical geometry, Computers and Fluids 46 (2011) 59–69.

[50] W. Bo, M. Shashkov, Adaptive reconnection-based arbitrary Lagrangian Eulerian method, Journal of Computational Physics 299 (2015) 902–939.

[51] J. Breil, T. Harribey, P. Maire, M. Shashkov, A multi-material ReALE method with MOF interface reconstruction, Computers and Fluids 83 (2013) 115–125.

[52] W. Boscheri, 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 (2017) 76–106.

[53] E. Gaburro, M. Dumbser, M. J. Castro, Direct arbitrary-lagrangian-eulerian finite volume schemes on moving nonconforming unstructured meshes, Computers and Fluids 159 (2017) 254–275.

[54] E. Gaburro, M. J. Castro, M. Dumbser, 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 (2018) 2251–2275.

[55] W. Boscheri, M. Dumbser, D. Balsara, High-order ader-weno ale schemes on unstructured triangular meshesapplication of several node solvers to hydrodynamics and magnetohydrodynamics, International Journal for Numerical Methods in Fluids 76 (2014) 737–778.

[56] W. Boscheri, M. Dumbser, Arbitrary–Lagrangian–Eulerian One–Step WENO Finite Volume Schemes on Unstructured Triangular Meshes, Communications in Computational Physics 14 (2013) 1174–1206.

[57] W. Boscheri, High order direct arbitrary-lagrangian–eulerian (ale) finite volume schemes for hyperbolic systems on unstructured meshes, Archives of Computational Methods in Engineering 24 (2017) 751–801.

[58] M. Dumbser, 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 (2014) 57–83.

[59] W. Boscheri, M. Dumbser, O. Zanotti, High Order Cell-Centered Lagrangian-Type Finite Volume Schemes with Time-Accurate Local Time Stepping on Unstructured Triangular Meshes, Journal of Computational Physics 291 (2015) 120–150.

[60] V. Springel, Moving-mesh hydrodynamics with the arepo code, Proceedings of the International Astronomical Union 6 (2010) 203–206.

[61] R. Pakmor, F. Marinacci, V. Springel, Magnetic fields in cosmological simulations of disk galaxies, The Astrophysical Journal Letters 783 (2014) L20.

[62] R. Pakmor, V. Springel, A. Bauer, P. Mocz, D. J. Munoz, S. T. Ohlmann, K. Schaal, C. Zhu, Improving the convergence properties of the moving-mesh code arepo, Monthly Notices of the Royal Astronomical Society 455 (2015) 1134–1143.

[63] S. R. Idelsohn, E. Oñate, F. D. Pin, 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 (2004) 964–984.

[64] F. D. Pin, S. R. Idelsohn, E. Oñate, R. Aubry, The ALE/Lagrangian Particle Finite Element Method: A new approach to computation of free-surface flows and fluid-object interactions, Computers and Fluids 36 (2007) 27–38.

[65] E. Oñate, S. Idelsohn, M. Celigueta, R. Rossi, 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 (2008) 1777–1800.

[66] A. Larese, R. Rossi, E. Oñate, S. Idelsohn, Validation of the Particle Finite Element Method (PFEM) for Simulation of the Free-Surface Flows, Engineering Computations 25 (2008) 385–425.

[67] S. Idelsohn, M. Mier-Torrecilla, E. Oñate, Multi–fluid flows with the Particle Finite Element Method, Comput. Methods Appl. Mech. Engrg. 198 (2009) 2750–2767.

[68] E. Oñate, M. Celigueta, S. Idelsohn, F. Salazar, B. Suarez, Possibilities of the Particle Finite Element Method for fluid–soil–structure interaction problems, Journal of Computational Mechanics 48 (2011) 307–318.

[69] M. Dumbser, D. Balsara, E. Toro, C. Munz, A unified framework for the construction of one–step finite–volume and discontinuous Galerkin schemes, Journal of Computational Physics 227 (2008) 8209–8253.

[70] A. Bowyer, Computing dirichlet tessellations, The computer journal 24 (1981) 162–166.

[71] D. F. Watson, Computing the n-dimensional delaunay tessellation with application to voronoi polytopes, The computer journal 24 (1981) 167–172.

[72] E. P. Mücke, I. Saias, B. Zhu, Fast randomized point location without preprocessing in two-and three-dimensional delaunay triangulations, Computational Geometry 12 (1999) 63–83.

[73] M. Dumbser, Arbitrary high order PNPM schemes on unstructured meshes for the compressible Navier–Stokes equations, Computers & Fluids 39 (2010) 60–76.

[74] H. Luo, L. Luo, R. Nourgaliev, V. Mousseau, N. Dinh, A reconstructed discontinuous Galerkin method for the compressible Navier–Stokes equations on arbitrary grids, Journal of Computational Physics 229 (2010) 6961–6978.

[75] H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, Z. Jiang, A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids , Journal of Computational Physics 236 (2013) 477–492.

[76] S. Godunov, Finite difference methods for the computation of discontinuous solutions of the equations of fluid dynamics, Mathematics of the USSR: Sbornik 47 (1959) 271–306.

[77] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, M2AN Math. Model. Numer. Anal. 33 (1999) 547–571.

[78] D. Levy, G. Puppo, G. Russo, A third order central WENO scheme for 2D conservation laws, Applied Numerical Mathematics 33 (2000) 415–421.

[79] D. Levy, G. Puppo, G. Russo, A fourth-order central WENO scheme for multidimensional hyperbolic systems of conservation laws, SIAM Journal on Scientific Computing 24 (2002) 480–506.

[80] I. Cravero, G. Puppo, M. Semplice, G. Visconti, Cweno: uniformly accurate reconstructions for balance laws, Mathematics of Computation 87 (2018) 1689–1719.

[81] M. Dumbser, W. Boscheri, M. Semplice, G. Russo, Central weighted eno schemes for hyperbolic conservation laws on fixed and moving unstructured meshes, SIAM Journal on Scientific Computing 39 (2017) A2564–A2591.

[82] T. Barth, P. Frederickson, Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA paper no. 90-0013 (1990).

[83] M. Käser, A. Iske, ADER schemes on adaptive triangular meshes for scalar conservation laws, Journal of Computational Physics 205 (2005)

486–508.

[84] M. Semplice, A. Coco, G. Russo, Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction, Journal of Scientific Computing 66 (2016) 692–724.

[85] M. Dumbser, M. Kaeser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, Journal of Computational Physics 221 (2007) 693 – 723.

[86] A. Stroud, Approximate Calculation of Multiple Integrals, Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971.

[87] M. Dumbser, M. Kaeser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, Journal of Computational Physics 221 (2007) 693–723.

[88] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics 126 (1996) 202–228.

[89] C. Hu, C. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, Journal of Computational Physics 150 (1999) 97–127.

[90] D. Balsara, S. Garain, C. Shu, An efficient class of WENO schemes with adaptive order, Journal of Computational Physics 326 (2016) 780–804.

[91] W. Boscheri, R. Loubère, M. Dumbser, Direct arbitrary-lagrangian–eulerian ader-mood finite volume schemes for multidimensional hyperbolic conservation laws, Journal of Computational Physics 292 (2015) 56–87.

[92] W. Boscheri, R. Loubère, 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 (2017) 271–312.

[93] V. Titarev, E. Toro, ADER: Arbitrary high order Godunov approach, Journal of Scientific Computing 17 (2002) 609–618.

[94] E. Toro, V. Titarev, Solution of the generalized Riemann problem for advection-reaction equations, Proc. Roy. Soc. London (2002) 271–281.

[95] V. Titarev, E. Toro, ADER schemes for three-dimensional nonlinear hyperbolic systems, Journal of Computational Physics 204 (2005) 715–736.

[96] T. Schwartzkopff, C. Munz, E. Toro, ADER: A high order approach for linear hyperbolic systems in 2d, Journal of Scientific Computing 17 (2002) 231–240.

[97] E. F. Toro, V. A. Titarev, Derivative Riemann solvers for systems of conservation laws and ADER methods, Journal of Computational Physics 212 (2006) 150–165.

[98] M. Dumbser, C. Enaux, E. Toro, Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws, Journal of Computational Physics 227 (2008) 3971–4001.

[99] S. Busto, J. Ferrín, E. F. Toro, M. E. Vázquez-Cendón, A projection hybrid high order finite volume/finite element method for incompressible turbulent flows, Journal of Computational Physics 353 (2018) 169–192.

[100] W. Boscheri, M. Dumbser, High order accurate direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume schemes on moving curvilinear unstructured meshes, Computers and Fluids 136 (2016) 48–66.

[101] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes, III, Journal of Computational Physics 71 (1987) 231–303.

[102] C. Parés, Numerical methods for nonconservative hyperbolic systems: a theoretical framework, SIAM Journal on Numerical Analysis 44 (2006) 300–321.

[103] M. Castro, J. Gallardo, C. Parés, 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 (2006) 1103–1134.

[104] M. Castro, J. Gallardo, J. López, C. Parés, Well-balanced high order extensions of godunov's method for semilinear balance laws, SIAM Journal of Numerical Analysis 46 (2008) 1012–1039.

[105] A. Hidalgo, M. Dumbser, Ader schemes for nonlinear systems of stiff advection–diffusion–reaction equations, Journal of Scientific Computing 48 (2011) 173–189.

[106] H. Jackson, On the eigenvalues of the ader-weno galerkin predictor, Journal of Computational Physics 333 (2017) 409–413.

[107] V. V. Rusanov, Calculation of Interaction of Non–Steady Shock Waves with Obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.

[108] S. Osher, F. Solomon, Upwind difference schemes for hyperbolic conservation laws, Math. Comput. 38 (1982) 339–374.

[109] M. Dumbser, E. F. Toro, On universal Osher–type schemes for general nonlinear hyperbolic conservation laws, Communications in Computational Physics 10 (2011) 635–671.

[110] B. Re, C. Dobrzynski, A. Guardone, An interpolation-free ale scheme for unsteady inviscid flows computations with large boundary displacements over three-dimensional adaptive grids, Journal of Computational Physics 340 (2017) 26–54.

[111] S. Clain, S. Diot, R. Loubère, A high-order finite volume method for systems of conservation lawsmulti-dimensional optimal order detection (MOOD), Journal of Computational Physics 230 (2011) 4028 – 4050.

[112] S. Diot, S. Clain, R. Loubère, Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials, Computers and Fluids 64 (2012) 43 – 63.

[113] S. Diot, R. Loubère, S. Clain, 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 (2013) 362–392.

[114] R. Loubere, M. Dumbser, S. Diot, 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 (2014) 718–763.

[115] W. Boscheri, R. Loubère, M. Dumbser, Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws, Journal of Computational Physics 292 (2015) 56–87.

[116] M. Dumbser, R. Loubère, 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 (2016) 163–199.

[117] M. Sonntag, C. Munz, Shock capturing for discontinuous galerkin methods using finite volume subcells, in: J. Fuhrmann, M. Ohlberger, C. Rohde (Eds.), Finite Volumes for Complex Applications VII, Springer, 2014, pp. 945–953.

[118] M. Dumbser, O. Zanotti, R. Loubère, S. Diot, A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws, Journal of Computational Physics 278 (2014) 47–75.

[119] O. Zanotti, F. Fambri, M. Dumbser, A. Hidalgo, Space–time adaptive ADER discontinuous Galerkin finite element schemes with a posteriori sub–cell finite volume limiting, Computers and Fluids 118 (2015) 204–224.

[120] W. Boscheri, M. Semplice, M. Dumbser, Central WENO Subcell Finite Volume Limiters for ADER Discontinuous Galerkin Schemes on Fixed and Moving Unstructured Meshes, Communications in Computational Physics 25 (2019) 311–346.

[121] L. Rannabauer, M. Dumbser, M. Bader, ADER-DG with a-posteriori finite-volume limiting to simulate tsunamis in a parallel adaptive mesh refinement framework, Computers and Fluids (2018). In press. DOI: 10.1016/j.compfluid.2018.01.031.

[122] J. N. de la Rosa, C.-D. Munz, Hybrid dg/fv schemes for magnetohydrodynamics and relativistic hydrodynamics, Computer Physics Communications 222 (2018) 113 – 135.

[123] C. Hu, C. Shu, A high-order weno finite difference scheme for the equations of ideal magnetohydrodynamics., Journal of Computational Physics 150 (1999) 561 – 594.

[124] E. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics: a Practical Introduction., Springer, 2009.

[125] L. Sedov, Similarity and Dimensional Methods in Mechanics, Academic Press, New York, 1959.

[126] N. Botta, R. Klein, S. Langenberg, S. Lützenkirchen, Well balanced finite volume methods for nearly hydrostatic flows, Journal of Computational Physics 196 (2004) 539–565.

[127] R. Käppeli, S. Mishra, Well-balanced schemes for the euler equations with gravitation, Journal of Computational Physics 259 (2014) 199–219.

[128] P. Chandrashekar, C. Klingenberg, A second order well-balanced finite volume scheme for Euler Equations with gravity, Journal on Scientific Computing 37 (2015) B382–B402.

[129] V. Desveaux, M. Zenk, C. Berthon, C. Klingenberg, A well-balanced scheme to capture non-explicit steady states in the euler equations with gravity, International Journal for Numerical Methods in Fluids 81 (2016) 104–127.

[130] C. Klingenberg, G. Puppo, M. Semplice, Arbitrary order finite volume well-balanced schemes for the Euler equations with gravity, SIAM Journal on Scientific Computing 41 (2019) A695A721.

[131] A. Mignone, C. Zanni, P. Tzeferacos, B. Van Straalen, P. Colella, G. Bodo, The pluto code for adaptive mesh computations in astrophysical fluid dynamics, The Astrophysical Journal Supplement Series 198 (2011) 7.

[132] R. Liska, B. Wendroff, Comparison of several difference schemes on 1d and 2d test problems for the euler equations, SIAM Journal on Scientific Computing 25 (2003) 995–1017.

[133] M. Tavelli, M. Dumbser, A high order semi-implicit discontinuous galerkin method for the two dimensional shallow water equations on staggered unstructured meshes, Applied Mathematics and Computation 234 (2014) 623–644.

[134] A. Dedner, F. Kemm, D. Kröner, C.-D. Munz, T. Schnitzer, M. Wesenberg, Hyperbolic divergence cleaning for the MHD equations, Journal of Computational Physics 175 (2002) 645–673.

[135] D. Balsara, M. Dumbser, Divergence-free MHD on unstructured meshes using high order finite volume schemes based on multidimensional Riemann solvers, Journal of Computational Physics (2015) 687–715.

[136] F. Fambri, M. Dumbser, O. Zanotti, Space–time adaptive ader-dg schemes for dissipative flows: Compressible navier–stokes and resistive mhd equations, Computer Physics Communications 220 (2017) 297–318.

[137] W. Boscheri, 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 (2017) 76–106.

[138] D. Balsara, Second-order accurate schemes for magnetohydrodynamics with divergence-free reconstruction, The Astrophysical Journal Supplement Series 151 (2004) 149–184.

[139] D. Balsara, D. Spicer, A staggered mesh algorithm using high order godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations, Journal of Computational Physics 149 (1999) 270–292.

[140] I. Peshkov, E. Romenski, A hyperbolic model for viscous Newtonian flows, Continuum Mechanics and Thermodynamics 28 (2016) 85–104.

[141] M. Dumbser, I. Peshkov, E. Romenski, O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of continuum mechanics: Viscous heat–conducting fluids and elastic solids, Journal of Computational Physics 314 (2016) 824–862.

[142] M. Dumbser, I. Peshkov, E. Romenski, O. Zanotti, High order ADER schemes for a unified first order hyperbolic formulation of Newtonian continuum mechanics coupled with electro–dynamics, Journal of Computational Physics 348 (2017) 298–342.

[143] E. Gaburro, M. J. Castro, M. Dumbser, A well balanced diffuse interface method for complex nonhydrostatic free surface flows, Computers & Fluids 175 (2018) 180–198.

[144] S. Del Pino, E. Labourasse, G. Morel, An asymptotic preserving multidimensional ale method for a system of two compressible flows coupled with friction, Journal of Computational Physics 363 (2018) 268–301.

[145] L. Grosheintz-Laval, R. Käppeli, High-order well-balanced finite volume schemes for the euler equations with gravitation, Journal of Computational Physics 378 (2019) 324–343.

[146] J. P. Berberich, P. Chandrashekar, C. Klingenberg, High order well-balanced finite volume methods for multi-dimensional systems of hyperbolic balance laws, arXiv preprint arXiv:1903.05154 (2019).

[147] S. Lloyd, Least squares quantization in pcm, IEEE transactions on information theory 28 (1982) 129–137.