Journal of Computational Physics 401 (2020) 108927

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

# Second-order invariant domain preserving ALE approximation of hyperbolic systems ☆

Jean-Luc Guermond [a,*], Bojan Popov [a], Laura Saavedra [b]

[a] Department of Mathematics, Texas A&M University 3368 TAMU, College Station, TX 77843, USA
[b] Departamento de Matemática Aplicada a la Ingeniería Aeroespacial, E.T.S.I. Aeronáutica y del Espacio, Universidad Politécnica de Madrid, 28040 Madrid, Spain

ARTICLE INFO

*Article history:*
Received 24 October 2018
Received in revised form 30 August 2019
Accepted 31 August 2019
Available online 18 September 2019

*Keywords:*
Conservation equations
Hyperbolic systems
Arbitrary Lagrangian Eulerian
Moving meshes
Invariant domains
Convex limiting

ABSTRACT

In this paper we introduce an invariant domain preserving arbitrary Lagrangian Eulerian method for solving hyperbolic systems. The time stepping is explicit and the approximation in space is done with continuous finite elements. The method is second-order in space and made invariant domain preserving by using a newly introduced convex limiting technique.


## 1. Introduction

The arbitrary Lagrangian Eulerian technique (ALE) was proposed in the early 1960's to solve the compressible Euler equations using finite differences; see for instance Noh [38]. This methodology has been adopted in the finite element literature and popularized by Donea and co-authors to solve problems with moving boundaries. This technique is particularly useful to simulate fluid-structure interactions. It has been shown to be very effective in the low-Mach and incompressible regimes. The reader is referred to Donea et al. [12] for a brief history of the method and a thorough review. However, the interest in this method to solve compressible fluid flow problems with finite elements has faded over the years. It is only recently that the interest for finite elements has been renewed in the scientific computing community by the landmark paper of Dobrev et al. [11]. In this groundbreaking work, the authors have shown that high-order finite elements have very good approximation properties to represent the mesh motion. In particular, high-order finite elements can sustain very large deformations without entangling. In parallel to [11], various other paths have been explored in the finite element literature to do Lagrangian hydrodynamics. Since it is out of the scope of this paper to cite all the work that has been done on this topic, we point the reader to a few directions that have come to our attention: for instance we refer to Barlow [3], where the finite element formalism has been combined with staggered grid hydrodynamics methods, to Vilar et al. [46], where a

☆ This material is based upon work supported in part by the National Science Foundation grants DMS 1619892, by the Air Force Office of Scientific Research, USAF, under grant/contract number FA9550-18-1-0397, by the Army Research Office under grant/contract number W911NF-15-1-0517 and W911NF-19-1-0431, and by the Spanish MCINN under Projects MTM2015-67030-P and TRA2016-75075-R. Draft version, September 17, 2019.
* Corresponding author.
*E-mail address:* guermond@math.tamu.edu (J.-L. Guermond).

https://doi.org/10.1016/j.jcp.2019.108927
0021-9991/

Discontinuous Galerkin scheme has been developed, and to Abgrall and Tokareva [1], where the residual distribution technique has been adapted to Lagrangian hydrodynamics. We also refer to Scovazzi [39] where linear stabilization techniques are invoked, but in view of Ern and Guermond [15], we conjecture (and it is our experience) that this type of stabilization may not be robust with respect to the various tuneable parameters involved unless it is turned off in regions where entropy is produced.

The objective of the present paper is to propose a second-order continuous finite element technique for solving hyperbolic systems in the ALE framework and to give precise mathematical statements regarding the properties of this method. This is done by revisiting the first-order method from Guermond et al. [24] and by adapting to the ALE setting the recently proposed convex limiting technique from Guermond et al. [25,27]. One novelty with respect to [24] is that the mesh motion is described with high-order polynomials, whereas to guarantee that the method is invariant domain preserving, the approximation of the conserved variables is done with piecewise linear finite elements. We confirm the claim made in Dobrev et al. [11] that using high-order elements to represent the mesh motion allows for very large deformations of the mesh without entangling. But, contrary to [11] and most of the above references (including our own work Guermond et al. [23]), where stabilization is heuristically done by adding some ad hoc artificial viscosity, the invariant domain properties of our low-order method are mathematically established (see Theorem 3.6). Another novelty is that the convex limiting technique from [25,27] is adapted to the ALE setting to produce an algorithm that is invariant domain preserving and formally second-order accurate in space. One key result of the paper is Theorem 4.6 where it is proved that once convex limiting is done, the method is guaranteed to be invariant domain preserving. To the best of our knowledge, this paper is the first one in the ALE literature where limiting is done on the specific entropy, and the (relaxed) minimum principle is guaranteed. Additional properties of the proposed approximation technique are as follows: it is formally second-order accurate in space (provided the user-defined ALE velocity is reasonable), it is conservative (Lemma 3.3), it satisfies the so-called discrete geometric conservation law (Lemma 3.4), and it is Galilean invariant (Lemma 3.8). The method is applicable to any hyperbolic system in any space dimension. The second-order accuracy is numerically shown to hold in the maximum norm for smooth solutions of nonlinear scalar conservation equations and with smooth solutions of the compressible Euler equations. The method is illustrated by testing it on benchmark problems that are standard in the ALE literature.

The paper is organized as follows. In §2, we introduce the problem and recall elementary (but important) properties of the 1D Riemann problem. We also introduce the finite element setting for the mesh motion and the finite element setting for the approximation of the hyperbolic system. The ALE algorithm is fully described in §3; more precisely, we describe in this section the mesh motion, the low-order invariant domain preserving method, and the high-order (possibly invariant domain violating) method. The convex limiting, whose purpose is to make the high-order method invariant domain preserving, is introduced in §4. Since the ALE velocity field is user-defined, it is not the purpose of the present paper to discuss its construction at length, but to give some perspective to the reader, we give in §5 a brief description of the technique we are using in our numerical examples. The proposed method is illustrated numerically in §6 on various benchmark problems.

## 2. Preliminaries

In this section we introduce the problem and define the notation for the finite element approximation of the mesh motion and of the hyperbolic system at hand. One originality of the present paper is that the approximation settings for the solution to the hyperbolic system and for the ALE motion are distinct.

### 2.1. The model problem

The objective of the present paper is to approximate the following hyperbolic system written in conservative form:

$$\begin{cases} \partial_t \boldsymbol{u} + \nabla\cdot\mathbb{f}(\boldsymbol{u}) = 0, & \text{for } (\boldsymbol{x},t)\in\mathbb{R}^d\times\mathbb{R}_+ \\ \boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}), & \text{for } \boldsymbol{x}\in\mathbb{R}^d. \end{cases} \tag{2.1}$$

Here, the dependent variable $\boldsymbol{u}$ takes values in $\mathbb{R}^m$ and is considered to be a column vector $\boldsymbol{u}=(u_1,\dots,u_m)^{\mathsf{T}}$. The space dimension $d$ is arbitrary. The flux $\mathbb{f}(\boldsymbol{u})$ is a matrix with entries $\mathbb{f}_{ik}(\boldsymbol{u})$, $1\le i\le m$, $1\le k\le d$. We denote $\mathbb{f}_i(\boldsymbol{u})$ the vector in $\mathbb{R}^d$ with entries $\mathbb{f}_{ik}(\boldsymbol{u})$, $1\le k\le d$. The flux is assumed to be Lipschitz. For any vector $\boldsymbol{n}=(n_1\dots,n_d)^{\mathsf{T}}$ in $\mathbb{R}^d$, we denote $\mathbb{f}(\boldsymbol{u})\boldsymbol{n}$ the column vector with entries $\sum_{1\le k\le d}\mathbb{f}_{ik}(\boldsymbol{u})n_k$, where $i\in\{1:m\}$. Here we recall that $\{1:m\}:=\{1,2,\dots,m\}$. We denote $\nabla\cdot\mathbb{f}(\boldsymbol{u})$ the column vector with entries $(\nabla\cdot\mathbb{f}(\boldsymbol{u}(\boldsymbol{x}))_i:=\sum_{1\le k\le d}\partial_{x_k}\mathbb{f}_{ik}(\boldsymbol{u}(\boldsymbol{x}))$. Finally, we also recall that for any entropy $\eta$, the associated entropy flux $\boldsymbol{F}$ is defined by $\nabla\boldsymbol{F}(\boldsymbol{u})=(\nabla\eta)^{\mathsf{T}}\nabla\mathbb{f}$; that is to say $\partial_{u_j}\boldsymbol{F}_k(\boldsymbol{u})=\sum_{1\le i\le m}\partial_{u_i}\eta(\boldsymbol{u})\partial_{u_j}\mathbb{f}_{ik}(\boldsymbol{u})$ for all $j\in\{1:m\}$ and all $k\in\{1:d\}$.

### 2.2. The Riemann problem

Instead of trying to give a meaning to solutions to the above system, we assume that there exists an admissible set $\mathcal{A}\subset\mathbb{R}^m$ such that the following one-dimensional Riemann problem is (uniquely) solvable

$$\partial_t\boldsymbol{v}+\partial_x(\mathbb{f}(\boldsymbol{v})\boldsymbol{n})=0,\quad (x,t)\in\mathbb{R}\times\mathbb{R}_+,\qquad \boldsymbol{v}(x,0)=\begin{cases}\boldsymbol{v}_L & \text{if } x<0\\ \boldsymbol{v}_R & \text{if } x>0,\end{cases} \tag{2.2}$$

for any unit vector $\boldsymbol{n}\in\mathbb{R}^d$ and any Riemann pair $(\boldsymbol{v}_L,\boldsymbol{v}_R)$ in $\mathcal{A}^2$. We assume also that there exist convex subsets of $\mathcal{A}$ that are invariant for (2.1). We say that a convex subset $\mathcal{B}\subset\mathcal{A}$ is invariant for (2.1) if for any Riemann pair in $\mathcal{B}$, the Riemann solution $\boldsymbol{v}(x,t)$ is in $\mathcal{B}$ for all $(x,t)\in\mathbb{R}\times\mathbb{R}_+$. The existence of such sets has been established by Chueh et al. [10] on a very large class of hyperbolic systems.

Given any Riemann pair $(\boldsymbol{v}_L,\boldsymbol{v}_R)$ in $\mathcal{A}^2$ and any unit vector $\boldsymbol{n}$ in $\mathbb{R}^d$, we denote by $\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ the fastest left wave speed and $\lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ the fastest right wave speed in the Riemann problem (2.2). These two wave speeds, henceforth called extreme left and right wave speeds, are such that $\boldsymbol{v}(x,t)=\boldsymbol{v}_L$ if $\frac{x}{t}\le\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ and $\boldsymbol{v}(x,t)=\boldsymbol{v}_R$ if $\lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)\le\frac{x}{t}$. Computing $\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ and $\lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ is often difficult and time consuming since it requires solving nonlinear problems. But, since in what follows it is sufficient to have bounds on these quantities, we denote $\lambda_L^{\min}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ and $\lambda_L^{\max}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ any quantities with the properties that

$$\lambda_L^{\min}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)\le\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R),\qquad \lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)\le\lambda_R^{\max}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R). \tag{2.3}$$

It is shown in Guermond and Popov [18], for the compressible Euler equation with co-volume equation of state, and in Guermond and Popov [21], for the shallow water equations, that one can find accurate bounds on the left and right wave speed that do not require solving iteratively nonlinear problems. We assume from now on that we have at hand a lower estimate of the left wave speed, $\lambda_L^{\min}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$, and an upper estimate on the right wave speed, $\lambda_R^{\max}(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$.

Since we are going to solve (2.2) with an ALE technique, we will have to estimate the extreme left and right wave speeds in the Riemann problem with the flux $\mathbb{f}$ replaced by $\mathbb{f}(\boldsymbol{v})-\boldsymbol{v}\otimes\mathbf{W}$ with $\mathbf{W}\in\mathbb{R}^d$ and $\boldsymbol{a}\otimes\boldsymbol{b}:=\boldsymbol{a}\boldsymbol{b}^{\mathsf{T}}$ for any $\boldsymbol{a}\in\mathbb{R}^m$ and $\boldsymbol{b}\in\mathbb{R}^d$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are seen as column vectors. This is the purpose of following statement whose proof is reported in the Appendix A.

**Lemma 2.1** *(Translation). Let $\mathbf{W}\in\mathbb{R}^d$, and let $\mathbb{g}(\boldsymbol{u}):=\mathbb{f}(\boldsymbol{u})-\boldsymbol{u}\otimes\mathbf{W}$. Then*

1. *$(\eta(\boldsymbol{u}),\boldsymbol{F}(\boldsymbol{u}))$ is an entropy pair for the flux $\mathbb{f}$ if and only if $(\eta(\boldsymbol{u}),\boldsymbol{F}(\boldsymbol{u})-\eta(\boldsymbol{u})\mathbf{W})$ is an entropy pair for the flux $\mathbb{g}$.*
2. *$\boldsymbol{w}(x,t)$ is the entropy solution to the Riemann problem (2.2) iff $\boldsymbol{w}(x+(\mathbf{W}\cdot\boldsymbol{n})t,t)$ is the entropy solution to the Riemann problem*

$$\partial_t\boldsymbol{v}+\partial_x(\mathbb{g}(\boldsymbol{v})\boldsymbol{n})=0,\quad (x,t)\in\mathbb{R}\times\mathbb{R}_+,\qquad \boldsymbol{v}(x,0)=\begin{cases}\boldsymbol{v}_L & \text{if } x<0\\ \boldsymbol{v}_R & \text{if } x>0.\end{cases} \tag{2.4}$$

3. *The two problems (2.2) and (2.4) have the same admissible sets and the same invariant sets.*
4. *Let $\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ and $\lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ be the extreme left and right wave speeds in the Riemann problem (2.2), then $\lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)-\mathbf{W}\cdot\boldsymbol{n}$ and $\lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)-\mathbf{W}\cdot\boldsymbol{n}$ are the extreme left and right wave speeds in Riemann problem (2.4).*

### 2.3. Geometric finite elements and mesh

We present in this section the setting for the approximation of the ALE motion. The finite element setting for the approximation of the hyperbolic system (2.1) is presented in §2.4. We are going to use the same notation as in Guermond et al. [24]. The reader who is familiar with finite elements and ALE motion is invited to move to §3.

Let $(\mathcal{T}_h^0)_{h>0}$ be a shape-regular sequence of matching meshes at the initial time. The symbol $^0$ in $\mathcal{T}_h^0$ refers to the initial configuration of the meshes. The meshes will deform over time by means of a user-defined velocity field. To keep track of the mesh motion we use the symbol $^n$ to mean that $\mathcal{T}_h^n$ is the mesh at time $t^n$. The mesh cells in $\mathcal{T}_h^n$ are assumed to be generated from a finite number of reference elements denoted $\widehat{K}_1,\dots,\widehat{K}_\varpi$. For example, the initial mesh could be composed of a combination of triangles and parallelograms in two space dimensions ($\varpi=2$ in this case); or it could be composed of a combination of tetrahedra, parallelepipeds, and triangular prisms in three space dimensions ($\varpi=3$ in this case). We now introduce a set of reference Lagrange finite elements $\{(\widehat{K}_r,\widehat{P}_r^{\mathrm{geo}},\widehat{\Sigma}_r^{\mathrm{geo}})\}_{1\le r\le\varpi}$. Here we use the notation (now standard) introduced by Ciarlet: $\widehat{K}_r$ is a reference element, $\widehat{P}_r^{\mathrm{geo}}$ is a reference polynomial space, and $\widehat{\Sigma}_r^{\mathrm{geo}}$ is the set of the linear forms that define the degrees of freedom. The index $r\in\{1:\varpi\}$ will be omitted in the rest of the paper to simplify the notation. We denote by $\{\widehat{\boldsymbol{a}}_i\}_{i\in\mathcal{N}^{\mathrm{geo}}}$ and $\{\widehat{\theta}_i^{\mathrm{geo}}\}_{i\in\mathcal{N}^{\mathrm{geo}}}$ the reference Lagrange nodes of $\widehat{K}$ and the associated reference Lagrange shape functions, with $\mathrm{card}(\mathcal{N}^{\mathrm{geo}}):=\dim\widehat{P}^{\mathrm{geo}}$. The purpose of the geometric reference element $(\widehat{K},\widehat{P}^{\mathrm{geo}},\widehat{\Sigma}^{\mathrm{geo}})$ is to construct the geometric mapping $T_K^n:\widehat{K}\longrightarrow K$. Let $\{\boldsymbol{a}_i^n\}_{i\in\mathcal{V}^{\mathrm{geo}}}$ be the collection of all the Lagrange nodes in the mesh $\mathcal{T}_h^n$, which we assume are organized in cells by means of the geometric connectivity array $\mathrm{j}^{\mathrm{geo}}:\mathcal{N}^{\mathrm{geo}}\times\mathcal{T}_h^n\longrightarrow\mathcal{V}^{\mathrm{geo}}$ (assumed to be independent of the time index $n$). More precisely, the connectivity array is defined such that, for any mesh cell $K\in\mathcal{T}_h^n$, $\{\boldsymbol{a}_{\mathrm{j}^{\mathrm{geo}}(i,K)}^n\}_{i\in\mathcal{N}^{\mathrm{geo}}}$ is the set of the Lagrange nodes describing $K$. The geometric mapping $T_K^n$ is then defined by

$$T_K^n(\widehat{\boldsymbol{x}}):=\sum_{i\in\mathcal{N}^{\mathrm{geo}}}\boldsymbol{a}_{\mathrm{j}^{\mathrm{geo}}(i,K)}^n\widehat{\theta}_i^{\mathrm{geo}}(\widehat{\boldsymbol{x}}); \tag{2.5}$$

that is, the geometric mapping is fully described once the location of the geometric Lagrange nodes is known at time $t^n$. The motion of the geometric Lagrange nodes is described in §3.2. In the two-dimensional numerical simulations reported

at the end of the paper we are going to use triangles, and $\widehat{P}^{\rm geo}$ will be one of the following polynomial spaces $\mathbb{P}_{1,2}$, $\mathbb{P}_{2,2}$, $\mathbb{P}_{3,2}$, $\mathbb{P}_{4,2}$, where $\mathbb{P}_{k,d}$ is the vector space composed of the $d$-variate polynomials of degree at most $k$.

Given a mesh $\mathcal{T}_h^n$, we denote by $D^n$ the computational domain generated by $\mathcal{T}_h^n$, and we define a scalar-valued space based on the geometric Lagrange finite elements $(\widehat{K},\widehat{P}^{\rm geo},\widehat{\Sigma}^{\rm geo})$: $P^{\rm geo}(\mathcal{T}_h^n):=\{v\in\mathcal{C}^0(D^n;\mathbb{R})\,|\,v_{|K}\circ T_K^n\in\widehat{P}^{\rm geo},\ \forall K\in\mathcal{T}_h^n\}$. We denote by $\{\varphi_i^{{\rm geo},n}\}_{i\in\mathcal{V}^{\rm geo}}$ the global shape functions in $P^{\rm geo}(\mathcal{T}_h^n)$. Recall that $\{\varphi_i^{{\rm geo},n}\}_{i\in\mathcal{V}^{\rm geo}}$ is a basis of $P^{\rm geo}(\mathcal{T}_h^n)$ and $\varphi^{{\rm geo},n}_{{\rm j}^{\rm geo}(i,K)}(\boldsymbol{x})=\widehat{\theta}_i^{\rm geo}((T_K^n)^{-1}(\boldsymbol{x}))$, for all $i\in\mathcal{N}^{\rm geo}$, all $K\in\mathcal{T}_h^n$ and all $\boldsymbol{x}\in K$. We finally introduce the following vector-valued space which we are going to use to represent the ALE mapping:

$$\boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h^n):=[P^{\rm geo}(\mathcal{T}_h^n)]^d. \tag{2.6}$$

For further reference we also introduce a subset of the Lagrange nodes $\{\boldsymbol{a}_i^n\}_{i\in\mathcal{V}^{\rm geo}}$, which we call vertices of the mesh $\mathcal{T}_h^n$ and denote $\{\boldsymbol{z}_i^n\}_{i\in\mathcal{Z}^{\rm geo}}$. The vertices are by definition the images by the geometric transformations of the vertices of the reference element $\widehat{K}$ (if $\widehat{K}$ is a triangle, $\widehat{K}$ has 3 vertices, if $\widehat{K}$ is a cube, $\widehat{K}$ has 8 vertices, etc.).

### 2.4. Approximating finite elements

The construction of the approximate solution to (2.1) is based on a set of reference finite elements $\{(\widehat{K}_r,\widehat{P}_r,\widehat{\Sigma}_r)\}_{1\le r\le\varpi}$. Again, the index $r\in\{1:\varpi\}$ is omitted in the rest of the paper to simplify the notation. It is important to understand at this point that $\widehat{P}^{\rm geo}$ and $\widehat{P}$ are different objects; $(\widehat{K},\widehat{P}^{\rm geo},\widehat{\Sigma}^{\rm geo})$ is a Lagrange element but $(\widehat{K},\widehat{P},\widehat{\Sigma})$ may not be; it may for instance be a Bernstein-Bezier finite element, see e.g., Lai and Schumaker [35, Chap. 2] or some other modal finite element. In the two-dimensional numerical simulations reported at the end of the paper we always take $\widehat{P}=\mathbb{P}_{1,2}$ but we use $\widehat{P}^{\rm geo}=\mathbb{P}_{k,2}$ with $k\in\{1,2,3,4\}$. The shape functions on the reference element are denoted $\{\widehat{\theta}_i\}_{i\in\mathcal{N}}$. We assume that the basis $\{\widehat{\theta}_i\}_{i\in\mathcal{N}}$ has the partition of unity property: $\sum_{i\in\mathcal{N}}\widehat{\theta}_i(\widehat{\boldsymbol{x}})=1$, for all $\widehat{\boldsymbol{x}}\in\widehat{K}$. We also assume that $\int_{\widehat{K}}\widehat{\theta}_i(\widehat{\boldsymbol{x}})\,{\rm d}\widehat{x}>0$ for all $i\in\mathcal{N}$, which is known to hold true for the shape functions in $\mathbb{P}_{1,d}$, $\mathbb{Q}_{1,d}$ and for the Bernstein spaces of any degree.

Given $\mathcal{T}_h^n$, we define $P(\mathcal{T}_h^n):=\{v\in\mathcal{C}^0(D^n;\mathbb{R})\,|\,v_{|K}\circ T_K^n\in\widehat{P},\ \forall K\in\mathcal{T}_h^n\}$, and we introduce the vector-valued spaces

$$\boldsymbol{P}_m(\mathcal{T}_h^n):=[P(\mathcal{T}_h^n)]^m,\qquad\text{and}\qquad\boldsymbol{P}_d(\mathcal{T}_h^n):=[P(\mathcal{T}_h^n)]^d. \tag{2.7}$$

The solution of (2.1) will be approximated in $\boldsymbol{P}_m(\mathcal{T}_h^n)$ and the ALE velocity field will be approximated in $\boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h^n)$ and in $\boldsymbol{P}_d(\mathcal{T}_h^n)$.

The global shape functions in $P(\mathcal{T}_h^n)$ are denoted by $\{\varphi_i^n\}_{i\in\mathcal{V}}$. Recall that these functions form a basis of $P(\mathcal{T}_h^n)$. We denote by ${\rm j}:\mathcal{N}\times\mathcal{T}_h^n\longrightarrow\mathcal{V}$ the connectivity array associated with the global shape functions $\{\varphi_i^n\}_{i\in\mathcal{V}}$. This array, which we assume to be independent of $n$, is defined such that $\varphi^n_{{\rm j}(i,K)}(\boldsymbol{x})=\widehat{\theta}_i((T_K^n)^{-1}(\boldsymbol{x}))$, for all $i\in\mathcal{N}$ and for all $K\in\mathcal{T}_h^n$. This definition implies the partition of unity property $\sum_{i\in\mathcal{V}}\varphi_i^n(\boldsymbol{x})=1$, for all $\boldsymbol{x}\in D^n$.

In the rest of the paper we are going to make frequent use of the following time-dependent vector-valued coefficients:

$$\boldsymbol{c}_{ij}^n:=\int_{D^n}\varphi_i^n(\boldsymbol{x})\nabla\varphi_j^n(\boldsymbol{x})\,{\rm d}x,\qquad\qquad\boldsymbol{n}_{ij}^n:=\frac{\boldsymbol{c}_{ij}^n}{\|\boldsymbol{c}_{ij}^n\|_{\ell^2}}. \tag{2.8}$$

Notice that the definition (2.8) together with the partition of unity property implies that $\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}^n=\boldsymbol{0}$. We are going to use repeatedly this identity.

For any $i\in\mathcal{V}$ we denote by $\mathcal{I}(i)$ the collection of the indices of the shape functions whose support has a nontrivial intersection with the shape function $\varphi_i$; that is, we set $\mathcal{I}(i):=\{j\in\mathcal{V}\,|\,|{\rm supp}(\varphi_i^n\varphi_j^n)|\neq 0\}$ where for any measurable set $E\subset D^n$, $|E|$ denotes the measure of $E$. Henceforth we call connectivity graph of $P(\mathcal{T}_h^n)$, the graph $(\mathcal{V},\mathcal{E})$ where the vertices are all the members of $\mathcal{V}$, and the edges are pairs $(i,j)$ in $\mathcal{V}^2$ such that $(i,j)$ is in $\mathcal{E}$ iff $j\in\mathcal{I}(i)$ and $i\in\mathcal{I}(j)$. Notice that, actually, $j\in\mathcal{I}(i)$ iff $i\in\mathcal{I}(j)$. The connectivity graph does not depend on $n$, since we assumed that the connectivity array ${\rm j}:\mathcal{N}\times\mathcal{T}_h^n\longrightarrow\mathcal{V}$ does not depend on $n$.

## 3. The Arbitrary Lagrangian Eulerian algorithm

We describe in this section two ALE approximations of (2.1); one is invariant domain preserving and entropy satisfying, the other one is at least second-order accurate in space but may violate the invariant domain property. We use continuous finite elements and explicit time stepping. We are going to use two different discrete settings: one for the mesh motion and one for the approximation of the hyperbolic system (2.1). The reader who is familiar with finite elements and ALE motion is invited to go to §3.2.

### 3.1. Preliminaries

Let $\mathcal{T}_h^0$ be the mesh at the initial time $t=0$. Let $\mathcal{T}_h^n$ be the mesh at time $t^n$. Let $\tau$ be the time step and let $t^{n+1}:=t^n+\tau$. The time step $\tau$ is not fixed and may vary at each time level $t^n$, but to simplify the already heavy notation, we write $\tau$ instead of $\tau^n$. We now describe the tools that are used to construct the mesh $\mathcal{T}_h^{n+1}$.

The key assumption we make in the rest of the paper is that the mesh motion at the time level $t^n$, say $\widetilde{\boldsymbol{w}}^n$, is given by the user as a discrete field represented in $\boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h^n)$; that is,

$$\widetilde{\boldsymbol{w}}^n = \sum_{i\in\mathcal{V}^{\rm geo}} \widetilde{\mathbf{W}}_i^n \varphi_i^{{\rm geo},n} \in \boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h^n). \tag{3.1}$$

Let us introduce $\boldsymbol{a}_i(t) = \boldsymbol{a}_i^n + (t-t^n)\widetilde{\boldsymbol{w}}^n(\boldsymbol{a}_i^n)$ for $t\in[t^n,t^{n+1}]$. We then define the ALE mapping $\boldsymbol{\Phi}_t : D^n \to \mathbb{R}^d$ by setting $\boldsymbol{\Phi}_t(\boldsymbol{\xi}) = \sum_{i\in\mathcal{V}^{\rm geo}} \boldsymbol{a}_i(t)\varphi_i^{{\rm geo},n}(\boldsymbol{\xi})$, for all $t\in[t^n,t^{n+1}]$ and all $\boldsymbol{\xi}\in D^n$. Recalling that $\varphi_{{\rm j}^{\rm geo}(i,K)}^{{\rm geo},n}(\boldsymbol{\xi}) := \widehat{\theta}_i((T_K^n)^{-1}(\boldsymbol{\xi}))$, this definition is equivalent to $\boldsymbol{\Phi}_{t|K}(\boldsymbol{\xi}) = \sum_{i\in\mathcal{N}^{\rm geo}} \boldsymbol{a}_{{\rm j}^{\rm geo}(i,K)}(t)\varphi_{{\rm j}^{\rm geo}(i,K)}^{{\rm geo},n}(\boldsymbol{\xi})$, for all $\boldsymbol{\xi}\in K$ and all $K\in\mathcal{T}_h^n$. We denote $D(t) := \boldsymbol{\Phi}_t(D^n)$. We assume here that either $\widetilde{\boldsymbol{w}}^n$ is a reasonably smooth field, or $\tau$ is small enough so that the mapping $\boldsymbol{\Phi}_t$ is invertible for all $t\in[t^n,t^{n+1}]$. We use $\boldsymbol{\Phi}_t$ to define the new mesh $\mathcal{T}_h^{n+1}$ by setting $D^{n+1} := \boldsymbol{\Phi}_{t^{n+1}}(D^n)$. We will be more precise in the next section.

Let $\boldsymbol{v}_{\rm A} : D^n\times[t^n,t^{n+1}] \longrightarrow \mathbb{R}^d$ be defined by setting $\boldsymbol{v}_{\rm A}(\boldsymbol{x},t) := (\partial_t\boldsymbol{\Phi})(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}),t)$, i.e., $\boldsymbol{v}_{\rm A}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t) := (\partial_t\boldsymbol{\Phi})(\boldsymbol{\xi},t)$. Notice that for the choice we made above for the mesh motion, we have $\boldsymbol{v}_{\rm A}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t) = \widetilde{\boldsymbol{w}}^n(\boldsymbol{x})$ for all $\boldsymbol{x}\in D^n$, as shown in [24, Lem. 4.3]. We are going to use the following classical result in the next section.

**Lemma 3.1** *(Liouville's formula). Let $\mathbb{J}(\boldsymbol{\xi},t) = \nabla\boldsymbol{\Phi}_t(\boldsymbol{\xi})$ be the Jacobian matrix of $\boldsymbol{\Phi}$, then $\partial_t\det(\mathbb{J}(\boldsymbol{\xi},t)) = (\nabla\cdot\boldsymbol{v}_{\rm A})(\boldsymbol{\Phi}(\boldsymbol{\xi},t),t)\times\det(\mathbb{J}(\boldsymbol{\xi},t))$.*

Let $\varphi_i^n$ be one of the global shape functions in $P_m(\mathcal{T}_h^n)$. Let $\varphi_i(\boldsymbol{x},t)$ be defined for all $t\in[t^n,t^{n+1}]$ by setting $\varphi_i(\boldsymbol{x},t) := \varphi_i^n(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}))$, then Liouville's formula implies that $\partial_t\int_{D(t)}\varphi_i(\boldsymbol{x},t)\,{\rm d}x = \int_{D(t)}\varphi_i(\boldsymbol{x},t)\nabla\cdot\widetilde{\boldsymbol{w}}^n(\boldsymbol{x})\,{\rm d}x$. This formula will be used in the next section to define the update of the mass transported by the shape functions. An important corollary of Liouville's formula is the following result which defines the ALE formulation of (2.1).

**Lemma 3.2.** *Let $\boldsymbol{u}$ be a weak solution to (2.1), then the following identity holds in the distribution sense (in time) over the interval $[t^n,t^{n+1}]$ for every function $\psi\in C_0^0(D^n;\mathbb{R})$ (with the notation $\varphi(\boldsymbol{x},t) := \psi(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}))$):*

$$\partial_t\int_{\mathbb{R}^d}\boldsymbol{u}(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,{\rm d}x = \int_{\mathbb{R}^d}\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{v}_{\rm A} - \mathbb{f}(\boldsymbol{u}))\varphi(\boldsymbol{x},t)\,{\rm d}x. \tag{3.2}$$

### 3.2. Generic algorithm

We introduce in this section a generic algorithm to approximate (2.1) using a graph viscosity and the ALE mesh motion introduced above. An invariant-domain preserving version of the algorithm is detailed in §3.3 and a high-order variant (possibly invariant-domain violating) of the algorithm is detailed in §3.4. We essentially follow the technique introduced in Guermond et al. [24, §4.3] with the important exception that now we allow the ALE velocity to be represented with a polynomial degree much higher than that used to approximate the solution to (2.1).

Let $(\mathrm{m}_i^0)_{i\in\mathcal{V}}$ be the approximation of the mass of the shape functions at time $t^0$ defined by $\mathrm{m}_i^0 := \int_{D^0}\varphi_i^0(\boldsymbol{x})\,{\rm d}x$. Note that $\mathrm{m}_i^0 > 0$ since $\int_{\widehat{K}}\widehat{\theta}_n(\widehat{\boldsymbol{x}})\,{\rm d}\widehat{x} > 0$ for all $i\in\mathcal{N}$. Let $\boldsymbol{u}_{h0} := \sum_{i\in\mathcal{V}}\mathbf{U}_i^0\varphi_i^0 \in \boldsymbol{P}_m(\mathcal{T}_h^0)$ be a reasonable approximation of the initial data $\boldsymbol{u}_0$. Let $(\mathrm{m}_i^n)_{i\in\mathcal{V}}$ be the approximations of the mass of the shape functions at time $t^n$, and $\boldsymbol{u}_h^n := \sum_{i\in\mathcal{V}}\mathbf{U}_i^n\varphi_i^n \in \boldsymbol{P}_m(\mathcal{T}_h^n)$ be the approximation of $\boldsymbol{u}$ at time $t^n$. Let $\widetilde{\boldsymbol{w}}^n \in \boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h^n)$ be the user-defined ALE velocity at time $t^n$, see (3.1).

The new mesh $\mathcal{T}_h^{n+1}$ is defined by setting $D^{n+1} := \boldsymbol{\Phi}_{t^{n+1}}(D^n)$. More precisely, since each cell in $\mathcal{T}_h^{n+1}$ is entirely defined by the location of its Lagrange nodes, it suffices to compute $\boldsymbol{\Phi}_{t^{n+1}}(\boldsymbol{a}_i^n)$ for all $i\in\mathcal{V}^{\rm geo}$ to define $\mathcal{T}_h^{n+1}$ completely. Denoting by $\boldsymbol{a}_i^{n+1} := \boldsymbol{\Phi}_{t^{n+1}}(\boldsymbol{a}_i^n)$ the new Lagrange nodes and using that $\boldsymbol{\Phi}_t(\boldsymbol{\xi}) = \sum_{i\in\mathcal{V}^{\rm geo}}\boldsymbol{a}_i(t)\varphi_i^{{\rm geo},n}(\boldsymbol{\xi})$, and $\varphi_j^{{\rm geo},n}(\boldsymbol{a}_i^n) = \delta_{ij}$, the Lagrange nodes of the new mesh are obtained by

$$\boldsymbol{a}_i^{n+1} := \boldsymbol{a}_i^n + \tau\,\widetilde{\boldsymbol{w}}^n(\boldsymbol{a}_i^n). \tag{3.3}$$

We now estimate the mass of the shape function $\varphi_i^{n+1} := \varphi_i^n\circ\boldsymbol{\Phi}_{t^{n+1}}^{-1}$. A natural definition could be to take $\int_{D^{n+1}}\varphi_i^{n+1}(\boldsymbol{x})\,{\rm d}x$ to be the mass at $t^{n+1}$, but, as explained in Guermond et al. [24, §4.4.2], it is not possible with this definition to construct strong stability preserving (SSP) extensions of the algorithm that are both conservative and invariant domain preserving. To be able to use higher-order SSP time stepping techniques and be both conservative and invariant domain preserving, we define $\mathrm{m}_i^{n+1}$ by approximating the identity $\partial_t\int_{D(t)}\varphi_i(\boldsymbol{x},t)\,{\rm d}x = \int_{D(t)}\varphi_i(\boldsymbol{x},t)\nabla\cdot\widetilde{\boldsymbol{w}}^n(\boldsymbol{x})\,{\rm d}x$, (which is a consequence of Liouville's theorem), with the forward Euler method. That is, we set $\mathrm{m}_i^{n+1} := \mathrm{m}_i^n + \tau\int_{D^n}\varphi_i^n(\boldsymbol{x})\nabla\cdot\boldsymbol{w}^n(\boldsymbol{x})\,{\rm d}x$, where $\boldsymbol{w}^n := \Pi_h(\widetilde{\boldsymbol{w}}^n) \in \boldsymbol{P}_d(\mathcal{T}_h)$ is a reasonable approximation of $\widetilde{\boldsymbol{w}}^n$ in $\boldsymbol{P}_d(\mathcal{T}_h)$. In all the numerical tests reported at the end of the paper $\Pi_h : \boldsymbol{P}_d^{\rm geo}(\mathcal{T}_h) \to \boldsymbol{P}_d(\mathcal{T}_h)$ is the Lagrange interpolation operator (more details are given in §5.3 and (5.7)). We henceforth set $\boldsymbol{w}^n = \sum_{i\in\mathcal{V}}\mathbf{W}_i^n\varphi_i^n$. Then using the definition of $\boldsymbol{c}_{ij}^n$, we obtain

$$\mathfrak{m}_i^{n+1} := \mathfrak{m}_i^n + \tau \sum_{j\in\mathcal{I}(i)} \mathbf{W}_j^n \cdot \boldsymbol{c}_{ij}^n. \tag{3.4}$$

Recall that the initialization is done by setting $\mathfrak{m}_i^0 := \int_{D^0} \varphi_i^0(\boldsymbol{x})\,\mathrm{d}x$. At this point we assume again that the user-defined ALE velocity field $\widetilde{\boldsymbol{w}}^n$ is reasonable or $\tau$ is small enough so that $\mathfrak{m}_i^{n+1}$ is positive. More specific implementation details regarding the construction of the ALE velocity are given in §5.

Finally we compute $\boldsymbol{u}_h^{n+1}$ by using the following explicit technique:

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n\mathbf{U}_i^n}{\tau} = \sum_{j\in\mathcal{I}(i)} (\mathbf{U}_j^n \otimes \mathbf{W}_j^n - \mathbb{f}(\mathbf{U}_j^n))\boldsymbol{c}_{ij}^n + d_{ij}^n(\mathbf{U}_j^n - \mathbf{U}_i^n), \tag{3.5}$$

where $\boldsymbol{u}_h^{n+1} := \sum_{i\in\mathcal{V}} \mathbf{U}_i^{n+1}\varphi_i^{n+1} \in \boldsymbol{P}_m(\mathcal{T}_h^{n+1})$.

We now try to give the reader some intuitive understanding for (3.4) and (3.5). As said above, (3.4) is a consequence of Liouville's theorem; it is the space and time discrete version of the equality $\partial_t \int_{D(t)} \varphi_i(\boldsymbol{x},t)\,\mathrm{d}x = \int_{D(t)} \varphi_i(\boldsymbol{x},t)\nabla\cdot\widetilde{\boldsymbol{w}}^n(\boldsymbol{x})\,\mathrm{d}x$. It is essential to use $\boldsymbol{w}_h^n$ instead of $\widetilde{\boldsymbol{w}}_h^n$ in (3.4) so that the important identity (3.7) proved in Lemma 3.4 holds. In (3.5) $\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n\mathbf{U}_i^n}{\tau}$ is the forward Euler approximation of the left-hand side in (3.2). Notice that we have replaced the consistent mass matrix by the approximate lumped mass matrix to approximate the time derivative. The expression $\sum_{j\in\mathcal{I}(i)}(\mathbf{U}_j^n \otimes \mathbf{W}_j^n - \mathbb{f}(\mathbf{U}_j^n))\boldsymbol{c}_{ij}^n$ is just the Galerkin approximation of the right-hand side in (3.2) where $\boldsymbol{u}\otimes\boldsymbol{\nu}_{\mathrm{A}} - \mathbb{f}(\boldsymbol{u})$ is approximated by $\sum_{j\in\mathcal{V}}(\mathbf{U}_j^n \otimes \mathbf{W}_j^n - \mathbb{f}(\mathbf{U}_j^n))\varphi_j^n$. For any $i\in\mathcal{V}$, the coefficient $d_{ij}^n$ is an artificial viscosity for the pair of degrees of freedom $(i,j)\in\mathcal{E}$ that will be clearly defined in §3.3 and §3.4. We call $d_{ij}^n$ graph viscosity since this coefficient only involves the connectivity graph $(\mathcal{V},\mathcal{E})$ of $P(\mathcal{T}_h^n)$. We henceforth assume that $d_{ij}^n = 0$ if $j\notin\mathcal{I}(i)$ and

$$d_{ij}^n \geq 0, \text{ if } i\neq j, \quad d_{ij}^n = d_{ji}^n, \quad \text{and} \quad d_{ii}^n := \sum_{j\in\mathcal{I}(i)\setminus\{i\}} -d_{ji}^n. \tag{3.6}$$

Notice that $d_{ii}^n$ does not really need to be defined for (3.6) to make sense; this quantity is nevertheless introduced to shorten the definition of the CFL number.

Let us now address the question of conservation. The following result is proved in Lemma 5.1 in [24].

**Lemma 3.3** *(Conservation). Assume that either periodic boundary conditions are enforced or $\mathbb{f}(\mathbf{U}_i^n) - \mathbf{U}_i^n \otimes \mathbf{W}_i^n = 0$ for every $i$ such that the support of $\varphi_i^n$ touches $\partial D^n$. Then the quantity $\sum_{i\in\{1:I\}} \mathfrak{m}_i^n\mathbf{U}_i^n$ is conserved, i.e., it is independent of $n$.*

The following lemma establishes the equivalence between the conservative form of the update (3.5) with the (seemingly) non-conservative form (3.7). This result is important to prove that the low-order method, which is introduced in §3.3, is invariant domain preserving. As a byproduct, this result implies the so-called discrete geometric conservation law (DGCL). The GCL acronym has been coined in Thomas and Lombard [44] and has been repeatedly used in the ALE literature since then, (see e.g., Eq. (26) in Farhat et al. [16] or Eq. (7) in Guillard and Farhat [28]).

**Lemma 3.4** *(Non-conservative form & DGCL). The following assertions hold:*

(i) *The scheme (3.3)–(3.4)–(3.5) is equivalent to*

$$\mathfrak{m}_i^{n+1}\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\tau} = \sum_{j\in\mathcal{I}(i)} ((\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes\mathbf{W}_j^n - \mathbb{f}(\mathbf{U}_j^n))\boldsymbol{c}_{ij}^n + d_{ij}^n(\mathbf{U}_j^n - \mathbf{U}_i^n). \tag{3.7}$$

(ii) *The scheme (3.3)–(3.4)–(3.5) satisfies the discrete geometric conservation law; that is, if $\mathbf{U}_j^n = \mathbf{U}_i^n$ for all $j\in\mathcal{I}(i)$, then $\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$.*

**Proof.** (i) Multiply (3.4) by $\mathbf{U}_i^n$ and subtract the result from (3.5); see also [24, Lem. 4.4] for further details. (ii) If $\mathbf{U}_j^n = \mathbf{U}_i^n$ for all $j\in\mathcal{I}(i)$, then (3.7) implies that $\frac{\mathfrak{m}_i^{n+1}}{\tau}(\mathbf{U}_i^{n+1} - \mathbf{U}_i^n) = \sum_{j\in\mathcal{I}(i)} -\mathbb{f}(\mathbf{U}_j^n)\boldsymbol{c}_{ij}^n = -\mathbb{f}(\mathbf{U}_i^n)\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}^n = \mathbf{0}$, where the identity $\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}^n = \mathbf{0}$ is a consequence of the definition (2.8) together with the partition of unity property. Hence $\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$. □

**Remark 3.5** *(Local conservation).* The conservation statement in Lemma 3.3 may look inappropriate to the reader who is more familiar with the finite volume or/and discontinuous Galerkin literature since the notion of conservation mentioned there is global. Actually, the definition (2.8) together with the partition of unity property implies that $\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}^n = \mathbf{0}$; hence, the update (3.5) can be re-written $\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} = \mathfrak{m}_i^n\mathbf{U}_i^n + \tau\sum_{j\in\mathcal{I}(i)}\mathbb{F}_{ij}^n$ where $\mathbb{F}_{ij}^n := (\mathbf{U}_j^n\otimes\mathbf{W}_j^n + \mathbf{U}_i^n\otimes\mathbf{W}_i^n - \mathbb{f}(\mathbf{U}_j^n) - \mathbb{f}(\mathbf{U}_i^n))\boldsymbol{c}_{ij}^n + d_{ij}^n(\mathbf{U}_j^n - \mathbf{U}_i^n)$. Moreover, integration by parts shows that $\boldsymbol{c}_{ij}^n = -\boldsymbol{c}_{ji}^n$ for any $j\in\mathcal{I}(i)$ if periodic boundary conditions are enforced or if the support of $\varphi_j^n$ does not touch $\partial D^n$; notice also that we assumed that $d_{ij}^n = d_{ji}^n$. As a result, (away from

the boundary) we have $\mathbb{F}_{ij}^n = -\mathbb{F}_{ji}^n$; that is to say, the mass flux from the degree of freedom $j$ to the degree of freedom $i$ is opposite to the mass flux from the degree of freedom $i$ to the degree of freedom $j$. This is the property that is usually understood as "local mass conservation" in the finite volume or/and discontinuous Galerkin literature. In conclusion, the update (3.5), i.e., $m_i \frac{\mathbf{U}_i^{n+1}-\mathbf{U}_i^n}{\tau} = \sum_{j\in\mathcal{I}(i)} \mathbb{F}_{ij}^n$, is "locally conservative" in the sense that $\mathbb{F}_{ij}^n = -\mathbb{F}_{ji}^n$ for any $j \in \mathcal{I}(i)$ and any $i \in \mathcal{V}$ (and for any $i \in \mathcal{I}(j)$ and any $j \in \mathcal{V}$) if periodic boundary conditions are enforced or away from the boundary $\partial D^n$ otherwise.

### 3.3. Low-order approximation (GMS-GV1)

We define in this section the artificial graph viscosity that makes the method (3.3)–(3.4)–(3.5) invariant domain preserving and entropy satisfying. The method is called GMS-GV1 because it is based on a Guaranteed Maximum Speed and a first order Graph Viscosity; see Guermond and Popov [19], Guermond et al. [24,25].

For the sake of completeness, we recall the construction done in [24, §4.3]. For any $j \in \mathcal{V}$, we define the flux $\mathbb{g}_j^n(\boldsymbol{v}) := \mathbb{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}_j^n$ and denote by $\lambda^{\max}(\mathbb{g}_j^n\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ some upper bound on the fastest wave speed in the following Riemann problem:

$$\partial_t \boldsymbol{v} + \partial_x(\mathbb{g}_j^n(\boldsymbol{v})\boldsymbol{n}_{ij}^n) = 0, \quad (x,t)\in\mathbb{R}\times\mathbb{R}_+, \quad \boldsymbol{v}(x,0) = \begin{cases} \mathbf{U}_i^n & \text{if } x<0 \\ \mathbf{U}_j^n & \text{if } x>0. \end{cases} \tag{3.8}$$

An upper bound on the fastest wave speed in (3.8) can be obtained by estimating upper bounds on the extreme left and right wave speeds in the Riemann problem (2.2) with the flux $\mathbb{f}(\boldsymbol{v})\boldsymbol{n}_{ij}^n$ and initial data $(\mathbf{U}_i^n, \mathbf{U}_j^n)$. More precisely, let $\lambda_L^{\max}(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ and $\lambda_R^{\max}(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ be some upper bounds for the extreme left and right wave speeds in (2.2). Then Lemma 2.1 implies that the following quantity is an upper bound on the extreme left and right wave speeds in (3.8):

$$\lambda^{\max}(\mathbb{g}_j^n\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) := \max(|\lambda_L^{\max}(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) - \mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|, |\lambda_R^{\max}(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) - \mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|). \tag{3.9}$$

A very fast algorithm to compute $\lambda_L(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ and $\lambda_R(\mathbb{f}\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ for the compressible Euler equations is described in [18]; we also refer the reader to Toro [45] for more details on this question. We now set

$$d_{ij}^{\mathrm{L},n} = \max(\lambda^{\max}(\mathbb{g}_j^n\boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)\|\boldsymbol{c}_{ij}^n\|_{\ell^2}, \lambda^{\max}(\mathbb{g}_i^n\boldsymbol{n}_{ji}^n, \mathbf{U}_j^n, \mathbf{U}_i^n)\|\boldsymbol{c}_{ji}^n\|_{\ell^2}). \tag{3.10}$$

The following result is established in [24, Thm. 5.2]:

**Theorem 3.6** *(Local invariance). Let $n>0$ and $i\in\mathcal{V}$. Let $\mathcal{B}\subset\mathcal{A}$ be any invariant convex set. Let $\mathbf{U}_i^{\mathrm{L},n+1}$ be given by (3.3)–(3.4)–(3.5) with the viscosity defined in (3.10). Assume that the time step $\tau$ is small enough so that $\mathfrak{m}_i^{n+1} > 0$ and $2\tau\frac{|d_{ii}^{\mathrm{L},n}|}{\mathfrak{m}_i^{n+1}} \le 1$. If $\{\mathbf{U}_j^n \mid j\in\mathcal{I}(i)\}\subset\mathcal{B}$, then $\mathbf{U}_i^{\mathrm{L},n+1}\in\mathcal{B}$.*

In addition to local invariance, it is proved in Theorem 5.6 in [24] that the algorithm (3.3)–(3.4)–(3.5) yields a discrete entropy inequality for every admissible entropy pair of the system (2.1).

**Definition 3.7** *(Galilean invariance).* We say that an ALE scheme is Galilean invariant if, for any $\mathbf{W}_G\in\mathbb{R}^d$, the approximation to $\partial_t\boldsymbol{u} + \nabla\cdot(\mathbb{f}(\boldsymbol{u})) = 0$ with the ALE velocity $\widetilde{\boldsymbol{w}}$ is the same as the approximation to $\partial_t\boldsymbol{v} + \nabla\cdot(\mathbb{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}_G) = 0$ with the ALE velocity $\widetilde{\boldsymbol{w}}^n - \mathbf{W}_G$ and the same initial data.

This definition is motivated by the observation that the solutions to the two Cauchy problems $\partial_t\boldsymbol{u} + \nabla\cdot(\mathbb{f}(\boldsymbol{u})) = 0$ and $\partial_t\boldsymbol{v} + \nabla\cdot(\mathbb{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}_G) = 0$ are related by $\boldsymbol{v}(\boldsymbol{x}_0,t) = \boldsymbol{u}(\boldsymbol{x}(\boldsymbol{x}_0,t),t)$ with $\boldsymbol{x}(\boldsymbol{x}_0,t) = \boldsymbol{x}_0 + \mathbf{W}_G t$. We refer to Scovazzi [40] where a similar notion is discussed in the context of linearly stabilized methods for approximating compressible flows.

**Lemma 3.8** *(Galilean invariance). The GMS-GV1 scheme (3.5) is Galilean invariant.*

**Proof.** (1) Let $\mathcal{T}_h^{n+1}(\widetilde{\boldsymbol{w}}^n)$ be the new mesh obtained by (3.3) and let $\mathcal{T}_h^{n+1}(\widetilde{\boldsymbol{w}}^n - \mathbf{W}_G)$ be the new mesh obtained by using $\widetilde{\boldsymbol{w}}^n - \mathbf{W}_G$ as ALE velocity. Since these two meshes only differ by a uniform translation, the coefficients $\boldsymbol{c}_{ij}^n$ defined in (2.8) by the corresponding ALE mappings are identical.

(2) Since $\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij} = 0$, we have $\sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}\mathbf{W}_j^n = \sum_{j\in\mathcal{I}(i)}\boldsymbol{c}_{ij}(\mathbf{W}_j^n - \mathbf{W}_G)$. Hence the mass update defined in (3.4) is the same whether the ALE velocity is $\widetilde{\boldsymbol{w}}^n$ or $\widetilde{\boldsymbol{w}}^n - \mathbf{W}_G$.

(3) We observe that changing $\mathbf{W}_j^n$ by $\mathbf{W}_j^n - \mathbf{W}_G$ and $\mathbb{f}(\mathbf{U}_j^n)$ by $\mathbb{f}(\mathbf{U}_j^n) - \mathbf{U}_j^n\otimes\mathbf{W}_G$ does not modify the flux term in (3.5).

(4) Moreover, changing $\mathbf{W}_j^n$ by $\mathbf{W}_j^n - \mathbf{W}_G$ and $\mathbb{f}(\mathbf{U}_j^n)$ by $\mathbb{f}(\mathbf{U}_j^n) - \mathbf{U}_j^n\otimes\mathbf{W}_G$ does not change the upper bound on the fastest wave speed (3.9); hence the first-order viscosity $d_{ij}^{\mathrm{L},n}$ defined in (3.10) is unchanged.

(5) In conclusion the update in (3.5) is not changed by replacing $\mathbf{W}_j^n$ by $\mathbf{W}_j^n - \mathbf{W}_G$ and replacing $\mathbb{f}(\mathbf{U}_j^n)$ by $\mathbb{f}(\mathbf{U}_j^n) - \mathbf{U}_j^n\otimes\mathbf{W}_G$, thereby proving that the GMS-GV1 scheme is Galilean invariant. □

### 3.4. Entropy viscosity commutator (GMS-EV)

We now describe a technique that is formally high-order accurate in space but may be invariant domain violating. The method is inspired from Guermond et al. [22] and is based on the estimation of a commutator; see [25,26].

Let $(\eta(\boldsymbol{v}), \boldsymbol{F}(\boldsymbol{v}))$ be an entropy pair for (2.1), then recall that Lemma 2.1 implies that $(\eta(\boldsymbol{v}), \boldsymbol{F}(\boldsymbol{v}) - \eta(\boldsymbol{v})\mathbf{W})$ is an entropy pair for the flux $\mathbb{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}$ for any $\mathbf{W}\in\mathbb{R}^d$. Recall that by definition $\mathbb{f}$, $\eta$, and $\boldsymbol{F}$ satisfy the identity $\nabla\boldsymbol{F} = (\nabla\eta)^{\mathsf{T}}\nabla\mathbb{f}$ which means $\sum_{i=1}^m \partial_{v_i}\eta(\boldsymbol{v})\partial_{v_j}\mathbb{f}_{ik}(\boldsymbol{v}) = \partial_{v_j}\boldsymbol{F}_k(\boldsymbol{v})$ for all $j\in\{1:m\}$ and all $k\in\{1:d\}$. Let $i\in\mathcal{V}$ and $n>0$. Since every entropy can be defined up to a translation by a linear functional of the conserved variables, see Serre and Vasseur [42], we are going to introduce a directional relative entropy $\eta_i^n(\boldsymbol{v})$ and define an entropy residual based on this relative entropy. More precisely we assume that there exists a unit vector $\boldsymbol{k}$ in $\mathbb{R}^m$ such that $\boldsymbol{k}\cdot\boldsymbol{v}\neq 0$ for any $\boldsymbol{v}\in\mathcal{A}$. Then, given some state $\mathbf{U}_i^n$, we define the directional relative entropy by $\eta_i^n(\boldsymbol{v}) := \eta(\boldsymbol{v}) - \frac{\boldsymbol{k}\cdot\boldsymbol{v}}{\boldsymbol{k}\cdot\mathbf{U}_i^n}\eta(\mathbf{U}_i^n)$. Note that the convexity properties of $\eta_i^n(\boldsymbol{v})$ and $\eta(\boldsymbol{v})$ are identical. The associated entropy flux is $\boldsymbol{F}_i^n(\boldsymbol{v}) = \boldsymbol{F}(\boldsymbol{v}) - \frac{\eta(\mathbf{U}_i^n)}{\boldsymbol{k}\cdot\mathbf{U}_i^n}\boldsymbol{k}^{\mathsf{T}}\mathbb{f}(\boldsymbol{v})$. For instance, for the compressible Euler equations with conserved variable $(\rho,\boldsymbol{m},E)$, where $\rho$ is the density, $\boldsymbol{m}$ the momentum, and $E$ the total energy, it is natural to take $\boldsymbol{k} = (1,\mathbf{0},0)^{\mathsf{T}}$; in this case the directional relative entropy is $\eta_i^n(\boldsymbol{v}) = \eta(\boldsymbol{v}) - \frac{\rho}{\rho_i^n}\eta(\mathbf{U}_i^n)$.

In the ALE setting it is natural to work with the flux $\mathbb{g}(\boldsymbol{v},\mathbf{W}) := \mathbb{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}$ in the time interval $[t^n,t^{n+1}]$, with $\mathbf{W}\in\mathbb{R}^d$. Then we define the corresponding entropy flux $\boldsymbol{G}_i^n(\boldsymbol{v},\mathbf{W}) := \boldsymbol{F}_i^n(\boldsymbol{v}) - \eta_i^n(\boldsymbol{v})\mathbf{W}$. Now we want to measure how well the relation $\nabla\cdot(\boldsymbol{G}_i(\boldsymbol{v},\mathbf{W})) = (\nabla\eta_i(\boldsymbol{v}))^{\mathsf{T}}\nabla\cdot\mathbb{g}(\boldsymbol{v},\mathbf{W}))$ is satisfied by the approximate solution at the time $t^n$; here $\mathbf{W}\in\mathbb{R}^d$ is just a parameter that does not depend on $\boldsymbol{x}$. We then consider the expression $\int_{D^n}(\nabla\cdot\boldsymbol{G}_i^n(\boldsymbol{u}_h^n,\mathbf{W}_i^n) - (\nabla\eta_i^n(\boldsymbol{u}_h^n))^{\mathsf{T}}\nabla\cdot\mathbb{g}(\boldsymbol{u}_h^n,\mathbf{W}_i^n))\varphi_i^n(\boldsymbol{x})\,\mathrm{d}x$ and approximate this quantity, which we call entropy commutator, by

$$N_i^n := \sum_{j\in\mathcal{I}(i)}\Big(\boldsymbol{G}_i^n(\mathbf{U}_j^n,\mathbf{W}_i^n) - (\nabla(\eta_i^n)(\mathbf{U}_i^n))^{\mathsf{T}}\mathbb{g}(\mathbf{U}_j^n,\mathbf{W}_i^n)\Big)\cdot\boldsymbol{c}_{ij}. \tag{3.11}$$

Actually, to make $N_i^n$ Galilean invariant, it is natural to replace $\mathbf{W}_i^n$ in (3.11) by $\Delta\mathbf{W}_i^n := \mathbf{W}_i^n - \overline{\mathbf{W}_i^n}$ where $\overline{\mathbf{W}_i^n} = \frac{1}{\mathrm{card}(\mathcal{I}(i))}\sum_{j\in\mathcal{I}(i)}\mathbf{W}_j^n$. The quantity $N_i^n$ measures how well the finite element approximation satisfies the chain rule: $\nabla\cdot(\boldsymbol{G}_i^n(\boldsymbol{v},\Delta\mathbf{W}_i^n)) = (\nabla\eta_i(\boldsymbol{v}))^{\mathsf{T}}\nabla\cdot\mathbb{g}(\boldsymbol{v},\Delta\mathbf{W}_i^n)$. If $\boldsymbol{u}_h^n$ is smooth then $N_i^n$ is small since the chain rule implies that $\sum_{l=1}^m\partial_{v_l}\eta_i^n(\boldsymbol{v})\partial_{v_j}\mathbb{g}_{lk}(\boldsymbol{v},\Delta\mathbf{W}_i^n) = \partial_{v_j}\boldsymbol{G}_{i,k}^n(\boldsymbol{v},\Delta\mathbf{W}_i^n)$ for all $j\in\{1:m\}$ and all $k\in\{1:d\}$. In order to construct a non-dimensional quantity we also define

$$D_i^n := \Big|\sum_{j\in\mathcal{I}(i)}\boldsymbol{G}_i^n(\mathbf{U}_j^n,\Delta\mathbf{W}_i^n))\cdot\boldsymbol{c}_{ij}\Big| + \sum_{l=1}^m\Big|\partial_{u_l}\eta_i^n(\mathbf{U}_i^n)\Big|\times\Big|\sum_{j\in\mathcal{I}(i)}\mathbb{g}_{u_l}(\mathbf{U}_j^n,\Delta\mathbf{W}_i^n))\cdot\boldsymbol{c}_{ij}\Big|. \tag{3.12}$$

where $\mathbb{g}_{u_1},\dots,\mathbb{g}_{u_m}$ are the $\mathbb{R}^d$-valued components of the flux $\mathbb{g}$.

We then construct a normalized entropy residual $R_i^n := \frac{|N_i^n|}{D_i^n}$ and define the high-order graph viscosity (or entropy viscosity, EV) by setting

$$d_{ij}^{\mathrm{H},n} := d_{ij}^{\mathrm{L},n}\max(R_i^n,R_j^n),\ i\neq j, \qquad d_{ii}^{\mathrm{H},n} := -\sum_{j\in\mathcal{I}(i)\setminus\{i\}} d_{ij}^{\mathrm{H},n}. \tag{3.13}$$

The provisional high-order approximation $\boldsymbol{u}_h^{\mathrm{H},n+1} = \sum_{i\in\mathcal{V}}\mathbf{U}_i^{\mathrm{H},n+1}\varphi_i^{n+1}$ is computed by using (3.3)–(3.4)–(3.5) with the graph entropy viscosity defined in (3.13). The method thus constructed is henceforth referred to as GMS-EV.

**Remark 3.9** *(Directional relative entropy).* Notice that the definition of $\eta_i^n(\boldsymbol{v})$ implies that $N_i^n = \sum_{j\in\mathcal{I}(i)}(\boldsymbol{G}_i^n(\mathbf{U}_i^n,\Delta\mathbf{W}_i^n) - (\nabla\eta(\mathbf{U}_i^n))^{\mathsf{T}}\mathbb{g}(\mathbf{U}_j^n,\Delta\mathbf{W}_i^n))\cdot\boldsymbol{c}_{ij}$; that is, the value of the commutator is the same for every directional relative entropy and it is equal to the value of the commutator with the original entropy. Hence using a directional relative entropy is essentially meant to normalize properly the entropy commutator.

## 4. Quasiconcavity-based limiting

Given a convex invariant set $\mathcal{B}\subset\mathcal{A}$, and assuming that $\mathbf{U}_i^n\in\mathcal{B}$ for all $i\in\mathcal{V}$ and some $n\geq 0$, there is no guarantee that the coordinate vector of the high-order update, $(\mathbf{U}_i^{\mathrm{H},n+1})_{i\in\mathcal{V}}$, does not step out of $\mathcal{B}$. We explain in this section how to push $(\mathbf{U}_i^{\mathrm{H},n+1})_{i\in\mathcal{V}}$ back in the invariant domain by using a convex limiting technique introduced in Guermond et al. [25,27]. Most of the material of this section is inspired from [25,27] and is very loosely based on the Flux Corrected Transport paradigm of Zalesak [48] (see also Boris and Book [5]).

### 4.1. Overview of the objectives

Instead of enforcing the high-order update to be in some global invariant set $\mathcal{B}\subset\mathcal{A}$, we are now going to localize the notion of invariant sets. We now give a brief overview of the strategy before going into the details. We assume that there exists a finite set $\mathcal{L}\subset\mathbb{N}$, independent of the mesh size, say $\mathcal{L}=\{1:\mathfrak{L}\}$, and a collection of subsets $\mathcal{A}_l\subset\mathcal{A}$, $l\in\mathcal{L}$, such that for any $l\in\mathcal{L}$, any $i\in\mathcal{V}$ and $n\geq 0$, one can construct some continuous functional $\Psi_l^{i,n}:\mathcal{A}_l\to\mathbb{R}$ such that the zero level set $\{\boldsymbol{v}\in\mathcal{A}_l\,|\,\Psi_l^{i,n}(\boldsymbol{v})>0\}$ is convex. Furthermore, for practical purposes, since $\mathbf{U}_i^{\mathrm{H},n+1}$ may step outside the admissible set $\mathcal{A}$, we assume that the domain of $\Psi_1^{i,n}$ is $\mathbb{R}^m$ and we set $\mathcal{A}_1:=\mathbb{R}^m$ so that $\Psi_1^{i,n}\in\mathcal{C}^0(\mathcal{A}_1;\mathbb{R})$. Moreover, for any $l\in\mathcal{L}\backslash\{1\}$, we assume that $\{\boldsymbol{v}\in\mathcal{A}_{l-1}\,|\,\Psi_{l-1}^{i,n}(\boldsymbol{v})>0\}\subset\mathcal{A}_l$. Then we define the set

$$\mathcal{B}^{i,n}:=\bigcap_{l\in\mathcal{L}}\{\boldsymbol{v}\in\mathcal{A}_l\,|\,\Psi_l^{i,n}(\boldsymbol{v})>0\}. \tag{4.1}$$

We are going to abuse the language by calling $\mathcal{B}^{i,n}$ local invariant set. To be useful, we assume that the above construction is done such that $\mathcal{B}^{i,n}\subset\mathcal{B}$ for any $i\in\mathcal{V}$. It turns out that this construction is indeed possible for most hyperbolic systems we are aware of. Examples illustrating this construction are given in §4.2 and §6.4.1; we also refer the reader to [25, §4] and [27, §7] for other details. To sum up, given $i\in\mathcal{V}$ and $n\geq 0$, we are going to construct a collection of functionals $(\Psi_l^{i,n})_{l\in\mathcal{L}}$ and sets $\mathcal{B}^{i,n}$ so that:

(i) $\mathbf{U}_j^{\mathrm{L},n+1}\subset\mathcal{B}^{i,n}$;
(ii) if $\mathbf{U}_i^{\mathrm{H},n+1}$ is not in $\mathcal{B}^{i,n}$, we are going to correct (limit) the high-order solution so that the after limiting $\mathbf{U}_i^{n+1}$ is in $\mathcal{B}^{i,n}$.

The limiting will be done sequentially: First we limit $\mathbf{U}_i^{\mathrm{H},n+1}$ w.r.t. $\Psi_1^{i,n}$ and construct $\mathbf{U}_i^{1,n+1}$ so that $\Psi_1^{i,n}(\mathbf{U}_i^{1,n+1})>0$, hence $\mathbf{U}_i^{1,n+1}\in\mathcal{A}_2$. Then, for any $l\in\mathcal{L}\backslash\{1\}$, we construct $\mathbf{U}_i^{l,n+1}$ by limiting $\mathbf{U}_i^{l-1,n+1}$ w.r.t. $\Psi_l^{i,n}$ so that $\Psi_l^{i,n}(\mathbf{U}_i^{l,n+1})>0$. Notice that this process makes sense since $\mathbf{U}_i^{l-1,n+1}\in\mathcal{A}_l$ for any $l\in\mathcal{L}\backslash\{1\}$. Eventually we obtain $\mathbf{U}_i^{n+1}:=\mathbf{U}_i^{\mathfrak{L},n+1}$.

Notice that if one can realize the above program, i.e., if one can guarantee that $\mathbf{U}_i^{n+1}\in\mathcal{B}^{i,n}$ for all $i\in\mathcal{V}$, then $\{\mathbf{U}_i^{n+1}\}_{i\in\mathcal{V}}\subset\mathcal{B}$; that is, the limited high-order solution is invariant domain preserving.

### 4.2. Functionals and bounds

In this section we address the question of the construction of the functions $\Psi_l^{i,n}$, $l\in\mathcal{L}$. We are going to proceed as in [25, §4] and [27, §7] and adapt the strategy to the present setting.

We start by recalling the definition of the notion of quasiconcavity which will help us clarify what we are trying to achieve.

**Definition 4.1** *(Quasiconcavity).* Given a convex set $\mathcal{A}\subset\mathbb{R}^m$, we say that a function $\Psi:\mathcal{A}\to\mathbb{R}$ is quasiconcave if every *upper* level set of $\Psi$ is *convex*; that is, the set $L_\lambda(\Psi):=\{\boldsymbol{v}\in\mathcal{A}\,|\,\Psi(\boldsymbol{v})\geq\lambda\}$ is convex for any $\lambda\in\mathbb{R}$.

For instance, for the compressible Euler equations we recall that the conserved variable is $\boldsymbol{u}:=(\rho,\boldsymbol{m},E)^{\mathsf{T}}$, where $\rho$ is the density, $\boldsymbol{m}$ the momentum, and $E$ the total energy. The functional $\Psi_1:\mathbb{R}^{d+2}\ni(\rho,\boldsymbol{m},E)\mapsto\rho\in\mathbb{R}$ is linear, hence concave, hence quasiconcave; this functional is also well defined over $\mathbb{R}^{d+2}$. Let use set $\mathcal{A}_1:=\{\boldsymbol{u}\in\mathbb{R}^{d+2}\,|\,\rho>0\}$, then another important example is the internal energy $\Psi_2:\mathcal{A}_1\ni(\rho,\boldsymbol{m},E)\mapsto\epsilon(\boldsymbol{u}):=E-\frac{1}{2}\frac{\boldsymbol{m}^2}{\rho}\in\mathbb{R}$. This functional is concave for any equation of state; hence, it is also quasiconcave. It follows that the functional $\Psi_3:\{\boldsymbol{u}\in\mathbb{R}^{d+2}\,|\,\rho>0\}\ni(\rho,\boldsymbol{m},E)\mapsto e(\boldsymbol{u}):=\frac{1}{\rho}\epsilon(\boldsymbol{u})\in\mathbb{R}$, which is the specific internal energy, is quasiconcave as well. Denoting by $\eta:\{\boldsymbol{u}\in\mathbb{R}^{d+2}\,|\,\rho>0,\ e(\boldsymbol{u})>0\}\mapsto\mathbb{R}$ any entropy of the compressible Euler equations (i.e., any generalized entropy of Harten et al. [29]) and using the convention that entropies are convex, the functional $\Psi_4:\{\boldsymbol{u}\in\mathbb{R}^{d+2}\,|\,\rho>0,\ e(\boldsymbol{u})>0\}\ni\boldsymbol{u}\mapsto-\eta(\boldsymbol{u})\in\mathbb{R}$ is concave, hence quasi-concave. It follows that the negative of the specific entropy $\Psi_5:\{\boldsymbol{u}\in\mathbb{R}^{d+2}\,|\,\rho>0,\ e(\boldsymbol{u})>0\}\ni\boldsymbol{u}\mapsto-\frac{1}{\rho}\eta(\boldsymbol{u})\in\mathbb{R}$ is quasiconcave as well. We refer the reader to Frid [17] and Serre [41, Thm. 8.2.2] for more details on these questions. We are going to use the above functionals to enforce the positivity of the density, $\rho>0$, the positivity of the internal specific energy, $e(\boldsymbol{u})$, and the minimum principle on the specific entropy, $\Psi_5(\boldsymbol{u})\geq\operatorname{ess\,inf}_{\boldsymbol{x}\in D}\Psi_5(\boldsymbol{u}_0)$ (with $\eta$ being the physical entropy). In the numerical examples reported at the end of the paper we are going to use $\Psi_1$ and $\Psi_5$.

Recalling that $\boldsymbol{n}_{ij}^n:=\boldsymbol{c}_{ij}^n/\|\boldsymbol{c}_{ij}^n\|_{\ell^2}$ is a unit vector in $\mathbb{R}^d$, for all $i\in\mathcal{V}$, $j\in\mathcal{I}(i)\backslash\{i\}$, let us introduce the auxiliary state $\bar{\mathbf{U}}_{ij}^{n+1}$ defined by

$$\bar{\mathbf{U}}_{ij}^n:=\Big(\mathbb{f}(\mathbf{U}_i^n)-\mathbb{f}(\mathbf{U}_j^n)-(\mathbf{U}_i^n-\mathbf{U}_j^n)\otimes\mathbf{W}_j^n\Big)\cdot\boldsymbol{n}_{ij}^n\frac{\|\boldsymbol{c}_{ij}^n\|_{\ell^2}}{2d_{ij}^{\mathrm{L},n}}+\frac{1}{2}(\mathbf{U}_i^n+\mathbf{U}_j^n). \tag{4.2}$$

We adopt the convention that $\bar{\mathbf{U}}_{ii}^n:=\mathbf{U}_i^n$. The states $\bar{\mathbf{U}}_{ij}^{n+1}$ will play an essential role in our limiting strategy. Provided the appropriate CFL condition is satisfied, these states have the right physical properties as explained in the following statement.

**Lemma 4.2.** *Let $n>0$ and $i\in\mathcal{V}$. Let $\mathcal{B}\subset\mathcal{A}$ be any convex invariant set of (2.1). Let $j\in\mathcal{I}(i)$. Assume that $\{\mathbf{U}_i^n,\mathbf{U}_j^n\}\subset\mathcal{B}$, then $\bar{\mathbf{U}}_{ij}^n\in\mathcal{B}$.*

**Proof.** This is a direct consequence of the definition (3.10), Lemma 2.1 and [24, Lem. 2.1]. □

**Remark 4.3** *(Key property of $\{\bar{\boldsymbol{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}$).* It is established in [24, Thm. 5.2] that the GMS-GV1 solution (i.e., $\mathbf{U}_i^{\mathrm{L},n+1}$ given by (3.3)–(3.4)–(3.5) with the viscosity defined in (3.10)) is a convex combination of the sates $\{\bar{\mathbf{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}$. Lemma 3.4 is crucial to prove this property.

We are now in position to explain how the functionals $\Psi_l^{i,n}$ and the sets $\mathcal{B}^{i,n}$ introduced in (4.1) are constructed. Let us assume that we have at hand a collection of quasi-concave functionals $\{\Psi_l\in C^0(\mathcal{A}_l;\mathbb{R})\}_{l\in\mathcal{L}}$, with $\mathcal{L}=\{1:\mathfrak{L}\}$, such that $\mathcal{A}_1=\mathbb{R}^m$ and that $\{\boldsymbol{v}\in\mathcal{A}_{l-1}\mid\Psi_{l-1}(\boldsymbol{v})>0\}\subset\mathcal{A}_l$ for all $l\in\mathcal{L}\backslash\{1\}$. Notice that using the zero level set of $\Psi_l$ to define $\mathcal{A}_l$ is purely conventional since translating any quasiconcave functional by any constant still gives a quasiconcave functional. We now define

$$\mathcal{B}_{\mathcal{L}}:=\bigcap_{l\in\mathcal{L}}\{\boldsymbol{v}\in\mathcal{A}_l\mid\Psi_l(\boldsymbol{v})>0\},\tag{4.3}$$

and we assume that $\mathcal{B}_{\mathcal{L}}$ is an invariant set of (2.1). For any $l\in\mathcal{L}$, any $i\in\mathcal{V}$, and $n\geq 0$, we define $\Psi_l^{i,n}\in C^0(\mathcal{A}_l;\mathbb{R})$ as follows:

$$\Psi_l^{i,n}(\mathbf{U}):=\Psi_l(\mathbf{U})-\Psi_l^{i,n,\min},\quad\text{with}\quad\Psi_l^{i,n,\min}:=\min_{\mathbf{V}\in\{\bar{\mathbf{U}}_j^n\}_{j\in\mathcal{I}(i)}}\Psi_l(\mathbf{V}).\tag{4.4}$$

Let $\mathcal{B}^{i,n}$ be defined as in (4.1).

**Lemma 4.4.** *Let $i\in\mathcal{V}$, and $n\geq 0$. Assume that $\mathbf{U}_j^n\in\mathcal{B}_{\mathcal{L}}$ for all $j\in\mathcal{I}(i)$. Then the following holds true:*

(i) $\mathcal{B}^{i,n}\subset\mathcal{B}_{\mathcal{L}}$;

(ii) *If $\tau$ is small enough so that $\mathfrak{m}_i^{n+1}>0$ and $2\tau\frac{|d_{ii}^{\mathrm{L},n}|}{\mathfrak{m}_i^{n+1}}\leq 1$, then $\mathbf{U}_i^{\mathrm{L},n+1}\in\mathcal{B}^{i,n}$.*

**Proof.** (i) Since $\mathcal{B}_{\mathcal{L}}$ is convex and we assumed that $\mathbf{U}_j^n\in\mathcal{B}_{\mathcal{L}}$ for all $j\in\mathcal{I}(i)$, Lemma 4.2 implies that $\bar{\mathbf{U}}_{ij}^n\in\mathcal{B}_{\mathcal{L}}$ for all $j\in\mathcal{I}(i)$. This in turn implies that the convex hull of the set $\{\bar{\mathbf{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}$ is a subset of $\mathcal{B}_{\mathcal{L}}$. Let $\boldsymbol{v}\in\mathcal{B}^{i,n}$. Then $\boldsymbol{v}\in\mathcal{A}_l$ and $\Psi_l^{i,n}(\boldsymbol{v})=\Psi_l(\boldsymbol{v})-\Psi_l^{i,n,\min}\geq 0$ for any $l\in\mathcal{L}$. Hence, $\Psi_l(\boldsymbol{v})\geq\Psi_l^{i,n,\min}$. But $\Psi_l^{i,n,\min}=\min_{\mathbf{V}\in\{\bar{\mathbf{U}}_j^n\}_{j\in\mathcal{I}(i)}}\Psi_l(\mathbf{V})$ and we have established that $\{\bar{\mathbf{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}\subset\mathcal{B}_{\mathcal{L}}\subset\mathcal{A}_l$; hence, $\Psi_l^{i,n,\min}\geq\min_{\boldsymbol{w}\in\mathcal{A}_l}\Psi_l(\boldsymbol{w})\geq 0$. In conclusion, $\Psi_l(\boldsymbol{v})\geq\Psi_l^{i,n,\min}\geq 0$, which proves that $\boldsymbol{v}\in\bigcap_{l\in\mathcal{L}}\{\boldsymbol{w}\in\mathcal{A}_l\mid\Psi_l(\boldsymbol{w})\geq 0\}=:\mathcal{B}_{\mathcal{L}}$.

(ii) Notice first that $\mathbf{U}_i^{\mathrm{L},n+1}=\mathbf{U}_i^n(1-2\tau\frac{|d_{ii}^{\mathrm{L},n}|}{\mathfrak{m}_i^{n+1}})+\sum_{i\in\mathcal{I}(i)\backslash\{i\}}2\tau\frac{d_{ij}^{\mathrm{L},n}}{\mathfrak{m}_i^{n+1}}\bar{\mathbf{U}}_{ij}^n$. Hence, the CFL condition $2\tau\frac{|d_{ii}^{\mathrm{L},n}|}{\mathfrak{m}_i^{n+1}}\leq 1$ and $\mathfrak{m}_i^{n+1}>0$, implies that $\mathbf{U}_i^{\mathrm{L},n+1}$ is in the convex hull of the set $\{\bar{\mathbf{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}$. But we proved in step (i) that the convex hull of the set $\{\bar{\mathbf{U}}_{ij}^n\}_{j\in\mathcal{I}(i)}$ is a subset of $\mathcal{B}_{\mathcal{L}}$; hence $\mathbf{U}_i^{\mathrm{L},n+1}\in\mathcal{B}_{\mathcal{L}}$. □

### 4.3. Limiting

We explain in this section the convex limiting strategy. As in the FCT methodology, we start by estimating the difference $\mathbf{U}_i^{\mathrm{H},n+1}-\mathbf{U}_i^{\mathrm{L},n+1}$. This is done by subtracting (3.5), written with the high-order viscosity $d_{ij}^{\mathrm{H},n}$, from (3.5), written with the low-order viscosity $d_{ij}^{\mathrm{L},n}$. We obtain

$$\frac{\mathfrak{m}_i^{n+1}}{\tau}(\mathbf{U}_i^{\mathrm{H},n+1}-\mathbf{U}_i^{\mathrm{L},n+1})=\sum_{j\in\mathcal{I}(i)}(d_{ij}^{\mathrm{H},n}-d_{ij}^{\mathrm{L},n})(\mathbf{U}_j^n-\mathbf{U}_i^n).$$

This identity can be re-written into the following abstract form:

$$\begin{cases}\mathfrak{m}_i(\mathbf{U}_i^{\mathrm{H},n+1}-\mathbf{U}_i^{\mathrm{L},n+1})=\sum_{j\in\mathcal{I}(i)}\boldsymbol{A}_{ij}^n\\ \boldsymbol{A}_{ij}^n:=\tau(d_{ij}^{\mathrm{H},n}-d_{ij}^{\mathrm{L},n})(\mathbf{U}_j^n-\mathbf{U}_i^n).\end{cases}\tag{4.5}$$

As usual, we notice that the matrix $\boldsymbol{A}^n$ is skew-symmetric, which immediately implies that the low-order solution and the high-order solution carry the same mass: $\sum_{i\in\mathcal{V}}\mathfrak{m}_i\mathbf{U}_i^{\mathrm{H},n+1}=\sum_{i\in\mathcal{V}}\mathfrak{m}_i\mathbf{U}_i^{\mathrm{L},n+1}$.

From now on our objective is to construct an update $\mathbf{U}_i^{n+1}$ that satisfies all the constraints $\Psi_l^{i,n}(\mathbf{U}_i^{n+1}) \geq 0$, $l \in \mathcal{L}$. At this point we adopt a strategy that is different from that of the FCT algorithm [48]. We introduce a set of positive coefficients $\{\theta_{ij}\}_{j\in\mathcal{I}(i)\backslash\{i\}}$ adding up to 1; for instance, one can take $\theta_{ij} := \frac{1}{\mathrm{card}(\mathcal{I}(i))-1}$, or $\theta_{ij} = \int_{D^n} \varphi_j^n(\boldsymbol{x})\varphi_i^n(\boldsymbol{x})\,\mathrm{d}x / (\sum_{j\in\mathcal{I}(i)\backslash\{i\}} \int_{D^n} \varphi_j^n(\boldsymbol{x})\varphi_i^n(\boldsymbol{x})\,\mathrm{d}x)$, for all $j \in \mathcal{I}(i)\backslash\{i\}$. In all the computation reported at the end of the paper we have adopted the first choice. Using that $\mathbf{U}_i^{\mathrm{L},n+1} = \sum_{j\in\mathcal{I}(i)\backslash\{i\}} \theta_{ij}\mathbf{U}_i^{\mathrm{L},n+1}$, we define the high-order update by setting

$$\mathbf{U}_i^{n+1} = \sum_{j\in\mathcal{I}(i)\backslash\{i\}} \theta_{ij}(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}\mathbf{P}_{ij}^n), \qquad \text{with} \qquad \mathbf{P}_{ij}^n := \frac{1}{\mathfrak{m}_i\theta_{ij}}\mathbf{A}_{ij}^n. \tag{4.6}$$

Note that $\mathbf{U}_i^{n+1} = \mathbf{U}_i^{\mathrm{L},n+1}$ if $\ell_{ij} = 0$ and $\mathbf{U}_i^{n+1} = \mathbf{U}_i^{\mathrm{H},n+1}$ if $\ell_{ij} = 1$. The parameter $\ell_{ij}$ is called limiter. The following lemma proved in [25, Lem. 4.4] is the workhorse of the limiting technique that we propose.

**Lemma 4.5.** *Let $B \subset \mathbb{R}^m$ and $\Psi \in C^0(B;\mathbb{R})$ be such that $\{\boldsymbol{v} \in B \mid \Psi(\boldsymbol{v}) \geq 0\}$ is convex. Let $i \in \mathcal{I}$ and $j \in \mathcal{I}(i)$. Assume that $\mathbf{U}_i^{\mathrm{L},n+1} \in B$ and $\Psi(\mathbf{U}_i^{\mathrm{L},n+1}) > 0$. Then*

(i) *There is a unique $\ell_j^i \in [0,1]$ such that defined by*

$$\ell_j^i = \begin{cases} 1 & \text{if } \Psi(\mathbf{U}_i^{\mathrm{L},n+1} + \boldsymbol{P}_{ij}^n) \geq 0 \\ \max\{\ell \in [0,1] \mid \Psi(\mathbf{U}_i^{\mathrm{L},n+1} + \ell\boldsymbol{P}_{ij}^n) \geq 0\} & \text{otherwise,} \end{cases} \tag{4.7}$$

(ii) $\Psi(\mathbf{U}_i^{\mathrm{L},n+1} + \ell\boldsymbol{P}_{ij}^n) \geq 0$ *for every* $\ell \in [0, \ell_j^i]$;

(iii) *Setting $\ell_{ij} = \min(\ell_j^i, \ell_i^j)$, we have $\Psi(\mathbf{U}_i^{\mathrm{L},n+1} + \ell_{ij}\boldsymbol{P}_{ij}^n) \geq 0$ and $\ell_{ij} = \ell_{ji}$.*

(iv) *Let $\mathbf{U}_i^{n+1}$ be defined by (4.6) with $\ell_{ij}$ defined above, then $\Psi(\mathbf{U}_i^{n+1}) \geq 0$.*

**Algorithm 1** Convex limiting.

**Require:** $\mathbf{U}^{\mathrm{L},n+1}$, $\mathbf{A}^n$, and $\mathbf{P}^n$
**Ensure:** limiter matrix $\mathbb{L} = (\ell_{ij})$
**for** $i \in \mathcal{I}$ **do**
  **for** $j \in \mathcal{I}(i)\backslash\{i\}$ **do**
    $\ell_{ij} = 1$
    **for** $l \in \mathcal{L}$ **do**
      $\ell_{ij} \leftarrow \max\{\ell \in [0, \ell_{ij}] \mid \Psi_l^{i,n}(\mathbf{U}_i^{\mathrm{L},n+1} + \ell\mathbf{P}_{ij}^n) \geq 0\}$
    **end for**
  **end for**
**end for**
$\mathbb{L} \leftarrow \min(\mathbb{L}, \mathbb{L}^{\mathsf{T}})$

Lemma 4.5 can be applied to any $l \in \mathcal{L}$ with $\Psi := \Psi_l^{i,n}$ and $B := \mathcal{A}_l$. Notice that the assumptions $\mathbf{U}_i^{\mathrm{L},n+1} \in B$ and $\Psi(\mathbf{U}_i^{\mathrm{L},n+1}) > 0$ have been shown to hold true in Lemma 4.4. The full convex limiting technique is summarized in Algorithm 1. Notice that the symmetrization of the limiter is done only at the very end of the algorithm. Putting together all the above results, we finally conclude.

**Theorem 4.6** *(Local invariance). Let $i \in \mathcal{V}$ and $n \geq 0$. Assume that $\boldsymbol{U}_j^n \in \mathcal{B}_{\mathcal{L}}$ for all $j \in \mathcal{I}(i)$. Assume that the time step $\tau$ is small enough so that $\mathfrak{m}_i^{n+1} > 0$ and $2\tau\frac{d_{ii}^{\mathrm{L},n}}{\mathfrak{m}_i^{n+1}} \leq 1$. Let $\boldsymbol{U}_i^{n+1}$ be defined by (4.6) with $\ell_{ij}$ defined in Algorithm 1, then $\boldsymbol{U}_i^{n+1} \in \mathcal{B}^{i,n} \subset \mathcal{B}_{\mathcal{L}}$.*

We stress here that the local invariance statement in Theorem 4.6 holds for any user-given (reasonably smooth) ALE velocity field $\widetilde{\boldsymbol{w}}^n$. Note also that this result is completely independent on the way the high-order viscosity $d_{ij}^{\mathrm{H},n}$ is defined. The statement $\mathbf{U}_i^{n+1} \in \mathcal{B}^{i,n} \subset \mathcal{B}_{\mathcal{L}}$ not only implies that the scheme is invariant-domain preserving, but it also gives the local bounds $\mathbf{U}_i^{n+1} \in \mathcal{B}^{i,n}$, which is far stronger that only stating $\mathbf{U}_i^{n+1} \in \mathcal{B}_{\mathcal{L}}$. To the best of knowledge, it seems that Theorem 4.6 is original and has not been established in the ALE literature before.

### 4.4. Relaxing the bounds

The technique presented above is excellent to obtain second-order accuracy in the $L^1$-norm, but the bound $\Psi_l^{i,\min}$ defined in (4.4) is too tight to make the method higher-order or even second-order in the $L^\infty$-norm in the presence of smooth extrema. As observed in Khobalatte and Perthame [32, §3.3] and explained in Guermond et al. [25, §4.7], one should relax

the bound $\Psi_l^{i,\min}$ to recover full accuracy in the $L^\infty$-norm for smooth solutions. To avoid repeating ourselves, we refer the reader to [25, §4.7] where techniques to relax $\Psi_i^{\min}$ are introduced. In a nutshell, one proceeds as follows: For each $l \in \mathcal{L}$ and all $i \in \mathcal{V}$, we set

$$\Delta^2\Psi_l^{i,n} = \frac{1}{\sum_{j\in\mathcal{I}(i)\setminus\{i\}}\beta_{ij}^n}\sum_{j\in\mathcal{I}(i)\setminus\{i\}}\beta_{ij}^n(\Psi_l(\mathbf{U}_j^n)-\Psi_l(\mathbf{U}_i^n)), \tag{4.8}$$

where the coefficients $\beta_{ij}^n$ are meant to make the computation linearity-preserving (see Remark 6.2 in [27]); for instance $\beta_{ij}^n = \int_{D^n}\nabla\varphi_i^n\cdot\nabla\varphi_j^n\,dx$. Then we compute the average

$$\overline{\Delta^2\Psi_l^{i,n}} := \frac{1}{2(\mathrm{card}(\mathcal{I}(i))-1)}\sum_{j\in\mathcal{I}(i)\setminus\{i\}}(\frac{1}{2}\Delta^2\Psi_l^{i,n}+\frac{1}{2}\Delta^2\Psi_l^{j,n}), \tag{4.9}$$

and finally relax the quantity $\Psi_l^{i,n,\min}$, which is used in the definition (4.4), by redefining $\Psi_l^{i,n,\min}$ as follows:

$$\Psi_l^{i,n,\min} \leftarrow \max((1-\mathrm{sign}(\Psi_l^{i,n,\min})r_i)\Psi_l^{i,n,\min},\ \Psi_l^{i,n,\min}-|\overline{\Delta^2\Psi_l^{i,n}}|), \tag{4.10}$$

where $r_i = (\frac{m_i}{|D|})^{\frac{1.5}{d}}$. The term $(1-\mathrm{sign}(\Psi_l^{i,n,\min})r_i)\Psi_l^{i,n,\min}$ is meant to be a safeguard on coarse meshes: since $r_i \in (0,1)$, $\Psi_l^{i,n,\min}$ cannot change sign. This somewhat ad hoc threshold $(1-\mathrm{sign}(\Psi_i^{i,n,\min})r_h)$ is never active when the mesh size is fine enough. For instance, for the compressible Euler equations, when $\Psi(\mathbf{U})$ is either the density (or the internal energy), this threshold guarantees positivity of the density (or the internal energy) because in this case $(1-\mathrm{sign}(\Psi_i^{\min})r_i \geq 0$. The exponent 1.5 is also somewhat ad hoc; in principle one could take $r_i = (\frac{m_i}{|D|})^{\frac{\delta}{d}}$ with any $\delta < 2$.

When the functional $\Psi_l$ is an entropy, the relaxation method must be more aggressive since this function is constant in smooth regions. The details are explained in [25, §4.7.2].

## 5. Construction of the ALE velocity

Although the ALE velocity field $\widetilde{\boldsymbol{w}}^n$ (or, equivalently, the mesh motion) is user-defined and should not be the topic of this paper, we give in this section some details regarding the construction of the ALE velocity and the mesh motion which could be useful to the reader. All the computations reported in §6 have been done with the techniques described here.

### 5.1. Reconstruction of the Lagrangian motion

The algorithm (3.3)–(3.4)–(3.5) uses two velocities

$$\widetilde{\boldsymbol{w}}^n = \sum_{i\in\mathcal{V}^{\mathrm{geo}}}\widetilde{\mathbf{W}}_i^n\varphi_i^{\mathrm{geo},n} \in \boldsymbol{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n), \qquad \boldsymbol{w}^n = \sum_{i\in\mathcal{V}}\mathbf{W}_i^n\varphi_i^n \in \boldsymbol{P}_d(\mathcal{T}_h^n).$$

In practice the polynomial degrees of the two spaces $\boldsymbol{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n)$ and $\boldsymbol{P}_d(\mathcal{T}_h^n)$ can be different. Actually, all the computations reported in this paper are done with continuous linear Lagrange elements for $\boldsymbol{P}_d(\mathcal{T}_h^n)$, whereas we use polynomials of degrees in the range 1 to 4 for $\boldsymbol{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n)$. It has been reported in the literature, and it is also our experience, that using higher-order polynomials to represent the mesh motion limits the risks of mesh entangling and, in the case of vortical motions, postpones the time when the mesh eventually entangles. For further arguments supporting this claim we refer the reader to Dobrev et al. [11], Bazilevs et al. [4], Anderson et al. [2]

One of the motivation for using ALE methods in compressible hydrodynamics is that making the mesh motion as close as possible to the actual fluid velocity, one significantly reduces the effects of artificial viscosity. Hence, at each time level one wants to use the approximate solution $\boldsymbol{u}_h^n$ to construct the velocity $\boldsymbol{w}^n$ and, in some way, reconstruct $\widetilde{\boldsymbol{w}}^n$ from $\boldsymbol{w}^n$. For instance for the Euler equations we have $\boldsymbol{u}_h^n = \sum_{i\in\mathcal{V}}(\varrho_i^n, \mathbf{M}_i^n, \mathsf{E}_i^n)\varphi_i^n$ and it is natural to set $\mathbf{W}_i^n = \frac{\mathbf{M}_i^n}{\varrho_i^n}$. Then, using $\boldsymbol{w}^n = \sum_{i\in\mathcal{V}}\mathbf{W}_i^n\varphi_i$ would make the scheme Lagrangian, but as explained above using $\widetilde{\boldsymbol{w}}^n = \boldsymbol{w}^n$ to move the mesh is not good enough since $\boldsymbol{w}^n$ is represented with piecewise linear polynomials, whereas we want $\widetilde{\boldsymbol{w}}^n$ to be higher-order. Hence, one has to reconstruct somehow the higher-order velocity $\widetilde{\boldsymbol{w}}^n$ from the low-order representation $\boldsymbol{w}^n$. This reconstruction problem is the question that we address in the next section.

### 5.2. The butterfly algorithm

Reconstructing a high-order field from a low order representation is a problem that has been thoroughly investigated in the computer graphics literature. We propose to reconstruct $\widetilde{\boldsymbol{w}}^n$ from $\boldsymbol{w}^n$ using an algorithm called the butterfly subdivision algorithm; we refer to Dyn et al. [14] for more details.

For simplicity we assume the space dimension is $d = 2$ and the approximation space $P^n(\mathcal{T}_h^n)$ is piecewise linear, $\widehat{P} = \mathbb{P}_{1,2}$. For the time being we also assume that the geometric finite element space is quadratic, $\widehat{P}^{\mathrm{geo}} = \mathbb{P}_{2,2}$. That is to say $P^n(\mathcal{T}_h^n)$

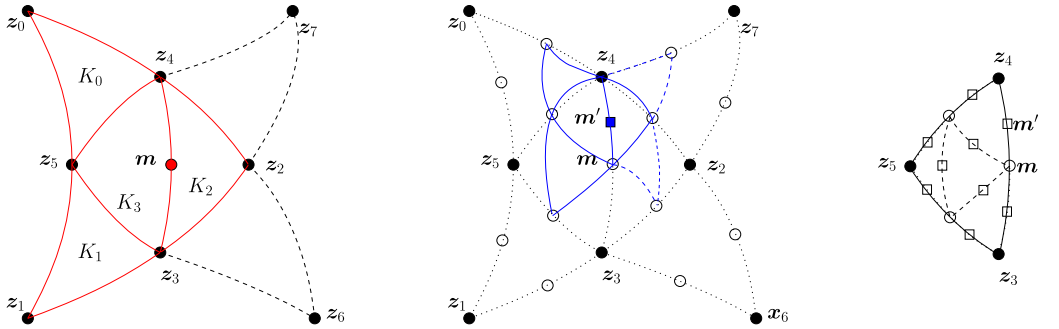


**Fig. 1.** Butterfly subdivision. Left: $\mathbb{P}_2$ subdivision. Center: $\mathbb{P}_4$ subdivision done after the $\mathbb{P}_2$ subdivision. Right: Details on the $\mathbb{P}_4$ subdivision.

is composed of continuous functions whose pullback by the geometric transformation on each cell is linear on the reference element and the geometric transformation $T_K^n : \widehat{K} \to K$ is quadratic for all $K \in \mathcal{T}_h^n$ on the reference element. Since the algorithm from [14] is valid only for triangles, if the approximation space is not based on triangles, one has to use the relevant generalization of the butterfly subdivision algorithm. There are many such algorithms available in the literature.

We now assume that we have at hand a scalar field $w \in P(\mathcal{T}_h^n)$ (here we argue with scalar-valued functions since we are going to apply the subdivision algorithm to each Cartesian component of $\boldsymbol{w}^n$). Let $K_3$ be a triangular cell with vertices $\boldsymbol{z}_3, \boldsymbol{z}_4, \boldsymbol{z}_5$. Let $K_0, K_1, K_2$ be the three triangular cells sharing a face with $K_3$, say the vertices of $K_0, K_1, K_2$ are $(\boldsymbol{z}_0, \boldsymbol{z}_4, \boldsymbol{z}_5)$, $(\boldsymbol{z}_1, \boldsymbol{z}_3, \boldsymbol{z}_5)$, $(\boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_4)$, respectively, as depicted in the left panel in Fig. 1. Recall that the geometric mappings $T_{K_0}^n, T_{K_1}^n, T_{K_2}^n$ are assumed to be quadratic, i.e., the faces of the cells in $\mathcal{T}_h^n$ are pieces of parabolas. Let us consider one face of $K_3$, say the face with the vertices $(\boldsymbol{z}_3, \boldsymbol{z}_4)$ and the midpoint $\boldsymbol{m}$ (see Fig. 1). The question we address here is to estimate a third-order approximation of $w^n$ at $\boldsymbol{m}$ using the values $w^n(\boldsymbol{z}_0), \dots, w^n(\boldsymbol{z}_5)$. The idea proposed in Dyn et al. [14] consists of viewing $\bigcup_{i=0}^3 K_i$ as a quadratic macro-element $K_{0123}$ with vertices $(\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2)$ and midpoints $\boldsymbol{z}_3, \boldsymbol{z}_4, \boldsymbol{z}_5$. Then, denoting by $T_{K_{0123}} : \widehat{K} \to K_{0123}$ the corresponding quadratic geometric mapping, one constructs the function $w_{0123} : K_{0123} \to \mathbb{R}$ whose pullback by $T_{K_{0123}}$ is quadratic on the reference element and whose values at the nodes $\boldsymbol{z}_0, \dots, \boldsymbol{z}_5$ are $w(\boldsymbol{z}_0), \dots, w(\boldsymbol{z}_5)$. A direct computation shows that $w_{0123}(\boldsymbol{m}) = -\frac{1}{8}w(\boldsymbol{z}_0) - \frac{1}{8}w(\boldsymbol{z}_1) + \frac{1}{2}w(\boldsymbol{z}_3) + \frac{1}{2}w(\boldsymbol{z}_4) + \frac{1}{4}w(\boldsymbol{z}_5)$. But after realizing that $\boldsymbol{m}$ also belongs to a face of the cell $K_2$, we can apply again the same strategy with the macro element constructed on $K_2$ using the cells $K_2, K_3$ and the cell $K_4$, $K_5$ with vertices $(\boldsymbol{z}_2, \boldsymbol{z}_3, \boldsymbol{z}_6)$ and $(\boldsymbol{z}_2, \boldsymbol{z}_4, \boldsymbol{z}_7)$, respectively. Then third-order accuracy is maintained by averaging the two estimations by setting:

$$\widetilde{w}(\boldsymbol{m}) = -\frac{1}{16}(w(\boldsymbol{z}_0) + w(\boldsymbol{z}_1) + w(\boldsymbol{z}_6) + w(\boldsymbol{z}_7)) + \frac{1}{8}(w(\boldsymbol{z}_2) + w(\boldsymbol{z}_5)) + \frac{1}{2}(w(\boldsymbol{z}_3) + w(\boldsymbol{z}_4)).$$

The above operation is done for every midpoints in the mesh. If the union of the two macro-elements (i.e., the butterfly) cannot be constructed because it would require cells outside the domain, then we set $\widetilde{w}(\boldsymbol{m}) = w_{0123}(\boldsymbol{m})$. If $\boldsymbol{m}$ is at the boundary of $D^n$, we assume that $\widetilde{w}(\boldsymbol{m})$ is given as a boundary condition.

If the geometric reference element is defined with $\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$, we first apply the butterfly algorithm to get $\widetilde{w}$ at the midpoints of $\mathcal{T}_h^n$. Then we consider the mesh $\mathcal{T}_{h/2}^n$ formed by subdividing every cell into four new cells by connecting the three midpoints, and estimate $\widetilde{w}$ at the new midpoints in $\mathcal{T}_{h/2}^n$ by apply again the butterfly algorithm. The process is summarized in the center and right panels in Fig. 1.

For further reference, we denote by $\mathcal{F}^{\text{btt}} : \boldsymbol{P}_d(\mathcal{T}_h) \to \boldsymbol{P}_d^{\text{geo}}(\mathcal{T}_h)$ the mapping described above, i.e., $\widetilde{\boldsymbol{w}} = \mathcal{F}^{\text{btt}}(\boldsymbol{w})$ for all $\boldsymbol{w} \in \boldsymbol{P}_d(\mathcal{T}_h)$.

### 5.3. Definition of the ALE velocity

Nonlinear conservation equations are notorious to have solutions that may develop shocks in finite time even if the initial data is smooth. For instance, in the case of scalar conservation equations, the method of characteristics with velocity $\mathbb{f}'(u)$ gives a multi-valued solution when shocks develop, and for a generic initial data this happens in finite time. Since, for scalar conservation equations, using $\mathbb{f}'(u)$ for the ALE velocity is somewhat similar to solving (2.1) by using the method of characteristics, the algorithm (3.3)–(3.4)–(3.5) may breakdown in finite time. (This claim is substantiated by numerical tests and arguments reported in [24, §6.1].) The breakdown manifests itself by a time step that goes to zero as the time level approaches the time of formation of the first shock. One way to avoid this breakdown is to use an ALE velocity that is a smoothed version of the Lagrangian velocity.

Many techniques to construct smooth ALE velocities have been proposed in the literature and we refer the reader to the abundant literature on this topic. Since it is not the purpose of the present paper to design an optimal ALE velocity, we content ourselves by adapting a technique proposed by Loubère et al. [36]. The method consists of blending the Lagrangian velocity with an averaging technique. The blending is done through a parameter that controls the local deformation of the cells.

Let us be more precise. We start by computing the Lagrangian velocity. For scalar conservation equations we set $\boldsymbol{w}_{\mathrm{Lg}}^n := \sum_{i\in\mathcal{V}} \mathbb{f}'(\mathsf{U}_i^n)\varphi_i^n$, and for the compressible Euler equations we set $\boldsymbol{w}_{\mathrm{Lg}}^n := \sum_{i\in\mathcal{V}} \frac{1}{\varrho_i^n}\mathbf{M}_i^n\varphi_i^n$. Recall that in both cases $\boldsymbol{w}_{\mathrm{Lg}}^n \in \boldsymbol{P}_d(\mathcal{T}_h^n)$. Then we reconstruct $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n \in \boldsymbol{P}_d^{\mathrm{geo}}(\mathcal{T}_h^n)$ as explained in §5.1 and §5.2, i.e., $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n := \mathcal{F}^{\mathrm{btt}}(\boldsymbol{w}_{\mathrm{Lg}}^n)$. With the Lagrangian velocity $\widetilde{\boldsymbol{w}}^n$ in hand, we compute the Lagrangian motion and an averaged version of the Lagrangian motion for all $i$ in $\mathcal{V}^{\mathrm{geo}}$ by setting

$$\boldsymbol{a}_{i,\mathrm{Lg}}^{n+1} := \boldsymbol{a}_i^n + \tau\,\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n(\boldsymbol{a}_i^n), \tag{5.1}$$

$$\boldsymbol{a}_{i,\mathrm{Sm}}^{n+1} := \frac{1}{\mathrm{card}(\mathcal{I}(i))-1}\sum_{j\in\mathcal{I}(i)\setminus\{i\}} \boldsymbol{a}_{j,\mathrm{Lg}}^{n+1}. \tag{5.2}$$

Finally, the actual ALE motion is defined for all $i$ in $\mathcal{V}^{\mathrm{geo}}$ by

$$\boldsymbol{a}_i^{n+1} := \omega_i^n\boldsymbol{a}_{i,\mathrm{Lg}}^{n+1} + (1-\omega_i^n)\boldsymbol{a}_{i,\mathrm{Sm}}^{n+1}, \tag{5.3}$$

where $\omega_i^n$ is a blending parameter that depends on the mesh deformation. In [24] we were using $\omega_i^n = 0.1$, but now we are going to use a slightly more sophisticated definition in order to be able to sustain highly deformed meshes. We now describe two possible ways to calculate $\omega_i^n$ that we have used in the numerical simulations reported at the end of the paper.

(i) **Method 1:** In every cell $K^n \in \mathcal{T}_h^n$, we measure the ratio between the area of the elements $K^n$ and that of the transformed element $K^{n+1}$ using (5.1) for all nodes in $K^n$ and set

$$\omega_K^n = \min\left(\frac{|K^{n+1}|}{|K^n|}, 1\right). \tag{5.4}$$

Then denoting $\mathcal{T}_i := \{K \in \mathcal{T}_h^n \mid \boldsymbol{a}_i^n \in K\}$ for all $i \in \mathcal{V}^{\mathrm{geo}}$, we define $\omega_i^n$ by averaging: $\omega_i^n = \mathrm{card}(\mathcal{T}_i)^{-1}\sum_{K\in\mathcal{T}_i}\omega_K^n$. Notice that the cardinality of $\mathcal{T}_i$ does not depend on $n$ since we assumed in §2.3 that the geometric connectivity is time-independent.

(ii) **Method 2:** The second method initially proposed in Loubère et al. [36] proceeds as follows. First we construct the Jacobian matrix of the mapping $\boldsymbol{x} \mapsto \boldsymbol{x} + \tau\,\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n(\boldsymbol{x})$. Since $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n$ is only piecewise continuous, the Jacobian matrix is discontinuous and its restriction to each cell $K \in \mathcal{T}_h^n$ is given by $\mathbb{F}_{|K}(\boldsymbol{x}) := \mathbb{I} + \tau\nabla\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n(\boldsymbol{x})$. Then we compute the right Cauchy-Green strain tensor $\mathbb{F}|_K^{\mathsf{T}}\mathbb{F}_{|K}$ and compute the two eigenvalues $\lambda_{1,K}(\boldsymbol{a}_i^n)$, $\lambda_{2,K}(\boldsymbol{a}_i^n)$ at all the geometric Lagrange nodes $\boldsymbol{a}_i^n \in K$, with the convention $\lambda_{1,K}(\boldsymbol{a}_i^n) \le \lambda_{2,K}(\boldsymbol{a}_i^n)$. Notice that if $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n$ generates a solid motion, then $\lambda_{1,K}(\boldsymbol{a}_i^n) = \lambda_{2,K}(\boldsymbol{a}_i^n)$. Then we define $\alpha_i^n := \mathrm{card}(\mathcal{T}_i)^{-1}\sum_{K\in\mathcal{T}_i}\lambda_{1,K}(\boldsymbol{a}_i^n)/\lambda_{2,K}(\boldsymbol{a}_i^n)$ and set

$$\omega_i^n := \phi(\alpha_i^n), \tag{5.5}$$

where $\phi \in C^0([0,1];[0,1])$ is a monotone function with $\phi(0) = 0$ and $\phi(1) = 1$. This function will be specified in each application.

In conclusion the ALE velocity $\widetilde{\boldsymbol{w}}^n$ is given by

$$\widetilde{\boldsymbol{w}}^n(\boldsymbol{a}_i^n) := \tau^{-1}(\boldsymbol{a}_i^{n+1} - \boldsymbol{a}_i^n), \tag{5.6}$$

where $\boldsymbol{a}_i^{n+1}$ is defined in (5.3). One deduces $\boldsymbol{w}^n$ from $\widetilde{\boldsymbol{w}}^n$ by using some interpolation operator. For instance, let us assume that $P(\mathcal{T}_h^n)$ is composed of continuous piecewise linear Lagrange elements (which is the case in all the numerical tests reported in the paper). Let $\{\boldsymbol{z}_i\}_{i\in\mathcal{Z}^{\mathrm{geo}}}$ be the set of the vertices of the geometric approximation (not to be confused with the Lagrange nodes $\{\boldsymbol{a}_i\}_{i\in\mathcal{V}^{\mathrm{geo}}}$). Let $\{\boldsymbol{z}_i\}_{i\in\mathcal{Z}}$ be the Lagrange nodes of $P(\mathcal{T}_h^n)$. Up to an isomorphism the two sets $\mathcal{Z}^{\mathrm{geo}}$ and $\mathcal{Z}$ are identical, and we use the same enumeration on both sets. The field $\boldsymbol{w}^n$ is then obtained from $\widetilde{\boldsymbol{w}}^n$ by setting the following for all $i \in \mathcal{Z}$:

$$\boldsymbol{w}^n(\boldsymbol{z}_i) = \widetilde{\boldsymbol{w}}^n(\boldsymbol{z}_i), \tag{5.7}$$

That is to say, $\boldsymbol{w}^n = \Pi_h(\widetilde{\boldsymbol{w}}^n)$, where $\Pi_h : C^0(\overline{D^n};\mathbb{R}^m) \to \boldsymbol{P}(\mathcal{T}_h)$ is the Lagrange interpolation operator.

## 6. Numerical tests

The objectives of this section is to illustrate the ALE method described in the present paper. This is done by estimating the accuracy of the method on scalar equations (linear and nonlinear) and on the compressible Euler equations.

**Table 1**
Swirling flow problem (6.1) at $t = 1$, CFL $= 1.0$, $\mathbb{P}_{2,2}$ motion. Convergence rate in $L^\infty$-norm.

| h | GMS-GV1 | | PG | | PG-CL | |
|---|---|---|---|---|---|---|
| 1/8 | 6.84E-01 | – | 1.42E-01 | – | 3.42E-01 | – |
| 1/16 | 3.92E-01 | 0.80 | 3.84E-02 | 1.89 | 3.84E-02 | 3.15 |
| 1/32 | 1.43E-01 | 1.45 | 9.70E-03 | 1.98 | 9.70E-03 | 1.98 |
| 1/64 | 4.19E-02 | 1.78 | 2.36E-03 | 2.04 | 2.36E-03 | 2.04 |
| 1/128 | 1.11E-02 | 1.92 | 5.93E-04 | 1.99 | 5.93E-04 | 1.99 |
| 1/256 | 2.83E-03 | 1.97 | 1.49E-04 | 1.99 | 1.49E-04 | 1.99 |

### 6.1. Numerical details

In all the numerical tests we use continuous Lagrange finite elements on triangles in two space dimensions. The one-dimensional shocktube problems reported in §6.5 and §6.6 are run in two space dimensions. The exact specification of the initial computational domain $D^0$ is given in each case. The reference polynomial space $\widehat{P}$ for the approximation space $P(\mathcal{T}_h)$ is $\mathbb{P}_{1,2}$. Depending on the situation, the reference geometric space $\widehat{P}^{\text{geo}}$ for the geometric space $P^{\text{geo}}(\mathcal{T}_h)$ is composed of polynomials of degree between 1 and 4, i.e., $\widehat{P}^{\text{geo}} = \mathbb{P}_{k,2}$ with $k \in \{1, 2, 3, 4\}$. In all the cases, the limiting is done exactly as explained in §4. The limiting is done on the density and the specific entropy. The limiting is done twice as explained in Algorithm 1 on page A3227 in [25]. The time stepping is done by using the third-order strong stability preserving Runge-Kutta, see Shu and Osher [43, Eq. (2.18)] and Kraaijevanger [33, Thm. 9.4]. The time step is recomputed at every time level by using the formula $\tau = \frac{1}{2}\text{CFL}\times \min_{i\in\mathcal{I}} \frac{\mathrm{m}_i}{|d_{ii}^{L,n}|}$ with $d_{ii}^{L,n} = -\sum_{ij} d_{ij}^{L,n}$ given in (3.10).

### 6.2. Analytical scalar-valued solution

We solve the linear transport equation $\partial_t u + \nabla\cdot(\boldsymbol{\beta} u) = 0$ in the domain $D^0 = (0,1)^2$ with the initial data $u_0(\boldsymbol{x}) = \sin(2\pi x_1)\sin(2\pi x_2)$ and the following incompressible velocity field:

$$\boldsymbol{\beta}(\boldsymbol{x}, t) = \cos(\pi t)\big(-\sin(2\pi x_2)\sin^2(\pi x_1), \sin(2\pi x_1)\sin^2(\pi x_2)\big)^{\mathsf{T}}. \tag{6.1}$$

This is an incompressible periodic swirling flow with period 2. Notice also that $u(\cdot, 1) = u_0(\cdot)$. The largest deformations occur at $t = \frac{1}{2}$ and $t = \frac{3}{2}$. The ALE velocity is chosen by setting $\mathbf{W}_i^n = \boldsymbol{\beta}(\boldsymbol{z}_i^n, t^n)$, for all $i \in \mathcal{Z} = \mathcal{V}$, i.e., $\boldsymbol{w}_h^n$ is the Lagrange interpolant of $\boldsymbol{\beta}$ in $\boldsymbol{P}_d(\mathcal{T}_h^n)$, and $\widetilde{\mathbf{W}}_i^n = \boldsymbol{\beta}(\boldsymbol{a}_i^n, t^n)$, for all $i \in \mathcal{V}^{\text{geo}}$, i.e., $\widetilde{\boldsymbol{w}}_h^n$ is the Lagrange interpolant of $\boldsymbol{\beta}$ in $\boldsymbol{P}^{\text{geo}}(\mathcal{T}_h^n)$. Notice that there is no issue with boundary condition since $\boldsymbol{\beta}\cdot\boldsymbol{n}_{|\partial D^0} = 0$. Notice also that here the mesh is not smoothed; the mesh motion is purely Lagrangian.

#### 6.2.1. Convergence tests

In this test the geometric reference polynomial space is $\widehat{P}^{\text{geo}} = \mathbb{P}_{2,2}$. Three different series of tests are done to measure the accuracy of the proposed method. In the first series we use the GMS-GV1 viscosity defined in (3.10). Notice that the particular choice of the ALE velocity implies that $\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) = |(\boldsymbol{\beta}(\boldsymbol{z}_i^n, t^n) - \boldsymbol{\beta}(\boldsymbol{z}_j^n, t^n))\cdot\boldsymbol{n}_{ij}^n|$; hence the viscosity is second-order in space instead of being first-order. This phenomenon makes the GMS-GV1 method second-order accurate in space; this is one of the reasons that makes Lagrangian methods attractive. In the second method, which we refer to as PG (for pure Galerkin method), we set $d_{ij}^n = 0$, i.e., the viscosity is removed. The third method, which we refer to PG-CL (for pure Galerkin with convex limiting), is the PG method with limiting. Here we apply the convex limiting technique discussed in §4 with $\Psi_1(\mathsf{U}) = \mathsf{U}$ and $\Psi_2(\mathsf{U}) = -\mathsf{U}$. Using $\Psi_1$ in the limiting enforces the local minimum principle, and using $\Psi_2$ enforces the local maximum principle.

The computation are done with CFL $= 1.0$ up to time $t = 1$ on various meshes with mesh size $h = 2^{-k}$ for $k \in \{3, \dots, 8\}$. We report in Table 1 the relative error measured in the $L^\infty$-norm for the three methods: GMS-GV1; PG; PG-CL. As expected all the methods are second-order accurate. We also observe that, owing to the relaxation introduced in §4.4, the limiting does not affect the convergence rate in the $L^\infty$-norm. We have verified that using cubic polynomials for the geometric approximation, $\widehat{P}^{\text{geo}} = \mathbb{P}_{3,2}$, does not change these results. When the computations are done using $\widehat{P}^{\text{geo}} = \mathbb{P}_{1,2}$, we observe that the mesh entangles before the final time. This problem can be solved by smoothing the mesh (results not shown here).

#### 6.2.2. Mesh motion

In order to demonstrate the benefit of using high-order elements for the mesh motion, we now solve the above problem with a velocity field that creates larger deformations than in the previous test case. We take

$$\boldsymbol{\beta}(\boldsymbol{x}, t) = \cos(\tfrac{\pi t}{2})\big(-\sin(2\pi x_2)\sin^2(\pi x_1), \sin(2\pi x_1)\sin^2(\pi x_2)\big)^{\mathsf{T}}. \tag{6.2}$$

The period of this flow is 4 and $u(\cdot, 2) = u_0(\cdot)$. The largest deformations occur at $t = 1$ and $t = 3$. In this type of flows, the longer the period the larger the deformations at the quarter period. Our goal is to show that high-order elements can sustain such large mesh distortions without collapsing with a pure Lagrangian motion.

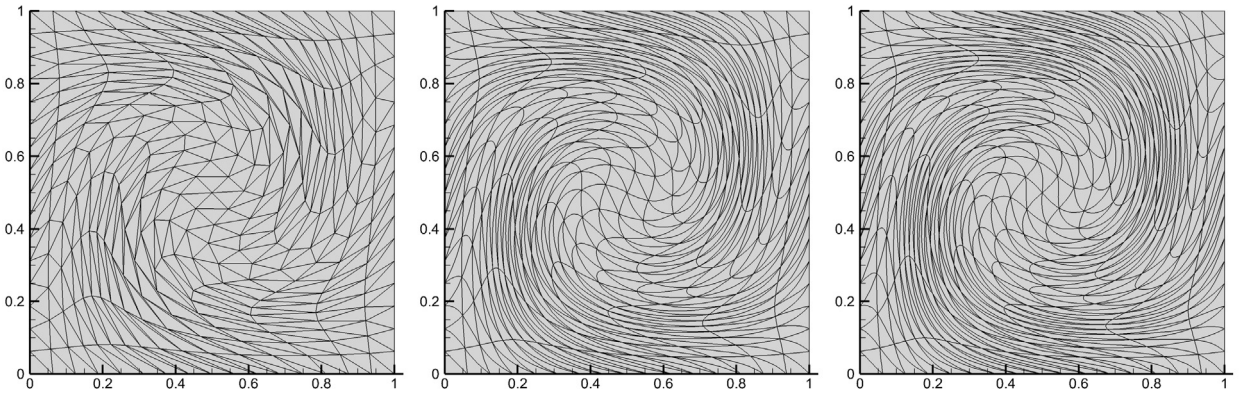

**Fig. 2.** Swirling flow problem with velocity (6.2). Lagrangian mesh motion with: $\mathbb{P}_{1,2}$ at collapse time $t=0.42$ (left); $\mathbb{P}_{2,2}$ at collapse time $t=0.74$ (middle); $\mathbb{P}_{3,2}$ at final time $t=1$ (right).

The computations are done with a mesh of mesh size $h=1/16$ until $t=1$. We test three different geometric approximations: $\widehat{P}^{\mathrm{geo}}=\mathbb{P}_{k,2}$, with $k\in\{1,2,3\}$. We show in Fig. 2 the meshes for the three different approximations considered. The simulations for the $\mathbb{P}_{1,2}$ and the $\mathbb{P}_{2,2}$ approximations cannot reach the final time $t=1$ without collapsing. The collapse time for the $\mathbb{P}_{1,2}$ approximation is $t=0.42$; the collapse time for the $\mathbb{P}_{2,2}$ approximation is $t=0.74$. The $\mathbb{P}_{3,2}$ approximation runs until the $t=1$ without any problems. (We recall that no smoothing is done on the meshes and the largest mesh deformations occur at $t=1$.)

### 6.3. Nonlinear scalar conservation equations

We now test the proposed method on nonlinear scalar conservation equations.

#### 6.3.1. Burgers' equation

We consider Burgers' equation in two space dimensions

$$\partial_t u+\nabla\cdot(\tfrac{1}{2}u^2\boldsymbol{\beta})=0,\quad u_0(\boldsymbol{x})=\mathbb{1}_S, \tag{6.3}$$

where $\boldsymbol{\beta}=(1,1)^{\mathsf{T}}$, and $\mathbb{1}_S$ denotes the indicator function of the set $S=(0,1)^2$. The solution to this problem is given in Guermond et al. [24, §6.2.2]. For completeness we recall the solution here. Let us set $\boldsymbol{x}:=(x_1,x_2)$. For any time $t>0$ if $x_2\le x_1$, then let $\alpha:=x_1-x_2$ and $\alpha_0:=1-\frac{t}{2}$. If $\alpha>1$, then $u(\boldsymbol{x},t)=0$.

$$\text{If } \alpha\le\alpha_0,\text{ then}\qquad u(\boldsymbol{x},t)=\begin{cases}\frac{x_2}{t} & \text{if } 0\le x_2<t\\ 1 & \text{if } t\le x_2<\frac{t}{2}+1-\alpha\\ 0 & \text{otherwise.}\end{cases} \tag{6.4}$$

$$\text{If } \alpha_0<\alpha\le 1,\text{ then}\quad u(\boldsymbol{x},t)=\begin{cases}\frac{x_2}{t} & \text{if } 0\le x_2<\sqrt{2t(1-\alpha)}\\ 0 & \text{otherwise.}\end{cases} \tag{6.5}$$

If $x_2>x_1$, then $u(x_1,x_2,t):=u(x_2,x_1,t)$.

The computations are done up to time $t=1$ with CFL $=0.1$ and the initial computational domain is $D^0=(-0.25,1.75)^2$. The boundary of $D^0$ does not move in the time interval $[0,1]$, i.e., $\partial D^0=\partial D^n$ for any $n\ge 0$ such that $t^n\in[0,1]$. The initial meshes are formed by $N\times N$ quadrilateral cells divided into two triangles, with $N\in\{8,16,32,64,128\}$. The Lagrangian velocity is obtained by setting $\boldsymbol{w}^n_{\mathrm{Lg}}:=\sum_{i\in\mathcal{V}}(\mathsf{U}^n_i,\mathsf{U}^n_i)^{\mathsf{T}}\varphi^n_i$, and $\widetilde{\boldsymbol{w}}^n_{\mathrm{Lg}}$ is obtained from $\boldsymbol{w}^n_{\mathrm{Lg}}$ by applying the butterfly algorithm, i.e., $\widetilde{\boldsymbol{w}}^n_{\mathrm{Lg}}=\mathcal{F}^{\mathrm{btt}}(\boldsymbol{w}^n_{\mathrm{Lg}})$. Then the smoothing technique called method 1 in §5.3 is used to smooth the mesh when $\min_{K\in\mathcal{T}^n_h}|K|\le\frac{1}{5}\min_{K\in\mathcal{T}^0_h}|K|$, which happens only near the end of the simulation.

For this test we use quadratic geometric elements, i.e., $\widehat{P}^{\mathrm{geo}}=\mathbb{P}_{2,2}$. We test four methods. The first one is the invariant domain preserving method GMS-GV1. The second one (referred to as SB in Table 2) is based on the smoothness-based viscosity introduced in Guermond and Popov [20, §4.3] (see also Jameson et al. [31, Eq. (12)], the second formula in the right column of page 1490 in Jameson [30], or Burman [8, Thm. 4.1]). The viscosity is defined by $d^{\mathrm{H},n}_{ij}=d^{\mathrm{L},n}_{ij}\max(\psi(\alpha^n_i),\psi(\alpha^n_j))$ where

$$\alpha^n_i:=\frac{\left|\sum_{j\in\mathcal{I}(i)}\beta_{ij}(\mathsf{U}^n_j-\mathsf{U}^n_i)\right|}{\max(\sum_{j\in\mathcal{I}(i)}|\beta_{ij}||\mathsf{U}^n_j-\mathsf{U}^n_i|,\epsilon_i)},\qquad \beta_{ij}:=\int_{D^n}\nabla\varphi^n_i\cdot\nabla\varphi^n_j\,\mathrm{d}x,\ i\ne j, \tag{6.6}$$

**Table 2**
Burgers' equation. $L^1$-norm of the error at $t=1$ with CFL $=0.1$: GMS-GV1 method (left); high-order SB method without limiting (center left); high-order SB method with convex limiting (center right); Right: Pure Galerkin with limiting.

| # dof | GMS-GV1 | | SB | | SB-CL | | PG-CL | |
|---|---|---|---|---|---|---|---|---|
| 81 | 6.37E01 | – | 5.25E-01 | – | 5.26E-01 | – | 4.54E-01 | – |
| 289 | 4.68E-01 | 0.44 | 3.03E-01 | 0.79 | 3.04E-01 | 0.79 | 2.78E-01 | 0.71 |
| 1089 | 2.82E-01 | 0.73 | 1.62E-01 | 0.90 | 1.62E-01 | 0.90 | 1.69E-01 | 0.72 |
| 4225 | 1.52E-01 | 0.89 | 9.39E-02 | 0.79 | 9.40E-02 | 0.79 | 1.15E-01 | 0.56 |
| 16641 | 7.42E-02 | 1.04 | 4.76E-02 | 0.98 | 4.76E-02 | 0.98 | 8.13E-02 | 0.50 |

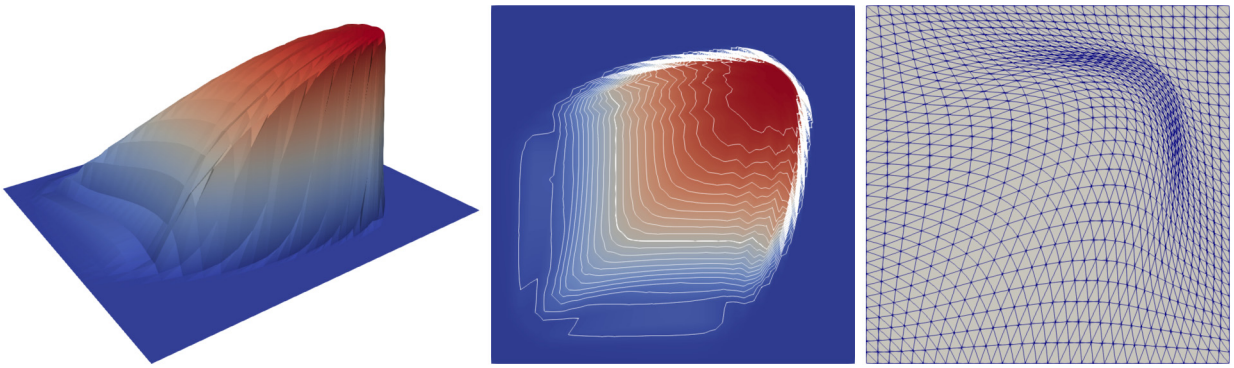

**Fig. 3.** Burger's equation solution and ALE mesh at $t=1$ with $16\times16$ grid. Smoothness-based high-order graph viscosity solution combined with the convex limiting technique.

where $\epsilon_i^n = 10^{-8}\max_{j\in\mathcal{I}(i)}|\mathsf{U}_j|$. The tests are done with $\psi(\alpha)=\alpha^6$. This second method is linearity preserving, but it may not be invariant domain preserving since the mesh distortions can make some of the coefficients $\beta_{ij}$ positive (see [20, §4.3] for details). The third method, called SB-CL in Table 2, has the same viscosity as the SB method but in addition the invariant domain property is enforced by convex limiting. The last (fourth) method (called PG-CL in Table 2) is defined by setting $d_{ij}^{\mathrm{H},n}=0$ and using convex limiting (PG-CL stands for pure Galerkin with convex limiting). We show in Table 2 the $L^1$-norm of the error at $t=1$ for the four methods on five successively refined meshes. As expected, the method SB-CL performs the best. We also observe/confirm, (as already mentioned many times in the literature) that using $d_{ij}^{\mathrm{H},n}=0$ and using convex limiting is not a good idea for nonlinear conservation equations, i.e., PG-CL method does not perform as well as SB-CL method.

Finally, we show in Fig. 3 the graph of the solution obtained on the $16\times16$ mesh at $t=1$ using the SB-CL method. We observe that the mesh is automatically refined in the shock region. Eventual entangling is avoided by smoothing the mesh as explained above.

*6.3.2. KPP rotating wave*

We solve in this section a nonlinear conservation equation with a nonconvex flux originally proposed in Kurganov et al. [34]:

$$\partial_t u+\nabla\cdot\boldsymbol{f}(u)=0,\quad \mathbf{f}(u)=(\sin u,\cos u)^{\mathsf{T}},\quad u_0(\boldsymbol{x})=\begin{cases}3.5\pi & \text{if } \|\boldsymbol{x}\|<1,\\ 0.25\pi & \text{otherwise.}\end{cases}\tag{6.7}$$

It is a challenging test for many high order numerical schemes because the solution has a two-dimensional composite wave structure. Many high-order methods produce solutions for this test that converge to weak solutions that are not the entropy solution. As shown in Ern and Guermond [15], this is specially the case of methods that use linear stabilization techniques.

The initial computational domain is $D^0=(-2.5,1.5)\times(-2.0,2.5)$. The initial mesh is composed of $128\times128$ quadrilateral cells divided into two triangles. The Lagrangian velocity is obtained by setting $\boldsymbol{w}_{\mathrm{Lg}}^n:=\sum_{i\in\mathcal{V}}\boldsymbol{f}'(\mathsf{U}_i^n)\varphi_i^n$, with $\boldsymbol{f}'(u)=(\cos u,-\sin u)$. The field $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n$ is obtained from $\boldsymbol{w}_{\mathrm{Lg}}^n$ by applying the butterfly algorithm, i.e., $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n=\mathcal{F}^{\mathrm{btt}}(\boldsymbol{w}_{\mathrm{Lg}}^n)$. Then the smoothing technique called method 1 in §5.3 is used to smooth the mesh when $\min_{K\in\mathcal{T}_h^n}|K|\le\frac{1}{5}\min_{K\in\mathcal{T}_h^0}|K|$. We use quadratic geometric elements for this test, i.e., $\widehat{P}^{\mathrm{geo}}=\mathbb{P}_{2,2}$. The computation is done up to time $t=1$ with CFL $=0.1$. We show in Fig. 4 the mesh and the solution obtained with the method SB-CL described in §6.3.1 (i.e., convex limiting and the smoothness-based viscosity with the little modification consisting of setting $\psi(\mathsf{U}_i^n)=\psi(\mathsf{U}_j^n)=1$ if $\boldsymbol{f}(u)\cdot\boldsymbol{n}_{ij}$ is not convex on the interval $[\min(\mathsf{U}_i^n,\mathsf{U}_j^n),\max(\mathsf{U}_i^n,\mathsf{U}_j^n)]$; see [20, §6.5] or [34, §4] for the details). The final mesh is shown in the left panel of the figure; the solution is shown in the right panel. The method performs as expected.

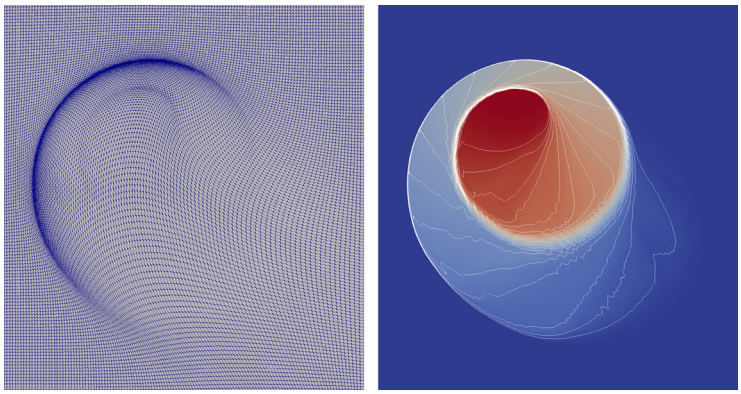

**Fig. 4.** KPP problem (6.7): Final mesh (left); Iso-contours at the solution (right).

### 6.4. Euler equations

In this section we present the results for the compressible Euler equations with the ideal gas equation of state $p(\boldsymbol{u}) = (\gamma-1)\rho e(\boldsymbol{u})$. We use the notation $e(\boldsymbol{u}) := E - \frac{1}{2}\rho^{-1}\boldsymbol{m}^2$, where we recall that $\boldsymbol{u} = (\rho, \boldsymbol{m}, E)^\intercal$ is the vector of the conserved variables. We illustrate the behavior of the proposed method on a series of traditional benchmark problems.

#### 6.4.1. Numerical details

When the exact solution of a specific test case is known, we compute the following relative error indicators for $q \in \{1, 2, \infty\}$.

$$\delta^q(t) := \frac{\|\rho_h(t) - \rho(t)\|_{L^q(D)}}{\|\rho(t)\|_{L^q(D)}} + \frac{\|\mathbf{m}_h(t) - \mathbf{m}(t)\|_{L^q(D)}}{\|\mathbf{m}(t)\|_{L^q(D)}} + \frac{\|E_h(t) - E(t)\|_{L^q(D)}}{\|E(t)\|_{L^q(D)}}, \tag{6.8}$$

The above norm are estimated by using a Gaussian quadrature rule of order 8 with 16 points.

The convex limiting is done as explained in §4. First, limiting is done on the density with the functionals $\Psi_1^{i,n}(\boldsymbol{u}) = \rho - \varrho_i^{n,\min}$ and $\Psi_2^{i,n}(\boldsymbol{u}) = -\rho - (-\varrho_i^{n,\max})$ with the notation $\mathbf{U}_i^n = (\varrho_i^n, \mathbf{M}_i^n, \mathsf{E}_i^n)$, $\varrho_i^{n,\min} = \min_{j\in\mathcal{I}(i)} \overline{\varrho}_{ij}^n$, and $-\varrho_i^{n,\max} = \min_{j\in\mathcal{I}(i)} -\overline{\varrho}_{ij}^n$. The quantities $\varrho_i^{n,\min}$ and $\varrho_i^{n,\min}$ are relaxed by using (4.10) as explained in §4.4. Enforcing $\Psi_1^{i,n}(\mathbf{U}_i^{n+1}) \geq 0$ gives the local minimum principle $\varrho_i^{n+1} \geq \varrho_i^{n,\min}$ and enforcing $\Psi_2^{i,n}(\mathbf{U}_i^{n+1}) \geq 0$ gives the local maximum principle $\varrho_i^{n+1} \leq \varrho_i^{n,\max}$. Then, limiting on the specific entropy is done with the concave functional $\Psi_3^{i,n}(\boldsymbol{u}) = \rho e(\boldsymbol{u}) - \mathsf{s}_i^{n,\min}\rho^\gamma$ with the notation $\mathsf{s}_i^{n,\min} = \min_{j\in\mathcal{I}(i)} \overline{\varrho}_{ij}^n e(\overline{\mathbf{U}_{ij}}^n)/(\overline{\varrho_{ij}}^n)^\gamma$. Enforcing $\Psi_3^{i,n}(\mathbf{U}_i^{n+1}) \geq 0$ gives the local minimum principle on the specific entropy $\varrho_i^{n+1} e(\mathbf{U}_i^{n+1})/(\varrho_i^{n+1})^\gamma \geq \mathsf{s}_i^{n,\min}$. The quantity $\mathsf{s}_i^{n,\min}$ is relaxed as explained in [25, §4.7.2].

The Lagrangian velocity is obtained by setting $\boldsymbol{w}_{\mathrm{Lg}}^n := \sum_{i\in\mathcal{V}} \frac{\mathbf{M}_i^n}{\varrho_i^n}\varphi_i^n$ and we compute $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n$ by applying the butterfly algorithm as explained in §5.2, i.e., $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n = \mathcal{F}^{\mathrm{btt}}(\boldsymbol{w}_{\mathrm{Lg}}^n)$.

#### 6.4.2. 2D isentropic vortex

The first test case is the isentropic vortex problem. The exact solution is isentropic and is given by

$$\rho(\mathbf{x}, t) = (T_\infty + \delta T)^{1/(\gamma-1)}, \qquad \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\infty + \delta\mathbf{u}, \qquad p(\mathbf{x}, t) = \rho^\gamma. \tag{6.9}$$

The free-steam conditions we use are $\rho_\infty = p_\infty = T_\infty = 1$ and $\boldsymbol{u}_\infty = (2, 0)^T$. The perturbations are

$$\delta\mathbf{u}(\mathbf{x}, t) = \frac{\beta}{2\pi} e^{\frac{1-r^2}{2}}(-\bar{x}_2, \bar{x}_1), \qquad \delta T(\mathbf{x}, t) = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2} e^{1-r^2}, \tag{6.10}$$

where $r = \|\mathbf{x} - \bar{\mathbf{x}}_c(t)\|$ is the Euclidean distance from the vortex center $\bar{\mathbf{x}}_c(t) := (x_1^0 + 2t, x_2^0)^\intercal$, $\beta = 5$ is a constant defining the vortex strength, and $\gamma = \frac{7}{5}$.

We set the initial computational domain to be $D^0 = (-5, 5)^2$. The first mesh consists of 20×20 squares divided in two triangles, then the mesh is refined uniformly five times. We set the density, the momentum and the energy to the free-stream values on the boundary of the computational domain at all times. We use the GMS-EV method with the entropy commutator computed with the generalized entropy $\eta(\boldsymbol{u}) = p^{\frac{1}{\gamma}}$ to set $d_{ij}^{\mathrm{H},n}$, see §3.4. Both the density and the specific entropy are limited.

To illustrate again the importance of using high-order approximation for the geometry, we perform two series of computations. In the first series we use $\widehat{P}^{\mathrm{geo}} = \mathbb{P}_{1,2}$ and set $\widetilde{\boldsymbol{w}}_{\mathrm{Lg}}^n = \boldsymbol{w}_{\mathrm{Lg}}^n$. In the second series with use $\widehat{P}^{\mathrm{geo}} = \mathbb{P}_{4,2}$ and compute

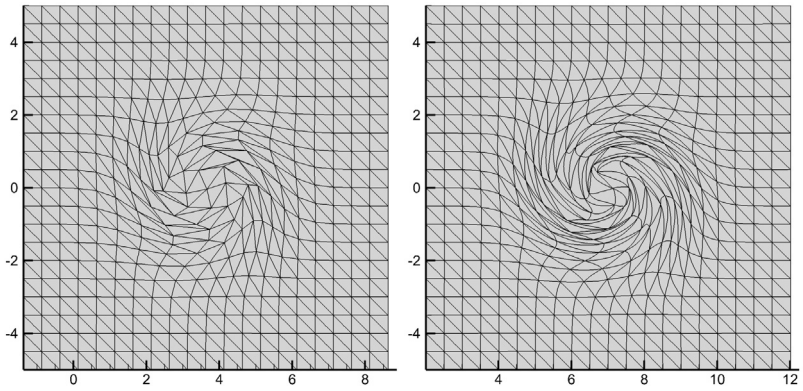


**Fig. 5.** Meshes obtained with pure Lagrangian motion. The meshes are shown at the instant before collapsing: $\widehat{P}^{\text{geo}} = \mathbb{P}_{1,2}$ at $t = 1.81$ (left) $\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$ at $t = 3.48$ (right).

**Table 3**
Isentropic vortex, Convergence test at $t = 2$ for the GMS-EV method, with limiting. CFL = 0.25.

| # dof | $\delta^1(t)$ | Rate | $\delta^2(t)$ | Rate | $\delta^\infty(t)$ | Rate |
|---|---|---|---|---|---|---|
| 441 | 1.62E-02 | – | 4.07E-02 | – | 3.65E-01 | – |
| 1681 | 4.97E-03 | 1.71 | 1.17E-02 | 1.80 | 9.64E-02 | 1.92 |
| 6561 | 1.33E-03 | 1.90 | 3.18E-03 | 1.88 | 3.03E-02 | 1.67 |
| 25921 | 3.38E-04 | 1.98 | 8.08E-04 | 1.98 | 8.11E-03 | 1.90 |
| 103041 | 8.49E-05 | 2.00 | 2.02E-04 | 2.00 | 2.10E-03 | 1.95 |

$\widetilde{\boldsymbol{w}}^n_{\text{Lg}}$ by applying twice the butterfly algorithm as explained in §5.2, i.e., $\widetilde{\boldsymbol{w}}^n_{\text{Lg}} = \mathcal{F}^{\text{btt}}(\boldsymbol{w}^n_{\text{Lg}})$. No smoothing is applied. Since the swirling always goes in the same direction and there is no smoothing, the meshes entangle at some point of time no matter what the level of refinement is or what the polynomial degree of the geometric approximation is. We show in Fig. 5 the deformation of the 20×20 mesh in the two cases right before the meshes entangle ($\widehat{P}^{\text{geo}} = \mathbb{P}_{1,2}$ in the left panel and $\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$ in the right panel). The collapse time for the linear mesh ($\widehat{P}^{\text{geo}} = \mathbb{P}_{1,2}$) mesh is $t = 1.81$. The collapse time for the other mesh ($\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$) is $t = 3.48$. In both cases we use CFL = 0.25. This experiment confirms again that the $\mathbb{P}_{4,2}$-mesh can sustain a lot more deformations than the $\mathbb{P}_{1,2}$-mesh.

In Table 3 we show the errors obtained with the GMS-EV approximation with limiting at $t = 2$. The CFL number is 0.25. We observe second-order accuracy in all the norms. There is no clipping effect on the $L^\infty$-norm.

### 6.5. Sod, Lax and Leblanc shocktubes

In this section we compute the solution of three standard one-dimensional Riemann problems usually referred to in the literature as Sod, Lax, and Leblanc shocktube problems. Since convergence tables for these tests are rarely given in the literature, we give in Appendix B all the details to compute the exact solutions to these problems so that readers without access to a Riemann solver can estimate the exact solution up to fifteen digits of accuracy.

The initial computational domain is $D^0 = (0, 1)\times(0, 0.4)$. Since the solution is one-dimensional, the convergence test are done on five meshes with refinements done in the $x$-direction only. The initial mesh is composed of 160×4 squares divided into 2 triangles. Because no wave reaches the left and right boundaries in the time interval considered, Dirichlet boundary conditions are enforced on the left and right boundaries of the domain. No boundary conditions are enforced on the upper and lower sides of the computational domain. The mesh motion is done with quadratic polynomials: $\widehat{P}^{\text{geo}} = \mathbb{P}_{2,2}$. Smoothing of the mesh is done with Method 1 for the Sod and Lax shocktubes and with Method 2 with $\phi(\alpha) = 1 - (1 - \alpha)^{1/3}$ for the Leblanc shocktube.

The errors are computed at $t = 0.15$ with CFL = 0.5 for the Lax problem, $t = 0.225$ with CFL = 0.25 for the Sod problem, and $t = 2/3$ with CFL = 0.2 for the Leblanc problem. The convergence results on the $\delta^1(t)$ error indicator are shown in Table 4. The convergence orders are close to 1, which is the optimal order, and the results are comparable with what is usually reported in the literature.

### 6.6. Woodward Colella blast wave problem

We now consider the so-called Woodward Colella blast wave problem [47]. Here again we use $\gamma = \frac{7}{5}$. This test involves multiple interactions of nonlinear waves, strong shocks and contact discontinuities. The initial condition consists of three constant states at rest in a one-dimensional box of length 1. The three states are $\mathbf{U}_L = (1, \mathbf{0}, 2.5 \times 10^3)^{\mathsf{T}}$, $\mathbf{U}_M = (1, \mathbf{0}, 2.5 \times 10^{-2})^{\mathsf{T}}$, and $\mathbf{U}_R = (1, \mathbf{0}, 2.5 \times 10^2)^{\mathsf{T}}$. The initial condition is

**Table 4**
Convergence tests for the shocktube problems. GMS-EV scheme with convex limiting, relaxing the bounds for the density and the entropy.

| | Lax | | Leblanc | | Sod | |
|---|---|---|---|---|---|---|
| # dof | $\delta^1(t)$ | rate | $\delta^1(t)$ | rate | $\delta^1(t)$ | rate |
| 805 | 6.45E-02 | – | 7.91E-02 | – | 5.67E-02 | – |
| 1605 | 3.40E-02 | 0.93 | 4.58E-02 | 0.79 | 3.11E-02 | 0.87 |
| 3205 | 1.82E-02 | 0.90 | 2.74E-02 | 0.74 | 1.79E-02 | 0.79 |
| 6405 | 9.19E-03 | 0.99 | 1.55E-02 | 0.82 | 8.83E-03 | 1.02 |
| 12805 | 4.77E-03 | 0.95 | 8.23E-03 | 0.91 | 4.42E-03 | 1.00 |

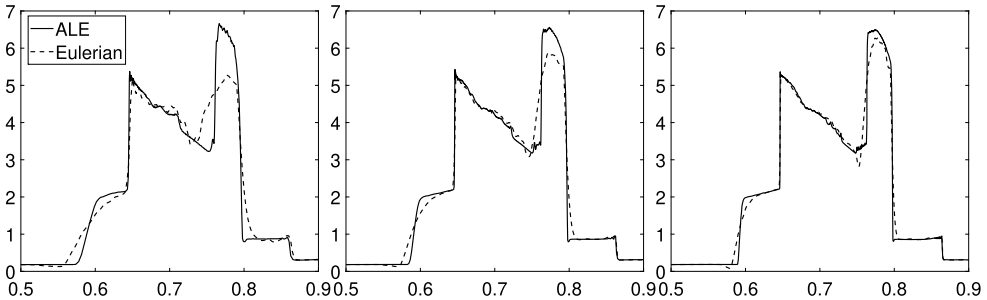


**Fig. 6.** Blast wave. Density along the center line $y = 0.1$ at $t = 0.038$, $CLF = 0.4$. ALE vs. Eulerian approximation: $I = 400$ (left); $I = 800$ (center); $I = 1600$ (right).

$$\mathbf{U}^{\mathbf{0}}(x) = \begin{cases} \mathbf{U}_L, & \text{if } 0 < x < 0.1, \\ \mathbf{U}_M, & \text{if } 0.1 < x < 0.9, \\ \mathbf{U}_R, & \text{if } 0.9 < x < 1. \end{cases} \tag{6.11}$$

The initial computational domain is $D^0 = (0,1)\times(0,0.2)$. The initial mesh is uniform and consists on $200\times5$ squares divided into 2 triangles. The mesh refinement is done is the $x$-direction only. The simulations are run from $t = 0$ to $t = 0.038$. We solve the problem with the GMS-EV scheme. Limiting is done on the density and the specific entropy with relaxation on the bounds. The mesh motion is done with quadratic polynomials: $\widehat{P}^{\text{geo}} = \mathbb{P}_{2,2}$. The mesh smoothing is done with Method 2 with $\phi(\alpha) = 1 - (1-\alpha)^{1/3}$.

We compare in Fig. 6 the density at the final time obtained with the proposed ALE method and with the Eulerian approximation ($\widetilde{\boldsymbol{w}}^n = \mathbf{0}$ and $\boldsymbol{w}^n = \mathbf{0}$). We observe that, with the same number of degrees of freedom, the contact discontinuity is better approximated with the ALE method than with the Eulerian one.

### 6.7. Noh problem

The next test case we consider is the so-called Noh problem, see Noh [37] and Caramana et al. [9, §5]. Here we take $\gamma = \frac{5}{3}$. The initial data is

$$\rho_0(\boldsymbol{x}) = 1.0, \quad \boldsymbol{u}_0(\boldsymbol{x}) = -\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, \quad p_0(\boldsymbol{x}) = 10^{-15}. \tag{6.12}$$

The solution to this problem is a shock wave propagating radially outwards with constant speed given by

$$\boldsymbol{u}(\boldsymbol{x},t) = \begin{cases} \left(16, -16\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|}, 8\right)^{\mathsf{T}} & \text{if } \|\boldsymbol{x}\| \le \frac{t}{3}, \\ \left(1 + \frac{t}{\|\boldsymbol{x}\|}, \mathbf{0}, \frac{1}{2}(1 + \frac{t}{\|\boldsymbol{x}\|})\right)^{\mathsf{T}} & \text{if } \frac{t}{3} < \|\boldsymbol{x}\|. \end{cases} \tag{6.13}$$

The computational domain at the initial time is $D^0 = (-1,1)^2$. Dirichlet conditions are enforced at the boundary of the domain. The ALE velocity at the boundary of the computational domain is set to be equal to the fluid velocity, i.e., the boundary moves inwards in the radial direction with speed 1. The final time is chosen to be $t = 0.6$ in order to avoid that the shockwave collides with the moving boundary of the computational domain. The mesh and the shock collide at $t = \frac{3}{4}$ since the shock moves radially outwards with speed $\frac{1}{3}$. We approximate the ALE velocity with Lagrange finite elements of degree 2, $\widehat{P}^{\text{geo}} = \mathbb{P}_{2,2}$, and we use Method 2 with $\phi(\alpha) = 1 - (1-\alpha)^{1/3}$ to smooth the mesh.

We show in Table 5 the $L^1$-error indicators $\delta^1$ obtained with the low-order method GMS-GV1, the unlimited GMS-EV scheme, and GMS-EV scheme with convex limiting on the density and the specific entropy. The initial meshes are composed of $N\times N$ square cells divided into two triangles. The computations are done with $N \in \{30, 60, 120, 240\}$. One observes that the errors obtained with the high-order methods are smaller than those with the low-order one. Furthermore, in addition

**Table 5**
Noh problem, convergence test, $t = 0.6$, CFL $= 0.4$.

| | GMS-GV1 | | GMS-EV | | GMS-EV limited | |
|---|---|---|---|---|---|---|
| # dof | $\delta^1(t)$ | rate | $\delta^1(t)$ | rate | $\delta^1(t)$ | rate |
| 961 | 8.52E-01 | – | 2.35E-01 | – | 2.89E-01 | – |
| 3721 | 5.48E-01 | 0.64 | 1.26E-01 | 0.89 | 1.51E-01 | 0.94 |
| 14641 | 3.14E-01 | 0.80 | 7.19E-02 | 0.81 | 8.07E-02 | 0.90 |
| 58081 | 1.47E-01 | 1.09 | 4.16E-02 | 0.79 | 4.18E-02 | 0.95 |
| 231361 | 7.72E-02 | 0.93 | 2.40E-02 | 0.79 | 2.18E-02 | 0.94 |

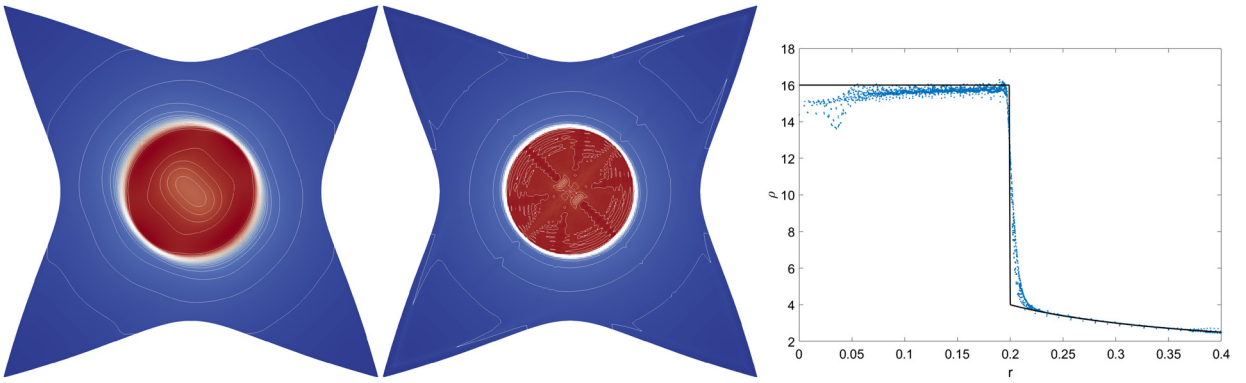


**Fig. 7.** Contours of the density field on a uniform mesh, $h = 1/120$: GMS-GV1 method, maximum of density is 14.59 (left); GMS-EV method with convex limiting on the density and the entropy, maximum of density is 16.29 (center and right); Scatter plot of the density at the vertices of the mesh as a function of the radial distance (right).

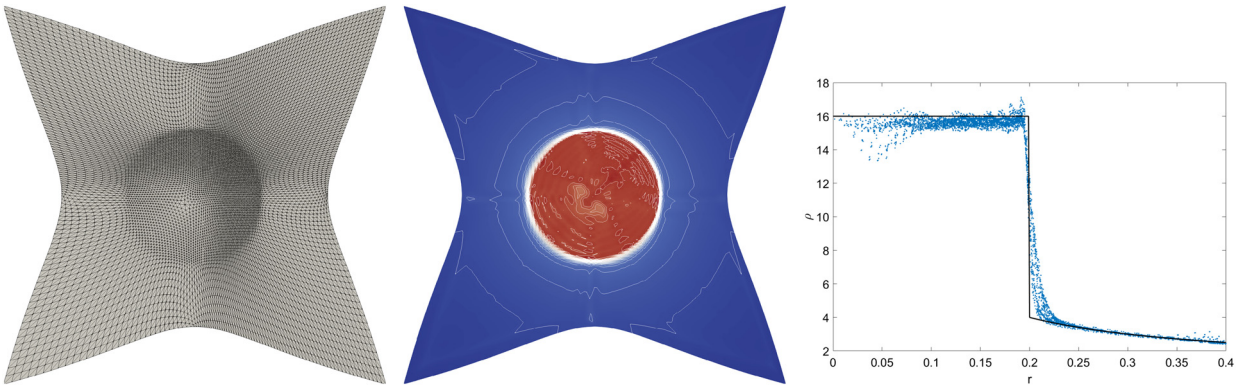


**Fig. 8.** Noh problem, $t = 0.6$, nonuniform mesh. Final mesh (left); Density (center) obtained with the GMS-EV method with convex limiting on density and specific entropy. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

of preserving the bounds, the GMS-EV scheme with convex limiting has a better rate of convergence than the unlimited method. The rate of convergence is around 0.95, which is close to the optimal rate of 1.

We show in Fig. 7 the density field obtained with GMS-GV1 (left panel) and with the GMS-EV scheme with convex limiting (center and right panel) the $t = 0.6$ on the uniform corresponding to $h = \frac{1}{120}$. The right panel shows a scatter plot of the density at the vertices of the mesh as a function of the radial distance (GMS-EV scheme only).

We show in Fig. 8 a solution obtained with the GMS-EV scheme with convex limiting on a nonuniform mesh suggested in [11, §8.4]. This test is meant to illustrate the robustness of the method with respect to the mesh regularity. The initial mesh is divided in four quadrants: the bottom left quadrant is composed of $32\times32$ squares; the top left is composed of $32\times64$ squares; the top right is composed of $64\times64$; and the bottom right is composed of $64\times32$ squares. Each square cell is divided into two triangles. We observe that the method behaves well and does not develop any hour-glass-like instability. A scatter plot of the density at the vertices of the mesh as a function of the radial distance is shown in the rightmost panel.

The results shown in Fig. 7 and Fig. 8 compare quite well with Dobrev et al. [11, Figs. 8.7], Guermond et al. [23, Fig. 15], and Boscheri et al. [6, Fig. 4].

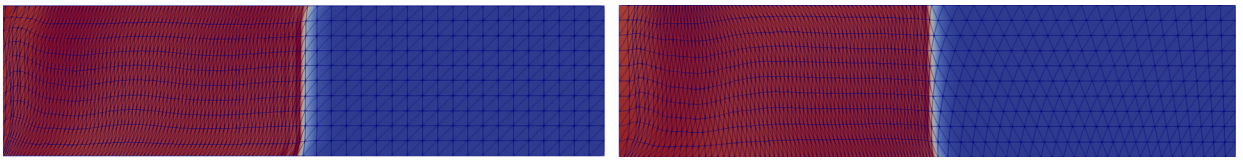

**Fig. 9.** Saltzman problem. Density field and mesh at $t = 0.6$. GMS-EV plus convex limiting with pure Lagrangian motion. Solution obtained on the uniform mesh (left) and on the modified mesh using the mapping $\boldsymbol{T}^{\text{ptb}}$(right). The length of the domain, initially equal to 1, is now equal to 0.4.

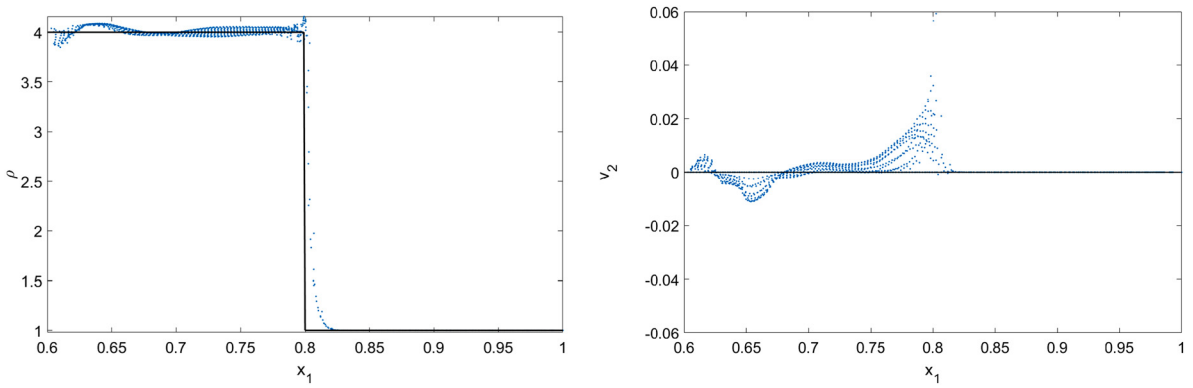


**Fig. 10.** Saltzman problem. The GMS-EV plus convex limiting solution is compared against the exact solution (solid line). The solutions are plotted as functions of the $x_1$ coordinate. On the left the density and on the right the vertical component of the velocity obtained on the modified mesh with pure Lagrangian motion.

### 6.8. Saltzman problem

We now consider the so-called Saltzman problem (see Dukowicz and Meltz [13, §2.2]). It is a shocktube problem in a two-dimensional rectangular box, $(0,1)\times(0,0.1)$ with $\gamma = \frac{5}{3}$. A strong shock wave is created by a piston moving to the right along the horizontal axis. The solution to this problem is obtained by solving the Riemann problem with $\mathbf{V}_L = (1, 2, (\gamma - 1)10^{-4})^{\mathsf{T}}$, $\mathbf{V}_R = (1, 0, (\gamma - 1)10^{-4})^{\mathsf{T}}$, with the convention that here $\boldsymbol{V} = (\rho, \boldsymbol{v}, e)$, where $\boldsymbol{v}$ is the velocity and $e$ the specific internal energy. The 3-wave of this Riemann problem is a shock moving with speed $\lambda_3 = 1.3334166614589844$. The 2-wave is a contact discontinuity moving with speed $v^* = 1$. The 1-wave is a shock moving with speed $\lambda_1^- = 0.66658333854101581$, Here we have $\rho_L^* = \rho_R^* = 3.9992502342988532$, $p^* = 1.3334833281256511$. The Saltzman problem consists of simulating only half of this problem by replacing the contact discontinuity with a wall moving at speed 1. The wall is initially located at $\{x = 0\}$. The initial data is $\boldsymbol{u}_0 = (1, \mathbf{0}, 10^{-4})^{\mathsf{T}}$, and the boundary condition is $\boldsymbol{v} = \boldsymbol{e}_x$ at the boundary $\{x = t\}$ and $\boldsymbol{v}\cdot\boldsymbol{n} = 0$ otherwise.

The mesh for the initial computational domain $D^0 = (0,1)\times(0,0.1)$ is composed of $100\times10$ uniformly distributed square cells each divided into two triangles. The mesh is then transformed by using the mapping $\boldsymbol{T}^{\text{ptb}} : (x_1, x_2)^{\mathsf{T}} \mapsto (x_1 + (0.1 - x_2)\sin(\pi x_1), x_2)^{\mathsf{T}}$. The mapping $\boldsymbol{T}^{\text{ptb}}$ is applied to all the Lagrange nodes $\{\boldsymbol{a}_i^0\}_{i\in\mathcal{V}^{\text{geo}}}$. This transformation is meant to bias the mesh in order to trigger hourglass instabilities. We use quadratic elements for the mesh motion: $\widehat{P}^{\text{geo}} = \mathbb{P}_{2,2}$. The motion is done without any smoothing; that is, $\omega_i^n = 0$ in (5.3), i.e., the motion is Lagrangian. The high-order viscosity $d_{ij}^{\text{H},n}$ is computed by using the entropy viscosity commutator explained in §3.4 using the entropy $\eta(\boldsymbol{u}) = \rho s$. Limiting on the density and the specific entropy is applied as explained in §6.4.1.

We show in Fig. 9 the density and the mesh at $t = 0.6$ and in Fig. 10 the density and the vertical component of the velocity as functions of the coordinate $x_1$. We observe that the GMS-EV scheme is robust with respect to the bias introduced in the mesh. Except for some small oscillations near the boundaries, the numerical and the exact solutions are in good agreement. These results compare well with Boscheri et al. [7, Fig. 10] and Scovazzi [39, Fig. 2].

### 6.9. Rayleigh-Taylor instability

The last problem is the Rayleigh-Taylor instability. This instability occurs at the interface between two fluids of different densities with the lighter fluid on top of the heavier one. The fluids are initially at rest. The initial equilibrium is unstable under the action of the gravity force: any perturbation at the interface breaks the balance.

The computational domain is a fixed rectangular box $D^0 = (0, d/2) \times (0, 3d)$. In the simulations we set $d = 1/3$. The gravitational force is $\mathbf{g} = (0, -0.1x_2)^{\mathsf{T}}$. The heavy fluid has density $\rho_{\max} = 2$ and the light fluid has density $\rho_{\min} = 1$. The interface between the two fluids at $t = 0$ is $x_2 = \eta(x) = 0.5 - 0.1d\cos(2\pi x_1/d)$. The growth of the perturbation at the

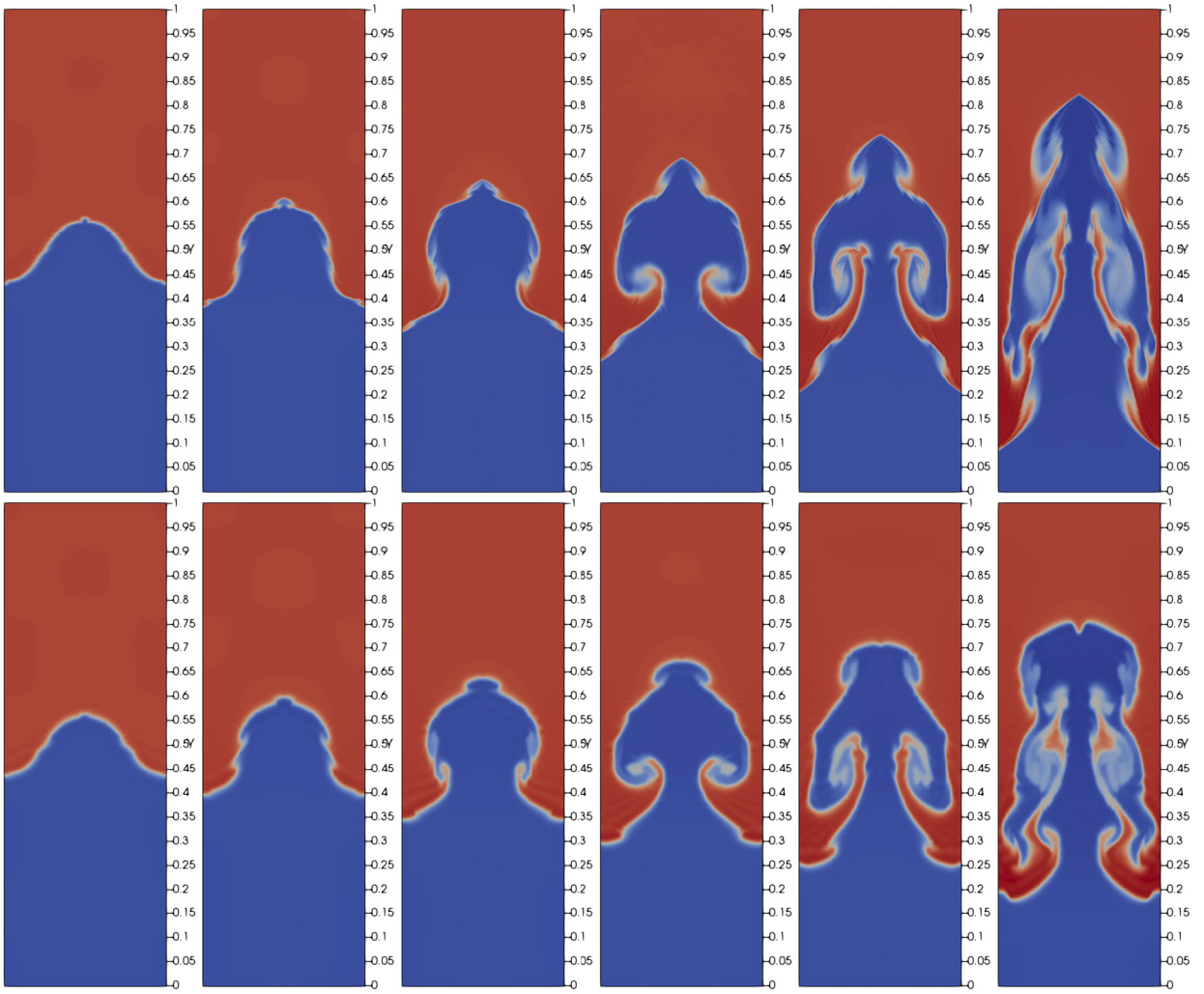

**Fig. 11.** Rayleigh Taylor instability. Density field at times $t = 2, 3, 4, 5, 6, 7.5$ (from left to right). ALE (top) and Eulerian (bottom). In both cases we use GMS-EV with convex limiting and relaxation on the entropy and the specific entropy; CFL = 0.6.

interface is exponential and the rate is $\exp(\alpha t)$ where $\alpha \propto \mathcal{A}$, and $\mathcal{A} = (\rho_{\max} - \rho_{\min})/(\rho_{\max} + \rho_{\min})$ is the Atwood number; here we take $\mathcal{A} = 1/3$. The initial density field is slightly regularized by setting

$$\rho_0(x) := 2 + \tanh\left(\frac{y - \eta(x)}{0.01d}\right). \tag{6.14}$$

The initial pressure is hydrostatic. The slip boundary condition is enforced on the four walls of $D^0$. The mesh is composed of 50×200 squares divided into two triangles. The ALE velocity is computed with $\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$ Lagrange finite elements. The simulations are run until $t = 7.5$ with CFL = 0.6. The mesh undergoes very large deformations in the time interval $[0, t]$. The mesh is smoothed with Method 2 using $\Phi(\alpha) = 1 - (1 - \alpha)^{\frac{1}{3}}$. We show in Fig. 11 the density field obtained with the GMS-EV scheme with limiting and relaxation on the density and the specific entropy. We compare the solution obtained with the ALE approach (top panels) with the solution obtained with the Eulerian approach (bottom panels).

Finally, in order to illustrate that the method is robust when using fourth-order polynomials to describe the mesh motion, i.e., $\widehat{P}^{\text{geo}} = \mathbb{P}_{4,2}$, we show in Fig. 12 the meshes and the amplitude of the velocity at the four times $t = 4, 5, 6, 7.5$. These pictures show that the mesh undergoes very large deformations in this test.

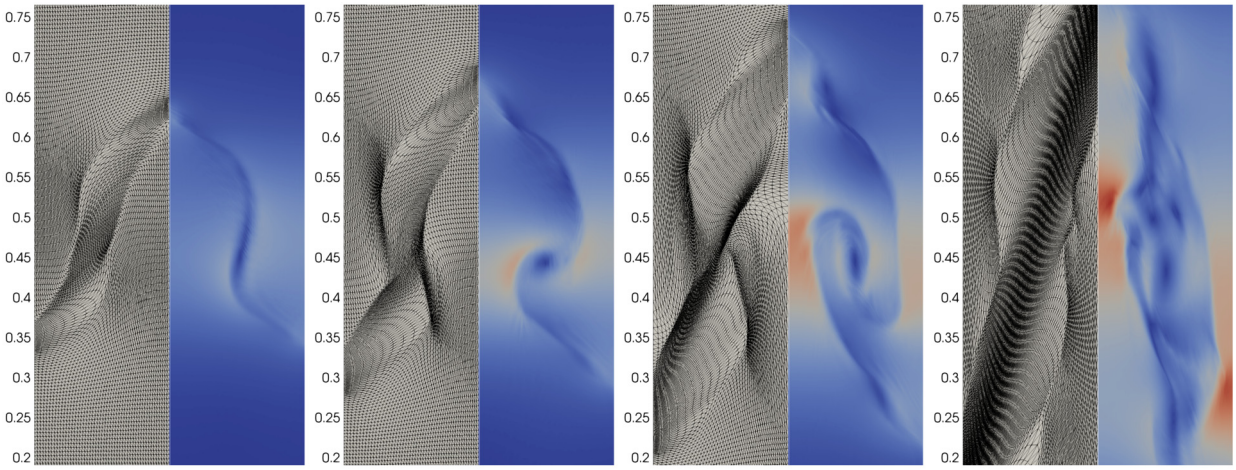

**Fig. 12.** Magnitude of the velocity field and mesh obtained with the ALE method with GMS-EV and limiting: $t = 4, 5, 6, 7.5$ (from left to right). Initial mesh is composed of $50 \times 200$ cells divided into two triangles; CFL $= 0.6$.

## Appendix A. Technical material

**Proof of Lemma 2.1.** (1) Observe that

$$\sum_{1\le i\le m} (\partial_{u_i}\eta(\boldsymbol{u}))\partial_{u_j}(u_i\boldsymbol{W}_k) = \sum_{1\le i\le m} (\partial_{u_i}\eta)\delta_{ij}\boldsymbol{W}_k = \sum_{1\le i\le m} (\partial_{u_j}\eta(\boldsymbol{u})\boldsymbol{W}_k).$$

Hence the entropy flux associated with $\boldsymbol{u}\otimes\boldsymbol{W}$ is $\eta(\boldsymbol{u})\boldsymbol{W}$. This proves (1).
(2) Let $\boldsymbol{v}(x,t)$ be the entropy solution to the Riemann problem (2.4), and let $\boldsymbol{w}(x,t)$ be the entropy solution to the Riemann problem (2.2). Let $(\eta(\boldsymbol{u}), \boldsymbol{F}(\boldsymbol{u}))$ be an entropy pair for the flux $\mathbb{f}$. Then

$$\int_0^\infty\int_{\mathbb{R}} (-\eta(\boldsymbol{w}(x,t))\partial_t\phi(x,t) - \boldsymbol{n}\cdot\boldsymbol{F}(\boldsymbol{w}(x,t))\partial_x\phi(x,t))\,\mathrm{d}x\,\mathrm{d}t - \int_{\mathbb{R}} \eta(\boldsymbol{w}_0(x))\phi(x,0)\,\mathrm{d}x \le 0, \tag{A.1}$$

for all $\phi\in C_0^1([0,\infty);\mathbb{R}^+)$. Making the change of variable $x := x' + (\mathbf{W}\cdot\boldsymbol{n})t$, setting $\psi(x',t) := \phi(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t)$, and observing that $\partial_t\psi(x',t) = (\partial_t\phi)(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t) + (\mathbf{W}\cdot\boldsymbol{n})(\partial_x\phi)(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t)$, we infer that

$$\begin{aligned}\int_0^\infty\int_{\mathbb{R}} &(-\eta(\boldsymbol{w}(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t))\partial_t\psi(x',t) - (\boldsymbol{n}\cdot\boldsymbol{F}(\boldsymbol{w}(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t)) - \boldsymbol{w}(x'+(\mathbf{W}\cdot\boldsymbol{n})t,t)(\mathbf{W}\cdot\boldsymbol{n}))\partial_x\psi(x',t))\,\mathrm{d}x'\,\mathrm{d}t\\ &- \int_{\mathbb{R}} \eta(\boldsymbol{w}_0(x'+(\mathbf{W}\cdot\boldsymbol{n})t))\psi(x',0)\,\mathrm{d}x' \le 0,\end{aligned} \tag{A.2}$$

for all $\psi\in C_0^1([0,\infty);\mathbb{R}^+)$. Using (1), i.e., the entropy flux associated with $(\mathbb{f}(\boldsymbol{w}) - \boldsymbol{w}\otimes\boldsymbol{W})\boldsymbol{n}$ is $\boldsymbol{n}\cdot\boldsymbol{F}(\boldsymbol{w}) - (\boldsymbol{W}\cdot\boldsymbol{n})\boldsymbol{w}$, the above computation shows that $\boldsymbol{w}$ is an entropy solution to (2.2) iff $\boldsymbol{w}(x+(\mathbf{W}\cdot\boldsymbol{n})t,t)$ is an entropy solution to (2.4), thereby proving (2). (Recall that using in (A.1) the family of entropies $\eta_i^\pm(\boldsymbol{u}) = \pm u_i$ and associated entropy fluxes $\boldsymbol{F}_i^\pm(\boldsymbol{u}) = \pm\mathbb{f}_i(\boldsymbol{u})$, $i\in\{1:m\}$, one deduces from (A.1) that $\boldsymbol{w}(x,t)$ is a weak solution to (2.2); similarly, (A.2) implies that $\boldsymbol{w}(x+(\boldsymbol{W}\cdot\boldsymbol{n})t,t)$ is a weak solution to (2.4).)
(3) This is a direct consequence of (2).
(4) The solution $\boldsymbol{w}$ to the Riemann problem is self-similar, i.e., there exists $\psi$ such that $\boldsymbol{w}(x,t) = \boldsymbol{\psi}(\frac{x}{t})$. Hence $\boldsymbol{v}(x,t) = \boldsymbol{w}(x+\mathbf{W}\cdot\boldsymbol{n}t,t) = \boldsymbol{\psi}(\frac{x}{t}+\mathbf{W}\cdot\boldsymbol{n})$. This implies that if $\frac{x}{t}+\mathbf{W}\cdot\boldsymbol{n} \le \lambda_L(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ then $\boldsymbol{v}(x,t) = \boldsymbol{v}_L$ and if $\frac{x}{t}+\mathbf{W}\cdot\boldsymbol{n} \ge \lambda_R(\mathbb{f}\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ then $\boldsymbol{v}(x,t) = \boldsymbol{v}_R$; which completes the proof. □

## Appendix B. Sod, Lax, and Leblanc shocktubes

We give here all the details that are needed to compute the exact solution to the Sod, Lax, and Leblanc shocktubes so that any reader without access to a Riemann solver can estimate the exact solution up to fifteen digits of accuracy. Denoting by $\rho, v, p$ the density, velocity, and pressure, and recalling that the equation of state is the gamma law $p = (\gamma-1)\rho e$, we give in Table 6 the Riemann data for these three tests.

**Table 6**
Data for Sod, Lax and Leblanc shocktubes.

| | $\rho_L$ | $\rho_R$ | $v_L$ | $v_R$ | $p_L$ | $p_R$ | $\gamma$ |
|---|---|---|---|---|---|---|---|
| Sod | 1.0 | 0.125 | 0.0 | 0.0 | 1.0 | 0.1 | $\frac{7}{5}$ |
| Lax | 0.445 | 0.5 | 0.698 | 0.0 | 3.528 | 0.571 | $\frac{7}{5}$ |
| Leblanc | 1.0 | 0.001 | 0.0 | 0.0 | $\frac{2}{3}10^{-1}$ | $\frac{2}{3}10^{-10}$ | $\frac{5}{3}$ |

**Table 7**
Solution of the Sod, Lax and Leblanc shocktubes.

| | $\xi \le \lambda_1^-$ | $\lambda_1^- < \xi \le \lambda_1^+$ | $\lambda_1^+ < \xi \le v^*$ | $v^* < \xi \le \lambda_3$ | $\lambda_3 < \xi$ |
|---|---|---|---|---|---|
| $\rho$ | $\rho_L$ | $\rho(\xi)$ | $\rho_L^*$ | $\rho_R^*$ | $\rho_R$ |
| $v$ | $v_L$ | $v(\xi)$ | $v^*$ | $v^*$ | $v_R$ |
| $p$ | $p_L$ | $p(\xi)$ | $p^*$ | $p^*$ | $p_R$ |

**Table 8**
Star states for Sod, Lax and Leblanc shocktubes.

| | Sod | Lax | Leblanc |
|---|---|---|---|
| $\lambda_1^-$ | −1.183215956619923 | −2.6335650740600323 | $-\frac{1}{3}$ |
| $\lambda_1^+$ | −0.07027281256118334 | −1.6366974421005713 | 0.49578489518897934 |
| $v^*$ | 0.92745262004894991 | 1.5287230266328840 | 0.62183867139173454 |
| $\lambda_3$ | 1.7521557320301779 | 2.4793214809898405 | 0.82911836253346982 |
| $\rho_L^*$ | 0.4263194281784952 | 0.34456847418960945 | $5.4079335349316249\times10^{-2}$ |
| $\rho_R^*$ | 0.26557371170530708 | 1.3040845320261998 | $3.9999980604299963\times10^{-3}$ |
| $p^*$ | 0.3031301780506468 | 2.4660979192073564 | $5.1557792765096996\times10^{-4}$ |

The solutions to these three problems have the same structure: the left-wave (or 1-wave) is an expansion, the middle wave is a contact discontinuity, and the right-wave (or 3-wave) is a shock. Denoting $\xi = (x - x_0)/t$ the self-similar variable, where $x_0 \in \mathbb{R}$ is the location of the discontinuity at $t = 0$, we introduce the following functions describing the left expansion wave (i.e., the 1-wave):

$$\rho(\xi) = \rho_L \left[ \frac{2}{\gamma+1} + \frac{\gamma-1}{(\gamma+1)c_L}(v_L - \xi) \right]^{\frac{2}{\gamma-1}}, \tag{B.1a}$$

$$v(\xi) = \frac{2}{\gamma+1} \left[ c_L + \frac{\gamma-1}{2} v_L + \xi \right], \tag{B.1b}$$

$$p(\xi) = p_L \left[ \frac{2}{\gamma+1} + \frac{\gamma-1}{(\gamma+1)c_L}(v_L - \xi) \right]^{\frac{2\gamma}{\gamma-1}}, \tag{B.1c}$$

where $c_L = \sqrt{\gamma\, p_L/\rho_L}$ is the sound speed of the left state. Then the full self-similar solution is described in Table 7 with the parameters $\lambda_1^-$, $\lambda_1^+$, $v^*$, $\lambda_3$, $\rho_L^*$, $\rho_R^*$, and $p^*$ given in Table 8.

## Appendix C. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcp.2019.108927.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

[1] R. Abgrall, S. Tokareva, Staggered grid residual distribution scheme for Lagrangian hydrodynamics, SIAM J. Sci. Comput. 39 (5) (2017) A2317–A2344.
[2] R.W. Anderson, V.A. Dobrev, T.V. Kolev, R.N. Rieben, V.Z. Tomov, High-order multi-material ALE hydrodynamics, SIAM J. Sci. Comput. 40 (1) (2018) B32–B58.
[3] A. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, Int. J. Numer. Methods Fluids 56 (2007) 953–964.
[4] Y. Bazilevs, I. Akkerman, D.J. Benson, G. Scovazzi, M.J. Shashkov, Isogeometric analysis of Lagrangian hydrodynamics, J. Comput. Phys. 243 (2013) 224–243.
[5] J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works, Comput. Phys. 11 (1) (1973) 38–69, J. Comput. Phys. 135 (2) (1997) 170–186, With an introduction by Steven T. Zalesak, Commemoration of the 30th anniversary of J. Comput. Phys.
[6] W. Boscheri, R. Loubère, M. Dumbser, Direct arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws, J. Comput. Phys. 292 (2015) 56–87.

[7] W. Boscheri, M. Dumbser, R. Loubère, P.-H. Maire, A second-order cell-centered Lagrangian ADER-MOOD finite volume scheme on multidimensional unstructured meshes for hydrodynamics, J. Comput. Phys. 358 (2018) 103–129.
[8] E. Burman, On nonlinear artificial viscosity, discrete maximum principle and hyperbolic conservation laws, BIT 47 (4) (2007) 715–733.
[9] E. Caramana, M. Shashkov, P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1) (1998) 70–97.
[10] K.N. Chueh, C.C. Conley, J.A. Smoller, Positively invariant regions for systems of nonlinear diffusion equations, Indiana Univ. Math. J. 26 (2) (1977) 373–392.
[11] V. Dobrev, T. Kolev, R. Rieben, High-order curvilinear finite element methods for Lagrangian hydrodynamics, SIAM J. Sci. Comput. 34 (5) (2012) B606–B641.
[12] J. Donea, A. Huerta, J.-P. Ponthot, A. Rodríguez-Ferran, Arbitrary Lagrangian–Eulerian Methods, John Wiley & Sons, Ltd, 2004, pp. 413–437, chapter 14.
[13] J.K. Dukowicz, B.J. Meltz, Vorticity errors in multidimensional lagrangian codes, J. Comput. Phys. 99 (1) (1992) 115–134.
[14] N. Dyn, D. Levine, J.A. Gregory, A butterfly subdivision scheme for surface interpolation with tension control, ACM Trans. Graph. 9 (2) (Apr. 1990) 160–169.
[15] A. Ern, J.-L. Guermond, Weighting the edge stabilization, SIAM J. Numer. Anal. 51 (3) (2013) 1655–1677.
[16] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids, J. Comput. Phys. 174 (2) (2001) 669–694.
[17] H. Frid, Maps of convex sets and invariant regions for finite-difference systems of conservation laws, Arch. Ration. Mech. Anal. 160 (3) (2001) 245–269.
[18] J.-L. Guermond, B. Popov, Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations, J. Comput. Phys. 321 (2016) 908–926.
[19] J.-L. Guermond, B. Popov, Invariant domains and first-order continuous finite element approximation for hyperbolic systems, SIAM J. Numer. Anal. 54 (4) (2016) 2466–2489.
[20] J.-L. Guermond, B. Popov, Invariant domains and second-order continuous finite element approximation for scalar conservation equations, SIAM J. Numer. Anal. 55 (6) (2017) 3120–3146.
[21] J.-L. Guermond, B. Popov, Estimation from above of the maximum wave speed in the Riemann problem for the Euler equations and related problems, 2017, preprint.
[22] J.-L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for nonlinear conservation laws, J. Comput. Phys. 230 (11) (2011) 4248–4267.
[23] J.-L. Guermond, B. Popov, V. Tomov, Entropy-viscosity method for the single material Euler equations in Lagrangian frame, Comput. Methods Appl. Mech. Eng. 300 (2016) 402–426.
[24] J.-L. Guermond, B. Popov, L. Saavedra, Y. Yang, Invariant domains preserving arbitrary Lagrangian Eulerian approximation of hyperbolic systems with continuous finite elements, SIAM J. Sci. Comput. 39 (2) (2017) A385–A414.
[25] J.-L. Guermond, M. Nazarov, B. Popov, I. Tomas, Second-order invariant domain preserving approximation of the Euler equations using convex limiting, SIAM J. Sci. Comput. 40 (5) (2018) A3211–A3239.
[26] J.-L. Guermond, M. Quezada de Luna, B. Popov, C.E. Kees, M.W. Farthing, Well-balanced second-order finite element approximation of the shallow water equations with friction, SIAM J. Sci. Comput. 40 (6) (2018) A3873–A3901.
[27] J.-L. Guermond, B. Popov, I. Tomas, Invariant domain preserving discretization-independent schemes and convex limiting for hyperbolic systems, Comput. Methods Appl. Mech. Eng. 347 (2019) 143–175.
[28] H. Guillard, C. Farhat, On the significance of the geometric conservation law for flow computations on moving meshes, Comput. Methods Appl. Mech. Eng. 190 (11–12) (2000) 1467–1482.
[29] A. Harten, P.D. Lax, C.D. Levermore, W.J. Morokoff, Convex entropies and hyperbolicity for general Euler equations, SIAM J. Numer. Anal. 35 (6) (1998) 2117–2127 (electronic).
[30] A. Jameson, Origins and further development of the Jameson-Schmidt-Turkel scheme, AIAA J. 55 (5) (2017).
[31] A. Jameson, W. Schmidt, E. Turkel, Numerical solution of the Euler equations by finite volume. Methods using Runge-Kutta time-stepping schemes, in: 14th AIAA Fluid and Plasma Dynamics Conference, June 1981, AIAA Paper 1981-1259.
[32] B. Khobalatte, B. Perthame, Maximum principle on the entropy and second-order kinetic schemes, Math. Comput. 62 (205) (1994) 119–131.
[33] J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods, BIT 31 (3) (1991) 482–528.
[34] A. Kurganov, G. Petrova, B. Popov, Adaptive semidiscrete central-upwind schemes for nonconvex hyperbolic conservation laws, SIAM J. Sci. Comput. 29 (6) (2007) 2381–2401.
[35] M.-J. Lai, L.L. Schumaker, Spline Functions on Triangulations, Encyclopedia of Mathematics and Its Applications, vol. 110, Cambridge University Press, Cambridge, 2007.
[36] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera ReALE, A reconnection-based arbitrary-Lagrangian-Eulerian method, J. Comput. Phys. 229 (12) (2010) 4724–4761.
[37] W. Noh, Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux, J. Comput. Phys. 72 (1) (1987) 78–120.
[38] W.F. Noh, CEL: a Time-Dependent, Two-Space-Dimensional, Coupled Eulerian-Lagrange Code, Technical Report UCRL-7463, Lawrence Radiation Lab., Univ. of California, Livermore, Livermore, California, Aug. 1963, http://www.osti.gov/scitech/servlets/purl/4621975.
[39] G. Scovazzi, Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations, Comput. Methods Appl. Mech. Eng. 196 (2007) 966–978.
[40] G. Scovazzi, Galilean invariance and stabilized methods for compressible flows, Int. J. Numer. Methods Fluids 54 (6–8) (2007) 757–778.
[41] D. Serre, Systems of Conservation Laws. 2. Geometric Structures, Oscillations, and Initial-Boundary Value Problems, Cambridge University Press, Cambridge, 2000, Translated from the 1996 French original by I.N. Sneddon.
[42] D. Serre, A.F. Vasseur, About the relative entropy method for hyperbolic systems of conservation laws, in: A Panorama of Mathematics: Pure and Applied, in: Contemp. Math., vol. 658, Amer. Math. Soc., Providence, RI, 2016, pp. 237–248.
[43] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, J. Comput. Phys. 77 (2) (1988) 439–471.
[44] P.D. Thomas, C.K. Lombard, Geometric conservation law and its application to flow computations on moving grids, AIAA J. 17 (10) (1979) 1030–1037.
[45] E.F. Toro, Riemann Solvers and Numerical Methods for Fluid Dynamics, third edition, Springer-Verlag, Berlin, 2009, A practical introduction.
[46] F. Vilar, P.-H. Maire, R. Abgrall, A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids, J. Comput. Phys. 276 (2014) 188–234.
[47] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1) (1984) 115–173.
[48] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335–362.