An Arbitrary Lagrangian-Eulerian Finite Element method preserving convex invariants of hyperbolic systems

Jean-Luc Guermond, Bojan Popov, Laura Saavedra, and Yong Yang

Abstract We present a conservative Arbitrary Lagrangian Eulerian method for solving nonlinear hyperbolic systems. The key characteristics of the method is that it preserves all the convex invariants of the hyperbolic system in question. The method is explicit in time, uses continuous finite elements and is first-order accurate in space and high-order in time. The stability of the method is obtained by introducing an artificial viscosity that is unambiguously defined irrespective of the mesh geometry/anisotropy and does not depend on any ad hoc parameter.

1 Introduction

This paper is the expanded version of a talk given at the University of Houston in February 2016 at a workshop honoring the 70th birthday of Olivier Pironneau and his long lasting contributions to Numerical Analysis and Scientific Computing, [19]. The topic of paper is in the continuation of the groundbreaking work done by Olivier Pironneau on the analysis of the method of characteristics for solving the transport equation, [18]. More specifically, our objective is to build a finite element approximation to the entropy solution of the following hyperbolic system written in conservative form:

Jean-Luc Guermond and Bojan Popov
Department of Mathematics, Texas A&M University 3368 TAMU, College Station, TX 77843, USA, e-mail: guermond@math.tamu.edu, popov@math.tamu.edu

Laura Saavedra
Departamento Fundamentos Matemáticos, Universidad Politécnica de Madrid, E.T.S.I. Aeronáuticos, 28040 Madrid, Spain, e-mail: laura.saavedra@upm.es

Yong Yang
Penn State University Mathematics Dept. University Park, State College, PA 16802, USA, e-mail: yytamu@gmail.com

$$\begin{cases} \partial_t \boldsymbol{u} + \nabla\cdot\boldsymbol{f}(\boldsymbol{u}) = 0, \quad \text{for } (\boldsymbol{x},t) \in \mathbb{R}^d\times\mathbb{R}_+. \\ \boldsymbol{u}(\boldsymbol{x},0) = \boldsymbol{u}_0(\boldsymbol{x}), \quad \text{for } \boldsymbol{x} \in \mathbb{R}^d, \end{cases} \tag{1}$$

where the dependent variable $\boldsymbol{u}$ is $\mathbb{R}^m$-valued and the flux $\boldsymbol{f}$ is $\mathbb{R}^{m\times d}$-valued. We investigate in this paper an approximation technique using an Arbitrary Lagrangian Eulerian (ALE) formulation with continuous finite elements and explicit time stepping on non-uniform meshes.

The paper is organized as follows. We introduce some notation and recall important properties about the one-dimensional Riemann problem in §2. We introduce notation relative to mesh motion and Lagrangian mappings in §3. The results established in §2 and §3 are standard and will be invoked in §4 and §5. It is proved in §5 that under the appropriate CFL condition the algorithm is conservative, satisfies a local entropy inequality for every admissible entropy pair and preserves invariant domains. The main results of this section are Theorem 1 and Theorem 2. The SSP RK3 extension of the method is tested numerically in §6 on scalar conservation equations and on the compressible Euler equations. The paper essentially reproduces the arguments developed in Guermond et al. [13]. We refer the reader to [13] for details, proofs and extensions of the material presented herein.

2 Riemann problem and invariant domain

We recall in this section elementary properties of Riemann problems that will be used in the paper.

2.1 Notation and boundary conditions

The dependent variable $\boldsymbol{u}$ in (1) is considered as a column vector $\boldsymbol{u} = (u_1,\dots,u_m)^{\mathsf{T}}$. The flux is a matrix with entries $f_{ij}(\boldsymbol{u})$, $1 \le i \le m$, $1 \le j \le d$. We denote by $\boldsymbol{f}_i$ the row vector $(f_{i1},\dots,f_{id})$, $i \in \{1{:}m\}$. We denote by $\nabla\cdot\boldsymbol{f}$ the column vector with entries $(\nabla\cdot\boldsymbol{f})_i = \sum_{1\le j\le d}\partial_{x_j} f_{ij}$. For any $\boldsymbol{n} = (n_1\dots,n_d)^{\mathsf{T}} \in \mathbb{R}^d$, we denote $\boldsymbol{f}(\boldsymbol{u})\cdot\boldsymbol{n}$ the column vector with entries $\boldsymbol{f}_i(\boldsymbol{u})\cdot\boldsymbol{n} = \sum_{1\le l\le d} n_l f_{il}(\boldsymbol{u})$, where $i \in \{1{:}m\}$. Given two vector fields, say $\boldsymbol{u} \in \mathbb{R}^m$ and $\boldsymbol{v} \in \mathbb{R}^d$, we define $\boldsymbol{u}\otimes\boldsymbol{v}$ to be the $m\times d$ matrix with entries $u_i v_j$, $i \in \{1{:}m\}$, $j \in \{1{:}d\}$. We also define $\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{v})$ to be the column vector with entries $\nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{v})_i = \sum_{j=1}^d \partial_j(u_i v_j)$. The unit sphere in $\mathbb{R}^d$ centered at 0 is denoted by $S^{d-1}(\mathbf{0},1)$.

To simplify questions regarding boundary conditions, we assume that either the initial data is constant outside a compact set and we solve the Cauchy problem in $\mathbb{R}^d$ or we use periodic boundary conditions.

2.2 One-dimensional Riemann problem

We are not going to try to define weak solutions to (1), but instead we assume that there is a clear notion for the solution of the Riemann problem. To stay general we introduce a generic hyperbolic flux $\boldsymbol{h}$ and we say that $(\eta,\boldsymbol{q})$ is an entropy pair associated with the flux $\boldsymbol{h}$ if η is convex and the following identity holds:

$$\partial_{v_k}(\boldsymbol{q}(\boldsymbol{v})\cdot\boldsymbol{n}) = \sum_{i=1}^{m} \partial_{v_i}\eta(\boldsymbol{v})\partial_{v_k}(\boldsymbol{h}_i(\boldsymbol{v})\cdot\boldsymbol{n}), \qquad \forall k\in\{1{:}m\},\ \forall \boldsymbol{n}\in S^{d-1}(\boldsymbol{0},1). \tag{2}$$

We refer to Chen [4, §2] for more details on convex entropies and symmetrization. In the rest of the paper we assume that there exists a nonempty admissible set $\mathscr{A}_{\boldsymbol{h}}\subset\mathbb{R}^m$ such that the following one-dimensional Riemann problem

$$\partial_t\boldsymbol{u}+\partial_x(\boldsymbol{h}(\boldsymbol{u})\cdot\boldsymbol{n})=0, \quad (x,t)\in\mathbb{R}\times\mathbb{R}_+, \qquad \boldsymbol{u}(x,0)=\begin{cases}\boldsymbol{u}_L, & \text{if } x<0\\ \boldsymbol{u}_R, & \text{if } x>0,\end{cases} \tag{3}$$

has a unique entropy satisfying solution for any pair of states $(\boldsymbol{u}_L,\boldsymbol{u}_R)\in\mathscr{A}_{\boldsymbol{h}}\times\mathscr{A}_{\boldsymbol{h}}$ and any unit vector $\boldsymbol{n}\in S^{d-1}(\boldsymbol{0},1)$. We henceforth denote the solution to this problem by $\boldsymbol{u}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R)$. We also say that $\boldsymbol{u}$ is an entropy satisfying solution of (3) if the following holds in the distribution sense for any entropy pair $(\eta,\boldsymbol{q})$:

$$\partial_t\eta(\boldsymbol{u})+\partial_x(\boldsymbol{q}(\boldsymbol{u})\cdot\boldsymbol{n})\le 0. \tag{4}$$

Since it is unrealistic to expect a general theory of the Riemann problem (3) for arbitrary nonlinear hyperbolic systems with large data, we instead make the following assumption: The unique solution of (3) has a finite speed of propagation for any $\boldsymbol{n}$ and any $(\boldsymbol{u}_L,\boldsymbol{u}_R)\in\mathscr{A}_{\boldsymbol{h}}\times\mathscr{A}_{\boldsymbol{h}}$, i.e., there are $\lambda_L(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R)\le\lambda_R(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R)$ s.t.

$$\boldsymbol{u}(x,t)=\begin{cases}\boldsymbol{u}_L, & \text{if } x\le t\lambda_L(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R)\\ \boldsymbol{u}_R, & \text{if } x\ge t\lambda_R(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R).\end{cases} \tag{5}$$

This assumption is known to hold for small data when the system is strictly hyperbolic with smooth flux and all the characteristic fields are either genuinely nonlinear or linearly degenerate, see e.g., Dafermos [6, Thm. 9.5.1]. The sector $\lambda_L t<x<\lambda_R t$, $0<t$, is henceforth referred to as the Riemann fan. The maximum wave speed in the Riemann fan is $\lambda_{\max}:=\lambda_{\max}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{u}_L,\boldsymbol{u}_R):=\max(|\lambda_L|,|\lambda_R|)$.

2.3 Invariant sets and domains

The following elementary result is a well-known and important consequence of the Riemann fan assumption (5):

Lemma 1. *Let $\boldsymbol{h}$ be a hyperbolic flux over the admissible set $\mathscr{A}_{\boldsymbol{h}}$ and satisfying the finite wave speed assumption* (5)*. Let $\boldsymbol{v}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)$ be the unique solution to the problem $\partial_t\boldsymbol{v}+\partial_x(\boldsymbol{h}(\boldsymbol{v})\cdot\boldsymbol{n})=0$ with initial data $\boldsymbol{v}_L,\boldsymbol{v}_R\in\mathscr{A}_{\boldsymbol{h}}$. Let $(\eta,\boldsymbol{q})$ be an entropy pair associated with the flux $\boldsymbol{h}$. Assume that $t\,\lambda_{\max}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)\le\frac{1}{2}$ and let $\overline{\boldsymbol{v}}(t,\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R):=\int_{-\frac{1}{2}}^{\frac{1}{2}}\boldsymbol{v}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)(x,t)\,\mathrm{d}\boldsymbol{x}$, then*

$$\overline{\boldsymbol{v}}(t,\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)=\tfrac{1}{2}(\boldsymbol{v}_L+\boldsymbol{v}_R)-t\big(\boldsymbol{h}(\boldsymbol{v}_R)\cdot\boldsymbol{n}-\boldsymbol{h}(\boldsymbol{v}_L)\cdot\boldsymbol{n}\big). \tag{6}$$

$$\eta(\overline{\boldsymbol{v}}(t,\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R))\le\tfrac{1}{2}(\eta(\boldsymbol{v}_L)+\eta(\boldsymbol{v}_R))-t(\boldsymbol{q}(\boldsymbol{v}_R)\cdot\boldsymbol{n}-\boldsymbol{q}(\boldsymbol{v}_L)\cdot\boldsymbol{n}). \tag{7}$$

We now introduce notions of invariant sets that are slightly different from what is usually done in the literature (see e.g., in Chueh et al. [5], Hoff [15], Frid [9].)

Definition 1 (Invariant set). Let $\boldsymbol{h}$ be a hyperbolic flux over the admissible set $\mathscr{A}_{\boldsymbol{h}}$ and satisfying the finite wave speed assumption (5). A convex set $A\subset\mathscr{A}_{\boldsymbol{h}}\subset\mathbb{R}^m$ is said to be invariant for the problem $\partial_t\boldsymbol{v}+\nabla\cdot\boldsymbol{h}(\boldsymbol{v})=0$ if for any pair $(\boldsymbol{v}_L,\boldsymbol{v}_R)\in A\times A$, any unit vector $\boldsymbol{n}\in S^{d-1}(\mathbf{0},1)$, the average of the entropy solution of the Riemann problem $\partial_t\boldsymbol{v}+\nabla\cdot(\boldsymbol{h}(\boldsymbol{v})\cdot\boldsymbol{n})=0$ over the Riemann fan $\frac{1}{t(\lambda_R-\lambda_L)}\int_{\lambda_Lt}^{\lambda_Rt}\boldsymbol{v}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)(x,t)\,\mathrm{d}\boldsymbol{x}$, remains in A for all $t>0$.

Remark 1. The above definition implies that $\frac{1}{I}\int_I\boldsymbol{v}(\boldsymbol{h},\boldsymbol{n},\boldsymbol{v}_L,\boldsymbol{v}_R)(x,t)\,\mathrm{d}\boldsymbol{x}\in A$ for any $t>0$ and any interval I such that $(\lambda_Lt,\lambda_Rt)\subset I$.

Lemma 2 (Translation). *Let $\mathbf{W}\in\mathbb{R}^d$ and let $\boldsymbol{g}(\boldsymbol{v}):=\boldsymbol{f}(\boldsymbol{v})-\boldsymbol{v}\otimes\mathbf{W}$.*

(i) The two problems: $\partial_t\boldsymbol{u}+\nabla\cdot\boldsymbol{f}(\boldsymbol{u})=0$ and $\partial_t\boldsymbol{v}+\nabla\cdot\boldsymbol{g}(\boldsymbol{v})=0$ have the same admissible sets and the same invariant sets.

(ii) $(\eta(\boldsymbol{u}),\boldsymbol{q}(\boldsymbol{u}))$ is an entropy pair for the flux $\boldsymbol{f}$ if and only if $(\eta(\boldsymbol{v}),\boldsymbol{q}(\boldsymbol{v})-\eta(\boldsymbol{v})\mathbf{W})$ is an entropy pair for the flux $\boldsymbol{g}$.

3 Geometric preliminaries

In this section we introduce some notation and recall well known results about Lagrangian mappings. The key results, which will be invoked in §4 and §5, are lemmas 3 and 4. The reader who is familiar with these notions is invited to skip this section and to go directly to §4.

3.1 Jacobian of the coordinate transformation

Let $\boldsymbol{\Phi}:\mathbb{R}^d\times\mathbb{R}_+\longrightarrow\mathbb{R}^d$ be a uniformly Lipschitz mapping, and assume that there is $t^*>0$ such that the mapping $\boldsymbol{\Phi}_t:\mathbb{R}^d\ni\boldsymbol{\xi}\longmapsto\boldsymbol{\Phi}_t(\boldsymbol{\xi}):=\boldsymbol{\Phi}(\boldsymbol{\xi},t)\in\mathbb{R}^d$ is invertible for all $t\in[0,t^*]$. Let $\boldsymbol{v}_{\mathrm{A}}:\mathbb{R}^d\times[0,t^*]\longrightarrow\mathbb{R}^d$ be the vector field implicitly defined by

$$\boldsymbol{v}_{\mathrm{A}}(\boldsymbol{\Phi}(\boldsymbol{\xi},t),t) := \partial_t \boldsymbol{\Phi}(\boldsymbol{\xi},t), \quad \forall (\boldsymbol{\xi},t) \in \mathbb{R}\times[0,t^*]. \tag{8}$$

This definition makes sense owing to the inversibility assumption on the mapping $\boldsymbol{\Phi}_t$; actually (8) is equivalent to $\boldsymbol{v}_{\mathrm{A}}(\boldsymbol{x},t) := \partial_t \boldsymbol{\Phi}(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}),t)$ for any $t \in [0,t^*]$.

Lemma 3 (Liouville's formula). *Let $\mathbb{J}(\boldsymbol{\xi},t) = \nabla_{\boldsymbol{\xi}} \boldsymbol{\Phi}(\boldsymbol{\xi},t)$ be the Jacobian matrix of $\boldsymbol{\Phi}$, then*

$$\partial_t \det(\mathbb{J}(\boldsymbol{\xi},t)) = (\nabla\cdot\boldsymbol{v}_{\mathrm{A}})(\boldsymbol{\Phi}(\boldsymbol{\xi},t),t)\det(\mathbb{J}(\boldsymbol{\xi},t)). \tag{9}$$

Note that the expression $(\nabla\cdot\boldsymbol{v}_{\mathrm{A}})(\boldsymbol{\Phi}(\boldsymbol{\xi},t),t)$ in (9) should not be confused with $\nabla\cdot(\boldsymbol{v}_{\mathrm{A}}(\boldsymbol{\Phi}(\boldsymbol{\xi},t),t),t))$.

3.2 Arbitrary Lagrangian Eulerian formulation

The following result is the main motivation for the arbitrary Lagrangian Eulerian formulation that we are going to use in the paper.

Lemma 4. *The following identity holds in the distribution sense (in time) over the interval $[0,t^*]$ for every function $\psi \in C_0^0(\mathbb{R}^d;\mathbb{R})$ (with the notation $\varphi(\boldsymbol{x},t) := \psi(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}))$):*

$$\partial_t \int_{\mathbb{R}^d} \boldsymbol{u}(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^d} \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{v}_{\mathrm{A}} - \boldsymbol{f}(\boldsymbol{u}))\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}. \tag{10}$$

Proof. Using the chain rule and Lemma 3, we have

$$\begin{aligned}
\partial_t \int_{\mathbb{R}^d} \boldsymbol{u}(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} &= \partial_t \int_{\mathbb{R}^d} \boldsymbol{u}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\det(\mathbb{J}(\boldsymbol{\xi},t))\psi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^d} \big\{\partial_t(\boldsymbol{u}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t))\det(\mathbb{J}(\boldsymbol{\xi},t)) + \boldsymbol{u}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\partial_t(\det(\mathbb{J}(\boldsymbol{\xi},t)))\big\}\psi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\
&= \int_{\mathbb{R}^d} \big\{(\partial_t\boldsymbol{u})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t) + \partial_t\boldsymbol{\Phi}(\boldsymbol{\xi},t)\cdot(\nabla\boldsymbol{u})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\big\}\det(\mathbb{J}(\boldsymbol{\xi},t))\psi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi} \\
&\qquad + \int_{\mathbb{R}^d} \boldsymbol{u}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)(\nabla\cdot\boldsymbol{v}_{\mathrm{A}})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\det(\mathbb{J}(\boldsymbol{\xi},t))\psi(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}.
\end{aligned}$$

Then using (1) and the definition of the vector field $\boldsymbol{v}_{\mathrm{A}}$ yields

$$\begin{aligned}
&\partial_t \int_{\mathbb{R}^d} \boldsymbol{u}(\boldsymbol{x},t)\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x} = \int_{\mathbb{R}^d} -\nabla\cdot\boldsymbol{f}(\boldsymbol{u})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\psi(\boldsymbol{\xi})\det(\mathbb{J}(\boldsymbol{\xi},t))\,\mathrm{d}\boldsymbol{\xi} \\
&+\int_{\mathbb{R}^d} \big\{\boldsymbol{v}_{\mathrm{A}}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\cdot(\nabla\boldsymbol{u})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t) + (\nabla\cdot\boldsymbol{v}_{\mathrm{A}})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\boldsymbol{u}(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\big\}\psi(\boldsymbol{\xi})\det(\mathbb{J}(\boldsymbol{\xi},t))\,\mathrm{d}\boldsymbol{\xi} \\
&\qquad = \int_{\mathbb{R}^d} \big\{-\nabla\cdot\boldsymbol{f}(\boldsymbol{u})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t) + \nabla\cdot(\boldsymbol{u}\otimes\boldsymbol{v}_{\mathrm{A}})(\boldsymbol{\Phi}_t(\boldsymbol{\xi}),t)\big\}\psi(\boldsymbol{\xi})\det(\mathbb{J}(\boldsymbol{\xi},t))\,\mathrm{d}\boldsymbol{\xi}.
\end{aligned}$$

We conclude by making the change of variable $\boldsymbol{x} = \boldsymbol{\Phi}(\boldsymbol{\xi},t)$. □

We now state a result regarding the notion of entropy solution in the ALE framework. The proof of this result is similar to that of Lemma 4.

Lemma 5. *Let $(\eta,\boldsymbol{q})$ be an entropy pair for* (1)*. The following inequality holds in the distribution sense (in time) over the interval $[0,t^*]$ for every non-negative function $\psi\in C_0^0(\mathbb{R}^d;\mathbb{R}_+)$ (with the notation $\varphi(\boldsymbol{x},t):=\psi(\boldsymbol{\Phi}_t^{-1}(\boldsymbol{x}))$):*

$$\partial_t\int_{\mathbb{R}^d}\eta(\boldsymbol{u}(\boldsymbol{x},t))\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}\le\int_{\mathbb{R}^d}\nabla\cdot(\eta(\boldsymbol{u})\boldsymbol{v}_{\mathrm{A}}-\boldsymbol{q}(\boldsymbol{u}))\varphi(\boldsymbol{x},t)\,\mathrm{d}\boldsymbol{x}. \tag{11}$$

4 The Arbitrary Lagrangian Eulerian algorithm

We describe in this section the ALE algorithm to approximate the solution of (1). We use continuous finite elements and explicit time stepping. We use two different discrete settings: one for the mesh motion and one for the approximation of (1).

4.1 Geometric finite elements and mesh

Let $(\mathscr{T}_h^0)_{h>0}$ be a shape-regular sequence of matching meshes. The symbol ^{0} in $\mathscr{T}_h^0$ refers to the initial configuration of the meshes. The meshes will deform over time, in a way that has yet to be defined, and we are going to use the symbol ^{n} to say that $\mathscr{T}_h^n$ is the mesh at time t^n for a given $h>0$. We assume that the elements in the mesh cells are generated from a finite number of reference elements denoted $\widehat{K}_1,\dots,\widehat{K}_{\varpi}$. For instance, $\mathscr{T}_h^0$ could be composed of a combination of triangles and parallelograms in two space dimensions ($\varpi=2$ in this case); the mesh $\mathscr{T}_h^0$ could also be composed of a combination of tetrahedra, parallelepipeds, and triangular prisms in three space dimensions ($\varpi=3$ in this case). The diffeomorphism mapping $\widehat{K}_r$ to an arbitrary element $K\in\mathscr{T}_h^n$ is denoted $T_K^n:\widehat{K}_r\longrightarrow K$ and its Jacobian matrix is denoted $\mathbb{J}_K^n$, $1\le r\le\varpi$. We now introduce a set of reference Lagrange finite elements $\{(\widehat{K}_r,\widehat{P}_r^{\mathrm{geo}},\widehat{\Sigma}_r^{\mathrm{geo}})\}_{1\le r\le\varpi}$ (the index $r\in\{1:\varpi\}$ will be omitted in the rest of the paper to alleviate the notation). Letting $n_{\mathrm{sh}}^{\mathrm{geo}}:=\dim\widehat{P}^{\mathrm{geo}}$, we denote by $\{\widehat{\boldsymbol{a}}_i\}_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}$ and $\{\widehat{\theta}_i^{\mathrm{geo}}\}_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}$ the Lagrange nodes of $\widehat{K}$ and the associated Lagrange shape functions.

The unique purpose of the geometric reference element $\{(\widehat{K},\widehat{P}^{\mathrm{geo}},\widehat{\Sigma}^{\mathrm{geo}})$ is to construct the geometric transformation T_K^n. Let $\{\boldsymbol{a}_i^n\}_{i\in\{1:I^{\mathrm{geo}}\}}$ be the collection of all the Lagrange nodes in the mesh $\mathscr{T}_h^n$. The Lagrange nodes are organized in cells by means of the geometric connectivity array $\mathrm{j}^{\mathrm{geo}}:\mathscr{T}_h^n\times\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}\longrightarrow\{1:I^{\mathrm{geo}}\}$ (assumed to be independent of the time index n). Given a mesh cell $K\in\mathscr{T}_h^n$, the connectivity array is defined such that $\{\boldsymbol{a}^n_{\mathrm{j}^{\mathrm{geo}}(i,K)}\}_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}$ is the set of the Lagrange nodes describing K^n. More precisely, upon defining the geometric transformation $T_K^n:\widehat{K}\longrightarrow K$ at time t^n by

$$T_K^n(\widehat{\boldsymbol{x}}) = \sum_{i\in\{1:n_{\rm sh}^{\rm geo}\}} \boldsymbol{a}^n_{{\rm j}^{\rm geo}(i,K)} \widehat{\theta}_i^{\rm geo}(\widehat{\boldsymbol{x}}) \tag{12}$$

we have $K := T_K^n(\widehat{K})$. In other words the geometric transformation is fully described by the motion of geometric Lagrange nodes. Recall that constructing the Jacobian matrix $\mathbb{J}_K^n$ from (12) is an elementary operation for any finite element code.

4.2 Approximating finite elements

We now introduce a set of reference finite elements $\{(\widehat{K}_r, \widehat{P}_r, \widehat{\Sigma}_r)\}_{1\le r\le \varpi}$ which we are going to use to construct an approximate solution to (1) (the index $r \in \{1:\varpi\}$ will be omitted in the rest of the paper to alleviate the notation). The shape functions on the reference element are denoted $\{\widehat{\theta}_i\}_{i\in\{1:n_{\rm sh}\}}$. We assume that the basis $\{\widehat{\theta}_i\}_{i\in\{1:n_{\rm sh}\}}$ has the following key properties:

$$\widehat{\theta}_i(\boldsymbol{x}) \ge 0, \quad \sum_{i\in\{1:n_{\rm sh}\}} \widehat{\theta}_i(\widehat{\boldsymbol{x}}) = 1, \quad \forall \widehat{\boldsymbol{x}} \in \widehat{K}. \tag{13}$$

These properties hold true for linear Lagrange elements and for Bernstein-Bezier finite elements, see e.g., Lai and Schumaker [16, Chap. 2], Ainsworth [1].

Given the mesh $\mathscr{T}_h^n$, we denote by D^n the computational domain generated by $\mathscr{T}_h^n$ and we define the scalar-valued space

$$P(\mathscr{T}_h^n) := \{v \in \mathscr{C}^0(D^n;\mathbb{R}) \mid v_{|K}\circ T_K^n \in \widehat{P},\ \forall K \in \mathscr{T}_h^n\}, \tag{14}$$

We also introduce the vector-valued spaces

$$\boldsymbol{P}_d(\mathscr{T}_h^n) := [P(\mathscr{T}_h^n)]^d, \quad \text{and} \quad \boldsymbol{P}_m(\mathscr{T}_h^n) := [P(\mathscr{T}_h^n)]^m. \tag{15}$$

We are going to approximate the ALE velocity in $\boldsymbol{P}_d(\mathscr{T}_h^n)$ and the solution of (1) in $\boldsymbol{P}_m(\mathscr{T}_h^n)$. The global shape functions in $P(\mathscr{T}_h^n)$ are denoted by $\{\psi_i^n\}_{i\in\{1:I\}}$. Recall that these functions form a basis of $P(\mathscr{T}_h^n)$. Let ${\rm j} : \mathscr{T}_h^n \times \{1:n_{\rm sh}\} \longrightarrow \{1:I\}$ be the connectivity array, assumed to be independent of n. This array is defined such that

$$\psi^n_{{\rm j}(i,K)}(\boldsymbol{x}) = \widehat{\theta}_i((T_K^n)^{-1}(\boldsymbol{x})), \quad \forall i \in \{1:n_{\rm sh}\},\ \forall K \in \mathscr{T}_h^n. \tag{16}$$

This definition together with (13) implies that

$$\psi_i^n(\boldsymbol{x}) \ge 0, \quad \sum_{i\in\{1:I\}} \psi_i^n(\boldsymbol{x}) = 1, \quad \forall \boldsymbol{x} \in \mathbb{R}^d. \tag{17}$$

We denote by S_i^n the support of ψ_i^n and by $|S_i^n|$ the measure of S_i, $i \in \{1:I\}$. We also define $S_{ij}^n := S_i^n \cap S_j^n$ the intersection of the two supports S_i^n and S_j^n. Let E be a union of cells in $\mathscr{T}_h^n$; we define $\mathscr{I}(E) := \{j \in \{1:I\} \mid |S_j^n \cap E| \ne 0\}$ the set that

contains the indices of all the shape functions whose support on E is of nonzero measure. Note that the index set $\mathscr{I}(E)$ does not depend on the time index n since we have assumed that the connectivity of the degrees of freedom is fixed once for all. We are going to regularly invoke $\mathscr{I}(K)$ and $\mathscr{I}(S_i^n)$ and the partition of unity property: $\sum_{i\in\mathscr{I}(K)}\psi_i^n(\boldsymbol{x})=1$ for all $\boldsymbol{x}\in K$.

Lemma 6. *For all $K\in\mathscr{T}_h^n$, all $\boldsymbol{x}\in K$, and all $\boldsymbol{v}_h:=\sum_{i\in\{1:I\}}\mathbf{V}_i\psi_i^n\in\boldsymbol{P}_m(\mathscr{T}_h^n)$, $\boldsymbol{v}_h(\boldsymbol{x})$ is in the convex hull of $(\mathbf{V}_i)_{i\in\mathscr{I}(K)}$ (henceforth denoted $\mathrm{conv}(\mathbf{V}_i)_{i\in\mathscr{I}(K)}$). Moreover for any convex set A in $\mathbb{R}^m$, we have*

$$\big((\mathbf{V}_i)_{i\in\mathscr{I}(K)}\in A\big)\Rightarrow(\boldsymbol{v}_h(\boldsymbol{x})\in A,\ \forall\boldsymbol{x}\in K)\,. \tag{18}$$

4.3 The ALE algorithm

Let $\mathscr{T}_h^0$ be the mesh at the initial time $t=0$. Let $(\mathfrak{m}_i^0)_{i\in\{1:I\}}$ be the approximations of the mass of the shape functions at time t^0 defined by $\mathfrak{m}_i^0=m_i^0:=\int_{\mathbb{R}^d}\psi_i^0(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. Let $\boldsymbol{u}_{h0}:=\sum_{i\in\{1:I\}}\mathbf{U}_i^0\psi_i^0\in\boldsymbol{P}_m(\mathscr{T}_h^0)$ be a reasonable approximation of the initial data $\boldsymbol{u}_0$ (we shall make a more precise statement later).

Let $\mathscr{T}_h^n$ be the mesh at time t^n, $(\mathfrak{m}_i^n)_{1\le i\le I}$ be the approximations of the mass of the shape functions at time t^n, and $\boldsymbol{u}_h^n:=\sum_{i\in\{1:I\}}\mathbf{U}_i^n\psi_i^n\in\boldsymbol{P}_m(\mathscr{T}_h^n)$ be the approximation of $\boldsymbol{u}$ at time t^n. We denote by $\mathfrak{M}^{L,n}$ the approximate lumped matrix, i.e., $\mathfrak{M}_{ij}^{L,n}=\mathfrak{m}_i^n\delta_{ij}$. We now make the assumption that the given ALE velocity field is a member of $\boldsymbol{P}_d(\mathscr{T}_h^n)$, i.e., $\boldsymbol{w}^n=\sum_{i\in\{1:I\}}\mathbf{W}_i^n\psi_i^n\in\boldsymbol{P}_d(\mathscr{T}_h^n)$. Then the Lagrange nodes of the mesh are moved by using the following rule:

$$\boldsymbol{a}_i^{n+1}=\boldsymbol{a}_i^n+\tau\boldsymbol{w}^n(\boldsymbol{a}_i^n). \tag{19}$$

This fully defines the mesh $\mathscr{T}_h^{n+1}$ as explained at the end of §4.1. Upon introducing $\psi^{\mathrm{geo}}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(\boldsymbol{\xi}):=\widehat{\theta}_i((T_K^n)^{-1}(\boldsymbol{\xi}))$ and $\boldsymbol{a}_i(t)=\boldsymbol{a}_i^n+(t-t^n)\boldsymbol{w}^n(\boldsymbol{a}_i^n)$ for $t\in[t^n,t^n+\tau]$, this also defines the ALE mapping

$$\boldsymbol{\Phi}_{t|K}(\boldsymbol{\xi})=\sum_{i\in\{1:n_{\mathrm{sh}}^{\mathrm{geo}}\}}\boldsymbol{a}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(t)\,\psi^{\mathrm{geo}}_{\mathrm{j}^{\mathrm{geo}}(i,K)}(\boldsymbol{\xi}),\qquad\forall\boldsymbol{\xi}\in K,\ \forall K\in\mathscr{T}_h^n. \tag{20}$$

We now estimate the mass of the shape function $\psi_i^{n+1}:=\psi_i^n\circ\boldsymbol{\Phi}_{t^{n+1}}$. Of course we could use $m_i^{n+1}=\int_{\mathbb{R}^d}\psi_i^{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$. This option leads to many difficulties that are explored in Guermond et al. [13]; in particular, extending the method to high-order in time with this definition is problematic. To have a method that is compatible with higher-order strong stability preserving (SSP) time stepping techniques, we define $\mathfrak{m}_i^{n+1}$ by approximating the following identity with a first-order quadrature rule:

$$\int_{\mathbb{R}^d}\psi^{n+1}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}-\int_{\mathbb{R}^d}\psi^n(\boldsymbol{\xi})\,\mathrm{d}\boldsymbol{\xi}=\int_{\mathbb{R}^d}\psi^n(\boldsymbol{\xi})\left[\int_{t^n}^{t^{n+1}}\partial_\zeta\det(\mathbb{J}(\boldsymbol{\xi},\zeta))\,\mathrm{d}\zeta\right]\mathrm{d}\boldsymbol{\xi}.$$

Note that $\det(\mathbb{J}(\boldsymbol{\xi},\zeta))$ is a polynomial function of ζ of degree d. The first-order approximation of the integral with respect to ζ in the above expression gives:

$$\mathfrak{m}_i^{n+1} = \mathfrak{m}_i^n + \tau \int_{S_i^n} \psi_i^n(\boldsymbol{x}) \nabla\cdot\boldsymbol{w}^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{21}$$

Taking inspiration from (10), we propose to compute $\boldsymbol{u}_h^{n+1}$ as follows:

$$\begin{aligned}\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n\mathbf{U}_i^n}{\tau} - \sum_{j\in\mathscr{I}(S_i^n)} d_{ij}^n\mathbf{U}_j^n \\ + \int_{\mathbb{R}^d} \nabla\cdot\Bigg(\sum_{j\in\{1:I\}} (\boldsymbol{f}(\mathbf{U}_j^n) - \mathbf{U}_j^n\otimes\mathbf{W}_j^n)\psi_j^n(\boldsymbol{x})\Bigg)\psi_i^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 0,\end{aligned} \tag{22}$$

where $\boldsymbol{u}_h^{n+1} := \sum_{i\in\{1:I\}} \mathbf{U}_i^{n+1}\psi_i^{n+1} \in \boldsymbol{P}_m(\mathscr{T}_h^{n+1})$. Notice that we have replaced the consistent mass matrix by an approximation of the lumped mass matrix to approximate the time derivative. The coefficient d_{ij}^n is an artificial viscosity for the pair of degrees of freedom (i,j) that will be identified by proceeding as in Guermond and Popov [12]. We henceforth assume that $d_{ij}^n = 0$ if $j\notin\mathscr{I}(S_i^n)$ and

$$d_{ij}^n \geq 0, \text{ if } i\neq j, \qquad d_{ij}^n = d_{ji}^n, \quad \text{and} \qquad d_{ii} := \sum_{i\neq j\in\mathscr{I}(S_i^n)} -d_{ji}^n. \tag{23}$$

The entire process is described in Algorithm 1.

Let us reformulate (22) in a form that is more suitable for computations. Let us introduce the vector-valued coefficients

$$\boldsymbol{c}_{ij}^n := \int_{S_i^n} \nabla\psi_j^n(\boldsymbol{x})\psi_i^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{24}$$

We define the unit vector $\boldsymbol{n}_{ij}^n := \frac{\boldsymbol{c}_{ij}^n}{\|\boldsymbol{c}_{ij}^n\|_{\ell^2}}$. Then we rewrite (22) as follows:

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n\mathbf{U}_i^n}{\tau} + \sum_{j\in\mathscr{I}(S_i^n)} (\boldsymbol{f}(\mathbf{U}_j^n) - \mathbf{U}_j^n\otimes\mathbf{W}_j^n)\cdot\boldsymbol{c}_{ij}^n - d_{ij}^n\mathbf{U}_j^n = 0. \tag{25}$$

It will be shown in the proof of Theorem 1 that an admissible choice for d_{ij}^n is

$$d_{ij}^n = \max(\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)\|\boldsymbol{c}_{ij}^n\|_{\ell^2}, \lambda_{\max}(\boldsymbol{g}_i^n, \boldsymbol{n}_{ji}^n, \mathbf{U}_j^n, \mathbf{U}_i^n)\|\boldsymbol{c}_{ji}^n\|_{\ell^2}). \tag{26}$$

where $\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ is the largest wave speed in the following one-dimensional Riemann problem with the flux $\boldsymbol{g}_j^n(\boldsymbol{v}) := \boldsymbol{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes\mathbf{W}_j^n$:

$$\partial_t\boldsymbol{v} + \partial_x(\boldsymbol{g}_j^n(\boldsymbol{v})\cdot\boldsymbol{n}_{ij}^n) = 0, \quad (x,t)\in\mathbb{R}\times\mathbb{R}_+, \quad \boldsymbol{v}(x,0) = \begin{cases}\mathbf{U}_i^n & \text{if } x<0\\ \mathbf{U}_j^n & \text{if } x>0.\end{cases} \tag{27}$$

Remark 2 (Fastest wave speed). The fastest wave speed in (27) can be obtained by estimating the fastest wave speed in the Riemann problem (3) with the flux $\boldsymbol{f}(\boldsymbol{v})\cdot\boldsymbol{n}_{ij}^n$ and the initial data $(\mathbf{U}_i^n,\mathbf{U}_j^n)$. Let $\lambda_L(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)$ and $\lambda_R(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)$ be the speed of the leftmost and the rightmost waves in (3), respectively. Then

$$\lambda_{\max}(\boldsymbol{g}_j^n,\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)=\max(|\lambda_L(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)-\mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|,\\ |\lambda_R(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)-\mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|). \quad (28)$$

A fast algorithm to compute $\lambda_L(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)$ and $\lambda_R(\boldsymbol{f},\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n)$ for the compressible Euler equations is given in Guermond and Popov [11]; see also Toro [20].

Algorithm 1

Require: $\boldsymbol{u}_h^0$ and $\mathfrak{M}^{L,0}$
1: **while** $t^n<T$ **do**
2: Use CFL condition to estimate τ.
3: **if** $t^n+\tau>T$ **then**
4: $\tau\leftarrow T-t^n$
5: **end if**
6: Estimate/choose $\boldsymbol{w}^n$ and make sure that the transformation $\boldsymbol{\Phi}_t$ defined in (20) is invertible over the interval $[t^n,t^{n+1}]$.
7: Move mesh from t^n to t^{n+1} using (19).
8: Compute $\mathfrak{m}_i^{n+1}$, see (21). Check $\mathfrak{m}_i^{n+1}>0$; otherwise, go to step 6, reduce τ.
9: Compute $\boldsymbol{c}_{ij}^n$ as in (24).
10: Compute d_{ij}^n, see (26) and (23).
11: Check $1-\sum_{i\neq j\in\mathscr{I}(S_i^n)}2d_{ij}^n\frac{\tau}{\mathfrak{m}_i^{n+1}}$ positive. Otherwise, go to step 6 and reduce τ.
12: Compute $\boldsymbol{u}_h^{n+1}$ by using (25).
13: $t^n\leftarrow t^n+\tau$
14: **end while**

Since it is important to compare $\mathbf{U}_j^{n+1}$ and $\mathbf{U}_j^n$ to establish the invariant domain property, we rewrite the scheme in a form that is more suitable for this purpose.

Lemma 7 (Non-conservative form). *The scheme* (22) *is equivalent to*

$$\mathfrak{m}_i^{n+1}\frac{\mathbf{U}_i^{n+1}-\mathbf{U}_i^n}{\tau}=\sum_{j\in\mathscr{I}(S_i^n)}((\mathbf{U}_j^n-\mathbf{U}_i^n)\otimes\mathbf{W}_j^n-\boldsymbol{f}(\mathbf{U}_j^n))\cdot\boldsymbol{c}_{ij}^n+d_{ij}^n\mathbf{U}_j^n, \quad (29)$$

Proof. We rewrite (25) as follows:

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

Then, recalling the expression $\boldsymbol{w}^n=\sum_{i\in\{1:I\}}\mathbf{W}_i^n\psi_i^n$, and using (21), we infer that $\mathfrak{m}_i^{n+1}=\mathfrak{m}_i^n+\tau\sum_{j\in\mathscr{I}(S_i^n)}\mathbf{W}_j^n\cdot\boldsymbol{c}_{ij}^n$, which in turn implies that

$$(\mathfrak{m}_i^{n+1}-\mathfrak{m}_i^n)\mathbf{U}_i^n=\tau\mathbf{U}_i^n\sum_{j\in\mathscr{I}(S_i^n)}\mathbf{W}_j^n\cdot\boldsymbol{c}_{ij}^n=\tau\sum_{j\in\mathscr{I}(S_i^n)}(\mathbf{U}_i^n\otimes\mathbf{W}_j^n)\cdot\boldsymbol{c}_{ij}^n. \qquad \square$$

Remark 3 (Other discretizations). The method for computing the artificial diffusion is quite generic, i.e., it is not specific to continuous finite elements. The above method can be applied to any type of discretization that can be put into the form (25).

4.4 SSP extension

Retaining the invariant domain property (see §5.1) and increasing the time accuracy can be done by using so-called Strong Stability Preserving (SSP) time discretization methods. The key is to achieve higher-order accuracy in time by making convex combination of solutions of forward Euler sub-steps. More precisely each time step of a SSP method is decomposed into substeps that are all forward Euler solutions, and the end of step solution is a convex combination of the intermediate solutions; see Ferracina and Spijker [8], Higueras [14], Gottlieb et al. [10] for reviews on SPP techniques. Algorithm 2 illustrates one Euler step of the scheme. SSP techniques are useful when combined with reasonable limitation strategies since the resulting methods are high-order, both in time and space, and invariant domain preserving.

Algorithm 2 Euler step

Require: $\mathscr{T}_h^0$, $\boldsymbol{u}_h^0$, ($\mathfrak{m}^0$ or m^0), $\boldsymbol{w}^0$, τ

1: Compute $\widetilde{\boldsymbol{a}}_i^1=\boldsymbol{a}_i^0+\tau\boldsymbol{w}^0$, ($\widetilde{\mathfrak{m}}^1$ or $\widetilde{m}^1$), $\widetilde{\boldsymbol{u}}_h^1$, and build new mesh $\widetilde{\mathscr{T}}_h^1$

2: **return** $\widetilde{\mathscr{T}}_h^1$, $\widetilde{\boldsymbol{u}}_h^1$, ($\widetilde{\mathfrak{m}}^1$ or $\widetilde{m}^1$)

As an illustration we describe the SSP RK3 implementation of the scheme in Algorithm 3. Generalizations to other SSP techniques are left to the reader.

Algorithm 3 SPP RK3

Require: $\mathscr{T}_h^0$, $\boldsymbol{u}_h^0$, $\mathfrak{m}^0$, t^0

1: Define the ALE velocity $\boldsymbol{w}^0$ at t^0

2: Call Euler step($\mathscr{T}_h^0$, $\boldsymbol{u}_h^0$, $\mathfrak{m}^0$, $\boldsymbol{w}^0$, τ, $\mathscr{T}_h^1$, $\boldsymbol{u}_h^1$, $\mathfrak{m}^1$)

3: Define the ALE velocity $\boldsymbol{w}^1$ at $t^0+\tau$

4: Call Euler step($\mathscr{T}_h^1$, $\boldsymbol{u}_h^1$, $\mathfrak{m}^1$, $\boldsymbol{w}^1$, τ, $\widetilde{\mathscr{T}}_h^2$, $\widetilde{\boldsymbol{u}}_h^2$, $\widetilde{\mathfrak{m}}^2$)

5: Set $\boldsymbol{a}^2=\frac{3}{4}\boldsymbol{a}^0+\frac{1}{4}\widetilde{\boldsymbol{a}}^2$, $\mathfrak{m}^2=\frac{3}{4}\mathfrak{m}^0+\frac{1}{4}\widetilde{\mathfrak{m}}^2$, build mesh $\mathscr{T}_h^2$, $\boldsymbol{u}_h^2=\frac{3}{4}\frac{\mathfrak{m}^0}{\mathfrak{m}^2}\boldsymbol{u}_h^0+\frac{1}{4}\frac{\widetilde{\mathfrak{m}}^2}{\mathfrak{m}^2}\widetilde{\boldsymbol{u}}_h^2$

6: Define the ALE velocity $\boldsymbol{w}^2$ at $t^0+\frac{1}{2}\tau$

7: Call Euler step($\mathscr{T}_h^2$, $\boldsymbol{u}_h^2$, $\mathfrak{m}^2$, $\boldsymbol{w}^2$, τ, $\widetilde{\mathscr{T}}_h^3$, $\widetilde{\boldsymbol{u}}_h^3$, $\widetilde{\mathfrak{m}}^3$)

8: Set $\boldsymbol{a}^3=\frac{1}{3}\boldsymbol{a}^0+\frac{2}{3}\widetilde{\boldsymbol{a}}^3$, $\mathfrak{m}^3=\frac{1}{3}\mathfrak{m}^0+\frac{2}{3}\widetilde{\mathfrak{m}}^3$, build mesh $\mathscr{T}_h^3$, $\boldsymbol{u}_h^3=\frac{1}{3}\frac{\mathfrak{m}^0}{\mathfrak{m}^3}\boldsymbol{u}_h^0+\frac{2}{3}\frac{\widetilde{\mathfrak{m}}^3}{\mathfrak{m}^3}\widetilde{\boldsymbol{u}}_h^3$

9: **return** $\mathscr{T}_h^3$, $\boldsymbol{u}_h^3$, $\mathfrak{m}^3$, $t^1=t^0+dt$

Note that $\boldsymbol{u}_h^2$ is a convex combination of $\boldsymbol{u}_h^0$ and $\widetilde{\boldsymbol{u}}_h^2$ since $1=\frac{3}{4}\frac{\mathrm{m}_i^0}{\mathrm{m}_i^2}+\frac{1}{4}\frac{\widetilde{\mathrm{m}}_i^2}{\mathrm{m}_i^2}$. The same observation holds true for $\boldsymbol{u}_h^3$, i.e., $\boldsymbol{u}_h^3$ is a convex combination of $\boldsymbol{u}_h^0$ and $\widetilde{\boldsymbol{u}}_h^3$ since $1=\frac{1}{3}\frac{\mathrm{m}_i^0}{\mathrm{m}_i^3}+\frac{2}{3}\frac{\widetilde{\mathrm{m}}_i^3}{\mathrm{m}_i^3}$, for any $i\in\{1{:}I\}$.

5 Stability analysis

We establish the conservation and the invariant domain property of the scheme (22).

5.1 Invariant domain property

We first discuss the conservation properties of the scheme.

Lemma 8. *For the scheme* (22)*, the quantity* $\sum_{i\in\{1:I\}}\mathrm{m}_i^n\mathbf{U}_i^n$ *is is independent of n, i.e., the total mass is conserved.*

We can now prove a result somewhat similar in spirit to Thm 5.1 from Farhat et al. [7], although the present result is more general since it applies to any hyperbolic system. We define the local minimum mesh size $\underline{h}_{ij}^n$ associated with an ordered pair of shape functions (ψ_i^n,ψ_j^n) as follows: $\underline{h}_{ij}^n:=\frac{1}{\|\|\nabla\varphi_j\|_{\ell^2}\|_{L^\infty(S_{ij}^n)}}$, where $S_{ij}^n=S_i^n\cap S_j^n$. We then define a local mesh size and a local mesh structure parameter κ_i^n by setting

$$\underline{h}_i^n=\min_{j\in\mathscr{I}(S_i^n)}\underline{h}_{ij}^n,\qquad \kappa_i^n:=\frac{\sum_{i\neq j\in\mathscr{I}(S_i^n)}\int_{S_{ij}^n}\psi_i^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}{\int_{S_i^n}\psi_i^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}}. \tag{30}$$

Note that the upper estimate $\kappa_i^n\le\max_{j\in\{1:I\}}\mathrm{card}(\mathscr{I}(S_j(0)))-1$ implies that κ_i^n is uniformly bounded with respect to n and i.

Theorem 1 (Local invariance). *Let* $n\ge 0$*, and* $i\in\{1{:}I\}$*. Assume the CFL condition*

$$2\tau\frac{\lambda_{i,\max}^n}{\underline{h}_i^n}\kappa_i^n\frac{\mathrm{m}_i^n}{\mathrm{m}_i^{n+1}}\le 1, \tag{31}$$

where $\lambda_{i,\max}^n:=\max_{j\in\mathscr{I}(S_i^n)}(\lambda_{\max}(\boldsymbol{g}_j^n,\boldsymbol{n}_{ij}^n,\mathbf{U}_i^n,\mathbf{U}_j^n),\ \lambda_{\max}(\boldsymbol{g}_i^n,\boldsymbol{n}_{ji}^n,\mathbf{U}_j^n,\mathbf{U}_i^n))$*. Let* $B\subset\mathscr{A}_{\boldsymbol{f}}$ *be a convex invariant set for the flux* $\boldsymbol{f}$*. If* $\{\mathbf{U}_j^n\mid j\in\mathscr{I}(S_i^n)\}\subset B$*, then* $\mathbf{U}_i^{n+1}\in B$*.*

Proof. Let $i\in\{1{:}I\}$ and invoke (29) from Lemma 7 to express $\mathbf{U}_i^{n+1}$ as follows:

$$\mathbf{U}_i^{n+1}=\mathbf{U}_i^n+\frac{\tau}{\mathrm{m}_i^{n+1}}\sum_{j\in\mathscr{I}(S_i^n)}((\mathbf{U}_j^n-\mathbf{U}_i^n)\otimes\mathbf{W}_j^n-\boldsymbol{f}(\mathbf{U}_j^n))\cdot\boldsymbol{c}_{ij}^n+d_{ij}^n\mathbf{U}_j^n.$$

Since the partition of unity property implies that $\sum_{j\in\mathscr{I}(S_i^n)} \boldsymbol{c}_{ij}^n = 0$ and we have $\sum_{j\in\mathscr{I}(S_i^n)} d_{ij}^n = 0$ from (23), we can rewrite the above equation as follows:

$$\begin{aligned}\mathbf{U}_i^{n+1} &= \mathbf{U}_i^n + \sum_{j\in\mathscr{I}(S_i^n)} d_{ij}^n(\mathbf{U}_i^n + \mathbf{U}_j^n) \\ &\qquad + \frac{\tau}{\mathfrak{m}_i^{n+1}} \sum_{j\in\mathscr{I}(S_i^n)} ((\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes \mathbf{W}_j^n + \boldsymbol{f}(\mathbf{U}_i^n) - \boldsymbol{f}(\mathbf{U}_j^n))\cdot \boldsymbol{c}_{ij}^n \\ &= \mathbf{U}_i^n\left(1 + 2d_{ii}^n \frac{\tau}{\mathfrak{m}_i^{n+1}}\right) + \sum_{i\neq j\in\mathscr{I}(S_i^n)} d_{ij}^n(\mathbf{U}_i^n + \mathbf{U}_j^n) \\ &+ \frac{\tau}{\mathfrak{m}_i^{n+1}} \sum_{i\neq j\in\mathscr{I}(S_i^n)} ((\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes \mathbf{W}_j^n + \boldsymbol{f}(\mathbf{U}_i^n) - \boldsymbol{f}(\mathbf{U}_j^n))\cdot \boldsymbol{c}_{ij}^n.\end{aligned}$$

Let us introduced the auxiliary state $\overline{\mathbf{U}}_{ij}^{n+1}$ defined by

$$\overline{\mathbf{U}}_{ij}^{n+1} = (\boldsymbol{f}(\mathbf{U}_i^n) - \boldsymbol{f}(\mathbf{U}_j^n) - (\mathbf{U}_i^n - \mathbf{U}_j^n)\otimes \mathbf{W}_j^n)\cdot \boldsymbol{n}_{ij}^n \frac{\|\boldsymbol{c}_{ij}^n\|_{\ell^2}}{2d_{ij}^n} + \frac{1}{2}(\mathbf{U}_i^n + \mathbf{U}_j^n),$$

where $\boldsymbol{n}_{ij}^n := \boldsymbol{c}_{ij}^n/\|\boldsymbol{c}_{ij}^n\|_{\ell^2}$. Then, provided we establish that $1 - \sum_{i\neq j\in\mathscr{I}(S_i^n)} 2d_{ij}^n \frac{\tau}{\mathfrak{m}_i^{n+1}} \geq 0$, we have proved that $\mathbf{U}_i^{n+1}$ is a convex combination of $\mathbf{U}_i^n$ and $(\overline{\mathbf{U}}_{ij}^{n+1})_{i\neq j\in\mathscr{I}(S_i^n)}$:

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n\left(1 - \sum_{i\neq j\in\mathscr{I}(S_i^n)} 2d_{ij}^n \frac{\tau}{\mathfrak{m}_i^{n+1}}\right) + \frac{\tau}{\mathfrak{m}_i^{n+1}} \sum_{i\neq j\in\mathscr{I}(S_i^n)} 2d_{ij}^n \overline{\mathbf{U}}_{ij}^{n+1}. \tag{32}$$

Let us now consider the Riemann problem (27). Let $\boldsymbol{v}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ be the solution to (27) with $\boldsymbol{g}_j^n(\boldsymbol{v}) := \boldsymbol{f}(\boldsymbol{v}) - \boldsymbol{v}\otimes \mathbf{W}_j^n$. Let $\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ be the fastest wave speed in (27), see (28). Using the notation of Lemma 1, we then observe that $\overline{\mathbf{U}}_{ij}^{n+1} = \overline{\boldsymbol{v}}(t, \boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n)$ with $t = \frac{\|\boldsymbol{c}_{ij}^n\|_{\ell^2}}{2d_{ij}^n}$, provided $t\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) \leq \frac{1}{2}$. Note that the definition of d_{ij}^n, (26), implies that the condition $t\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_i^n) \leq \frac{1}{2}$ is satisfied. Since B is an invariant set for the flux $\boldsymbol{f}$, by Lemma 2, B is also an invariant set for the flux $\boldsymbol{g}_j^n$. Since, in addition, B contains the data $(\mathbf{U}_i^n, \mathbf{U}_j^n)$, we conclude that $\overline{\mathbf{U}}_{ij}^{n+1} = \overline{\boldsymbol{v}}(t, \boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) \in B$; see Remark 1. In conclusion, $\mathbf{U}_i^{n+1} \in B$ since $\mathbf{U}_i^{n+1}$ is a convex combination of objects in B. The rest of the proof consists of verifying that (31) indeed implies $1 - \sum_{i\neq j\in\mathscr{I}(S_i^n)} 2d_{ij}^n \frac{\tau}{\mathfrak{m}_i^{n+1}} \geq 0$. □

Corollary 1. *Let $n \in \mathbb{N}$. Assume that τ is small enough so that the CFL condition* (31) *holds for all $i \in \{1{:}I\}$. Let $B \subset \mathscr{A}_{\boldsymbol{f}}$ be a convex invariant set. Assume that $\{\mathbf{U}_i^n \mid i \in \{1{:}I\}\} \subset B$. Then* (i) $\{\mathbf{U}_i^{n+1} \mid i \in \{1{:}I\}\} \subset B$; (ii) $\boldsymbol{u}_h^n \in B$ *and* $\boldsymbol{u}_h^{n+1} \in B$.

Proof. The statement (i) is a direct consequence of Theorem 1. The statement (ii) is a consequence of (18) from Lemma 6. □

Corollary 2. *Let $B \subset \mathscr{A}_{\boldsymbol{f}}$ be a convex invariant set containing the initial data $\boldsymbol{u}_0$. Assume that $\{\mathbf{U}_i^0 \mid i \in \{1{:}I\}\} \subset B$. Let $N \in \mathbb{N}$. Assume that τ is small enough so that the CFL condition* (31) *holds for all $i \in \{1{:}I\}$ and all $n \in \{0{:}N\}$. Then $\{\mathbf{U}_i^n \mid i \in \{1{:}I\}\} \subset B$ and $\boldsymbol{u}_h^n \in B$ for all $n \in \{0{:}N+1\}$.*

Remark 4 (Construction of $\boldsymbol{u}_h^0$). Let $B \subset \mathscr{A}_{\boldsymbol{f}}$ be a convex invariant set containing the initial data $\boldsymbol{u}_0$. If $\boldsymbol{P}_m(\mathscr{T}_h^0)$ is composed of piecewise Lagrange elements, then defining $\boldsymbol{u}_h^0$ to be the Lagrange interpolant of $\boldsymbol{u}_0$, we have $\{\mathbf{U}_i^0 \mid i \in \{1{:}I\}\} \subset B$. Similarly if $\boldsymbol{P}_m(\mathscr{T}_h^0)$ is composed of Bernstein finite elements of degree two and higher, then defining $\boldsymbol{u}_h^0$ to be the Bernstein interpolant of $\boldsymbol{u}_0$ we have $\{\mathbf{U}_i^0 \mid i \in \{1{:}I\}\} \subset B$; see Lai and Schumaker [16, Eq. (2.72)]. In both cases the assumptions of Corollary 2 hold true.

5.2 Discrete Geometric Conservation Law

The ALE scheme (22) preserves constant states. This property is known in the literature as the Discrete Geometric Conservation Law (DGCL).

Corollary 3 (DGCL). *The scheme* (22) *preserves constant states. In particular if $\mathbf{U}_j^n = \mathbf{U}_i^n$ for all $j \in \mathscr{I}(S_i^n)$, then $\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$.*

Proof. The partition of unity property implies that $\sum_{j\in\mathscr{I}(S_i^n)} \boldsymbol{c}_{ij}^n = 0$. Moreover, the definition d_{ij}^n implies that $\sum_{j\in\mathscr{I}(S_i^n)} d_{ij}^n = 0$ (see (23)). Since Lemma 7 implies that

$$\begin{aligned}\mathbf{U}_i^{n+1} = \mathbf{U}_i^n &+ d_{ij}^n(\mathbf{U}_j^n - \mathbf{U}_i^n)\\ &+ \frac{\tau}{\mathfrak{m}_i^{n+1}} \sum_{j\in\mathscr{I}(S_i^n)} ((\mathbf{U}_j^n - \mathbf{U}_i^n)\otimes \mathbf{W}_j^n + \boldsymbol{f}(\mathbf{U}_i^n) - \boldsymbol{f}(\mathbf{U}_j^n))\cdot \boldsymbol{c}_{ij}^n,\end{aligned}$$

it is now clear that if $\mathbf{U}_j^n = \mathbf{U}_i^n$ for all $j \in \mathscr{I}(S_i^n)$, then $\mathbf{U}_i^{n+1} = \mathbf{U}_i^n$. □

Remark 5 (DGCL). Note that although the DGCL seems to be given some importance in the literature, Corollary 3 has no particular significance. It is a direct consequence of the definition of the mass update (21) which is invoked to rewrite the scheme (22) from the conservative form to the equivalent nonconservative form (29). This equivalence is essential to prove the invariant domain property. In other words, *the DGCL is just a consequence of the equivalence of the discrete conservative and nonconservative formulations.*

5.3 Discrete entropy inequality

In this section we prove a discrete entropy inequality which is consistent with the inequality stated in Lemma 5.

Theorem 2. *Let $(\eta,\boldsymbol{q})$ be an entropy pair for* (1)*. Let $n\in\mathbb{N}$ and $i\in\{1{:}I\}$. Assume that all the assumptions of Theorem 1 hold. Then the following discrete entropy inequality holds:*

$$\frac{1}{\tau}\big(\mathfrak{m}_i^{n+1}\eta(\mathbf{U}_i^{n+1})-\mathfrak{m}_i^n\eta(\mathbf{U}_i^n)\big)\le-\sum_{j\in\mathscr{I}(S_i^n)}d_{ij}^n\eta(\mathbf{U}_j^n)$$
$$-\int_{\mathbb{R}^d}\nabla\cdot\bigg(\sum_{j\in\mathscr{I}(S_i^n)}(\boldsymbol{q}(\mathbf{U}_j^n)-\eta(\mathbf{U}_j^n)\mathbf{W}_j^n)\psi_j^n(\boldsymbol{x})\bigg)\psi_i^n(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}.\quad(33)$$

Proof. Let $(\eta,\boldsymbol{q})$ be an entropy pair for the hyperbolic system (1). Let $i\in\{1{:}I\}$ and let $n\in\mathbb{N}$. Then using (32), the CFL condition and the convexity of η, we have

$$\eta(\mathbf{U}_i^{n+1})\le\eta(\mathbf{U}_i^n)\bigg(1-\sum_{i\neq j\in\mathscr{I}(S_i^n)}2d_{ij}^n\frac{\tau}{m_i^{n+1}}\bigg)+\frac{\tau}{m_i^{n+1}}\sum_{i\neq j\in\mathscr{I}(S_i^n)}2d_{ij}^n\eta(\overline{\mathbf{U}}_{ij}^{n+1}).$$

This can also be rewritten as follows:

$$\frac{m_i^{n+1}}{\tau}\big(\eta(\mathbf{U}_i^{n+1})-\eta(\mathbf{U}_i^n)\big)\le\sum_{i\neq j\in\mathscr{I}(S_i^n)}2d_{ij}^n(\eta(\overline{\mathbf{U}}_{ij}^{n+1})-\eta(\mathbf{U}_i^n)).$$

Owing to (7) from Lemma 1, and recalling that the entropy flux of the Riemann problem (27) is $(\boldsymbol{q}(v)-\eta(v)\mathbf{W}_j^n)\cdot\boldsymbol{n}_{ij}^n$ we infer that

$$\eta(\overline{\mathbf{U}}_{ij}^{n+1})\le\tfrac{1}{2}(\eta(\mathbf{U}_i^n)+\eta(\mathbf{U}_j^n))-t\big(\boldsymbol{q}(\mathbf{U}_j^n)-\eta(\mathbf{U}_j^n)\mathbf{W}_j^n-\boldsymbol{q}(\mathbf{U}_i^n)+\eta(\mathbf{U}_i^n)\mathbf{W}_j^n\big)\cdot\boldsymbol{n}_{ij}^n$$

with $t=\|\boldsymbol{c}_{ij}^n\|_{\ell^2}/2d_{ij}^n$. Inserting this inequality in the first one, we have

$$\frac{m_i^{n+1}}{\tau}\big(\eta(\mathbf{U}_i^{n+1})-\eta(\mathbf{U}_i^n)\big)\le\sum_{j\in\mathscr{I}(S_i^n)}d_{ij}^n(\eta(\mathbf{U}_j^n)-\eta(\mathbf{U}_i^n))$$
$$-\sum_{j\in\mathscr{I}(S_i^n)}\|\boldsymbol{c}_{ij}^n\|_{\ell^2}\big(\boldsymbol{q}(\mathbf{U}_j^n)-\boldsymbol{q}(\mathbf{U}_i^n)-(\eta(\mathbf{U}_j^n)-\eta(\mathbf{U}_i^n))\mathbf{W}_j^n\big)\cdot\boldsymbol{n}_{ij}^n.$$

By proceeding as in the proof of Lemma 7, we observe that $\frac{\mathfrak{m}_i^{n+1}-\mathfrak{m}_i^n}{\tau}=\sum_{j\in\mathscr{I}(S_i^n)}\mathbf{W}_j^n\cdot\boldsymbol{c}_{ij}^n$. Then using that $\|\boldsymbol{c}_{ij}^n\|_{\ell^2}\boldsymbol{n}_{ij}^n=\boldsymbol{c}_{ij}^n$, we obtain (33). This concludes the proof. □

6 Numerical tests

In this section, we numerically illustrate the performance of the proposed method using SSP RK3. All the tests have been done with two different codes. One code is written in F95 and uses $\mathbb{P}_1$ Lagrange elements on triangles. The other code is based on deal.ii [2], is written in C++ and uses $\mathbb{Q}_1$ Lagrange elements on quadrangles.

The mesh composed of triangles is obtained by dividing all the quadrangles into two triangles. The same numbers of degrees of freedom are used for both codes.

6.1 Analytical scalar-valued solution

To test the convergence property of the SSP RK3 version of the method, as described in Algorithm 3, we solve the linear transport equation in the domain $D^0 = (0,1)^2$:

$$\partial_t u + \nabla\cdot(\boldsymbol{\beta} u) = 0, \quad u_0(\boldsymbol{x}) = x_1 + x_2, \tag{34}$$

where $\boldsymbol{\beta} = (\sin(\pi x_1)\cos(\pi x_2)\cos(2\pi t), -\cos(\pi x_1)\sin(\pi x_2)\cos(2\pi t))^{\mathsf{T}}$. In both codes the ALE velocity is chosen by setting $\mathbf{W}_i^n = \boldsymbol{\beta}(\boldsymbol{a}_i^n)$, i.e., $\boldsymbol{w}_h^n$ is the Lagrange interpolant of $\boldsymbol{\beta}$ on $\mathscr{T}_h^n$. Notice that there is no issue with boundary condition since $\boldsymbol{\beta}\cdot\boldsymbol{n}_{|\partial D^0} = 0$. We first test the accuracy in time of the algorithm by setting $d_{ij}^n = 0$,

Table 1 Rotation problem (34) with Lagrangian formulation, CFL=1.0

	Without viscosity				With viscosity			
# dofs	$\mathbb{Q}_1$, L^1-norm		$\mathbb{P}_1$, L^1-norm		$\mathbb{Q}_1$, L^1-norm		$\mathbb{P}_1$, L^1-norm	
81	6.46E-04	-	1.76E-03	-	1.31E-02	-	1.13E-02	-
289	1.16E-04	2.48	2.46E-04	2.85	4.28E-03	1.61	3.63E-03	1.64
1089	1.41E-05	3.03	3.23E-05	2.93	1.23E-03	1.80	1.04E-03	1.80
4225	1.76E-06	3.01	4.20E-06	2.94	3.29E-04	1.90	2.78E-04	1.90
16641	2.26E-07	2.96	5.76E-07	2.87	8.50E-05	1.95	7.19E-05	1.95
66049	2.82E-08	3.00	9.57E-08	2.59	2.16E-05	1.97	1.83E-05	1.98

i.e., the viscosity is removed. The computations are done with $CFL = 1$. The error measured in the L^1-norm at time $t = 0.5$ is reported in the left part of Table 1. The third-order convergence in time is confirmed. Note that there is no space error due to the particular choice for the ALE velocity and the initial data.

In the second test we put back the viscosity d_{ij}^n. Notice that the particular choice of the ALE velocity implies that $\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) = |(\boldsymbol{\beta}_i^n - \boldsymbol{\beta}_j^n)\cdot\boldsymbol{n}_{ij}^n|$; hence the viscosity is second-order in space instead of being first-order. This phenomenon makes the algorithm second-order in space (in addition to being conservative and maximum principle preserving). The error in the L^1-norm at time $t = 0.5$ is shown in the right part of Table 1.

6.2 Nonlinear scalar conservation equations

We now test the method on nonlinear scalar conservation equations.

6.2.1 Definition of the ALE velocity

In nonlinear conservation equations, solutions may develop shocks in finite time. In this case, using the purely Lagrangian velocity leads to a breakdown of the method in finite time which manifests itself by a time step that goes to zero as the current time approaches the time of formation of the shock. One way to avoid this breakdown is to use an ALE velocity that is a modified version of the Lagrangian velocity.

We now propose an algorithm to compute an ALE velocity based on Loubère et al. [17]. The only purpose of this algorithm is to be able to run the nonlinear simulations past the time of formation of shocks. We refer the reader to the abundant ALE literature to design other ALE velocities that better suit the reader's goals.

We first deform the mesh by using the Lagrangian motion, i.e., we set $\boldsymbol{a}_{i,\mathrm{Lg}}^{n+1} = \boldsymbol{a}_i^n + \tau \nabla_u \boldsymbol{f}(\mathbf{U}_i^n)$; we recall that $\mathbf{U}_i^n \in \mathbb{R}$ and $\nabla_u \boldsymbol{f}(\mathbf{U}_i^n) \in \mathbb{R}^d$ for scalar equations. Then, given $L \in \mathbb{N} \setminus \{0\}$, we define a smooth version of the Lagrangian mesh by smoothing the position of the geometric Lagrange nodes as follows:

$$\begin{cases} \boldsymbol{a}_i^{n+1,0} := \boldsymbol{a}_{i,\mathrm{Lg}}^{n+1},\ i \in \{1{:}I\} \\ \left(\boldsymbol{a}_i^{n+1,l} := \dfrac{1}{|\mathscr{I}(\mathscr{S}_i)|-1} \displaystyle\sum_{i \neq j \in \mathscr{I}(\mathscr{S}_i)} \boldsymbol{a}_j^{n+1,l-1},\ i \in \{1{:}I\} \right),\ l \in \{1{:}L\} \\ \boldsymbol{a}_{i,\mathrm{Sm}}^{n+1} := \boldsymbol{a}_i^{n+1,L},\ i \in \{1{:}I\}. \end{cases} \tag{35}$$

Finally, the actual ALE motion is defined by $\boldsymbol{a}_i^{n+1} = \omega \boldsymbol{a}_{i,\mathrm{Lg}}^{n+1} + (1-\omega)\boldsymbol{a}_{i,\mathrm{Sm}}^{n+1}, \quad i \in \{1{:}I\}$ where ω is a user-defined constant. In all our computations, we use $\omega = 0.9$ and $L = 2$. As mentioned in [17], a more advanced method consists of choosing ω pointwise by using the right Cauchy-Green strain tensor. We have not implemented this version of the method since the purpose of the tests in the next sections is just to show that the present method works as advertised for any reasonable ALE velocity.

6.2.2 Burgers equation

We consider the inviscid Burgers equation in two space dimensions

$$\partial_t u + \nabla \cdot (\tfrac{1}{2} u^2 \boldsymbol{\beta}) = 0, \quad u_0(\boldsymbol{x}) = \mathbb{1}_{\{\|\boldsymbol{x}\|_{\ell^2}\}}, \tag{36}$$

where $\boldsymbol{\beta} = (1,1)^\intercal$ and $\mathbb{1}_E$ denotes the characteristic function of the set $E \subset \mathbb{R}^d$. The solution to this problem at time $t > 0$ and at $\boldsymbol{x} = (x_1, x_2)$ is given as follows. Assume first that $x_2 \leq x_1$, then define $\alpha = x_1 - x_2$ and let $\alpha_0 = 1 - \frac{t}{2}$. There are three cases. If $\alpha > 1$, then $u(x_1, x_2, t) = 0$.

$$\text{If } \alpha \le \alpha_0, \text{ then } \quad u(x_1,x_2,t) = \begin{cases} \frac{x_2}{t} & \text{if } 0 \le x_2 < t \\ 1 & \text{if } t \le x_2 < \frac{t}{2} + 1 - \alpha \\ 0 & \text{otherwise.} \end{cases} \tag{37}$$

$$\text{If } \alpha_0 < \alpha \le 1, \text{ then } \quad u(x_1,x_2,t) = \begin{cases} \frac{x_2}{t} & \text{if } 0 \le x_2 < \sqrt{2t(1-\alpha)} \\ 0 & \text{otherwise.} \end{cases} \tag{38}$$

If $x_2 > x_1$, then $u(x_1,x_2,t) := u(x_2,x_1,t)$. The computation are done up to $T = 1$ in the initial computational domain $D^0 = (-0.25, 1.75)^2$. The boundary of D^n does not move in the time interval $(0,1)$, i.e., $\partial D^0 = \partial D^n$ for any $n \ge 0$. The results of the convergence tests are reported in Table 2. The solution is computed on a 128×128 mesh. The $\mathbb{Q}_1$ and $\mathbb{P}_1$ meshes at $T = 1$ are shown in Figure 1.

Table 2 Burgers equation, convergence tests, $CFL = 0.1$

	$\mathbb{Q}_1$				$\mathbb{P}_1$			
# dofs	L^2-error		L^1-error		L^2-error		L^1-error	
81	5.79E-01	-	6.00E-01	-	5.80E-01	-	6.17E-01	-
289	4.20E-01	0.46	3.88E-01	0.63	4.43E-01	0.39	4.68E-01	0.40
1089	2.96E-01	0.51	2.32E-01	0.74	3.12E-01	0.51	2.86E-01	0.71
4225	2.14E-01	0.47	1.32E-01	0.82	2.17E-01	0.53	1.55E-01	0.88
16641	1.56E-02	0.45	7.40E-02	0.83	1.23E-01	0.82	7.57E-02	1.04

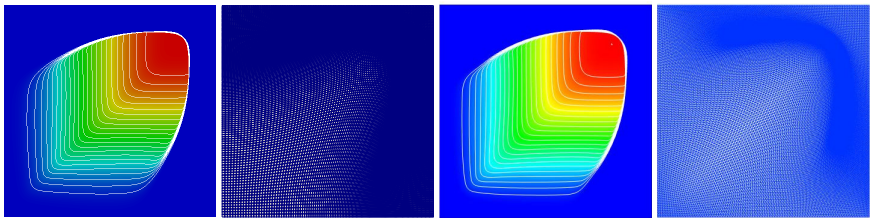

Fig. 1 Burgers equation. Left: $\mathbb{Q}_1$ FEM with 25 contours; Center left: Final $\mathbb{Q}_1$ mesh; Center right: $\mathbb{P}_1$ FEM with 25 contours; Right: Final $\mathbb{P}_1$ mesh.

6.3 Compressible Euler equations

We finish the series of tests by solving the compressible Euler equations in $\mathbb{R}^2$

$$\begin{cases} \partial_t \rho + \nabla\cdot(\rho \boldsymbol{u}) & = 0, \\ \partial_t(\rho \boldsymbol{u}) + \nabla\cdot(\rho \boldsymbol{u}\otimes\boldsymbol{u} + p\mathbb{I}) & = 0, \\ \partial_t E + \nabla\cdot(\boldsymbol{u}(E+p)) & = 0, \end{cases} \tag{39}$$

with the ideal gas equation of state, $p = (\gamma - 1)(E - \frac{1}{2}\rho\|\boldsymbol{u}\|_{\ell^2}^2)$ where $\gamma > 1$, and appropriate initial and boundary conditions. The motion of the mesh is done as described in (35) with $\boldsymbol{a}_{i,\mathrm{Lg}}^{n+1} = \boldsymbol{a}_i^n + \tau \boldsymbol{u}_h^n(\boldsymbol{a}_i^n)$ where $\boldsymbol{u}_h^n$ is the approximate fluid velocity.

We consider the so-called Noh problem, see e.g., Caramana et al. [3, §5]. The computational domain at the initial time is $D^0 = (-1,1)^2$ and the initial data is

$$\rho_0(\boldsymbol{x}) = 1.0, \quad \boldsymbol{u}_0(\boldsymbol{x}) = -\frac{\boldsymbol{x}}{\|\boldsymbol{x}\|_{\ell^2}}, \quad p_0(\boldsymbol{x}) = 10^{-15}. \tag{40}$$

A Dirichlet boundary condition is enforced on all the dependent variables at the boundary of the domain. We use $\gamma = \frac{5}{3}$. The ALE velocity at the boundary of the computational domain is prescribed 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 which happens at $t = \frac{3}{4}$ since the shock moves radially outwards with speed $\frac{1}{3}$.

Table 3 Noh problem, convergence test, $T = 0.6$, $CFL = 0.2$

	$\mathbb{Q}_1$				$\mathbb{P}_1$			
# dofs	L^2-norm		L^1-norm		L^2-norm		L^1-norm	
961	2.60	-	1.44	-	2.89	-	1.71	-
3721	1.81	0.52	8.45E-01	0.77	2.21	0.39	1.09	0.64
14641	1.16	0.64	4.21E-01	1.01	1.42	0.64	5.15E-01	1.08
58081	7.66E-01	0.60	2.10E-01	0.99	9.39E-01	0.59	2.60E-01	0.99
231361	5.21E-01	0.56	1.06E-01	0.98	6.33E-01	0.57	1.28E-01	1.02

The solution to this problem is known. We show in Table 3 the L^1- and the L^2-norm of the error on the density for various meshes which are uniform at $t = 0$: 30×30, 60×60, etc.

7 Concluding remarks

In this paper we have developed a framework for constructing ALE algorithms using continuous finite elements. The method is invariant domain preserving on any mesh in arbitrary space dimension. The methodology applies to any hyperbolic system which has such intrinsic property. If the system at hand has an entropy pair, then the method also satisfies a discrete entropy inequality. The time accuracy of the method can be increased by using SSP time discretization techniques. The equivalence between the conservative and non-conservative formulations implies the that DGCL condition holds (preservation of constant states). The new methods have been tested on a series of benchmark problems and the observed convergence orders and numerical performance are compatible with what is reported in the literature.

References

[1] M. Ainsworth. Pyramid algorithms for Bernstein-Bézier finite elements of high, nonuniform order in any dimension. *SIAM J. Sci. Comput.*, 36(2):A543–A569, 2014.

[2] W. Bangerth, R. Hartmann, and G. Kanschat. deal.II – a general purpose object oriented finite element library. *ACM Trans. Math. Softw.*, 33(4):24/1–24/27, 2007.

[3] E. Caramana, M. Shashkov, and P. Whalen. Formulations of artificial viscosity for multi-dimensional shock wave computations. *J. Comput. Phys.*, 144(1):70–97, 1998.

[4] G.-Q. Chen. Euler equations and related hyperbolic conservation laws. In *Evolutionary equations. Vol. II*, Handb. Differ. Equ., pages 1–104. Elsevier/North-Holland, Amsterdam, 2005.

[5] K. N. Chueh, C. C. Conley, and J. A. Smoller. Positively invariant regions for systems of nonlinear diffusion equations. *Indiana Univ. Math. J.*, 26(2):373–392, 1977.

[6] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]*. Springer-Verlag, Berlin, 2000.

[7] C. Farhat, P. Geuzaine, and 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):669–694, 2001.

[8] L. Ferracina and M. N. Spijker. An extension and analysis of the Shu-Osher representation of Runge-Kutta methods. *Math. Comp.*, 74(249):201–219, 2005.

[9] H. Frid. Maps of convex sets and invariant regions for finite-difference systems of conservation laws. *Arch. Ration. Mech. Anal.*, 160(3):245–269, 2001.

[10] S. Gottlieb, D. I. Ketcheson, and C.-W. Shu. High order strong stability preserving time discretizations. *J. Sci. Comput.*, 38(3):251–289, 2009.

[11] J.-L. Guermond and B. Popov. Fast estimation of the maximum wave speed in the Riemann problem for the euler equations. *J. Comput. Phys.*, 321:908–926, 2016. arXiv:1511.02756, Submitted.

[12] J.-L. Guermond and B. Popov. Invariant domains and first-order continuous finite element approximation for hyperbolic systems. *SIAM J. Numer. Anal.*, 2016. arXiv:1509.07461, in press.

[13] J.-L. Guermond, B. Popov, L. Saavedra, and Y. Yang. Invariant domains preserving ale approximation of hyperbolic systems with continuous finite elements. *SIAM J. Numer. Anal.*, 39(2):A385–A414, 2017.

[14] I. Higueras. Representations of Runge-Kutta methods and strong stability preserving methods. *SIAM J. Numer. Anal.*, 43(3):924–948, 2005.

[15] D. Hoff. Invariant regions for systems of conservation laws. *Trans. Amer. Math. Soc.*, 289 (2):591–610, 1985.

[16] M.-J. Lai and L. L. Schumaker. *Spline functions on triangulations*, volume 110 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 2007.

[17] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: A reconnection-based arbitrary-Lagrangian-Eulerian method. *Journal of Computational Physics*, 229(12):4724 – 4761, 2010.

[18] O. Pironneau. On the transport-diffusion algorithm and its applications to the Navier- Stokes equations. *Numer. Math.*, 38:309–332, 1982.

[19] O. Pironneau. *Finite element methods for fluids*. John Wiley & Sons, Ltd., Chichester; Masson, Paris, 1989. Translated from the French.

[20] E. F. Toro. *Riemann solvers and numerical methods for fluid dynamics*. Springer-Verlag, Berlin, third edition, 2009. A practical introduction.