Communications on Applied Mathematics and Computation (2023) 5:923–945
https://doi.org/10.1007/s42967-021-00165-y

ORIGINAL PAPER

Second-Order Invariant Domain Preserving ALE Approximation of Euler Equations

Jean-Luc Guermond[1] · Bojan Popov[1] · Laura Saavedra[2]

Received: 3 December 2020 / Revised: 20 August 2021 / Accepted: 21 August 2021 /
Published online: 23 December 2021
© The Author(s) 2021

Abstract

An invariant domain preserving arbitrary Lagrangian-Eulerian method for solving nonlinear hyperbolic systems is developed. The numerical scheme is explicit in time and the approximation in space is done with continuous finite elements. The method is made invariant domain preserving for the Euler equations using convex limiting and is tested on various benchmarks.

Keywords Conservation equations · Hyperbolic systems · Arbitrary Lagrangian-Eulerian · Moving meshes · Invariant domains · High-order method · Convex limiting · Finite element method

Mathematics Subject Classification 35L52 · 65M60 · 35Q31

1 Introduction

An arbitrary Lagrangian-Eulerian (ALE) technique is developed in this paper in an attempt to combine the advantages of classical kinematic descriptions, and to minimize their drawbacks as much as possible. The computational mesh given by an ALE formulation, allows us to handle greater distortions of the fluid with more precision than that obtained by a purely Eulerian formulation. In our previous work, Guermond et al. [7, 9], we investigated the artificial viscosity problem with hyperbolic systems using an ALE formulation, continuous finite elements and explicit time stepping. In this work, we propose a first-order artificial viscosity that does not depend on any ad hoc parameters and that leads to precise

✉ Laura Saavedra
laura.saavedra@upm.es

Jean-Luc Guermond
guermond@math.tamu.edu

Bojan Popov
popov@math.tamu.edu

[1] Department of Mathematics, Texas A&M University, College Station, USA

[2] Departamento de Matemática Aplicada a la Ingeniería Aeroespacial, Universidad Politécnica de Madrid, Madrid, Spain

invariant domain properties and entropy inequalities. Combining a low-order method with an entropy consistent high-order method via a convex limiting process, we describe a high-order method that preserves the invariant domains.

We consider the following hyperbolic system in conservative form:

$$\begin{cases} \partial_t u + \nabla{\cdot}f(u) = 0, (x,t) \in \mathbb{R}^d{\times}\mathbb{R}_+, \\ u(x,0) = u_0(x),\ x \in \mathbb{R}^d, \end{cases} \tag{1}$$

where the dependent variable u is $\mathbb{R}^m$-valued, the flux is $f \in C^1(\mathbb{R}^m;\mathbb{R}^{m\times d})$, and u_0 is any admissible initial data. For simplicity in the presentation of the theoretical results, we consider periodic boundary conditions or assume that u_0 is constant outside a compact set. Some examples of well-known equations that can be written in this general form are the Euler equations, shallow water equations, and Burgers' equation.

The main objective of this paper is to present a simplified version of our high-order ALE schemes for Euler equations, mainly focusing on the computational aspects and key features of the ALE motion description and omitting the details of the precise mathematical formulation necessary to rigorously demonstrate the properties of the methods. For this reason, this paper describes in details the algorithm and the implementation of our high-order invariant domain preserving method in a finite element code. One of the key issues in the definition of the ALE motion is the mesh optimization process. We solve it in two stages: (i) high-order reconstruction of the Lagrangian velocity; (ii) smoothing of the mesh. The smoothing stage is driven by information provided by a simulation. We define a blending parameter to smooth the mesh using an adaptation of the method proposed by Loubère et al. [11] which has also been used by other research groups (see, e.g., Boscheri and Balsara [1], Boscheri et al. [2]). There are other ways to approximate the ALE movement, where the ALE velocity is defined not only using the velocity of the fluid, but also more information from the simulation, such as the gradients of the variables or shock positions, see e.g., Dobrev et al. [4], Zhao et al. [15], but we do not explore this direction in this paper. One novelty here, with respect to Guermond et al. [7, 9], is that we present new numerical tests in which we compare results obtained using ALE and Eulerian formulations of our high-order scheme.

The remainder of this paper is organized as follows. The ALE motion and its approximation are described in details in Sect. 2. Specifically, we present the main difference between purely Lagrangian and ALE descriptions and describe the finite element spaces used to approximate the ALE motion as well as our method of computing the ALE velocity. To compute the ALE velocity, we reconstruct the approximation of the Lagrangian velocity and then apply a smoothing procedure. The general ALE scheme is introduced in Sect. 3, where the low-order invariant domain preserving method and the high-order (possibly invariant domain violating) method are briefly described. The convex limiting technique, whose purpose is to make the high-order method invariant domain preserving, is described in Sect. 4. The new ALE method is illustrated numerically in Sect. 5 on various benchmark problems and compared with the same method in Eulerian coordinates.

2 ALE Motion and Its Approximation

We consider an ALE motion $X_{\mathrm{A}} : \mathbb{R}^d \times \mathbb{R}_+ \to \mathbb{R}^d$ as a uniformly Lipschitz mapping, and define the spatial description of the associated velocity as

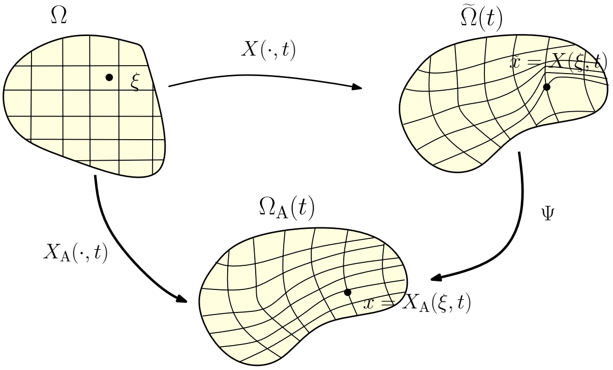

Fig. 1 Description of an ALE motion and its relation with the Lagrangian motion of a body

$$\mathbf{v}_{\mathrm{A}}(x,t) := \partial_t X_{\mathrm{A}}(X^{-1}_{\mathrm{A},t}(x),t), \tag{2}$$

where $X_{\mathrm{A},t} : \mathbb{R}^d \to \mathbb{R}^d$, $X_{\mathrm{A},t}(x) := X_{\mathrm{A}}(x,t)$. In the notation used herein, the subindex A stands for arbitrary.

Remark 1 Arbitrary Lagrangian Eulerian vs. Lagrangian motion (Fig. 1). To illustrate the differences between ALE and Lagrangian descriptions, we consider the transport equation

$$\begin{cases} \partial_t u + \mathbf{v}\cdot\nabla u = 0 & \text{for } (x,t)\in\mathbb{R}^d\times\mathbb{R}_+, \\ u(x,0) = u_0(x) & \text{for } x\in\mathbb{R}^d, \end{cases} \tag{3}$$

where $\mathbf{v}$ is the velocity of the fluid.

By computing the time derivative of u along the characteristic curves of the Lagrangian motion, we get

$$\frac{\mathrm{d}u}{\mathrm{d}t}(X(\xi,t),t) = \frac{\partial u}{\partial t}(X(\xi,t),t) + \mathbf{v}(X(\xi,t),t)\cdot\nabla u(X(\xi,t),t) = 0. \tag{4}$$

If we discretize (4) using the explicit Euler method, we obtain

$$u^{n+1}(x^{n+1}) = u^n(x^n),$$

where $x^i = X(\xi,t^i), i = n, n+1$; that means that the solution is translated along the characteristic curves.

By writing the material derivative along the characteristic curves of the ALE motion, we get

$$\begin{aligned}\frac{\mathrm{d}u}{\mathrm{d}t}(X_{\mathrm{A}}(\xi,t),t) =& \frac{\partial u}{\partial t}(X_{\mathrm{A}}(\xi,t),t) + \mathbf{v}_{\mathrm{A}}(X_{\mathrm{A}}(\xi,t),t)\cdot\nabla u(X_{\mathrm{A}}(\xi,t),t) \\ =&(\mathbf{v}_{\mathrm{A}} - \mathbf{v})(X_{\mathrm{A}}(\xi,t),t)\cdot\nabla u(X_{\mathrm{A}}(\xi,t),t).\end{aligned} \tag{5}$$

Therefore, using the explicit Euler method, we obtain

$$u^{n+1}(x^{n+1}) = u^n(x^n) + \Delta t(\mathbf{v}_{\mathrm{A}}(x^n) - \mathbf{v}(x^n))\cdot\nabla u^n(x^n), \tag{6}$$

where $X_{\mathrm{A}}^{n+1}(\xi) = x^{n+1}$. The new solution is the solution at the previous time step moved along the characteristics plus a correction term due to the difference between the ALE velocity and the fluid velocity.

2.1 Geometric Finite Elements and Mesh

We now present the discrete setting of the ALE motion. Let $\mathcal{T}_h^n$, $h > 0$, $n = 0, 1, \cdots, N$ be a sequence of shape-regular matching meshes, where $\mathcal{T}_h^0$ denotes the initial mesh and $\mathcal{T}_h^n$ is the deformed mesh by means of the user-defined ALE motion at time t^n. Given mesh $\mathcal{T}_h^n$, we denote by D^n the computational domain generated by $\mathcal{T}_h^n$. To simplify the presentation, we assume that our meshes are formed by triangles in the case of two space dimensions and tetrahedra in the case of three space dimensions. In Guermond et al. [7], a more general setting for the configuration of the meshes that can be used was described.

We introduce the reference Lagrangian finite element $(\widehat{K}, \widehat{P}^{\mathrm{geo}}, \widehat{\Sigma}^{\mathrm{geo}})$, where $\widehat{K}$ is the reference element, $\widehat{P}^{\mathrm{geo}}$ is the reference polynomial space, and $\widehat{\Sigma}^{\mathrm{geo}}$ is the set of linear forms that define the degrees of freedom. We denote by $\{\widehat{a}_i\}_{i\in\{1:\mathcal{N}^{\mathrm{geo}}\}}$ and $\{\widehat{\theta}_i^{\mathrm{geo}}\}_{i\in\{1:\mathcal{N}^{\mathrm{geo}}\}}$ the Lagrange nodes of $\widehat{K}$ and the associated Lagrange shape functions, respectively, where $\mathcal{N}^{\mathrm{geo}} := \dim \widehat{P}^{\mathrm{geo}}$. Let $\{a_i\}_{i\in\{1:\mathcal{V}^{\mathrm{geo}}\}}$ be the collection of all Lagrange nodes of mesh $\mathcal{T}_h^n$ and let $\mathrm{j}^{\mathrm{geo}} : \mathcal{T}_h^n \times \{1:\mathcal{N}^{\mathrm{geo}}\} \longrightarrow \{1:\mathcal{V}^{\mathrm{geo}}\}$ be the geometric connectivity array (assumed to be independent of the time). Then the geometric transformation $T_K^n : \widehat{K} \to K \in \mathcal{T}_h^n$ is defined by

$$T_K^n(\widehat{x}) = \sum_{i\in\{1:\mathcal{N}^{\mathrm{geo}}\}} a^n_{\mathrm{j}^{\mathrm{geo}}(K,i)} \widehat{\theta}_i^{\mathrm{geo}}(\widehat{x}). \tag{7}$$

We now define the finite-dimensional spaces based on the geometric Lagrange finite elements

$$P^{\mathrm{geo}}(\mathcal{T}_h^n) := \{v \in \mathcal{C}^0(D^n;\mathbb{R}) | v_{|K} \circ T_K^n \in \widehat{P}^{\mathrm{geo}}, \forall K \in \mathcal{T}_h^n\} \tag{8}$$

and the following vector-valued spaces which we use to represent the ALE motion:

$$\boldsymbol{P}^{\mathrm{geo}}(\mathcal{T}_h^n) := [P^{\mathrm{geo}}(\mathcal{T}_h^n)]^d. \tag{9}$$

By considering the approximate ALE motion mapping $X_{\mathrm{A}}^{n,n+1} : D^n \to D^{n+1}$ as a function of $\boldsymbol{P}^{\mathrm{geo}}(\mathcal{T}_h^n)$, we can write

$$X_{\mathrm{A}}^{n,n+1}(x) = \sum_{i\in\{1:\mathcal{V}^{\mathrm{geo}}\}} X_{\mathrm{A}}^{n,n+1}(a_i^n)\varphi_i^{\mathrm{geo},n}(x), \quad x \in D^n. \tag{10}$$

That is, the approximate ALE motion is fully described by the position of geometric Lagrange nodes at time t^{n+1}: $a_i^{n+1} := X_{\mathrm{A}}^{n,n+1}(a_i^n)$.

In the numerical simulations presented at the end of this paper, we use triangles and $P^{\mathrm{geo}}(\mathcal{T}_h^n)$ is one of the following polynomial spaces: $\mathbb{P}_i$, $i = 1, 2, 4$.

2.2 Approximation of the ALE Motion

Although the ALE motion of the mesh, or equivalently the ALE velocity field, is user-defined, we give in this section some details on how we compute the ALE velocity which

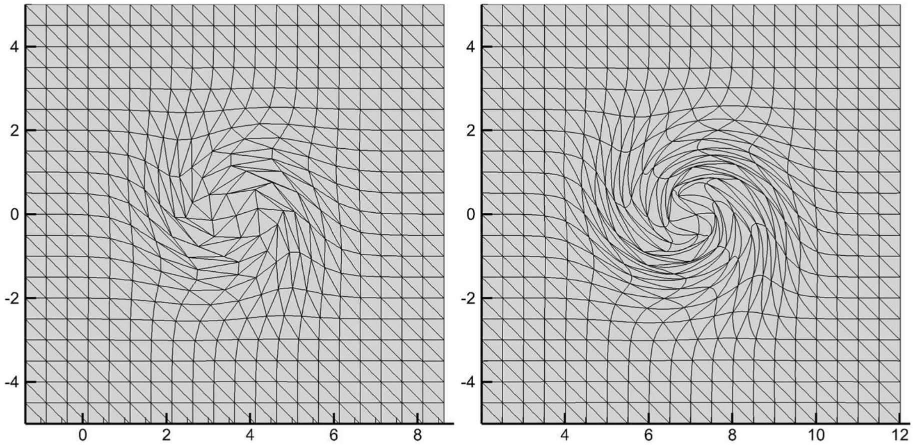

Fig. 2 Isentropic vortex. Meshes plotted at the instant before collapsing. Left panel: mesh at $t = 1.81$ obtained using $v^n \in \mathbb{P}_1$; right panel: mesh at $t = 3.48$ using $v^n \in \mathbb{P}_4$

may be useful to the reader. All the numerical results presented in this paper have been obtained using the techniques described below.

We compute the new position of the geometric Lagrange nodes of the mesh using an explicit Euler step, that is

$$X^{n,n+1}(a_i^n) = a_i^{n+1} = a_i^n + \Delta t v_{\mathrm{A}}^n(a_i^n), \tag{11}$$

where v_{A}^n is the approximate ALE velocity at time t^n and $\Delta t := t^{n+1} - t^n$ is the time step.

We implement the calculation of the ALE velocity v_{A}^n in two stages: reconstruction and smoothing of the ALE velocity. A brief description of the two stages is as follows.

2.2.1 Reconstruction of the ALE Velocity

One of the motivations for using ALE methods in compressible hydrodynamics is that by making the mesh motion as close as possible to the actual fluid motion, one can significantly reduce the effects of the artificial viscosity. Hence, at each time t^n, the approximate solution of the hyperbolic system should be used to reconstruct the fluid velocity, the ALE velocity should be defined from that. Let v^n be the approximated velocity of the fluid obtained by solving our hyperbolic system. The easiest way to define the ALE velocity is $v_{\mathrm{A}}^n = v^n$ which make the scheme Lagrangian. However, if we use low-order finite elements to solve our system, v_{A}^n, and therefore $X_{\mathrm{A}}^{n,n+1}$, would be also in a low-order space and would not represent well smooth mesh motions with vorticity. It has been reported in the literature and 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. To illustrate this observation, we solve an isentropic vortex problem[1] where the analytical velocity is known, and we move the mesh using two different approximations of this velocity: piecewise linear and piecewise quadratic. In Fig. 2, we plot the mesh at the instant

[1] In Sect. 5.1 the analytical solution of the isentropic vortex problem is described.

before collapsing, which occurs at $t = 1.81$ when using linear polynomial approximation of the velocity and at $t = 3.48$ when using high-order (quadratic) polynomial approximation of the velocity.

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 the ALE velocity from v^n using an algorithm called the butterfly subdivision algorithm, initially described in Dyn et al. [5] for surface reconstruction. We refer the reader to our previous work [9] for more details on how this algorithm is used to reconstruct the ALE velocity.

Once the velocity is reconstructed, we compute the new position of the mesh nodes using

$$a^{n+1}_{i,\mathrm{Lag}} = a^n_i + \Delta t v^n_{\mathrm{Lag}}(a^n_i), \tag{12}$$

where $v^n_{\mathrm{Lag}} \in P^{\mathrm{geo}}(\mathcal{T}^n_h)$ is the reconstructed velocity. The notation $_{\mathrm{Lag}}$ is used to indicate that nodes are moved following the fluid motion as no mesh smoothing is applied yet.

2.2.2 Smoothing of the ALE Velocity (Maintaining the Connectivity)

When the fluid undergoes large deformations, for instance, when a vortex or a shock wave is formed, a purely Lagrangian algorithm can experience a loss of accuracy or the mesh will eventually tangle. Therefore, it is not possible to describe the computational domain motion without doing remeshing operations. One way to avoid this breakdown is to use in the algorithm an ALE velocity which is a smoothed version of the Lagrangian velocity.

We propose a method that involves blending the Lagrangian velocity using an averaging technique. The blending is done through a parameter that controls the local deformation of the cells; the definition of this parameter is inspired by a technique proposed by Loubére et al. [11]. From the Lagrangian motion given by (12), we compute an averaged version $a^{n+1}_{i,\mathrm{Sm}}$ as follows:

$$a^{n+1}_{i,\mathrm{Sm}} := \frac{1}{\mathrm{card}(\mathcal{I}(i)) - 1} \sum_{j \in \mathcal{I}(i), j \neq i} a^{n+1}_{j,\mathrm{Lag}}, \tag{13}$$

where $i \in \mathcal{V}^{\mathrm{geo}}$, and $\mathcal{I}(i)$ denotes the collection of indices of the neighboring nodes of i. This smoothing technique is not sufficient to avoid mesh tangling when the fluid motion is complex; a more sophisticated curvilinear mesh smoothing is a topic of our ongoing research. Finally, the actual position of the geometric nodes is defined by

$$a^{n+1}_i := \omega_i a^{n+1}_{i,\mathrm{Lag}} + (1 - \omega_i) a^{n+1}_{i,\mathrm{Sm}}, \tag{14}$$

where $\omega_i \in [0, 1]$ is a blending parameter.

To define the blending parameter, we first construct first the Jacobian matrix of the mapping $x \mapsto x + \Delta t v^n_{\mathrm{Lag}}(x)$ in each cell $K \in \mathcal{T}^n_h$ by $\mathbb{F}_{|K}(x) := \mathbb{I} + \Delta t \nabla v^n_{\mathrm{Lag}}(x)$. Then, we compute the right Cauchy-Green strain tensor $\mathbb{F}|^{\mathsf{T}}_K \mathbb{F}_{|K}$ and the two eigenvalues $\lambda_{1,K}(a^n_i)$, $\lambda_{2,K}(a^n_i)$ at all the geometric Lagrange nodes $a^n_i \in K$, with the convention $\lambda_{1,K}(a^n_i) \leqslant \lambda_{2,K}(a^n_i)$. We define $\mathcal{T}_i := \{K \in \mathcal{T}^n_h \mid a^n_i \in K\}$ for all $i \in \mathcal{V}^{\mathrm{geo}}$, and compute the blending parameter as follows:

$$\omega^n_i := \left(\mathrm{card}(\mathcal{T}_i)^{-1} \sum_{K \in \mathcal{T}_i} \lambda_{1,K}(a^n_i) / \lambda_{2,K}(a^n_i) \right)^{\frac{1}{s}}, \tag{15}$$

where $s \in \mathbb{N}$. In each application, we will specify this natural number, which can be larger if we want our ALE motion to be closer to the Lagrangian motion. Finally, after the two stages, the ALE motion is given by

$$X_{\mathrm{A}}^{n,n+1}(x) = \sum_{i \in \mathcal{V}^{\mathrm{geo}}} a_i^{n+1} \varphi_i^n(x), \tag{16}$$

and the ALE velocity of the geometric nodes is given by

$$v_{\mathrm{A}}^n(a_i^n) = (a_i^{n+1} - a_i^n)/\Delta t. \tag{17}$$

3 ALE Scheme

We describe in this section two ALE approximations of (1). One is invariant domain preserving and entropy satisfying, while the other is at least second-order accurate in space but may violate the invariant domain property. We use continuous finite elements and explicit time stepping.

If $\mathbf{u}$ is a weak solution of the hyperbolic system (1), then the following result that we demonstrated in Guermond et al. [7] is the main motivation for the ALE formulation.

Lemma 1 *The following identity holds in the distribution sense* (*in time*) *over the interval* $[0, t^*]$ *for every function* $\psi \in C_0^0(\mathbb{R}^d; \mathbb{R})$:

$$\partial_t \int_{\mathbb{R}^d} \mathbf{u}(x,t)\varphi(x,t)\mathrm{d}x = \int_{\mathbb{R}^d} \nabla\cdot(\mathbf{u}(x,t) \otimes \mathbf{v}_{\mathrm{A}} - \mathbf{f}(\mathbf{u}))\varphi(x,t)\mathrm{d}x, \tag{18}$$

where $\varphi(x,t) := \psi(X_{\mathrm{A},t}^{-1}(x))$.

3.1 Approximating Finite Elements

The construction of the approximate solution of (1) is based on a reference finite element $\{(\widehat{K}, \widehat{P}, \widehat{\Sigma})\}$. It is important to note that $\widehat{P}^{\mathrm{geo}}$ and $\widehat{P}$ are different objects. $(\widehat{K}, \widehat{P}^{\mathrm{geo}}, \widehat{\Sigma}^{\mathrm{geo}})$ 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 for example Lai and Schumaker [10, Chap. 2] or another modal finite element. In our code, we use Lagrange finite elements; specifically we always take $\widehat{P} = \mathbb{P}_1$ but we use $\widehat{P}^{\mathrm{geo}} = \mathbb{P}_k$ with $k \in \{1, 2, 4\}$.

The shape functions of the reference element are denoted by $\{\widehat{\theta}_i\}_{i \in \mathcal{N}}$. Given $\mathcal{T}_h^n$, we define $P(\mathcal{T}_h^n) := \{v \in \mathcal{C}^0(D^n; \mathbb{R}) \mid v_{|K} \circ T_K^n \in \widehat{P},\ \forall K \in \mathcal{T}_h^n\}$ and introduce the vector-valued spaces

$$P_m(\mathcal{T}_h^n) := [P(\mathcal{T}_h^n)]^m, \qquad \text{and} \qquad P_d(\mathcal{T}_h^n) := [P(\mathcal{T}_h^n)]^d. \tag{19}$$

The global shape functions in $P(\mathcal{T}_h^n)$ are denoted by $\{\psi_i^n\}_{i \in \mathcal{V}}$. Recall that these functions form a basis of $P(\mathcal{T}_h^n)$. We denote by $\mathrm{j} : \mathcal{N} \times \mathcal{T}_h^n \longrightarrow \mathcal{V}$ the connectivity array associated with the global shape functions $\{\psi_i^n\}_{i \in \mathcal{V}}$. This array, which we assume to be independent of n, is defined such that $\psi_{\mathrm{j}(i,K)}^n(x) = \widehat{\theta}_i((T_K^n)^{-1}(x))$, for all $i \in \mathcal{N}$ and for all $K \in \mathcal{T}_h^n$. For any $i \in \mathcal{V}$, $\mathcal{I}(i)$ is the collection of indices of the shape functions whose support has a nontrivial intersection with the shape function ψ_i; that is, we set $\mathcal{I}(i) = \{j \in \mathcal{V} \mid |\mathrm{supp}(\psi_i^n \psi_j^n)| \neq 0\}$

where for any measurable set $E \subset D^n$, $|E|$ denotes the measure of E. Henceforth we call the connectivity graph of $P(\mathcal{T}_h^n)$, the graph $(\mathcal{V}, \mathcal{E})$ where the vertices are all 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 $\mathsf{j} : \mathcal{N} \times \mathcal{T}_h^n \longrightarrow \mathcal{V}$ does not depend on n.

The solution of (1) is approximated in $P_m(\mathcal{T}_h^n)$, so we can write

$$\mathbf{u}(x, t_n) \approx u_h^n(x) := \sum_{i=1}^{\mathcal{V}} \mathbf{U}_i^n \psi_i^n(x), \forall x \in D^n. \tag{20}$$

3.2 Generic Algorithm

We introduce in this section a generic algorithm to obtain u_n^{n+1}, the approximate solution of (1) at time t^{n+1}. This algorithm was introduced in Guermond et al. [7] and was sightly modified in Guermond et al. [9] to allow the ALE velocity to be represented with a polynomial degree much higher than that used to approximate the solution of (1). A detailed description of this method and its conservation properties can be found in those references; here we describe only the main steps of the method.

Let $(\mathfrak{m}_i^0)_{i \in \mathcal{V}}$ be the approximation of the mass of the shape functions at time t^0 defined by $\mathfrak{m}_i^0 := \int_{D^0} \varphi_i^0(x)\,\mathrm{d}x$. Let $u_{h0} := \sum_{i \in \mathcal{V}} \mathbf{U}_i^0 \varphi_i^0 \in P_m(\mathcal{T}_h^0)$ be a reasonable approximation of the initial data u_0, and let $(\mathfrak{m}_i^n)_{i \in \mathcal{V}}$ be the approximations of the mass of the shape functions at time t^n. If N is the number of time steps, for each $n \in \{0, 1, \cdots, N\}$, we do the following steps.

(i) Compute the ALE velocity v_{A}^n. This can be done following the stages described in Sect. 2.2.

(ii) Move the mesh: $a_i^{n+1} = a_i^n + \Delta t v_{\mathrm{A}}^n(a_i^n)$.

(iii) Approximate the ALE velocity, $\boldsymbol{w}^n := \Pi_h(\boldsymbol{w}^n) \in P_d(\mathcal{T}_h)$, where $\Pi_h : P_d^{\mathrm{geo}}(\mathcal{T}_h) \to P_d(\mathcal{T}_h)$ is the Lagrange interpolation operator. We henceforth set $\boldsymbol{w}_h^n = \sum_{i=1}^{N} \mathbf{W}_i^n \psi_i^n \in P_d(\mathcal{T}_h^n)$.

(iv) Update the mass matrix

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

As explained in Guermond et al. [7, Sect. 4.4.2], to be able to use higher-order SSP time stepping techniques and be both conservative and invariant domain preserving, we define $\mathfrak{m}_i^{n+1}$ by approximating the identity $\partial_t \int_{D(t)} \varphi_i(x,t)\,\mathrm{d}x = \int_{D(t)} \varphi_i(x,t) \nabla \cdot \boldsymbol{w}^n(x)\,\mathrm{d}x$ (which is a consequence of Liouville's theorem), with the forward Euler method to get (21).

(v) Compute u_h^{n+1} using

$$\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1} - \mathfrak{m}_i^n \mathbf{U}_i^n}{\Delta t} + \sum_{j \in \mathcal{I}(i)} (f(\mathbf{U}_j^n) - \mathbf{U}_j^n \otimes \mathbf{W}_j^n) \cdot \boldsymbol{c}_{ij}^n - d_{ij}^n(\mathbf{U}_j^n - \mathbf{U}_i^n) = 0, \tag{22}$$

where

$$\boldsymbol{c}_{ij}^n := \int_{D^n} \nabla\psi_j^n(x)\psi_i^n(x)\,\mathrm{d}x, \tag{23}$$

and d_{ij}^n is an artificial viscosity for the pair of degrees of freedom (i, j), that is clearly defined in Sects. 3.3 and 3.4. We call d_{ij}^n the graph viscosity since this coefficient only involves the connectivity graph of $P(\mathcal{T}_h^n)$. We henceforth assume that $d_{ij}^n = 0$ if $j \notin \mathcal{I}(i)$ and

$$d_{ij}^n \geqslant 0,\ \text{if } i \neq j, \quad d_{ij}^n = d_{ji}^n, \quad \text{and} \quad d_{ii}^n := \sum_{j\in\mathcal{I}(i)\backslash\{i\}} -d_{ji}^n. \tag{24}$$

Notice that d_{ii}^n does not really need to be defined for (24) to make sense; this quantity is nevertheless introduced to shorten the definition of the CFL number.

Remark 2 Recall that the initialization is done by setting $\mathfrak{m}_i^0 := \int_{D^0} \varphi_i^0(x)\,\mathrm{d}x$. At this point we assume again that the user-defined ALE velocity field $\boldsymbol{w}^n$ is reasonable or that Δt is small enough that $\mathfrak{m}_i^{n+1}$ is positive.

Remark 3 In (22), $\frac{\mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1}-\mathfrak{m}_i^n\mathbf{U}_i^n}{\Delta t}$ is the forward Euler approximation of the left-hand side in (18). Notice that we have replaced the consistent mass matrix by the approximate lumped mass matrix to approximate the time derivative. The expression $\sum_{j\in\mathcal{I}(i)}(\mathbf{U}_j^n \otimes \mathbf{W}_j^n - \mathbf{f}(\mathbf{U}_j^n)) \cdot \boldsymbol{c}_{ij}^n$ is just the Galerkin approximation of the right-hand side of (18).

3.3 First-Order Viscosity: GMS-GV Method

We define in this section the artificial graph viscosity that makes the method described in the previous section to be invariant domain preserving and entropy satisfying. The method is called GMS-GV because it is based on the guaranteed maximum speed (GMS) and first-order graph viscosity (GV); see Guermond and Popov [6], Guermond et al. [7, 8].

We consider the 1D Riemann problem:

$$\partial_t v + \partial_x(\boldsymbol{g}_j^n(v)\cdot\boldsymbol{n}_{ij}^n) = 0, \quad (x,t) \in \mathbb{R}\times\mathbb{R}_+, \quad v(x,0) = \begin{cases} \mathbf{U}_i^n, & \text{if } x < 0, \\ \mathbf{U}_j^n, & \text{if } x > 0, \end{cases} \tag{25}$$

where $\boldsymbol{g}_j^n(v) := f(v) - v \otimes \mathbf{W}_j^n$ and $\boldsymbol{n}_{ij}^n = \boldsymbol{c}_{ij}^n/\|\boldsymbol{c}_{ij}^n\|_{\ell^2}$. The following quantity is an upper bound on the extreme left and right wave speeds in (25):

$$\lambda_{\max}(\boldsymbol{g}_j^n, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) = \max(|\lambda_{\mathrm{L}}(f, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) - \mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|, |\lambda_{\mathrm{R}}(f, \boldsymbol{n}_{ij}^n, \mathbf{U}_i^n, \mathbf{U}_j^n) - \mathbf{W}_j^n\cdot\boldsymbol{n}_{ij}^n|).$$

We now set

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

The following theorem was proved by Guermond et al. [7]:

Theorem 1 *Let $n > 0$ and $\mathbf{U}_i^{L,n+1}$ be given by* (21) *and* (23) *with the viscosity given by* (26). *Then, the total mass $\sum_{i\in\{1:\mathcal{V}\}} \mathfrak{m}_i^{n+1}\mathbf{U}_i^{n+1}$ is conserved. Provided that Δt is small enough that $\mathfrak{m}_i^{n+1} > 0$ and $(1 - \frac{2\Delta t}{\mathfrak{m}_i^{n+1}}|d_{ii}^{L,n}|) \geqslant 0$, the following properties are satisfied.*

(i) *Local invariance*: $\boldsymbol{U}_i^{L,n+1} \in \mathrm{Conv}\{\overline{\boldsymbol{U}}(\boldsymbol{U}_i^{L,n}, \boldsymbol{U}_j^{L,n}) \mid j \in \mathcal{I}(S_i)\}$.

(ii) *Global invariance. Let A be a convex invariant set. Assume* $\boldsymbol{U}_0 \in A$, *then* $\boldsymbol{U}_i^{L,n+1} \in A$ *for all* $n \geqslant 0$. *The scheme preserves all the convex invariant sets.*

(iii) *Discrete entropy inequality for any entropy pair* $(\eta, \boldsymbol{q})$:

$$\frac{1}{\Delta t}\left(\mathfrak{m}_i^{n+1}\eta(\mathbf{U}_i^{L,n+1}) - \mathfrak{m}_i^n\eta(\mathbf{U}_i^{L,n})\right)$$
$$\leqslant - \sum_{j\in\mathcal{I}(S_i^n)} d_{ij}^n \eta(\mathbf{U}_j^{L,n}) - \int_{\mathbb{R}^d} \nabla\cdot\left(\sum_{j\in\mathcal{I}(i)} (\boldsymbol{q}(\mathbf{U}_j^{L,n}) - \eta(\mathbf{U}_j^{L,n})\mathbf{W}_j^n)\psi_j^n(x)\right)\psi_i^n(x)\,\mathrm{d}x.$$

3.4 Second-Order Viscosity: GMS-EV Method

We now describe a technique that is formally high-order accurate in space but may be invariant domain violating. Let $(\eta(\mathbf{U}), \mathbf{F}(\mathbf{U}))$ be an entropy pair for the hyperbolic system. Then

$$(\eta(\mathbf{U}), \mathbf{G}(\mathbf{U}) := \mathbf{F}(\mathbf{U}) - \eta(\mathbf{U})\mathbf{W}) \tag{27}$$

is an entropy pair for the flux $\mathbf{g}(\mathbf{U}) = f(\mathbf{U}) - \mathbf{U} \otimes \mathbf{W}$. The relation

$$\nabla \cdot (\mathbf{G}(\mathbf{u})) = (\nabla\eta(\mathbf{u}))^{\mathrm{T}}\nabla\mathbf{g}(\mathbf{u}) \tag{28}$$

holds for $\mathbf{u}$ the exact solution of the hyperbolic system. We define the entropy commutator

$$N_i^n := \sum_{j\in\mathcal{I}(S_i^n)} \left(\mathbf{G}(\mathbf{U}_j^n, \mathbf{W}_i^n) - (\nabla\eta(\mathbf{U}_i^n))^{\mathrm{T}}\mathbf{g}(\mathbf{U}_j^n, \mathbf{W}_i^n)\right) \cdot \mathbf{c}_{ij}, \tag{29}$$

which measures how well the approximate solution verifies the relation (28). From (29) we construct a normalized entropy residual $R_i^n \in [0, 1]$ (see details in Guermond et al. [9, Sect. 3.4]) and the entropy viscosity (EV) is defined by

$$d_{ij}^{H,n} := d_{ij}^{L,n}\max(R_i^n, R_j^n)), i \neq j, \text{ and } d_{ii}^{H,n} := - \sum_{j\in\mathcal{I}(S_i^n)\backslash\{i\}} d_{ij}^{H,n}. \tag{30}$$

In view of this definition of the high-order viscosity, we call the method GMS-EV (guaranteed maximum speed entropy viscosity).

4 Convex Limiting Technique: a Second-Order Invariant Domain Preserving Scheme for Euler Equations

Some limitations must be applied to the high-order update, $\mathbf{U}_i^{H,n+1}$, given by (21) and (22) with the viscosity given by (30) to guarantee the preservation of the physical bounds and to be invariant domain preserving.

Given a convex invariant set $B \subset A$, and assuming that $\mathbf{U}_i^n \in B, \forall i \in \mathcal{V}$ and some $n \geqslant 0$, we explain in this section how to push back $\mathbf{U}_i^{H,n+1}$ in the invariant domain using a convex limiting technique. For simplicity, we will give a brief overview of the limiting strategy for Euler equations; the precise description of the method for a general hyperbolic system and the rigorous mathematical framework can be seen in Guermond et al. [9]. We note that when the equation of state (EOS) implies finite compressibility of the fluid (like with the co-volume EOS), we can automatically enforce that with our method and this seems to be difficult with other formulations.

We start by estimating the difference $\mathbf{U}_i^{H,n} - \mathbf{U}_i^{L,n}$. This is done by subtracting (22), written with the high-order viscosity $d_{ij}^{H,n}$, from (22), written with the low-order viscosity $d_{ij}^{L,n}$. We obtain

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

Defining $\lambda_i := \dfrac{1}{\operatorname{card}(\mathcal{I}(S_i^n)) - 1}$, $j \in \mathcal{I}(S_i^n)\backslash\{i\}$, we can write

$$\mathbf{U}_i^{H,n+1} = \sum_{j\in\mathcal{I}(i)\backslash\{i\}} \lambda_j(\mathbf{U}_i^{L,n+1} + \mathbf{P}_{ij}), \tag{31}$$

where

$$\mathbf{P}_{ij} := \frac{\Delta t}{\mathbf{m}_i^{n+1}}(d_{ij}^{H,n} - d_{ij}^{L,n})(\mathbf{U}_j^n - \mathbf{U}_i^n). \tag{32}$$

The main objective of the limiting method is to find symmetric limiting parameters $l_{ij} \in [0, 1]$ so that the new limited solution

$$\mathbf{U}_i^{n+1} := \sum_{j\in\mathcal{I}(i)\backslash\{i\}} \lambda_j(\mathbf{U}_i^{L,n+1} + l_{ij}\mathbf{P}_{ij}) \tag{33}$$

satisfies the expected bounds.

The following lemma proved in Guermond et al. [8, Lem. 4.4] is the workhorse of the limiting technique that we propose.

Lemma 2 *Let $B \subset \mathbb{R}^m$ and $\Psi \in C^0(B;\mathbb{R})$ be such that $\{v \in B \mid \Psi(v) \geqslant 0\}$ is convex (Ψ quasiconcave functional). Let $i \in \mathcal{I}$ and $j \in \mathcal{I}(i)$. Assume that $\mathbf{U}_i^{L,n+1} \in B$ and $\Psi(\mathbf{U}_i^{L,n+1}) > 0$, then there is a unique $l_j^i \in [0, 1]$ defined by*

$$l_j^i = \begin{cases} 1, & \textit{if } \Psi(\mathbf{U}_i^{L,n+1} + \mathbf{P}_{ij}^n) \geqslant 0, \\ \max\{l \in [0,1] \mid \Psi(\mathbf{U}_i^{L,n+1} + l\mathbf{P}_{ij}^n) \geqslant 0\}, & \textit{otherwise}, \end{cases} \tag{34}$$

such that

$$\Psi(\mathbf{U}_i^{L,n+1} + l\mathbf{P}_{ij}^n) \geqslant 0, \qquad \forall l \in [0, l_j^i]. \tag{35}$$

Now we apply this result to limit the high-order scheme to solve Euler equations with an ideal gas EOS given by $p = (\gamma - 1)(E - \frac{1}{2}\rho\|u\|_{\ell^2}^2)$ where $\gamma > 1$. Euler equations can be written as

$$\partial_t \mathbf{U} + \nabla\cdot\mathbf{f}(\mathbf{U}) = 0 \tag{36}$$

with $\mathbf{U} = (\rho, \mathbf{m}, E)^{\mathrm{T}} \in \mathbb{R}^{d+2}$ and $\mathbf{f}(\mathbf{U}) := (\mathbf{m}, \mathbf{v} \otimes \mathbf{m} + p\mathbb{I}, \mathbf{v}(E + p))^{\mathrm{T}}$. The specific energy is

$$e(\mathbf{U}) = \frac{1}{\rho}E - \frac{\mathbf{m}^2}{2\rho^2} \tag{37}$$

and the specific entropy can be defined as

$$s(\rho, e) = \log(e^{\frac{1}{\gamma-1}}\rho^{-1}). \tag{38}$$

The set $\mathcal{A} := \{(\rho, \mathbf{m}, E) | \rho > 0, e > 0, s \geqslant s_{\min}\}$ is an invariant set for the Euler equations for any value of $s_{\min}$. Locally we want to enforce the following bounds:

$$\rho_i^{n,\min} \leqslant \rho_i^{n+1} \leqslant \rho_i^{n,\max}, E_i^{n,\min} \leqslant E_i^{n+1} \leqslant E_i^{n,\max}, s(\rho_i^{n+1}, e(\mathbf{U}_i^{n+1})) \geqslant s_i^{n,\min}, \tag{39}$$

where

$$u_i^{n,\min} = \min_{j\in\mathcal{I}(S_i^n)}(\overline{u}_{ij}^n),\ u_i^{n,\max} = \max_{j\in\mathcal{I}(S_i^n)}(\overline{u}_{ij}^n) \tag{40}$$

and

$$\overline{\mathbf{U}}_{ij}^n = \Big(\mathbf{f}(\mathbf{U}_i^n) - \mathbf{f}(\mathbf{U}_j^n) - (\mathbf{U}_i^n - \mathbf{U}_j^n) \otimes \mathbf{W}_j^n\Big) \cdot \boldsymbol{n}_{ij}^n \frac{\|\boldsymbol{c}_{ij}^n\|_{l^2}}{2d_{ij}^{L,n}} + \frac{1}{2}(\mathbf{U}_i^n + \mathbf{U}_j^n) \tag{41}$$

are the so-called auxiliary (bar) states. Under the CFL condition, the bar state $\overline{\mathbf{U}}_{ij}^n$ is the average of the exact solutions of a Riemann problem with a left state $\mathbf{U}_i^n$ and a right state $\mathbf{U}_j^n$, see Theorem 1 for details. Therefore, as long as the initial states $\mathbf{U}_i^n$ and $\mathbf{U}_j^n$ are in an invariant set of the Euler system, the average $\overline{\mathbf{U}}_{ij}^n$ will stay in that invariant set (recall that all invariant sets are convex).

Now we define the quasiconcave functionals to apply Lemma 2 to our invariant domains. First, we define two functionals to enforce the bounds on the density, $\Psi_1(\mathbf{U}) := \rho - \rho_i^{n,\min} \geqslant 0$ and $\Psi_2(\mathbf{U}) := -\rho + \rho_i^{n,\max} \geqslant 0$. Then we define $c_i^{\min} := \exp(s_i^{\min}(\gamma - 1))$. The inequality $s(\rho, e(\mathbf{U})) \geqslant s_i^{\min}$ is satisfied if and only if

$$\Psi_3(\mathbf{U}) := \rho e(\mathbf{U}) - c_i^{\min}\rho^\gamma \geqslant 0. \tag{42}$$

Defining $\mathcal{B}_i := \{\mathbf{v} \in \mathcal{A} | \Psi_i(\mathbf{v}) \geqslant 0, i = 1, 2, 3\}$, we proved in Guermond et al. [9] that if $\mathbf{U}_j^n \in \mathcal{B}_i$ for all $j \in \mathcal{I}(i)$ and $2\Delta t d_{ii}^{L,n}/\mathfrak{m}_i^{n+1} \leqslant 1$, then

$$\mathbf{U}_i^{n+1} = \sum_{j\in\mathcal{I}(i)\backslash\{i\}} \lambda_j(\mathbf{U}_i^{L,n+1} + l_{ij}\mathbf{P}_{ij}) \in \mathcal{B}_i \tag{43}$$

for $i \in N, n \geqslant 0$.

That means that, for any smooth ALE velocity, the scheme is not only invariant domain preserving ($\mathbf{U}_i^{n+1} \in \mathcal{A}$), but also preserves the local domain ($\mathbf{U}_i^{n+1} \in \mathcal{B}_i$).

Remark 4 In the numerical tests presented in Sect. 5, the quantity $s_i^{n,\min}$ is relaxed as explained in Guermond et al. [8, Sect. 4.7.2], to maintain the second-order accuracy of the method.

Remark 5 Imposing local lower and upper bounds on the density, with the bounds coming from the bar sates, automatically enforces positivity and in case of the co-volume EOS, it also enforces the maximum compressibility of the method.

5 Numerical Results

In this section, we consider a set of test problems for the Euler equations, which are designed to verify the theoretical conservation and invariant domain properties of our high-order ALE finite element scheme. We first solve problems with known solutions to compare the errors obtained with our low- and high-order methods (with and without limiting). Then, we present a more challenging set of problems, where the mesh motion is more complex and we focus the presentation on the robustness of our ALE method. All simulations use RK3 strong stability preserving time integration derived from the explicit Euler scheme given in (22). The EOS for all test cases is the ideal gas law, $p = (\gamma - 1)\rho e$, where the value of γ is given for each test case.

When the exact solution of a specific test case is known, we compute the following relative error indicator:

$$\delta^1(t) := \frac{\|\rho_h(t) - \rho(t)\|_{L^1(D)}}{\|\rho(t)\|_{L^1(D)}} + \frac{\|\mathbf{m}_h(t) - \mathbf{m}(t)\|_{L^1(D)}}{\|\mathbf{m}(t)\|_{L^1(D)}} + \frac{\|E_h(t) - E(t)\|_{L^1(D)}}{\|E(t)\|_{L^1(D)}}. \tag{44}$$

The above norm is estimated using local Gaussian quadrature rules of order 8 with 16 points.

5.1 Isentropic Vortex

The first test case we consider is the so-called isentropic vortex problem. The exact solution is isentropic and is given by

$$\rho(\mathbf{x}, t) = (T_\infty + \delta T)^{1/(\gamma-1)}, \qquad \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_\infty + \delta\mathbf{u}, \qquad p(\mathbf{x}, t) = \rho^\gamma. \tag{45}$$

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

$$\delta\mathbf{u}(\mathbf{x}, t) = \frac{\beta}{2\pi}\mathrm{e}^{\frac{1-r^2}{2}}(-\overline{x}_2, \overline{x}_1), \qquad \delta T(\mathbf{x}, t) = -\frac{(\gamma-1)\beta^2}{8\gamma\pi^2}\mathrm{e}^{1-r^2}, \tag{46}$$

where $r = \|\mathbf{x} - \overline{\mathbf{x}}_c(t)\|$ is the Euclidean distance from the vortex center $\overline{\mathbf{x}}_c(t) := (x_1^0 + 2t, x_2^0)^{\mathrm{T}}$, $\beta = 5$ 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 into two triangles, then the mesh is refined uniformly five times. We set the density, momentum and total 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 $\eta(u) = p^{\frac{1}{\gamma}}$ to set $d_{ij}^{H,n}$, see Sect. 3.4.

Table 1 displays the errors obtained with our ALE schemes at $t = 2$, where the CFL number is 0.25. We observe the first-order accuracy with the GMS-GV method and second-order accuracy with the GMS-EV method with and without limiting. The solution of the isentropic vortex is smooth so the limiting technique does not offer an additional advantage over the initial GMS-EV method; however, we have verified that it does not significantly increase the computational cost or increase the error of the method.

The CPU time used for the three methods is provided in Table 2. It can be observed that the limiting technique increases the CPU time by about 3% when the GMS-EV is used.

Table 1 Isentropic vortex. Convergence test at $t = 2$. CFL = 0.25

# dof	GMS-GV1		GMS-EV		GMS-EV limited	
	$\delta^1(t)$	Rate	$\delta^1(t)$	Rate	$\delta^1(t)$	Rate
441	9.18E−02	–	1.61E−02	–	1.62E−02	–
1 681	7.23E−02	0.34	4.97E−03	1.69	4.97E−03	1.70
6 561	4.98E−02	0.54	1.33E−03	1.90	1.33E−03	1.90
25 921	3.07E−02	0.70	3.38E−04	1.98	3.38E−04	1.98
103 041	1.74E−02	0.82	8.49E−05	2.00	8.49E−05	2.00

Table 2 Isentropic vortex. Comparison of the computational cost between the ALE methods when #dof = 14 641, CFL = 0.25 and the final time is $t = 2$

Method	CPU time	CPU time/(N#dof)
GMS-GV	2 m 21.27 s	4.00E−05
GMS-EV	2 m 39.38 s	4.02E−05
GMS-EV + limiting	2 m 44.30 s	4.14E−05

Table 3 Isentropic vortex. Convergence test at $t = 2$ obtained with GMS-EV with limiting. ALE vs. Eulerian ($v_{\mathrm{A}} = 0$) approximations. CFL = 0.25

# dof	ALE		Eulerian	
	$\delta^1(t)$	Rate	$\delta^1(t)$	Rate
561	1.34E−02	–	3.40E−02	–
2 145	4.05E−03	1.73	9.62E−03	1.82
8 385	1.08E−03	1.90	2.48E−03	1.95
33 153	2.77E−04	1.97	6.33E−04	1.97

Notice that the time step changes in each simulation, because it depends on the viscosity and the fluid velocity. Therefore, a better indicator of the computational cost of each method would be the ratio CPU time/(N#dof) where N is the number of time steps done. From the values of this parameter displayed in Table 2, we can deduce that the computational cost of the low- and high-order methods is similar.

Finally, to illustrate how the accuracy of the methods increases when the ALE description is used, and Table 3 displayed the error indicator obtained with the GMS-EV scheme with limiting using the Eulerian (setting $v_{\mathrm{A}} = 0$) and ALE approaches. When the Eulerian approach is used, as the mesh is fixed and the vortex is moving to $x > 0$ direction, we have to increase the computational domain to ensure that the vortex remains in the domain. We set $D^0 = (-5, 10) \times (-5, 5)$. As in the previous simulations, we consider meshes formed by squares divided into two triangles, and we use the same meshes for both techniques. It can be observed that the errors obtained with the ALE approach are smaller than those obtained with the Eulerian approach, whereas the rate of convergence is the same.

5.2 Noh Problem

Noh problem is a test problem with a known exact solution; a shock wave propagating radially outwards at a constant speed given by

Table 4 Simulations data for the Noh problem

Domain	$D^0 = (-1,1)^2$
Final time	$t_f = 0.6$
# dof	$\mathcal{N} = 921, 3\,721, 14\,641, 58\,081, 231\,361$
Finite elements	Motion $\mathbb{P}_2$, hyperbolic system $\mathbb{P}_1$
Scheme	GMS-GV, GMS-EV, GMS-EV + convex limiting
CFL	0.4

Table 5 Noh problem, convergence test, L^1-norm relative error at $t = 0.6$

# dof	GMS-GV1		GMS-EV		GMS-EV limited	
	$\delta^1(t)$	Rate	$\delta^1(t)$	Rate	$\delta^1(t)$	Rate
961	8.52E−01	–	2.35E−01	–	2.89E−01	–
3 721	5.48E−01	0.64	1.26E−01	0.89	1.51E−01	0.94
14 641	3.14E−01	0.80	7.19E−02	0.81	8.07E−02	0.90
58 081	1.47E−01	1.09	4.16E−02	0.79	4.18E−02	0.95
231 361	7.72E−02	0.93	2.40E−02	0.79	2.18E−02	0.94

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

See Noh [12] and Caramana et al. [3, Sect. 5], for instance. This problem can test the ability of our algorithm to handle discontinuities and still provide a solution consistent with the exact solution, while the invariant domains are preserved.

The initial computational domain is $D^0 = (-1,1) \times (-1,1)$ and we do the computations up to $t_f = 0.6$, with CFL = 0.4 and $\gamma = \frac{5}{3}$. The calculations were performed in a sequence of uniform meshes generated from $N{\times}N, N \in \{30, 60, 120, 240\}$ squares divided into two triangles. A summary of the data used in our simulations is provided in Table 4.

Convergence rates and L^1 relative errors are presented in Table 5. With both schemes, GMS-GV and GMS-EV limited, the convergence rates are close to first order which is consistent with the literature on this problem. As expected, the errors obtained using the high-order scheme are smaller. However, when using the GMS-EV high-order scheme without applying the limiting technique, the convergence rates are worse. This is consistent with the results shown in Fig. 3. On the left panel of Fig. 3, a scatter plot of the density field is plotted for three ALE methods: GMS-GV, GMS-EV and GMS-EV with convex limiting. To obtain these plots, we compute the distance to the center of each mesh vertex and plot it versus the value of the density at this point. We compare our results with the exact solution. If we look at the results obtained with the GMS-EV with and without limiting, plotted with green and red solid lines, respectively, we observe that the non-limited solution is more precise, but it exhibits undershoots and overshoots, as expected.

To examine the behavior of our ALE technique, Fig. 4 compares the solution obtained with our best scheme (GMS-EV + limiting) with the same method using an Eulerian description of the motion. It can be concluded that the solution with the Eulerian description is much more diffusive.

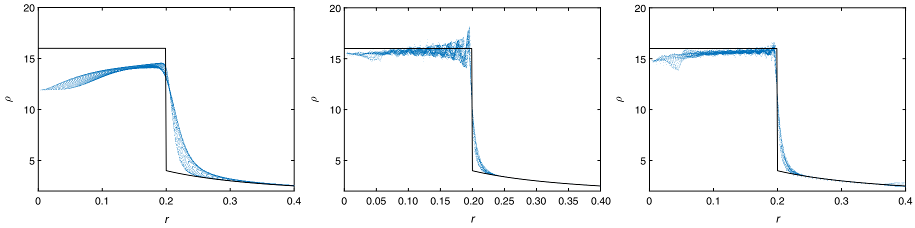

Fig. 3 Noh problem. Scatter plot of the density at the vertices of the mesh as a function of the radial distance, computed on the $120 \times 120 \times 2$ element mesh. From left to right: ALE GMS-GV, GMS-EV and GMS-EV with limiting schemes

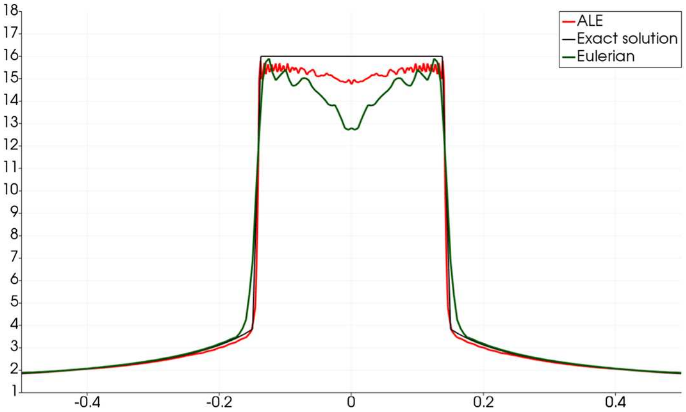

Fig. 4 Noh problem. Density along the diagonal line from $(-0.5, 0.5)$ to $(0.5,\ 0.5)$, computed on the $240 \times 240 \times 2$ element mesh. ALE vs. Eulerian GMS-EV algorithm with limiting

To illustrate the robustness of our ALE method with respect to the mesh regularity, we solve the problem on a non-uniform mesh. The initial mesh is divided into four quadrants: the bottom left quadrant is composed of 32×32 squares; the top left is composed of 32×64 squares; the top right is composed of 64×64; and the bottom right is composed of 64×32 squares. Each square cell is divided into two triangles. Figure 5 indicates that our ALE high-order method preserves the radial symmetry of the solution and does not develop any hour-glass-like instability with a non-uniform mesh.

5.3 Radial Sod Problem

We run the well-known Sod shock tube problem using a radial configuration, so the initial condition is

$$u(x, 0) = \begin{cases} (1, 0, 0, 0.25), & \text{if } \sqrt{x_1^2 + x_2^2} \leqslant 0.5, \\ (0.125, 0, 0, 2.5), & \text{if } \sqrt{x_1^2 + x_2^2} > 0.5, \end{cases} \tag{47}$$

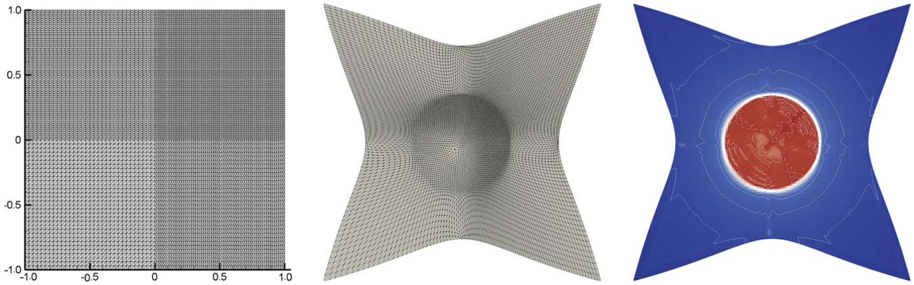

Fig. 5 Noh problem. Solution obtained with the GMS-EV method and the convex limiting technique using the non-uniform mesh. From left to right: initial mesh, mesh and density contours at $t_f = 0.6$

and we consider the domain $D = (0, 1) \times (0, 1)$ and impose symmetry conditions at $x = 0$ and $y = 0$. Here, we take $\gamma = 1.4$. We perform the calculations on a uniform mesh formed by 100×100 squares divided into two triangles each. The final time in our simulations is $t_f = 0.225$ and the CFL number is 0.25.

For this problem, we use the GMS-EV method with an entropy commutator computed with the physical entropy $\eta(u) = s$ to set $d_{ij}^{H,n}$. This high-order viscosity, together with the limiting technique, allows us to use a purely Lagrangian motion for the mesh without smoothing the velocity. Thus, for this test problem, in the definition of the ALE velocity, we only do the reconstruction stage using $\mathbb{P}_2$ finite elements.

The results for the ALE (Lagrangian in this case) and Eulerian GMS-EV schemes are presented in Fig. 6. The density fields for both methods are plotted together with the final mesh obtained by the ALE method. The Eulerian versions of the scheme suffer from a strong numerical diffusion, whereas with the ALE method, the mesh moves to better capture the shock wave and the contact discontinuity. The same results can be observed from the surface plots of the density in Fig. 7.

5.4 Double Mach Reflection Problem

The double Mach reflection problem involves a strong shock (Mach 10) moving in air ($\gamma = 1.4$) that impinges a wall with 60 degree angle. When the shock runs up the wall, a

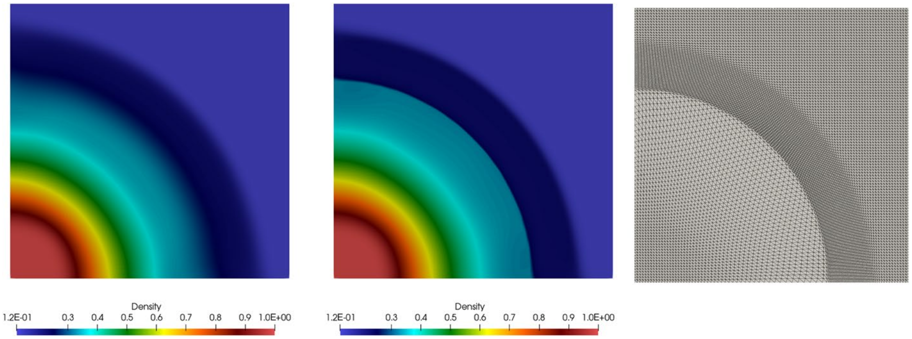

Fig. 6 Radial Sod problem. Density field at $t_f = 0.225$ computed by the limited GMS-EV scheme with Eulerian (left) and ALE (center) descriptions. On the left, the final mesh for the ALE scheme is plotted

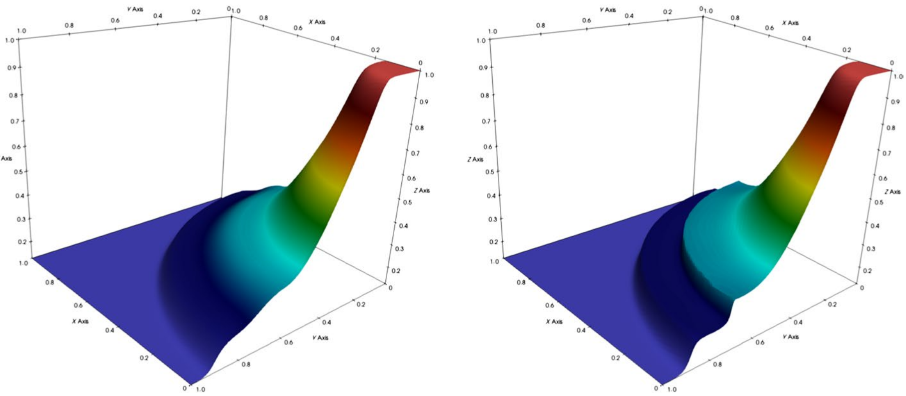

Fig. 7 Radial Sod problem. Surface plot of density at $t = 0.225$ obtained with the GMS-EV method combined with the convex limiting technique. From left to right: Eulerian vs. ALE description

self-similar structure (with two triple points) emerges. This is a standard benchmark test popularized by Woodward and Colella [14] and then by many others for evaluating the resolution of Euler codes. Different setups are proposed in the literature for this problem to prevent the formation of undesirable numerical artifacts (see Vevek et al. [13]). The setup in the case of an ALE code is even more problematic, we use a setting similar to that of Zhao et al. [15] but allow the non-wall boundaries to move with the flow.

We solve this problem in the initial computational domain $D^0 = (-1.5, 4) \times (0, 1)$, and with the shock initially positioned at $x_1 = 1/6$ where the horizontal wall begins. A free-slip boundary condition is enforced on $x_2 = 0, x_1 \geqslant 1/6$, at the top boundary, the flow values are set to describe the exact motion of the Mach 10 shock, and inflow and outflow boundary conditions are enforced on the left and right boundaries, respectively. The initial condition is

$$u(x, 0) = \begin{cases} (8, 57.151\,97, -33.001\,2, 563.544), & \text{if } x_2 \geqslant g(x_1, 0), \\ (1.4, 0, 0, 2.5), & \text{if } x_2 < g(x_1, 0), \end{cases} \tag{48}$$

where $g(x_1, t) = \sqrt{3}(x_1 - \frac{1}{6}) - 20t$ is the position of the shock at instant t. The flow is computed at time $t = 0.2$ with $\mathrm{CFL} = 0.5$. The initial mesh is a uniform mesh formed by 65 142 triangles and 33 072 $\mathbb{P}_1$ nodes. Regarding the mesh motion, we allow the inflow and the top boundary nodes to move right with the flow. Because part of the bottom boundary is a wall, the second component of the velocity in the wall nodes is 0. Therefore, in order to maintain the computational domain as a square, we set on all boundaries $(\mathbf{v}_\mathrm{A})_{x_2} = 0$.

Figure 8 presents the density field obtained with the GMS-EV scheme using ALE and Eulerian approaches. In the works of Zhao et al. [15] and Boscheri and Balsara [1], the double Mach reflection problem is also solved using ALE techniques. The mesh regularization stage in our ALE movement is much simpler than that used in these two references where a much more sophisticated mesh motion is defined. However, we believe that our results are in good agreement with the results of Boscheri and Balsara [15], Zhao et al. [1] as the final results for the density are similar. The result from the GMS-EV scheme using ALE mesh motion appears to be superior to the Eulerian one. The triple points are resolved better and the jet is well defined (Fig. 9).

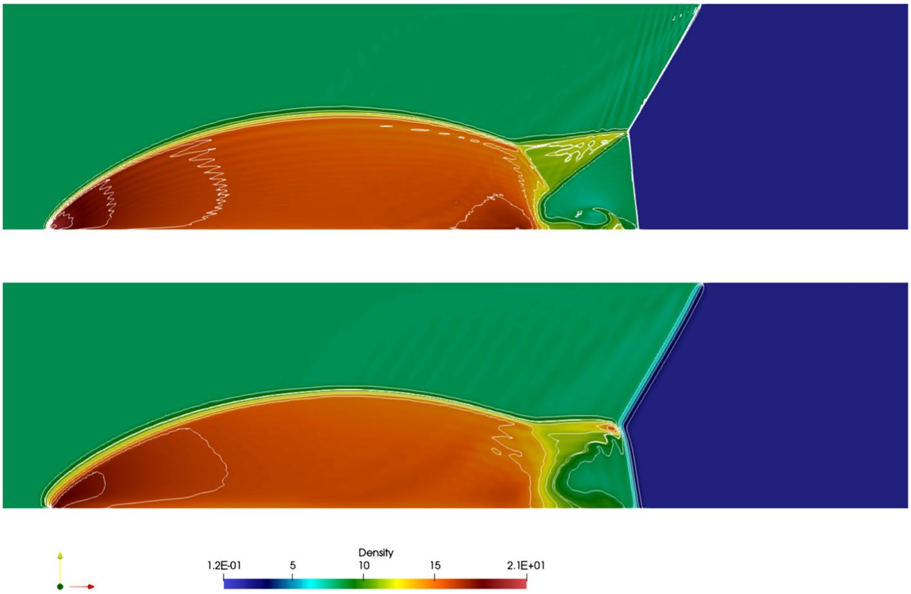

Fig. 8 Double Mach reflection problem. Density field and contours of the density at $t_f = 0.2$ obtained with the GMS-EV scheme with limiting using ALE (top) and Eulerian (bottom) approaches

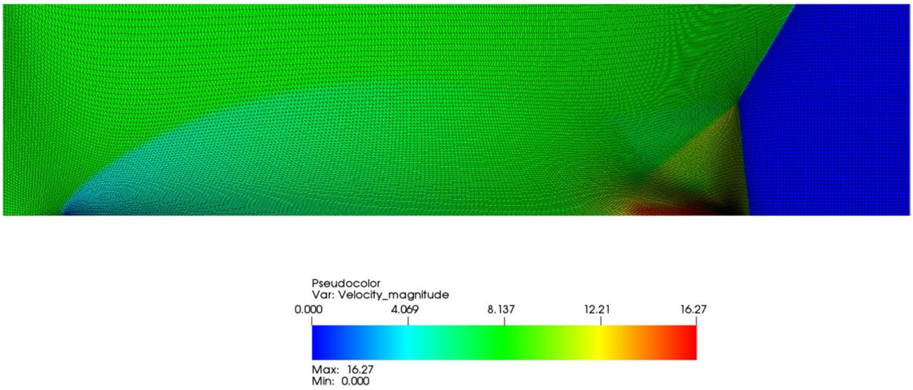

Fig. 9 Double Mach reflection problem. Velocity magnitude field and mesh obtained at $t_f = 0.2$ with the limited ALE GMS-EV scheme

5.5 Rayleigh-Taylor Instability

This instability occurs in the interface between two fluids of different densities when the more dense fluid is on top. Under the force of gravity, the initial equilibrium is unstable to any perturbation of the interface and the heavier fluid goes into the lighter one developing the so-called Rayleigh-Taylor instability.

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 $\mathbf{g} = (0, -0.1x_2)^{\mathrm{T}}$. The

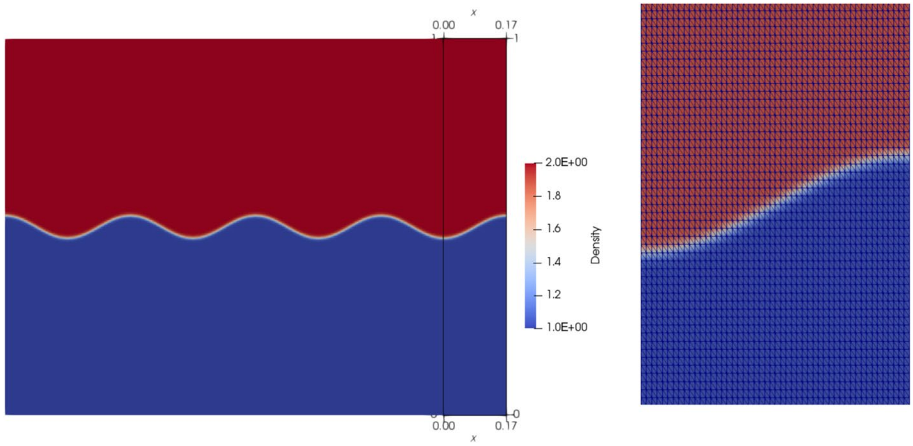

Fig. 10 RT problem. Initial configuration for our simulations (left) and initial mesh of the domain close to the interface (right)

Table 6 Data for the simulations of the Rayleigh-Taylor instability

Domain	$D = (0, 1/6) \times (0, 1)$
Final time	$t_f = 7.5$
Mesh	50×200 squares divided into two triangles
Finite elements	Motion $\mathbb{P}_4$, hyperbolic system $\mathbb{P}_1$
Scheme	GMS-EV + convex limiting, CFL = 0.6

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 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

$$\rho_0(x) := 2 + \tanh\left(\frac{y - \eta(x)}{0.01d}\right). \tag{49}$$

The initial pressure is hydrostatic. The slip boundary condition is enforced on the four walls of D^0. In Fig. 10, we can see the computational domain and the initial conditions, as well as the details of the initial mesh around the interface. 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 $= 0.6$. A summary of the data used in the simulations is given in Table 6.

The mesh undergoes very large deformations in the time interval $[0, t]$ and is smoothed using $s = 3$ to define the blending parameter given by (15). Figure 11 displays the density field obtained with the GMS-EV scheme with limiting using ALE and Eulerian approaches. Finally, to illustrate that the method is robust when using fourth-order polynomials to describe the mesh motion, that is $P^{\text{geo}} = \mathbb{P}_{4,2}$, Fig. 12 displays the meshes and the density field at six time: $t = 2, 3, 4, 5, 6, 7.5$. These pictures show that the mesh undergoes very large deformations in this test.

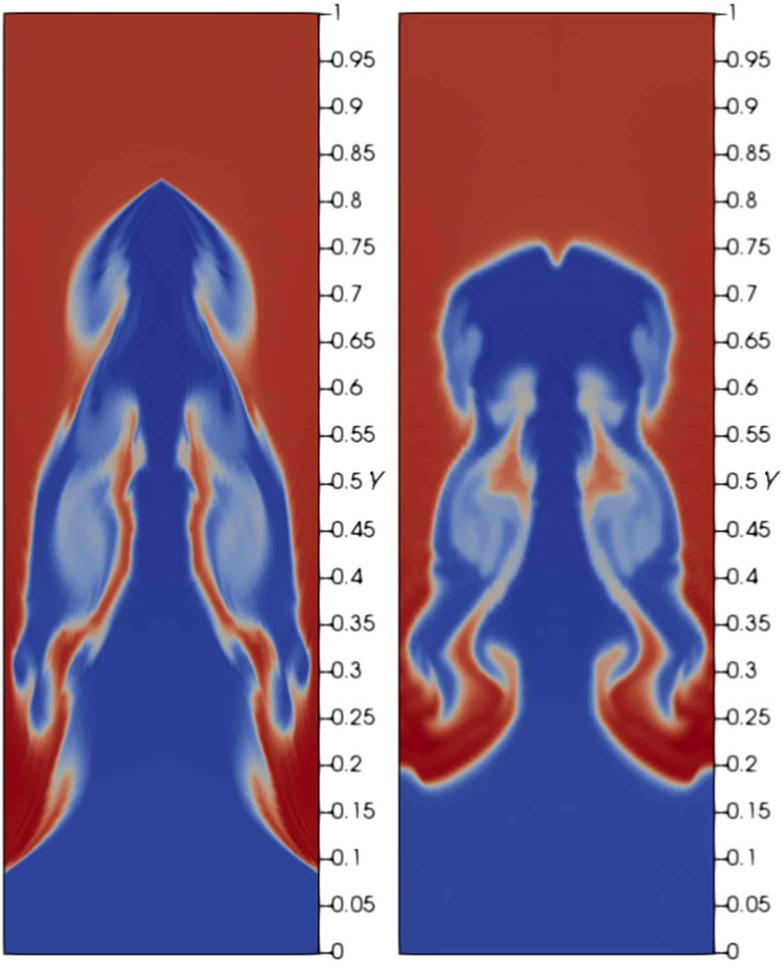

Fig. 11 Rayleigh-Taylor instability. Density field at $t = 7.5$ s obtained with GMS-EV method with limiting. ALE (left panel) vs. Eulerian (right panel) approach

We do not compare here the solutions obtained by the different schemes proposed, as it was done in the previous sections. This is because when we try to solve this problem with the low-order method, with the mesh described above, the interface between the two fluids is completely diffused. On the other hand, when we use the GMS-EV scheme (not limited), the solution is not between the physical bounds. As a consequence the mesh collapses at a certain time (before the final time of $t = 7.5$ s), even when doing the smoothing, because the ALE velocity is not smooth enough to keep the mesh untangled.

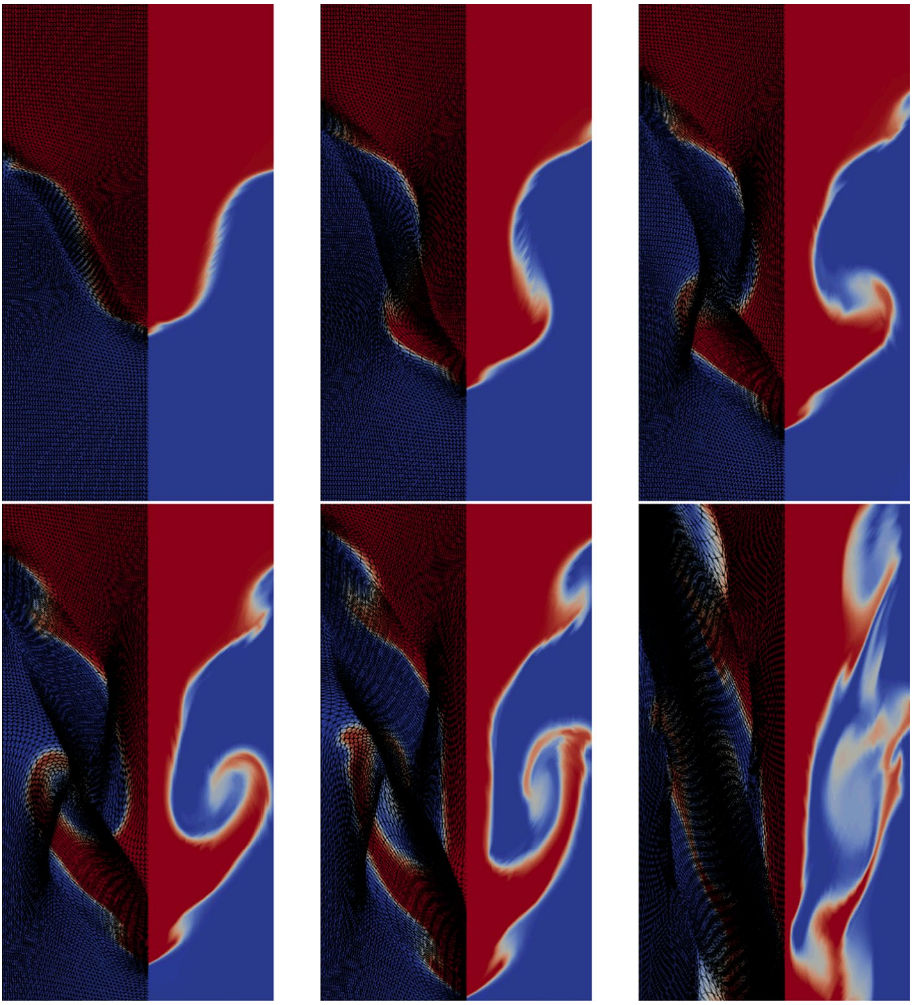

Fig. 12 Rayleigh-Taylor instability. Density field and mesh at time $t = 2, 3, 4, 5, 6, 7.5$ (from left to right and from top to bottom). We use ALE GMS-EV with convex limiting; CFL $= 0.6$

6 Conclusion

We have extended in this paper our work from Guermond et al. [7, 9]. The main novelty is that we give complete details on the implementation of the method for the Euler equations and present new numerical tests. In particular, we compare the performance of the ALE and Eulerian formulations of our high-order scheme for the radial Sod and the double Mach reflection problems. The results clearly show the advantage of using the ALE method and confirm the robustness of our methodology. More sophisticated mesh motions and handling of multiple materials will be addressed in future works.

Acknowledgements This material is based upon work supported in part by a “Computational R&D in Support of Stockpile Stewardship” Grant from Lawrence Livermore National Laboratory, the National

Science Foundation Grants DMS-1619892, by the Air Force Office of Scientific Research, USAF, under Grant/contract number FA99550-12-0358, and by the Army Research Office under Grant/contract number W911NF-15-1-0517 and by the Spanish MCINN under Project PGC2018-097565-B-I00.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature.

Compliance with Ethical Standards

Conflict of Interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Boscheri, W., Balsara, D.S.: High order direct Arbitrary-Lagrangian-Eulerian (ALE) PNPM schemes with WENO adaptive-order reconstruction on unstructured meshes. J. Comput. Phys. **398**, 108899 (2019)
2. Boscheri, W., Balsara, D.S., Dumbser, M.: Lagrangian ADER-WENO finite volume schemes on unstructured triangular meshes based on genuinely multidimensional HLL Riemann solvers. J. Comput. Phys. **267**, 112–138 (2014)
3. Caramana, E., Shashkov, M., Whalen, P.: Formulations of artificial viscosity for multi-dimensional shock wave computations. J. Comput. Phys. **144**(1), 70–97 (1998)
4. Dobrev, V., Knupp, P., Kolev, T., Mittal, K., Rieben, R., Tomov, V.: Simulation-driven optimization of high-order meshes in ALE hydrodynamics. Comput. Fluids **208**, 104602 (2020)
5. Dyn, N., Levine, D., Gregory, J.A.: A butterfly subdivision scheme for surface interpolation with tension control. ACM Trans. Graph. **9**(2), 160–169 (1990)
6. Guermond, J.-L., Popov, B.: Invariant domains and first-order continuous finite element approximation for hyperbolic systems. SIAM J. Numer. Anal. **54**(4), 2466–2489 (2016)
7. Guermond, J.-L., Popov, B., Saavedra, L., Yang, Y.: Invariant domains preserving arbitrary Lagrangian Eulerian approximation of hyperbolic systems with continuous finite elements. SIAM J. Sci. Comput. **39**(2), A385–A414 (2017)
8. Guermond, J.-L., Nazarov, M., Popov, B., Tomas, I.: Second-order invariant domain preserving approximation of the Euler equations using convex limiting. SIAM J. Sci. Comput. **40**(5), A3211–A3239 (2018)
9. Guermond, J.-L., Popov, B., Saavedra, L.: Second-order invariant domain preserving ALE approximation of hyperbolic systems. J. Comput. Phys. **401**, 108927 (2020)
10. Lai, M.-J., Schumaker, L.L.: Spline Functions on Triangulations. Encyclopedia of Mathematics and Its Applications, vol. 110. Cambridge University Press, Cambridge (2007)
11. Loubère, R., Maire, P.-H., Shashkov, M., Breil, J., Galera, S.: ReALE: A reconnection-based arbitrary-Lagrangian-Eulerian method. J. Comput. Phys. **229**(12), 4724–4761 (2010)
12. Noh, W.: Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux. J. Comput. Phys. **72**(1), 78–120 (1987)
13. Vevek, U., Zang, B., New, T.: On alternative setups of the double Mach reflection problem. J. Comput. Phys. **78**(1), 1291–1303 (2019)
14. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**(1), 115–173 (1984)
15. Zhao, X., Yu, X., Duan, M., Qing, F., Zou, S.: An arbitrary Lagrangian-Eulerian RKDG method for compressible Euler equations on unstructured meshes: single-material flow. J. Comput. Phys. **396**, 451–469 (2019)