ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/322261767

A second-order cell-centered Lagrangian ADER-MOOD finite volume scheme on multidimensional unstructured meshes for hydrodynamics

Article *in* Journal of Computational Physics · April 2018

DOI: 10.1016/j.jcp.2017.12.040

CITATIONS
9

READS
434

4 authors, including:

Walter Boscheri
University of Ferrara
45 PUBLICATIONS **732** CITATIONS
SEE PROFILE

Raphaël Loubère
French National Centre for Scientific Research
96 PUBLICATIONS **2,158** CITATIONS
SEE PROFILE

Pierre-Henri Maire
Atomic Energy and Alternative Energies Commission
93 PUBLICATIONS **2,759** CITATIONS
SEE PROFILE

Some of the authors of this publication are also working on these related projects:

very high order finite volume for linear PDE View project

Multimaterial Arbitrary Lagrangian-Eulerian Methods for High-speed compressible flows View project

All content following this page was uploaded by Raphaël Loubère on 10 January 2018.

The user has requested enhancement of the downloaded file.

A second-order cell-centered Lagrangian ADER-MOOD finite volume scheme on multidimensional unstructured meshes for hydrodynamics

Walter Boscheri [a] , Michael Dumbser [b] , Raphaël Loubère [c] , Pierre-Henri Maire [d] .

[a] *Faculty of Science and Technology*
Free University of Bozen, Piazza Universitá 1, I-39100 Bolzano, Italy

[b] *Laboratory of Applied Mathematics*
Department of Civil, Environmental and Mechanical Engineering
University of Trento, Via Mesiano 77, I-38123 Trento, Italy

[c] *CNRS and Institut de Mathématiques de Bordeaux (IMB), France*

[d] *CEA-CESTA, Le Barp, France*

Abstract

In this paper we develop a conservative cell-centered Lagrangian finite volume scheme for the solution of the hydrodynamics equations on unstructured multidimensional grids. The method is derived from the Eucclhyd scheme discussed in [47,43,45]. It is second-order accurate in space and is combined with the *a posteriori* Multidimensional Optimal Order Detection (MOOD) limiting strategy to ensure robustness and stability at shock waves. Second-order of accuracy in time is achieved via the ADER (Arbitrary high order schemes using DERivatives) approach. A large set of numerical test cases is proposed to assess the ability of the method to achieve effective second order of accuracy on smooth flows, maintaining an essentially non-oscillatory behavior on discontinuous profiles, general robustness ensuring physical admissibility of the numerical solution, and precision where appropriate.

Key words: Cell-centered Lagrangian finite volume schemes, Eucclhyd, moving unstructured meshes, a posteriori MOOD limiting, ADER approach

Email addresses: walter.boscheri@unibz.it (Walter Boscheri), michael.dumbser@unitn.it (Michael Dumbser), raphael.loubere@u-bordeaux.fr (Raphaël Loubère), pierre-henri.maire@cea.fr (Pierre-Henri Maire).

1 Introduction

This paper considers the hydrodynamics system of conservation laws written in Lagrangian formulation, which dates back to 1940. Lagrangian numerical methods are a moving mesh discretization wherein the computational grid moves with the flow velocity. It thus requires the computation of a nodal velocity. The most natural way to obtain it relies on a staggered discretization for which the kinematic variables are located at nodes whereas the thermodynamic ones are placed at cell centers. The seminal work of von Neumann and Richtmyer [51] has designed the one-dimensional framework of staggered discretization which has been later extended to two-dimensional geometry by Wilkins [61]. The compatibility of staggered discretization with the second law of thermodynamics was ensured by adding an artificial viscosity to the internal energy equation allowing the dissipation of kinetic energy into internal energy through shock waves. It is worth mentioning that this space and time staggered discretization cannot conserve total energy. Burton and his co-authors [12] have cured this flaw by constructing a compatible hydrodynamics method based on the one hand on a predictor-corrector time discretization which conserves total energy and, on the other hand, on the adjointness property of the discrete gradient and divergence operators. The interested reader might refer to Després book page 241 [11] or to the review papers [1] for a modern presentation of this topic. In spite of the long history of staggered Lagrangian methods based on artificial viscosity, an increasing interest in cell-centered Lagrangian schemes dedicated to solve the hydrodynamics equations has been observed since 2005 with the seminal works of Després et al.[21] and Maire et al.[47]. Their Lagrangian cell-centered formulations have been originally developed for the 2D hydrodynamics equations on quadrilateral meshes, later extended to 3D hexahedral grids and further to general unstructured meshes [14,43]. As a purely Lagrangian formulation, no mass flux is allowed across cell interfaces, and the cells move and deform according to the action of the neighborhood. The main novelty in their approach is the definition of a multidimensional nodal solver taking into account all surrounding cells to uniquely determine the node velocity. As such they ensure the compatibility between the motion of the cell (and its associated volume as a by-product), and the cell deformation computed through PDEs involving the discretization of the velocity divergence. This so-called Geometrical Conservation Law (GCL) allows to derive numerical methods on a more firm base, and re-use many of the advances from the well-established Godunov-type finite volume schemes on fixed grids.
Indeed, these cell-centered Lagrangian algorithms are Godunov-type schemes on moving meshes, accurate up to second-order in time through Runge-Kutta time stepping, Predictor-Corrector strategy or Generalized Riemann problem methodology and second-order in space via limited piecewise-linear reconstruction of velocity and pressure fields. They are solving the conservation laws for mass, momentum and total energy. Contrarily, classical staggered Lagrangian schemes solve the specific internal energy equation, retrieving the total energy conservation by a compatible derivation of the div-grad operator [12]. The differences between nowadays cell-centered Lagrangian schemes reside in the way how the nodal solver and the limiter are designed and derived. The limiters employed to ensure robustness and preserve flow symmetries are based on classical slope limiters [54] for scalar fields and frame-invariant limiters for vector fields [40,41]. These schemes are in fact used as the engine of indirect Arbitrary-Lagrangian-Eulerian (ALE) simulation codes, see [31] for instance. High order of accuracy in space has been first introduced in [15,38,17,18], where a third order ENO reconstruction operator is employed in the context

of cell-centered Lagrangian-type finite volume schemes. High order of accuracy in time was guaranteed either by the use of a Runge-Kutta or by a LaxWendroff-type time stepping. Parallel to these developments, a new generation of multi-dimensional cell-centered direct ALE schemes based on the Eulerian formulation of the governing equations has emerged [6,10]. Following the concepts of (Arbitrary high order schemes using DERivatives) ADER discretization, approximate Riemann solvers, high order accurate polynomial reconstructions, WENO or *a posteriori* limiting, these numerical methods have proven to be both accurate and robust for classical Lagrangian-like test cases. However the way these methods are derived prevents to determine an associated strictly Lagrangian scheme, because mass flux may exist among computational cells. Nonetheless, by re-using the Lagrangian nodal solvers [47] within those direct ALE schemes, some authors have shown [10,5] that this results in an effectively quasi-Lagrangian scheme which presents good properties: (very) high numerical accuracy on smooth flows, sharp contact discontinuities and interfaces, shock waves spread over one or two cells.
Consequently, the question arises whether it is possible to derive also a *truly* Lagrangian scheme without intercell mass flux based on these concepts developed in the Eulerian and ALE framework. This is the purpose of this work, which makes use of the Eucclhyd scheme [47,43,45] as a solid starting point for dealing with Lagrangian hydrodynamics. Therefore we will focus on deriving the Lagrangian version of the hydrodynamics equations, adapting the direct ALE numerical method to the situation of zero mass flux, leading *de facto* to a genuinely Lagrangian scheme. Unstructured 2D and 3D meshes made of simplex elements are only considered in this work even if quadrangles/hexahedra could be employed. In this work we wish to maintain simple cell shapes with straight edges, and, as such, we assume that the velocity is at most linear. Consequently the proposed scheme can not exceed a second-order of accuracy. If we want to go beyond second-order of accuracy, one has to cope with curvilinear geometry, see for instance [29,24,23]. Extension to curved meshes is beyond the scope if this work, although it has already been proposed in the context of direct ALE algorithms [7,8] and Lagrangian-type schemes [16]. To give a brief overview, the numerical method presented in this paper is built relying on the vertex-based Eucclhyd finite volume scheme, the ADER strategy for obtaining a second-order space-time discretization and the *a posteriori* Multidimensional Optimal Order Detection (MOOD) limiting to switch from pure second-order to first-order of accuracy without needing to employ any classical *a priori* slope limiter.

To the best knowledge of the authors, this is the first time that the *a posteriori* MOOD limiting strategy is applied in the context of pure Lagrangian cell-centered Godunov-type finite volume schemes on unstructured meshes in two and three space dimensions. The *a posteriori* limiting approach is substantially different from existing *a priori* limiters, which are typically used in pure Lagrangian schemes. The new method proposed here is by construction positivity preserving, since it relies on a provably positivity preserving first order fallback scheme, and is nominally second order accurate in space and time.

This paper first presents the context and the governing equations. In Section 2, the numerical scheme is introduced with emphasis on the ADER approach, the nodal solver and the *a posteriori* limiting strategy. Also the main properties of the new method are pointed out. All numerical tests are gathered in Section 3. We simulate the following test problems: Kidder, Sod, Saltzmann, Sedov, and the linear phase of a Richtmyer-Meshkov instability. All tests are run in 2D and 3D. Conclusions and perspectives are drawn in the last section.

2 Numerical method

In this paper we consider the equations of Lagrangian hydrodynamics which can be written in the control volume formulation as

$$\frac{d}{dt}\int\limits_{\Omega(t)} \rho\, dV = 0, \tag{1a}$$

$$\frac{d}{dt}\int\limits_{\Omega(t)} dV - \int\limits_{\partial\Omega(t)} \mathbf{u}\cdot\mathbf{n}\, dS = 0, \tag{1b}$$

$$\frac{d}{dt}\int\limits_{\Omega(t)} \rho\mathbf{u}\, dV + \int\limits_{\partial\Omega(t)} p\mathbf{n}\, dS = 0, \tag{1c}$$

$$\frac{d}{dt}\int\limits_{\Omega(t)} \rho E\, dV + \int\limits_{\partial\Omega(t)} p\mathbf{u}\cdot\mathbf{n}\, dS = 0. \tag{1d}$$

Here, $\Omega(t) \subset \mathbb{R}^d$ denotes the time-dependent control volume in $d \in \{2, 3\}$ space dimensions and $\partial\Omega(t)$ its surface defined by the outward pointing unit normal vector $\mathbf{n}$. Furthermore, ρ is the fluid density, $\mathbf{u} = (u, v, w)$ represents the velocity vector in the physical space, $\mathbf{x} = (x, y, z)$ is the coordinate vector, p is the fluid pressure and E the specific total energy. Equations (1a)-(1d) express the conservation of mass, volume, momentum and total energy, respectively. The balance law (1b) is typically referred to as Geometric Conservation Law (GCL) and the velocity of a Lagrangian particle is governed by the local kinematic equation (trajectory equation)

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}, \qquad \mathbf{x}(t=0) = \mathbf{x}_0, \tag{2}$$

where $\mathbf{x}_0$ stands for the coordinates for any point in the initial control volume $\Omega(t=0)$, while $\mathbf{x}$ describes its evolution for $t > 0$. This equivalence often leads to ignoring (1b) and only solving the trajectory equation(2) [1]. System (1) is closed by an equation of state which, for an ideal gas, reads

$$p = (\gamma - 1)\rho\varepsilon, \qquad \varepsilon := \left(E - \frac{1}{2}\mathbf{u}^2\right), \tag{3}$$

with γ representing the ratio of specific heats and ε the specific internal energy.
We denote the vector of variables $\mathbf{Q} = (1, \frac{1}{\rho}, \mathbf{u}, E)$ and the admissible states are defined by the set

$$\mathcal{A} = \left\{ \mathbf{Q} = (1, \frac{1}{\rho}, \mathbf{u}, E),\ \text{s.t. } \rho > 0 \text{ and } \varepsilon > 0 \right\}, \tag{4}$$

which determines the basic property for a physical state. Such condition is in principle necessary but not sufficient, because one should also consider other constraints like entropy inequality or other requirements demanded by the governing equations.
System (1) can be written more compactly at the aid of vector $\mathbf{Q}$ and its associated fluxes

[1] However, we must ensure that this equivalence is maintained by the numerical scheme, which may not always be the case.

$\mathbf{F}(\mathbf{Q}) = (0, -\mathbf{u}, p, p\mathbf{u})$, thus

$$\frac{d}{dt} \int\limits_{\Omega(t)} \rho \mathbf{Q} \, dV + \int\limits_{\partial\Omega(t)} \mathbf{F}(\mathbf{Q}) \cdot \mathbf{n} \, dS = 0. \tag{5}$$

2.1 Discretization of the physical conservation laws

2.1.1 Mesh and coordinates

The computational domain $\Omega(t)$ is discretized at time t by a set of non-overlapping simplicial control volumes $T_i(t)$, that can be either triangles for $d = 2$ or tetrahedra for $d = 3$. N_E denotes the total number of elements contained in the domain. For any current discrete time t^n the union of all elements $T_i^n := T_i(t^n)$ is called the *current mesh configuration* $\mathcal{T}_\Omega^n$ of the domain

$$\mathcal{T}_\Omega^n = \bigcup_{i=1}^{N_E} T_i^n. \tag{6}$$

Each control volume is defined in the *physical* space $\mathbf{x} = (x, y, z)$ and it can be mapped onto a reference element T_e in the reference coordinate system $\boldsymbol{\xi} = (\xi, \eta, \zeta)$, see Figure 1. The reference element is the unit triangle in 2D with vertices $\boldsymbol{\xi}_{e,1} = (\xi_{e,1}, \eta_{e,1}) = (0, 0)$, $\boldsymbol{\xi}_{e,2} = (\xi_{e,2}, \eta_{e,2}) = (1, 0)$ and $\boldsymbol{\xi}_{e,3} = (\xi_{e,3}, \eta_{e,3}) = (0, 1)$, or the unit tetrahedron in 3D with nodes $\boldsymbol{\xi}_{e,1} = (\xi_{e,1}, \eta_{e,1}, \zeta_{e,1}) = (0, 0, 0)$, $\boldsymbol{\xi}_{e,2} = (\xi_{e,2}, \eta_{e,2}, \zeta_{e,2}) = (1, 0, 0)$, $\boldsymbol{\xi}_{e,3} = (\xi_{e,3}, \eta_{e,3}, \zeta_{e,3}) = (0, 1, 0)$ and $\boldsymbol{\xi}_{e,4} = (\xi_{e,4}, \eta_{e,4}, \zeta_{e,4}) = (0, 0, 1)$. The spatial mapping in 3D reads

$$\mathbf{x} = \mathbf{X}_{1,i}^n + \left(\mathbf{X}_{2,i}^n - \mathbf{X}_{1,i}^n\right) \xi + \left(\mathbf{X}_{3,i}^n - \mathbf{X}_{1,i}^n\right) \eta + \left(\mathbf{X}_{4,i}^n - \mathbf{X}_{1,i}^n\right) \zeta, \tag{7}$$

where $\mathbf{X}_{k,i}^n = (X_{k,i}^n, Y_{k,i}^n, Z_{k,i}^n)$ represents the vector of physical spatial coordinates of the k-th vertex of element T_i^n for $k = 1, 2, 3, 4$.

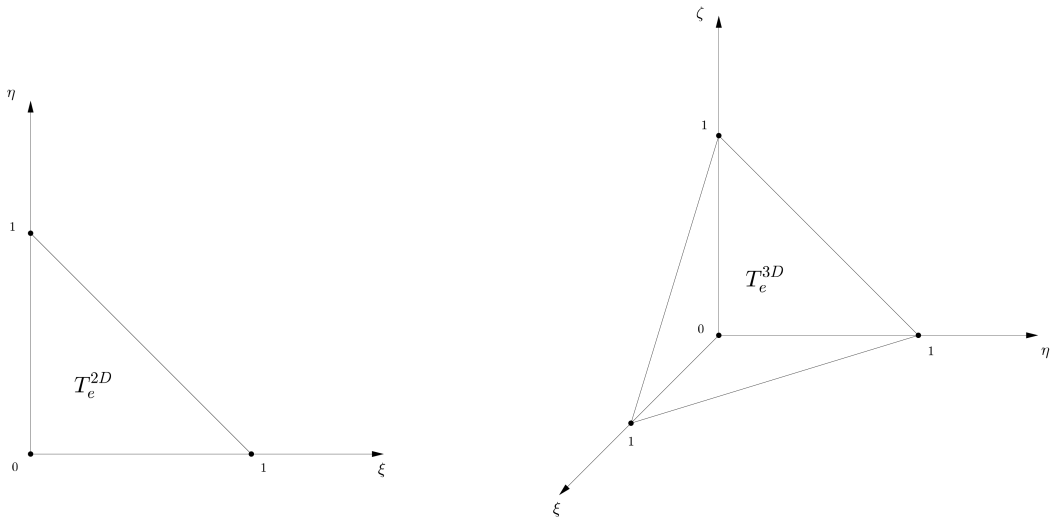

Fig. 1. Reference element T_e in $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ for $d = 2$ (left) and $d = 3$ (right).

2.2 Local form of the governing equations

System (5) is considered for each element $T_i(t)$, hence

$$\frac{d}{dt}\int\limits_{T_i(t)} \rho\mathbf{Q}\,dV + \int\limits_{\partial T_i(t)} \mathbf{F}(\mathbf{Q})\cdot\mathbf{n}\,dS = 0. \tag{8}$$

Recall that the first equation of (8) implies that the cell mass m_i remains constant in time, which is defined as

$$m_i := \int\limits_{T_i(t)} \rho\,dV. \tag{9}$$

Following the approach detailed in [42,49], let us define the mass averaged value of $\mathbf{Q}$ over the element $T_i(t)$ as

$$\mathbf{Q}_i \equiv \mathbf{Q}_i(t) = \frac{1}{m_i}\int\limits_{T_i(t)} \rho\mathbf{Q}\,dV, \tag{10}$$

from which a reformulation of system (8), after the use of the Reynolds transport theorem (see [44] for an exhaustive derivation and justification), reads

$$m_i\frac{d}{dt}\mathbf{Q}_i + \int\limits_{\partial T_i(t)} \mathbf{F}(\mathbf{Q})\cdot\mathbf{n}\,dS = 0, \tag{11}$$

with $\mathbf{Q}_i = \left(\frac{1}{\rho_i}, \mathbf{u}_i, E_i\right)$. For the sake of completeness let us define the cell volume as

$$|T_i| \equiv |T_i(t)| = \int\limits_{T_i(t)} dV, \tag{12}$$

which allows the cell density to be determined as $\rho_i(t) = m_i/|T_i(t)|$ for any time t. Note that the mass conservation (1a) is implicitly taken into account in system (11), while the specific volume $\frac{1}{\rho_i}$ is evolved in the geometric conservation law (1b). In the next section we derive our Lagrangian scheme by employing the ADER technology (Arbitrary high order schemes using DERivatives).

2.3 ADER methodology for Lagrangian hydrodynamics

The ADER ("Arbitrary high order schemes using DERivatives") approach originates from the work of Toro and collaborators [50,56,27]. The first ADER algorithms [50,55,36,56] followed the concept of Ben-Artzi and Falcovitz [3] based on an approximate solution of the generalized Riemann problem (GRP) at zone boundaries. The time evolution is carried out by using repeatedly the governing conservation law in differential form to replace time derivatives by space derivatives, which is the so-called Cauchy-Kovalewski or Lax-Wendroff procedure. The idea behind the GRP approach is a temporal Taylor series expansion of the state at the interface and this has been also used in the context of Lagrangian hydrodynamics [42] to achieve second order of accuracy in time.

In this work, we adopt the more recent formulation proposed in [27,26] in the Eulerian framework on fixed grids and in [6] for moving mesh schemes. A *weak* integral formulation

of the conservation law in space-time is used to obtain a space-time accurate representation of the solution within each cell.

2.3.1 Space-time coordinate vectors and nodal basis functions

First, let $\widetilde{\mathbf{x}} = (x, y, z, t)$ and $\widetilde{\boldsymbol{\xi}} = (\xi, \eta, \zeta, \tau)$ be the space-time coordinate vectors in the physical and in the reference system, respectively. The spatial mapping is given by (7), while the mapping in time is simply linear

$$t = t^n + \tau\,\Delta t, \qquad \tau \in [0, 1], \tag{13}$$

where t^n and Δt represent the current time and current time step, respectively. For evaluating the magnitude of Δt we use a classical CFL condition and a criterion to avoid a too large increase of cell volume in a single timestep [47,42].

The vector of state variables $\mathbf{Q}$ is known at t^n, therefore, recall (10), in each cell T_i^n and in a finite volume mass averaged sense, the piecewise constant data are given by

$$\mathbf{Q}_i^n \equiv \mathbf{Q}_i(t^n) = \frac{1}{m_i} \int_{T_i^n} \rho(\mathbf{x}, t^n)\mathbf{Q}(\mathbf{x}, t^n)\, dV. \tag{14}$$

Let us introduce a space-time polynomial $\mathbf{q}_h$ defined between t^n and t^{n+1}, that is between $\tau = 0$ and $\tau = 1$, which is expanded onto a nodal basis constituted of piecewise linear space-time basis functions $\theta(\widetilde{\boldsymbol{\xi}})$, that is

$$\mathbf{q}_h = \sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\widehat{\mathbf{q}}_{l,i}, \tag{15}$$

where $\widehat{\mathbf{q}}_{l,i}$ are the $\mathcal{L} = 2\mathcal{M}$ degrees of freedom with $\mathcal{M} = (d + 1)$. The space-time nodes which define the basis are simply given by the spatial nodes $\boldsymbol{\xi}_l$ at times $\tau = 0$ and $\tau = 1$, which correspond to the tensor product of the spatial nodes of classical conforming second order finite elements and the Newton-Cotes points in time. As a consequence, the space-time basis functions $\theta_l(\widetilde{\boldsymbol{\xi}})$ have the following form in multiple space dimensions:

$$\theta_l^{2D}(\widetilde{\boldsymbol{\xi}}) = \begin{cases} \theta_1 = (1 - \xi - \eta)(1 - \tau) \\ \theta_2 = \xi(1 - \tau) \\ \theta_3 = \eta(1 - \tau) \\ \theta_4 = (1 - \xi - \eta)\tau \\ \theta_5 = \xi\tau \\ \theta_6 = \eta\tau \end{cases}, \qquad \theta_l^{3D}(\widetilde{\boldsymbol{\xi}}) = \begin{cases} \theta_1 = (1 - \xi - \eta - \zeta)(1 - \tau) \\ \theta_2 = \xi(1 - \tau) \\ \theta_3 = \eta(1 - \tau) \\ \theta_4 = \zeta(1 - \tau) \\ \theta_5 = (1 - \xi - \eta - \zeta)\tau \\ \theta_6 = \xi\tau \\ \theta_7 = \eta\tau \\ \theta_8 = \zeta\tau \end{cases}, \tag{16}$$

Notice that, for $\tau = 0$, the space-time degrees of freedom $\widehat{\mathbf{q}}_{l,i}^0$ coincide with the degrees of freedom at time t^n. For instance, if a piecewise polynomial reconstruction of $\mathbf{q}$ was known

at t^n, then $\hat{\mathbf{q}}^0_{l,i}$ would be its components while $\hat{\mathbf{q}}^1_{l,i}$ would be the remaining components at t^{n+1}, see Figure 2 for an illustration in 2D.

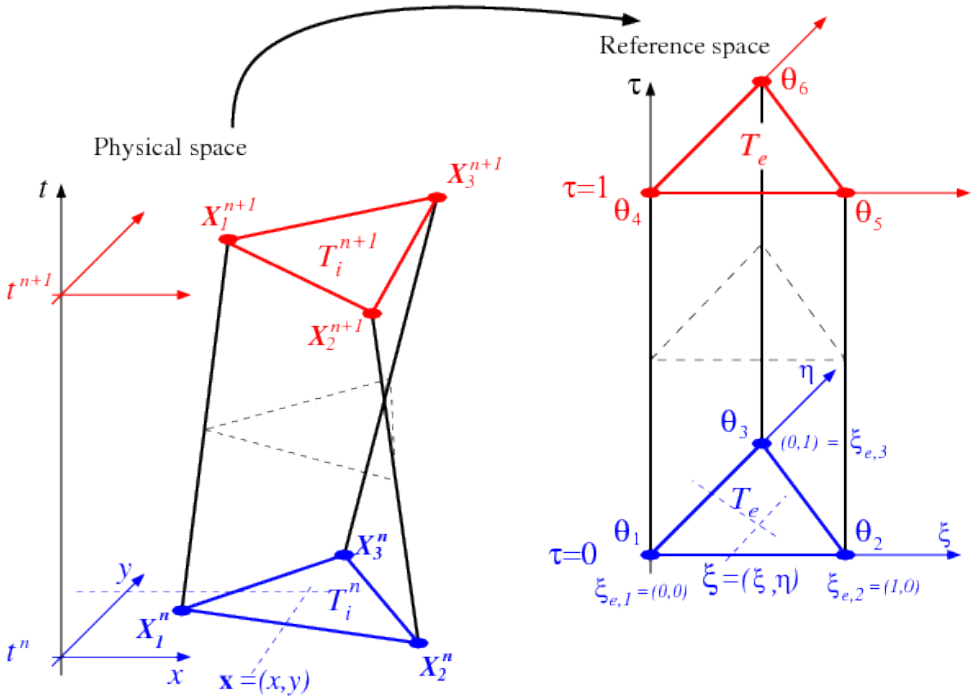

Fig. 2. 2D space-time mapping from the physical element $T_i(t)$ evolving from t^n to t^{n+1} and defined by vertex coordinates $\mathbf{X}_k^n = (x_k^n, y_k^n)$ and $\mathbf{X}_k^{n+1} = (x_k^{n+1}, y_k^{n+1})$ for $k = 1, 2, 3$ to the unit reference element $T_e \times [0, 1]$. On the reference space-time element are defined space-time basis functions θ_l with $l = 1, 2, 3$ for space-time degrees of freedom at time $\tau = 0$ and, θ_l with $l = 4, 5, 6$ for space-time degrees of freedom at time $\tau = 1$.

2.3.2 Piecewise linear reconstruction at t^n

A reconstruction procedure is carried out for obtaining a piecewise linear polynomial $\mathbf{w}_h^n$ within each cell T_i^n from the piecewise constant states $\mathbf{Q}_i^n$ in the neighborhood. This reconstruction polynomial is expressed in terms of a set of piecewise linear spatial basis functions $\psi_l(\boldsymbol{\xi})$, which are taken as $\theta_l(\boldsymbol{\xi}, 0)$, hence

$$\mathbf{w}_h^n = \sum_{l=1}^{\mathcal{M}} \psi_l(\boldsymbol{\xi}) \widehat{\mathbf{w}}_{l,i}^n, \tag{17}$$

where the $\mathbf{w}_{l,i}^n$ are the $\mathcal{M} = (d+1)$ *unknown* degrees of freedom (expansion coefficients). In order to perform the reconstruction procedure, we consider a so-called reconstruction stencil $\mathcal{S}_i = \bigcup_{j=1}^{n_e} T_{m(j)}^n$, where $1 \leq j \leq n_e$ is a local index that counts the elements belonging to the stencil, while $m(j)$ maps the local index to the global element numbers used in the mesh configuration (6). The stencil contains a total number of $n_e = d\mathcal{M}$ elements and is filled by the Voronoi neighbors of T_i^n (i.e. the neighbor elements sharing at least one vertex with element T_i^n). If n_e is not achieved then an algorithm recursively continues adding the neighbors of these neighbors until n_e is reached.
The reconstruction procedure is carried out in reference coordinates $\boldsymbol{\xi}$, and, integral con-

servation is required on each element $T_j^n \in \mathcal{S}_i$, that is

$$\frac{1}{|T_j^n|}\int\limits_{T_j^n} \sum_{l=1}^{\mathcal{M}} \psi_l(\boldsymbol{\xi})\widehat{\mathbf{w}}_{l,i}^n \, dV = \mathbf{Q}_j^n, \qquad \forall T_j^n \in \mathcal{S}_i, \tag{18}$$

where $|T_j^n|$ represents the volume of element T_j^n at time t^n. The above system (18) is overdetermined since $n_e > \mathcal{M}$ and therefore requires the use of a least squares technique. However, to enforce conservation of the reconstruction polynomial we must at least require that the above expression holds exactly for cell T_i^n, that is $\frac{1}{|T_i^n|}\int\limits_{T_i^n} \sum\limits_{l=1}^{\mathcal{M}} \psi_l(\boldsymbol{\xi})\widehat{\mathbf{w}}_{l,i}^n \, dV = \mathbf{Q}_i^n$ is added as a linear constraint to the overdetermined system (18). The resulting *constrained* least squares problem is then solved with a classical Lagrangian multiplier approach, see [28]. To optimize the construction of the reconstruction matrix, which is given by the integrals appearing in (18), an exact *analytical* integration is performed by inserting the transformation (7) into the above expression (18). The resulting formula is a function of the physical vertex coordinates $\mathbf{X}_{k,j}^n$ of the control volume T_j^n.

At the end of this reconstruction step, we have access to piecewise linear reconstructed state variables (17) at time t^n in each cell T_i^n. This reconstruction will be used as initial condition for the following ADER predictor step.

2.3.3 Space-time ADER predictor

To perform the predictor stage within the ADER technology, system (1) is first written in a *non-conservative* form as

$$\frac{d}{dt}\left(\frac{1}{\rho}\right) - \frac{1}{\rho}\left(\nabla \cdot \mathbf{u}\right) = 0, \qquad \frac{d}{dt}\mathbf{u} + \frac{1}{\rho}\left(\nabla p\right) = 0, \qquad \frac{d}{dt}E + \frac{1}{\rho}\left(\nabla \cdot (p\mathbf{u})\right) = 0, \tag{19}$$

where $\frac{d}{dt}$ denotes the material derivative. It further reads compactly as follows

$$\frac{d}{dt}\mathbf{Q} + \frac{1}{\rho}\left(\nabla \cdot \mathbf{F}(\mathbf{Q})\right) = \mathbf{0}. \tag{20}$$

Then, the governing equations (20) are expressed in the reference system as

$$\frac{\partial}{\partial \tau}\mathbf{Q} + \Delta t\, \mathbf{H} = \mathbf{0}, \tag{21}$$

with the abbreviation

$$\mathbf{H} = \left(\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\xi}} \cdot \frac{\partial \boldsymbol{\xi}}{\partial t} + \frac{1}{\rho} \cdot \left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^T \nabla_{\boldsymbol{\xi}} \mathbf{F}(\mathbf{Q})\right). \tag{22}$$

Here, the ∇ operator in $\boldsymbol{\xi}$ is used and it is defined by

$$\nabla = \begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix} = \begin{pmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} & \frac{\partial \zeta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} & \frac{\partial \zeta}{\partial y} \\ \frac{\partial \xi}{\partial z} & \frac{\partial \eta}{\partial z} & \frac{\partial \zeta}{\partial z} \end{pmatrix} \begin{pmatrix} \frac{\partial}{\partial \xi} \\ \frac{\partial}{\partial \eta} \\ \frac{\partial}{\partial \zeta} \end{pmatrix} := \left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^T \nabla_{\boldsymbol{\xi}}. \tag{23}$$

In (22) $\mathbf{Q}$ and $\mathbf{H}$ are approximated for a cell T_i by an expansion $\mathbf{q}_{h,i}$ and $\mathbf{h}_{h,i}$ over the space-time basis functions, thus

$$\mathbf{q}_{h,i} = \sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\widehat{\mathbf{q}}_{l,i}, \qquad \mathbf{h}_{h,i} = \sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\widehat{\mathbf{h}}_{l,i}. \tag{24}$$

We insert the above expressions into the governing equations (21), hence multiplying by a test function $\theta_k(\widetilde{\boldsymbol{\xi}})$ one gets

$$\theta_k(\widetilde{\boldsymbol{\xi}})\frac{\partial}{\partial\tau}\left(\sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\,\widehat{\mathbf{q}}_{l,i}\right) + \Delta t\,\theta_k(\widetilde{\boldsymbol{\xi}})\left(\sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\,\widehat{\mathbf{h}}_{l,i}\right) = \mathbf{0}. \tag{25}$$

Then, (25) is integrated over the space-time reference element $T_e \times [0,1]$, obtaining the weak formulation of the non-conservative governing system (20)

$$\mathbf{K}_\tau \widehat{\mathbf{q}}_{l,i} = -\Delta t\,\mathbf{M}\widehat{\mathbf{h}}_{l,i}, \tag{26}$$

where the following abbreviations have been used:

$$\mathbf{K}_\tau = \int_0^1 \int_{T_e} \theta_k \frac{\partial \theta_l}{\partial \tau}\, d\boldsymbol{\xi}\, d\tau, \quad \mathbf{M} = \int_0^1 \int_{T_e} \theta_k\,\theta_l\, d\boldsymbol{\xi}\, d\tau. \tag{27}$$

To solve (26), all space and time degrees of freedom $\widehat{\mathbf{q}}_{l,i}$ must be determined. However, at relative time $\tau = 0$ the reconstruction polynomial $\mathbf{w}_h$ (see Section 2.3.2) directly provides part of the unknown degrees of freedom, i.e. $\widehat{\mathbf{q}}^0_{l,i}$. As such, those degrees of freedom are moved to the right hand side of (26), while the remaining degrees $\widehat{\mathbf{q}}^1_{l,i}$ at time $\tau = 1$ are the true unknowns and have to be evaluated by solving iteratively the nonlinear algebraic equation system

$$\mathbf{K}_\tau \widehat{\mathbf{q}}^{r+1}_{l,i} = -\Delta t\,\mathbf{M}\widehat{\mathbf{h}}^{r}_{l,i}, \tag{28}$$

with subscript r denoting the iteration number of the non-linear solver. In [26] it is demonstrated that, for linear homogeneous hyperbolic PDE, system (28) admits a unique solution that is obtained after at most $\mathcal{O}$ iterations, with $\mathcal{O}$ denoting the order of accuracy of the numerical scheme. Together with the solution $\mathbf{q}_h$, we also have to consider the geometry evolution of element T_i that is governed by the trajectory equation (2), which links spatial positions and velocity field. The trajectory equation is solved here relying on the same approach used for the numerical solution (28). Indeed, the space variable and velocity field inside element $T_i(t)$ are spanned over the space-time basis as

$$\mathbf{x}_h = \sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\widehat{\mathbf{x}}_{l,i}, \qquad \mathbf{u}_h = \sum_{l=1}^{\mathcal{L}} \theta_l(\widetilde{\boldsymbol{\xi}})\widehat{\mathbf{u}}_{l,i}, \tag{29}$$

and equation (2) is multiplied by a test function θ_k, integrated over the space-time reference element and solved by the iterative method

$$\mathbf{K}_\tau \widehat{\mathbf{x}}^{r+1}_{l,i} = -\Delta t\,\mathbf{M}\widehat{\mathbf{u}}^{r}_{l,i}. \tag{30}$$

Also in this case the degrees of freedom $\widehat{\mathbf{x}}^0_{l,i}$ at time $\tau = 0$ are known because they are the vertex coordinates of element T_i at time t^n, i.e. $\widehat{\mathbf{x}}^0_{l,i} = \mathbf{X}^n_k$.

The resulting system (28)-(30) is solved in a *coupled* manner until the residuals of both expressions are less than a prescribed tolerance, typically set to 10^{-12}. Once the above

procedure has been carried out for all elements of the computational domain, we end up with an element-local second order predictor for the numerical solution $\mathbf{q}_h$ on a moving and deforming cell defined by space-time coordinates $\mathbf{x}_h$. This prediction of the solution will be the main ingredient for the corrector step of the ADER methodology, i.e. the node-based finite volume scheme, and substantially differs from the GRP-type approach proposed in [48,42].

2.4 Cell-centered Lagrangian FV scheme based on nodal solver

Equipped with a local piecewise linear space-time prediction of the solution $\mathbf{q}_h$ in each cell, we proceed by deriving a pure cell-centered Lagrangian scheme using the concepts originally to be found in [47,14,42]. More precisely, our version closely follows the work done in [42], but for the sake of consistency, we briefly remind the main steps of the derivation.

2.4.1 Cell-centered Lagrangian scheme — first-order discretization

Since 2005 all these newly developed cell-centered Lagrangian schemes are founded on a derivation that maintains the consistency between the purely Lagrangian nature (no mass flux), the mesh displacement and the conservative update of the physical quantities (momentum and total energy). These numerical methods rely on a so-called *nodal solver*, which could be interpreted as a local approximate multidimensional Riemann solver at a given node of the mesh. The first-order discretization strictly follows the Eucclhyd scheme proposed in [42], where one can find all the details of its derivation.

Let us summarize the algorithm to compute $\mathbf{Q}_i^{n+1}, \mathcal{T}_\Omega^{n+1}$ from $\mathbf{Q}_i^n, \mathcal{T}_\Omega^n$ in a conservative and Lagrangian manner, assuming that the discretization in time is a simple first-order in time forward Euler scheme:

$$\left(\frac{1}{\rho_i}\right)^{n+1} = \left(\frac{1}{\rho_i}\right)^{n} + \frac{\Delta t}{m_i} \sum_{r\in\mathcal{P}(i)} \mathbf{u}_r^n \cdot L_{r,i}^n \mathbf{n}_{r,i}^n, \tag{31a}$$

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \frac{\Delta t}{m_i} \sum_{r\in\mathcal{P}(i)} \boldsymbol{\mathcal{F}}_{r,i}^n, \tag{31b}$$

$$E_i^{n+1} = E_i^n - \frac{\Delta t}{m_i} \sum_{r\in\mathcal{P}(i)} \boldsymbol{\mathcal{F}}_{r,i}^n \cdot \mathbf{u}_r^n. \tag{31c}$$

where $\mathcal{P}(i)$ is the set of nodes of cell i, and $L_{r,i}\mathbf{n}_{r,i}$ is its outward pointing normal associated to node $\mathbf{X}_{r,i}$, $r = 1, \cdots, d+1$. In other words, this normal is an average of the d directions of faces/edges of T_i impinging on node $\mathbf{X}_{r,i}$, see Figure 3 for a 2D illustration. The node velocity $\mathbf{u}_r^n$ and the subcell force $\boldsymbol{\mathcal{F}}_{r,i}^n$ are given by

$$\mathbf{u}_r^n = \mathrm{M}_r^{-1} \left(\sum_{i\in\mathcal{I}(r)} p_i^n \, L_{r,i}^n \mathbf{n}_{r,i}^n + \mathrm{M}_{r,i}^n \mathbf{u}_i^n \right), \qquad \boldsymbol{\mathcal{F}}_{r,i}^n = p_i^n \, L_{r,i}^n \mathbf{n}_{r,i}^n - \mathrm{M}_{r,i}^n (\mathbf{u}_r^n - \mathbf{u}_i^n), \tag{32}$$

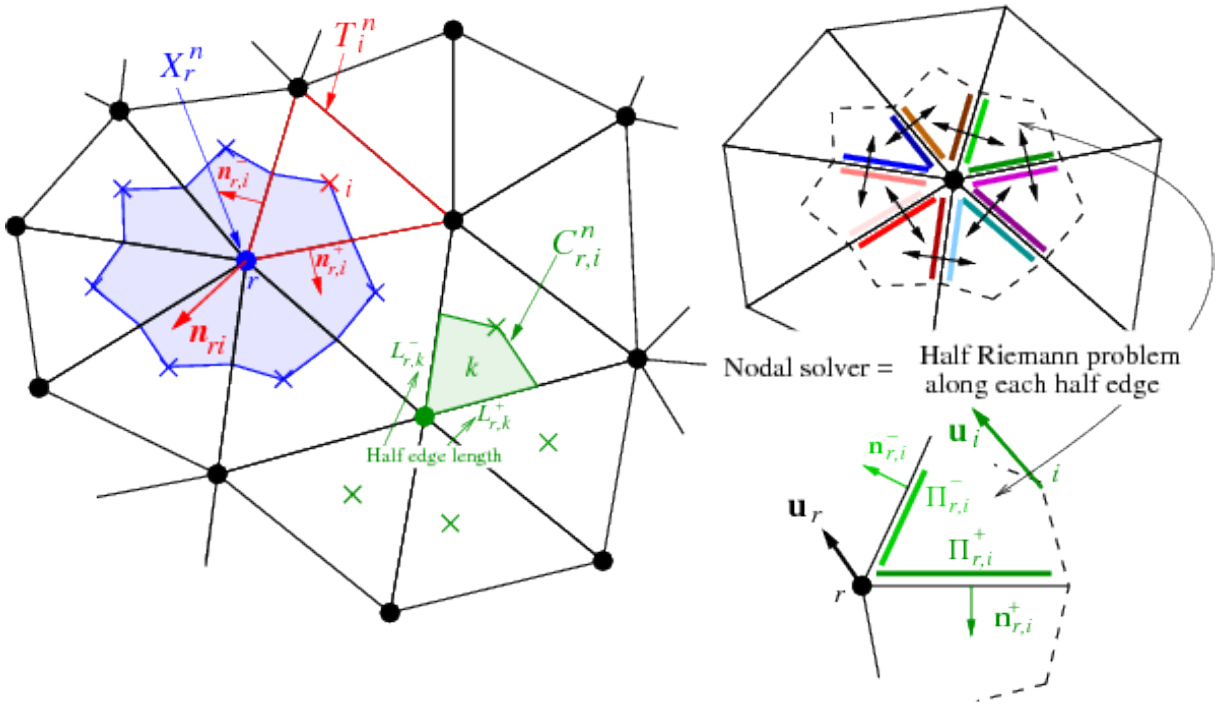

Fig. 3. Notation for a 2D mesh composed of triangles. Left: a cell T_i^n (red) is decomposed into three subcells $C_{r,i}^n$ (joining the cell center i, two mid-edge point and the current node r). The dual cell is the union of all subcells surrounding node r (blue). A given subcell has two outward pointing normals to its edges $\mathbf{n}_{r,i}^-$, $\mathbf{n}_{r,i}^+$ which sum up to a geometrical vector $\mathbf{n}_{r,i}$. The half edge lengths are denoted by $L_{r,i}^-$ and $L_{r,i}^+$. Right: sketch of the nodal solver, each subcell has two half-edge pressures (color bars) denoted by $\Pi_{r,i}^+$, $\Pi_{r,i}^-$ and the nodal solver computes the nodal velocity $\mathbf{u}_r$ knowing the pressures Π_i and the surrounding cell velocities $\mathbf{u}_i$.

with $\mathcal{I}(r)$ denoting the set of indices of the cells surrounding node r. The definitions of the matrices $\mathrm{M}_{r,i}^n$ and M_r follow from [42] and they read

$$\mathrm{M}_{r,i} = z_c \left[L_{r,i}^+ (\mathbf{n}_{r,i}^+ \otimes \mathbf{n}_{r,i}^+) + L_{r,i}^- (\mathbf{n}_{r,i}^- \otimes \mathbf{n}_{r,i}^-) \right], \qquad \mathrm{M}_r = \sum_{i \in \mathcal{I}(r)} \mathrm{M}_{r,i}. \tag{33}$$

Finally, the new node position for any vertex r is given by

$$\mathbf{x}_r^{n+1} = \mathbf{x}_r^n + \Delta t \, \mathbf{u}_r^n. \tag{34}$$

2.4.2 Second-order in space and time cell-centered Lagrangian ADER scheme

A second-order accurate extension in space of the previous solver is typically obtained by feeding the subcell forces and the node values (32) with a piecewise linear reconstruction of the pressure and velocity, evaluated at node $\mathbf{X}_r$:

$$\mathbf{u}_r^* = \mathrm{M}_r^{-1} \left(\sum_{i \in \mathcal{I}(r)} p_i^*(\mathbf{X}_r) \, L_{r,i}^n \mathbf{n}_{r,i}^n + \mathrm{M}_{r,i}^n \mathbf{u}_i^*(\mathbf{X}_r) \right), \quad \boldsymbol{\mathcal{F}}_{r,i}^* = p_i^*(\mathbf{X}_r) \, L_{r,i}^n \mathbf{n}_{r,i}^n - \mathrm{M}_{r,i}^n \left(\mathbf{u}_r^* - \mathbf{u}_i^*(\mathbf{X}_r) \right), \tag{35}$$

where for all $\mathbf{X}$ in cell T_i^n we have

$$p_i^*(\mathbf{X}) = \sum_{l=1}^{\mathcal{M}} \widehat{p}_{l,i}^* \Psi_l(\mathbf{X}), \qquad \mathbf{u}_i^*(\mathbf{X}) = \sum_{l=1}^{\mathcal{M}} \widehat{\mathbf{u}}_{l,i}^* \Psi_l(\mathbf{X}). \tag{36}$$

As previously seen, the first-order in time scheme considers $t^* = t^n$. A second-order accurate scheme in time can be obtained by employing either a predictor-corrector strategy or a GRP methodology [42]. Both techniques produce a prediction of the solution at t^{n+1}, which is later averaged in time at $t^* = t^{n+1/2} = \frac{1}{2}(t^n + t^{n+1})$. In the case of a predictor-corrector scheme in time, the geometry is considered at time $t^{n+1/2}$ mainly because the predictor step provides a valid solution and a valid mesh at t^{n+1}. In the GRP methodology the mesh position and the associated geometrical vectors can be easily frozen at time t^n as well as the subcell matrix which depends on the density and the sound speed. Nevertheless, even though the computation might become cumbersome, also in the GRP context it is possible to take into account the variation of the geometry and the corner matrix, especially for the purpose of a numerical convergence analysis, see [44]. In our approach the geometry is classically kept at time t^n for the GRP solution.

In this work, the second-order extension of the previous Lagrangian scheme is obtained by the use of the space and time ADER predictor solution $\mathbf{q}_h$ evaluated at each vertex location $\mathbf{X}_r$ $r = 1, 2, ..., d+1$ and at the half time level $t^{n+\frac{1}{2}}$. Since the predictor is linear within each cell, this simply consists in evaluating $\mathbf{u}_h$ and $\mathbf{p}_h$ in reference space as

$$\mathbf{p}_i^{n+1/2}(\mathbf{X}_r) = \frac{1}{2}\left(\widehat{\mathbf{p}}_{r,i} + \widehat{\mathbf{p}}_{r+d+1,i}\right), \quad \mathbf{u}_i^{n+1/2}(\mathbf{X}_r) = \frac{1}{2}\left(\widehat{\mathbf{u}}_{r,i} + \widehat{\mathbf{u}}_{r+d+1,i}\right), \quad r = 1, 2, ..., d+1.$$

These values feed the nodal solver (32) for which geometrical vectors and subcell matrices are evaluated at t^n. Then, the conservative variables are advanced in time by means of (31), while the node position by

$$\mathbf{x}_r^{n+1} = \mathbf{x}_r^n + \Delta t\, \mathbf{u}_r^*, \tag{38}$$

which ends the second-order extension of the numerical method.

2.5 a posteriori limiting

Nominally, the previously described numerical method is second-order accurate in space and time. At this point, this scheme has no embedded limiting or nonlinear stabilization procedure to handle shocks and steep gradients. As such, this accurate Lagrangian scheme will inexorably present spurious numerical oscillations known as Gibbs phenomenon.

Classical Eulerian ADER schemes usually employ WENO or slope limiters to add extra dissipation to the construction of the predictor. The same solution is also adopted for classical cell-centered Lagrangian schemes. Slope limiters are used when velocity and pressure are reconstructed. The lack of mesh symmetry when employing component-by-component velocity limiting has led to the development of frame-invariant tensorial limiting for vector fields [40,41].

Most of the limiting strategies are based on the belief that the solution at t^n is sufficient to predict where, and how much, dissipation is needed to compute a valid solution at t^{n+1} in the sense of the previous properties. *A contrario*, a recent methodology, called MOOD [19], follows an *a posteriori* philosophy: a candidate solution is computed at t^{n+1}, this solution is checked against detection criteria to determine which cells demand more dissipation, and, these troubled cell values are recomputed starting again from t^n us-

ing locally a more dissipative scheme. This approach has been successfully employed to stabilize Eulerian finite volume schemes [19,39], Discontinuous Galerkin schemes [63,8], Smoothed Particle Hydrodynamics (SPH) [52], direct ALE FV schemes [10,22,5], Asymptotic Preserving FV schemes [4], for different systems of PDEs, namely hydrodynamics, magneto-hydrodynamics [39], shallow-water [20], *etc.*
We propose to adopt such *a posteriori* limiting strategy for our Lagrangian ADER scheme by employing the unlimited second-order scheme for valid cells, and the first-order accurate one for the troubled cells. This strategy is entirely based on the assumption of a first-order scheme which exhibits good properties, such as positivity and monotonicity. A proof of the positiveness of density and internal energy for the cell-centered first-order discretization (31) is given in [60], which we rely on. Since the first order scheme is positivity preserving, the MOOD extension to second order is *by construction* positivity preserving, since positivity of the solution is explicitly checked *a posteriori* in the physical admissibility criteria. Whenever and wherever the discrete solution is detected to be not positive, the corresponding cells will be updated by the positivity-preserving first order scheme. This makes the *a posteriori* limiting approach used in the present paper *substantially different* from existing *a priori* limiting strategies classically employed in pure Lagrangian schemes.

2.5.1 MOOD loop

Starting at t^n with valid cell-centered data $\mathbf{Q}_i^n$ and mesh T_i^n, one proceeds with the very first step of the MOOD loop which is the computation of the so-called unlimited candidate solution $\mathbf{Q}_h^{n+1,HO}$, $T_i^{n+1,HO}$ for all cells. Each cell is assigned its cell polynomial degree $M_i = 1$ stating that piecewise linear reconstructions are employed.
The second step consists in testing the candidate solution and mesh against detection criteria, see next section, to determine if the solution is acceptable. In this case the cell is flagged as acceptable, otherwise the cell is marked as troubled.
If there exist troubled cell(s), during the so-called decrementing step, the cell polynomial degree for troubled cells is dropped to $M_i = 0$ in order to use locally the first order accurate scheme. However, for this cell, it is mandatory that the three nodal solvers are fed with first-order accurate reconstructions. Hence, the effective polynomial degree for the reconstructions at node $\mathbf{X}_{i,k}$ is the minimum of the cell polynomial degrees in its surrounding

$$M_{i,r} = \min_{j \in \mathcal{I}(r)} M_j, \tag{39}$$

see Figure 4 for an illustration. As a consequence the three nodal solvers of a given cell, which has not been detected as troubled, may not use the same polynomial degrees. Such cells are referred to as *mixed cells*, since they exhibit some vertexes updated with the second-order scheme, and the remaining ones with the first-order approximation. Therefore a cell can be updated either with a second-order, first-order accurate scheme, or a mix between those two (respectively the blue cell, the black one or other color cell of Figure 4). In fact the set of troubled cells sent back in time for re-computation is constituted of the invalid cells, the degree of which has dropped, and any of their neighbor cells. Finally, the last step consists in discarding the numerical solution for the cells belonging to the set of troubled ones, and, recomputing the solution, starting again at t^n, but using the new polynomial degrees $M_{i,r}$ which have been updated during the decrementing stage.
The second iteration of the MOOD loop then begins. A new candidate solution at t^{n+1}

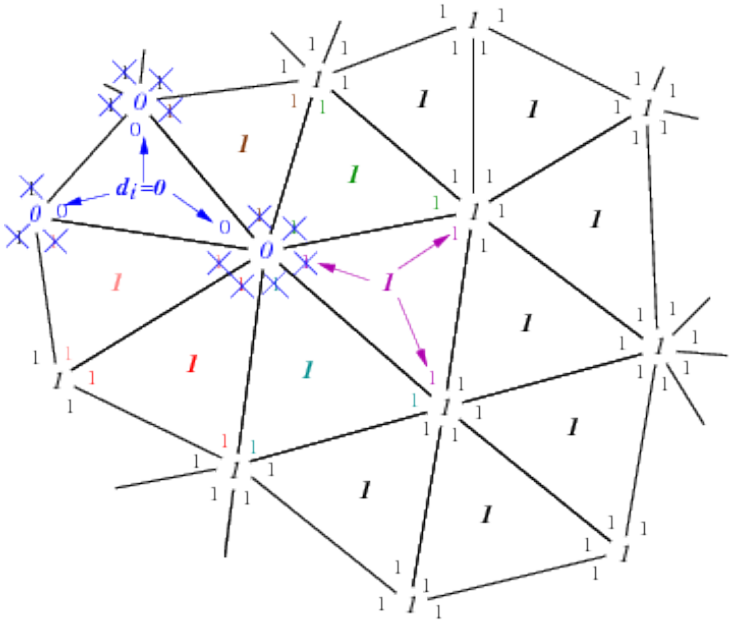

Fig. 4. Illustration of the cell polynomial degree M_i decrementing and its influence on the effective polynomial degree used at each node. The blue cell is flagged for $M_i = 0$ (first order accurate) while the purple one with $M_j = 1$ (second order accurate). For the blue cell to be updated with the first order accurate scheme then its three nodal solvers must be fed with first-order reconstructions, hence the reduction of the node polynomial degree to 0 for all surrounding cells. Therefore the purple cell uses second order accurate reconstructions for two nodes, and first order accurate for one node in contact with the blue cell.

in those troubled cells is computed. Then this solution in troubled cells is tested against the detection criteria. If troubled cells are still present, the MOOD loop pursues with decrementing and re-computation until no more bad cells are detected. A cell with $M_i = 0$ is considered as valid because the first-order accurate scheme is the most dissipative scheme to be tried, and, its solution must be accepted as a valid one. Note that more than two iterations may be needed because mixed cells could be destabilized and detected as troubled, and, consequently, may demand a re-computation. Nonetheless the MOOD loop always converges because either all cells are updated with $M_i = 0$ after a possibly large but finite number of re-computations, or, the solution is detected as a valid one before. The observations have shown that, on average, few percents of cells are troubled per timestep. Rarely, depending on the test case, the percentage may exceed 20%. In this work we have only performed two iterations because the dissipation spreads rapidly; from the cell degree 0 to the three nodal solvers, and, partially to all neighbor cells.

2.5.2 Detection criteria

Some fundamental objects in a MOOD loop are the criteria chosen to detect the occurrence of a troubled cell. In this work we rely on

- physical admissibility criteria: positivity of density, specific internal energy (and pressure), and cell volumes;
- computer admissibility criteria: representability of numerical values (Not-a-Number (NaN), Infinites (Inf), etc.);
- numerical admissibility criteria: essentially non-oscillatory behavior by means of a Relaxed Discrete Maximum Principle on conservative variables.

More precisely, a cell is flagged as troubled at t^{n+1} if one can not ensure either its physical admissibility

$$\rho_i^{n+1} > 0, \quad \text{or} \quad E_i^{n+1} - \frac{1}{2}\|\mathbf{u}\|^2 > 0, \tag{40}$$

or its geometrical admissibility

$$T_i^{n+1} \text{ is unfolded,} \tag{41}$$

or its computer admissibility

$$\mathbf{Q}_i^{n+1} \text{ and } \mathbf{F}(\mathbf{Q}_i^{n+1}) \text{ are representable numbers in the computer,} \tag{42}$$

or its numerical admissibility

$$-\delta(\mathbf{Q}_{max,i} - \mathbf{Q}_{min,i}) + \mathbf{Q}_{min,i} \leq \mathbf{Q}_i^{n+1} \leq \mathbf{Q}_{max,i} + \delta(\mathbf{Q}_{max,i} - \mathbf{Q}_{min,i}), \tag{43}$$

where $\mathbf{Q}_{max,i} = \max_{j\in\mathcal{S}_i} \mathbf{Q}_j^n$ and $\mathbf{Q}_{min,i} = \min_{j\in\mathcal{S}_i} \mathbf{Q}_j^n$ with $\delta = 10^{-3}$. Otherwise the cell is valid and accepted.

Notice that other non-oscillatory detectors could be designed in a more or less complex fashion, but we have observed that this relatively simple design (43) provides already reasonable results. We point out that Eqn.(43) does not lead to a frame invariant limiter regarding the velocity vector, hence the symmetry present in a physical problem could be not captured properly. To overcome this issue, one might consider for example the VIP limiter strategy proposed in [30].

Also, we would like to remark that the occurrence of Not-a-Number or Infinite can be cured in our *a posteriori* treatment, while in classical *a priori* ones, such occurrence leads to a code crashing. Finally, the positivity is trivially ensured if the first-order accurate numerical scheme is positivity preserving, which is the case provided that the time step is properly monitored [60]. For the sake of clarity, the physical admissibility conditions given by (40) depend on the physical phenomenon, namely the governing balance laws, the associated equation of state, some constraints like a divergence-free velocity or magnetic vector field, *etc.*

3 Test problems

In the following we present the results for a set of benchmark test cases specifically dedicated to test and verify Lagrangian ADER-MOOD finite volume schemes proposed in this paper. Note that due to the Lagrangian motion, the test cases should not involve large vorticity and shear to avoid mesh tangling. Boundary conditions are imposed in a compatible way with our numerical scheme, see Appendix A for details.

For each test problem the CFL stability coefficient is assumed to be CFL = 0.4 and CFL = 0.25 for two and three space dimensions, respectively. The time-dependent computational domain is addressed with $\Omega(t)$, while $\mathbf{W}(\mathbf{x}, t = 0) = (1/\rho_0, u_0, v_0, w_0, p_0)$ denotes the vector of primitive variables which is typically used to provide the initial condition for the forthcoming test problems.

3.1 Kidder problem

The Kidder test case [37] describes the isentropic compression of a shell filled with perfect gas, that is initially bounded between the internal radius $r_i = 0.9$ and the external radius $r_e = 1.0$ of the shell, with the general radial coordinate $r = \sqrt{\mathbf{x}^2}$ with $r_i(t) \leq r \leq r_e(t)$. The ratio of specific heats is $\gamma = 1 + \frac{2}{d}$ and the initial condition in primitive variables reads

$$\mathbf{W}(r,0) := \begin{pmatrix} 1/\rho(r) \\ \mathbf{v}(r) \\ p(r) \end{pmatrix} = \begin{pmatrix} \left(\frac{r_e^2 - r^2}{r_e^2 - r_i^2} \rho_i^{\gamma-1} + \frac{r^2 - r_i^2}{r_e^2 - r_e^2} \rho_e^{\gamma-1} \right)^{-\frac{1}{\gamma-1}} \\ 0 \\ s\rho(r)^\gamma \end{pmatrix}, \tag{44}$$

where $\rho_i = 1$ and $\rho_e = 2$ are the initial values of density at the corresponding frontier of the shell, while the initial entropy distribution is assumed to be uniform, hence $s = \frac{p}{\rho^\gamma} = 1$. The final time $t_{\text{final}} = \frac{\sqrt{3}}{2}\tau$ is determined in such a way that the shell is bounded by $0.45 \leq r \leq 0.5$. t_{final} and it depends on the focalisation time $\tau = \sqrt{\frac{\gamma-1}{2} \frac{(r_e^2 - r_i^2)}{c_e^2 - c_i^2}}$, with the internal and external sound speeds $c_\alpha = \sqrt{\gamma \frac{p_\alpha}{\rho_\alpha}}$, where α stands for e or i. Pressure boundary conditions are imposed on the internal and external frontiers of the shell according to the exact solution.
This multidimensional test case is appropriate to observe the numerical convergence on a sequence of successively refined computational meshes. The error ϵ between the cell-centered numerical solution $\mathbf{W}_i$ at position r_i at t_{final}, and the analytical solution $\mathbf{W}^{ex}(r_i, t_{\text{final}})$, available in [37] is reported in this section. The error is evaluated using L_1, L_2 and L_∞ norms at the final time t_{final} according to

$$\epsilon_{L_1} = \sum_{i=1}^{N_E} |\mathbf{W}_i^{ex} - \mathbf{W}_i| \, |T_i|, \quad \epsilon_{L_2} = \sum_{i=1}^{N_E} \sqrt{(\mathbf{W}_i^{ex} - \mathbf{W}_i)^2 \, |T_i|}, \quad \epsilon_{L_\infty} = \max_i |\mathbf{W}_i^{ex} - \mathbf{W}_i| \,,$$

with $|T_i|$ denoting the volume of cell T_i at the final time t_{final}. The resulting numerical errors and associated convergence rates are listed in Table 1 for pressure as well as for the location of the internal and external frontier (radius). The corresponding errors $\epsilon_{R_{int}}, \epsilon_{R_{ext}}$

are estimated as the arithmetic average of the difference between the analytical and the numerical radial coordinate for each node lying on the internal and external frontier. We observe that second order of accuracy is achieved in the L_1 and L_2 norm both in 2D and in 3D on fully unstructured meshes, with $h(\Omega(t_{\text{final}})) = \max_i |T_i|^{\frac{1}{d}}$ representing the characteristic mesh size at the final time. Finally, Figure 5 shows the initial and final pressure distribution and the mesh configuration in 2D (left panels) and 3D (right panels). Note that the detection procedure does not flag any cell both in 2D and 3D, meaning that the schemes are run in their optimal second-order accurate mode. This is appropriate because the flow is smooth and our mesh fine enough to capture its features, as highlighted in Figure 5. We would like to emphasize that the fact that all cells are updated with the second order scheme is a *result* of the computation and the limiter detects by itself that this test problem is smooth and needs no limiting. In other words: also in this smooth test problem, *all* cells are explicitly checked *a posteriori* by the MOOD limiter for the numerical and physical admissibility criteria, but no cell violates these criteria. We will observe in those tests where shocks and steep gradients are involved that the limiting procedure will appropriately detect bad cells.

2D	**Pressure**						**Radius**			
$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	$\epsilon_{R_{int}}$	$\mathcal{O}(R_{int})$	$\epsilon_{R_{ext}}$	$\mathcal{O}(R_{ext})$
2.40E-02	2.71E+00	-	6.06E+00	-	2.24E+01	-	2.01E-02	-	5.84E-02	-
1.37E-02	9.34E-01	1.9	2.56E+00	1.5	1.61E+01	0.6	1.57E-02	0.4	5.55E-02	0.1
4.51E-03	5.72E-02	2.5	2.10E-01	2.2	2.79E+00	1.6	2.50E-03	1.7	8.77E-03	1.7

3D	**Pressure**						**Radius**			
$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	$\epsilon_{R_{int}}$	$\mathcal{O}(R_{int})$	$\epsilon_{R_{ext}}$	$\mathcal{O}(R_{ext})$
2.18E-02	3.57E+00	-	7.47E+00	-	5.55E+01	-	3.93E-02	-	1.35E-01	-
1.66E-02	2.34E+00	1.6	5.08E+00	1.4	4.68E+01	0.6	2.65E-02	1.5	1.31E-01	0.1
1.20E-02	1.11E+00	2.3	2.76E+00	1.9	3.77E+01	0.7	1.19E-02	2.4	6.74E-02	2.0

Table 1
Numerical errors and convergence rates for the Kidder problem computed with second order of accuracy Lagrange ADER scheme. The error norms refer to the variable p (pressure) or to the internal and external radial position at the final time $t = t_{\text{final}}$.

3.2 Smooth vortex problem

In the previous test case, no problematic cells were detected, hence running the simulation with the full second-order accurate scheme. Let now consider the smooth vortex evolution problem proposed in [34,64], that consists in a mean flow with $\rho = u = v = p = 1$ onto which some perturbations are superimposed. The initial computational domain is the square $\Omega(t = 0) = [0; 10] \times [0; 10]$ with periodic boundaries set everywhere. According to [64], the final time of the simulation is $t_{\text{final}} = 0.01$ and the vortex strength is chosen to be $\epsilon = 10.0820$, so that the lowest density and pressure of the exact solution are very close to zero, i.e. 7.8×10^{-15} and 1.7×10^{-20}. The exact solution is given by the initial condition, which is specified in [64], convected up to the final time with the mean vortex velocity $\mathbf{v}_c = (u, v) = (1, 1)$. Table 2 shows the error norms for density and pressure at time t_{final} as well as the number of cells which have been detected as problematic in the *first* iterations of each simulation. After one or at most two time steps no more bad cells are present, hence running this test problem with all cells updated with the full second

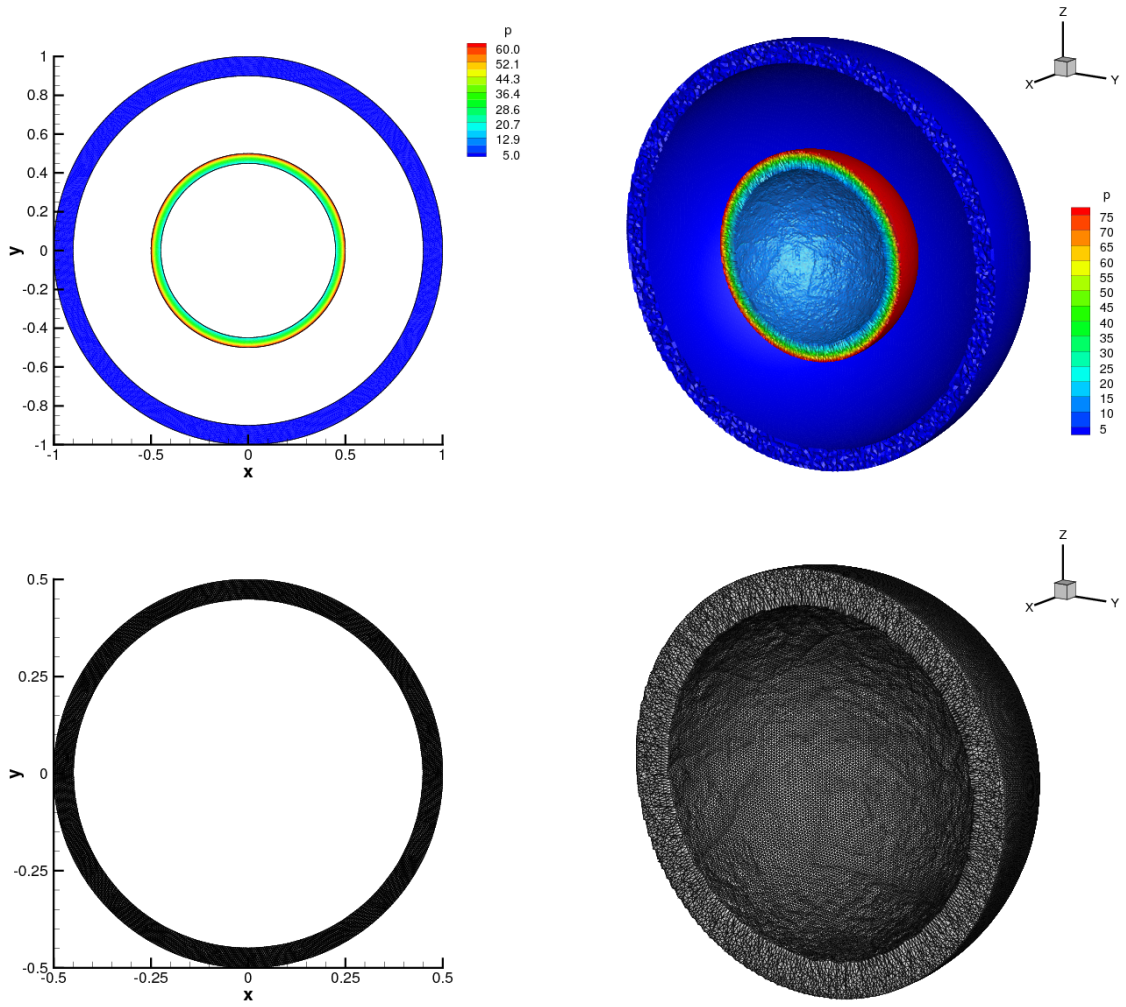

Fig. 5. Numerical results for the Kidder problem in 2D (left column) and in 3D (right column) — Top row: pressure distribution at time $t = 0$ and $t = t_{\text{final}}$ — Bottom row: mesh configuration at final time $t = t_{\text{final}}$ only.

order scheme. Even in this case, with more than 50% of problematic control volumes in the first iteration steps, our algorithm is able to achieve second order of accuracy in space and time.

Density

$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
3.26E-01	6.82E-02	-	1.88E-02	-	1.34E-02	-
2.48E-01	3.92E-02	2.0	9.90E-03	2.3	7.13E-03	2.3
1.63E-01	2.13E-02	1.5	5.20E-03	1.5	4.59E-03	1.0
1.28E-01	1.37E-02	1.8	3.17E-03	2.1	2.84E-03	2.0

Pressure

$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$
3.26E-01	7.98E-02	-	2.12E-02	-	1.38E-02	-
2.48E-01	4.68E-02	1.9	1.14E-02	2.3	8.47E-03	1.8
1.63E-01	2.55E-02	1.5	5.99E-03	1.5	5.09E-03	1.2
1.28E-01	1.65E-02	1.8	3.59E-03	2.1	2.69E-03	2.6

Bad cells

$h(\Omega(t_{\text{final}}))$	iteration number	# bad cells	% bad cells
3.26E-01	1	827	63.71
2.48E-01	1	1487	64.87
1.63E-01	1	3236	62.47
	2	482	9.3
1.28E-01	1	5709	62.11
	2	2958	32.18

Table 2
Numerical errors and convergence rates for the smooth vortex problem computed with second order of accuracy Lagrange ADER scheme. The error norms refer to the variable ρ (density) and p (pressure) computed at the final time $t = t_{\text{final}}$, while $h(\Omega(t_{\text{final}}))$ is the final mesh size. Finally, also the number of bad cells is reported for each simulation with the corresponding percentage w.r.t. the total number of elements contained in the computational mesh. After one or at most two iterations no more problematic cells are detected.

3.3 Multidimensional Sod problem

This test problem [58] involves simple cylindrical/spherical waves: a shock wave, a contact discontinuity as well as a rarefaction fan. The initial computational domain $\Omega(t=0)$ is the two-dimensional unit circle or the three-dimensional unit sphere of radius $R=1$, which is accordingly paved with a characteristic mesh size of $h=1/100$ yielding a total number of $N_E = 68324$ triangles or $N_E = 2538426$ tetrahedra. Two different states, namely $\mathbf{W}_i$ (internal) and $\mathbf{W}_e$ (external), are separated by the cylindrical/spherical interface of radius $R_I = 0.5$ and given by

$$\mathbf{W}(\mathbf{x},0) = \begin{cases} \mathbf{W}_i = (1,0,0,0,1), & \text{if } \; r \leq R_I, \\ \mathbf{W}_e = (1/0.125,0,0,0,0.1), & \text{if } \; r > R_I \end{cases} \tag{45}$$

with r the radius of point $\mathbf{x}$. The inner state $\mathbf{W}_i$ is characterized by high density and high pressure, which generate a diverging shock and a contact discontinuity towards the external part of the domain, filled with a low density and low pressure gas. A rarefaction wave is moving to the opposite direction. The reference solution can be computed according to [58] by solving a one-dimensional system with geometric source terms and it is used to compare the numerical results obtained with the Lagrangian ADER-MOOD finite volume schemes. Zero velocity boundary condition is set on the external surface, while the final time of the simulation is $t_{\text{final}} = 0.25$ and the ratio of specific heats is assumed to be $\gamma = 1.4$. Figures 6 and 7 present the results in two and three space dimensions, respectively. A very good agreement with the reference solution is achieved and one can appreciate the gain in accuracy of the second order version of the scheme, which provides significantly more accurate results in the rarefaction wave.

In Figure 8 we plot the percentage of bad cells detected for each time step in 2D (left panel) and 3D (right panel, zoom on the first 10000 time steps). We observe that in 2D the percentage of bad cells is of the order 3 to 5%, while in 3D it starts at about 35% and constantly decreases to 10% after 14 time steps, then below 5% (after 3251 time steps), and asymptotically converges to 0.67% at the final time. The lower percentage of bad cells in the 3D case is probably due to the use of coarse cells enduring more dissipation, hence demanding less action of from our limiting procedure.

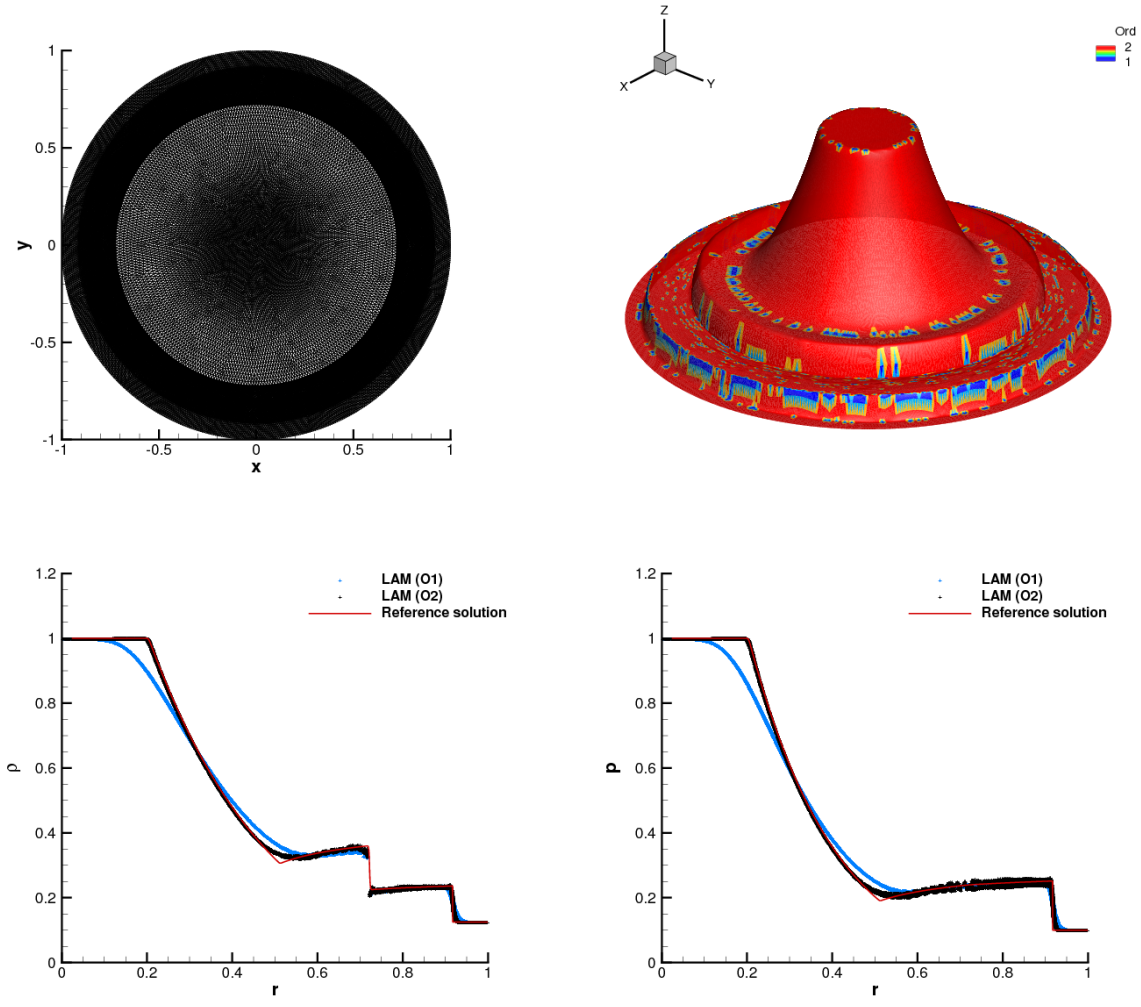

Fig. 6. Numerical results for the circular explosion problem in 2D — Top row: mesh configuration (left) and density distribution with cell order map (right) at the final time $t_{\text{final}} = 0.25$ — Bottom row: scatter plot of cell density (left) and pressure (right) compared against the exact solution.

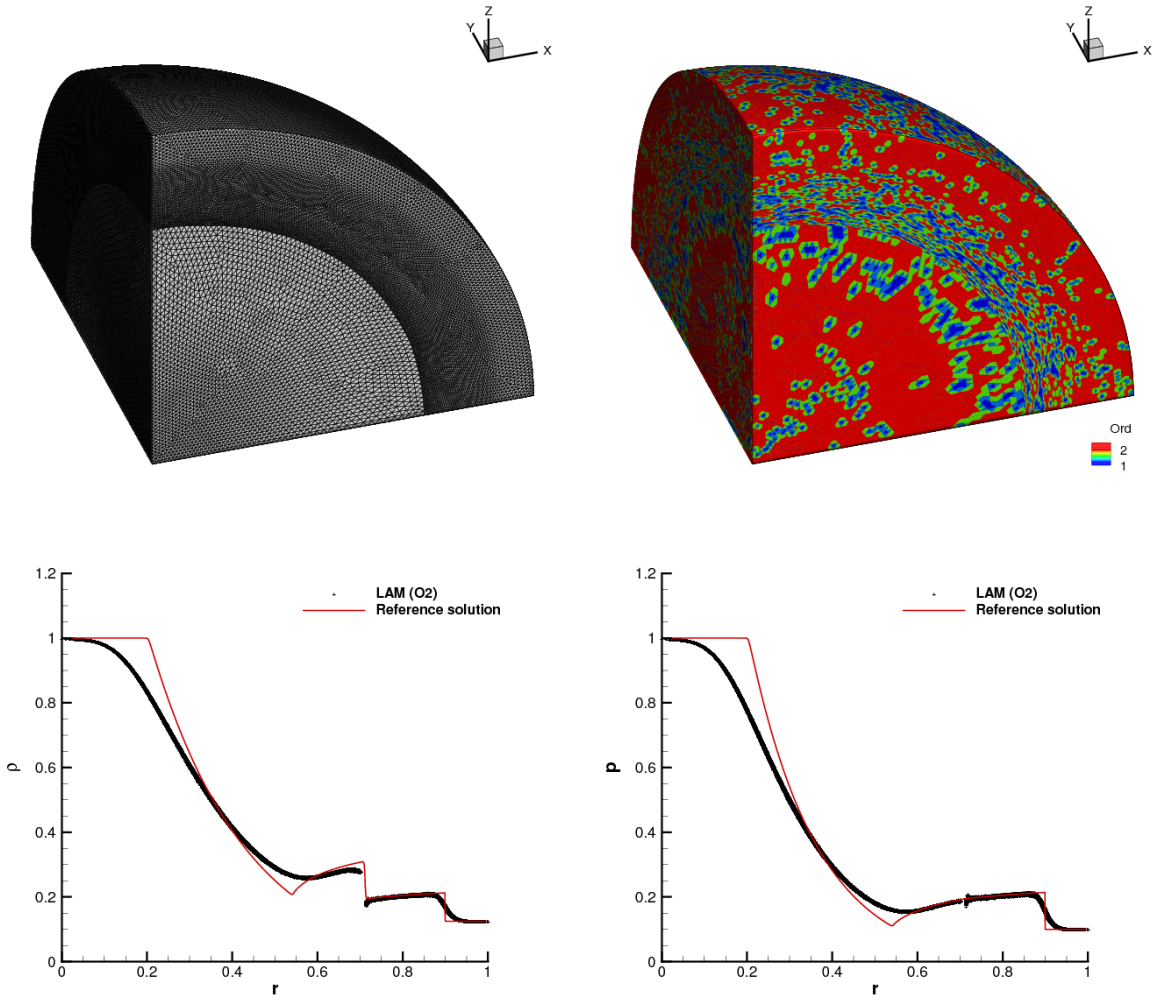

Fig. 7. Numerical results for the Sod problem in 3D — Top row: mesh configuration (left) and cell order map (right) at the final time $t_{\text{final}} = 0.25$ — Bottom row: scatter plot of cell density (left) and pressure (right) compared against the exact solution.

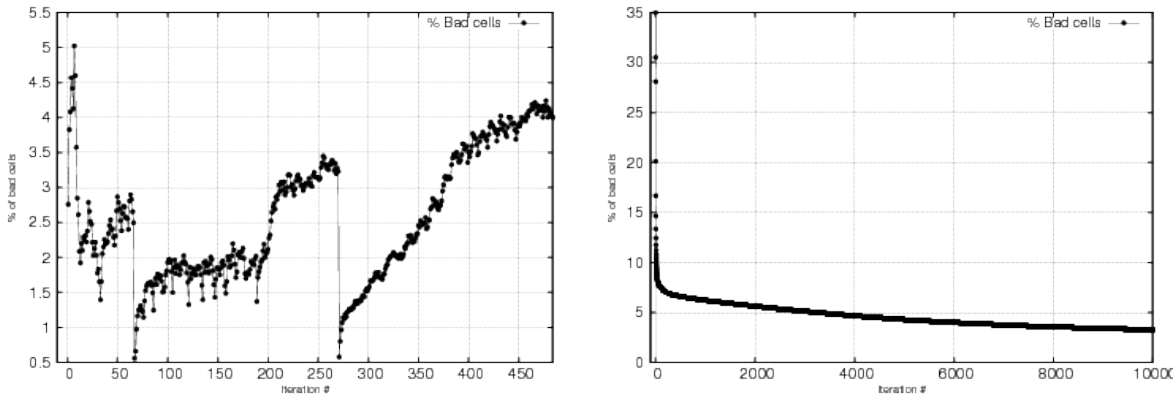

Fig. 8. Numerical results for the Sod problem in 2D (left) and 3D (right) — Percent of bad cells detected at each time iteration.

3.4 Piston problem

We consider now the motion of a piston which is pushing a perfect gas at rest contained in a box, hence generating a strong shock wave that highly compresses the cells of the computational mesh. This is a challenging test case first proposed by Saltzmann [25,13] on skewed quadrangular mesh to verify the robustness of Lagrangian schemes when the mesh is not aligned with the fluid flow. The initial computational domain, given by $\Omega(t=0) = [0;1]\times[0;0.1]$ in 2D and $\Omega(t=0) = [0;1]\times[0;0.1]\times[0;0.1]$ in 3D, is discretized with a characteristic mesh size of $h = 1/200$ with $N_E = 8824$ triangles, while $N_E = 74793$ tetrahedra with $h = 1/100$ in 3D. Here, we do not transform the grid configuration according to the skewness detailed in [25,13] because it leads to tangled elements when applied to fully unstructured meshes composed by simplex cells. Nevertheless the computational mesh does not exhibit any face aligned with the fluid flow, as can be observed in Figure 9. The gas is initially assigned the condition $\mathbf{W}(\mathbf{x},0) = (1,0,0,0,10^{-6})$ with $\gamma = 5/3$ and

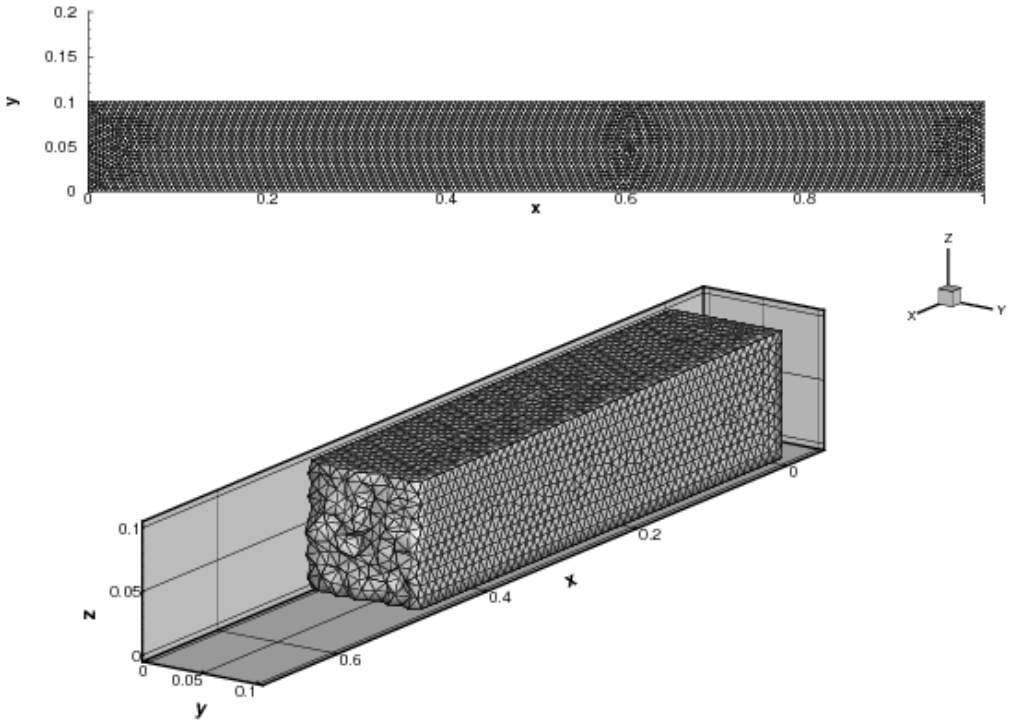

Fig. 9. Initial mesh configuration for the piston problem composed by fully unstructured triangles (top) and tetrahedra (bottom) — No element face is aligned along the $x-$axis which is the main flow direction.

the piston lies on the left side of the domain and moves with velocity $\mathbf{v}_p = (1,0,0)$. Velocity boundary condition is imposed on the piston, as well as on the remaining sides which are given slip wall boundary conditions. The final time of the simulation is $t_{\text{final}} = 0.75$. Note that the CFL number is set to CFL $= 0.01$ up to time $t = 0.01$ in order to prevent the generation of invalid elements in those cells lying near the piston that are instantaneously highly compressed for $t > 0$. Later on CFL $= 0.25$. One might also implement a *dynamic time step* strategy, in which the CFL number is adjusted automatically in order to preserve unfolded control volumes. The numerical results are depicted in Figures 10 and 11 at time $t = 0.6$, showing a good approximation of the shock plateau and the shock

wave location compared to the analytical solution [6,58]. The decrease of density near the piston is due to the well known *wall-heating problem*, see [57]. At time $t = 0.75$ the shock wave reaches the right boundary of the domain and is reflected backwards, hence traveling towards the piston. The impact between the piston and the shock occurs at time $t = 0.9$. Such time history is given in Figure 12 for $d = 2$, where we also plot the cell order map to confirm that the number of problematic elements is low and mainly located in the vicinity of the shock wave. As such these plots show that the limiter is able to detect the zones where the formal accuracy of the scheme must be reduced, hence increasing its numerical dissipation, and selecting the maximal accuracy in a large portion of the domain, without lack of robustness.

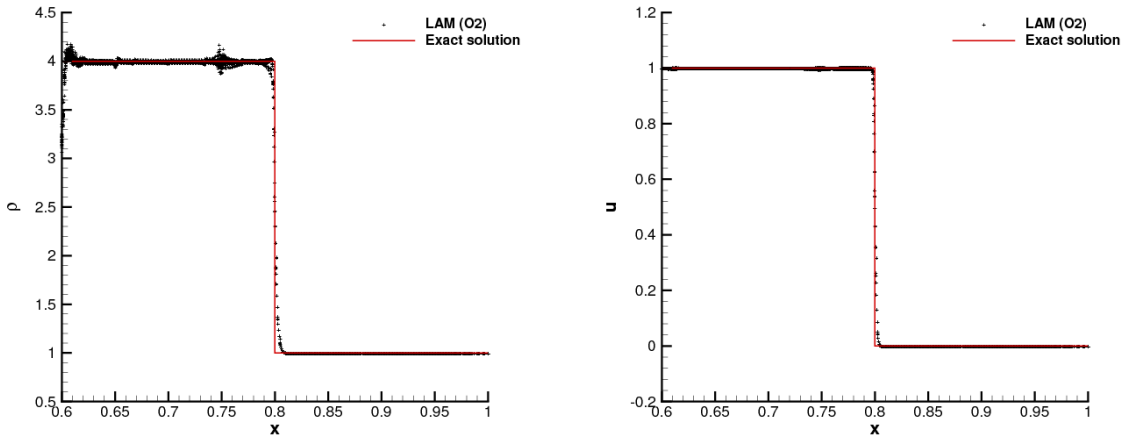

Fig. 10. Numerical results for the piston problem in 2D. Scatter plot of cell density (left) and horizontal velocity (right) compared against the exact solution at time $t = 0.6$.

In Figure 13 we plot the percentage of bad cells detected for each time iteration in 2D (left panel) and 3D (right panel, zoom on the first 1000 cycles). We observe that in 2D the percentage of bad cells is less than 1%, while in 3D, apart from the very first iterations, this number drops rapidly to almost zero percent.

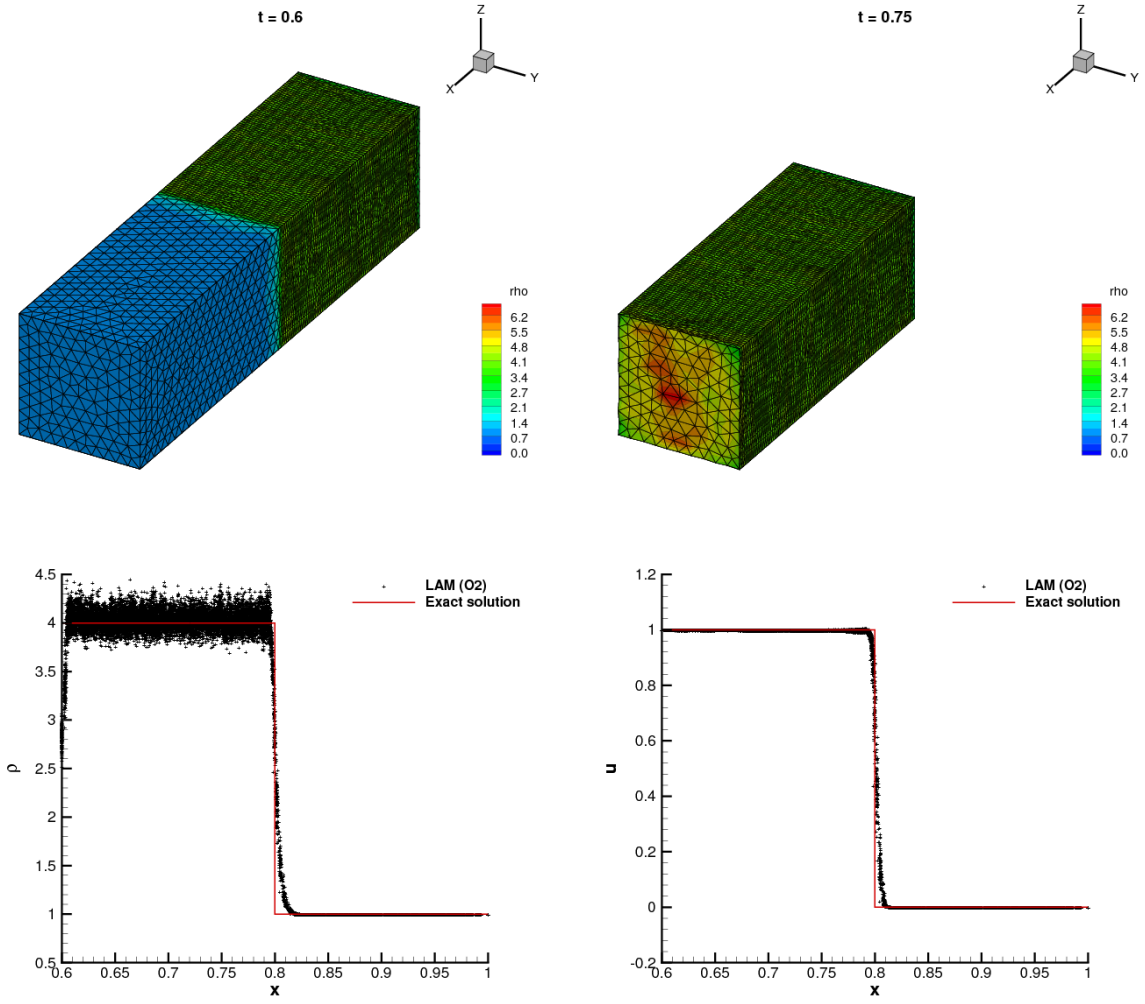

Fig. 11. Numerical results for the piston problem in 3D — Top row: density distribution and mesh configuration at output times $t = 0.6$ (left) and $t = 0.75$ (right) — Bottom row: scatter plot of cell density (left) and horizontal velocity (right) compared against the exact solution at time $t = 0.6$.

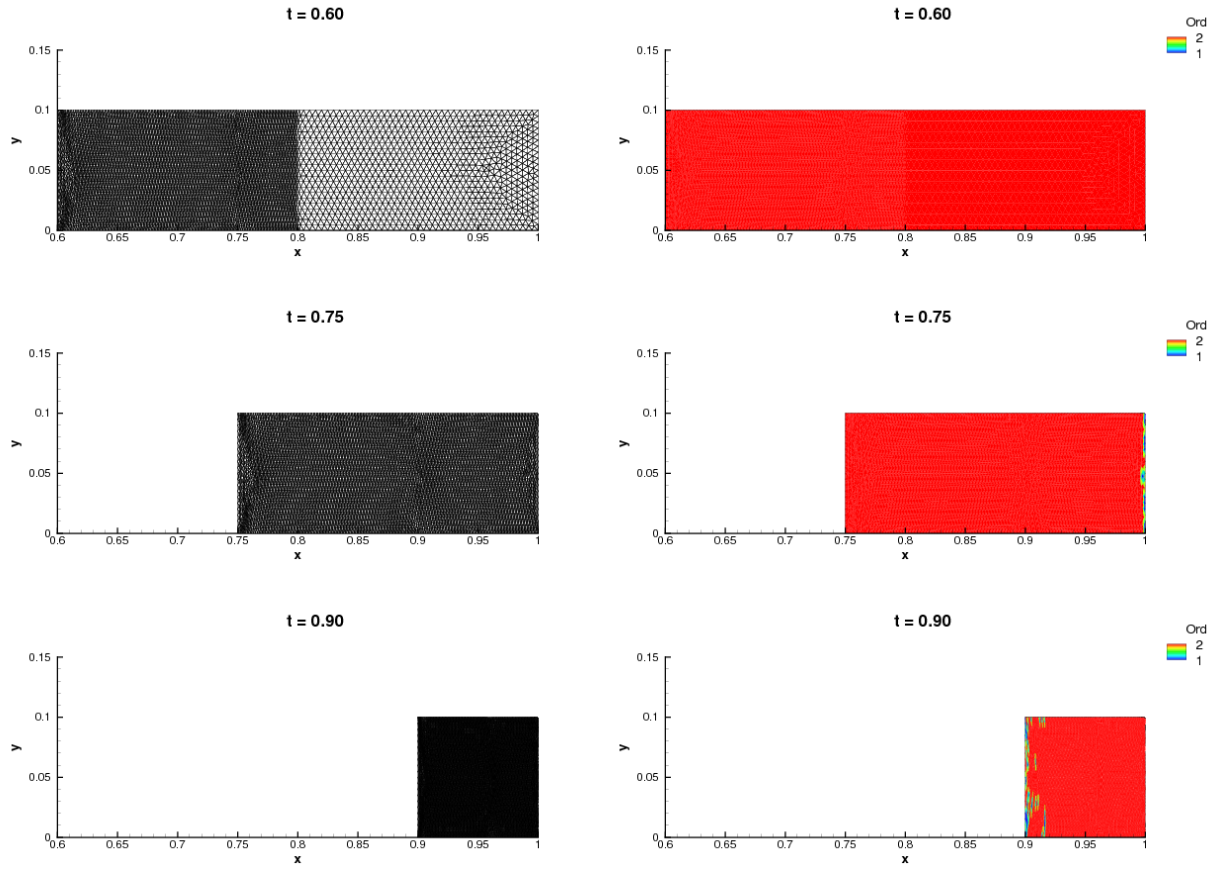

Fig. 12. Numerical results for the piston problem in 2D — Mesh configuration (left column) and cell order map (right column) at output times $t = 0.6$ (top row), $t = 0.75$ (middle row) and $t = 0.9$ (bottom row).

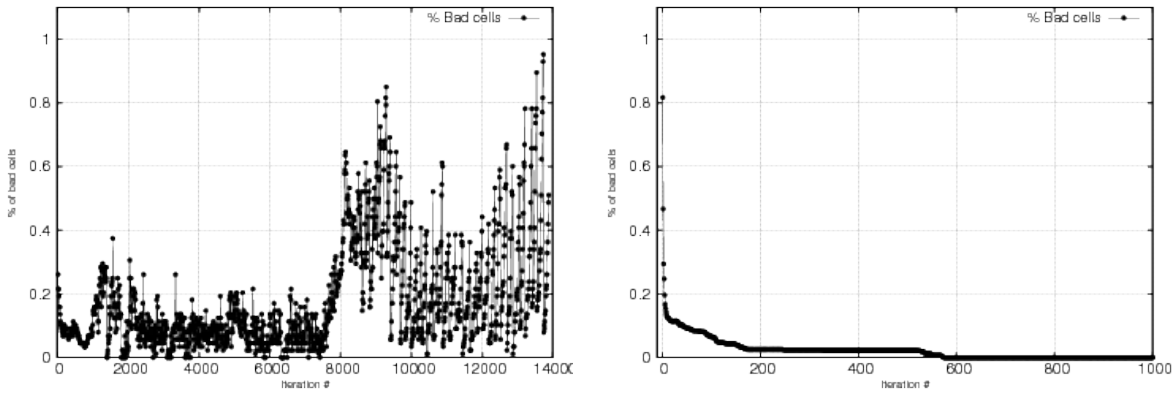

Fig. 13. Numerical results for the piston problem in 2D (left) and zoom in 3D on the first 1000 iterations (right) — Percent of bad cells detected at each time iteration.

3.5 Sedov problem

The Sedov problem describes the evolution of a blast wave with cylindrical or spherical symmetry. An analytical solution can be derived from self-similarity arguments [35], making this test a widely used one in literature [42]. The initial computational domain is the box $\Omega(0) = [0; 1.2]^d$ and we set symmetry boundary conditions on those faces which share the origin $\mathbf{O} = (x, y, z) = (0, 0, 0)$ of the domain, while the remaining sides are treated as slip walls. We use $N_E = 32400$ triangles and $N_E = 1552278$ tetrahedra to perform the simulation up to the final time $t_{\text{final}} = 1.0$. The initial condition reads $\mathbf{W}(\mathbf{x}, 0) = (1, 0, 0, 0, 10^{-6})$ in the entire computational domain, apart in those cells containing the origin $\mathbf{O}$ where the pressure is prescribed as

$$p_{or} = (\gamma - 1)\rho_0 \frac{E_{tot}}{\alpha\, V_{or}} \quad \text{with} \quad E_{tot} = \begin{cases} 0.244816 & \text{if } d = 2, \\ 0.851072 & \text{if } d = 3, \end{cases} \tag{46}$$

with $\gamma = 1.4$ and V_{or} denoting the volume of the elements attached to the origin. The factor α takes into account the cylindrical or spherical symmetry and is set to $\alpha = 4$ and $\alpha = 8$, respectively. The final mesh configuration is shown in Figures 14-15 in 2D and 3D respectively. An excellent agreement with the exact shock position at $r = 1$ is achieved. Furthermore, in Figure 14 one displays the colored density variable (bottom-left), and the problematic cells (bottom-right). We observe that bad cells are mainly located on the shock frontier, as expected, most of the cells being then updated with the second-order unlimited scheme. In 2D some cells are highly compressed at 45 degrees, which activates the limiter. This is consistent with observations for other Lagrangian schemes for which the elements in the diagonal are also problematic ones (see for instance [42]). Last, in Figure 16 we present the scatter plot of cell density as a function of cell radius for all cells in the computational domain against the exact solution in 2D (left panel) and 3D (right panel). This scatter plot confirms the preservation of the cylindrical and spherical symmetry and the relative sharpness of the numerical shock wave.

In Figure 17 we plot the percentage of bad cells detected for each time iteration in 2D (left panel) and 3D (right panel, zoom on the first 1000 cycles). The percentage of bad cells is less than 1%, while in 3D almost no bad cell is detected, which again proves that in multiple space dimensions the amount of work demanded by bad cells rapidly drops to zero compared to the cost of the unlimited scheme.

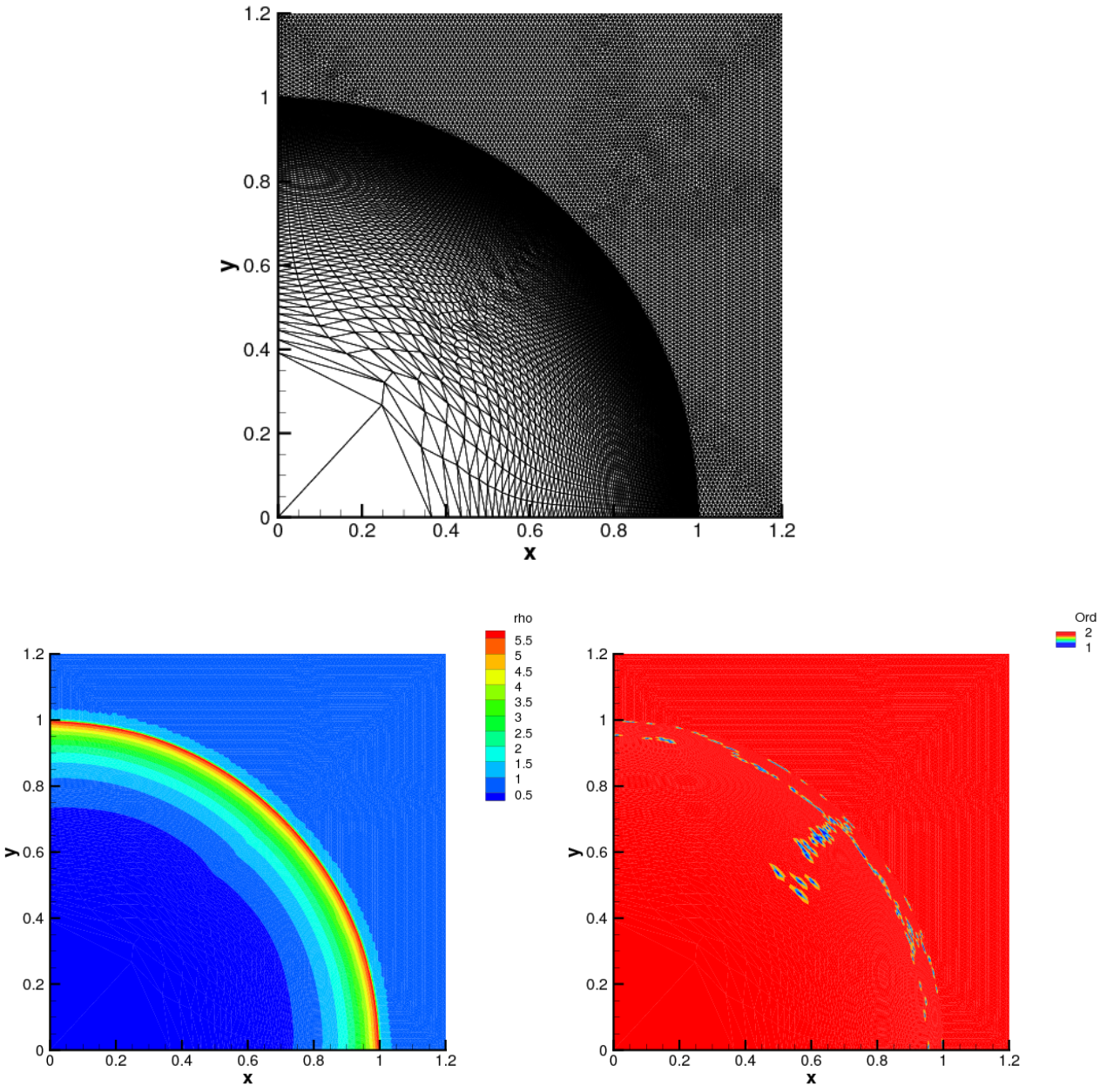

Fig. 14. Numerical results for the Sedov problem in 2D at the final time $t = 1.0$. Top: mesh configuration. Bottom: density distribution (left) and cell order map (right).

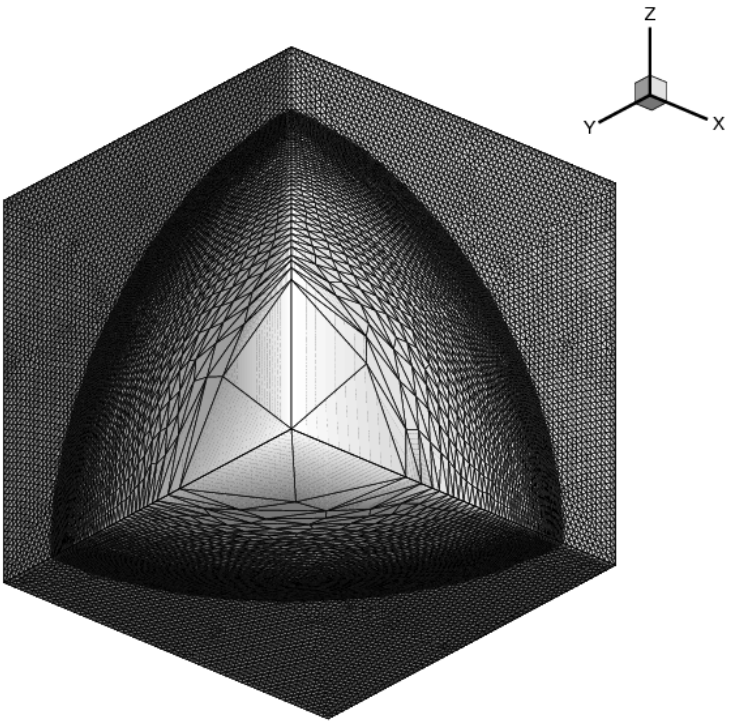

Fig. 15. Numerical results for the Sedov problem in 3D at the final time $t = 1.0$ — Mesh configuration.

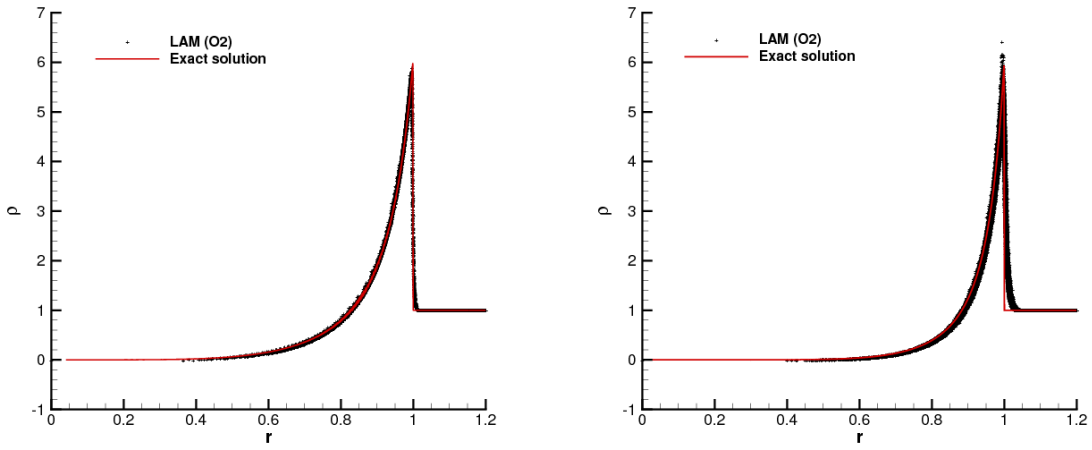

Fig. 16. Numerical results for the Sedov problem at the final time $t = 1.0$. Scatter plot of cell density compared against the exact solution in 2D (left) and 3D (right).

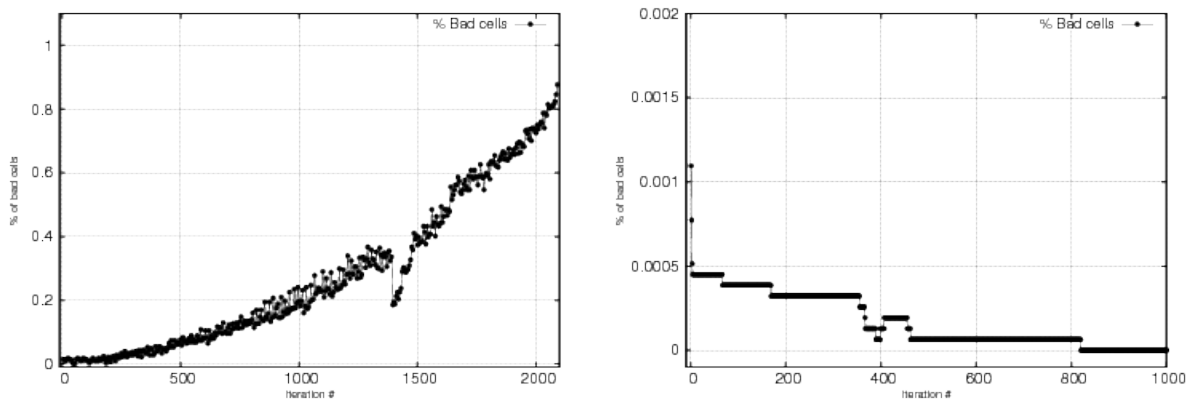

Fig. 17. Numerical results for the Sedov problem in 2D (left) and zoom in 3D on the first 1000 iterations (right) — Percent of bad cells detected at each time iteration.

3.6 Linear phase of the Richtmyer-Meshkov instability

Here, we simulate the Richtmyer-Meshkov instability for a piston-driven flow, following the work done in [42]. The initial computational domain is given by $\Omega(0) = [-5; 4.2] \times [0; 0.5]$ and is discretized with a characteristic mesh size of $h = 1/100$ yielding $N_E = 103334$ triangles. For the three-dimensional case we use $N_E = 2026881$ tetrahedra ($h = 1/50$) to pave the box $\Omega(0) = [-5; 4.2] \times [0; 0.5] \times [0; 0.5]$. The piston coincides with the left boundary of the domain and it moves with constant velocity $\mathbf{v}_p = (0.603, 0, 0)$, while the interface located at $x = 0$ splits the initial condition into a left state $\mathbf{W}_L(\mathbf{x}, 0) = (1, 0, 0, 0, 1)$ with $\gamma_L = 1.5$ and a right state $\mathbf{W}_R(\mathbf{x}, 0) = (1/2, 0, 0, 0, 3)$ with $\gamma_R = 3.0$. Symmetric boundary conditions are imposed on the planes $y = 0$ and $z = 0$, while the other sides are given a slip wall condition, apart from the piston. The final time of the simulation is $t_{\text{final}} = 6.0$ and at time $t_s = 3.015$ the piston hits the interface, hence generating transmitted and reflected shock waves that further interact with both the piston and the right boundary wall, as clearly highlighted by the $t - x$ diagram in Figure 18. The scatter plot of cell density is compared against the analytical solution at time $t = 5$ in two and three space dimensions, see Figures 18-19, where one can note that the transmitted and the reflected shock waves are well reproduced by the Lagrangian ADER-MOOD scheme.

Now, according to [42], we perform the same simulation but assigning an initially *perturbed* interface profile. The interface, originally located at $x = 0$, is moved according to a small perturbation $\alpha_0 = 10^{-4}$ with

$$x(y, z) = \alpha_0 \cos\left(\frac{2\pi}{\lambda} y\right), \tag{47}$$

where $\lambda = 1$ is the wavelength of the perturbation. If the perturbation is small enough, the linear theory [62] predicts that the growth of the perturbation amplitude $\alpha(t)$ is a linear function of time, after the shock-interface interaction. Therefore, the amplitude $\alpha(t)$ is measured as

$$\alpha(t) = \frac{x_p(t) - x_u(t)}{\alpha_0}, \tag{48}$$

where $x_p(t)$ and $x_u(t)$ represent the time history of the $x-$coordinate of the point initially located at $\mathbf{x} = (0, 0, 0)$ for the perturbed and unperturbed simulation, respectively. The study of the linear behavior is performed for $d = 2$ and the results are plotted in Figure 20, showing that the second order version of the Lagrangian ADER-MOOD scheme approximates the linear profile correctly. Contrarily the first order scheme is too diffusive and damps the linear growth of the instability, changing *de facto* the underlying physics at this resolution.

In Figure 21 we plot the percentage of bad cells detected for each time iteration in 2D (left panel) and 3D (right panel, zoom on the first 1000 cycles). The percentage of bad cells in 2D is of the order of 0.01% (~ 10 bad cells) and few "spikes" at about 0.17% (~ 176 bad cells) and 0.42% (~ 430 bad cells). In 3D we observe very few bad cells (maximal 17 bad cells, and, after iteration 1046, no more bad cell is ever detected) but this seems enough to maintain non-oscillatory results. The very small number of bad cells is the main reason why the recomputation step in the MOOD approach is computationally very cheap.

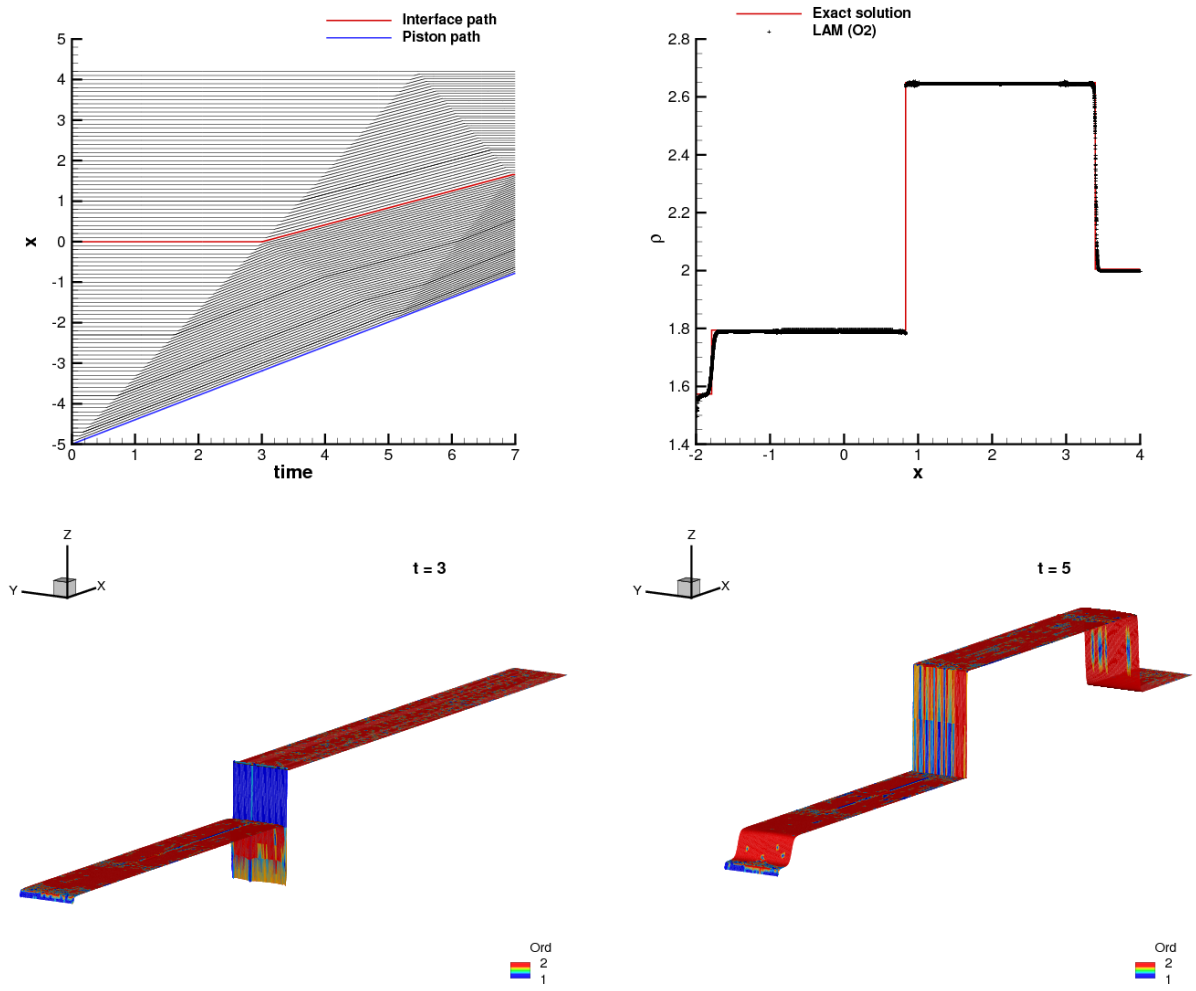

Fig. 18. Numerical results for the Richtmyer-Meshkov instability test case in 2D. Top row: $t-x$ diagram showing the time history of the shock-contact interaction (left) and scatter plot of cell density at time $t = 5$ compared against the exact solution (right). Bottom row: density profile (elevation) and cell order map (colors) at output times $t = 3$ (right) and $t = 5$ (left).

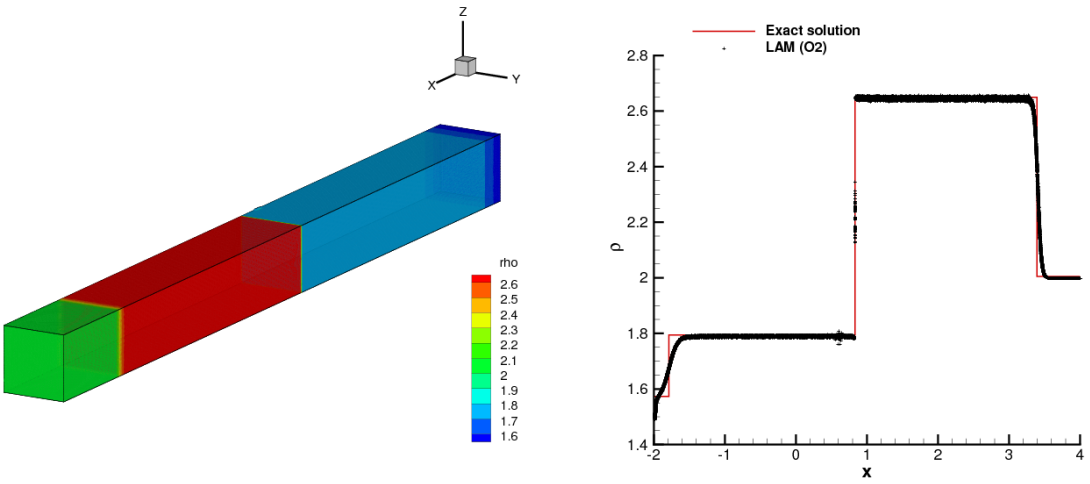

Fig. 19. Numerical results for the Richtmyer-Meshkov instability test case in 3D at time $t = 5$. Left: domain configuration and density contours. Right: scatter plot of cell density compared against the exact solution.

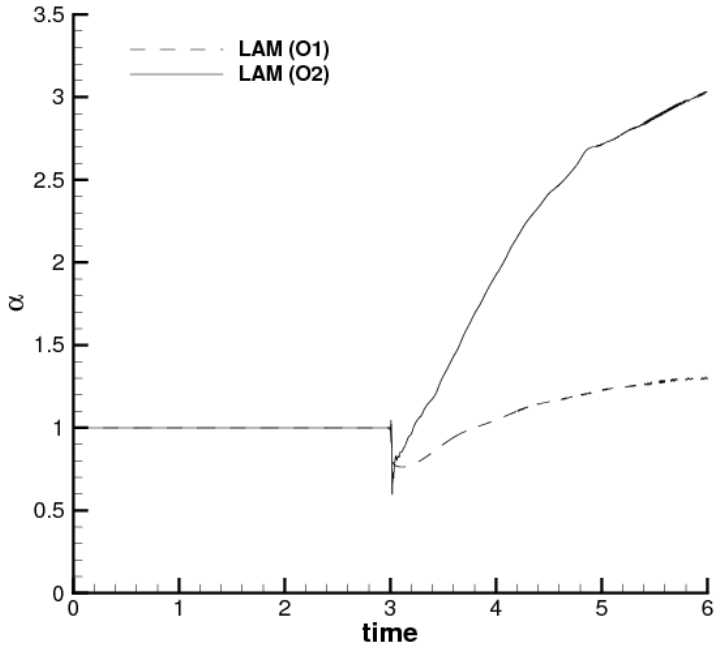

Fig. 20. Numerical results for the Richtmyer-Meshkov instability test case in 2D. Numerical perturbation amplitude as function of time for first and second order accurate Lagrangian ADER–MOOD scheme.

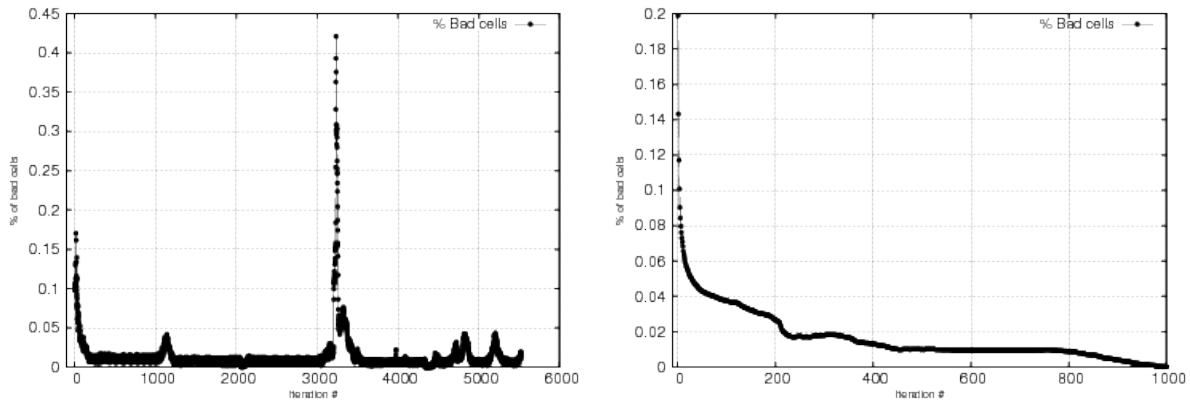

Fig. 21. Numerical results for the Richtmyer-Meshkov instability test case in 2D (left) and in 3D on the first 1000 time iterations (right) — Percent of bad cells detected at each time iteration.

3.7 Comparison with Eucclhyd scheme

In this section we show a comparison between the results obtained with the method presented in this paper and the ones computed with the *Eucclhyd* scheme outlined in [42]. We strictly follow the algorithm detailed in [42], applying it to unstructured meshes made by triangles and tetrahedra in order to achieve a fair comparison.

First, we consider the well-known Sod shock tube problem in the two-dimensional computational domain $\Omega(0) = [-0.5; 0.5] \times [-0.05; 0.05]$, which is discretized with a characteristic mesh size of $h = 1/100$. The initial condition is given by (45) with the initial discontinuity located at $x = 0$, while the final time is set to $t_{\text{final}} = 0.2$. Even though the Sod shock tube problem presents a one-dimensional setup, it becomes fully multidimensional when applied to unstructured meshes. We run this test case with the following schemes:

- FO: first order scheme ($M = 0$);
- LAM (Lagrangian ADER MOOD): the method presented in this paper with *a posteriori* limiting;
- ADER BJ / VK: here we use the ADER approach presented in Section 2.3.3 with an *a priori* limiting strategy, namely either Barth and Jespersen (BJ) limiter [2] or Venkatakrishnan (VK) limiter [59];
- *Eucclhyd* BJ / VK: the *Eucclhyd* method with either BJ or VK limiter.

The errors are evaluated according to (45) against the exact solution of the Riemann problem and they are reported for each scheme in Table 3. The error norms are of the same order of magnitude, and the minimum error is given by *Eucclhyd* BJ, followed by MOOD BJ. The Venkatakrishnan (VK) limiter seems to be too aggressive, hence generating the biggest errors. Figure 22 shows the scatter plot of density distribution at the final time of the simulation obtained with *Eucclhyd* BJ and the ADER MOOD approach discussed in this paper.

	Density		**Velocity**		**Pressure**		**CPU time**
	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_1}	ϵ_{L_2}	ϵ_{L_1}	ϵ_{L_2}	
FO	5.2474E-03	7.1137E-04	1.3864E-03	1.2756E-02	7.9593E-04	6.4253E-03	2.1804E+01
LAM	1.6235E-03	1.7213E-04	3.4280E-04	6.8285E-03	1.9043E-04	1.8568E-03	4.8092E+01
ADER BJ	1.5054E-03	1.1964E-04	3.1054E-04	7.6444E-03	1.4807E-04	1.6778E-03	4.4480E+01
ADER VK	1.6838E-03	1.7097E-04	3.8416E-04	7.5928E-03	1.9254E-04	1.9632E-03	4.0648E+01
***Eucclhyd* BJ**	1.5046E-03	1.1923E-04	3.0903E-04	7.6499E-03	1.4743E-04	1.6748E-03	3.0570E+01
***Eucclhyd* VK**	1.6806E-03	1.7060E-04	3.8322E-04	7.5908E-03	1.9214E-04	1.9607E-03	2.9951E+01

Table 3
Error norms (ϵ_{L_1},ϵ_{L_2}) for the the two-dimensional Sod shock tube problem computed with six different numerical schemes: FO, ADER MOOD, ADER BJ, ADER VK, *Eucclhyd* BJ, *Eucclhyd* VK. Computational time is also reported in the last column.

Next, we perform a convergence analysis using again the Kidder problem described in Section 3.1. In Table 4 we compare the error norms for density obtained with LAM and *Eucclhyd* schemes as well as the computational time needed for running each simulation. Due to the smoothness of this test case, which does not include any discontinuity, we run the *Eucclhyd* scheme without any limiter, while the unlimited second order scheme is automatically recovered in the LAM method because no bad cells are detected, hence not activating the *a posteriori* limiter. Looking at the results in Table 4, the lowest errors

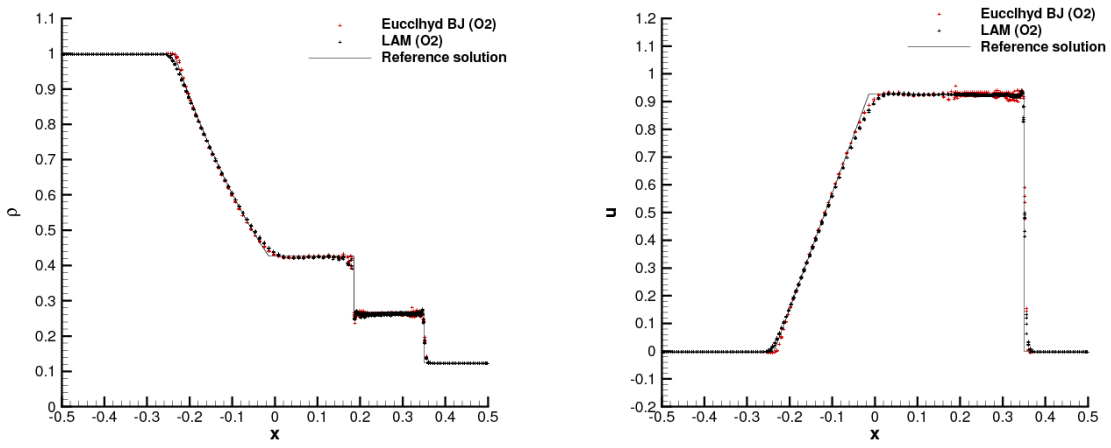

Fig. 22. Scatter plot of density (left) and horizontal velocity (right) distribution obtained for the two-dimensional Sod shock tube problem with LAM (black) scheme and *Eucclhyd* BJ scheme.

are achieved by the LAM scheme, which for most grid sizes presents a smaller value w.r.t. the results computed by the unlimited *Eucclhyd* method.

Eucclhyd

$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	CPU time
1.24E-02	2.2066E-02	-	6.5584E-02	-	3.6786E-01	-	2.2093E+01
7.66E-03	1.7788E-02	0.4	5.0944E-02	0.5	3.1132E-01	0.3	3.5668E+01
5.73E-03	9.0194E-03	2.3	2.6558E-02	2.2	1.9516E-01	1.6	7.7093E+01
4.71E-03	6.5608E-03	1.6	1.8843E-02	1.8	1.7286E-01	0.6	1.8940E+02

LAM

$h(\Omega(t_{\text{final}}))$	ϵ_{L_1}	$\mathcal{O}(L_1)$	ϵ_{L_2}	$\mathcal{O}(L_2)$	ϵ_{L_∞}	$\mathcal{O}(L_\infty)$	CPU time
1.47E-02	2.6834E-02	-	7.3494E-02	-	4.9616E-01	-	3.4573E+01
8.08E-03	1.7682E-02	0.7	4.7413E-02	0.7	2.2211E-01	1.3	6.0369E+01
6.13E-03	9.7763E-03	2.2	2.5907E-02	2.2	1.2667E-01	2.0	9.8604E+01
4.56E-03	5.6752E-03	1.8	1.5026E-02	1.8	6.5501E-02	2.2	3.5568E+02

Table 4
Numerical convergence studies for the Kidder problem in 2D carried out with *Eucclhyd* scheme and the LAM algorithm described in this paper. The error norms refer to density and are computed at the final time of the simulation.

For the sake of completeness one should also note that the *Eucclhyd* scheme is always computationally more efficient than the proposed LAM algorithm, as evident from Tables 3 and 4. Figure 23 depicts the dependency of the error norm on both the mesh size and the CPU time: the lower errors obtained with the LAM scheme require almost twice the computational time of the *Eucclhyd* scheme.

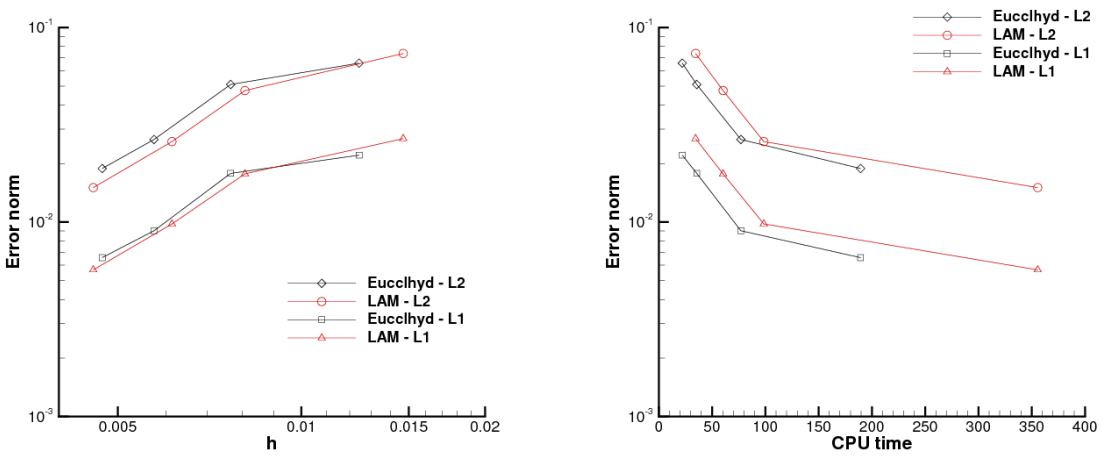

Fig. 23. Comparison between unlimited *Eucclhyd* scheme and the LAM scheme for the Kidder problem in 2D. Left: dependency of the error norm on the mesh size. Right: dependency of the error norm on the CPU time.

4 Conclusions and perspectives

This paper has presented a second-order accurate cell-centered Lagrangian scheme for the hydrodynamics system of conservation laws. The method is based on a conservative Lagrangian formulation in mass, momentum and total energy. A so-called nodal solver allows the determination of a vertex velocity which builds a consistent discretization between the trajectory equation and the geometrical conservation law. Second-order of accuracy in space and time are achieved via an ADER procedure, which differs from previous second order cell centered Lagrangian schemes. The ADER technology generates a predictor solution that can further be used inside the classical subcell force based Lagrangian scheme to reach a formal second-order of accuracy in space and time.
Robustness and stability are gained by the use of an *a posteriori* MOOD limiting strategy. A second-order unlimited candidate solution at t^{n+1} is tested against appropriate detection criteria to determine invalid cells. The solution in those cells is discarded and recomputed starting again at t^n but using the first-order accurate scheme.
This numerical scheme has been developed and tested on unstructured simplicial meshes in 2D and 3D on a large panel of test cases:

- the Kidder problem to assess the formal accuracy even when the limiter is on,
- a multidimensional Sod-type problem to show the ability of the method to capture non-aligned simple cylindrical/spherical waves and the difference between first and second order accurate schemes,
- a piston problem to show the ability of the method to maintain the prediction capability even when a shock travels across the whole domain several times,
- the Sedov problem to show the preservation of cylindrical/spherical symmetric waves with low dispersion/diffusion,
- the linear phase of a Richtmyer-Meshkov instability to verify the ability of the scheme to maintain a sufficient accuracy even in the presence of shocks when the limiter is on.

This test suite has been systematically simulated in 2D and 3D, showing that this cell-centered Lagrangian scheme is robust, essentially non-oscillatory and, at the same time maintains an almost optimal precision by a careful utilization of the high order scheme where appropriate and the low order one in the vicinity of problematic zones. Finally, a study is proposed in order to compare the popular *Eucclhyd* scheme with the new LAM scheme presented in this paper. Both smooth and discontinuous test cases are considered in order to fairly compare the two numerical schemes.
So far this numerical method has been only tested on the PDEs of compressible gas dynamics and in the future we plan to extend it also to more complex models for nonlinear elasto-plastic solids, e.g. the hypo-elastic Wilkins-type model [61,46], as well as the hyper-elastic model of Godunov and Romenski [33,53,9].

Moreover we plan to couple this Lagrangian scheme with a rezone and remap module to construct an indirect Arbitrary Lagrangian-Eulerian scheme. We also note that the extension of the proposed method to quadrangular/hexahedral or general polygonal/polyhedral meshes does not present any particular difficulty. Last but not least, the combination of the ADER approach with *a posteriori* MOOD limiting may become particularly attractive for the development of more general kth-order accurate purely Lagrangian schemes on moving curvilinear grids.

Acknowledgments

The material of this research has been partly built during the SHARK FV workshops which took place in May 2015, 2016 and 2017 in Ofir, Portugal
(`http://www.math.univ-toulouse.fr/SHARK-FV/`).
M.D. and W.B. have been financed by the European Research Council (ERC) under the European Union's Seventh Framework Programme (FP7/2007-2013) with the research project *STiMulUs*, ERC Grant agreement no. 278267. The authors would like to acknowledge PRACE for awarding access to the SuperMUC supercomputer based in Munich, Germany at the Leibniz Rechenzentrum (LRZ) and the HLRS supercomputing center in Stuttgart, Germany, for awarding access to the HazelHen supercomputer.

The authors would also like to thank the referees for their constructive comments and suggestions which definitely helped to improve the quality and the clarity of the manuscript.

A Boundary conditions

In our cell-centered Lagrangian scheme, boundary conditions are imposed *directly* in the nodal solver (35). Let $\mathcal{I}^{BC}(r)$ represent the set of boundary edges (or faces in 3D) surrounding node r. There are three type of boundary conditions that can be set:

- *pressure boundary condition*: a prescribed pressure p_f^{BC} on a boundary face f is taken into account as an additional term on the right hand side of (35), hence

$$\mathbf{u}_r^* = \mathrm{M}_r^{-1} \left(\sum_{i \in \mathcal{I}(r)} p_i^*(\mathbf{X}_r)\, L_{r,i}^n \mathbf{n}_{r,i}^n + \mathrm{M}_{r,i}^n \mathbf{u}_i - \sum_{f \in \mathcal{I}^{BC}(r)} p_f^{BC}\, L_{r,f}^n \mathbf{n}_{r,f}^n \right); \tag{A.1}$$

- *velocity boundary condition*: the prescribed velocity $v_f^{BC} := \mathbf{v}_f^{BC} \cdot \mathbf{n}_f^n$ can be interpreted again as a pressure boundary condition. The equivalent pressure force $p_{\mathbf{v}}^{BC}$ is given by

$$\begin{aligned} p_{\mathbf{v}}^{BC} &= \frac{\mathrm{M}_r^{-1} \left(\sum_{i \in \mathcal{I}(r)} p_i^*(\mathbf{X}_r)\, L_{r,i}^n \mathbf{n}_{r,i}^n + \mathrm{M}_{r,i}^n \mathbf{u}_i \right) \cdot \mathbf{d}_r - \sum_{f \in \mathcal{I}^{BC}(r)} v_f^{BC}\, L_{r,f}^n}{\mathrm{M}_r^{-1} \mathbf{d}_r \cdot \mathbf{d}_r}, \\ \mathbf{d}_r &= \sum_{f \in \mathcal{I}^{BC}(r)} L_{r,f}^n \mathbf{n}_{r,f}^n. \end{aligned} \tag{A.2}$$

  Then, the node velocity is evaluated by considering

$$\mathbf{u}_r^* = \mathrm{M}_r^{-1} \left(\sum_{i \in \mathcal{I}(r)} p_i^*(\mathbf{X}_r)\, L_{r,i}^n \mathbf{n}_{r,i}^n + \mathrm{M}_{r,i}^n \mathbf{u}_i \right) - p_{\mathbf{v}}^{BC} \mathbf{d}_r; \tag{A.3}$$

- *symmetry boundary conditions*: such requirements are concerned with geometric considerations. Specifically, the symmetry might involve either a symmetry plane defined by an orthonormal basis $(\boldsymbol{\tau}_1, \boldsymbol{\tau}_2)$, or a symmetry line along a direction vector $\boldsymbol{\tau}_1$, or even a symmetry point where we simply set $\mathbf{u}_r^* = \mathbf{0}$.

  If a symmetry plane is considered, then the node velocity writes $\mathbf{u}_r^* = \alpha_1 \boldsymbol{\tau}_1 + \alpha_2 \boldsymbol{\tau}_2$

and the momentum balance equation (35) becomes

$$\mathrm{M}_r(\alpha_1\boldsymbol{\tau}_1+\alpha_2\boldsymbol{\tau}_2)=\left(\sum_{i\in\mathcal{I}(r)} p_i^*(\mathbf{X}_r)\,L_{r,i}^n\mathbf{n}_{r,i}^n+\mathrm{M}_{r,i}^n\mathbf{u}_i\right), \tag{A.4}$$

which is solved by successive projection on $\boldsymbol{\tau}_1$ and $\boldsymbol{\tau}_2$.

On a symmetry line one has $\mathbf{u}_r^*=\alpha_1\boldsymbol{\tau}_1$ and the node velocity is simply obtained by

$$\alpha_1\mathrm{M}_r\cdot\boldsymbol{\tau}_1=\left(\sum_{i\in\mathcal{I}(r)} p_i^*(\mathbf{X}_r)\,L_{r,i}^n\mathbf{n}_{r,i}^n+\mathrm{M}_{r,i}^n\mathbf{u}_i\right). \tag{A.5}$$

Here, we have only recalled what is needed for a consistent implementation of boundary conditions with the nodal solver presented in Section 2.4. For further details and comments we refer the reader to [47,32].

References

[1] A.J. Barlow, P.-H. Maire, W.J. Rider, R.N. Rieben, and M.J. Shashkov. Arbitrary lagrangian- eulerian methods for modeling high-speed compressible multimaterial flows. *J. Comput. Phys.*, 322:603–665, 2016.

[2] T.J. Barth and D.C. Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper 89-0366*, pages 1–12, 1989.

[3] M. Ben-Artzi and J. Falcovitz. A second-order godunov-type scheme for compressible fluid dynamics. *Journal of Computational Physics*, 55:1–32, 1984.

[4] F. Blachère and R. Turpault. An admissibility and asymptotic preserving scheme for systems of conservation laws with source term on 2D unstructured meshes with high-order MOOD reconstruction. *Computer Methods in Applied Mechanics and Engineering*, 317:836 – 867, 2017.

[5] W. Boscheri. An efficient high order direct ALE ADER finite volume scheme with a posteriori limiting for hydrodynamics and magnetohydrodynamics. *International Journal for Numerical Methods in Fluids*, 134-135:111–129, 2016.

[6] W. Boscheri. High Order Direct Arbitrary-LagrangianEulerian (ALE) Finite Volume Schemes for Hyperbolic Systems on Unstructured Meshes. *Archives of Computational Methods in Engineering*, 24(4):751–801, 2017.

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

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

[9] W. Boscheri, M. Dumbser, and R. Loubère. Cell centered direct Arbitrary-Lagrangian-Eulerian ADER-WENO finite volume schemes for nonlinear hyperelasticity. *Computers and Fluids*, 134-135:111–129, 2016.

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

[11] Després Bruno. *Numerical Methods for Eulerian and Lagrangian Conservation Laws.* Frontiers in Mathematics. Springer International Publishing, 2017.

[12] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146(1):227–262, 1998.

[13] E.J. Caramana, C.L. Rousculp, and D.E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *Journal of Computational Physics*, 157:89 – 119, 2000.

[14] G. Carré, S. Del Pino, B. Després, and E. Labourasse. A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension. *Journal of Computational Physics*, 228:5160–5183, 2009.

[15] J. Cheng and C.W. Shu. A high order ENO conservative Lagrangian type scheme for the compressible Euler equations. *Journal of Computational Physics*, 227:1567–1596, 2007.

[16] J. Cheng and C.W. Shu. A third order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equation. *Communications in Computational Physics*, 4:1008–1024, 2008.

[17] J. Cheng and C.W. Shu. A cell-centered Lagrangian scheme with the preservation of symmetry and conservation properties for compressible fluid flows in two-dimensional cylindrical geometry. *Journal of Computational Physics*, 229:7191–7206, 2010.

[18] J. Cheng and C.W. Shu. Improvement on spherical symmetry in two-dimensional cylindrical coordinates for a class of control volume Lagrangian schemes. *Communications in Computational Physics*, 11:1144–1168, 2012.

[19] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation lawsmulti-dimensional optimal order detection (MOOD). *Journal of Computational Physics*, 230(10):4028 – 4050, 2011.

[20] S. Clain and J. Figueiredo. The MOOD method for the non-conservative shallow-water system. *Computers and Fluids*, 145:99 – 128, 2017.

[21] B. Després and C. Mazeran. Lagrangian gas dynamics in two dimensions and Lagrangian systems. *Arch. Rational Mech. Anal.*, 178:327–372, 2005.

[22] S. Diot, M.M. Francois, and E.D. Dendy. A higher-order unsplit 2D direct Eulerian finite volume method for two-material compressible flows based on the MOOD paradigms. *International Journal for Numerical Methods in Fluids*, 76(12):1064–1087, 2014.

[23] V. Dobrev, T. Kolev, and R. Rieben. High-order curvilinear finite element methods for lagrangian hydrodynamics. *SIAM Journal on Scientific Computing*, 34(5):B606–B641, 2012.

[24] V.A. Dobrev, T.E. Ellis, Tz.V. Kolev, and R.N. Rieben. Curvilinear Finite elements for Lagrangian hydrodynamics. *International Journal for Numerical Methods in Fluids*, 65:1295–1310, 2011.

[25] J.K. Dukovicz and B. Meltz. Vorticity errors in multidimensional Lagrangian codes. *Journal of Computational Physics*, 99:115 – 134, 1992.

[26] M. Dumbser, D.S. Balsara, E.F. Toro, and C.-D. Munz. A unified framework for the construction of one-step finite volume and discontinuous Galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209 – 8253, 2008.

[27] M. Dumbser, C. Enaux, and E.F. Toro. Finite volume schemes of very high order of accuracy for stiff hyperbolic balance laws. *Journal of Computational Physics*, 227:3971–4001, 2008.

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

[29] F.Vilar, P.-H. Maire, and R. Abgrall. A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids. *Journal of Computational Physics*, 276:188–234, 2014.

[30] Luttwak G. and Falcovitz J. Slope limiting for vectors: A novel vector limiting algorithm. *International Journal for Numerical Methods in Fluids*, 65(12):1365–1375, 2011.

[31] S. Galera, P.-H. Maire, and J. Breil. A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction. *Journal of Computational Physics*, 229:5755–5787, 2010.

[32] G. Georges, J. Breil, and P.-H. Maire. A 3D GCL compatible cell-centered Lagrangian scheme for solving gas dynamics equations. *Journal of Computational Physics*, 305:921–941, 2016.

[33] S.K. Godunov and E.I. Romenski. Nonstationary equations of the nonlinear theory of elasticity in Euler coordinates. *Journal of Applied Mechanics and Technical Physics*, 13:868–885, 1972.

[34] C. Hu and C.W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.

[35] J.R. Kamm and F.X. Timmes. On efficient generation of numerically robust sedov solutions. *Technical Report LA-UR-07-2849,LANL*, 2007.

[36] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486–508, 2005.

[37] R.E. Kidder. Laser-driven compression of hollow shells: power requirements and stability limitations. *Nucl. Fus.*, 1:3 – 14, 1976.

[38] W. Liu, J. Cheng, and C.W. Shu. High order conservative lagrangian schemes with lax-wendroff type time discretization for the compressible euler equations. *Journal of Computational Physics*, 228:8872–8891, 2009.

[39] R. Loubère, M. Dumbser, and S. Diot. A New Family of High Order Unstructured MOOD and ADER Finite Volume Schemes for Multidimensional Systems of Hyperbolic Conservation Laws. *Communication in Computational Physics*, 16:718–763, 2014.

[40] R. Loubère, P.-H. Maire, and P. Váchal. Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydro-dynamics scheme. *Communication in Computational Physics*, 10(4):940–978, 2011.

[41] G. Luttwak and J. Falcovitz. Slope limiting for vectors: A novel limiting algorithm. *Int. J. Numer. Meth. Fluid*, 65:1365–1375, 2011.

[42] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. *Journal of Computational Physics*, 228:2391–2425, 2009.

[43] P.-H. Maire. A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grids. *International Journal for Numerical Methods in Fluids*, 65:1281–1294, 2011.

[44] P.-H. Maire. *Contribution to the numerical modeling of Inertial Confinement Fusion.* PhD thesis, University of Bordeaux, France, 2011. Habilitation à diriger des recherches, available at `https://www.researchgate.net/publication/304111873_Habilitation_Defense_Contribution_to_the_Numerical_Modeling_of_Inertial_Confinement_Fusion`.

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

[46] P.-H. Maire, R. Abgrall, J. Breil, R. Loubère, and B. Rebourcet. A Nominally Second-order Cell-centered Lagrangian Scheme for Simulating Elastic-plastic Flows on Two-dimensional Unstructured Grids. *J. Comput. Phys.*, 235(C):626–665, February 2013.

[47] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM Journal on Scientific Computing*, 29:1781–1824, 2007.

[48] P.-H. Maire and J. Breil. A second-order cell-centered lagrangian scheme for two-dimensional compressible flow problems. *International Journal for Numerical Methods in Fluids*, 56:1417–1423, 2007.

[49] P.-H. Maire and B. Nkonga. Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics. *Journal of Computational Physics*, 228:799–821, 2009.

[50] R.C. Millington, E.F. Toro, and L.A.M. Nejad. *Arbitrary High Order Methods for Conservation Laws I: The One Dimensional Scalar Case.* PhD thesis, Manchester Metropolitan University, Department of Computing and Mathematics, June 1999.

[51] J. Von Neumann and R. D. Richtmyer. A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics*, 21:232–237, 1950.

[52] X. Nogueira, L. Ramirez, S. Clain, R. Loubère, L. Cueto-Felgueroso, and I. Colominas. High-accurate SPH method with Multidimensional Optimal Order Detection limiting. *Computer Methods in Applied Mechanics and Engineering*, 310:134 – 155, 2016.

[53] I. Peshkov and E. Romenski. A hyperbolic model for viscous newtonian flows. *Continuum Mechanics and Thermodynamics*, 28(1):85–104, 2016.

[54] W.J. Rider and L. G. Margolin. Simple modifications of monotonicity-preserving limiter. *Journal of Computational Physics*, 174(1):473 – 488, 2001.

[55] T. Schwartzkopff, C.D. Munz, and E.F. Toro. ADER: A High Order Approach For Linear Hyperbolic Systems in 2D. *Journal of Scientific Computing*, 17(1-4):231–240, 2002.

[56] V.A. Titarev and E.F. Toro. ADER: Arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17(1-4):609–618, December 2002.

[57] E.F. Toro. Anomalies of conservative methods: analysis, numerical evidence and possible cures. *International Journal of Computational Fluid Dynamics*, 11:128–143, 2002.

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

[59] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *Journal of Computational Physics*, 118:120–130, 1995.

[60] F. Vilar, C.W. Shu, and P.-H. Maire. Positivity-preserving cell-centered Lagrangian schemes for multi-material compressible fl ows: From first-order to high-orders. Part II: The two-dimensional case. *Journal of Computational Physics*, 312:416–442, 2016.

[61] M.L. Wilkins. Calculation of elastic plastic flow. In B. Alder, S. Fernbach, and M. Rotenberg, editors, *Methods in Computational Physics*, volume 3, pages 211–263. Academic Press, New York, 1964.

[62] Y. Yang, Q. Zhang, and D.H. Sharp. Small amplitude theory of RichtmyerMeshkov instability. *Physics of Fluids*, 6:1856–1873, 1994.

[63] O. Zanotti, M. Dumbser, R. Loubère, and S.Diot. A posteriori subcell limiting for Discontinuous Galerkin Finite Element method for hyperbolic system of conservation laws. *Journal of Computational Physics*, 278:47–75, 2014.

[64] X. Zhang and C.W. Shu. Positivity–preserving high order finite difference WENO schemes for compressible Euler equations . *Journal of Computational Physics*, 231:2245–2258, 2012.

View publication stats