High order accurate direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for non-conservative hyperbolic systems with stiff source terms

Walter Boscheri[1,*], Raphaël Loubère[2]

[1] *Laboratory of Applied Mathematics, Department of Civil, Environmental and Mechanical Engineering, University of Trento, I-38123 Trento, Italy.*
[2] *CNRS and Institut de Mathématiques de Toulouse (IMT) Université Paul-Sabatier, Toulouse, France*

Abstract. In this paper we present a 2D/3D high order accurate finite volume scheme in the context of direct Arbitrary-Lagrangian-Eulerian algorithms for general hyperbolic systems of partial differential equations with non-conservative products and stiff source terms. This scheme is constructed with a single stencil polynomial reconstruction operator, a one-step space-time ADER integration which is suitably designed for dealing even with stiff sources, a nodal solver with relaxation to determine the mesh motion, a path-conservative integration technique for the treatment of non-conservative products and an *a posteriori* stabilization procedure derived from the so-called Multi-dimensional Optimal Order Detection (MOOD) paradigm. In this work we consider the seven equation Baer-Nunziato model of compressible multi-phase flows as a representative model involving non-conservative products as well as relaxation source terms which are allowed to become stiff. The new scheme is validated against a set of test cases on 2D/3D unstructured moving meshes on parallel machines and the high order of accuracy achieved by the method is demonstrated by performing a numerical convergence study. Classical Riemann problems and explosion problems with exact solutions are simulated in 2D and 3D. The overall numerical code is also profiled to provide an estimate of the computational cost required by each component of the whole algorithm.

Key words: direct Arbitrary-Lagrangian-Eulerian, a posteriori MOOD stabilization, Baer-Nunziato model, stiff source terms, non-conservative products, unstructured mesh, ADER, high order of accuracy in space and time, high performance computing (HPC), hyperbolic conservation laws

*Corresponding author. *Email addresses:* walter.boscheri@unitn.it (W. Boscheri), raphael.loubere@math.univ-toulouse.fr (R. Loubère).
Partially supported by ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02

1 Introduction

In this work we pursue the development of the direct cell-centered Arbitrary-Lagrangian-Eulerian (ALE) ADER algorithm [**?**] supplemented with *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) stabilization technique. In [**?**] the ALE-ADER-MOOD numerical scheme has been solely tested and validated on the hydrodynamics system of conservation laws in multiple space dimensions on unstructured meshes. The hydrodynamics equations have been used as a first non-trivial hyperbolic system generating complex physical features (vorticity, shocks, contacts, wave interactions, etc.) which might be difficult to capture with a numerical scheme. We have shown that the ALE-ADER-MOOD approach can accurately solve such a system in 2D and 3D on unstructured moving grids.

Nonetheless, the Euler equations of compressible gas dynamics are relatively simple compared to more advanced models used in different fields of computational physics. Unfortunately the complexity of these models immutably leads to more equations to be solved as well as to more demanding terms, such as stiff sources, non-conservative products and/or constraints (divergence free, positivity). Dealing with such systems is an interesting challenge for high-order numerical schemes on moving unstructured meshes.

In this work we consider the seven-equation Baer-Nunziato (BN) system [2] that models compressible two-phase flows. The BN model consists in the combination of two systems of compressible Euler equations, one for each phase, coupled together via non-conservative products and relaxation sources. Exact numerical solutions, in ideal situations, can also be derived, which help the validation of our numerical method. The Baer-Nunziato system is used in this paper as a representative example of more complex physical systems of PDEs encountered in other fields of physics, hence requiring efficient, accurate and robust algorithms to be solved numerically.

Our aim is to show that the *a posteriori* MOOD stabilization technique developed in [13, 21, 22, 43**?**] can deal with such complex balance laws for which the occurrence of non-conservative and/or (stiff) source terms is part of modeling, and, as such, must be properly solved. The general framework for our study originates from [**?**], where the *a priori* WENO technique is then replaced by an *a posteriori* MOOD technique [**?**]. Within the *a posteriori* MOOD context several difficult properties are more easily fulfilled such as the positivity preservation of some physical quantities (density, internal energy and pressure). This substitution must be adapted to the presence of non-conservative products and stiff sources.

The rest of this paper is organized as follows. The second section briefly presents the direct high order accurate unlimited ADER Arbitrary-Lagrangian-Eulerian (ALE) scheme that was previously designed in [**?**], while in Section 3 the *a posteriori* MOOD technique is fully described, including the detection criteria which are properly selected and chosen for the Baer-Nunziato model. Decrementing technique as well as implementation issues and developer choices (cascade of schemes, parachute bullet-proof scheme) are discussed there. In Section 4 we gather the numerical results for a set of test cases

run in 2D and 3D. Complex flows involving interacting shock waves, contact discontinuities and rarefaction waves are considered. A profiling of the overall direct ALE code is provided in order to measure the relative cost of its main components. From these tests and a code profiling we will be able to assess the validity, efficiency and robustness of our high order accurate ALE scheme supplemented with the *a posteriori* MOOD detection and decrementing technique. Conclusions and perspectives are proposed in Section 5.

2 Direct high accurate unlimited ADER Arbitrary-Lagrangian-Eulerian scheme

In this paper we consider nonlinear systems of hyperbolic balance laws which may contain non-conservative products and source terms that might become stiff. For the sake of completeness we present in the following the three dimensional formulation of the governing equations, knowing that the two dimensional equations can be obtained by neglecting the z component and all its associated variables. A general formulation of such systems is given by

$$\frac{\partial \mathbf{Q}}{\partial t}+\nabla\cdot\mathbf{F}(\mathbf{Q})+\mathbf{B}(\mathbf{Q})\cdot\nabla\mathbf{Q}=\mathbf{S}(\mathbf{Q}),\qquad \mathbf{x}\in\Omega\subset\mathbb{R}^3,t\in\mathbb{R}_0^+, \tag{2.1}$$

where $\mathbf{Q}=(q_1,q_2,\cdots,q_v)$ is the vector of v conserved variables, $\mathbf{F}=(\mathbf{f},\mathbf{g},\mathbf{h})$ is the conservative nonlinear flux tensor, $\mathbf{B}=(\mathbf{B}_1,\mathbf{B}_2,\mathbf{B}_3)$ is the purely non-conservative part of the system written in block-matrix notation and $\mathbf{S}(\mathbf{Q})$ is the vector of algebraic source terms. We furthermore introduce the abbreviation $\mathbf{P}=\mathbf{P}(\mathbf{Q},\nabla\mathbf{Q})=\mathbf{B}(\mathbf{Q})\cdot\nabla\mathbf{Q}$ to ease notation in some parts of the manuscript.

In our moving mesh framework the computational domain $\Omega(t)\subset\mathbb{R}^3$ is discretized at any time level t^n by a set of moving and deforming tetrahedral elements T_i^n. N_E denotes the total number of elements and the union of all elements is referred to as the tetrahedrization $\mathcal{T}_\Omega^n$ of the domain:

$$\mathcal{T}_\Omega^n=\bigcup_{i=1}^{N_E}T_i^n. \tag{2.2}$$

We assume that the mesh configuration continuously changes in time. Consequently we adopt the mapping between the physical element T_i^n to the reference element T_e defined in the reference coordinate system $\xi-\eta-\zeta$, see Fig. 1.

The spatial reference element T_e is the unit tetrahedron defined by the vertices $\boldsymbol{\xi}_{e,1}=(\xi_{e,1},\eta_{e,1},\zeta_{e,1})=(0,0,0)$, $\boldsymbol{\xi}_{e,2}=(\xi_{e,2},\eta_{e,2},\zeta_{e,2})=(1,0,0)$, $\boldsymbol{\xi}_{e,3}=(\xi_{e,3},\eta_{e,3},\zeta_{e,3})=(0,1,0)$ and $\boldsymbol{\xi}_{e,4}=(\xi_{e,4},\eta_{e,4},\zeta_{e,4})=(0,0,1)$, where $\boldsymbol{\xi}=(\xi,\eta,\zeta)$ and $\mathbf{x}=(x,y,z)$ are the vector of the spatial coordinates in the reference system and the position vector in the physical system, respectively. Let $\mathbf{X}_{k,i}^n=(X_{k,i}^n,Y_{k,i}^n,Z_{k,i}^n)$ be the vector of physical spatial coordinates of the k-th vertex of tetrahedron T_i^n, then the linear mapping from T_i^n to T_e reads

$$\mathbf{x}=\mathbf{X}_{1,i}^n+\left(\mathbf{X}_{2,i}^n-\mathbf{X}_{1,i}^n\right)\xi+\left(\mathbf{X}_{3,i}^n-\mathbf{X}_{1,i}^n\right)\eta+\left(\mathbf{X}_{4,i}^n-\mathbf{X}_{1,i}^n\right)\zeta. \tag{2.3}$$

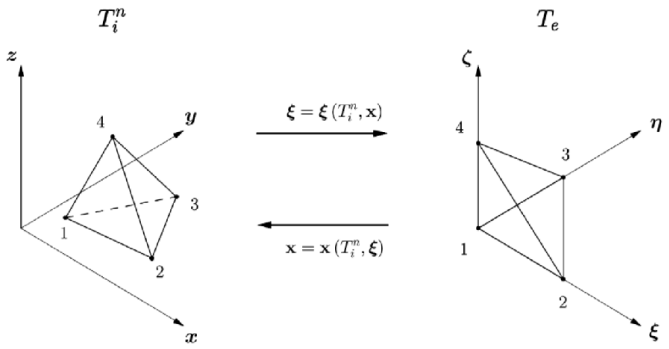

Figure 1: Spatial mapping from the physical element T_i^n defined with $\mathbf{x}=(x,y,z)$ to the unit reference tetrahedron T_e in $\boldsymbol{\xi}=(\xi,\eta,\zeta)$.

For any finite volume scheme data are represented by piecewise constant cell averages at a given time level. As a consequence, we define at each time level t^n within the control volume T_i^n the mean value of the state vector $\mathbf{Q}_i^n$ as

$$\mathbf{Q}_i^n = \frac{1}{|T_i^n|} \int_{T_i^n} \mathbf{Q}(\mathbf{x},t^n)\, d\mathbf{x}, \tag{2.4}$$

where $|T_i^n|$ is the volume of element T_i^n.

In the next section a polynomial reconstruction technique is described and used to obtain piecewise high order unlimited polynomials $\mathbf{w}_h(\mathbf{x},t^n)$ from the known cell averages $\mathbf{Q}_i^n$. Contrarily to [25**?**] we do not employ a WENO technique to limit the reconstruction, instead we use an iterative *a posteriori* MOOD loop presented in Section 3. High order of accuracy in time is achieved by applying a local space-time discontinuous Galerkin predictor method starting from the high order reconstruction polynomials $\mathbf{w}_h(\mathbf{x},t^n)$ (see Section 2.2).

2.1 Polynomial reconstruction

The reconstruction operator generates piecewise polynomials $\mathbf{w}_h(\mathbf{x},t^n)$ of degree M which are computed for each element T_i^n considering the so-called reconstruction stencil $\mathcal{S}_i$ and its associated known cell averages. The reconstruction stencil $\mathcal{S}_i$ is composed of a number n_e of neighbor elements of T_i^n, which is bigger than smallest number $\mathcal{M} = (M+1)(M+2)(M+3)/6$ needed to reach the nominal order of accuracy $M+1$, according to [6, 41, 47]. As suggested in [29, 30], for an unstructured mesh we usually take $n_e \simeq d \cdot \mathcal{M}$, with $d \in [2,3]$ representing the number of space dimensions. The stencil is recursively constructed: first the Voronoi neighborhood of the current cell T_i^n is listed in stencil $\mathcal{S}_i$, then, if the number of elements has not reached the prescribed number n_e, the Voronoi neighborhood of any cell already stored in $\mathcal{S}_i$ is added to this list, etc.

In the WENO approach adopted in [8, 25, 25**? ? ? ?**], nine reconstruction stencils are first determined and further used to compute nine different polynomials for each cell of the computational domain. These stencils are supposed to cover sufficiently enough "directions" for the polynomials in order to "catch" local phenomena. Next, these nine polynomials are blended with nonlinear weights to obtain the actual high order WENO limited polynomials $\mathbf{w}_h(\mathbf{x},t^n)$.

In this work only one central stencil is considered with n_e elements, hence we compute only one high order reconstruction polynomial per cell, as done in [**?**]. Let us call this unique stencil $\mathcal{S}_i$, hence

$$\mathcal{S}_i = \bigcup_{j=1}^{n_e} T^n_{m(j)}, \tag{2.5}$$

where $1 \leq j \leq n_e$ is a local index counting the elements in the stencil and $m(j)$ is a mapping from the local index j to the global index of the element in $\mathcal{T}^n_\Omega$. We rely on the orthogonal Dubiner-type basis functions $\psi_l(\xi,\eta,\zeta)$ [17, 23, 40], defined on the reference element T_e, to explicitly write the high order reconstruction polynomial as

$$\mathbf{w}_h(\mathbf{x},t^n) = \sum_{l=1}^{\mathcal{M}} \psi_l(\boldsymbol{\xi})\hat{\mathbf{w}}^n_{l,i} := \psi_l(\boldsymbol{\xi})\hat{\mathbf{w}}^n_{l,i}, \tag{2.6}$$

where the mapping to the reference coordinate system is given by (2.3) and $\hat{\mathbf{w}}^n_{l,i}$ denote the unknown degrees of freedom, also called expansion coefficients, of the reconstruction polynomial on stencil $\mathcal{S}_i$ for element T^n_i at time t^n. In the rest of the paper we will use the tensor index notation based on the Einstein summation convention implying summation over two equal indices.

We require integral conservation for the reconstruction on each element T^n_j belonging to stencil $\mathcal{S}_i$, therefore

$$\frac{1}{|T^n_j|}\int_{T^n_j} \psi_l(\boldsymbol{\xi})\hat{\mathbf{w}}^n_{l,i} d\mathbf{x} = \mathbf{Q}^n_j, \qquad \forall T^n_j \in \mathcal{S}_i. \tag{2.7}$$

This yields an overdetermined linear system of equations for the unknowns $\hat{\mathbf{w}}^n_{l,i}$ that can be solved using either a least squares technique (LSQ), see [30], or a more sophisticated singular value decomposition (SVD) algorithm. The so-called reconstruction matrix, that is given by the integrals of the linear system (2.7), depends on the current configuration of the elements in stencil $\mathcal{S}_i$ which are moving in an ALE framework. As a consequence the reconstruction matrix can not be inverted and stored in the preprocessing step as done in the Eulerian framework [30, 43]. Here, system (2.7) is solved on-the-fly at *each* timestep for each reconstruction stencil $\mathcal{S}_i$, whose topology remains constant in time since we assume that the mesh connectivity does not change. For further details on the reconstruction technique we refer the reader to [30**?**].

2.2 Local space-time Discontinuous Galerkin predictor on moving curved meshes

The reconstructed polynomials $\mathbf{w}_h(\mathbf{x},t^n)$ computed at time t^n are then *evolved* during one time step *locally* within each element $T_i(t)$, without needing any neighbor information, but still solving the original PDEs (2.1). As a result one obtains piecewise space-time polynomials of degree M, denoted by $\mathbf{q}_h(\mathbf{x},t)$, that allows the scheme to achieve high order of accuracy even in time.

An *element-local* weak space-time formulation of the governing equations (2.1) is employed, following the approach developed in the Eulerian framework on fixed grids by Dumbser et al. in [24, 27, 28, 36]. Later, such technique has been extended to the Lagrangian framework on moving grids in 1D and 2D [8, 25, 32**?** **?** **?**] and to moving tetrahedral meshes in 3D [**?**]. All the details regarding the description of the local space-time Galerkin predictor can be found in the aforementioned references, hence we limit us to briefly recall it in the following.

According to [27, 33, 36**?**] we use the local space-time Discontinuous Galerkin predictor method due to the presence of stiff source terms. Let $\mathbf{x}=(x,y,z)$ and $\boldsymbol{\xi}=(\xi,\eta,\zeta)$ be the spatial coordinate vectors defined in the physical and in the reference system, respectively, and let $\tilde{\mathbf{x}}=(x,y,z,t)$ and $\tilde{\boldsymbol{\xi}}=(\xi,\eta,\zeta,\tau)$ be the corresponding space-time coordinate vectors. Let furthermore $\theta_l=\theta_l(\tilde{\boldsymbol{\xi}})=\theta_l(\xi,\eta,\zeta,\tau)$ be a space-time basis function defined by the Lagrange interpolation polynomials passing through the space-time nodes $\tilde{\boldsymbol{\xi}}_m=(\xi_m,\eta_m,\zeta_m,\tau_m)$, which are defined by the tensor product of the spatial nodes of classical conforming high order finite elements in space and the Gauss-Legendre quadrature points in time.

Since the Lagrange interpolation polynomials lead to a *nodal* basis, the functions θ_l satisfy the following useful interpolation property:

$$\theta_l(\tilde{\boldsymbol{\xi}}_m)=\delta_{lm}, \tag{2.8}$$

where δ_{lm} denotes the Kronecker symbol. Following [24], the local solution $\mathbf{q}_h$, the fluxes $\mathbf{F}_h=(\mathbf{f}_h,\mathbf{g}_h,\mathbf{h}_h)$, the source term $\mathbf{S}_h$ and the non-conservative products $\mathbf{P}_h=\mathbf{B}(\mathbf{q}_h)\cdot\nabla\mathbf{q}_h$, are approximated within the space-time element $T_i(t)\times[t^n;t^{n+1}]$ with

$$\mathbf{q}_h=\mathbf{q}_h(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{q}}_{l,i},\quad \mathbf{F}_h=\mathbf{F}_h(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{F}}_{l,i},\quad \mathbf{S}_h=\mathbf{S}_h(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{S}}_{l,i},\quad \mathbf{P}_h=\mathbf{P}_h(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{P}}_{l,i}. \tag{2.9}$$

Using (2.8) we evaluate in a pointwise manner the degrees of freedom of $\mathbf{F}_h$, $\mathbf{S}_h$ and $\mathbf{P}_h$ from $\mathbf{q}_h$ as

$$\widehat{\mathbf{F}}_{l,i}=\mathbf{F}(\widehat{\mathbf{q}}_{l,i}),\quad \widehat{\mathbf{S}}_{l,i}=\mathbf{S}(\widehat{\mathbf{q}}_{l,i}),\quad \widehat{\mathbf{P}}_{l,i}=\mathbf{P}(\widehat{\mathbf{q}}_{l,i},\nabla\widehat{\mathbf{q}}_{l,i}),\quad \nabla\widehat{\mathbf{q}}_{l,i}=\nabla\theta_m(\tilde{\boldsymbol{\xi}}_l)\widehat{\mathbf{q}}_{m,i}, \tag{2.10}$$

with $\nabla\widehat{\mathbf{q}}_{l,i}$ representing the gradient of $\mathbf{q}_h$ at node $\tilde{\boldsymbol{\xi}}_l$.

An isoparametric approach is adopted, where the mapping between the physical space-time coordinate vector $\tilde{\mathbf{x}}$ and the reference space-time coordinate vector $\tilde{\boldsymbol{\xi}}$ is represented by the *same* basis functions θ_l used for the discrete solution $\mathbf{q}_h$. Consequently we have

$$\mathbf{x}(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{x}}_{l,i},\qquad t(\tilde{\boldsymbol{\xi}})=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{t}_l, \tag{2.11}$$

where $\widehat{\mathbf{x}}_{l,i}=(\widehat{x}_{l,i},\widehat{y}_{l,i},\widehat{z}_{l,i})$ are the degrees of freedom of the spatial physical coordinates of the moving space-time control volume, which are unknown, while $\widehat{t}_l$ denote the *known* degrees of freedom of the physical time at each space-time node $\tilde{\mathbf{x}}_{l,i}=(\widehat{x}_{l,i},\widehat{y}_{l,i},\widehat{z}_{l,i},\widehat{t}_l)$. The mapping in time is simply linear

$$t=t_n+\tau\Delta t, \qquad \tau=\frac{t-t^n}{\Delta t}, \qquad \Rightarrow \qquad \widehat{t}_l=t_n+\tau_l\Delta t, \tag{2.12}$$

with t^n denoting the current time. Δt is the time step and it is computed under a classical Courant-Friedrichs-Levy number (CFL) stability condition, i.e.

$$\Delta t=\text{CFL}\min_{T_i^n}\frac{d_i}{|\lambda_{\max,i}|}, \qquad \forall T_i^n\in\Omega^n, \tag{2.13}$$

where d_i is the insphere diameter of tetrahedron T_i^n and $|\lambda_{\max,i}|$ corresponds to the maximum absolute value of the eigenvalues computed from the solution $\mathbf{Q}_i^n$ in T_i^n. On unstructured meshes the CFL stability condition must satisfy the more restrictive inequality $\text{CFL}\leq\frac{1}{d}$.

The Jacobian of the space-time transformation from the physical to the reference element and its inverse read

$$J_{st}=\frac{\partial\tilde{\mathbf{x}}}{\partial\tilde{\boldsymbol{\xi}}}=\begin{pmatrix} x_\xi & x_\eta & x_\zeta & x_\tau \\ y_\xi & y_\eta & y_\zeta & y_\tau \\ z_\xi & z_\eta & z_\zeta & z_\tau \\ 0 & 0 & 0 & \Delta t \end{pmatrix}, \qquad J_{st}^{-1}=\frac{\partial\tilde{\boldsymbol{\xi}}}{\partial\tilde{\mathbf{x}}}=\begin{pmatrix} \xi_x & \xi_y & \xi_z & \xi_t \\ \eta_x & \eta_y & \eta_z & \eta_t \\ \zeta_x & \zeta_y & \zeta_z & \zeta_t \\ 0 & 0 & 0 & \frac{1}{\Delta t} \end{pmatrix}. \tag{2.14}$$

We point out that in the Jacobian matrix $t_\xi=t_\eta=t_\zeta=0$ and $t_\tau=\Delta t$, as can be easily derived from the time mapping (2.12).

Let us introduce the nabla operator ∇ in the reference space $\boldsymbol{\xi}=(\xi,\eta,\zeta)$ and in the physical space $\mathbf{x}=(x,y,z)$:

$$\nabla_{\boldsymbol{\xi}}=\begin{pmatrix} \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{pmatrix}, \qquad \nabla=\begin{pmatrix} \frac{\partial}{\partial x} \\ \frac{\partial}{\partial y} \\ \frac{\partial}{\partial z} \end{pmatrix}=\begin{pmatrix} \xi_x & \eta_x & \zeta_x \\ \xi_y & \eta_y & \zeta_y \\ \xi_z & \eta_z & \zeta_z \end{pmatrix}\begin{pmatrix} \frac{\partial}{\partial\xi} \\ \frac{\partial}{\partial\eta} \\ \frac{\partial}{\partial\zeta} \end{pmatrix}=\left(\frac{\partial\boldsymbol{\xi}}{\partial\mathbf{x}}\right)^T\nabla_{\boldsymbol{\xi}}, \tag{2.15}$$

and two integral operators

$$[f,g]^\tau=\int_{T_e} f(\xi,\eta,\zeta,\tau)g(\xi,\eta,\zeta,\tau)d\boldsymbol{\xi}, \qquad \langle f,g\rangle=\int_0^1\int_{T_e} f(\xi,\eta,\zeta,\tau)g(\xi,\eta,\zeta,\tau)d\boldsymbol{\xi}d\tau, \tag{2.16}$$

that denote the scalar products of two functions f and g over the spatial reference element T_e at time τ and over the space-time reference element $T_e\times[0,1]$, respectively.

The system of balance laws (2.1) is then reformulated in the reference coordinate system (ξ,η,ζ) using the inverse of the associated Jacobian matrix (2.14) adopting the gradient notation (2.15) as

$$\frac{\partial \mathbf{Q}}{\partial \tau}+\Delta t\left[\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\xi}}\cdot\frac{\partial \boldsymbol{\xi}}{\partial t}+\left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^{T}\nabla_{\boldsymbol{\xi}}\cdot\mathbf{F}+\mathbf{B}(\mathbf{Q})\cdot\left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^{T}\nabla_{\boldsymbol{\xi}}\mathbf{Q}\right]=\Delta t\,\mathbf{S}(\mathbf{Q}). \tag{2.17}$$

By introducing the abbreviation $\mathbf{H}=\frac{\partial \mathbf{Q}}{\partial \boldsymbol{\xi}}\cdot\frac{\partial \boldsymbol{\xi}}{\partial t}+\left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^{T}\nabla_{\boldsymbol{\xi}}\cdot\mathbf{F}+\mathbf{B}(\mathbf{Q})\cdot\left(\frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\right)^{T}\nabla_{\boldsymbol{\xi}}\mathbf{Q}$, then (2.17) simplifies to

$$\frac{\partial \mathbf{Q}}{\partial \tau}+\Delta t\,\mathbf{H}=\Delta t\,\mathbf{S}(\mathbf{Q}). \tag{2.18}$$

The numerical approximation of $\mathbf{H}$ is computed by the same isoparametric approach (2.9), i.e. $\mathbf{H}_h=\theta_l(\tilde{\boldsymbol{\xi}})\widehat{\mathbf{H}}_{l,i}$.

Inserting this approximation and (2.9) into (2.18), then multiplying (2.18) with a space-time test function $\theta_k(\boldsymbol{\xi})$ and further integrating the resulting equation over the space-time reference element $T_e\times[0,1]$, one obtains a weak formulation of the original governing PDE (2.1):

$$\left\langle \theta_k,\frac{\partial \theta_l}{\partial \tau}\right\rangle\widehat{\mathbf{q}}_{l,i}=\langle\theta_k,\theta_l\rangle\,\Delta t\left(\widehat{\mathbf{S}}_{l,i}-\widehat{\mathbf{H}}_{l,i}\right).$$

The term on the left hand side can be integrated by parts in time, yielding

$$[\theta_k(\boldsymbol{\xi},1),\theta_l(\boldsymbol{\xi},1)]^1\widehat{\mathbf{q}}_{l,i}-\left\langle\frac{\partial \theta_k}{\partial \tau},\theta_l\right\rangle\widehat{\mathbf{q}}_{l,i}=[\theta_k(\boldsymbol{\xi},0),\psi_l(\boldsymbol{\xi})]^0\hat{\mathbf{w}}^n_{l,i}+\langle\theta_k,\theta_l\rangle\,\Delta t\left(\widehat{\mathbf{S}}_{l,i}-\widehat{\mathbf{H}}_{l,i}\right), \tag{2.19}$$

where the initial condition of the local Cauchy problem $\mathbf{w}^n_h$ has been introduced in a weak form. The integrals appearing in (2.19), which are given definition (2.16), are evaluated on the space-time reference element where also the basis functions $\theta_l(\tilde{\boldsymbol{\xi}})$ are defined. Gaussian quadrature formulae of suitable order of accuracy are employed, see [54] for details. Adopting the following more compact matrix-vector notation

$$\mathbf{K}_1=[\theta_k(\boldsymbol{\xi},1),\theta_l(\boldsymbol{\xi},1)]^1-\left\langle\frac{\partial \theta_k}{\partial \tau},\theta_l\right\rangle,\quad \mathbf{F}_0=[\theta_k(\boldsymbol{\xi},0),\psi_l(\boldsymbol{\xi})],\quad \mathbf{M}=\langle\theta_k,\theta_l\rangle, \tag{2.20}$$

we can rewrite system (2.19) into

$$\mathbf{K}_1\widehat{\mathbf{q}}_{l,i}=\mathbf{F}_0\hat{\mathbf{w}}^n_{l,i}+\Delta t\,\mathbf{M}\left(\widehat{\mathbf{S}}_{l,i}-\widehat{\mathbf{H}}_{l,i}\right). \tag{2.21}$$

Equation (2.21) constitutes *de facto* an element-local nonlinear system of algebraic equations for the unknown space-time expansion coefficients $\widehat{\mathbf{q}}_{l,i}$. It is solved using the following iterative scheme

$$\widehat{\mathbf{q}}^{r+1}_{l,i}-\Delta t\,\mathbf{K}_1^{-1}\mathbf{M}\widehat{\mathbf{S}}^{r+1}_{l,i}=\mathbf{K}_1^{-1}\left(\mathbf{F}_0\hat{\mathbf{w}}^n_{l,i}-\Delta t\,\mathbf{M}\widehat{\mathbf{H}}^r_{l,i}\right), \tag{2.22}$$

where r denotes the iteration number. Stiff algebraic source terms $\mathbf{S}$ must be *implicitly* discretized, see [27, 32, 33, 36].

Together with the solution, we have to evolve the geometry of the space-time control volume which moves in time. We choose to consider only the vertex coordinates of element T_i^n, whose motion is described by the ODE system

$$\frac{d\mathbf{x}}{dt} = \mathbf{V}(\mathbf{Q},\mathbf{x},t), \tag{2.23}$$

with $\mathbf{V} = \mathbf{V}(\mathbf{Q},\mathbf{x},t)$ denoting the local mesh velocity. Our *Arbitrary-Lagrangian-Eulerian* (ALE) method allows the mesh velocity to be chosen independently from the fluid velocity, so that the scheme may reduce either to a pure Eulerian approach in the case where $\mathbf{V} = 0$ or to an almost Lagrangian algorithm if $\mathbf{V}$ coincides with the fluid velocity $\mathbf{v}$. Any other choice for the mesh velocity is also possible leading to an ALE regime. Following the same philosophy as previously, the velocity inside element $T_i(t)$ is also expressed in terms of the space-time basis functions θ_l as $\mathbf{V}_h = \theta_l(\boldsymbol{\xi},\tau)\widehat{\mathbf{V}}_{l,i}$, with the notation $\widehat{\mathbf{V}}_{l,i} = \mathbf{V}(\hat{\mathbf{q}}_{l,i},\hat{\mathbf{x}}_{l,i},\hat{t}_l)$. This approach gives a full *space-time* description of the time evolution of the element geometry, because the solution of the trajectory equation (2.23) provides the velocity vector for *each* space-time degree of freedom $\tilde{\mathbf{x}}_{l,i} := (\hat{\mathbf{x}}_{l,i},\hat{t}_l)$. Consequently, by means of the isoparametric definition (2.11), one has a complete and continuous description of the *high order* space-time volume $T_i(t)\times\Delta t$.

The local space-time DG method is used again to solve (2.23) for the unknown coordinate vector $\tilde{\mathbf{x}}_l = (x_l,y_l,z_l,t_l)$, according to [8, 25, 32], hence

$$\mathbf{K}_1\tilde{\mathbf{x}}_l = [\theta_k(\boldsymbol{\xi},0),\mathbf{x}(\boldsymbol{\xi},t^n)]^0 + \Delta t\,\mathbf{M}\widehat{\mathbf{V}}_{l,i}, \tag{2.24}$$

where $\mathbf{x}(\boldsymbol{\xi},t^n)$ is given by the mapping (2.3) based on the known vertex coordinates of tetrahedron T_i^n at time t^n. The above system is iteratively solved together with (2.22) until the residuals of the predicted solution at iteration r are less than a user-given tolerance, typically set to 10^{-12}. Since the initial condition for the predictor stage is given by the unlimited reconstruction polynomials $\mathbf{w}_h(\mathbf{x},t^n)$ the iterative process described above might fail, hence never achieving convergence. It may be related to positivity problems, as well as to unphysical oscillations or Gibbs phenomena. Such issues will be properly treated *a posteriori* by the MOOD loop, illustrated in Section 3, therefore we set also a maximum number of iterations r_{max} for the solution of the coupled nonlinear system (2.22)-(2.24). According to [24], for the linear homogeneous case at most $(M+1)$ iterations are needed to reach convergence. Because $M \leq 5$ in our work we typically set $r_{max} = 20$ to be more generous in the case of this nonlinear inhomogeneous PDEs with stiff sources like system (3.1). In this way the predictor stage ends either because the prescribed tolerance has been reached or because the maximum number of iterations has been exceeded.

Once the above procedure is performed for all cells, an element-local predictor for the numerical solution $\mathbf{q}_h$, for the fluxes $\mathbf{F}_h = (\mathbf{f}_h,\mathbf{g}_h,\mathbf{h}_h)$, for the source term $\mathbf{S}_h$ and also for the mesh velocity $\mathbf{V}_h$ is available. This procedure is *locally* carried out for each cell,

consequently it remains to update the mesh motion *globally*, by assigning a unique velocity vector to each node. To address this issue, in the next section a local nodal solver algorithm for the velocity together with an embedded rezoning technique are presented.

2.3 Mesh motion

The aim of any ALE scheme is to follow as closely as possible the flow motion. The fluid flow can generate highly deformed cells that may drastically reduce the admissible timestep, according to (2.47), or even lead to tangled elements. In order to guarantee excellent resolution properties for contact waves and material interfaces together with a good mesh quality, the mesh velocity must be chosen carefully. For this reason a suitable Lagrangian nodal solver technique [20, 45] is used to assign to each node k a unique velocity vector $\overline{\mathbf{V}}_k$ accurately representing the "true" fluid velocity, therefore leading to the new vertex Lagrangian coordinates

$$\mathbf{X}_k^{Lag} = \mathbf{X}_k^n + \Delta t \overline{\mathbf{V}}_k, \tag{2.25}$$

where $\mathbf{X}_k^n$ are the coordinates of node k at the current time level t^n.

To maintain an overall acceptable geometrical mesh quality we rely on a local rezoning algorithm [42, 60] which yields a new vertex position $\mathbf{X}_k^{Rez}$ that does not take into account the underlying flow features but only local geometrical characteristics. In this work we use the algorithm proposed in [42] which carries on a minimization of a nodal based local objective function.

The final node position is given by a weighted linear combination between the Lagrangian coordinates $\mathbf{X}_k^{Lag}$ and the rezoned coordinates $\mathbf{X}_k^{Rez}$ using the relaxation algorithm from [35, 44], that is

$$\mathbf{X}_k^{n+1} = \mathbf{X}_k^{Lag} + \omega_k \left(\mathbf{X}_k^{Rez} - \mathbf{X}_k^{Lag} \right), \tag{2.26}$$

where ω_k is a node-based coefficient associated to the deformation of the Lagrangian grid over the time step Δt, see [35, 44] for the details.

Since we are dealing with a direct ALE formulation the mesh velocity is a degree of freedom. As a consequence we could run our ALE code in a pure Eulerian regime by setting $\mathbf{X}_k^{n+1} = \mathbf{X}_k^0$ or in an almost Lagrangian regime with $\mathbf{X}_k^{n+1} = \mathbf{X}_k^{Lag}$. We could also force $\omega_k = 1$ leading to a pure geometrically rezoned mesh with coordinates $\mathbf{X}_k^{n+1} = \mathbf{X}_k^{Rez}$ or we might even adopt any user-given mesh velocity according to a prescribed boundary velocity.

2.4 Finite volume scheme

The same approach already developed in two and three space dimensions presented in [8, 25**?**] is briefly summarized here. To begin with, the governing PDE (2.1) is more

compactly reformulated using a space-time divergence operator $\tilde{\nabla}$

$$\tilde{\nabla}\cdot\tilde{\mathbf{F}}+\tilde{\mathbf{B}}(\mathbf{Q})\cdot\tilde{\nabla}\mathbf{Q}=\mathbf{S}(\mathbf{Q}),\qquad \tilde{\nabla}=\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z},\frac{\partial}{\partial t}\right)^{T}, \tag{2.27}$$

where the space-time flux tensor $\tilde{\mathbf{F}}$ and the system matrix $\tilde{\mathbf{B}}$ are given by $\tilde{\mathbf{F}}=(\mathbf{f},\mathbf{g},\mathbf{h},\mathbf{Q})$ and $\tilde{\mathbf{B}}=(\mathbf{B}_1,\mathbf{B}_2,\mathbf{B}_3,0)$. For the computation of the state vector at the new time level $\mathbf{Q}^{n+1}$, the balance law (2.27) is integrated over a four-dimensional space-time control volume $\mathcal{C}_i^n=T_i(t)\times\left[t^n;t^{n+1}\right]$, which after the application of the theorem of Gauss yields

$$\int_{\partial\mathcal{C}_i^n}\tilde{\mathbf{F}}\cdot\,\tilde{\mathbf{n}}\,dS+\int_{\mathcal{C}_i^n}\tilde{\mathbf{B}}(\mathbf{Q})\cdot\tilde{\nabla}\mathbf{Q}\,d\mathbf{x}dt=\int_{\mathcal{C}_i^n}\mathbf{S}(\mathbf{Q})\,d\mathbf{x}dt. \tag{2.28}$$

The non-conservative products are treated with a path-conservative approach [11, 12, 25, 26, 28, 31, 48, 49, 57], hence leading to

$$\int_{\partial\mathcal{C}_i^n}\left(\tilde{\mathbf{F}}+\tilde{\mathbf{D}}\right)\cdot\,\tilde{\mathbf{n}}\,dS+\int_{\mathcal{C}_i^n\setminus\partial\mathcal{C}_i^n}\tilde{\mathbf{B}}(\mathbf{Q})\cdot\tilde{\nabla}\mathbf{Q}\,d\mathbf{x}dt=\int_{\mathcal{C}_i^n}\mathbf{S}(\mathbf{Q})\,d\mathbf{x}dt, \tag{2.29}$$

where a new term $\tilde{\mathbf{D}}$ has been introduced in order to take into account potential jumps of the solution $\mathbf{Q}$ on the space-time element boundaries $\partial\mathcal{C}_i^n$. This term is computed by the path integral

$$\tilde{\mathbf{D}}\cdot\tilde{\mathbf{n}}=\frac{1}{2}\int_0^1\tilde{\mathbf{B}}\left(\boldsymbol{\Psi}(\mathbf{Q}^-,\mathbf{Q}^+,s)\right)\cdot\tilde{\mathbf{n}}\frac{\partial\boldsymbol{\Psi}}{\partial s}ds=\frac{1}{2}\left(\int_0^1\tilde{\mathbf{B}}\left(\boldsymbol{\Psi}(\mathbf{Q}^-,\mathbf{Q}^+,s)\right)\cdot\tilde{\mathbf{n}}ds\right)\left(\mathbf{Q}^+-\mathbf{Q}^-\right), \tag{2.30}$$

where the integration path $\boldsymbol{\Psi}$ in (2.30) is chosen to be a simple straight-line segment $\boldsymbol{\Psi}(\mathbf{Q}^-,\mathbf{Q}^+,s)=\mathbf{Q}^-+s(\mathbf{Q}^+-\mathbf{Q}^-)$ [12, 28, 31, 48] and $(\mathbf{Q}^-,\mathbf{Q}^+)$ are the conserved variables in element T_i^n and its direct neighbor T_j^n, respectively. In (2.29) the first integral is written as the sum of the fluxes computed over the three-dimensional space-time volume $\partial\mathcal{C}_i^n$, given by the evolution of each face of tetrahedron $T_i(t)$ within the timestep Δt. Here $\tilde{\mathbf{n}}=(\tilde{n}_x,\tilde{n}_y,\tilde{n}_z,\tilde{n}_t)$ denotes the outward pointing space-time unit normal vector on the varying space-time face ∂C_i^n.

Let $\mathcal{N}_i$ denote the *Neumann neighborhood* of tetrahedron $T_i(t)$, which is the set of directly adjacent neighbors $T_j(t)$ that share a common face $\partial T_{ij}(t)$ with tetrahedron $T_i(t)$. The space-time volume ∂C_i^n is composed by four space-time sub-volumes ∂C_{ij}^n, each of them defined for each face of tetrahedron $T_i(t)$, and two more space-time sub-volumes, T_i^n and T_i^{n+1}, that represent the tetrahedron configuration at times t^n and t^{n+1}, respectively (see [**?**] for details). Hence, the space-time volume ∂C_i^n involves overall a total number of six space-time sub-volumes, i.e.

$$\partial C_i^n=\left(\bigcup_{T_j(t)\in\mathcal{N}_i}\partial C_{ij}^n\right)\cup T_i^n\cup T_i^{n+1}. \tag{2.31}$$

Each of the space-time sub-volumes is mapped to a reference element in order to simplify the integral computation. For the configurations at the current and at the new time level, T_i^n and T_i^{n+1}, we use the mapping (2.3) with $(\xi,\eta,\zeta) \in [0;1]$. The space-time unit normal vectors simply read $\tilde{\mathbf{n}} = (0,0,0,-1)$ for T_i^n and $\tilde{\mathbf{n}} = (0,0,0,1)$ for T_i^{n+1}, since these volumes are orthogonal to the time coordinate. For the lateral sub-volumes ∂C_{ij}^n we adopt a linear parametrization to map the physical volume to a four-dimensional space-time reference prism.

Starting from the old vertex coordinates $\mathbf{X}_{ik}^n$ and the new ones $\mathbf{X}_{ik}^{n+1}$, that are *known* from the mesh motion algorithm described in Section 2.3, the lateral sub-volumes are parametrized using a set of linear basis functions $\beta_k(\chi_1,\chi_2,\tau)$ that are defined on a local reference system (χ_1,χ_2,τ) which is oriented orthogonally w.r.t. the face $\partial T_{ij}(t)$ of tetrahedron T_i^n, e.g. the reference time coordinate τ is orthogonal to the reference space coordinates (χ_1,χ_2) that lie on $\partial T_{ij}(t)$. The temporal mapping is simply given by $t = t^n + \tau\Delta t$, hence $t_{\chi_1} = t_{\chi_2} = 0$ and $t_\tau = \Delta t$. The lateral space-time volume ∂C_{ij}^n is defined by six vertices of physical coordinates $\tilde{\mathbf{X}}_{ij,k}^n$. The first three vectors $(\mathbf{X}_{ij,1}^n, \mathbf{X}_{ij,2}^n, \mathbf{X}_{ij,3}^n)$ are the nodes defining the common face $\partial T_{ij}(t^n)$ at time t^n, while the same procedure applies at the new time level t^{n+1}. Therefore the six vectors $\tilde{\mathbf{X}}_{ij,k}^n$ are given by

$$\tilde{\mathbf{X}}_{ij,1}^n = \left(\mathbf{X}_{ij,1}^n, t^n\right), \qquad \tilde{\mathbf{X}}_{ij,2}^n = \left(\mathbf{X}_{ij,2}^n, t^n\right), \qquad \tilde{\mathbf{X}}_{ij,3}^n = \left(\mathbf{X}_{ij,3}^n, t^n\right),$$

$$\tilde{\mathbf{X}}_{ij,4}^n = \left(\mathbf{X}_{ij,1}^{n+1}, t^{n+1}\right), \qquad \tilde{\mathbf{X}}_{ij,5}^n = \left(\mathbf{X}_{ij,2}^{n+1}, t^{n+1}\right), \qquad \tilde{\mathbf{X}}_{ij,6}^n = \left(\mathbf{X}_{ij,3}^{n+1}, t^{n+1}\right), \tag{2.32}$$

and the parametrization for ∂C_{ij}^n reads

$$\partial C_{ij}^n = \tilde{\mathbf{x}}(\chi_1,\chi_2,\tau) = \sum_{k=1}^{6} \beta_k(\chi_1,\chi_2,\tau)\tilde{\mathbf{X}}_{ij,k}^n, \tag{2.33}$$

with $0 \le \chi_1 \le 1$, $0 \le \chi_2 \le 1-\chi_1$ and $0 \le \tau \le 1$. The linear basis functions $\beta_k(\chi_1,\chi_2,\tau)$ are given by

$$\begin{aligned}
\beta_1(\chi_1,\chi_2,\tau) &= (1-\chi_1-\chi_2)(1-\tau), & \beta_4(\chi_1,\chi_2,\tau) &= (1-\chi_1-\chi_2)(\tau) \\
\beta_2(\chi_1,\chi_2,\tau) &= \chi_1(1-\tau), & \beta_5(\chi_1,\chi_2,\tau) &= \chi_1\tau, \\
\beta_3(\chi_1,\chi_2,\tau) &= \chi_2(1-\tau), & \beta_6(\chi_1,\chi_2,\tau) &= \chi_2\tau.
\end{aligned} \tag{2.34}$$

The coordinate transformation is associated with a matrix $\mathcal{T}$ that reads

$$\mathcal{T} = \left(\hat{\mathbf{e}}, \frac{\partial\tilde{\mathbf{x}}}{\partial\chi_1}, \frac{\partial\tilde{\mathbf{x}}}{\partial\chi_2}, \frac{\partial\tilde{\mathbf{x}}}{\partial\tau}\right)^T, \tag{2.35}$$

with $\hat{\mathbf{e}} = (\hat{\mathbf{e}}_1,\hat{\mathbf{e}}_2,\hat{\mathbf{e}}_3,\hat{\mathbf{e}}_4)$. Let $\hat{\mathbf{e}}_p$ represent the unit vector aligned with the p-th axis of the physical coordinate system (x,y,z,t) and let $\tilde{x}_q$ denote the q-th component of vector $\tilde{\mathbf{x}}$.

The determinant of $\mathcal{T}$ produces at the same time the space-time volume $|\partial C_{ij}^n|$ of the space-time sub-volume ∂C_{ij}^n and the space-time normal vector $\tilde{\mathbf{n}}_{ij}$, as

$$\tilde{\mathbf{n}}_{ij} = \left(\epsilon_{pqrs} \hat{\mathbf{e}}_p \frac{\partial \tilde{x}_q}{\partial \chi_1} \frac{\partial \tilde{x}_r}{\partial \chi_2} \frac{\partial \tilde{x}_s}{\partial \tau} \right) / |\partial C_{ij}^n|, \tag{2.36}$$

where the *Levi-Civita* symbol has been used according to the usual definition

$$\epsilon_{pqrs} = \begin{cases} +1, & \text{if } (p,q,r,s) \text{ is an } \textit{even} \text{ permutation of } (1,2,3,4), \\ -1, & \text{if } (p,q,r,s) \text{ is an } \textit{odd} \text{ permutation of } (1,2,3,4), \\ 0, & \text{otherwise,} \end{cases} \tag{2.37}$$

and with

$$|\partial C_{ij}^n| = \left\| \epsilon_{pqrs} \hat{\mathbf{e}}_p \frac{\partial \tilde{x}_q}{\partial \chi_1} \frac{\partial \tilde{x}_r}{\partial \chi_2} \frac{\partial \tilde{x}_s}{\partial \tau} \right\|. \tag{2.38}$$

The final one-step direct ALE finite volume scheme takes the following form:

$$|T_i^{n+1}| \mathbf{Q}_i^{n+1} = |T_i^n| \mathbf{Q}_i^n - \sum_{T_j \in \mathcal{N}_i} \int_0^1 \int_0^1 \int_0^{1-\chi_1} |\partial C_{ij}^n| \tilde{\mathbf{G}}_{ij} \cdot \tilde{\mathbf{n}}_{ij} d\chi_2 d\chi_1 d\tau + \int_{\mathcal{C}_i^n \setminus \partial \mathcal{C}_i^n} (\mathbf{S}_h - \mathbf{P}_h) \, d\mathbf{x} dt, \tag{2.39}$$

where in the term $\tilde{\mathbf{G}}_{ij} \cdot \tilde{\mathbf{n}}_{ij}$ we embed the Arbitrary-Lagrangian-Eulerian numerical flux function as well as the path-conservative jump term, which allows the discontinuity of the predictor solution $\mathbf{q}_h$ that occurs at the space-time sub-volume ∂C_{ij}^n to be properly resolved. The volume integrals appearing in (2.39) are approximated using multidimensional Gaussian quadrature rules [54] of suitable order of accuracy and the term $\tilde{\mathbf{G}}_{ij}$ is evaluated relying on a simple ALE Rusanov-type scheme [8, 32] as

$$\tilde{\mathbf{G}}_{ij} = \frac{1}{2} \left(\tilde{\mathbf{F}}(\mathbf{q}_h^+) + \tilde{\mathbf{F}}(\mathbf{q}_h^-) \right) \cdot \tilde{\mathbf{n}}_{ij} + \frac{1}{2} \left(\int_0^1 \tilde{\mathbf{B}}(\Psi) \cdot \tilde{\mathbf{n}} \, ds - |\lambda_{\max}| \mathbf{I} \right) \left(\mathbf{q}_h^+ - \mathbf{q}_h^- \right), \tag{2.40}$$

where $\mathbf{q}_h^-$ and $\mathbf{q}_h^+$ are the local space-time predictor solution inside element $T_i(t)$ and the neighbor $T_j(t)$, respectively, and $|\lambda_{\max}|$ denotes the maximum absolute value of the eigenvalues of the matrix $\tilde{\mathbf{A}} \cdot \tilde{\mathbf{n}}$ in space-time normal direction. Using the normal mesh velocity $\mathbf{V} \cdot \mathbf{n}$, matrix $\tilde{\mathbf{A}}_{\tilde{\mathbf{n}}}$ reads

$$\tilde{\mathbf{A}}_{\tilde{\mathbf{n}}} = \tilde{\mathbf{A}} \cdot \tilde{\mathbf{n}} = \left(\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2 + \tilde{n}_z^2} \right) \left[\left(\frac{\partial \mathbf{F}}{\partial \mathbf{Q}} + \mathbf{B} \right) \cdot \mathbf{n} - (\mathbf{V} \cdot \mathbf{n}) \mathbf{I} \right], \tag{2.41}$$

with $\mathbf{I}$ denoting the $\nu \times \nu$ identity matrix, $\mathbf{A} = \partial \mathbf{F} / \partial \mathbf{Q} + \mathbf{B}$ representing the classical Eulerian system matrix and $\mathbf{n}$ being the spatial unit normal vector given by

$$\mathbf{n} = \frac{(\tilde{n}_x, \tilde{n}_y, \tilde{n}_z)^T}{\sqrt{\tilde{n}_x^2 + \tilde{n}_y^2 + \tilde{n}_z^2}}. \tag{2.42}$$

Finally we remark that integration over a closed space-time control volume, as done in this scheme, automatically respects the geometric conservation law (GCL), since application of Gauss' theorem yields

$$\int_{\partial \mathcal{C}_i^n} \tilde{\mathbf{n}} dS = 0. \tag{2.43}$$

As already pointed out in [**?**] the numerical method in general allows a mass flux even for $\mathbf{V} = \mathbf{v}$. Consequently there is no associated pure Lagrangian scheme *in sensu stricto* to this numerical method. Nonetheless genuinely accurate results can be achieved with this high accurate numerical method as can be seen in [25**?**].

2.5 Timestep constraint

The timestep Δt, which is needed for the discretization of the governing equations (2.39), is computed taking into account two different criteria, namely a classical CFL stability condition and a user-defined geometrical limitation.

The Courant-Friedrichs-Levy (CFL) stability condition is given by (2.13), while the second criteria is based on the limitation of the rate of change of the element volume within one timestep, i.e. the volume of each cell T_i^n is not allowed either to increase or to decrease more than a certain threshold which is provided by the user at the beginning of the computation. Such a limitation is typically adopted in the Lagrangian and ALE framework [45, 46]. As clearly stated by (2.13), the timestep tends to become very small when the elements are highly compressed or stretched. Therefore, when compression occurs, we impose the condition

$$|T_i^{n+1}| \leq C_v |T_i^n|, \tag{2.44}$$

where C_v is a coefficient which sets the maximum admissible variation of volume for the cell. For our applications we choose $C_v = 0.8$ and the volume at the new time level $|T_i^{n+1}|$ is conveniently estimated by using the *current* vertex velocities $\overline{\mathbf{V}}_k^n$ for each vertex k of T_i^n to approximate the new vertex positions $\mathbf{X}_k^{n+1}$, hence obtaining

$$\mathbf{X}_k^{n+1} \approx \mathbf{X}_k^n + \Delta t_{V,i} \overline{\mathbf{V}}_k^n \qquad \forall k \in T_i^n, \tag{2.45}$$

with $\Delta t_{V,i}$ denoting the unknown timestep which satisfies the volume criterion (2.44). By using expression (2.45) to explicitly derive a formula for the new cell volume $|T_i^{n+1}|$, the volume restriction (2.44) constitutes a second or third order algebraic equation for the unknown $\Delta t_{V,i}$, in two and three space dimensions, respectively. The volume timestep is assumed to be as always the minimum between all $\Delta t_{V,i}$, i.e.

$$\Delta t_V = \min_{T_i^n} \Delta t_{V,i}. \tag{2.46}$$

According to [45] the final expression for the next timestep Δt^{n+1} is given by

$$\Delta t^{n+1} = \min\left(\Delta t_{CFL}, \Delta t_V, C_M \Delta t^n\right), \tag{2.47}$$

where Δt^n is the current timestep and $C_M = 1.01$ is a coefficient which allows the timestep to smoothly increase avoiding large and quick modifications of Δt.

2.6 Remarks

Last but not least, we would like to state clearly that within the family of high order one-step direct ALE methods proposed in this paper, the choices of the Riemann solver, the reconstruction technique and the mesh velocity are deliberately independent from each other, hence the method in general allows a mass flux. This means that even for $\mathbf{V} = \mathbf{v}$ the proposed scheme is *not* meant to be a pure Lagrangian method *in sensu stricto*. However, the family of schemes presented in this paper is able to resolve material interfaces and contact waves very well, much better than traditional high order Eulerian methods on fixed meshes.

3 Direct *a posteriori* limited ADER-MOOD Arbitrary-Lagrangian-Eulerian scheme

The direct ADER ALE numerical method previously described is of high accuracy both in space and time through the construction of high order accurate space and time polynomials. Without any spatial limiting the order of accuracy of the scheme is formally $M+1$. However, to ensure stability and robustness of the scheme, some dissipative mechanism must be added to deal with steep fronts or shock waves. In [8, 25**?**] the limiting was introduced by the use of one of the classical WENO reconstruction techniques [29, 30, 37, 38, 41]. The numerical results in [25**?**] show that this approach maintains the overall accuracy and provides the necessary robustness of the scheme. In [**?**] an alternative technique to WENO in this ALE context using the so-called *a posteriori* MOOD (Multi-dimensional Optimal Order Detection) method was adopted. The use of the *a posteriori* MOOD technique in replacement of WENO polynomial reconstruction in general improves the numerical efficiency of the overall method. Some reduction of memory and CPU time consumption along with some improvements of the overall accuracy are clearly observed, see [**?**]. Moreover some properties which previously were fulfilled relying on some special techniques are now built into the MOOD paradigm, for instance the WENO technique may demand the introduction of a flattener variable used to smear out the oscillations and to ensure positivity of density and pressure [5**?**].

In this paper we extend the MOOD approach to new different system of conservation laws with non-conservative products and stiff source terms. Basically the unlimited numerical technique described in the previous section must be supplemented with

some sort of limiting that we choose to build in an *a posteriori* framework. The *a posteriori* MOOD paradigm is based on the observation that it seems relatively easy to check against some criteria the validity of a numerical discrete solution provided by a numerical scheme solving a nonlinear system of equations *at the end of a timestep*. If some cells do not pass the check, they can be re-updated, that is re-computed starting from data at time t^n using a more dissipative numerical method. If these cells are still detected as problematic, then, at last, an ultra-dissipative and bullet-proof robust scheme is used. These ones are then re-updated for a last time within the current timestep starting again with data at t^n. This so-called iterative MOOD loop ends with cells updated either with a high accurate but less robust scheme, or, with a lower accurate but more robust one.

MOOD has been designed originally on fixed grid for Euler equations [13, 14, 21, 22] and more recently it has been successfully substituted to WENO within a 3D ADER high order finite volume scheme solving different systems of conservation laws [43]. The *a posteriori* MOOD concept has also been successfully employed as an *a posteriori* subcell finite volume limiter for high order Discontinuous Galerkin finite element schemes in [62, 63**?**], or as an efficient high-order finite volume solver for convection-diffusion problems [15, 16, 50] and also to construct all-entropy finite volume schemes [58, 59].

In the following we briefly review the MOOD approach as a stabilization technique along with the important *detection criteria* designed for each system of equations solved in this article.

3.1 MOOD paradigm as an *a posteriori* stabilization technique

The direct ALE framework previously presented is a nominally high-order accurate scheme in space and time. In Section 2.1 we have presented a polynomial reconstruction technique with no extra dissipation, meaning that the discrete numerical solution and the displaced mesh at time t^{n+1} may not be free from spurious oscillations. The MOOD loop is composed by the following three entities:

1. *Detection criteria*. The detection criteria set is a list of properties which are checked to assess if a numerical solution in a cell is acceptable at the end of a timestep. The first set of criteria is based on the physics solved by the simulation and correspond to properties that must be fulfilled to ensure physical admissibility of a numerical solution. They are called the *Physical Admissible Detection (PAD) criteria* and can not be decorrelated from the system of equations which is solved.

   The second set of criteria is based on numerical properties. These are called *Numerical Admissible Detection (NAD) criteria* and they ensure that the numerical solution is essentially non-oscillatory. In [13, 21, 22, 43, 62, 63] the NAD criteria are mostly built relying on the relaxed discrete maximum principle (DMP). Also we check if the computed solution is a representable datum, that is to say we check from a computer science point of view for *Not-a-Number* (NaN) or *infinite* (inf)situations.

If one of the PAD or the NAD criteria is not fulfilled, then *locally* some action must be taken to supplement the scheme with more dissipation.

2. *Parachute* or *bullet-proof* scheme. The last available scheme is called the *parachute* to express the fact that in any extremely difficult case this scheme is used as a last resort. As such, the candidate numerical solution provided must always be considered as an acceptable one. In most works involving a MOOD loop [13, 14, 21, 22, 43], the first-order Godunov finite volume scheme is chosen.

3. *Cascade*. The cascade is a list of ordered numerical schemes which one would like to try. Generally they are ranked from the most accurate and prone to instability scheme up to the least accurate but robust parachute scheme [43]. Here this sequence is related to the accuracy of the polynomial reconstructions. Precisely one sets a maximal polynomial degree $d_{\max}$ and, in an ideal situation, the corresponding method is a space-time $d_{\max}$-th order scheme. Several intermediate polynomial degrees for the reconstructions can be adopted, e.g. we can use the following decrementing list: $\mathbb{P}_{d_{\max}} \rightarrow \mathbb{P}_2 \rightarrow \mathbb{P}_1 \rightarrow \mathbb{P}_0$.

In an ALE implementation [**?**] as well as in the Eulerian context [43], the MOOD loop embraces the main evolution routines of the high order unlimited numerical scheme and, possibly, it iterates to recompute some cells marked as problematic by the detection criteria, see Figure 2.

3.2 Baer-Nunziato (BN) model

Next we consider the so-called Baer-Nunziato (BN) model for compressible two-phase flows [2] describing detonation waves in solid-gas combustion processes. Here the first phase is the solid phase with subscript 1 or s, while the second one is the gas phase (subscript 2 or g). Let us denote by k=1,2 the phase number and by ϕ_k the volume fraction of phase k fulfilling the condition $\phi_1 + \phi_2 = 1$, while ρ_k and p_k are accordingly the phase density and pressure. Furthermore the velocity vector of each phase is $\mathbf{u}_k = (u_k, v_k, w_k)$.

The seven-equation[†] Baer-Nunziato model with relaxation source terms constitutes

[†]In 3D the seven-equation model involves *de facto* 11 equations, hence leading to a relatively demanding system to solve numerically.

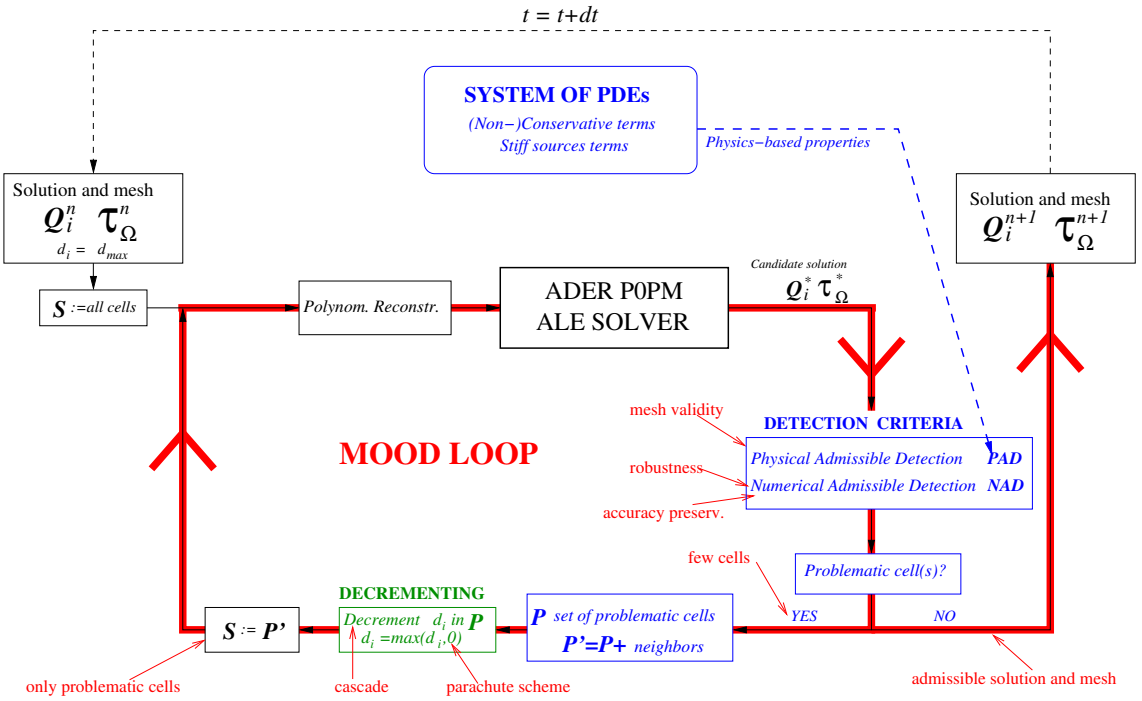

Figure 2: Sketch of the MOOD loop embracing an existing ALE ADER-$\mathbb{P}_0\mathbb{P}_M$ solver. The three main ingredients of the MOOD loop is the Detection Criteria (blue box), the cascade of schemes illustrated by the Decrementing procedure (green box) which possible ends with the parachute scheme (the scheme using $\mathbb{P}_0$ data, that is to say the first order Godunov finite volume scheme in our situation).

the following non-conservative system of nonlinear hyperbolic PDEs:

$$\begin{cases} \dfrac{\partial}{\partial t}(\phi_1\rho_1)+\nabla\cdot(\phi_1\rho_1\mathbf{u_1}) & = 0, \\ \dfrac{\partial}{\partial t}(\phi_1\rho_1\mathbf{u_1})+\nabla\cdot(\phi_1\rho_1\mathbf{u_1}\mathbf{u_1})+\nabla\phi_1 p_1 & = p_I\nabla\phi_1 \quad -\lambda(\mathbf{u_1}-\mathbf{u_2}), \\ \dfrac{\partial}{\partial t}(\phi_1\rho_1 E_1)+\nabla\cdot((\phi_1\rho_1 E_1+\phi_1 p_1)\mathbf{u_1}) & = -p_I\partial_t\phi_1 \quad -\lambda\,\mathbf{u_I}\cdot(\mathbf{u_1}-\mathbf{u_2}), \\ \dfrac{\partial}{\partial t}(\phi_2\rho_2)+\nabla\cdot(\phi_2\rho_2\mathbf{u_2}) & = 0, \\ \dfrac{\partial}{\partial t}(\phi_2\rho_2\mathbf{u_2})+\nabla\cdot(\phi_2\rho_2\mathbf{u_2}\mathbf{u_2})+\nabla\phi_2 p_2 & = p_I\nabla\phi_2 \quad -\lambda\,(\mathbf{u_2}-\mathbf{u_1}), \\ \dfrac{\partial}{\partial t}(\phi_2\rho_2 E_2)+\nabla\cdot((\phi_2\rho_2 E_2+\phi_2 p_2)\mathbf{u_2}) & = p_I\partial_t\phi_1 \quad -\lambda\,\mathbf{u_I}\cdot(\mathbf{u_2}-\mathbf{u_1}), \\ \dfrac{\partial}{\partial t}\phi_1+\mathbf{u_I}\nabla\phi_1 & = \quad \mu(p_1-p_2). \end{cases} \tag{3.1}$$

In this work we consider drastically simplified interphase drag and pressure relaxation source terms. Alternative formulations and choices for these terms can be found in [39]. To close the system the stiffened gas equation of state is used for each phase:

$$p_k=(\gamma_k-1)\rho_k e_k-\gamma_k\pi_k, \tag{3.2}$$

where e_k denotes the internal energy of phase k and the specific total energy is given by $E_k=e_k+\frac{1}{2}\mathbf{u_k}^2$. Finally the parameter μ characterizes the pressure relaxation and λ the friction between the two phases. Following [2, 39] the velocity at the interface I is assumed to be the solid velocity $\mathbf{u_I}=\mathbf{u_1}$, while for the interface pressure we choose the gas pressure $p_I=p_2$. Note that many other choices are possible [51, 52]. In any case the proper resolution of material interfaces, which are given by jumps in the volume fractions, is a challenging task for any numerical method solving (3.1). For this system the Physical Admissible Detection (PAD) and Numerical Admissible Detection (NAD) criteria are designed as follows.

3.2.1 Physical Admissible Detection criteria (PAD)

The Physical Admissible Detection (PAD) criteria state that for a numerical solution to be valid it must be "admissible". Several physical constraints for a numerical solution to be admissible are easily identified for the BN model: first, the phase density ρ_k and the pressure p_k must always be positive in a cell i, i.e.

$$\rho^*_{k,i}>0, \qquad p^*_{k,i}>0, \tag{3.3}$$

then, the volume fractions of each phase must verify $0\leq\phi_k\leq 1$. In fact only ϕ_1 is tested since ϕ_2 is deduced from the equation $\phi_1+\phi_2=1$. Furthermore in a moving framework the volume of any cell i must also be strictly positive, that is:

$$|T^*_i|>0. \tag{3.4}$$

3.2.2 Numerical Admissible Detection criteria (NAD)

The Numerical Admissible Detection (NAD) criteria are based on the relaxed Discrete Maximum Principle (DMP) with the so-called u2 criterion [22**?**] applied on each conservative variable $(\phi_k\rho_k, \phi_k\rho_k\mathbf{u}_k, \phi_k\rho_k E_k, \phi_1)$.

The DMP+u2 detection process acts on a generic variable A for a candidate solution A_i^* at time t^{n+1} in cell i for a given set of neighbor cells with index $j \in \mathcal{V}_i$. We define the set of vertex neighbors $\mathcal{V}_i$, which contains all neighbors of cell T_i that have a common vertex with T_i. First, if A_i^* fulfills the DMP, that is

$$\min_{j\in\mathcal{V}_i}(A_j^n, A_i^n) \le A_i^* \le \max_{j\in\mathcal{V}_i}(A_j^n, A_i^n), \tag{3.5}$$

then the cell is valid for this variable. If the DMP is not fulfilled, then one checks the u2 criterion [21, 22]. A candidate solution A_i^* in cell T_i^* which violates the DMP is nonetheless eligible if the following conditions hold:

$$\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0 \quad \text{and} \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \ge 1-\varepsilon, \tag{3.6}$$

where ε is a smoothness parameter set to $1/2$ and $\mathcal{X}_i^*$ represents a "measure" of the local discrete directional curvature, e.g. the second derivative of the third order polynomial reconstruction evaluated at the space barycenter of the control volume T_i^n. Specifically, we have

$$\begin{aligned} \mathcal{X}_i^* &= \frac{\partial A_i^n}{\partial x^2} + 2\frac{\partial A_i^n}{\partial xy} + \frac{\partial A_i^n}{\partial y^2} && \text{in 2D ,} \\ \mathcal{X}_i^* &= \frac{\partial A_i^n}{\partial x^2} + \frac{\partial A_i^n}{\partial y^2} + \frac{\partial A_i^n}{\partial z^2} + 2\frac{\partial A_i^n}{\partial xy} + 2\frac{\partial A_i^n}{\partial xz} + 2\frac{\partial A_i^n}{\partial yz} && \text{in 3D.} \end{aligned} \tag{3.7}$$

Note that the continuous approximation of the conserved variable A_i^n is provided by the high order reconstruction polynomial $\mathbf{w}_h^n$ computed at time t^n. Since we need at least a third order accurate scheme for evaluating the second derivatives appearing in (3.7), if the local order has been decremented for cell i, then the u2 criterion is no longer applied for that cell. The minimal and maximal values of (3.6) are defined as

$$\mathcal{X}_i^{min} = \min_{j\in\mathcal{V}_i}\left(\mathcal{X}_i^*, \mathcal{X}_j^*\right) \text{ and } \mathcal{X}_i^{max} = \max_{j\in\mathcal{V}_i}\left(\mathcal{X}_i^*, \mathcal{X}_j^*\right), \tag{3.8}$$

considering the Voronoi neighborhood $\mathcal{V}_i$ of cell i, i.e. all those neighbors which share at least one node with element T_i. Note that the detection of smooth local extrema has also been discussed in the context of extremum preserving PPM schemes [18] and MPWENO schemes [4, 55].

Last we add a test for impossible discrete values: NaN (Not-a-Number) and Inf (Infinite).

In Table 1 we summarize the detection criteria enforced by the physics underlying the Baer-Nunziato model. The positivity of the solution is implied if the PAD criteria are fulfilled.

System	Non-cons. product	(Stiff) source	Physical Admissibility Detection (PAD) *for appropriate variables*	Numerical Admissibility Detection (NAD) *for all conservative variables*
Baer-Nunziato system	✓	✓	$\rho_k > 0, \quad p_k > 0$ $0 \leq \phi_k \leq 1$	DMP+u2 NaN Inf

Table 1: Summary of the detection criteria for the MOOD method for the Baer-Nunziato model considered in this work. DMP: Discrete Maximum Principle check, u2: curvature regularity check [21, 22], NaN: Not-a-Number check, Inf: Infinite check.

3.3 Cascade, parachute and summary

In this work we have mimicked the cascade already used in the Eulerian and ALE frameworks [43**?**]: $\mathbb{P}_{d_{\max}} \to \mathbb{P}_1^{\text{LIM}} \to \mathbb{P}_0$. The maximal degree is set to $d_{\max} = 3,4$ and the $\mathbb{P}_1^{\text{LIM}}$ scheme uses $\mathbb{P}_1$ reconstructions on conservative variables with Barth & Jespersen slope limiting [7].

The MOOD loop first computes the unlimited $\mathbb{P}_{d_{\max}}$ candidate solution $\boldsymbol{Q}_i^*$ for each cell i and checks if any cell is problematic according to the detection criteria PAD and NAD. Then the solution in troubled/problematic cells $\boldsymbol{Q}_i^*$ is discarded and recomputed starting at time t^n with $\mathbb{P}_1^{\text{LIM}}$ reconstructions. For the numerical flux evaluation we also need $\mathbb{P}_1^{\text{LIM}}$ reconstructions for each direct neighbor T_j of T_i to assure that the cell is updated with same order fluxes for each face. Moreover the local space-time DG predictor is recomputed with a first order accurate time integration to secure the stiff source term integration. This new candidate solution in troubled cells is checked again for validity. At this point the candidate solution has possible $\mathbb{P}_{d_{\max}}$-updated cells and $\mathbb{P}_1^{\text{LIM}}$-updated ones. However some cells can still be invalid because the $\mathbb{P}_1^{\text{LIM}}$ may still produce some non-admissible states especially when complex hyperbolic systems are simulated and positivity is not ensured by construction. Last, for these still problematic cells, we rely on the parachute scheme using $\mathbb{P}_0$ "reconstructions", that is first order in space and time. Being our bullet-proof scheme, these remaining problematic cells are recomputed and valid by construction, in the sense that the PAD criteria are fulfilled[‡]. In the worst case scenario all cells in the domain are updated with the first-order $\mathbb{P}_0$ scheme. Contrarily, in the best possible situation (globally smooth flow), all cells are updated with the unlimited $\mathbb{P}_{d_{\max}}$ scheme. Usually, a numerical solution at the end of a timestep ends up with a mix of cells that have been updated either via $\mathbb{P}_{d_{\max}}$, $\mathbb{P}_1^{\text{LIM}}$ or $\mathbb{P}_0$. Note that the MOOD loop always converges to an acceptable discrete solution provided that the parachute scheme can produce such a solution.

In our implementation if any of the conservative variables does not fulfill the detection criteria, then one considers that all variables need correction on the next MOOD iteration, hence implying that one unique cell polynomial degree is decremented for all

[‡]Recall that the parachute scheme must be chosen so that PAD criteria are fulfilled at minima. Being the last resort scheme used to ensure that the simulation can still run, any arrangement with numerics and physics is allowed at this stage.

variables.

4 Numerical experiments

This section introduces and describes a list of representative test cases for the BN system of PDEs. Numerical solutions given by the direct ALE-MOOD numerical method of $M+1$th order of accuracy are presented ($M=3,4$). The CFL is generally set to $1/d$ if not stated otherwise. All tests are run on unstructured meshes made of triangles or tetrahedra with our code implemented under MPI framework. Moreover all simulations are run *at minima* on several cores on local machines or massively parallel ones (SuperMUC supercomputer based in Munich, Germany, at the Leibniz Rechenzentrum (LRZ)). For all the test problems we set the mesh velocity to be equal to the solid phase velocity, which also represents the interface velocity according to our assumptions, see Section 3.2. Therefore, $\mathbf{V}=\mathbf{u}_1=\mathbf{u_I}$. Two-dimensional simulations are performed setting $M=4$, while we adopt $M=3$ for 3D runs. The meshes are automatically created by an external software by providing a characteristics mesh size denoted in the following by h.

The methodology of validation and verification involves the following test cases:

- 2D and 3D smooth non-trivial solutions of the BN model are simulated to assess the effective orders of accuracy of the numerical method. This test also proves that the *a posteriori* detection procedure is able to determine smooth solution, avoiding any unnecessary artificial limiting and its associated numerical dissipation. A convergence study is performed for different nominal order of accuracy (maximal polynomial degree of the reconstruction $M=d_{\max}$);

- two 1D Riemann problems solved on a 2D mesh (addressed as RP2 and RP3 according to [25]), involving simple separated waves generated by discontinuous initial data possibly enhanced by stiff source terms. These test cases can be found in [19, 25, 28] and their initial data are listed in Table 4.2. The exact solutions of these Riemann problems have been derived in [1, 19, 53] and are used in this work for comparison purposes;

- a 2D double Mach reflection problem which is designed for compressible two-phase flow. This test problem has been proposed in [28], starting from the original version of Woodward and Colella [61] for the compressible Euler equations. Here it becomes more complex as it involves reflection and refraction of shocks and contact with solid boundary conditions;

- two 3D spherical explosion problems based on the previous 1D Riemann problems are set for the compressible Baer-Nunziato model. They are called "Explosion problems" (EP). The first problem EP2 (according to the notation used in [25]) is the spherical version of RP2, whereas the second problem EP3 adopts a stiff interphase drag leading to non-trivial source terms. The reference solutions are obtained by

solving an equivalent non-conservative one-dimensional PDE in spherical geometry (with geometric reaction source terms) using a path-conservative second order TVD scheme [28] on an extremely fine *fixed* 1D mesh;

- as a last test we propose some quick profiling of the numerical code to show the amount of time spent in each main part of the algorithm.

4.1 Convergence studies

In this section a numerical convergence study is performed in two and three space dimensions. This test problem for the BN model can be found in [3, 25, 28, 31, 37]. First, an exact *stationary* and rotationally symmetric solution of the governing PDE is sought and then the problem is made *unsteady* by superimposing a constant, uniform velocity field $\bar{\mathbf{v}} = (\bar{u}, \bar{v})$ using the principle of Galilean invariance of Newtonian mechanics. The exact solution is then simply given by the advection of the non-trivial initial condition with the superimposed constant velocity field $\bar{\mathbf{v}}$. The rotationally symmetric solution is found by writing the governing BN equations (3.1) in polar coordinates and by imposing angular symmetry, see [28] for details. An ODE system in the radial coordinate only remains and it can be solved analytically. Using this manufactured analytical solution we are able to calculate the convergence rates when the mesh is successively refined. For the computational setup, we use the parameters from [25, 28]:

$$\gamma_1 = 1.4, \gamma_2 = 1.35, \pi_1 = \pi_2 = 0, \bar{u} = \bar{v} = 2, \mu = \lambda = 0, \rho_1 = 1, \rho_2 = 2, p_{10} = 1, p_{20} = \frac{3}{2}, s_1 = \frac{3}{2}, s_2 = \frac{7}{5}.$$

The computational domain is the square $\Omega = [-10;10] \times [-10;10]$ in 2D and $\Omega = [-10;10] \times [-10;10] \times [-5;5]$ in 3D discretized with triangular/tetrahedral meshes with periodic boundary conditions. The numerical convergence rates are computed for the solid volume fraction ϕ_s at the final time $t = 2.0$ on a sequence of successively refined meshes obtained by fixing a number of elements N_G along each direction. We simulate this problem with the ALE-MOOD-$\mathbb{P}_k$ schemes, $k = 2,3,4$ with *a posteriori* MOOD limiting acting. The results are listed in Table 4.1.

We observe that the schemes reach their designed order of accuracy even if the *a posteriori* stabilization process is active. In other words the MOOD procedure is able to detect that the numerical solution is smooth enough and do not need any limiting. Indeed, in this case the number of bad cells is exactly zero, meaning that the ALE scheme is run completely unlimited.

d	**ALE-MOOD**-$\mathbb{P}_2$			**ALE-MOOD**-$\mathbb{P}_3$			**ALE-MOOD**-$\mathbb{P}_4$		
	h	ϵ_{L_2}	$\mathcal{O}(L_2)$	h	ϵ_{L_2}	$\mathcal{O}(L_2)$	h	ϵ_{L_2}	$\mathcal{O}(L_2)$
2D	3.3188E-01	4.2739E-03	—	3.3291E-01	9.7121E-04	—	3.3413E-01	8.9462E-04	—
	2.5341E-01	1.8596E-03	3.1	2.5451E-01	3.1389E-04	4.2	2.5447E-01	2.2806E-04	5.0
	1.6911E-01	6.0825E-04	2.8	1.6917E-01	6.3690E-05	3.9	1.6918E-01	3.6773E-05	4.5
	1.2483E-01	2.5379E-04	2.9	1.2453E-01	1.8413E-05	4.1	1.2462E-01	9.2487E-06	4.5
	Expect.		3	Expect.		4	Expect.		5
3D	3.7135E-01	6.8289E-03	—	3.7057E-01	1.3726E-03	—	3.7081E-01	7.5658E-04	—
	2.3024E-01	1.9897E-03	2.6	2.3024E-01	2.6186E-04	3.5	2.3023E-01	1.0363E-04	4.2
	1.8053E-01	9.0159E-04	3.3	1.8066E-01	8.6867E-05	4.5	1.8066E-01	2.8278E-05	5.4
	Expect.		3	Expect.		4	Expect.		5

Table 2: Numerical convergence results for the compressible Baer–Nunziato model using the third to fifth order version of the Arbitrary–Lagrangian–Eulerian one–step finite volume schemes supplemented with *a posteriori* MOOD stabilization technique. The error norms refer to the variable ϕ_s (solid volume fraction) at time $t=2.0$. h refers to the characteristics mesh length.

	ρ_s	u_s	p_s	ρ_g	u_g	p_g	ϕ_s	t_f
RP2 [19]	$\gamma_s=3.0$,	$\pi_s=100$,	$\gamma_g=1.4$,	$\pi_g=0$,	$\lambda=\mu=0$			
L	800.0	0.0	500.0	1.5	0.0	2.0	0.4	0.10
R	1000.0	0.0	600.0	1.0	0.0	1.0	0.3	
RP3 [28]	$\gamma_s=1.4$,	$\pi_s=0$,	$\gamma_g=1.67$,	$\pi_g=0$,	$\lambda=10^3, \mu=10^2$			
L	1.0	0.0	1.0	1.0	0.0	1.0	0.99	0.2
R	0.125	0.0	0.1	0.125	0.0	0.1	0.01	

Table 3: Initial condition for the Riemann problems using the Baer-Nunziato model run in this paper. The left state (L) and the right state (R) are provided as well as model values for γ_k, π_k, λ,μ and the final time t_f.

4.2 Riemann problems

The initial computational domain is the box $\Omega(0)=[-0.5;0.5]\times[-0.05;0.05]$ and the initial discontinuity between the left state $\mathbf{Q}_L$ and the right state $\mathbf{Q}_R$ is located at $x_0=0$. The domain is paved with triangles using a characteristic mesh size of $h=1/200$, hence obtaining a total number of $N_E=8862$ elements. Periodic boundary conditions have been imposed in y direction, while we use transmissive boundaries along the x direction.

Friction and pressure relaxation are neglected in the first Riemann problem RP2, while for RP3 we use a stiff interphase drag $\lambda=10^3$ and pressure relaxation $\mu=10^2$. Please note that RP3 involves two almost pure ideal gases that differ in their value of γ. As done in [25, 28] the exact solution for RP3 is computed using the exact Riemann solver for the Euler equations of compressible gas dynamics [56] with two different values of γ on the left and on the right of the contact discontinuity, respectively.

The numerical results have been obtained using the fifth order version of our ALE finite volume schemes together with *a posteriori* MOOD stabilization procedure.

Figures 3-6 show a comparison between the reference solution and a one-dimensional cut through the reconstructed numerical solution $\mathbf{w}_h$. For RP3 we show the *mixture* density $\rho=\phi_s\rho_s+(1-\phi_s)\rho_g$ in Figure 6, whereas velocity and pressure distribution for the two phases are overlapping due to the stiff relaxation source terms. In all cases one can note a good agreement between numerical and reference solution. The material contact is well resolved in all cases and the overall accuracy is adequate remembering that the underlying scheme runs in a fifth order accurate mode on smooth parts of the flow and remains robust in the vicinity of discontinuities where it runs on a first order accurate basis.

In Figure 3 we have also provided the bad cell count (in percents) as a function of number of cycles. The blue line shows the percent of cells updated with the parachute first order scheme along with a polynomial fit, while the red line presents the percent of bad cells updated with the second order scheme. As expected, as the waves move away from the $x=0$ location more cells are detected as problematic and subsequently updated with a more dissipative scheme. Nonetheless the total number of troubled cells is at max-

imum around 30% and according to the last panel of this figure we can observe that these bad cells are distributed in the central part of the domain where all the discontinuities are concentrated.

A key feature of the MOOD approach is that it allows the user to be free to set and provide the main ingredients of the MOOD loop. Therefore, we also run the Riemann problem RP2 considering a *reduced* cascade of schemes, namely $\mathbb{P}_{d_{\max}} \rightarrow \mathbb{P}_1^{\text{LIM}}$. We consider a sequence of four refined meshes with characteristic mesh size of h=1/100, h=1/200, h=1/300 and h=1/400. Figure 4 depicts the associated cell order distribution for each mesh and we can notice that the location of the bad cells is closer to the discontinuities as the mesh is refined, still having some wrongly detected bad cells. Moreover, in Figure 5 we show that the percentage of cells updated with the parachute $\mathbb{P}_1^{\text{LIM}}$ scheme consistently decreases when the mesh is refined, from a value of $\approx$30% to $\approx$12%.

Figure 6 depicts the number of bad cells detected for each timestep and further updated with one of the scheme in the cascade. We observe that for this RP the total number of bad cells is around 15% of the total number of cells. Moreover the polynomial degrees used for the last timestep are shown in the bottom panel of this figure: due to the dissipation of the scheme we can see that the shock wave and the head of the rarefaction are particularly marked for decrementing whereas the rest of the domain only presents some randomly located troubled cells.

From the cell order distribution given in the bottom panel of Figures 3 and 6 one can note that shock and contact waves are not clearly marked and isolated by the detection criteria and this might be due to the switching from a high order to a low order scheme occurring across element boundaries. Moreover, in a smooth region close to a discontinuity we will inevitably produce a jump in the numerical reconstruction, between the cell marked as "problematic", which lies on the discontinuity, and the first cell on the plateau or on a smooth region, which is *correctly* not detected as problematic by the MOOD detection criteria. This phenomenon has already been observed in [38] in the context of high order ENO schemes and such a problem could be partially overcome by refining the computational mesh, as highlighted in Figure 4. A more fine grid allows the discontinuities to be located more precisely in our ALE context and they are better identified by the detection procedure of the MOOD approach, hence leading to a lower number of bad cells which are "better distributed" on the computational mesh, i.e. close to the shock/contact waves.

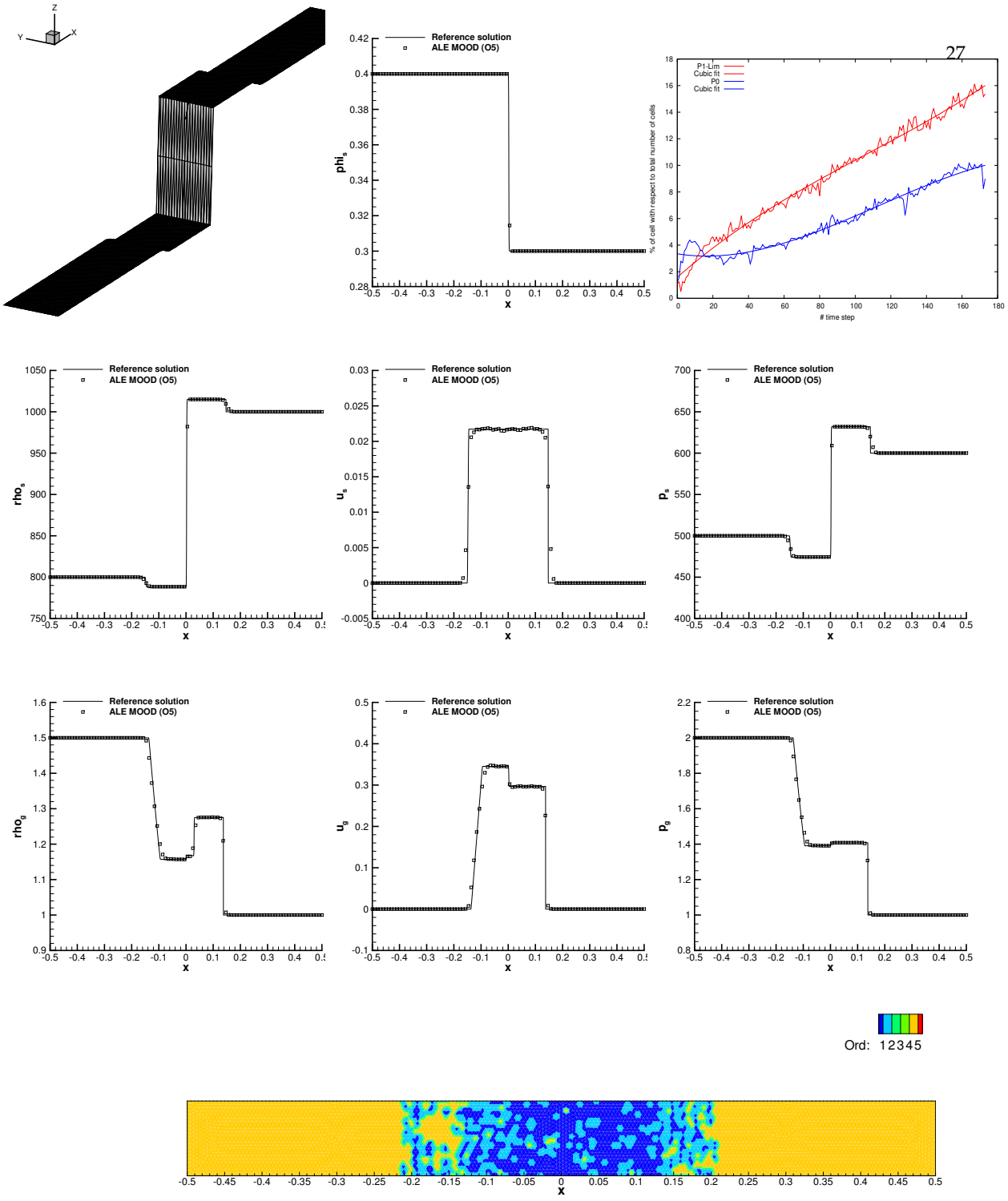

Figure 3: Fifth order numerical results for Riemann problem RP2 of the seven-equation Baer-Nunziato model at time $t = 0.1$ and comparison with the reference solution. From top-left to bottom-right: mesh, solid volume fraction, bad cell percents as a function of number of cycle (blue: first order, red: second order), solid variables (density, x component of the velocity, pressure) and gas variables. Bottom panel represents the polynomial degrees used for the last cycle by the MOOD procedure (blue cells are updated with a first order scheme while yellow with the unlimited 5th order).

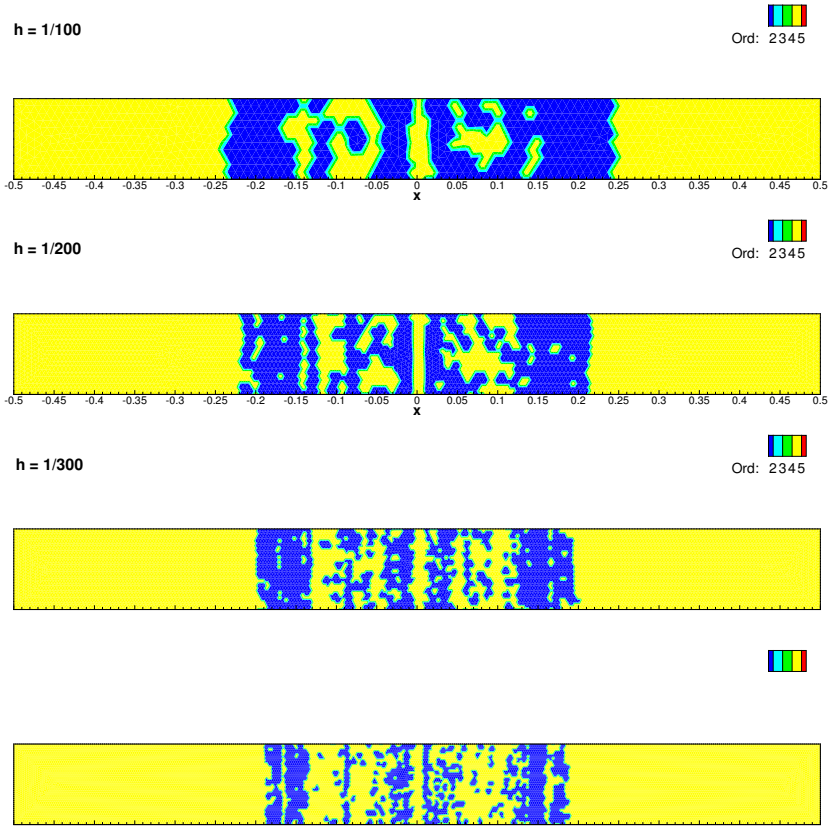

Figure 4: Cell polynomial order used for the last cycle by the MOOD procedure for Riemann problem RP2 solved with a fifth order accurate scheme on a sequence of refined unstructured meshes of characteristic mesh size h. From top to bottom: $h=1/100$, $h=1/200$, $h=1/300$ and $h=1/400$. The parachute scheme is chosen to be the $\mathbb{P}_1^{\mathrm{LIM}}$ method.

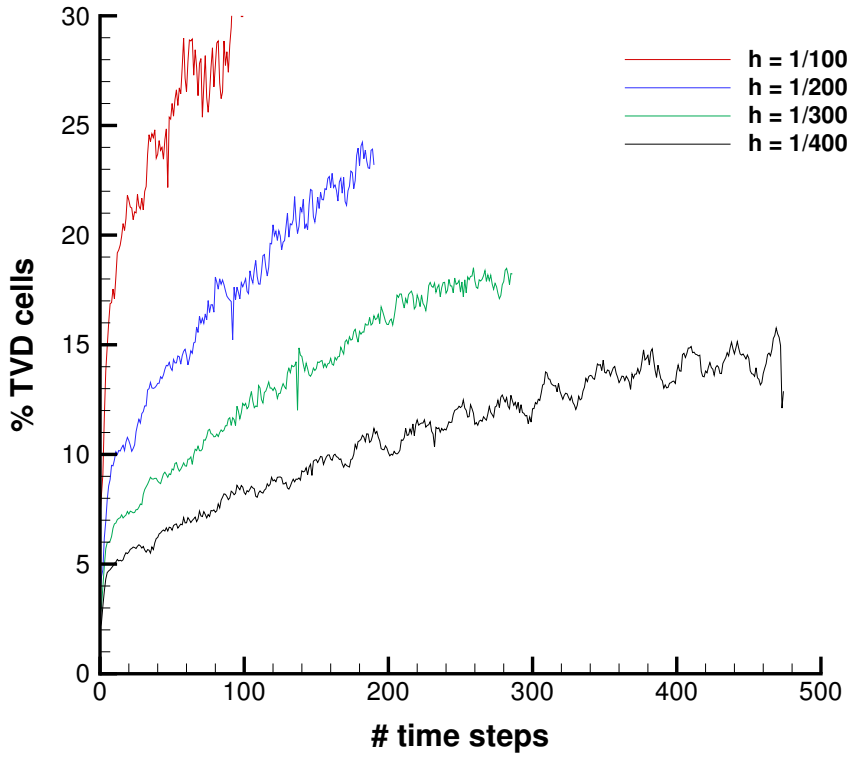

Figure 5: Percentage of cells updated with the parachute $\mathbb{P}_1^{\mathrm{LIM}}$ scheme for different characteristic mesh size ($h=1/100$, $h=1/200$, $h=1/300$ and $h=1/400$). The numerical results refer to Riemann problem RP2 run with degree $M=4$.

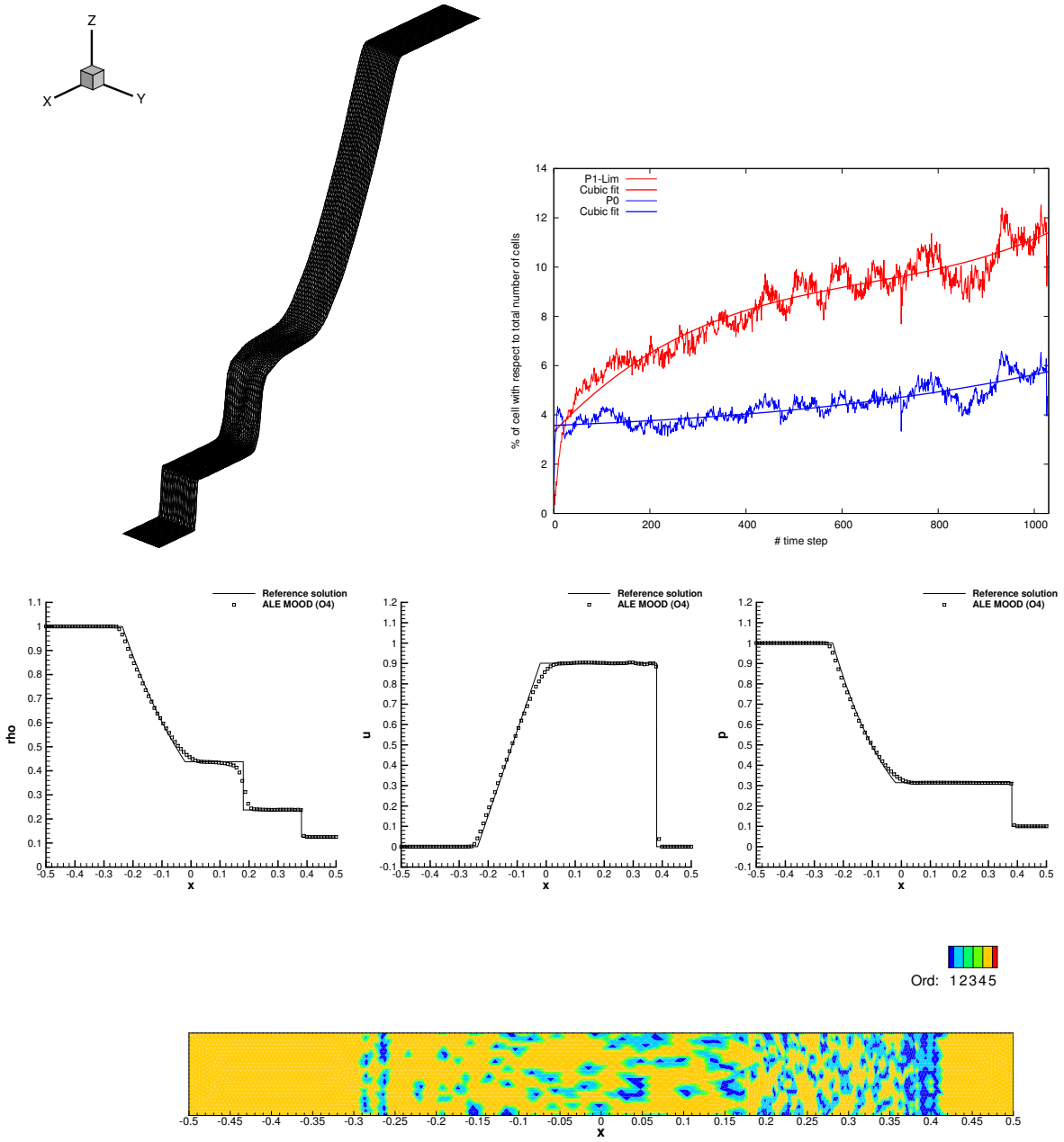

Figure 6: Fifth order numerical results for Riemann problem RP3 of the seven-equation Baer-Nunziato model with drag and pressure relaxation ($\lambda = 10^3, \mu = 10^2$) at time $t = 0.2$ and comparison with the reference solution. From top-left to bottom-right: mesh, bad cell percents as a function of number of cycle (blue: first order, red: second order), mixture density, mixture x component of the velocity and mixture pressure. Bottom panel represents the polynomial degrees used for the last cycle by the MOOD procedure (blue cells are updated with a first order scheme while yellow with the unlimited 5th order).

4.3 Double Mach reflection problem for compressible two phase flow

Here we run a double Mach reflection problem [61] adapted to compressible two phase flows [28]. It consists of a discontinuity initially located at $x = 0$ moving with velocity $\mathbf{v}_d = (10,0)$ to the right and hitting a 30^o wedge. The left state behind the discontinuity is calculated from the generalized Ranking Hugoniot conditions. With $\gamma_s = 3$, $\pi_s = 2$, $\gamma_g = 1.4$, $\pi_g = 0$, we obtain the following initial condition for the conserved variables in 2D: $\boldsymbol{W}_L = (0.4587156,2.087156,0,11.31078,4.285714,35.35714,0,301.875,0.25)$ and $\boldsymbol{W}_R = (0.25;0,0,0.875,0.75,0,0,1.3392857,0.25)$. The right state corresponds to $\rho_s = \rho_g = 1$ and $p_g = 1/\gamma_g$, $p_s = 1$. Transmissive boundary conditions are imposed in x-direction whereas we set solid wall conditions on the upper and lower boundaries. The final time is set to $t = 0.2$ and we use two different meshes with characteristic mesh spacing of $h = 1/50$ and $h = 1/100$, leading to a total number of triangles $N_E = 41456$ and $N_E = 167950$, respectively. Note that the meshes are on purpose chosen relatively coarse because a fifth order accurate ALE scheme is employed. Results are gathered in Figure 7 (polynomial degrees, gas and solid densities are displayed). We observe that the initial discontinuity moves with the correct speed, since it attains its correct final position $x = 2$. For the gas phase (undisturbed sound speed $c_g = 1$) we retrieve the typical flow structures present in the double Mach reflection problem at a Mach number of $M_g = 10$ [37, 61]. Contrarily, for the solid phase, the sound speed being larger $c_s = 3$, the Mach number $M_s = 10/3$ is lower. Hence, we only see the flow phenomena of a classical single Mach reflection. As already observed in [28], both phases completely decouple from each other.

One can also notice that the polynomial degrees are 0 or 1 in the vicinity of the discontinuities. From these plots we can truly see the structure of the main waves in the domain. Obviously when more cells are employed the numerical results do improve in quality. Nonetheless the detection capability of the scheme is not degradated when using a coarse mesh. Also a lot of cells are updated with the unlimited scheme (yellow cells) which justifies that the detection is parsimoniously and appropriately acting.

In Figure 8 we present the percentages of bad cells detected for each timestep for the two resolutions $h = 1/50$ on the left panel, and $h = 1/100$ on the right one. The number of troubled cells never exceeds 4% for second order cells and 3% for first order ones. On average only 5% of cells are detected and re-updated by our scheme for this problem. When the mesh resolution increases this number even drops.

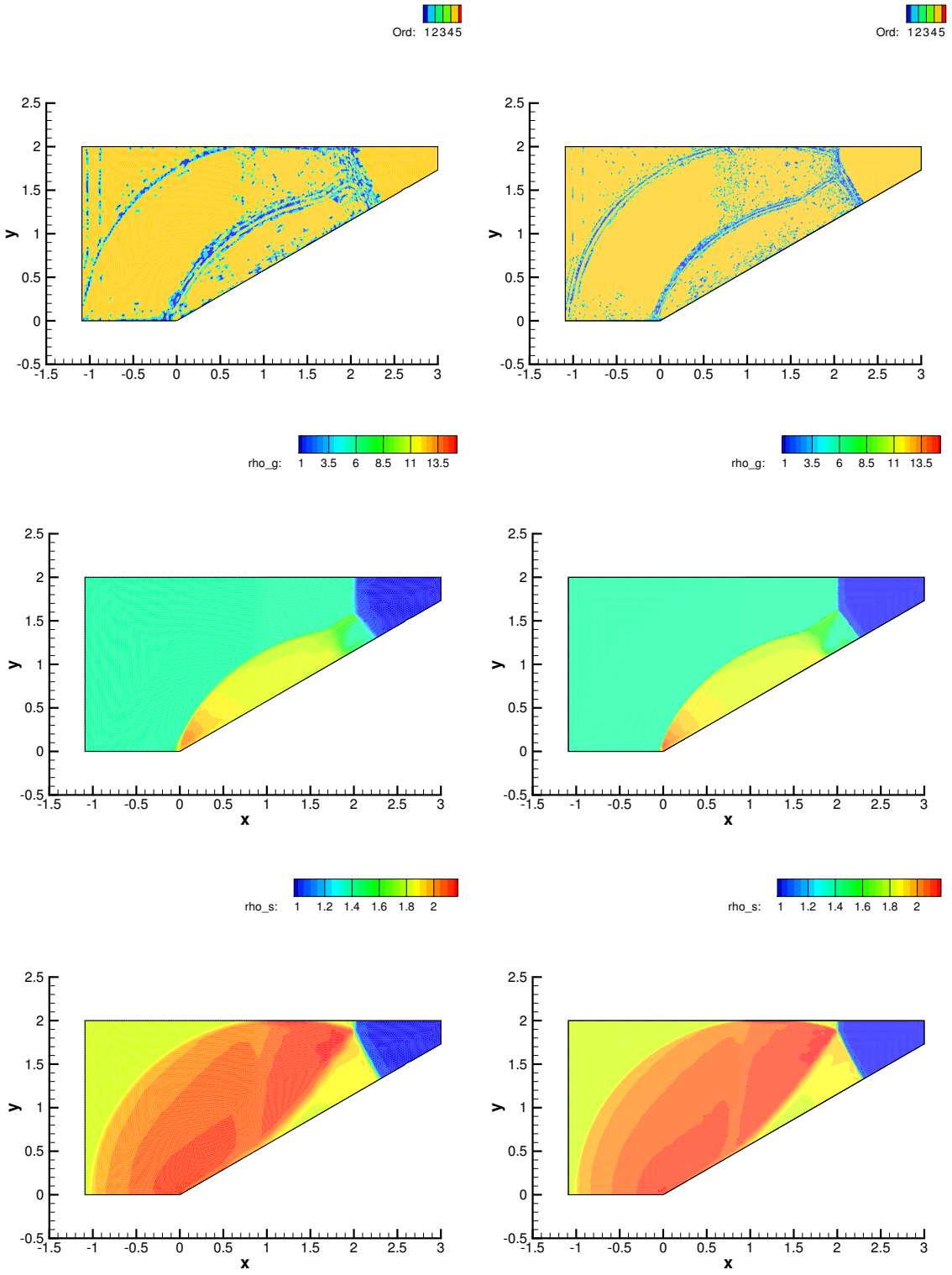

Figure 7: Double Mach reflection problem for compressible two phase flows for $h = 1/50$ (left panels) and $h = 1/100$ (right panels) at final time $t = 0.2$ — Results for the fifth order accurate ALE scheme with *a posteriori* MOOD stabilization — From top to bottom: polynomial degrees, gas density and solid density.

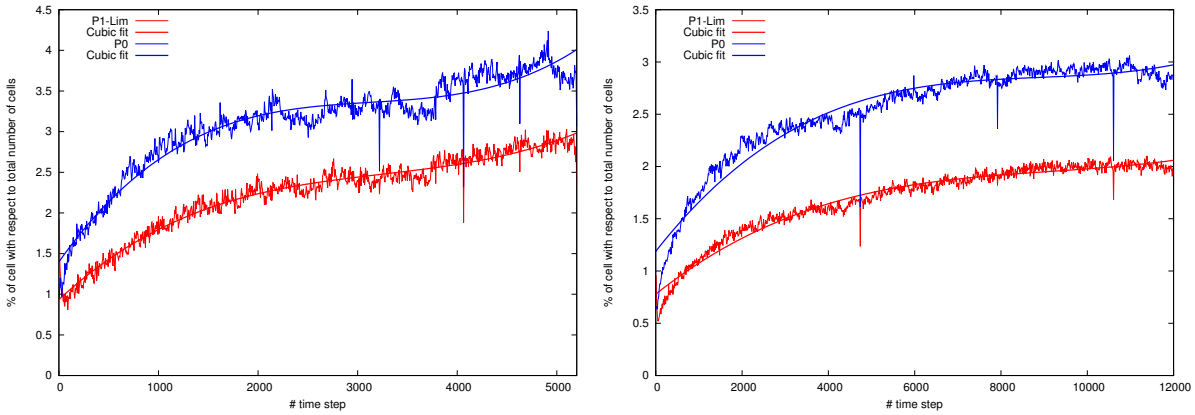

Figure 8: Double Mach reflection problem for compressible two phase flows for $h=1/50$ (left panel) and $h=1/100$ (right panel) at final time $t=0.2$ — Results for the fifth order accurate ALE scheme with *a posteriori* MOOD stabilization — Bad cell percents as a function of number of cycle (blue: first order, red: second order) and cubic fits (straight lines).

4.4 Spherical explosion problems

We use the same initial condition given for the Riemann problems listed in Table 4.2 to solve two different spherical explosion problems with the compressible Baer-Nunziato model (3.1). The computational domain $\Omega(0)$ is initially the sphere of radius $R = 0.9$, which has been discretized with a characteristic mesh size of $h = 1/100$ for $r \leq r_c$ and $h = 1/50$ for $r > r_c$ for a total number of tetrahedra of $N_E = 2632305$. $r = \sqrt{x^2+y^2+z^2}$ represents the generic radial position. In all cases the initial state $\mathbf{Q}(\mathbf{x},0)$ is assigned taking

$$\mathbf{Q}(\mathbf{x},0) = \begin{cases} \mathbf{Q}_i, & \text{if} \quad |\mathbf{x}| < r_c \\ \mathbf{Q}_o, & \text{else} \end{cases}, \tag{4.1}$$

with $r_c = 0.5$ representing the location of the initial discontinuity. The left state reported in Table 4.2 is assumed to be the inner state $\mathbf{Q}_i$, while the right state represents here the outer state $\mathbf{Q}_o$. In particular the initial condition of Riemann problem RP2 is used for the first explosion problem EP2. In the third explosion problem EP3 we use again the initial values of RP2 and we set $\lambda = 10^5$ and $\mu = 0$, hence adopting a stiff interphase drag. The final time is set to $t_f = 0.15$ for EP2, while $t_f = 0.18$ is used for EP3. Figures 9 - 10 show a comparison between the numerical results obtained with a fourth order ALE-MOOD scheme and the one-dimensional reference solution.

Since the mesh is moving with the interface velocity $\mathbf{u_I}$, i.e. $\mathbf{V} = \mathbf{u_I} = \mathbf{u}_1$, the contact discontinuity of the solid phase ϕ_1 is relatively well resolved in both cases. Along with classical comparison against exact/reference solution, we also display a cut in the sphere to observe the mesh and, more important, the polynomial degrees actually used for the last iteration. Moreover we present the percentage of bad cells detected as a function of the time iteration for the entire simulation (second panels of Figures 9 and 10)). For EP2 the number of bad cells is of the order 2 to 4 percents both for second order and first order cells, whereas it reaches 20% for second order updated cells and about 2% for first order ones. The relatively low number of bad cells justifies why testing an unlimited scheme as a first stage is an efficient strategy: more than 80% of the total number of cells do not need any limiting! On the remaining cells more time must be invested to properly stabilize the simulation, and here it is done by decrementing the polynomial degree of the reconstruction.

We would like to remark that the plots of the polynomial degrees used in the simulation (first panels of Figures 9 and 10) clearly have the imprint of the principal wave structure of the problem which states that the *a posteriori* detection procedure is performing well in observing discontinuities (and subsequently decrementing the polynomial degree of the reconstructions).

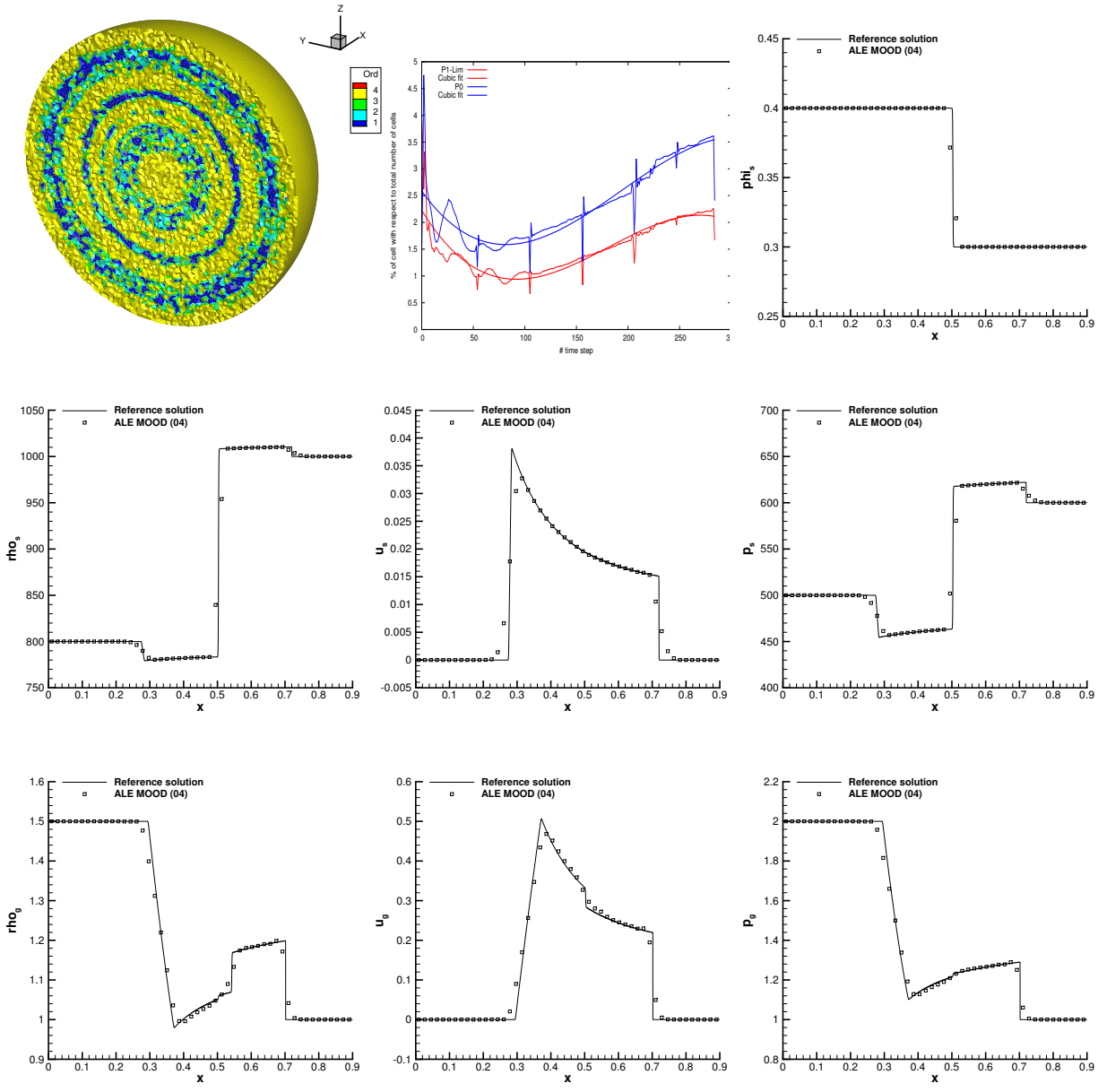

Figure 9: Fourth order numerical results for the 3D explosion problem EP2 of the seven-equation Baer-Nunziato model at time $t = 0.15$ and comparison with the reference solution.

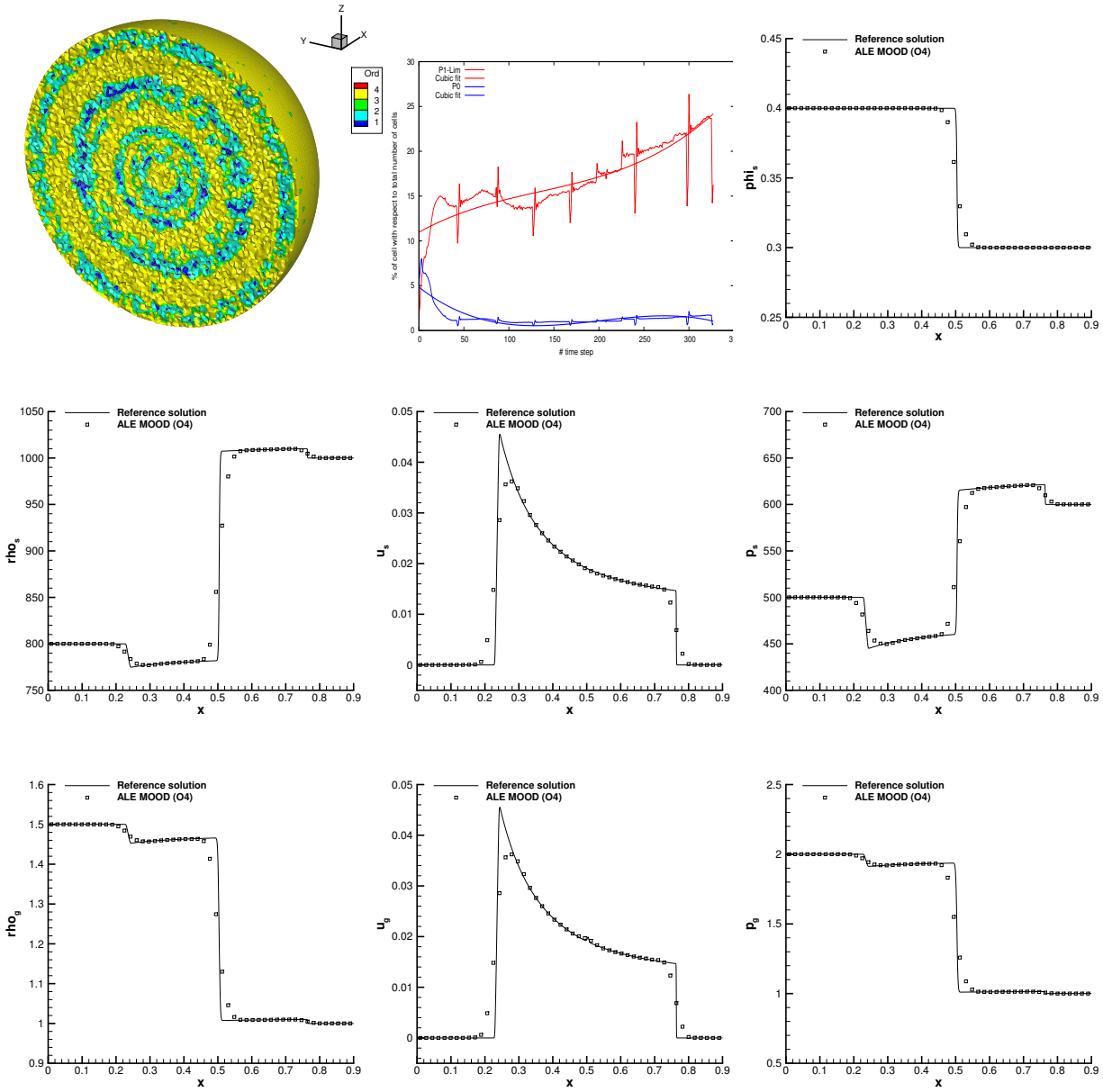

Figure 10: Fourth order numerical results for the 3D explosion problem EP3 with $\lambda = 10^5$ of the seven-equation Baer-Nunziato model at time $t = 0.18$ and comparison with the reference solution.

4.5 Profiling and CPU time consumption

We estimate the relative amount of time spent in the main components of the ALE-MOOD code for the RP3 test case which presents a lot of bad cells that need to be decremented and properly treated within the MOOD loop. It is obvious that the profiling of our *a posteriori* stabilized code tremendously depends both on the test case considered and on the technical implementation. If the isentropic vortex was chosen then no bad cell would ever been detected and the cost of the MOOD loop would be null. Here, on purpose, we choose the most demanding test of this paper in terms of number of bad cells. To evaluate the computational cost of the main components of the code, we have monitored the computational time used for the following parts:

1. the unlimited (first) iteration. This is the first iteration carried out using the unlimited polynomial reconstruction. If no problematic cells are detected, then this iteration terminates the timestep. The code is split here into three sub-steps:
   (a) *Polynomial reconstruction*: this is the unlimited polynomial reconstruction procedure illustrated in Section 2.1;
   (b) *Space-Time Prediction*: this corresponds to the algorithm described in Section 2.2;
   (c) *Finite Volume scheme*: we sum up the space-time flux integration presented in Section 2.4 and the mesh motion explained in Section 2.3.

   We will measure the time spent in each of these routines and the associated relative amount.

2. the subsequent MOOD procedure. The "Detection" step decides if problematic cells are present for each iteration. We closely monitor the amount of time of this procedure described in Section 3.3. If some problematic cells are detected, then some *Re-update* must be done. We globally measure the amount of time spent in the detection and decrementing processes as well as for re-updating those cells which have been marked as "problematic".

The third, fourth and fifth order accurate ALE-MOOD code is used to simulate the Riemann problem RP3 with the Baer-Nunziato model. We remark that RP3 involves stiff sources, hence requiring the local space-time predictor to use an implicit solver for the source term prediction which definitely is computationally expensive. The same unstructured mesh made of $N_E = 2246$ elements is used for the three runs. Results of this profiling are provided in Table 4. From this table (and the numerical results previously shown) we confirm that the number of troubled cells is below 30%, while the detection step consumes less than 1% of the total amount of the simulation.

We have also computed the number of degrees of freedom (d.o.f.) and the cost per cell, per degree of freedom per timestep according to

$$T_{\text{d.o.f.}} = T_{\text{CPU}} / N_{\text{d.o.f.}} / N_{\text{cycle}}, \tag{4.2}$$

	Ord.	Unlimited first iteration			MOOD procedure			T	N	T
		Pol.Recons.	*Predict.*	*FV scheme*	*Detect.*	*Re-upd.*	*Bad cells*	**CPU**	d.o.f.	/d.o.f.
ALE-MOOD	2	0.05 s	1.18 s	0.85 s	0.01 s	1.8 s	≃625	3.88 s	18	0.22 s
		1.3%	30.8%	22.3%	0.0%	45.6%	27.7%	100%		
		54.4%			45.6%					
	3	0.05 s	2.15 s	3.31 s	0.00 s	4.93 s	≃730	10.4 s	40	0.26 s
		0.5%	21.0%	32.7%	0.0%	45.8%	32.5%	100%		
		54.2%			45.8%					
	4	0.21 s	9.89 s	23.90 s	0.00 s	28.24 s	≃700	62.23 s	75	0.83 s
		0.3%	16.1%	39.2%	0.0%	44.4%	31.2%	100%		
		55.6%			44.4%					

Table 4: Profiling of the ALE-MOOD code solving the RP3 problem described above for the Baer-Nunziato system of equations. Horizontal lines: third, fourth and fifth order numerical scheme results. Vertical lines: percentage of total time for each procedure. The "Bad cells" column displays the percentage of bad cells detected on average for the entire simulation with respect to the total number of cells. (Note that the same cell can be detected several times per timestep.)

where the number of time steps is $N_{\text{cycle}} = 770$ for the simulation at order 2, 757 for order 3 and 736 for order 4.

In Figure 11 (top panels) we show the time spent in the unlimited scheme (sum of Reconstruction, Predictor, Finite Volume scheme) and in the MOOD loop (Detection, Re-update) in percent by respect to the total time and as a function of the time index. From left to right we display the results for the third, fourth and fifth order accurate schemes. From these plots we observe that the unlimited scheme covers about $50-60\%$ of the total cost. Consequently the stabilization of the simulation represents on average $40-50\%$ of the CPU time. This rather large cost is mainly due to the fact that

- this test has a relative large number of bad cell. Not only the troubled cells must be re-updated but also their direct neighbors, in order to have a consistency with the space-time flux integration along the space-time boundaries of each control volume. Another mild test would have less bad cells, leading to a relative lower stabilization cost;

- in our choice of implementation, no matter the polynomial degree of the reconstruction, all cells are updated with the highest integration rules needed for the unlimited scheme. In other words, if the fifth order scheme is used, then a bad cell is re-updated with a nominally fifth order accurate space-time integration, that is with a lot of unnecessary space-time integration points. Of course a "better implementation" would demand that only one space-time point is used for a first and second order accurate re-updated bad cell;

- in the BN model seven conservative variables are tested for NAD. This considerably increases the chances to detect a false bad cell. Moreover the presence of possible

seven discontinuous waves also makes those simulations extremely delicate. The detection procedure is consequently more strict, leading to more time spent in the stabilization procedures.

In Figure 11 (bottom panels) we present the same histograms but for each component of the code: Reconstruction, Predictor, Finite Volume (unlimited scheme) and Detection, Re-update (MOOD loop). From these plots one can note that the cost of the Detection and Reconstruction are negligible which might not be obvious at first glance. Finally, the Prediction step cost decreases with the accuracy of the scheme, while the Finite Volume update increases. This is clearly due to the fact that the higher the accuracy, the larger the number of integration points needed between neighbor cells. These points require a special treatment by the finite volume scheme (flux, Riemann solver, etc.). We must also remark that in the case of a non-conservative system with stiff source terms like the BN model, the treatment of the non-conservative products and the stiff sources are also computationally expensive.

We also perform the same study for the three-dimensional case, considering the spherical explosion problem EP3 which also involves a stiff relaxation source term. The computational domain is discretized using a total number of $N_E = 830713$ tetrahedra and we run this test case with orders $M = 2,3$. The final time of the simulation has been reached with a total number of time steps $N_{\text{cycle}} = 377$ for the third order and $N_{\text{cycle}} = 395$ for the fourth order of accuracy. Finally, the results are reported in Table 4.5, where one can note that in 3D most of the computational time is again required by the re-updating stage, consistently with the two space dimensions case. About 25% of the time is covered in the unlimited scheme by the numerical flux evaluation of the finite volume algorithm, hence explaining the high amount of time needed to re-update those cells which have been detected as problematic. Indeed, the higher the number of troubled cells, the higher the number of numerical fluxes that has to be computed and re-computed. Further research and work would be necessary to consider a quadrature-free approach, as the one suggested in [**?**], or design some other strategy in order to improve the efficiency of the ALE flux evaluation.

For the sake of clarity we have also performed the same simulations with the WENO ADER-ALE algorithm described in [25] and it turns out that, for such complex systems, the WENO scheme is faster and approximately two times more efficient, as clearly notable in Table 6. This result is in disagreement with what we observed in [**?**], where the ALE-MOOD code has proved to be much more efficient than the original WENO algorithm [25**?**]. Such a behavior is mainly due to the aforementioned reasons.

For comparison purposes we propose in Figure 12 the same diagnostics but when the two-dimensional hydrodynamics equations are solved and the classical Sod shock tube problem is simulated on the same mesh used previously. Because the hydrodynamics system of conservation laws has only four equations in 2D and neither source terms nor non-conservative products, the amount of time spent in each part of the code does change. For instance the amount of time spent in the reconstruction procedure is now

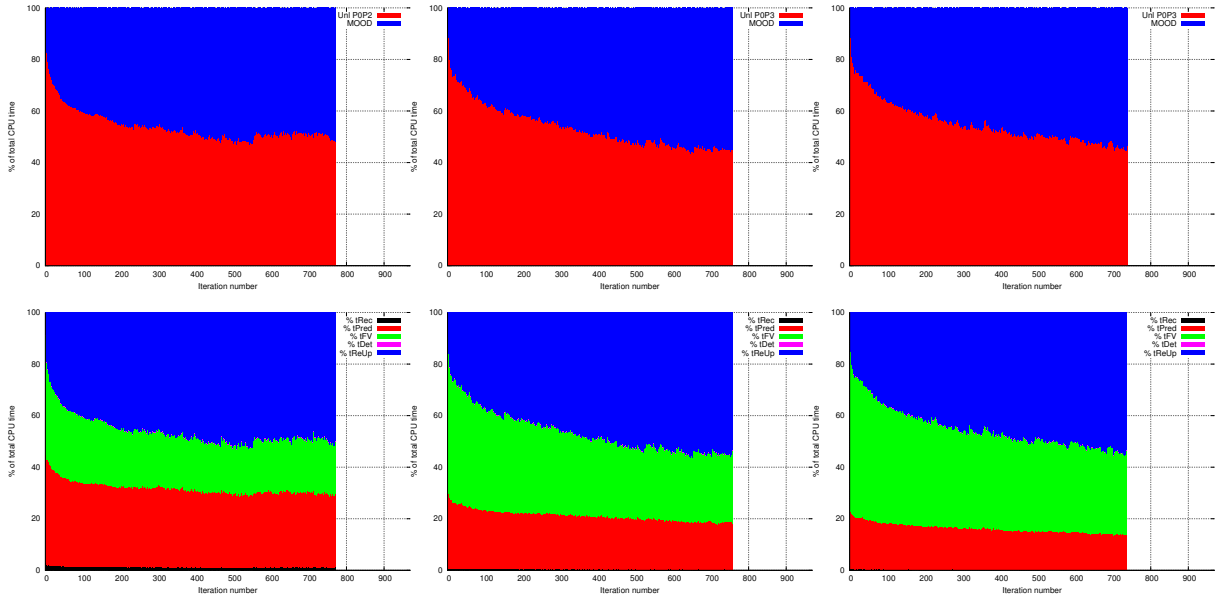

Figure 11: Profiling of the third, fourth and fifth order accurate ADER-ALE-MOOD codes from left to right for the BN model on the RP3 test case — Top line: Percent of the time spent in the unlimited scheme (sum of Reconstruction, Predictor, Finite Volume scheme) and in the MOOD loop (Detection, Re-update) — Bottom line Percent of the time spent in each component of the code : Reconstruction, Predictor, Finite Volume (unlimited scheme) and Detection, Re-update (MOOD loop) of bad cells — The abscissa represent the iteration index.

	Ord.	Unlimited first iteration			MOOD procedure			T	N	T
		Pol.Recons.	*Predict.*	*FV scheme*	*Detect.*	*Re-upd.*	*Bad cells*	**CPU**	d.o.f.	/d.o.f.
ALE-MOOD	2	0.05 s	1.70 s	2.70 s	0.00 s	6.8 s	≃115450	11.25 s	30	0.37 s
		0.5%	15.1%	24.1%	0.0%	60.3%	13.9%	100%		
		39.7%			60.3%					
	3	0.27 s	6.30 s	29.69 s	0.00 s	58.53 s	≃171125	94.79 s	80	1.18 s
		0.3%	6.6%	31.4%	0.0%	61.7%	20.6%	100%		
		38.3%			61.7%					

Table 5: Profiling of the ALE-MOOD code solving the EP3 problem described above for the Baer-Nunziato system of equations. Horizontal lines: third and fourth order numerical scheme results. Vertical lines: percentage of total time for each procedure. The "Bad cells" column displays the percentage of bad cells detected on average for the entire simulation with respect to the total number of cells. (Note that the same cell can be detected several times per timestep.)

		2D - RP3			3D - EP3		
ALE-WENO	Ord.	T **CPU**	N d.o.f.	T /d.o.f.	T **CPU**	N d.o.f.	T /d.o.f.
	2	1.74 s	18	0.10 s	4.05 s	30	0.14 s
	3	5.75 s	40	0.14 s	35.88 s	80	0.45 s
	4	29.25 s	75	0.39 s	-	-	-

Table 6: Profiling of the ALE-WENO code solving the test problems RP3 in 2D and EP3 in 3D described above for the Baer-Nunziato system of equations. We show the total computational time per timestep, the number of degrees of freedom and the computational time needed to update each degree of freedom per timestep. As done for the ALE-MOOD code, we consider $M=2,3,4$.

not negligible anymore (about 20%) and it increases with a high order of accuracy as expected. The relative amount of time spent in re-updating the bad cells drops to 30% on average. This is also not negligible because the space-time integration is still performed with high accuracy as for the BN model.

5 Conclusion and Perspectives

This paper follows [**?**] in which the novel *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) approach was applied to the context of direct Arbitrary-Lagrangian-Eulerian (ALE) ADER schemes solving the hydrodynamics equations in multi-dimensions on moving meshes. In this work we have shown how to extend such ALE-ADER-MOOD approach to different systems of equations involving non-conservative products and stiff source terms. The candidate model is the seven-equation Baer-Nunziato model of multi-phase flows with inter-phase drag and pressure relaxation terms.

Within the unlimited ALE-ADER-WENO scheme designed in [**?**] to solve 3D hydrodynamics problems on unstructured simplicial grids, we have substituted the MOOD paradigm [13, 21, 22, 43] to the *a priori* WENO polynomial limiting technique.

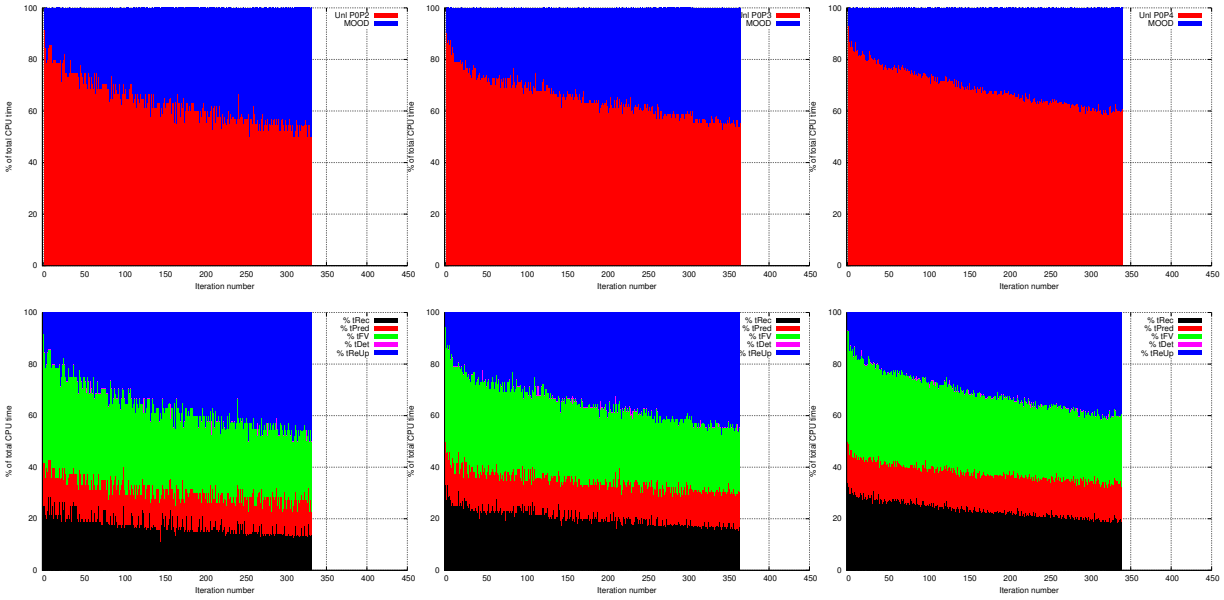

Figure 12: Profiling of the third, fourth and fifth order accurate ADER-ALE-MOOD codes from left to right for the hydrodynamics model on the Sod problem — Top line: Percent of the time spent in the unlimited scheme (sum of Reconstruction, Predictor, Finite Volume scheme) and in the MOOD loop (Detection, Re-update) — Bottom line Percent of the time spent in each component of the code : Reconstruction, Predictor, Finite Volume (unlimited scheme) and Detection, Re-update (MOOD loop) of bad cells — The abscissa represent the iteration index.

The ALE-MOOD scheme has been implemented within a 2D/3D parallel MPI code dedicated to solve hyperbolic PDE on fixed and moving unstructured grids, see [**?** **?**]. Some modifications have been designed to take into account the presence of non-conservative products and source terms. A test suite in 2D and 3D has been simulated and results are reported in this work. Generally speaking the ALE-MOOD numerical method produces accurate results on these complex systems of equations. An evolution of the number of problematic cells is systematically provided in order to visualize the extra-work brought by the use of the MOOD paradigm. The numerical results presented in this paper indicate that the high order direct ADER-ALE finite volume schemes supplemented with *a posteriori* MOOD stabilization paradigm constitute a valid numerical method to solve complex hyperbolic PDE systems with nonconservative products and stiff source terms in multiple space dimensions on unstructured meshes on parallel machines.

Moreover we have provided a crude but representative profiling of the overall numerical method to estimate the relative cost of the main components of the code. The new approach is less efficient than the original WENO formulation [**?** **?**], although it is easier to code and permits to save memory consumption because it only needs one reconstruction stencil in two and three space dimensions, see [**?**] for details. Therefore in the near future we plan to optimize further the Re-update component of the MOOD loop by adapting the space-time integration to the polynomial degree chosen or even developing a quadrature-free integration as done in [**?**]. We also plan to employ the ADER-ALE-MOOD machinery to deal with astrophysical and Inertial Confinement Fusion (ICF) problems for which highly accurate moving mesh techniques may become mandatory when the computational domain may drastically change its size.

Acknowledgments

W.B. has 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. R.L. has been partially funded by the ANR under the JCJC project "ALE INC(ubator) 3D" JS01-012-01 and the "International Centre for Mathematics and Computer Science in Toulouse" (CIMI) partially supported by ANR-11-LABX-0040-CIMI within the program ANR-11-IDEX-0002-02. The authors would like to acknowledge PRACE for awarding access to the SuperMUC supercomputer based in Munich, Germany at the Leibniz Rechenzentrum (LRZ). Parts of the material contained in this work have been elaborated, gathered and tested while W.B. visited the Mathematical Institute of Toulouse for three months and R.L. visited the Dipartimento di Ingegneria Civile Ambientale e Meccanica in Trento for three months.

The authors would also like to thank the two anonymous referees for their suggestions and comments, that helped to improve the quality and clarity of this article.

References

[1] N. Andrianov and G. Warnecke. The riemann problem for the baer-nunziato two-phase flow model. *Journal of Computational Physics*, 212:434–464, 2004. (Cited page 22.)

[2] M.R. Baer and J.W. Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (DDT) in reactive granular materials. *J. Multiphase Flow*, 12:861–889, 1986. (Cited page 2, 17 et 19.)

[3] D. Balsara. Second-order accurate schemes for magnetohydrodynamics with divergence-free reconstruction. *The Astrophysical Journal Supplement Series*, 151:149–184, 2004. (Cited page 23.)

[4] D. Balsara and C.W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000. (Cited page 20.)

[5] D.S. Balsara. Self-adjusting, positivity preserving high order schemes for hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231:75047517, 2011. (Cited page 15.)

[6] T.J. Barth and P.O. Frederickson. Higher order solution of the euler equations on unstructured grids using quadratic reconstruction. *28th Aerospace Sciences Meeting*, pages AIAA paper no. 90–0013, January 1990. (Cited page 4.)

[7] 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. (Cited page 21.)

[8] W. Boscheri and M. Dumbser. Arbitrary–Lagrangian–Eulerian One–Step WENO Finite Volume Schemes on Unstructured Triangular Meshes. *Communications in Computational Physics*, 14:1174–1206, 2013. (Cited page 5, 6, 9, 10, 13 et 15.)

[9] Walter Boscheri and Michael Dumbser. A direct arbitrary-lagrangian-eulerian ader-weno finite volume scheme on unstructured tetrahedral meshes for conservative and non-conservative hyperbolic systems in 3d. *Journal of Computational Physics*, 275(0):484 – 523, 2014.

[10] Walter Boscheri, Raphaël Loubère, and Michael Dumbser. Multi-dimensional direct arbitrary-lagrangian-eulerian high accurate ader-mood finite volume schemes. *Journal of Computational Physics*, 275(0):484 – 523, 2015. submitted.

[11] M.J. Castro, J.M. Gallardo, J.A. López, and C. Parés. Well-balanced high order extensions of godunov's method for semilinear balance laws. *SIAM Journal of Numerical Analysis*, 46:1012–1039, 2008. (Cited page 11.)

[12] M.J. Castro, J.M. Gallardo, and C. Parés. High-order finite volume schemes based on reconstruction of states for solving hyperbolic systems with nonconservative products. applications to shallow-water systems. *Mathematics of Computation*, 75:1103–1134, 2006. (Cited page 11.)

[13] 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. (Cited page 2, 16, 17 et 41.)

[14] S. Clain, S. Diot, and R. Loubère. Multi-dimensional optimal order detection (mood) a very high-order finite volume scheme for conservation laws on unstructured meshes. In Fort Fürst Halama Herbin Hubert (Eds.), editor, *FVCA 6, International Symposium, Prague, June 6-10*, volume 4 of *Series: Springer Proceedings in Mathematics*, 2011. 1st Edition. XVII, 1065 p. 106 illus. in color. (Cited page 16 et 17.)

[15] S. Clain and G. Machado. A very high-order finite volume method for the one-dimensional time-dependent convection-diffusion problem. *Computers and Mathematics with Applications*, 2014. in press. (Cited page 16.)

[16] Stéphane Clain, Gaspar J Machado, JM Nóbrega, and RMS Pereira. A sixth-order finite volume method for multidomain convection–diffusion problem with discontinuous coefficients. *Computer Methods in Applied Mechanics and Engineering*, 267:43–64, 2013. (Cited page 16.)

[17] B. Cockburn, G. E. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods*. Lecture Notes in Computational Science and Engineering. Springer, 2000. (Cited page 5.)

[18] P. Colella and M.D. Sekora. A limiter for PPM that preserves accuracy at smooth extrema. *Journal of Computational Physics*, 227:7069–7076, 2008. (Cited page 20.)

[19] V. Deledicque and M.V. Papalexandris. An exact riemann solver for compressible two-phase flow models containing non-conservative products. *Journal of Computational Physics*, 222:217–245, 2007. (Cited page 22 et 25.)

[20] B. Després and C. Mazeran. Lagrangian gas dynamics in two-dimensions and Lagrangian systems. *Archive for Rational Mechanics and Analysis*, 178:327–372, 2005. (Cited page 10.)

[21] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. *Computers and Fluids*, 64:43 – 63, 2012. (Cited page 2, 16, 17, 20, 21 et 41.)

[22] S. Diot, R. Loubère, and S. Clain. The MOOD method in the three-dimensional case: Very-high-order finite volume method for hyperbolic systems. *International Journal of Numerical Methods in Fluids*, 73:362–392, 2013. (Cited page 2, 16, 17, 20, 21 et 41.)

[23] M. Dubiner. Spectral methods on triangles and other domains. *Journal of Scientific Computing*, 6:345–390, 1991. (Cited page 5.)

[24] 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. (Cited page 6 et 9.)

[25] M. Dumbser and W. Boscheri. High-order unstructured Lagrangian one–step WENO finite volume schemes for non–conservative hyperbolic systems: Applications to compressible multi–phase flows. *Computers and Fluids*, 86:405 – 432, 2013. (Cited page 4, 5, 6, 9, 10, 11, 14, 15, 22, 23, 25 et 39.)

[26] M. Dumbser, M. Castro, C. Parés, and E.F. Toro. ADER schemes on unstructured meshes for non-conservative hyperbolic systems: Applications to geophysical flows. *Computers and Fluids*, 38:1731–1748, 2009. (Cited page 11.)

[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. (Cited page 6 et 9.)

[28] M. Dumbser, A. Hidalgo, M. Castro, C. Parés, and E.F. Toro. FORCE schemes on unstructured meshes II: Non–conservative hyperbolic systems. *Computer Methods in Applied Mechanics and Engineering*, 199:625–647, 2010. (Cited page 6, 11, 22, 23, 25 et 31.)

[29] M. Dumbser, M. Kaeser, V.A. Titarev, and E.F. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204 – 243, 2007. (Cited page 4 et 15.)

[30] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007. (Cited page 4, 5 et 15.)

[31] M. Dumbser and E. F. Toro. A simple extension of the Osher Riemann solver to non-conservative hyperbolic systems. *Journal of Scientific Computing*, 48:70–88, 2011. (Cited page 11 et 23.)

[32] M. Dumbser, A. Uuriintsetseg, and O. Zanotti. On Arbitrary–Lagrangian–Eulerian One–Step WENO Schemes for Stiff Hyperbolic Balance Laws. *Communications in Computational Physics*, 14:301–327, 2013. (Cited page 6, 9 et 13.)

[33] M. Dumbser and O. Zanotti. Very high order PNPM schemes on unstructured meshes for the resistive relativistic MHD equations. *Journal of Computational Physics*, 228:6991–7006, 2009. (Cited page 6 et 9.)

[34] Michael Dumbser and Walter Boscheri. High-order unstructured lagrangian one-step {WENO} finite volume schemes for non-conservative hyperbolic systems: Applications to compressible multi-phase flows. *Computers & Fluids*, 86(0):405 – 432, 2013.

[35] 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. (Cited page 10.)

[36] A. Hidalgo and M. Dumbser. ADER schemes for nonlinear systems of stiff advection-diffusion-reaction equations. *Journal of Scientific Computing*, 48:173–189, 2011. (Cited page 6 et 9.)

[37] C. Hu and C.W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999. (Cited page 15, 23 et 31.)

[38] G.-S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996. (Cited page 15 et 26.)

[39] A.K. Kapila, R. Menikoff, J.B. Bdzil, S.F. Son, and D.S. Stewart. Two-phase modelling of DDT in granular materials: reduced equations. *Physics of Fluids*, 13:3002–3024, 2001. (Cited page 19.)

[40] G. E. Karniadakis and S. J. Sherwin. *Spectral/hp Element Methods in CFD*. Oxford University Press, 1999. (Cited page 5.)

[41] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486–508, 2005. (Cited page 4 et 15.)

[42] P.M. Knupp. Achieving finite element mesh quality via optimization of the jacobian matrix norm and associated quantities. part ii – a framework for volume mesh optimization and the condition number of the jacobian matrix. *Int. J. Numer. Meth. Engng.*, 48:1165 – 1185, 2000. (Cited page 10.)

[43] 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. (Cited page 2, 5, 16, 17, 21 et 41.)

[44] Raphaël Loubère, Pierre-Henri Maire, Mikhail Shashkov, Jérôme Breil, and Stéphane Galera. Reale: A reconnection-based Arbitrary-Lagrangian-Eulerian method. *J. Comput. Phys.*, 229(12):4724 – 4761, 2010. (Cited page 10.)

[45] 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. (Cited page 10, 14 et 15.)

[46] 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. (Cited page 14.)

[47] C. Olliver-Gooch and M. Van Altena. A high-order–accurate unstructured mesh finite–volume scheme for the advection–diffusion equation. *Journal of Computational Physics*, 181:729 – 752, 2002. (Cited page 4.)

[48] C. Parés. Numerical methods for nonconservative hyperbolic systems: a theoretical framework. *SIAM Journal on Numerical Analysis*, 44:300–321, 2006. (Cited page 11.)

[49] S. Rhebergen, O. Bokhove, and J.J.W. van der Vegt. Discontinuous Galerkin finite element methods for hyperbolic nonconservative partial differential equations. *Journal of Computational Physics*, 227:1887–1922, 2008. (Cited page 11.)

[50] Stéphane Clain Ricardo Costa, Gaspar J. Machado. Sixth-order finite volume method for the 1d biharmonic operator: application to the intramedullary nail simulation. *International Journal of Applied Mathematics and Computer Science (AMCS)*, 2014. (Cited page 16.)

[51] R. Saurel and R. Abgrall. A multiphase godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics*, 150:425–467, 1999. (Cited page 19.)

[52] R. Saurel, S. Gavrilyuk, and F. Renaud. A multiphase model with internal degrees of freedom: Application to shock-bubble interaction. *Journal of Fluid Mechanics*, 495:283–321, 2003. (Cited page 19.)

[53] D.W. Schwendeman, C.W. Wahle, and A.K. Kapila. The riemann problem and a high-resolution godunov method for a model of compressible two-phase flow. *Journal of Computational Physics*, 212:490–526, 2006. (Cited page 22.)

[54] A.H. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice-Hall Inc., Englewood Cliffs, New Jersey, 1971. (Cited page 8 et 13.)

[55] A. Suresh and H.T. Huynh. Accurate monotonicity-preserving schemes with runge-kutta time stepping. *Journal of Computational Physics*, 136:83–99, 1997. (Cited page 20.)

[56] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer, second edition, 1999. (Cited page 25.)

[57] I. Toumi. A weak formulation of roes approximate riemann solver. *Journal of Computational Physics*, 102:360–373, 1992. (Cited page 11.)

[58] V.Desveaux. *Contribution à l'approximation numérique des systèmes hyperboliques*. PhD thesis, Université de Nantes, 2013. (Cited page 16.)

[59] V.Desveaux and C.Berthon. An entropic mood scheme for the euler equations. *International Journal of Finite Volumes*, 2013. (Cited page 16.)

[60] Alan M. Winslow. Numerical solution of the quasilinear poisson equation in a nonuniform triangle mesh. *J. Comput. Phys.*, 135(2):128–138, August 1997. (Cited page 10.)

[61] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984. (Cited page 22 et 31.)

[62] 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. *J. Comput. Phys.*, 278:47–75, 2014. (Cited page 16.)

[63] Olindo Zanotti, Francesco Fambri, Michael Dumbser, and Arturo Hidalgo. Space-time adaptive {ADER} discontinuous galerkin finite element schemes with a posteriori sub-cell finite volume limiting. *Computers & Fluids*, 118(0):204 – 224, 2015. (Cited page 16.)