



# Adaptive-Mesh-Refinement for hyperbolic systems of conservation laws based on a posteriori stabilized high order polynomial reconstructions

Matteo Semplice, Raphaël Loubère

## ► To cite this version:

Matteo Semplice, Raphaël Loubère. Adaptive-Mesh-Refinement for hyperbolic systems of conservation laws based on a posteriori stabilized high order polynomial reconstructions. *Journal of Computational Physics*, Elsevier, 2018. hal-01485587

HAL Id: hal-01485587

<https://hal.archives-ouvertes.fr/hal-01485587>

Submitted on 9 Mar 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Adaptive-Mesh-Refinement for hyperbolic systems of conservation laws based on *a posteriori* stabilized high order polynomial reconstructions

Matteo Semplice<sup>a</sup>, Raphaël Loubère<sup>\*b</sup>

<sup>a</sup>Dipartimento di Matematica G. Peano Università di Torino Via C. Alberto, 10 10123 Torino, Italy

<sup>b</sup>CNRS and Institut de Mathématiques de Bordeaux (IMB) Université de Bordeaux, France

---

## Abstract

In this paper we propose a third order accurate finite volume scheme based on polynomial reconstruction along with *a posteriori* limiting within an Adaptive-Mesh-Refinement (AMR) simulation code for hydrodynamics equations in 2D. The *a posteriori* limiting is based on the detection of problematic cells on a so-called candidate solution computed at each stage of a third order Runge-Kutta scheme. Such detection may include different properties, derived from physics, such as positivity, from numerics, such as a non-oscillatory behavior, or from computer requirements such as the absence of NaN's. Troubled cell values are discarded and re-computed starting again from the previous time-step using this time a more dissipating scheme but only locally to these cells. By decrementing the degree of the polynomial reconstructions from 2 to 0 we switch from a third-order to a first-order accurate scheme. Ultimately some troubled cells may be updated with a first order accurate scheme for instance close to steep gradients. The entropy indicator sensor is used to refine/coarsen the mesh. This sensor is also employed in an *a posteriori* manner because if some refinement is needed at the end of a time step, then the current time-step is recomputed but only locally with such refined mesh. We show on a large set of numerical tests that this *a posteriori* limiting procedure coupled with the entropy-based AMR technology not only can maintain optimal accuracy on smooth flows but also stability on discontinuous profiles such as shock waves, contacts, interfaces, etc. Moreover numerical evidences show that this approach is comparable in terms of accuracy and cost to a more classical CWENO approach within the same AMR context.

**Key words:** Adaptive Mesh Refinement, *a posteriori* limiter, MOOD paradigm, High order of accuracy in space and time, Entropy production, Hyperbolic conservation laws, Hydrodynamics, h/p adaptation.

---

## 1. Introduction

The quest for developing reliable, robust, accurate and efficient simulation codes for general hyperbolic systems of conservation laws is a work in progress since the very first simulations on computers around 1942, along with the first implemented numerical methods. Efficient simulations are based on several building-blocks or paradigms.

The first one, and, presumably, the most obvious one, is the development of robust and accurate conservative numerical methods based on the seminal works of Lax and Wendoff [LW60, LW64], Godunov [S.K59] and many others. There exist highly popular numerical schemes for time and space discretizations

---

<sup>\*</sup>Corresponding author

Email addresses: [matteo.semplice@unito.it](mailto:matteo.semplice@unito.it) (Matteo Semplice), [raphael.loubere@math.u-bordeaux.fr](mailto:raphael.loubere@math.u-bordeaux.fr) (Raphaël Loubère\*)

and, it is a matter of taste to determine the more appropriate one for a given situation. In this work we have chosen a third-order accurate Runge-Kutta time integration scheme, and, third-order accurate polynomial reconstruction with local Lax-Friedrichs approximate Riemann solver: this scheme assures a nominal third-order of accuracy. What remains to design is a valid strategy of limiting to ensure robustness and an essentially non-oscillatory behavior. In fact, any scheme of high accuracy inexorably develops spurious oscillations which must be damped by locally reducing the accuracy of the scheme to first order in the vicinity of any discontinuity. Most of these damping techniques rely on some sort of artificial viscosity, should it be described in a von Neumann and Richtmyer fashion [NR50], in terms of Riemann solvers [Tor99], (slope) limiters [van74, van79, Ven95, MB05, Kol10] or stabilization techniques [HST10]. All have in common the desire to detect within a solution where the numerical oscillations are or may appear, —in others words, to locate where some artificial viscosity or numerical damping is needed—, and, how much of dissipation is sufficient to avoid the growth of spurious oscillations. All of the above-mentioned techniques, more or less, answer these two questions, sometimes independently. In this work, we rely on the *a posteriori* MOOD paradigm which was developed initially in [CDL11a, DCL12, DLC13] and further extended to different contexts for instance in [LDD14, ZDLS14, VC13, ZFDH15a, ZFDH15b, NRCL16]. This *a posteriori* procedure checks at the end of the timestep for troubled cell, and further recomputes them with a more dissipative scheme. This procedure allows to maintain the physical admissibility of the solution, preserves an essentially-non-oscillatory behavior and even cures pathological behavior (like unrepresentable numbers such as NaN's and Inf's).

The resolution of any numerical simulation depends on the mesh (and the degrees of freedom) employed and on the ability of the numerical method to employ the best possible accuracy (high on smooth flow regions, low close to discontinuities). Thus, if the first ingredient to perform accurate simulations is the numerical method, the second one is certainly the proper use of the available computer resources especially on large meshes. While we tend to separate them, clearly, both ingredients are not uncorrelated from each other. The first possibility relies on parallel computing using vectorization, recent hybrid architectures (CPU/GPU), light parallelization with shared memory (OpenMP), or massively parallel machinery with distributed memory (MPI) for instance. Parallel computing is a genuine efficient way to improve even further the resolution of numerical simulations.

A second possibility relies on redirecting the available resolution resources (i.e. the computational cells, the degrees of freedom) to important locations of the simulation. Since this technique can be later coupled with parallelization, we focus on it for the purposes of this paper. Many different solutions have been explored since the 70's, which fall into two main classes: *adaptive mesh redistribution* and *adaptive mesh refinement* (AMR). The redistribution implies that the code runs with a fixed number of cells but they are continuously and automatically relocated following the flow (Lagrangian or Arbitrary-Lagrangian-Eulerian schemes (ALE) [HAC74, Mai09, Lou13]) or manually (Moving Mesh techniques [TT03]). Some techniques allow local changes of connectivity [LMS<sup>+</sup>10, Pin11], or, simply, emancipate from the strict notion of mesh, like for particle methods such as the Smoothed Particle Hydrodynamics (SPH) method [NRCL16]. Contrarily, the refinement strategy adds new cells where appropriate and removes old cells which are no longer required. This technique is known under the acronym AMR and many improvements and publications are available since the 80's [BO84, BC89]. A drastic reduction of the costs without sacrificing the level of accuracy is the main reason why those techniques were developed and are still in use today [SCR15]. In this work we rely on AMR technology using the so-called numerical entropy production refinement criterion [Pup04, PS11, SCR15], with a subtle but important difference compared to classical AMR procedures: each timestep is performed with the best possible refined mesh driven by the entropy production between  $t^n$  and  $t^{n+1}$ . In other words, if the numerical solution associated to its AMR mesh at time  $t^{n+1}$  could be improved (*dixit* the entropy production criterion) then, the mesh is appropriately refined, and, second, the solution is sent back at  $t^n$  for re-computation. Consequently a solution is always computed with the more appropriately refined mesh possible according to the entropy production criteria.

The main difficulty is to ensure that the association of computer technologies (parallelization and/or AMR like techniques) with a specific numerical method leads to a simulation code that is

Robust: intense shocks, wave interactions should not lead to a failure of the code;

Accurate: it is desirable to avoid inappropriate and sometimes excessive numerical diffusion and dissipation especially in regions of smooth flow. This is cured with clever numerical methods and/or adaptive mesh refinement techniques;

Efficient: the computer resources are continuously growing. Nonetheless they are finite, and thus precious enough not to be wasted. To build an efficient code, any part of development matters; from code architecture to parallelization environment, thrifty numerical schemes, AMR technology, implementing skills, efficient compiler and computers, etc.

Many other properties could be listed, such as re-usability, non-invasive add-ons, portability, etc. but in this work we simply focus on those three.

Our goal in this work is to show that, for 2D hydrodynamics equation, the coupling of *a posteriori* limiting strategy with *a posteriori* AMR technology leads to improvement in terms of resolution capability and effectiveness. This work will prove that the *a posteriori* limiting and AMR technology are relatively non-invasive to an existing serial code and also they pair together efficiently, by avoiding any prediction from data at the beginning of the timestep, but rather observing an updated numerical solution and possibly re-computing the same timestep if and where troubles have been detected.

Classically any *a priori* technique relies on data at discrete time  $t^n$  and on the ability of the mathematics to predict the behavior of the scheme at  $t^{n+1}$ , and, consequently, ensure good properties for the associated numerical solution. When a non-linear system of PDEs is solved by a non-linear high accurate scheme, such predictions are no more based on firm mathematical demonstrations, but often extrapolated from linear theory. Because we believe that observing problematic situations is easier than predicting their possible occurrence, we rely on *a posteriori* paradigms. Once problematic situations are detected at  $t^{n+1}$ , then, starting again from data at  $t^n$ , the numerical method must adapt (locally), and, provide a better suited numerical solution. In this work, both the AMR procedure based on entropy production diagnostics and the MOOD limiting procedure for the hydrodynamics solver rely on such *a posteriori* paradigm.

The rest of this paper is organized as follows. The second section briefly presents the system of PDEs solved, namely Euler equations in 2D on Cartesian geometry. Next in section 3 we present the third-order accurate finite volume scheme under Adaptive Mesh Refinement (AMR) framework. Space and time accurate discretizations are described supplemented by the AMR technology and, more specifically, the numerical entropy production used as the AMR threshold variable. Then in section 4 the *a posteriori* MOOD technique is presented as a limiting strategy to stabilize the previous third-order accurate scheme independently of the AMR procedure. The specific detection criteria, decrementing technique are discussed in the same section. This overall numerical method (2D third-order accurate FV scheme under AMR) is further validated and tested in section 5. Here, we gather the numerical results for a large set of test cases involving complex flows, interacting shock waves, contact discontinuities, rarefaction waves and re-circulation areas involving vortices. These tests assess the ability of the numerical method to maintain the optimal order of accuracy on smooth flow, and, an essentially non-oscillatory behavior on discontinuous solutions. Moreover the AMR technology permits to drastically increase the resolution of the computational mesh by appropriately allocating available computer resources. This is done at little cost compared to a fully refined mesh simulation. Conclusions and perspectives are drawn in section 6.

## 2. Hydrodynamics system of conservation laws

The model considered in this paper is the Euler equations of compressible gas dynamics. This system of partial differential equations (PDEs) reads

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho \mathbf{v} \\ \rho E \end{pmatrix} + \nabla \cdot \begin{pmatrix} \rho \mathbf{v} \\ \rho \mathbf{v} \otimes \mathbf{v} + p \mathbf{I} \\ \mathbf{v}(\rho E + p) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \quad (1)$$

where  $\rho$  denotes the mass density,  $\mathbf{v} = (u, v)$  the velocity vector,  $p$  the fluid pressure,  $E$  the total energy density and  $\mathbf{I}$  the  $2 \times 2$  identity matrix.  $\mathbf{v} \otimes \mathbf{v}$  is the dyadic product of the velocity vector with itself. The system is closed thanks to a perfect gas law (the equation of state (EOS)) with ratio of specific heats  $\gamma$ :

$$p = (\gamma - 1) \left( \rho E - \frac{1}{2} \rho \mathbf{v}^2 \right). \quad (2)$$

Classically the pressure is a function of two thermodynamics variables, namely the density and the specific internal energy  $\varepsilon = E - \frac{1}{2} \mathbf{v}^2$  and the gas constant  $\gamma$ . The sound speed is defined as  $a^2 = \gamma p / \rho = \gamma(\gamma - 1)\varepsilon$ . From this definition one deduces that physically admissible states are such that  $\rho > 0$  and  $p > 0$ , or, equivalently  $\rho > 0$  and  $\varepsilon > 0$ .

A general formulation of this nonlinear system of hyperbolic balance laws is given by

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{U}) = \mathbf{0}, \quad \mathbf{x} \in \Omega \subset \mathbb{R}^2, \quad t \in \mathbb{R}_0^+, \quad (3)$$

with appropriate initial and boundary conditions where  $\mathbf{x} = (x, y)$  is the coordinate vector,  $\mathbf{U} = (\rho, \rho u, \rho v, \rho E)$  is the vector of 4 conserved variables,  $\mathbf{F} = (\mathbf{f}, \mathbf{g})$  is the conservative nonlinear flux tensor depending on  $\mathbf{U}$ . The computational domain  $\Omega \subset \mathbb{R}^2$  is discretized by a set of  $N_E$  not necessarily conforming quadrangular elements  $\Omega_i$ , see figure 1. The union of all elements is referred to as the mesh or grid,  $\mathcal{M}_\Omega = \bigcup_{i=1}^{N_E} \Omega_i$ . The time is also discretized into cells  $\Theta^{n+1/2} = [t^n, t^{n+1}]$  defined by  $n = 0, \dots, N$  discrete times  $t^n < t^{n+1}$  with the time step  $\Delta t = t^{n+1} - t^n$  and the initial and final times being  $t^0 = t^{\text{init}}$  and  $t^N = t^{\text{final}}$ . We will see later that the time step, which need not be constant, is restricted by stability condition which depends on the numerical scheme employed. A so-called *finite volume* is one space-time cell  $\Omega_i \times \Theta^{n+1/2}$  onto which the numerical scheme will operate by considering interactions with neighboring finite volumes.

## 3. High accurate finite volume scheme for the Euler system of PDEs

In this paper we solve system of equations (3) by a highly accurate finite volume scheme. We first introduce the cell averages of the solution as

$$\mathbf{U}_i(t) = \frac{1}{|\Omega_i|} \int_{\Omega_i} \mathbf{U}(\mathbf{x}, t) d\mathbf{x}. \quad (4)$$

Furthermore, let  $\mathbf{u}_i^n$  be the approximation of  $\mathbf{U}_i(t^n)$  computed by the numerical scheme. In this section we describe the algorithm that computes  $\mathbf{u}_i^{n+1}$  from the cell averages at time level  $t^n$ . To this end, we consider the following semidiscretization of (3):

$$\frac{d}{dt} \mathbf{u}_i(t) = - \int_{\partial \Omega_i} \mathbf{F}(\mathbf{u}(s, t)) \cdot \mathbf{n}(s) ds, \quad (5)$$

where  $\mathbf{n}$  denotes the outgoing unit normal to the boundary of  $\Omega_i$ . Next, we will describe first the high order discretization  $\mathbf{L}_i(\mathbf{u}(t))$  of the spatial operator on the right hand side of (5) and then the time discretization will be achieved with a third order time accurate Runge-Kutta (RK3) scheme maintaining, at the same

time, better than second order of accuracy in space and time. At a difference from [SCR15, CS15], here stabilization of the high accurate reconstructions is obtained by means of an *a posteriori* MOOD limiting [CDL11a, DCL12, DLC13, LDD14] under the classical CFL condition of a RK3 scheme. At last an adaptive mesh refinement (AMR) technique is employed [SCR15] to enhance even further the accuracy of the overall scheme.

Next sub-sections describe the main aspects of this numerical method.

### 3.1. Polynomial reconstruction operator

The main ingredient of the proposed numerical method to reach high order of accuracy in space is the reconstruction operator on structured or unstructured meshes. In order to simplify the formulas, we drop the time dependence from our notation. The task of the reconstruction operator  $\mathcal{R}$  is to compute a piece-wise polynomial approximation of the data represented by the cell averages  $\mathbf{u}_i$  over the whole computational domain. More precisely, the restriction of  $\mathcal{R}$  to a generic cell  $\Omega_i$  is

$$\mathcal{R}(\mathbf{x})|_{\Omega_i} = \mathbf{w}_i(\mathbf{x}) = \mathbf{u}_i + \sum_{k=1}^{n_K} \hat{\mathbf{w}}_{i,k}^n \Psi_{i,k}(\mathbf{x}), \quad \forall i = 1, \dots, N_E. \quad (6)$$

In order to obtain third order accuracy, it is sufficient to take  $n_K = 5$  and we choose the basis functions

$$\Psi_{i,1} = x - x_i, \quad \Psi_{i,2} = y - y_i, \quad (7a)$$

$$\Psi_{i,3} = (x - x_i)^2 - \frac{1}{12} \Delta x_i^2, \quad \Psi_{i,4} = (y - y_i)^2 - \frac{1}{12} \Delta y_i^2, \quad \Psi_{i,5} = (x - x_i)(y - y_i), \quad (7b)$$

where  $\mathbf{x}_i = (x_i, y_i)$  is the centroid of cell  $\Omega_i$ , the size of which is  $\Delta x_i \times \Delta y_i$ . Following [SCR15], we have introduced the constant terms in  $\Psi_{i,3}$  and  $\Psi_{i,4}$  so that all basis functions have null cell average.

The reconstruction on element  $\Omega_i$  requires a so-called *reconstruction stencil*  $\mathcal{S}_i$ , that is an appropriate set including  $\Omega_i$  and a number of its neighbors that we denote as

$$\mathcal{S}_i = \bigcup_{k=1}^{n_e} \Omega_{j_i(k)}, \quad (8)$$

where  $n_e(i)$  is the number cells contained. Here  $k$  is a local index listing the elements in the stencil, and  $j_i(k)$  represents the mapping from local index  $k$  is  $\mathcal{S}_i$  to the global indexation in mesh  $\mathcal{M}_\Omega$ . Let us further assume that  $j_i(1) = i$ . In order to reach the nominal order of accuracy  $M + 1$ , we must choose  $n_e(i) \geq (M + 1)(M + 2)/2$ , see [BF90, OGA02, KI05]. In our case we include in  $\mathcal{S}_i$  all the cells intersecting  $\Omega_i$  at least in a point (see [SCR15] for a proof that there are at least 5 neighbors in a generic quad-tree mesh and for a discussion of the three-dimensional generalization). The local polynomial  $\mathbf{w}_i$  is then defined by the requirements that

$$\frac{1}{\Omega_{j_i(k)}} \int_{\Omega_{j_i(k)}} \mathbf{w}_i(\mathbf{x}) d\mathbf{x} = \mathbf{u}_{j_i(k)} \quad \forall k = 1, \dots, n_e(i) \quad (9)$$

The above, since  $n_e$  is larger than the number of coefficients of a degree 2 polynomial in 2 space variables, represents an over-determined system of linear equations in the coefficients of  $\mathbf{w}_i$ . This system should be solved in a least square sense, but, for conservation, the equation for  $k = 1$  should be satisfied exactly. This is facilitated by our choice of basis functions with zero average, so that it is sufficient to solve the equations (9) for  $k \geq 2$  as an unconstrained least squares problem.

Finally, we point out that the stencil  $\mathcal{S}_i$  employed in this work is strictly smaller than the stencils employed by [DK07, DKTT07a] for the same level of accuracy.  $\mathcal{S}_i$  is the same as the stencil of the CWENO3 reconstruction of [SCR15]. Here, by employing a single central unlimited polynomial, we save the extra-cost of several polynomial reconstructions and the associated cost of blending them, but we will need to devise an appropriate a-posteriori limiting strategy in order to avoid the onset of spurious oscillations.

### 3.2. Finite volume spatial discretization

The nature and accuracy of the finite volume scheme employed to solve (5) entirely depends on the way the integrals on its right hand side are approximated. Let us assume that  $\partial\Omega_i$  is decomposed into edges  $e_{ij}$  separating  $\Omega_i$  from a neighbor cell  $\Omega_j$ . Usually solving the spatial flux in (5) requires the use of a quadrature formula along each edge  $e_{ij}$  so that

$$\int_{\partial\Omega_i} \mathbf{F}(\mathbf{U}) \cdot \mathbf{n}(s) ds \approx \sum_{e_{ij}} |e_{ij}| \sum_{g=1}^G \omega_g \mathbf{F}(\mathbf{U}(\xi_g)) \cdot \mathbf{n}_{ij}(\xi_g), \quad (10)$$

where  $\omega_g, \xi_g$  are respectively the weights and integration points of the quadrature formula having  $G$  integration points and  $|e_{ij}|$  is the length of edge  $e_{ij}$ . In this work, we employ the 2-point Gaussian formula.

On the two sides of any edge two reconstructions are defined, namely  $\mathbf{w}_i(s)$  and  $\mathbf{w}_j(s)$ , with  $s \in e_{ij}$ , and, usually, they do not coincide. A numerical flux  $\mathcal{F}$  is then employed to choose uniquely the flux at the point  $s$ , to finally get

$$\mathbf{L}_i(\mathbf{u}) = \frac{1}{|\Omega_i|} \sum_{e_{ij}} |e_{ij}| \sum_{g=1}^G \omega_g \mathcal{F}(\mathbf{w}_i(\xi_g), \mathbf{w}_j(\xi_g), \mathbf{n}_{ij}(\xi_g)). \quad (11)$$

In this work we employ the simple Local Lax-Friedrichs (LLF) flux, i.e.  $\mathcal{F}(\mathbf{a}, \mathbf{b}) = (\mathbf{F}(\mathbf{a}) + \mathbf{F}(\mathbf{b}) - \alpha(\mathbf{b} - \mathbf{a}))/2$ , where  $\alpha$  is the largest of the eigenvalues of  $\mathbf{F}'(\mathbf{a})$  and  $\mathbf{F}'(\mathbf{b})$ . Of course other exact or approximate Riemann solver could be employed for this matter [Tor99], like the classical HLLC solver.

The accuracy in space of this finite volume scheme is related to the accuracy with which representations  $\mathbf{w}_i$  are built, assuming that the quadrature formula is of appropriate accuracy. The procedure to compute the operator  $\mathbf{L}_i(\mathbf{u})$  at given time  $t$  is: (i) compute the polynomial reconstructions  $\mathbf{w}_i(x)$  for every cell in the mesh using the cell averages at time  $t$  as data, (ii) compute the approximate values of  $\mathbf{w}_i$  and  $\mathbf{w}_j$  at all integration points  $\xi_g$  on each side of every edge  $e_{ij}$ , (iii) solve the Riemann problem at integration points, or, equivalently compute the associated numerical flux  $\mathcal{F}(\mathbf{w}_i^n(\xi_g), \mathbf{w}_j^n(\xi_g), \mathbf{n}_{ij}(\xi_g))$ , (iv) gather all fluxes in (11).

### 3.3. Time discretization

The accuracy in time is entirely related to the ability of the scheme to accurately integrate in time the semi-discrete equation (5). In this work we employ the classical TVD Runge-Kutta scheme of order 3 introduced in [GS98]. More in details our fully discrete scheme reads

$$\mathbf{u}_i^{(1)} = \mathbf{u}_i^n - \Delta t \mathbf{L}_i(\mathbf{u}^n), \quad (12a)$$

$$\mathbf{u}_i^{(2)} = \frac{3}{4} \mathbf{u}_i^n + \frac{1}{4} \mathbf{u}_i^{(1)} - \frac{1}{4} \Delta t \mathbf{L}_i(\mathbf{u}^{(1)}), \quad (12b)$$

$$\mathbf{u}_i^{n+1} = \frac{1}{3} \mathbf{u}_i^n + \frac{2}{3} \mathbf{u}_i^{(2)} - \frac{2}{3} \Delta t \mathbf{L}_i(\mathbf{u}^{(2)}), \quad (12c)$$

where  $\Delta t = t^{n+1} - t^n$  and the spatial operator is defined in (11).

The method (12) is by construction 3rd order accurate in time for smooth solutions. Its stability is guaranteed under a classical CFL condition

$$\Delta t \leq \lambda \min_i h_i \quad (13)$$

where  $h_i$  is the characteristic length of cell  $\Omega_i$ . The value of  $\lambda$  is computed easily from the  $\alpha$ 's employed in the LLF flux, since they record the maximal eigenvalues of the flux function.

### 3.4. Formal accuracy and robustness

Starting from  $\mathbf{u}^n$  and using the previous numerical method, we formally obtain  $\mathbf{u}^{n+1}$  which is a high-order accurate solution. More precisely the order of accuracy of the scheme is 3 in both space and time. Of course there would be no theoretical difficulty in considering spatial reconstruction operators with polynomials of larger degree, ensuring a higher spatial accuracy, but large values of  $M$  imply the use of associated larger stencils  $S_i$  which may render the scheme much less compact than the present one.

The scheme as described above is “unlimited”, in the sense that there is no embedded extra-dissipating mechanism which can assure that shock waves or steep gradients will not lead to spurious oscillations. Obviously a limiting procedure is to be employed to assure robustness and an essentially non-oscillatory behavior. Our chosen stabilization technique is described in section 4. However, any such procedure ineluctably reduces the accuracy of the method to 1 in the vicinity of the steep gradients.

To increase the accuracy even further we consider the AMR technique described in section 3.5. This technique coarsens the computational grid in locations where the flow can be described with less degrees of freedom. Conversely the AMR procedure refines the grid in locations where more cells are needed to capture small scale features and compensates for the local order reduction often induced in those areas by the limiting procedure.

### 3.5. Adaptive Mesh Refinement (AMR)

The adaptive grid is recursively generated starting by a coarse uniform, two-dimensional, Cartesian mesh of grid size  $H$  at level  $\ell = 0$ . According to some criterion each cell can be recursively subdivided into four equal squares. At the end of the recursive subdivision, the grid structure is described by a quad-tree. One example of subdivision of an original cell with  $\ell^{\max} = 3$  levels of refinement is illustrated in figure 1, together with the corresponding quad-tree. The cell corresponding to the level  $\ell = 0$  is the root of the quad-tree. Each cell of level  $\ell \in \{1, \dots, \ell^{\max} = 3\}$  has a father cell, which corresponds to its neighbor node in the quad-tree at level  $\ell - 1$ . The four nodes connected to the father node are called the children of the node. The children of each subdivided cell are given some prescribed ordering, in our case it is counter-clockwise starting from the upper-right cell.

Such subdivisions may lead to non-conformal cells as can be observed on the right panel of figure 1. When integration along the boundary of a non-conformal cell is invoked, in order to attain a third order accuracy, at least two quadrature points per intersection are needed. An intersection is defined as being a co-dimension 1 intersection between two adjacent cells. On the example of the right panel of figure 1 the blue cell has six neighbor cells, each demanding at least two quadrature points, leading to 12 flux evaluations to update this cell in each of the stages of (12). At each quadrature point one uses the cell centered polynomial reconstructions to extrapolate the physical state variables on each side of the intersection to feed the numerical flux function.

Any AMR procedure must have a mechanism which decides where to refine or derefine the mesh. In the present paper we employ the numerical entropy production, that was introduced in [Pup04] and further extended in [PS11]. Numerical entropy is available for any system of conservation laws with an entropy inequality. Moreover it scales as the truncation error in the regular regions, and its behavior allows to distinguish between contact discontinuities and shocks. Its definition considers an admissible entropy pair  $(\eta, \psi)$  and forms the quantity

$$S_i^n = \frac{1}{\Delta t} \left| \eta_i^{n+1} - \eta_i^n + \frac{\Delta t}{|\Omega_i|} \sum_{e_{ij}} |e_{ij}| \sum_{g=1}^G \omega_g \Psi(\xi_g) \right|, \quad (14)$$

where  $\Psi(\xi_g)$  is the numerical entropy flux function evaluated at integration point  $\xi_g$  using the reconstructions on both sides of intersection  $e_{ij}$  and  $\eta_i^n$ ,  $\eta_i^{n+1}$  denote the cell averages of the entropy on the numerical solution. Note that the computation of  $S_i^n$  is non-intrusive in a numerical scheme, since it amounts to the

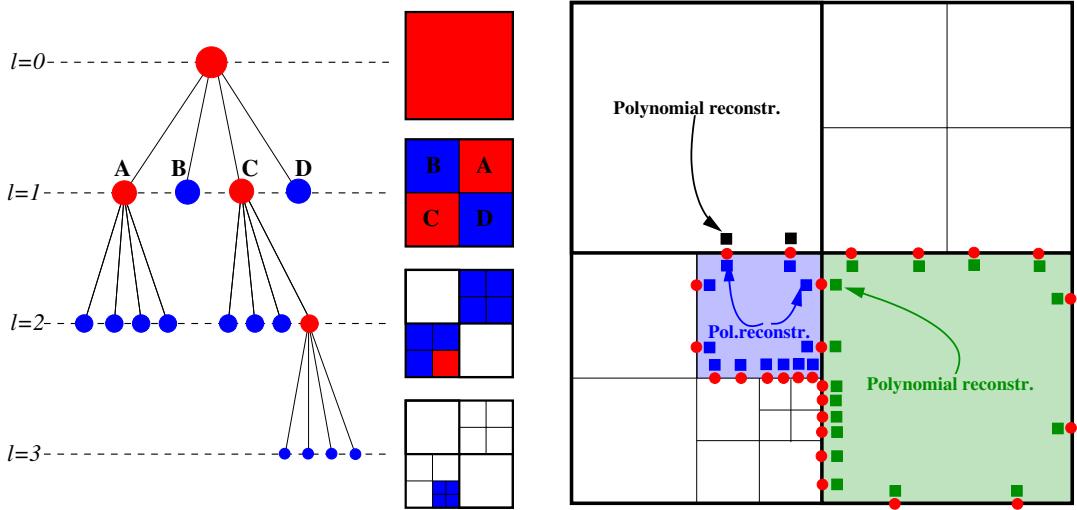


Figure 1: Left: Example of subdivision of an original cell with  $\ell^{\max} = 3$  levels of refinement: quad tree (left) and associate grid (right) — Right: Example of subdivision of an original cell and associated non-conformal mesh. Quadrature points are represented by red bullets. The blue cell has six neighbor cells each demanding at minima two quadrature points to reach third order accurate integration along the cell boundary. At each quadrature point one uses the polynomial reconstructions to extrapolate physical states (colored squares) on each side of the intersection to feed the numerical flux function.

addition of an extra variable in the vector  $\mathbf{u}$ , whose numerical flux  $\Psi$  can be computed using the same intermediate results (notably the polynomial reconstructions) that are needed to be fed into  $\mathcal{F}$ .

Following [PS11] if the solution is locally smooth then  $S_i^n = O(h^r)$  where  $r$  is the order of the scheme and  $h$  a characteristics length. If  $S_i^n = O(h)$  (resp.  $O(1/h)$ ) then a contact discontinuity (resp. shock) is located in  $\Omega_i$ .

At the end of each time-step, the quantity (14) is computed in every cell  $i$ , and compared to a threshold  $S^{\text{ref}}$ . If  $S_i^n > S^{\text{ref}}$  and the maximal refinement level  $\ell^{\max}$  is not yet reached then the cell is refined, and the cell averages in the newly created cells are set by averaging the reconstruction of  $\mathbf{w}_i^n$  and the time-step recomputed locally, that is  $\mathbf{u}_j^{n+1}$  is recomputed only in the numerical domain of dependence of the cell  $\Omega_i$  (see figure 2). When no further refinement is needed (or possible), a coarsening pass checks if all 4 direct children of a previously refined cell have an entropy production lower than a given coarsening threshold, i.e.  $S_i^n < S^{\text{coa}}$ . In this situation the 4 children are then replaced with their ancestor cell, where one sets  $\mathbf{U}^{n+1}$  to the average of the cell averages from the children cells. Following [PS11] we employ  $S^{\text{coa}} = S^{\text{ref}}/4$ .

More explicitly the AMR procedure consists in the following algorithm

- 1. Initialization  
Choose a maximal number of refinement  $\ell^{\max}$ , an initial mesh  $\mathcal{M}^{n=0}$ , threshold values  $S^{\text{ref}}, S^{\text{coa}}$  and set the associated data  $\mathbf{u}_i^{n=0}$ ;
- 0. Mark cells for re-computation  
Mark cells having a new value  $\mathbf{u}_i^n$ , and store their index in set  $C$ ;
- 1. Compute the candidate solution  
Compute a candidate solution  $\mathbf{u}^{n+1}$  and the diagnostics  $S_i^{n+1/2}$  on mesh  $\mathcal{M}^n$  from data  $\mathbf{u}^n$  for all cells in the domain of dependence of cells in  $C$
- 2. Decision on refining  
Do for all cell  $i$

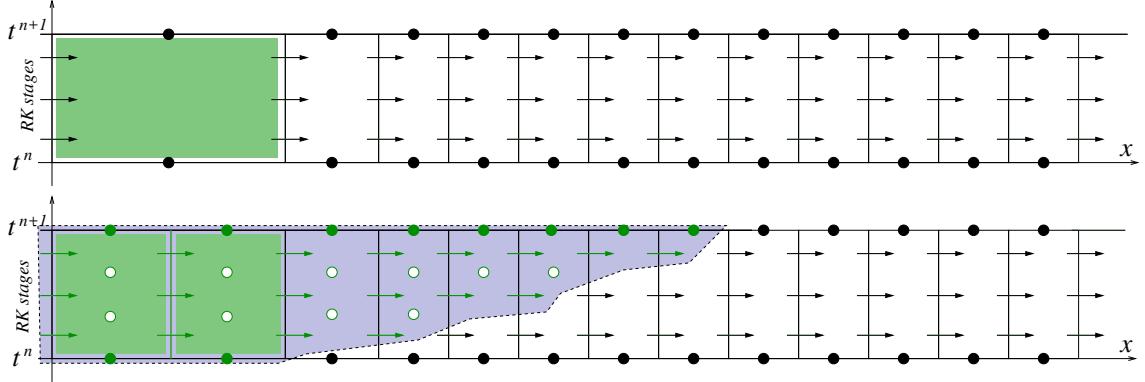


Figure 2: Re-computation after refinement — The top row depicts the tentative computation of the time advancement after which the red shaded cell is marked for refinement. Bullets (black or green) represent the cell averages and the arrows the numerical fluxes computed during the three Runge-Kutta stages — The bottom row depicts the re-computation performed after splitting the shaded cell. The dashed line indicates the numerical domain of dependence originating from the split cell. The stage values of the Runge-Kutta scheme that are recomputed are indicated by empty circles. The other stage values are not indicated for easier reading.

- (a) If  $S_i^{n+1/2} > S^{\text{ref}}$  and  $\ell_i < \ell^{\max}$ , re-computation is needed;
  - (b) Then split cell  $i$  into children cells, associate new children data averaging the polynomial reconstruction  $w_i(\mathbf{u}^n)$ ;
  - (c) Mark the children cells as having a new value  $\mathbf{u}_j^n$ ;
- End do for.
3. Decision on re-computation
    - If re-computation is needed,  $\mathcal{M}^n$  and  $\mathbf{u}^n$  have changed, go back to 1;
    - If re-computation is not needed then accept  $\mathbf{u}^{n+1}$  and continue;
  4. Decision on coarsening
    - Do for all set of sister cells (emanating from a common ancestor)
      - (a) If  $S_i^{n+1/2} < S^{\text{coa}}$  and  $\ell_i > 0$  for all sister cells, then coarsen the children cells into their common ancestor cell;
      - (b) Average sister data to set ancestor data into  $\mathbf{u}^{n+1}$ ;

Step -1 is performed only at the beginning of the time evolution, while steps 0–4 are performed for each time-step. In order to increase the accuracy, only at time  $t = 0$ , in step 2b the children values are set with the initial condition.

### 3.6. Summary

Up to now we have described a third-order accurate in space and time finite volume scheme within our AMR framework.

As such this scheme has no embedded limiting procedure to dissipate the occurrence of spurious numerical oscillations due to the Gibbs phenomenon. Therefore the scheme as it stands will produce numerical oscillations in the vicinity of shock waves and steep fronts. Moreover the generated oscillating solution would spoil the refining and coarsening procedures of the AMR framework.

In previous works, CWENO polynomial reconstructions have been used to limit and stabilize this scheme [PS11, SCR15, CS15]. In this work we will use an *a posteriori* stabilization procedure based on a troubled cell detector and subsequent re-computations with a second- or first-order finite volume scheme. This procedure is based on the so called MOOD paradigm, see [CDL11a, DCL12, DLC13, LDD14].

#### 4. *a posteriori* limited Finite Volume scheme

Classical WENO or CWENO reconstruction techniques have been developed elsewhere [JS96, HS99, DKTT07b, KI05, DK07, CS15] and, it has been shown that this approach maintains the overall accuracy and provides the necessary robustness of the scheme. In this paper we rely on an alternative technique using the so-called *a posteriori* MOOD (Multi-dimensional Optimal Order Detection) method. MOOD has been designed originally on fixed grids for Euler equations [CDL11a, DCL12, DLC13, CDL11b]. The *a posteriori* MOOD concept has also been used as a limiter for ADER schemes [LDD14], as a subcell limiter for high accurate Discontinuous Galerkin schemes in [ZDLS14, ZFDH15a] or as a high-order finite volume solver for convection-diffusion problems [CMNP13, CM14] and also to construct all-entropy finite volume schemes [VC13, VD13].

Most classical limiting procedures (slope limiter, artificial viscosity, WENO, CWENO, etc.) are *a priori* procedures as for they only use data at time  $t^n$  to perform two conceptually distinct operations: 1- predict in which cells extra-dissipation is needed, and, 2- define how much of such dissipation is sufficient. Ideally one demands that those operations lead to a provably robust scheme, free from spurious (possibly lethal) oscillations, and, equally importantly, maintaining the physical admissibility of the numerical solution. Moreover we desire that this limiting procedure reduce the accuracy as locally as possible. Maintaining the physical admissibility of the numerical solution is generally a difficult task for *a priori* limiting strategies specifically when a complex system of PDEs is solved.

Contrarily the *a posteriori* MOOD paradigm is based on the viewpoint that it is easier to observe an invalid numerical solution at  $t^{n+1}$  instead of predicting its occurrence from data at  $t^n$ . In the MOOD philosophy, a candidate solution is computed at  $t^{n+1}$ , this solution is tested through a list of admissible criteria (physically- or numerically-based), then troubled cells are sent back for re-computation with a more dissipating scheme, starting again from valid data at  $t^n$ . Several iterations (MOOD loop) may be needed because adding dissipation on one cell only in the vicinity of shocks may not be sufficient, and enlarging the stencil for dissipation is often mandatory. Let us describe in more details the approach.

##### 4.1. MOOD paradigm as an *a posteriori* stabilization technique

An *a posteriori* MOOD limiting strategy is therefore based on three objects: (i) a list of detection criteria, (ii) an ultra dissipative and robust scheme, the parachute scheme, (iii) a cascade of numerical schemes ordered from the more accurate to the parachute scheme.

1. The *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 time-step. The first set of criteria is based on the underlying physics, these 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 PDEs which are solved. Here, for the hydrodynamics system of conservation laws, the PAD criteria are the positivity of the density and the specific internal energy.

The second set of criteria relies on numerical properties. These are called Numerical Admissible Detection (NAD) criteria and they ensure that the numerical solution is essentially non-oscillatory. In [CDL11a, DCL12, DLC13, LDD14, ZDLS14] the NAD criteria are based on a 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 the occurrence of non-numeric values like *Not-a-Number* (NaN) or *Infinite* (Inf). If either the PAD or the NAD criteria are not fulfilled, then, locally, some action must be taken to supplement the scheme with more dissipation.

2. The *parachute* or *bullet-proof* scheme. The last scheme which one uses is called the *parachute* to express the fact that, for extremely difficult cases, this scheme is used as a last resort scheme. The candidate numerical solution provided by this scheme must always be considered as an acceptable one. As in most works involving a MOOD loop, as instance [CDL11a, DCL12, DLC13, CDL11b, LDD14], also in this work the first-order Godunov finite volume scheme is used.

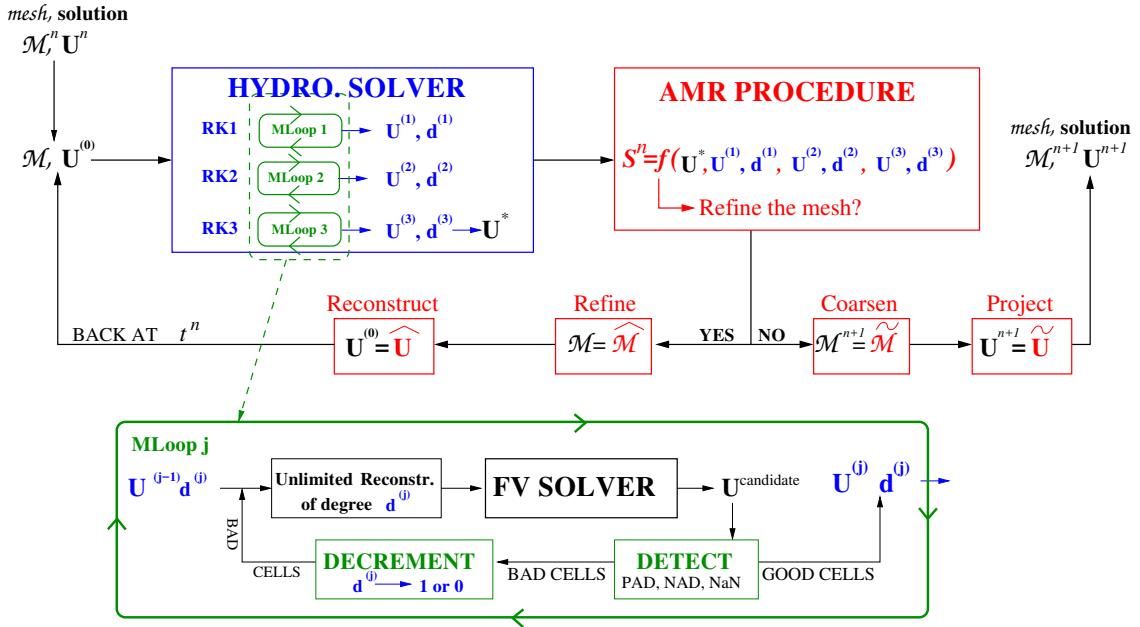


Figure 3: Sketch of our 3rd order Finite Volume AMR scheme. A single MOOD loop is depicted on the bottom part of the figure.

3. The *cascade*. The cascade is a list of ordered numerical schemes from the most accurate and prone to instability scheme, up to the least accurate but robust one [LDD14]. Here this sequence is related to the degree of the polynomial reconstructions. Precisely one sets a maximal polynomial degree  $d_{\max} = 2$  and we use the simplest sequence  $\mathbb{P}_2 \rightarrow \mathbb{P}_0$ <sup>1</sup>.

The MOOD loop embraces the main evolution routines of the high order numerical scheme, and, possibly, iterates to recompute some cells marked as problematic by the detection criteria, see figure 3. At the end of the MOOD loop the numerical solution is composed of good cells only, i.e. cells which have passed the detection criteria. This loop operates for each of the Runge-Kutta steps, and, furnished a valid numerical solution to the AMR procedure, which may require to refine the mesh and recompute the current time-step with this more adapted mesh.

The gain in efficiency brought by using the *a posteriori* MOOD paradigm is mainly due to the fact that usually few cells need decrementing the local polynomial degree. Therefore the extra-work implied by the MOOD loop to recompute a new candidate solution on those few problematic cells is genuinely low.

Without any doubt the detection criteria is the most important entity of the *a posteriori* MOOD procedure. It is based on physical and numerical properties considered as mandatory to accept the numerical solution within a cell.

#### 4.2. Physical Admissible Detection criteria (PAD)

The Physical Admissible Detection criteria state that for a candidate numerical solution at  $t^{n+1}$  to be valid in the case of Euler equations with perfect gas equation of state, the solution must obey the following positivity criteria:

$$\rho_i^{\text{candidate}} > 0, \quad \text{and} \quad \varepsilon_i^{\text{candidate}} = E_i^{\text{candidate}} - \frac{1}{2} |\mathbf{v}_i^{\text{candidate}}|^2 > 0, \quad (15)$$

<sup>1</sup> Note that more advance cascades have been employed elsewhere [CDL11a, DCL12, DLC13, LDD14], but, here we would like to demonstrate that even without fine tuning, obvious improvements are attained.

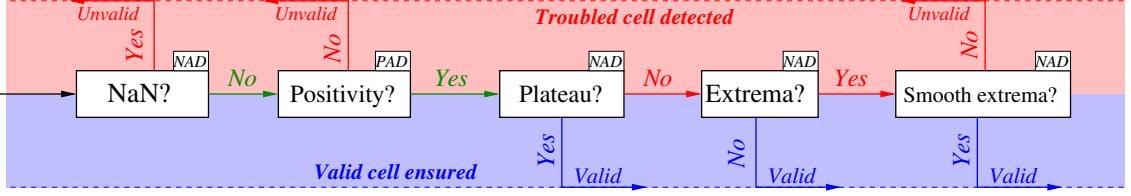


Figure 4: Sketch of the chain detector employed in the MOOD loop. A candidate solution in cell enters the chain from the left (black arrow), and successively passes through detection boxes. Each detection box answers by Yes or No, some answers directly validate the candidate solution (for instance: Is the cell value on a plateau?), or invalidate the solution (for instance: Is the solution polluted by a NaN?) which is flagged as problematic and sent back for re-computation.

so that pressure and sound-speed are unambiguously determined.

#### 4.3. Numerical Admissible Detection criteria (NAD)

The Numerical Admissible Detection criteria are based on the Discrete Maximum Principle (DMP) with the so-called relaxed u2 criteria [DLC13, BLD15], a plateau detection and a NaN (Not-A-Number) detection. All these NAD criteria must detect spurious numerical oscillations or lethal situations but ignoring smooth extrema or too small oscillations.

The DMP+u2 detection procedure acts on a generic variable  $w = (\mathbf{u}_h)$ . For a candidate solution  $w_i^{\text{candidate}}$  at time  $t^{n+1}$  in cell  $i$  for a given set of neighbor cells of index  $j \in \mathcal{V}_i$ , we first check if  $w_i^{\text{candidate}}$  fulfills the DMP. That is, if the local min/max are defined by  $m_i^n = \min_{j \in \mathcal{V}_i} (w_j^n, w_i^n)$  and  $M_i^n = \max_{j \in \mathcal{V}_i} (w_j^n, w_i^n)$ , the solution is considered as valid if  $m_i^n \leq w_i^{\text{candidate}} \leq M_i^n$ . The set of neighbors  $\mathcal{V}_i$  contains all neighbors of cell  $T_i$  sharing at least a common vertex with  $T_i$ . If the DMP is not fulfilled, then one must determine if this new extremum represents a smooth underlying function. To do so, one checks the u2 criterion described in [DCL12, DLC13] which states that a candidate solution which violates the DMP is nonetheless eligible if:  $X_i^{\max} X_i^{\min} > 0$  and  $\left| \frac{X_i^{\min}}{X_i^{\max}} \right| \geq 1 - \varepsilon$ , where  $\varepsilon$  is a smoothness parameter set to  $1/2$ ,  $X_i^{\text{candidate}}$  represents a measure of local discrete directional curvature. In our case the second derivative in the  $x$  direction of the local third order polynomial reconstruction is used. The minimal/maximal values are defined as  $X_i^{\min} = \min_{j \in \mathcal{V}_i} (X_j^*, X_i^*)$  likewise for the max value. The same check is done for the  $y$  component. In our case  $w$  is chosen as  $\rho$  only.

A second numerical detection criteria consists in ignoring cells on a numerical plateau. Obviously, the notion of a numerical plateau or flat area is not properly defined, but in this work we consider that a cell is always valid if  $(M_i^n - m_i^n) < h_i^3$ , where  $h_i$  is the (smallest) local characteristics length of current cell  $i$  if  $h_i < 1$ .

We also add a test for undefined or unrepresentable discrete values such as NaN (Not-a-Number) and Inf (Infinite). This last test allows the code to restart at the previous time step at any occurrence of non-numeric floating point values and thus genuinely helps for the robustness of the whole numerical method.

These detection checks are finally ordered into a chain, see figure 4, which allows fast exits to avoid unnecessary computations.

#### 4.4. Decrementing

In general the decrementing designs the procedure that determines the next scheme of the cascade that will be tried on troubled cells and their neighbors. In this work the numerical fluxes can be computed with only two possibility, either the cell is flagged with degree  $d_i = 2$  as 'high-accurate', then all fluxes are computed with  $\mathbb{P}_2$  reconstructions, or the cell is flagged by  $d_i = 0$  as 'low-accurate' then all fluxes are computed with piece-wise constant data, see figure 5 for an illustration. Therefore when two neighbor cells are assigned a different cell polynomial degree, say  $d_i = 2$  and  $d_j = 0$  (black and blue cells in the figure),

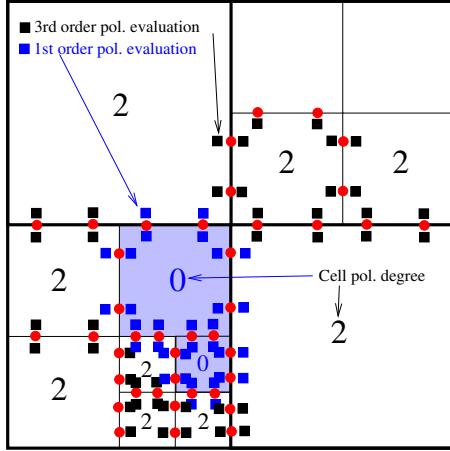


Figure 5: Illustration of the decrementing procedure. When two neighbor cells are assigned different polynomial degrees, then the reconstructions at their interface uses the smallest degree. When two cells have the same degree, then the reconstructions on both sides also employ this degree.

then the effective polynomial evaluations at both sides of the common intersection  $e_{ij}$  are made using the lowest degree  $d_{ij} = \min(d_i, d_j) = 0$ , refer to a blue/black cell intersection in figure 5.

For an AMR mesh, this decrementing is slightly more demanding because several control points must be considered around the current cell, each being assigned a polynomial degree according to the current degree and the degree of the cell across. With such a procedure we ensure that if a cell is assigned a degree 0, then, all its fluxes are computed with a first-order of accuracy. Some cells assigned a degree 2 are *de facto* updated with a mix of third-order and first-order accurate fluxes. However this has no impact on conservation because the numerical method is written in flux form, therefore each flux at interface point is used on each side with opposite signs.

#### 4.5. Iterative MOOD loop

For each stage of the RK3 scheme an iterative MOOP loop is run until convergence, as

0. High order unlimited solution  
Compute the third order accurate candidate solution  $\mathbf{U}^{(j)}$ , where  $d_i = 2$  for all cell  $i$  starting from  $\mathbf{U}^{(j-1)}$  ( $\mathbf{U}^{(0)} = \mathbf{u}^n$ );  
(a) Detect and build a list of  $N_b$  bad cells  $\mathcal{B}^0$ .  
(b) Set  $d_i = 0$  for all  $i \in \mathcal{B}^0$ .
1. MOOD loop  
Do while  $N_b > 0$   
Compute the candidate solution  $\mathbf{U}^{(j)}$  in the domain of dependence of the newly decremented cells  
(a) Detect and build a list of  $N_b$  bad cells  $\mathcal{B}^k$ .  
(b) Set  $d_i = 0$  for all  $i \in \mathcal{B}^k$ .  
End do while.

Notice that this iterative loop always converges if the first order FV scheme is compatible with the detection criteria. Moreover we set as valid any cell computed with  $d_i = 0$ , in other word, the first order Godunov FV scheme is supposed to always furnish an acceptable solution. As such, at the end of each RK3 stage, the provided solution is either high-order accurate and acceptable with respect to the detection criteria, or is locally first-order accurate.

## 5. Numerical experiments

This section introduces and describes a list of representative test cases for the hydrodynamics system of PDEs. Numerical solutions given by the AMR-MOOD numerical method are compared with the results from the classical AMR-CWENO numerical method to assess the gain brought by the use of the new limiting strategy. Next, we illustrate the gain brought by increasing the number of level of refinement within our AMR-MOOD framework.

The goals of this section on numerical test cases can be summarized as follows.

1. We want to numerically validate that the coupling of MOOD and AMR techniques provides an accurate and robust numerical method in 2D
  - for smooth solutions, we want to show that the effective optimal accuracy is attained with and without AMR, see the isentropic vortex in motion test case;
  - for non smooth flows (shock tubes, double Mach reflection, forward facing step) we would like to observe that the AMR-MOOD scheme is able to provide non oscillatory and accurate numerical solutions. Two complementary effects are acting: (i) maintaining optimal 3rd order of accuracy on regions where smooth flows are detected, and using a first-order accurate scheme only for few cells where steep gradients are detected, and, (ii) adaptation of the mesh by refining/coarsening where appropriate.
2. We will show that this AMR-MOOD numerical scheme is efficient in terms of CPU time, and accuracy. The comparison is made against the classical AMR-CWENO numerical method in 2D of [SCR15].
3. We will monitor the number of troubled cells detected by MOOD to prove that the detection procedure does not flag many cells, and, as such, limit the number of MOOD iterations.
4. Last, we will show that the *a posteriori* treatment in the numerical scheme renders the overall method robust when notoriously difficult tests are simulated (double Mach reflection or forward facing step problems).

In order to perform our tests, the numerical scheme described in this paper was coded with the help of the DUNE numerics library [BBD<sup>+</sup>08], employing the ALUGrid quad-tree mesh [ADKN16]. The simulations with the AMR-CWENO method were performed with the `dune-fv` module [SC14].

### 5.1. Initialization

In an AMR simulation, the initial mesh made of  $N^0$  cells in any direction and with  $\ell$  refinement levels must be adapted to the initial condition (IC). From these data we define  $N^{eq} = N^0 \times \ell$  which is the number of cells if a full refined mesh were to be employed. Our preferred strategy consists in recomputing the first time step several times, by letting the AMR machinery refine where appropriate. An alternative solution consists in adapting the mesh to the initial gradients contained in the IC if any. As such we can produce initial refined grids like the ones depicted in figure 6 with  $\ell = 5$  levels. The colors correspond to data values in order to show how the AMR procedure does adapt the mesh to the presence of gradient (first two panels), or, to waves emanating during the very first time iteration (third panel). (Those meshes will be employed in the following sections.)

### 5.2. Isentropic vortex in motion

The isentropic vortex problem [Shu97] tests the accuracy of numerical methods since an exact, smooth and analytic solution exists. The computational domain is set to  $\Omega = [-5, 5] \times [-5, 5]$ . The ambient flow is characterized by  $\rho_\infty = 1.0$ ,  $u_\infty = 1.0$ ,  $v_\infty = 1.0$  and  $p_\infty = 1.0$ , with a normalized ambient temperature  $T_\infty^* = 1.0$ . The perfect gas equation of state with  $\gamma = 1.4$  is considered. At the initial time  $t = 0$ , onto

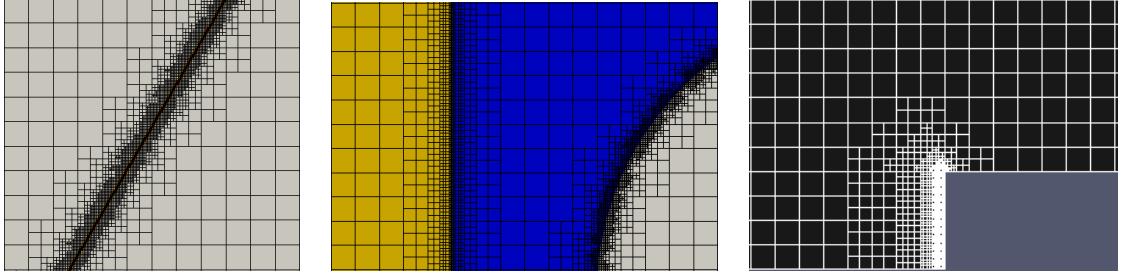


Figure 6: Examples of initial meshes (zooms) obtained with the AMR machinery at  $t = 0$  showing the adaptation to gradients in the initial conditions (left-middle panels) or to the waves emanating during the very first time-step (right panel).

this ambient flow is superimposed a vortex centered at  $(x_{\text{vortex}}, y_{\text{vortex}}) = (0, 0)$  with the following state:  $u = u_{\infty} + \delta u$ ,  $v = v_{\infty} + \delta v$ ,  $w = w_{\infty}$ ,  $T^* = T_{\infty}^* + \delta T^*$ , where the increments are given by

$$\delta u = -y' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta v = x' \frac{\beta}{2\pi} \exp\left(\frac{1-r^2}{2}\right), \quad \delta T^* = -\frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp(1-r^2),$$

with  $r = \sqrt{x'^2 + y'^2}$  and  $x' = x - x_{\text{vortex}}$ ,  $y' = y - y_{\text{vortex}}$ . The so-called strength of the vortex is set to  $\beta = 5.0$  and the initial density is given by

$$\rho = \rho_{\infty} \left( \frac{T^*}{T_{\infty}^*} \right)^{\frac{1}{\gamma-1}} = \left( 1 - \frac{(\gamma-1)\beta}{8\gamma\pi^2} \exp(1-r^2) \right)^{\frac{1}{\gamma-1}}. \quad (16)$$

Periodic boundary conditions are prescribed. At final time  $t = t_{\text{final}} = 10$  the vortex is back to its original position. Therefore the initial and final numerical solutions should match up to precision of the numerical method. Because the solution is a smooth solution, then it must be simulated with effective optimal high accuracy, in other words the limiting/stabilization procedure employed in the scheme should not have any effect along all the time evolution.

The simulations reported here for this test were computed with a fixed  $\Delta t = 0.125 \min_{Q_i^0} \{h_i\}$ . The results with the classical CFL restricted time-step were analogous are not reported. For each simulation, we compute the discrete  $L^2$  and  $L^{\infty}$  norm errors between the initial and final piece-wise constant data for the density variable.

### 5.2.1. Fixed grids

Successively refined Cartesian grids made of  $N_c = N^2$  squares are constructed given a number of cells in  $x/y$  directions starting from  $N = 32$  (coarse grid) up to  $N = 512$  (fine grid). For this test we shut off the AMR capability of the schemes because the goal is to observe if the effective high accuracy is attained when the *a posteriori* stabilization technique is active.

In Table 1 we report these errors and the associated orders of convergence for three schemes as a function of the CPU time. First, the simulation is ran with the unlimited 3rd order accurate scheme (with PAD safety check). This is the most accurate, less expensive and physically valid scheme we could employ for such a smooth problem. Second, we test the 3rd order accurate CWENO scheme and, third, the proposed MOOD schemes of same nominal accuracy. We run several MOOD versions which differ in the detection criteria used. We have employed a full DMP on all conservative variables, DMP+u2 on density and DMP+u2 on entropy. Three runs are made with the same scheme configuration and the CPU times to complete the simulations are averaged in order to mitigate unavoidable fluctuations in the speed of computers (in all cases the standard deviation of the data was below 1% of their average). CWENO and MOOD schemes

have an embedded limiting strategy, therefore they must be slightly more expensive than the unlimited scheme, and, possibly less accurate if the limiting strategy is activated when it should not. We can observe from the errors and orders of convergence that the three schemes converge with an asymptotic third order of accuracy both in  $L^2$  and  $L^\infty$  norms. We also deduce that the *a posteriori* MOOD limiting is not activated as the errors are exactly the same as the unlimited results ones. This is one key feature of the *a posteriori* stabilization technique. Contrarily, the CWENO procedure is always active and therefore the errors are slightly different. This table furnishes a base results for the next table where the AMR strategy is employed. Note that the very same results are obtained when entropy is checked instead of density or if the DMP on all variables is to be checked. Consequently we omit these results in this table. In Table 2 we propose

$N_c$	Unlimited (PAD)			CWENO			MOOD $\rho$ $\mathbb{P}_2 \rightarrow \mathbb{P}_0$					
	$L^2$ error	$L^\infty$ error		$L^2$ error	$L^\infty$ error		$L^2$ error	$L^\infty$ error				
$32^2$	2.47E-02	—	3.29E-01	—	2.44E-02	—	3.27E-01	—	2.47E-02	—	3.29E-01	—
$64^2$	1.16E-02	1.10	1.88E-01	0.81	1.02E-02	1.26	1.68E-01	0.96	1.16E-02	1.10	1.87E-01	0.81
$128^2$	2.36E-03	2.30	3.03E-02	2.63	2.12E-03	2.26	2.92E-02	2.53	2.36E-03	2.30	3.03E-02	2.63
$256^2$	4.14E-04	2.51	5.07E-03	2.58	4.02E-04	2.40	6.33E-03	2.21	4.14E-04	2.51	5.07E-03	2.58
$512^2$	5.61E-05	2.88	6.78E-04	2.90	5.44E-05	2.88	6.09E-04	3.38	5.61E-05	2.88	6.78E-04	2.90
Expected order	3		3		3		3		3		3	

Table 1:  $L^2$  and  $L^\infty$  errors and convergence rate for the isentropic vortex problem for the unlimited 3rd order scheme, the MOOD  $\mathbb{P}_2 \rightarrow \mathbb{P}_0$  scheme and the CWENO  $\mathbb{P}_2$  scheme.

the cost in terms of CPU time for the unlimited (with PAD check), CWENO and MOOD schemes, all being nominally 3rd order accurate. From this data we can again observe that the unlimited and MOOD schemes are rather equivalent as they produce comparable timings. (Recall that these timings derive from an averaging procedure.) CWENO is slightly more expensive in terms CPU time. This is due to the fact that the whole CWENO procedure is acting while only the detection part of the MOOD procedure is operating for this test because no bad cell is ever detected. Last in figure 7 we present in curves the data from the

$N_c$	CPU time [s]					← scheme	← detection
	Unlimit. PAD	CWENO	MOOD $\rho$	MOOD $\eta$	MOOD DMP		
$32^2$	7.00	7.25	6.87	6.79	6.93		
$64^2$	54.24	56.61	53.90	53.71	54.91		
$128^2$	445.41	475.05	443.46	447.65	444.94		
$256^2$	3632.84	3775.71	3622.88	3662.43	3956.82		
$512^2$	30006.20	30391.90	29434.95	29424.60	32860.72		

Table 2: Wall-clock times and memory consumption for the vortex problem and CWENO and MOOD methods compared to the unlimited scheme — All schemes are nominally 3rd order accurate ones. The unlimited scheme results are considered as the basis to compute the ratio of computer resource consumption for CWENO and MOOD schemes.

previous tables, namely the errors as a function of CPU time for  $L^1$ ,  $L^2$  and  $L^\infty$  norms. The data for uniform grids are plotted with symbols linked by straight line. As already mentioned all schemes tested are almost equivalent for uniform grids, as the lower errors obtained by CWENO (Table 1) are counterbalanced by the longer computational times (Table 2).

### 5.2.2. AMR grids

In this section we allow the AMR during the simulation. The grid is initially set to be uniform made of  $N^0 \times N^0$  square cells. Here we employ  $N^0 = 32, 64$  and  $128$ . Next, this uniform grid is further adapted to the flow, with the parameter controlling the AMR set such that the entire vortex structure is refined with

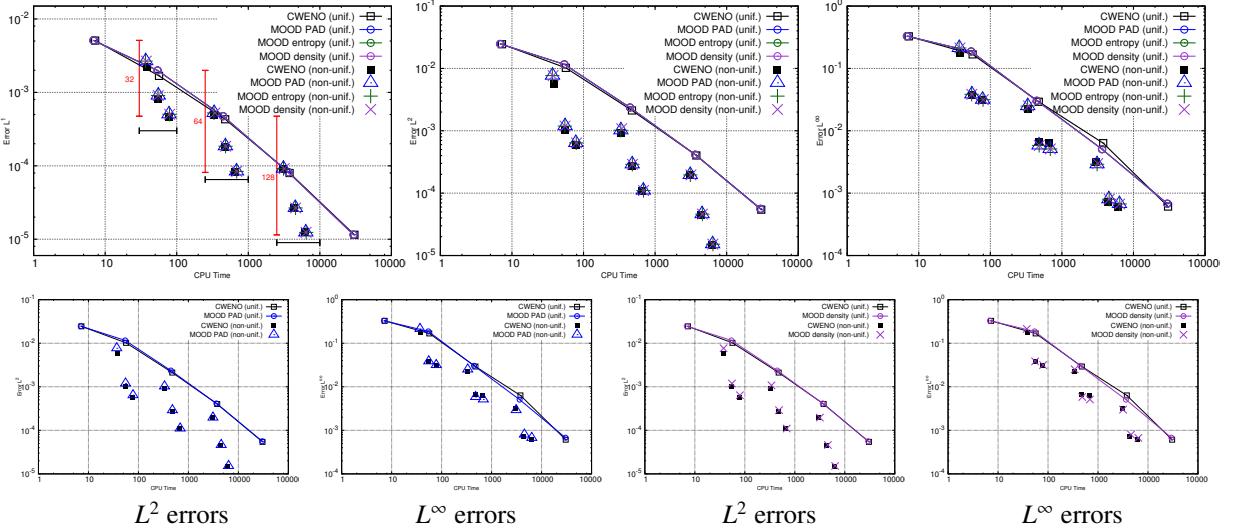


Figure 7: Isentropic vortex problem —  $L^1$ ,  $L^2$  and  $L^\infty$  error curves for CWENO and MOOD simulations as a function of CPU time — Top panels: results for uniform (line) and AMR (symbols) grids for all simulations for  $L^1$  (left),  $L^2$  (middle) and  $L^\infty$  (right) error norms. Bottom panels: comparison CWENO vs MOOD with PAD detection (first and second panels), and CWENO vs MOOD with  $u_2$  on density variable (third and fourth panels) — Each symbol corresponds to a specific threshold parameter, while each set of three symbols employs a grid made of  $N^0 = 32, 64$  or  $128$  cells with  $\ell = 3$  refinement levels.

$\ell = 3$  levels, leading to  $N^{eq} = 256, 512$  and  $1024$  respectively. The AMR procedure strongly depends on the threshold values  $S^{ref}$ .

First, the value  $S^{ref} = e = 10^{-3}$  was made for the run with  $N^0 = 32$  and  $\ell = 3$ . Since the error indicator on smooth flows scales as the formal order of the scheme, i.e.  $O(h^3)$ , the corresponding choice for the  $N^0 = 64$  and  $N^0 = 128$  runs are thus  $e/8$  and  $e/64$  (see [PS11, SCR15]). In order to better assess the influence of the refinement parameter on the results, each of these reference runs was compared with other ones with a 10-fold increase and 10-fold decrease in  $S^{ref}$  (see Table 3).

As expected the AMR grid initially covers, and later follows, the vortex structure during its drift for any

$N^0$	Threshold value. Base: $e = 10^{-3}$		
	Run 1	Run 2	Run 3
32	10e	e	0.1e
64	10e/8	e/8	0.1e/8
128	10e/64	e/64	0.1e/64

Table 3: Isentropic vortex problem — Threshold values used for different meshes.

( $N^0, \ell$ ), see the meshes and density (color) in figure 8 at final time for the three threshold values from table 3 and  $N^0 = 64, \ell = 3$ . We observe that the threshold value reduces or increases the layer of fine cells around the vortex structure for all schemes. Its choice hence drives the number of fine cells employed to capture small scale structures, and, as a consequence, the CPU time needed to complete the simulation. In table 4 are gathered the  $L^2$  and  $L^\infty$  errors when the initial coarse grid is made of  $N^0$  cells per direction and  $\ell$  refinement levels are used leading to the effective finest grid of  $N^{eq}$  cells. Ideally the error for  $N^0 = 32$  and  $\ell = 3$  in this table should be comparable with the results from table 1 with  $N_c = N^{eq} \times N^{eq} = 128^2$ , but the computational time is expected to be lower. In this table we also report the final total number of cells  $N_c(t)$

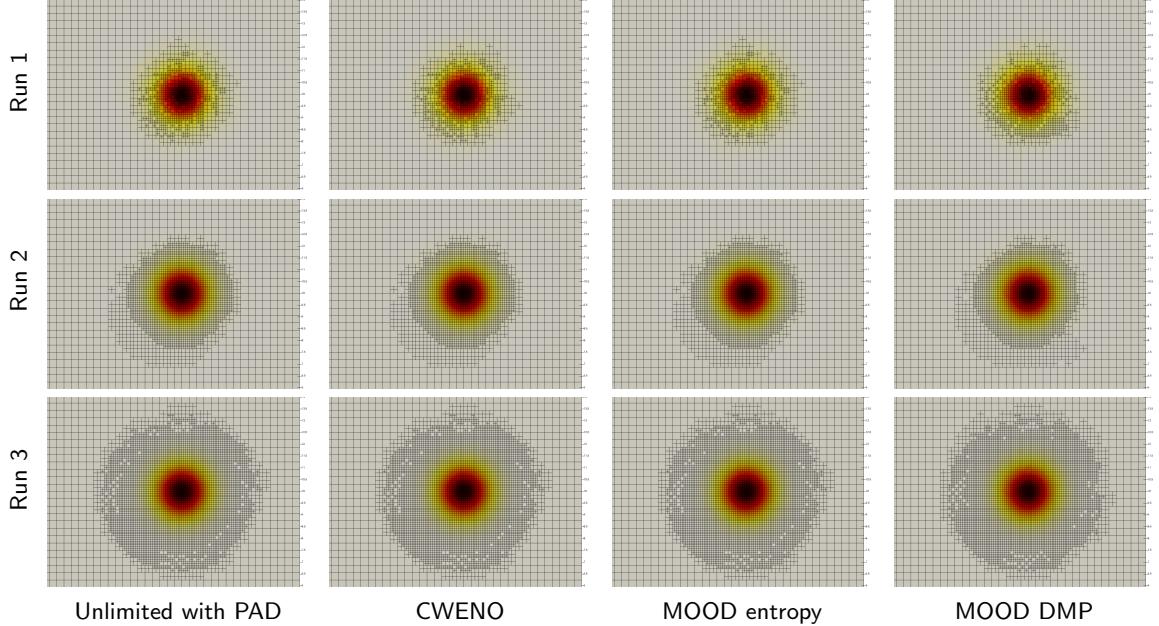


Figure 8: Isentropic vortex problem — Zoom on the final AMR grids for different threshold values (Run 1, 2, 3) starting with  $64 \times 64$  uniform cells and  $\ell = 3$  levels of refinement — From left to right panels: Unlimited (with PAD check), CWENO, MOOD with u2 on entropy, MOOD with DMP on all variables.

and the CPU time needed to complete the simulations. Data are reported for the three threshold values from table 3 and referred to as Run 1, 2, and 3.

To ease the analysis of these data, we have also reported them in figure 7 as free symbols. We can observe that they gather as triplets ordered from the smallest to the largest threshold values. The first triplet, below 100 units of time corresponds to  $N^0 = 32$ , the second one to  $N^0 = 64$  and the last one to  $N^0 = 128$ . For each triplet, the error decreases while the CPU time increases, because the number of fine and total cells also increases. Nevertheless, all these AMR simulation data are located below the uniform curves (line) meaning that the AMR simulations are more accurate and less expensive. Moreover we have plot on the  $L^1$  norm error curve the expected range of error (red segments) into which the AMR simulations should lie. These segments correspond to the range of error for uniform grid simulations when the grid is refined two times. For instance the first segment corresponds to the range between the first (uniform grid with  $N = 32$ ) and third ( $N = 128$ ) data, because any AMR simulation with  $(N^0, \ell) = (32, 3)$  must provide a better accuracy than the uniform 32 grid simulation, and can not exceed the accuracy of a  $N^{eq} = 128$  grid.

Next we propose in figure 9 the evolution of the number and the percentage of cells in each level of refinement during the MOOD with u2 on density simulation when  $N^0 = 32$  and  $\ell = 3$ . We observe an almost constant total number of cells and the number of cells in each refinement level. More specifically the middle level N1 is less populated due to the fact that these cells pave a transition layer between coarse and fine levels. Because the vortex does change neither its shape nor its size, the distributions of coarse, intermediate and fine cells are expected to behave as observed. The very same behavior is reported for all schemes for this relative mild test, therefore we omit these figures. Let us emphasize that the MOOD detection process does not mark any bad cell, or so few that it is barely of no impact on accuracy, nor on computational timings.

This isentropic vortex test case has shown that in the case of a smooth problem the finite volume numer-

$(N^0, \ell), N^{eq}$	Run	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU
Unlimited (with PAD check)													
$(32,3), 128$	1	1117	7.68E-03	2.13E-01	36.84	1150	5.63E-03	1.77E-01	38.24	1117	7.68E-03	2.13E-01	37.63
	2	1678	1.19E-03	3.84E-02	55.25	1687	1.04E-03	3.77E-02	54.96	1678	1.19E-03	3.84E-02	55.14
	3	2407	6.40E-04	3.15E-02	77.97	2380	5.83E-04	3.07E-02	77.22	2407	6.40E-04	3.15E-02	79.69
$(64,3), 256$	1	5254	1.02E-03	2.49E-01	333.26	5305	9.26E-04	2.26E-02	332.62	5194	1.08E-03	2.51E-02	341.80
	2	6757	2.90E-04	5.83E-03	478.43	6832	2.76E-04	6.53E-03	477.02	6757	2.90E-04	5.83E-03	483.21
	3	9277	1.10E-04	5.12E-03	684.24	9505	1.08E-04	6.34E-03	662.18	9277	1.10E-04	5.12E-03	689.91
$(128,3), 512$	1	21496	1.96E-04	2.91E-03	3099.37	21559	1.96E-04	3.13E-03	3036.33	21514	1.96E-04	2.99E-03	3173.55
	2	27757	4.61E-05	7.99E-04	4530.64	27901	4.48E-05	7.32E-04	4347.57	27757	4.61E-05	7.99E-04	4583.56
	3	36712	1.52E-05	6.71E-04	6364.56	36664	1.48E-05	6.09E-04	6145.22	36712	1.52E-05	6.71E-04	6415.37

Table 4: Final number of cells  $N_c(t)$ ,  $L^2$  and  $L^\infty$  errors, CPU time, for the isentropic vortex problem at  $t = 10$  simulated by limited/unlimited 3rd order schemes using AMR capability. Results for the unlimited (with PAD check), CWENO and MOOD with u2 detection on density schemes.  $N^0$  is the number of cell in each direction for the initial grid,  $\ell$  if the number of refinement level leading to  $N^{eq}$ , the number of cell for each direction in an equivalent fixed grid.

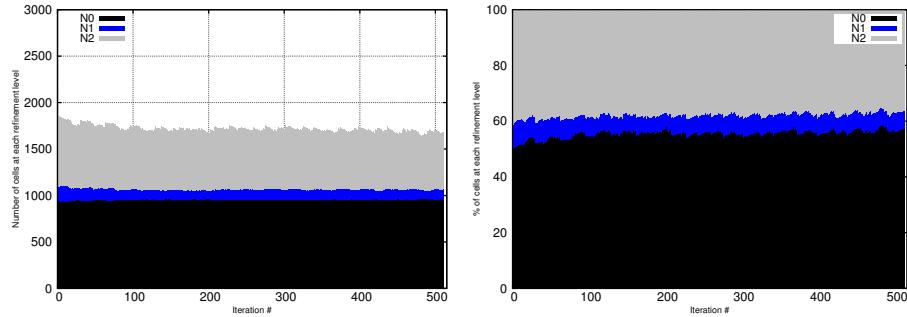


Figure 9: Evolution of the number of cells for each level of refinement as a function of the iteration number for the vortex problem — AMR simulation for MOOD with u2 detection on density for  $N^0 = 32$  and  $\ell = 3$  — Left panel: number of cell — Right panel: percentage of cell.

ical scheme supplemented with *a posteriori* MOOD stabilization is able to maintain its effective optimal accuracy at a reasonable cost in terms of CPU time. The coupling of *a posteriori* MOOD limiting with AMR capability produces comparable results with standard CWENO results. Playing with the threshold parameter we can still fine tune the amount of cells, which is a classical situation for any AMR simulation. In the next section we deal with discontinuous solutions for which some amount of limiting is mandatory.

### 5.3. Radial Sod shock tube

Next we run the cylindrical Sod shock tube problem in 2D. We plan to assess the ability of the scheme to capture simple but non-aligned cylindrical waves without producing spurious oscillations. The initial conditions for velocity component are  $u, v = 0$  everywhere, while density and pressure are  $(\rho_H, p_H) = (1, 1)$  for the central region  $r = \sqrt{x^2 + y^2} < 0.5$  and  $(\rho_L, p_L) = (0.125, 0.1)$  elsewhere. The ratio of specific heats is  $\gamma = 1.4$ , along with the perfect gas equation of state. The computational domain is set to  $\Omega = [0; 1] \times [0; 1]$  with symmetry boundary conditions. The final time is set to  $t_{\text{final}} = 0.2$ . This one-dimensional radial Riemann problem can be solved following for instance [Tor99] and our reference solution is computed using a classical 1D MUSCL TVD scheme with 25000 cells.

The initial grid is either uniform, made of  $N \times N$  quadrangular cells with  $N = 32, 64, 128, 256$  or  $512$ . AMR type of grids is also employed starting from  $N^0 = 32$  and  $\ell = 3, 4$  or  $5$  levels of refinements. Likewise for the previous test, the grid is refined and adapted to the discontinuity with  $\ell$  levels. We run the MOOD and CWENO versions of the code. The MOOD schemes employs the u2 detection either on density or entropy.

The AMR procedure strongly depends on the threshold values  $S^{\text{ref}}$ . Again, the threshold value  $S^{\text{ref}}$  was

selected as  $\epsilon = 10^{-3}$  by examining few  $(N^0, \ell) = (32, 3)$  runs. The thresholds for the other grids were then set in the same fashion as for the previous test, except that, due to the different scaling of the error on the discontinuities, the correct scaling factor is now 2 (and not 8) per each level added [PS11, SCR15]. The values of  $S^{\text{ref}}$  actually employed are gathered in table 5. However we will focus on results produced by the bold faced values.

$(N^0, \ell)$	<b>Threshold value.</b> Base: $\epsilon = 10^{-3}$		
	Run 1	Run 2	Run 3
(32,3)	<b><math>\epsilon</math></b>	0.1 $\epsilon$	0.01 $\epsilon$
(32,4)	<b><math>\epsilon/2</math></b>	0.1 $\epsilon/2$	0.01 $\epsilon/2$
(32,5)	<b><math>\epsilon/4</math></b>	0.1 $\epsilon/4$	0.01 $\epsilon/4$

Table 5: Isentropic vortex problem — Threshold values used for different refined meshes. We focus on bold faced value simulations.

### 5.3.1. Uniform grids

In figure 10 we present the density for CWENO (left) and MOOD with  $u_2$  on entropy (middle) or  $u_2$  on density (right) schemes. Each panel is decomposed into four quadrants and each quadrant is dedicated

to one particular uniform mesh:

NW $N = 256$	NE $N = 32$
SW $N = 128$	SE $N = 64$

. As such in figure 10 we have plotted the results for

12 simulations. The same color scale is used for all top panels where the density is displayed. Moreover the MOOD marker is also plotted on bottom panels. Red cells are problematic ones and therefore updated by a first order scheme, while blue ones are updated with the unlimited third-order accurate scheme. Note that the  $256 \times 256$  mesh is not plotted for the NW panel to ease the visualization. We observe that the troubled cells are on average located along the main waves but the detection criteria, density or entropy may produce different patterns. The density criteria seems to better capture the cylindricity of the wave. Note that when enough dissipation is deposited, then the detection criteria may not mark any cell for limiting anymore, therefore not marking the cells on known discontinuous waves does not imply that the detection fails, rather it implies that, at this particular time of visualization, the scheme does not consider those cells as problematic. As such the very next time step could produce different patterns of troubled cells.

Next in figure 11 we present the density as a function of the cell center radius for all cells and all meshes. Moreover we focus on the contact and shock zones in the middle and bottom panels of this figure. From these plots we can observe that, as expected when the mesh is refined the waves (contact, shock) and the tail/head of the rarefaction are sharper. It is also clear that for any scheme or mesh the waves seem to converge to the same place, which is adequate according to the reference solution. On average, the same defaults can be attributed to all schemes: when an excessive diffusion is not observed ( $N \geq 128$ ) then over- and under-shoots occur. While these are not observed at the same location for the three tested scheme, nonetheless they are a general characteristics.

At last, we present the error in  $L^2$  and  $L^\infty$  norms in table 6. Because of the presence of discontinuities we can expect a maximal first order of convergence of all schemes on this test for  $L^2$  norm. The data in this table show that the three schemes provide the same accuracy even if MOOD schemes seem slightly more accurate. As expected the errors decrease almost systematically in  $L^2$  and  $L^\infty$  norms. Due to the presence of discontinuity we can not expect a convergence in  $L^\infty$  norm at any rate, as observed.

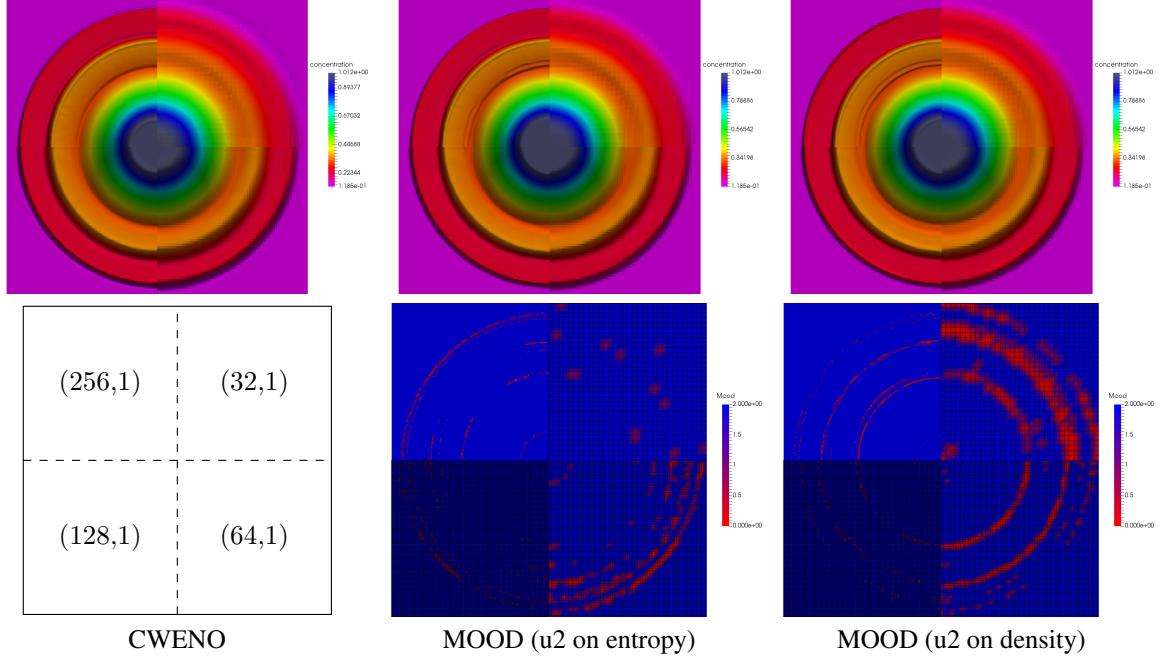


Figure 10: 2D radial Sod shock tube problem at  $t_{\text{final}} = 0.2$  — Uniform  $N \times N$  quadrangular mesh with  $N = 32$  (NE), 64 (SE), 128 (SW), 256 (NW) — Top panels: Colored density as a function of cell center location — Bottom panels: meshes and valid/traubled cells (blue/red). Left: CWENO, middle: MOOD with  $u_2$  on entropy, right: MOOD with  $u_2$  on density results.

$N_c$	CWENO		MOOD density		MOOD entropy	
	$L^2$ error	$L^\infty$ error	$L^2$ error	$L^\infty$ error	$L^2$ error	$L^\infty$ error
$32^2$	6.21E-02	—	8.82E-02	5.18E-02	—	7.50E-02
$64^2$	4.01E-02	0.63	6.19E-02	3.48E-02	0.57	5.95E-02
$128^2$	2.70E-02	0.57	5.85E-02	2.41E-02	0.53	5.78E-02
$256^2$	1.84E-02	0.55	5.40E-02	1.66E-02	0.54	5.48E-02
$512^2$	1.21E-02	0.60	4.86E-02	1.09E-02	0.61	4.71E-02
Expected order	1			1		1

Table 6:  $L^2$  and  $L^\infty$  errors and convergence rate for the radial Sod problem for the MOOD  $\mathbb{P}_2 \rightarrow \mathbb{P}_0$  schemes (entropy or density detection) and the CWENO  $\mathbb{P}_2$  scheme.

### 5.3.2. AMR grids

Next we employ the AMR procedure and produce the results for CWENO and MOOD schemes. In

figure 13 we present the results in four quadrants for each panel:

NW Unif.256	NE (32, 3)
SW (32, 5)	SE (32, 4)

can see, from top to bottom, the colored density, the troubled cells detected by MOOD and the meshes. The same three schemes are tested as before. Recall that the uniform mesh result with  $N = 256$  (NW) must be compared to the AMR results (32, 4) (SE). The first obvious comment is related to the number of cells, which is drastically reduced with the AMR procedure. Moreover the refinement zones seem to be located

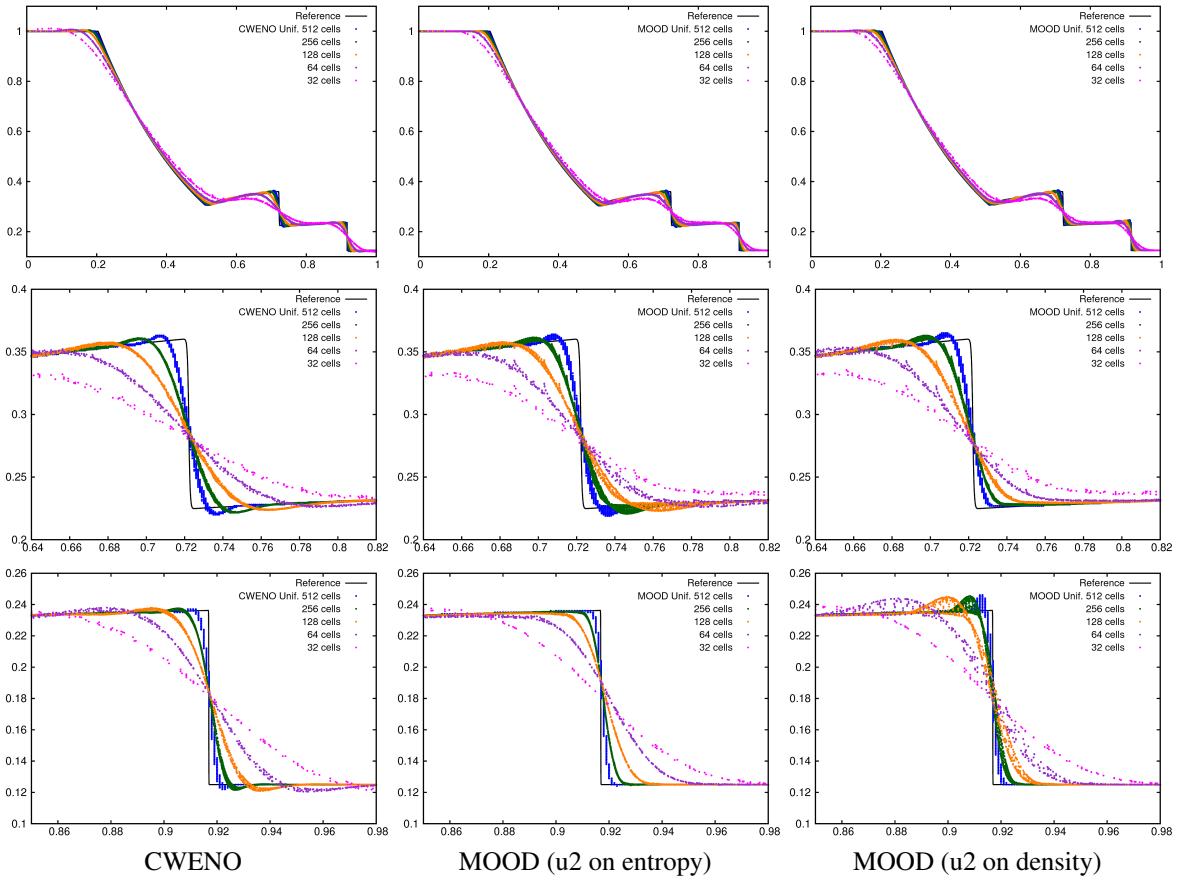


Figure 11: 2D radial Sod shock tube problem at  $t_{\text{final}} = 0.2$  — Uniform  $N \times N$  quadrangular mesh with  $N = 32, 64, 128, 256$  — Density as a function of cell center radius for all cells — Left: CWENO, middle: MOOD with  $u_2$  on entropy, right: MOOD with  $u_2$  on density — Top: full view — Middle: Zoom on the contact — Bottom: Zoom on the shock.

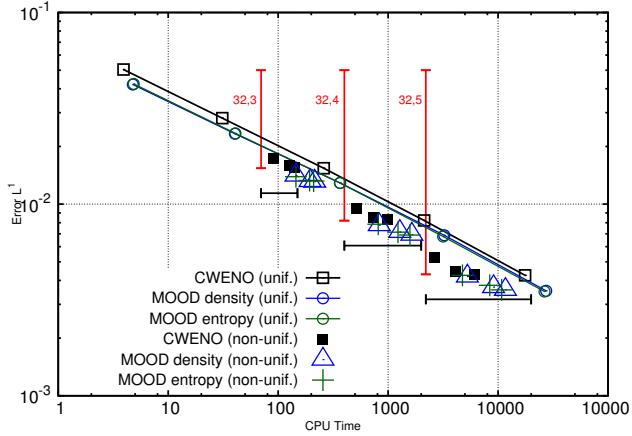


Figure 12: Radial Sod problem —  $L^1$  error curves for CWENO and MOOD simulations as a function of CPU time — Results for uniform (line) and AMR (symbols) grids for all simulations — Each symbols corresponds to a specific threshold parameter, while each set of three symbols employs a grid made of (32, 3), (32, 4) or (32, 5) cells.

along cylindrical waves as expected, apart from unexpected patches in the MOOD results certainly due to a non-perfect choice of the threshold parameters. The troubled cells detected by MOOD procedure (middle panel) are also properly located along these waves, meaning that few cells are actually limited. All blue cells are updated with the unlimited third-order accurate scheme. These behaviors tend to prove that the AMR results are equivalent in terms of accuracy to the uniform grid simulations for the previous section. The AMR procedure couples nicely with the *a posteriori* MOOD limiting, even if needlessly refined zones can be sometimes observed. Next in figure 14 we plot the 1D results as a function of the cell radius following the display of figure 11. The same conclusion as in the uniform case holds.

To assess that the AMR simulations are as accurate as uniform ones, we observe their comparison in figure 15 for the MOOD scheme with  $u_2$  on entropy. Both simulations are visually extremely close, meaning that there is no obvious loss of accuracy on the discontinuous waves. We do not report zooms on rarefaction fan but the results are alike. Moreover the same behavior is observed for CWENO and MOOD with  $u_2$  on density schemes, these results are omitted accordingly.

Visually we have observed that AMR and uniform mesh simulations seem to perform alike in terms of accuracy. Moreover the final spacial distribution of troubled cells seem to be coherent. In figure 16 we present, for MOOD schemes ( $u_2$  on entropy or density), the histograms of the number of cells  $N^k$  in each of the refinement level as a function of the time iteration for the cases  $\ell = 3$ ,  $\ell = 4$  and  $\ell = 5$ . We observe that the majority of cells are on the finest level in all cases. The number of cells in all other levels are slightly increasing during the simulation apart from the number in the initial level. The initial linear increase in the total number of cells is consistent with the smallest cells being located on the discontinuities, which are expanding during time evolution. Later on, the number of small cells levels out as a result of the balance between new ones being created along the expanding shock and contact discontinuity and other ones being coarsened in the smooth part of the rarefaction.

Note that the number of time-step is approximately doubled from one simulation to the next one which is coherent with the fact that the smallest cell characteristics length is divided by two. We observe that the results for the CWENO present the same behaviors and we omit them. On the same figure the number of bad cells detected by MOOD are represented with red symbols, (the axis is on the right of the figures in red). After the same high increase around the 50th iteration, the number of troubled cell seems to approximately follow the behavior of the number of fine cells.

In figure 17 we plot the evolution of the total number of troubled cells as a function of the time. For the

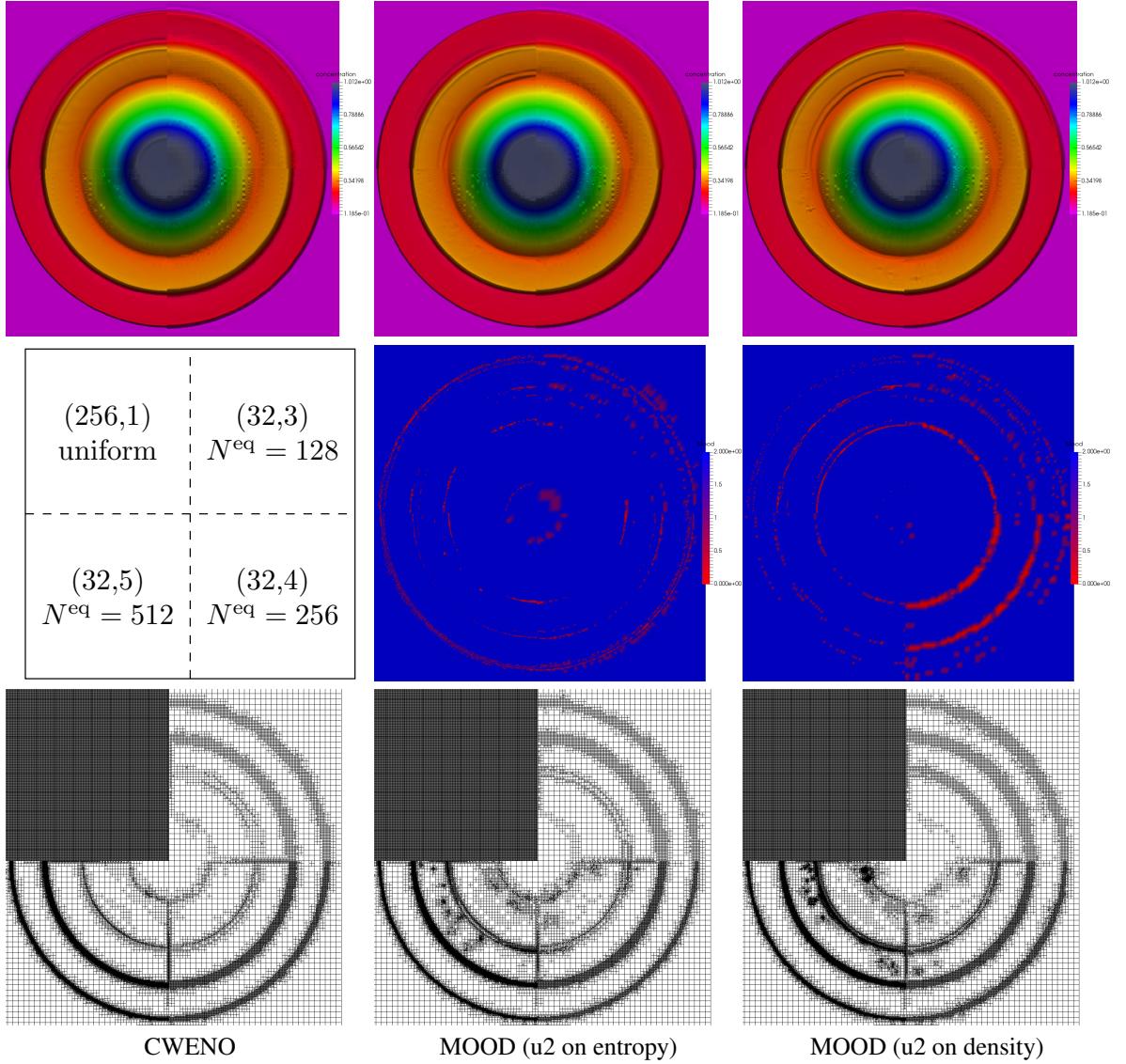


Figure 13: 2D radial Sod shock tube problem at  $t_{\text{final}} = 0.2$  — AMR simulations with  $N^0 = 32$  and  $\ell = 3$  (NE),  $\ell = 4$  (SE),  $\ell = 5$  (SW), and uniform  $256 \times 256$  mesh results (NW) — Colored density as a function of cell center location (top), MOOD marker (middle) and meshes — Left: CWENO, middle: MOOD with  $u_2$  on entropy, right: MOOD with  $u_2$  on density.

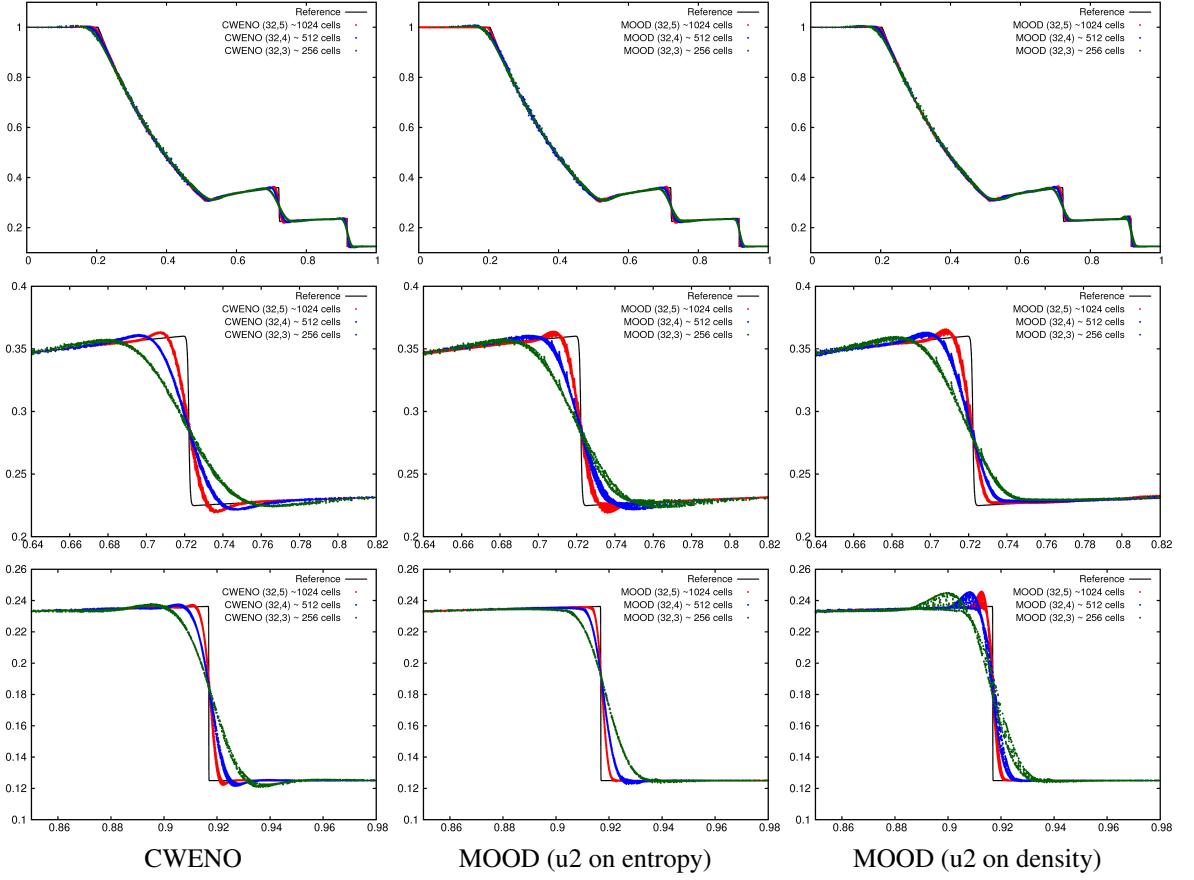


Figure 14: 2D radial Sod shock tube problem at  $t_{\text{final}} = 0.2$  — AMR simulations with  $N^0 = 32$  and  $\ell = 3$  (NE),  $\ell = 4$  (SE),  $\ell = 5$  (SW), and uniform  $256 \times 256$  mesh results (NW) — Density as a function of cell center radius for all cells — Left: CWENO, middle: MOOD with  $u_2$  on entropy, right: MOOD with  $u_2$  on density — Top: full view — Middle: Zoom on the contact — Bottom: Zoom on the shock.

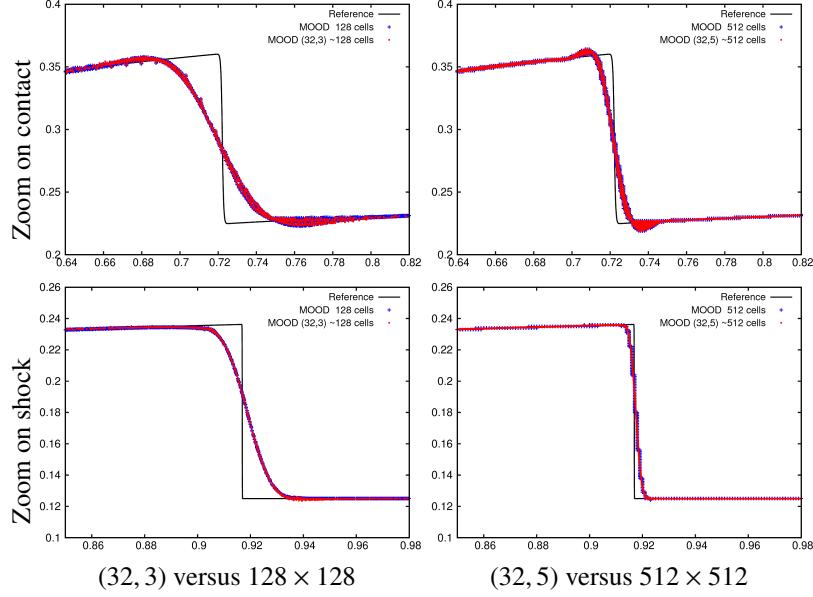


Figure 15: 2D radial Sod shock tube problem at  $t_{\text{final}} = 0.2$  — MOOD with  $u_2$  on entropy simulations — AMR simulations (red symbols) versus uniform ones (blue symbols) — Density as a function of cell center radius for all cells, zooms on contact (top panels) and shock (bottom panels) waves — Top:  $(32, 3)$  versus  $128 \times 128$  uniform mesh — Bottom:  $(32, 5)$  versus  $512 \times 512$  uniform mesh.

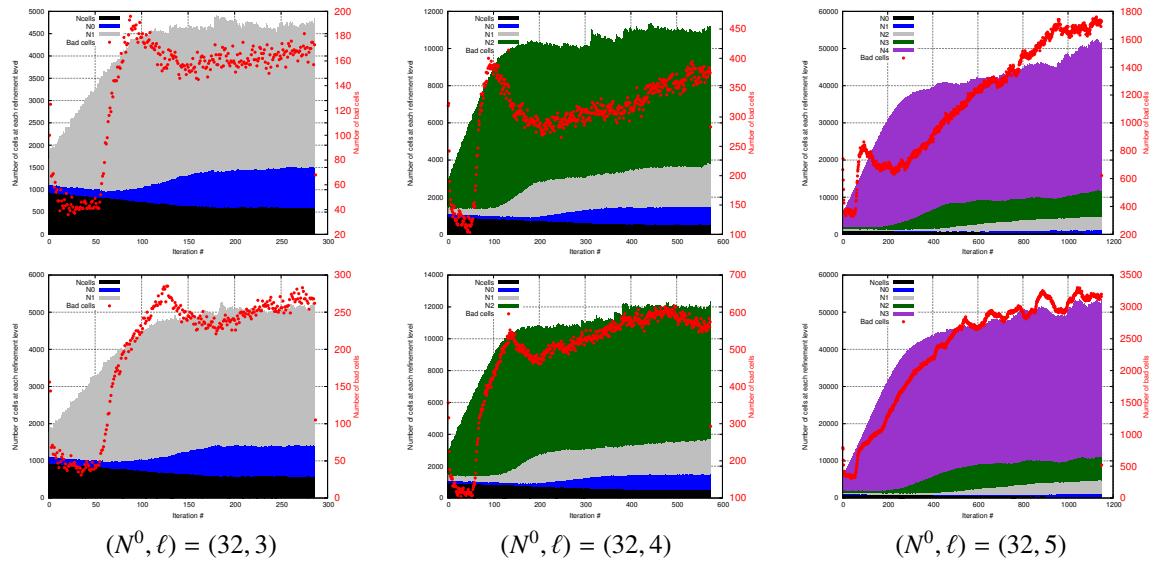


Figure 16: 2D radial Sod shock tube problem — MOOD scheme with  $u_2$  on entropy (top panels) or density (bottom panels) — Results for a mesh with 3 (left), 4 (middle) or 5 (right) levels of refinements — Colored histograms for the number  $N_k$  ( $k = 0, \dots, 4$ ) of cells of each level (black vertical scale on the left) and number of bad cells (red symbols and red vertical scale on the right) as a function of time iteration.

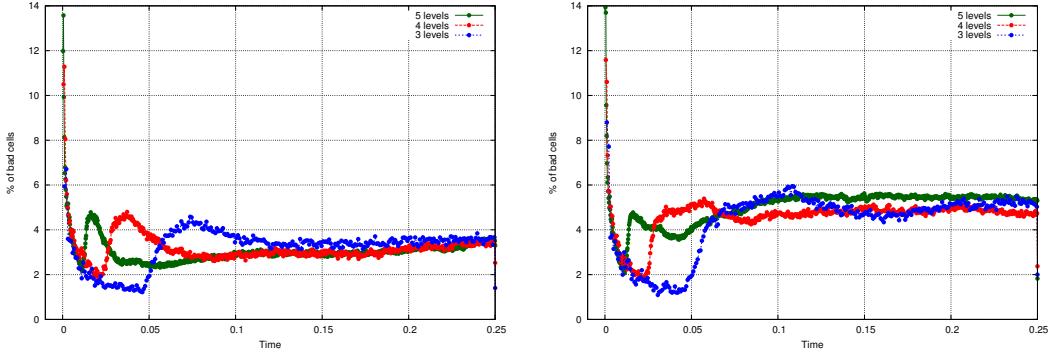


Figure 17: 2D radial Sod shock tube problem — Evolution of the total number of bad cells as a function of time for the MOOD schemes using 3, 4 or 5 levels of refinements and  $N^0 = 32$ . Left: MOOD with  $u_2$  on entropy — Right: MOOD with  $u_2$  on density.

two MOOD schemes the percentage of limited cells is of the order 3% to 5% apart from the first iterations for which the percentage can reach up to 14%. As such, for this test we conclude that the MOOD limiting procedure acts on a small amount of cells for the most part of the simulations. Note that in the last time-step much less troubled cells are detected. This is due to the fact that typically this time-step is shorter than the  $\Delta t$  allowed by the CFL condition, since it is cut short to meet exactly the final time for this simulation.

In table 7 we report the final number of cells  $N_c(t)$ ,  $L^2$  and  $L^\infty$  errors and, CPU time needed by the three schemes under AMR strategy. (Recall that  $(N^0, 1)$  simulations are uniform grid simulations.) Moreover we have computed the acceleration (Acc.), that is the ratio between the CPU time needed to solve the  $N^{eq}$  uniform grid simulation, and the current CPU time needed by the AMR simulation. We can note that for the AMR simulations, the number of cells may be truly different even if the threshold parameters are the same for given  $(N^0, \ell)$ . For instance MOOD with  $u_2$  on entropy is the scheme which refine the most, CWENO the less. Recalling that the AMR is driven by the numerical entropy production, it is to be expected that different spatial discretizations lead to different levels of entropy production and thus to different refinement pattern. Of course, one could obtain a MOOD simulation with the same number of cells as a CWENO one by tweaking the AMR threshold parameter. More important is the fact that the errors in both norms are of the same order for all scheme at equivalent mesh resolution. But the acceleration is about 2, 4 and between 5 and 6 for respectively  $\ell = 3, 4$ , and 5. This means that the more levels of refinement allowed, the more efficient the AMR strategy is. The gain in terms of number of cells oscillates between 2 and 10 at final time, whereas at  $t = 0$ , these ratio of cells are of the order 7, 17 and 36 with respect to  $\ell$ . The  $L^1$  error norm

$(N^0, \ell), N^{eq}$	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU	Acc.	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU	Acc.	$N_c(t)$	$L^2$ error	$L^\infty$ error	CPU	Acc.
CWENO															
(128,1),128	16384	2.70E-02	5.85E-02	259.8	—	16384	2.41E-03	6.30E-02	365.8	—	16384	2.42E-03	5.78E-02	364.5	—
(32,3),128	4540	1.66E-02	5.88E-02	91.4	2.8	5251	1.53E-02	6.13E-02	145.3	2.5	4852	1.33E-02	5.81E-02	145.1	2.5
MOOD $u_2$ on density															
(256,1),256	65536	1.84E-02	5.40E-02	2126.2	—	65536	1.66E-02	5.75E-02	3242.5	—	65536	1.65E-02	5.48E-02	3165.1	—
(32,4),256	10936	6.90E-03	5.41E-02	517.8	4.1	9709	7.41E-03	5.64E-02	830.4	3.9	18910	5.94E-03	5.39E-02	813.2	3.9
MOOD $u_2$ on entropy															
(512,1),512	262144	1.21E-02	4.86E-02	17618.2	—	262144	1.09E-02	5.08E-02	27384.7	—	262144	1.08E-02	4.71E-02	26456.8	—
(32,5),512	25054	3.76E-03	4.83E-02	2621.05	6.7	32011	3.18E-03	5.10E-02	5258.2	5.2	51796	3.07E-03	4.62E-02	4744.5	5.6

Table 7: Final number of cells  $N_c(t)$ ,  $L^2$  and  $L^\infty$  errors, CPU time, for the radial Sod problem simulated by several 3rd order schemes using AMR capability. Results for the CWENO and MOOD with  $u_2$  detection or density schemes.  $N^0$  is the number of cell in each direction for the initial grid,  $\ell$  if the number of refinement level leading to  $N^{eq}$ , the number of cell for each direction in an equivalent fixed grid. Acc. is the acceleration, i.e. the ratio between the CPU time (needed to solve the  $(N^0, 1)$  uniform grid simulation) and the current CPU time (ie needed for the AMR  $(N^0, \ell)$  simulation).

obtained from different threshold values, see table 5, as a function of the CPU time needed to complete the simulation are reported in figure 12 with symbols. This figure proves that the AMR simulations are

more efficient than uniform ones. This also shows that MOOD simulations are slightly more efficient than CWENO.

This test has shown that in the case of non smooth problem the numerical scheme supplemented with MOOD is able to maintain its essential non-oscillatory property at reasonable cost. Compared to classical CWENO strategy from [SCR15], MOOD approach does not behave as nicely with the AMR framework, mostly due to the fact that best working parameters for CWENO do not coincide with those for MOOD. Nevertheless, this issue is classical for any scheme working under AMR. We have observed that the number of *a posteriori* detected bad cells which are further recomputed with a first order accurate scheme is of the order of few percents. The efficiency of MOOD is however shown as the errors as a function of CPU time is generally lower than CWENO. Because none of the two MOOD schemes based either on entropy or density variable is clearly superior, we only present results obtained with limiting on density for the rest of the paper. The radial Sod shock tube has assessed the ability of the method to capture non-aligned smooth and discontinuous waves in an accurate way by automatically refining the mesh in the vicinity of those waves. The solution is accurately captured and the AMR strategy with MOOD approach provide an efficient couple to tackle more advanced problems. For the next cases, we increase the difficulty in order to observe how the numerical scheme will adapt when robustness is the main criteria and accuracy a secondary one.

#### 5.4. Double Mach reflection problem

The 2D double Mach reflection problem of a strong shock was proposed in [WC84]. This problem involves a Mach-10 traveling shock in a perfect gas ( $\gamma = 1.4$ ) which further hits a ramp at  $30^\circ$  with the  $x$ -axis. The initial conditions in front of and after the shock wave are given by

$$(\rho, u, v, p)(\mathbf{x}, t = 0) = \begin{cases} \frac{1}{\gamma}(8.0, 8.25, 0.0, 116.5), & \text{if } x' < 0.1, \\ (1.0, 0.0, 0.0, \frac{1}{\gamma}), & \text{if } x' \geq 0.1, \end{cases} \quad (17)$$

where  $x'$  is the coordinate in a rotated coordinate system. Reflecting wall boundary conditions are prescribed on the bottom and the exact solution of an isolated moving oblique shock wave with Mach number  $M_s = 10$  is imposed on the upper boundary. Inflow and outflow boundary conditions are set on the left and the right sides. The computational domain is  $\Omega = [0; 3.5] \times [0; 1]$  and the problem has been rotated so that the ramp is aligned with the  $x$  axis.

The mesh is built starting from a uniform coarse grid employing  $M = 30$  square cells on length  $[0; 1]$  leading to an effective length  $h = 1/M$ . Next, the grid is adapted to the initial condition with 4, 5 and 6 levels of refinement, leading to 4954, 7411 and 12247 squares respectively, see figure 18 for zooms. Consequently we employ the AMR grids  $(N^0, \ell) = (30, 4), (30, 5)$  and  $(30, 6)$  which should be equivalent to the fixed grids made of  $840 \times 240$ ,  $1680 \times 480$  and  $3360 \times 960$ . The threshold value is fixed to 0.05 for  $\ell = 4$  and consequently to 0.025 for  $\ell = 5$  and 0.01 for  $\ell = 6$  in order to refine approximately on the same features also in the finer runs. The MOOD detection criteria considers the DMP+u2 criteria on the density variable only.

The results are presented in color for the density variable at  $t_{\text{final}} = 0.2$ , see figure 19. In the same figure we present also the final AMR meshes along with the location of the bad cells detected by the MOOD procedure (red cells). Zoom are proposed on the bottom panels of the same figure to enhance the meshes, the bad cells and the re-circulation zones. Left panels gather the results for the  $\ell = 5$  and right ones for  $\ell = 6$  levels of refinement.

We observe that one more level of refinement improves the sharpness of the small scale structures such as the shocks, the roll-ups and the reflected/refracted waves. The mesh is appropriately refined to cover the main waves (even if erroneously refined cells can be observed along the boundaries for instance). To enhance the differences between these three simulations we plot the results on figure 20 to compare  $\ell = 4$  with  $\ell = 6$  (left panel) and  $\ell = 5$  and  $\ell = 6$  (right panel). Contrarily to the previous 2D colored view,

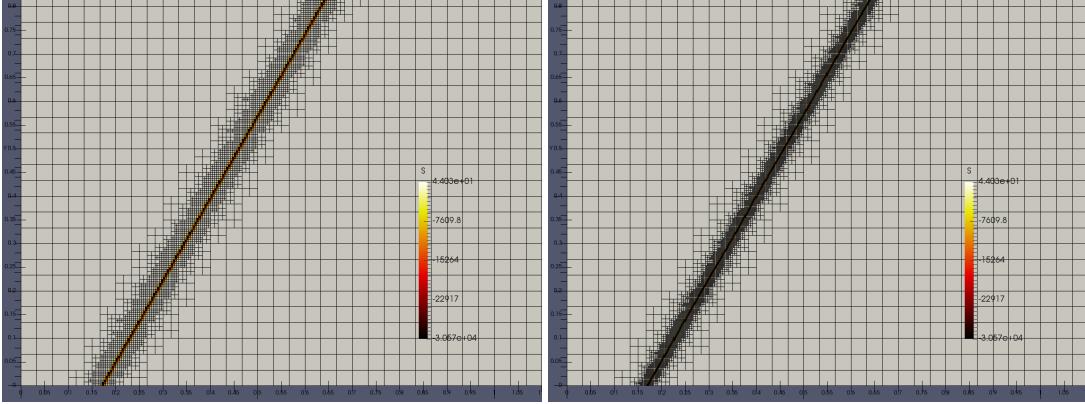


Figure 18: Double Mach reflection problem — Initial adapted AMR grid ( $N^0, \ell$ ) starting from a coarse mesh of characteristics length  $h = 1/N^0$  and  $\ell$  levels of refinement — Left: situation  $(N^0, \ell) = (30, 4)$  — Right: situation  $(N^0, \ell) = (30, 5)$ .

here the data are extruded in 3D and the plots are seen from above with light, enhancing the steep gradients with shadowing. We can clearly observe that the discontinuities are sharper and some waves are (better or simply) captured with a finer mesh. Most of the waves are Kelvin-Helmholtz unstable and only the finest resolution starts to capture those instability. Also some spurious effects seems more present with finer mesh. This behavior is consistently observed for intermediate times (not shown here).

Next we focus on diagnostics on the number of cells of each level  $N^k$ ,  $k = 0, \dots, 4$  or 5. On figure 21 we present the histograms for each  $N^k$  as a function of the time iteration for the cases  $\ell = 4$ ,  $\ell = 5$  and  $\ell = 6$ . We observe that the majority of cells are on the finest level in all cases, and this number increases in time as more structures to be refined appear and/or get bigger. The number of cells in all other levels are slightly increasing during the simulation. Note that the number of time-step is approximately doubled from one simulation to the next one. On the same figure we also plot with red symbols the number of bad cells detected by MOOD (red axis on the right of the figures). The number of troubled cell seems to increase with the number of fine cells.

Next in figure 22 one presents the percent of bad cells at each time-step for the three simulations. The full view is presented on the left panel while a zoom is proposed on the right one. As expected the percent of troubled cells detected by MOOD and further updated with a first order scheme, is of the order of 3% and exceed 7% only for very few first iterations after initialization.

Next in table 8 we gather some diagnostics for these AMR meshes: number of cells (final, average and

$(N^0, \ell)$	$N^{eq}$ unif.	$N_c(t), \bar{N}_c, N_c(t=0)$	Ratio	CPU	CPU unif.	Ratio
(30,4)	$840 \times 240 \approx 2 \times 10^5$	30253, 17982, 4954	7, 11, 41	3.3h (3h)	$\approx 16.4h$ (16h)	5
(30,5)	$1680 \times 480 \approx 8 \times 10^5$	79549, 46310, 7411	10, 17, 109	20h (1d)	$\approx 149h$ (6d)	7.5
(30,6)	$3360 \times 960 \approx 32 \times 10^5$	274989, 136978, 12247	12, 24, 263	187h (8d)	$\approx 1346h$ (56d)	7

Table 8: Double Mach Reflection problem — Equivalent uniform runs; final, average and initial number of AMR cells (and ratio); CPU times of the AMR run, expected CPU time of the equivalent uniform run (extrapolated) and ratio.

initial), ratio versus the equivalent uniform number of cells, the CPU time, the CPU time for a uniform equivalent number of cells, and the ratio of CPU time. Our simulation code demands 3h19mn for  $\ell = 4$ , 19h55mn for  $\ell = 5$  and 187h20mn for  $\ell = 6$  on a single CPU. In order to compare the execution times of AMR and uniform runs, we measured the execution times of the (30, 1), (60, 1) and (120, 1) runs and extrapolated the data to the  $N^{eq}$  grids. The ratio is around 5 for the (30, 4) grid and 7 for the two finer runs, indicating that AMR simulations is five to seven times faster than uniform simulations. The efficiency of the

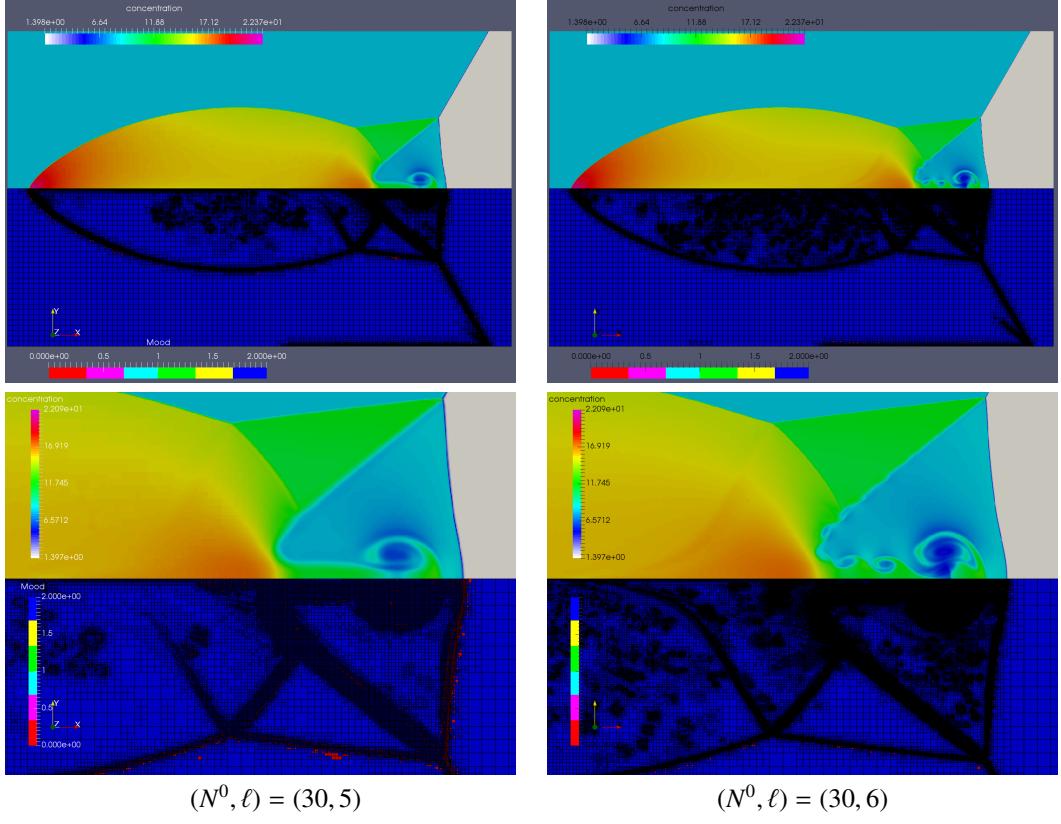


Figure 19: Double Mach reflection problem at  $t_{\text{final}} = 0.2$  with AMR — All panels: final colored density (top part) and mesh and troubled cells (bottom part) — Results for a mesh with  $\ell = 5$  (left), and  $\ell = 6$  (right) levels of refinements. Top: full view, bottom: zoom.

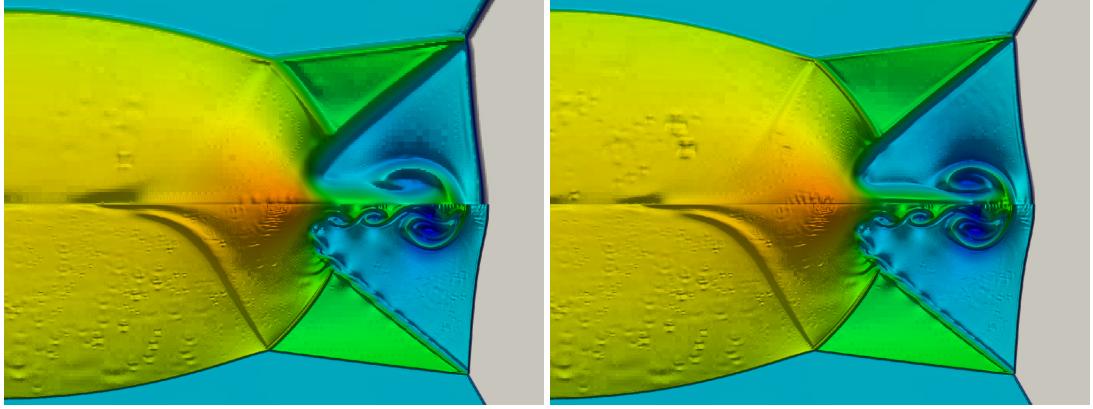


Figure 20: Double Mach reflection problem at  $t_{\text{final}} = 0.2$  with AMR — Density for a mesh with 4 (left-top), 5 (right-top) and 6 (left-right-bottom) levels of refinements.

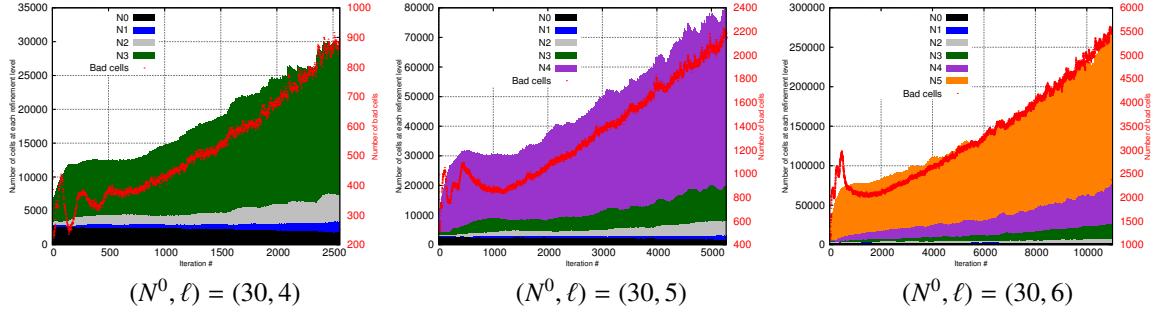


Figure 21: Double Mach reflection problem at  $t_{\text{final}} = 0.2$  with AMR — Results for a mesh with 4 (left), 5 (middle) or 6 (right) levels of refinements — Colored histograms for the number of cells of each level ( $N_k$ ,  $k = 0, \dots, 4$ ) and number of bad cells (red symbols) as a function of time iteration

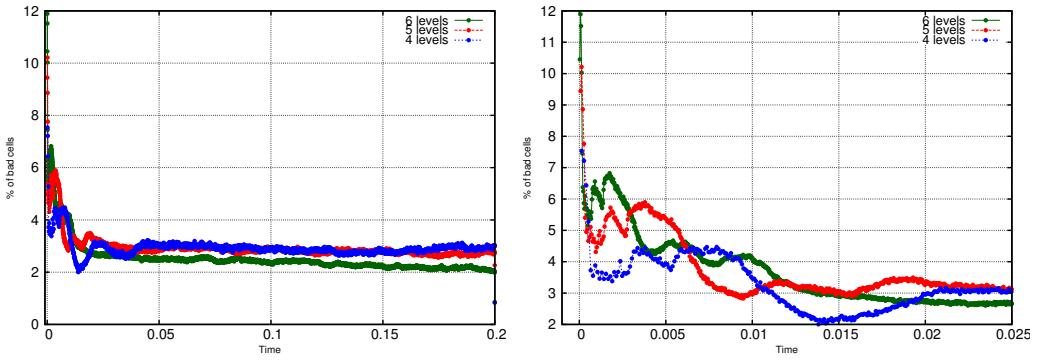


Figure 22: Double Mach reflection problem at  $t_{\text{final}} = 0.2$  with AMR — Results for a mesh with 4 (left), 5 (middle) or 6 (right) levels of refinements — Percent of bad cells detected by MOOD loop as a function of iteration (left) and time (right). Left: full view — Right: zoom on the first 700 iterations.

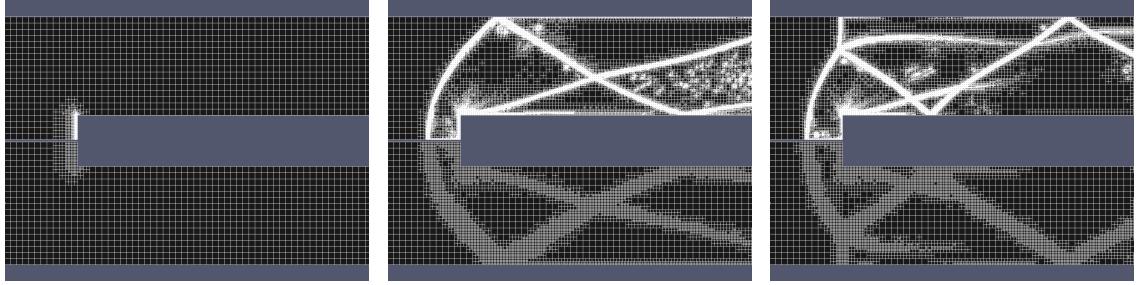


Figure 23: Forward Facing Step problem — Refined meshes with  $\ell = 3$  (bottom part of each panel) and  $\ell = 5$  (top parts) levels of refinement — Initial, intermediate ( $t \approx 1.2$ ) and final refined meshes.

AMR scheme is linked to its ability to perform the computation with 10 to 25 times less cells on average. The ratios of AMR cells versus  $N^{\text{eq}}$  keeps increasing when finer and finer levels are allowed because the main waves in this problem are well separated.

This difficult problem exhibits strong shock waves and smooth flow features. The detection procedure properly identifies the discontinuities associated with the shock waves. However, the *a posteriori* detection procedure is able to ignore regions of complex but nonetheless smooth vortex-type flow region, thus maintaining the maximum accuracy of the scheme. Moreover the AMR procedure keeps the finest allowed mesh in this zone and in the vicinity of strong waves, leading to the maximal possible accuracy. Note that falsely detected cells are sometimes associated to falsely refined regions. While this does misuse computer resources, it does not seem to pollute the numerical solution.

### 5.5. Forward facing step

The forward facing step (FFS) problem is considered next, see [WC84]. This problem describes a Mach 3 wind tunnel with a step filled with a uniform gas with density  $\rho = \gamma$ , pressure  $p = 1$ , velocity components  $u = 3, v = 0$  and  $\gamma = 1.4$ . The computational domain is fixed to  $\Omega = [0; 3] \times [0; 1] \setminus [0.6; 3] \times [0; 0.2]$ . Reflective wall boundary conditions are applied on the upper/lower boundaries of the domain, whereas inflow and outflow boundary conditions are imposed at the entrance and the exit, respectively on the left and right parts of the computational domain. The final time is set to  $t_{\text{final}} = 4$ . As reported in the literature, the solution of this problem involves shock waves interacting with themselves after bouncing on the boundaries. No exact solution exists nowadays, but reference ones can be found in the literature.

The initial coarse mesh is uniform with characteristics length  $h = 1/20$  in both directions. Next it is refined up to the finest level close to the corner of the domain from where the first fluid structures will emanate. We employ  $\ell = 3$  or 5 levels of refinement leading to an equivalent mesh made of about  $h' = \frac{h}{2^\ell}$ , that is to say  $h' = \frac{h}{8}$  or  $h' = \frac{h}{32}$ , see figure 23-left panels. In this section, for comparison purposes we plot the case  $\ell = 5$  on the top part of each panel and the case  $\ell = 3$  is symetrized with respect to the  $x$  axis, therefore being plotted on the bottom part of each panel, see figure 23. The threshold parameter is fixed to  $S^{\text{ref}} = 0.05$  and 0.0125 for the two refinements respectively.

In figure 23 are plotted the initial, intermediate ( $t \approx 1.2$ ) and final refined meshes. As expected the refinement zones follow the main waves and structures and the  $\ell = 5$  level simulation is a lot sharper in capturing those dynamical elements of the flow. In figure 24 we display the results for three times  $t = 0.4$ ,  $t = 1.2$  and final time  $t = 4.0$  from top to bottom. Left panels present the density variable, the middle ones the bad cell MOOD marker, and the right ones the density and mesh in a 3D elevation fashion to enhance the location and the complex structures of the flow, and the fact that the mesh is refined in appropriate locations. From this figure we can clearly see several smooth zones separated by the shock waves. The refinement regions are properly covering the waves and adapt to their motion in time. As expected, the great majority

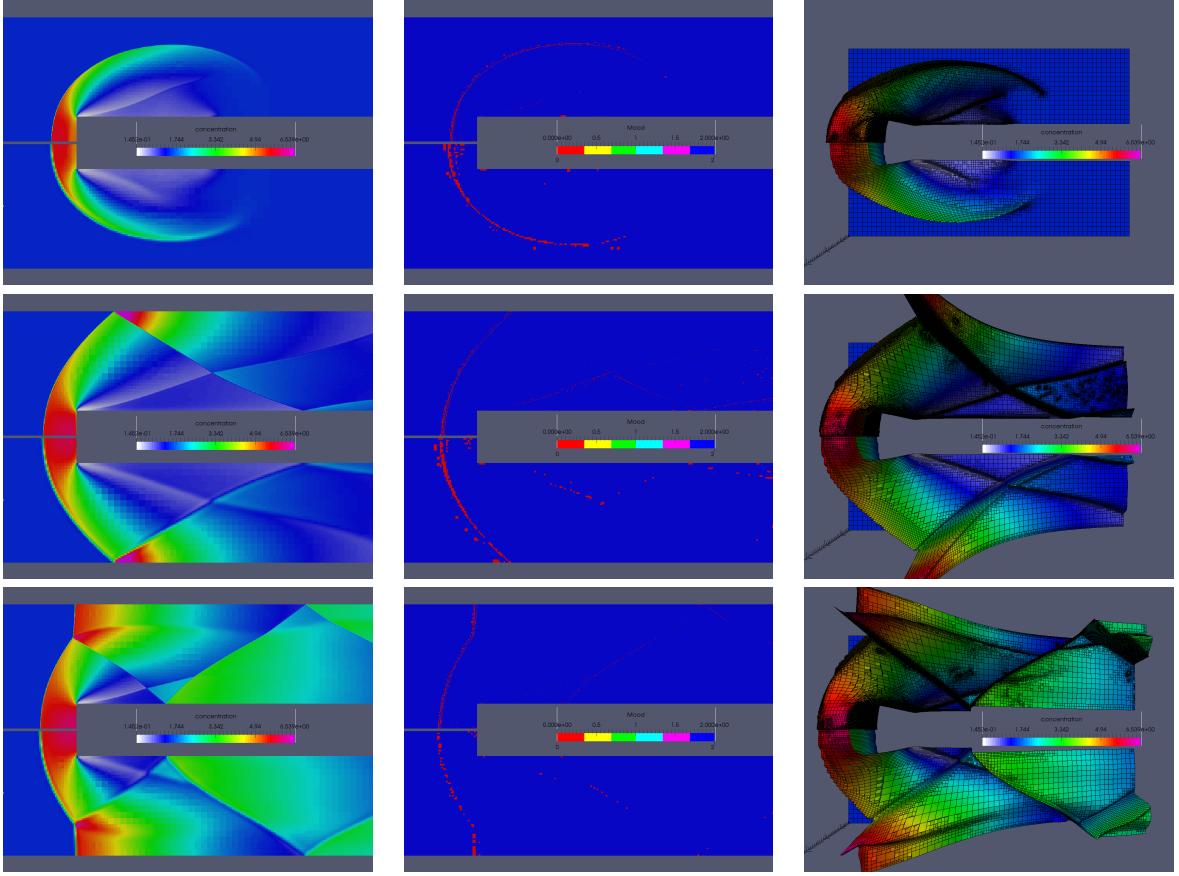


Figure 24: Forward Facing Step problem — Results for  $\ell = 3$  (bottom of each panel) and  $\ell = 5$  (top of each panel) levels of refinement — From top to bottom: data at time  $t = 0.4$ ,  $t = 1.2$  and final time  $t = 4.0$  — Left: density variable — Middle: bad cell MOOD marker — Right: 3D elevation of density and mesh.

of the computational domain is computed with the unlimited 3rd order accurate scheme (blue cells in middle panel), apart from the main and strongest shock waves, where the troubled cells are mostly located (red cells). The fact that MOOD is not decrementing most of the cells in the secondary apparently discontinuous waves means that, at this time, the numerical scheme has already deposited enough numerical dissipation to produce an acceptable solution even if 3rd order accurate unlimited polynomial reconstructions are employed.

Following the previous test case, we plot the number of cells of each level  $N^k$ ,  $k = 0, \dots, 2$  or 4. On figure 25, left and middle panels, we present the histograms for each  $N^k$  as a function of the time iteration for the case  $\ell = 3$  and  $\ell = 5$ . We observe that the majority of cells are on the finest level in both cases, and this number stagnates in time as soon as the main structures are in place, approximately after 1/5th of the total simulation time. The number of cells in all other levels follows the same behavior. As expected the number of time-step is approximately multiplied by a factor 4 from one simulation to the other, from 12685 to 51646 iterations due to the two extra levels of refinement when  $\ell = 5$ . With red symbols one represents the evolution of the number of bad cells detected by MOOD (red axis on the right of the figures). The number of troubled cell seems to follow the evolution of the number of fine cells. On the right panel we present the percent of bad cells at each time-step for both simulations which is of the order of 2% and

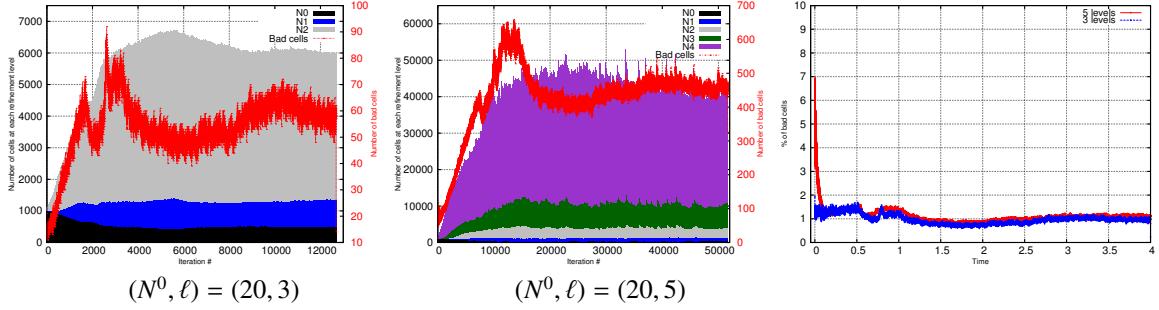


Figure 25: Forward Facing Step problem — Results for a mesh with 3 (left) or 5 (middle) levels of refinements — Left-middle: Colored histograms for the number of cells of each level ( $N_k$ ,  $k = 0, \dots, 2$  or 4) and number of bad cells (red symbols) as a function of time iteration — Right: percent of bad cells detected by MOOD loop at each time-step.

never exceeds 7%. As expected the percent of troubled cells (detected by MOOD and recomputed) is very low. This was already observable on figure 24 on the middle panels.

$(N^0, \ell)$	$N^{eq}$ unif.	$N_c(t), \bar{N}_c, N_c(t = \Delta t)$	Ratio	CPU	CPU unif.	Ratio
(20,3)	16128	5977, 5694, 1225	2.7, 2.8, 13	2.8h	21.1h (1d)	7.5
(20,4)	64512	15208, 14895, 1465	4.2, 4.3, 44	17h(3/4d)	373h (2w)	22
(20,5)	258048	40588, 39523, 1909	6.4, 6.6, 135	130h(5d)	$\approx 6591h$ (9m)	51
					true factor:17.67	
(20,3)	16128	5977, 5694, 1225	2.7, 2.8, 13	2.8h	21.1h (1d)	7.5
(20,4)	64512	15208, 14895, 1465	4.2, 4.3, 44	17h(3/4d)	$\approx 169h$ (1w)	9.9
(20,5)	258048	40588, 39523, 1909	6.4, 6.6, 135	130h(5d)	$\approx 1350h$ (1.5m)	10.4
					theor.factor:8	

Table 9: Forward Facing Step problem — Equivalent uniform runs; final, average and initial number of AMR cells (and ratio); CPU times of the AMR run, expected CPU time of the equivalent uniform run (extrapolated) and ratio (h: hour, d: day, w:week, m:month). Top part: CPU uniform time extrapolated with the true observed factor 17.67 from the first two uniform runs. Bottom part: CPU uniform time extrapolated with the theoretical factor 8.

For this test our simulation code demands about 129h36mn (about 5 days) for  $\ell = 5$  and 40000 cells, and about 2h49mn for  $\ell = 3$  and 6000 cells. In table 9 are reported some timing statistics about these runs. We point out that the initial data for this test is constant in space and thus the simple gradient-based indicator is not able to refine the mesh at time  $t = 0$ . All AMR runs therefore start with a uniform mesh of 1009 cells and the initial mesh adaption is entirely done during the first time-step. The data reported for the initial mesh size in the table and the mesh depicted in the left panel of Figure 23 refer thus to the mesh at the end of the first time-step. The data in this table show that the ratio of mesh sizes is of the order 3 to 6.5 and the acceleration is of the order 10 of the AMR simulations compared to uniform ones. The simulation times with the uniform mesh should scale with a factor 8 (bottom part of the table), leading to ratio of the order 7 to 10. However, for unclear reasons, the observed uniform mesh simulation times are larger, see the top part of the table, and seem to scale with a factor 17.67, leading to ratio up to 50. In both cases, however, the AMR simulations spare computer resources.

These results assess that the AMR simulation tool presented in this paper is well suited to capture physical flow fields associating interacting shock waves as well as fine re-circulation structures. The value of the threshold parameter could be possibly fine-tuned to obtain even better results. However, thanks to

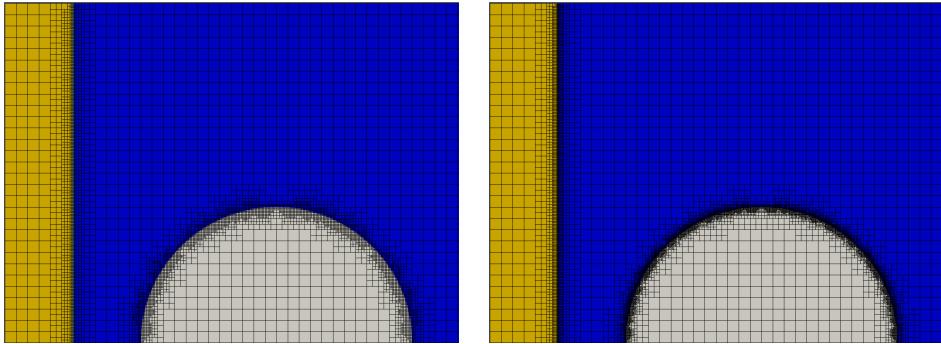


Figure 26: Shock-bubble interaction problem — Zooms on the initial refined meshes when  $\ell = 4$  (left) or  $5$  (right) levels of refinement.

the fail-safe MOOD approach, this parameter does not play a fundamental role to ensure the robustness. As a consequence, for this test, the AMR technology couples well with the *a posteriori* MOOD checking maintaining the code both accurate where possible, and, robust otherwise.

### 5.6. Shock bubble interaction

The last test of this test suite is the classical shock-bubble interaction problem. One considers an initial datum with a right-moving shock that impinges on a standing bubble of gas at low pressure, see [CT09, SCR15]. The computational domain is set to  $\Omega = [-0.1, 1.6] \times [0.5, 0.5]$ . Three distinct areas are considered: the left region (A) for  $x < 0$ , the bubble (B) of center  $(0.3, 0.0)$  and radius 0.2 and the right region (C) of all points with  $x > 0$  that are outside the bubble. The initial data are

$$\begin{aligned} \rho &= \frac{11}{3}, \quad u = 2.7136021011998722, \quad v = 0.0, \quad p = 10.0 && \text{if } (x, y) \in A, \\ \rho &= 0.1, \quad u = v = 0.0, \quad p = 1.0 && \text{if } (x, y) \in B, \\ \rho &= 1.0, \quad u = v = 0.0, \quad p = 1.0 && \text{if } (x, y) \in C. \end{aligned}$$

Boundary conditions are of Dirichlet type on the left, free-flow on the right, reflecting on  $y = \pm 0.5$ . The symmetry in the  $y$  variable permits a half domain computation (with  $y > 0$ ) considering symmetry boundary conditions at  $y = 0$ . The final time is  $t_{\text{final}} = 0.4$ .

Accordingly to the previous tests, we start with a coarse grid made with squares of characteristics length  $h = 1/N^0$  with  $N^0 = 30$ . This mesh is further adapted to the initial condition with  $\ell > 0$  levels of refinements, see figure 26 for the case  $\ell = 4, 5$ . The threshold parameter is fixed to 0.03 for  $\ell = 5$  and 0.1 for  $\ell = 4$ .

We display in figure 27 the results for the case  $\ell = 5$  at three times  $t = 0.2, 0.3$  and final time 0.4 from top to bottom. Left panels present the density in color, the norm of velocity gradient (reflected to  $y = 0$  on gray scale) and the bad cells detected by MOOD (in red color, other gray cells are valid ones). On right panels are plotted the cell refinement levels (top in colors) and the adapted mesh (bottom). Note that in the right panel the same type of information is displayed but in different formats. For these figures we can observe that the deformation of the bubble agrees well with already published results. Moreover the numerical solution accuracy competes with already obtained solution in [SCR15] for instance. The refinement follows the different wave structures (shocks, vortexes, interacting waves, etc) as can be seen on the gradient of the velocity. The results show that the numerical method is able to capture unstable vortex structures and diffracting shock waves. Still the AMR technology is able to keep derefined some of the internal zones between those waves (green and light blue cells in the cell level plot). Moreover, we can observe that the MOOD procedure, which is dedicated to maintain the non-oscillatory behavior of the numerical solution, marks cells along the main up-front discontinuous shock wave. Apart from the main shock region, that is in

the zone after the shock, the MOOD procedure does not require much limiting. Nonetheless, the procedure mostly marks cells along secondary waves. Consequently, most of the cells in the computational domain are effectively calculated with the unlimited third-order scheme, therefore without any extra-cost brought by the limiting apart from the MOOD detection. The refinement and detection strategies seem to perform well and the sharpness of small scale structures can be captured, maintained and released if they were to disappear. The final number of cells reaches about 125000 cells (starting at 12000) for  $\ell = 5$ .

For comparison purposes we propose in figure 28 the same results but with one less level of refinement:  $\ell = 4$ . The same general conclusions as for the  $\ell = 5$  case apply. However it is clear that the sharpness of the fluid structures is less pronounced. The final number of cells reaches about 33000 cells (starting at 7000) for  $\ell = 4$ .

The two final solutions are plotted side-by-side on figure 29 where  $\ell = 4$  (left) and  $\ell = 5$  (right) results are plotted as: 3D elevation colored density seen from above (top part of the panels) and the mesh colored by the density (bottom part). This figure clearly enhances the differences between the two simulations in terms of the number of small scale structures captured when adding one level of refinement.

Next on figure 30 we plot the statistics on the number of cells for each refinement level (left for  $\ell = 4$  and right for  $\ell = 5$ ) and the number of bad cells during the simulation (right panel). First of all, the number of iteration is respectively 4500 and 9000. Consistently with the previous tests, the number of cells for the two finer levels increases. The finest level represents the majority of cells. The number of bad cells after the initial hundreds of iterations is about 350 and 1100 respectively. The ratio between the number of bad cells and the total number of cells is of the order of 3% after  $t = 0.025$ , and about 10% before. In other words, on average, only 4% – 5% of the cells demand to be recomputed through the MOOD loop, leading to an efficient way to limit and stabilize numerical schemes.

As expected, the final accuracy is linked to the number of refinement levels  $\ell$  for fixed initial uniform mesh (driven by  $N^0$ ) before refinement. In summary the  $h = 1/N^0$  parameter drives the coarsest cell length, while  $h = 1/(N^0 2^\ell)$  is the finest one. The threshold parameter value controls the spreading of the refined zones.

In table 10 we gather the CPU time needed to perform the simulations. Note that, for these cases, the

$(N^0, \ell)$	$N^{eq}$ unif.	$N_c(t), \bar{N}_c, N_c(t=0)$	Ratio	CPU	CPU unif.	Ratio
(30,4)	$408 \times 240 \approx 98000$	33414, 20000, 7080	2.9, 4.9, 13.8	6.5h	$\approx 29.4h$ (1d)	4.5
(30,5)	$816 \times 480 \approx 392000$	124776, 80000, 11859	3.1, 4.9, 33.1	56h(2.5d)	$\approx 290h$ (12d)	5.2

Table 10: Shock-bubble interaction problem — Equivalent uniform runs; final, average and initial number of AMR cells (and ratio); CPU times of the AMR run, expected CPU time of the equivalent uniform run (extrapolated) and ratio (h: hour, d: day, w:week, m:month).

equivalent meshes would be made of  $(1.7 \times N^0 \times 2^{\ell-1}) \times (N^0 \times 2^{\ell-1})$  cells, which is 97920 cells for  $\ell = 4$  and 391680 cells for  $\ell = 5$  and  $N^0 = 30$ . In order to compare the execution times of AMR and uniform runs, we measured the execution times of the (30, 1), (60, 1) and (120, 1) runs and extrapolated the data to the  $N^{eq}$  grids. The ratio is about 4 – 5 for both runs, indicating that AMR simulations run 4 to 5 times faster than the equivalent uniform simulation. The efficiency of the AMR scheme is linked to its ability to perform the computation with 5 times less cells on average. Note also that the number of cells in the AMR simulations is very variable, since the AMR code employs as few as 1/14th (resp. 1/30th) of cells at initial time, when the waves are very localized.

From this test we have observed that the numerical method is robust, can nonetheless maintain a high accuracy, capture small-scale flow structures using the AMR technology. The *a posteriori* stabilization technology which allows for an almost optimal use of the 3rd order accurate scheme away from steep gradients works fine within the AMR framework as the number of bad cells remains of the order of few percents, and can preserve non-oscillatory numerical solutions.

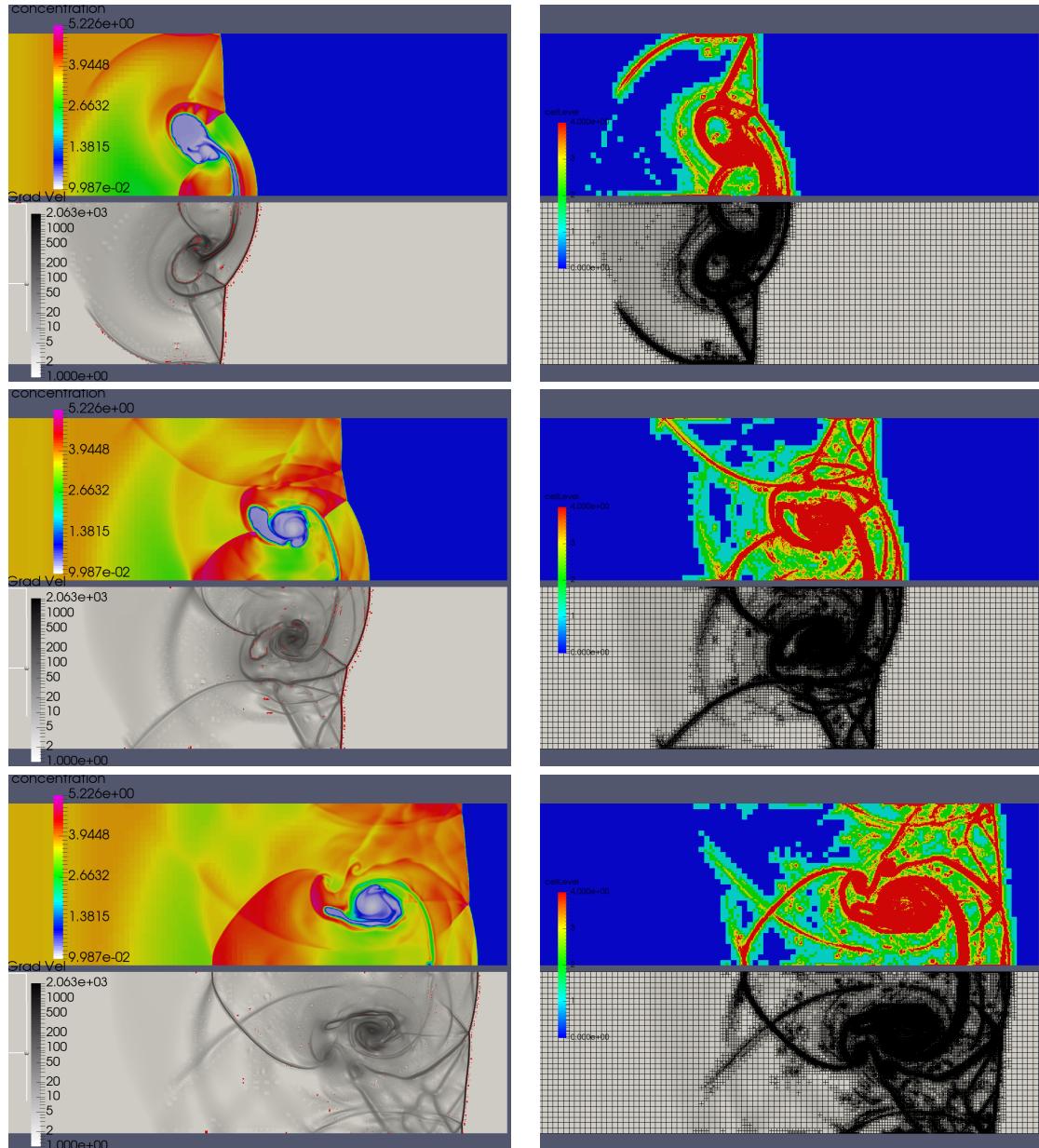


Figure 27: Shock-bubble interaction problem — Mesh with  $\ell = 5$  refinement levels — Top-bottom: time  $t = 0.2, 0.3$  and final  $0.4$  — Left panels: Density (top with colors), norm of velocity gradients (bottom with gray) and bad cells detected by MOOD (bottom in red) — Right panels: Cell levels (top with colors), adapted mesh (bottom)

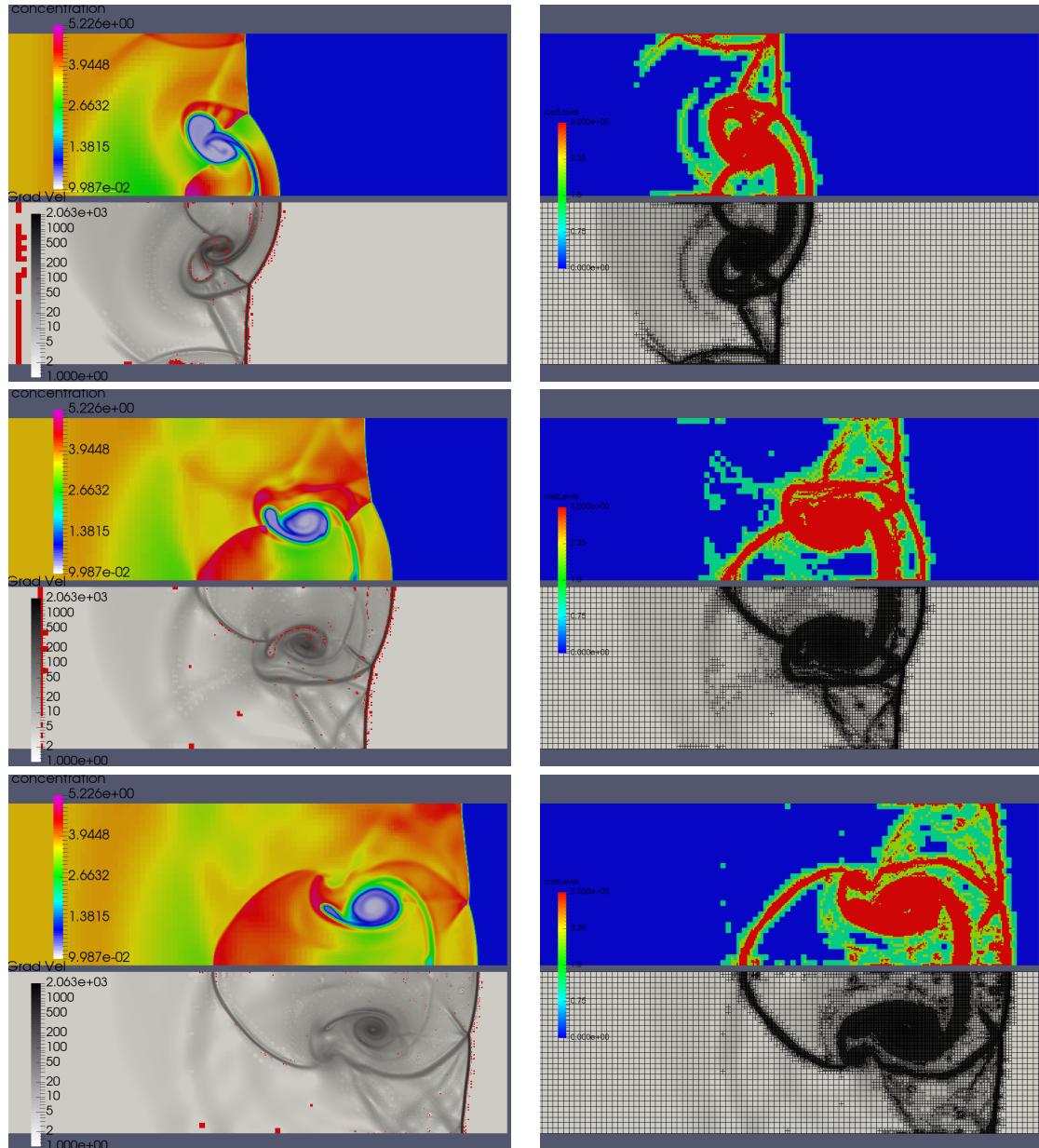


Figure 28: Shock-bubble interaction problem — Mesh with  $\ell = 4$  refinement levels — Top-bottom: time  $t = 0.2, 0.3$  and final  $0.4$  — Left panels: Density (top with colors), norm of velocity gradients (bottom with gray) and bad cells detected by MOOD (bottom in red) — Right panels: Cell levels (top with colors), adapted mesh (bottom)

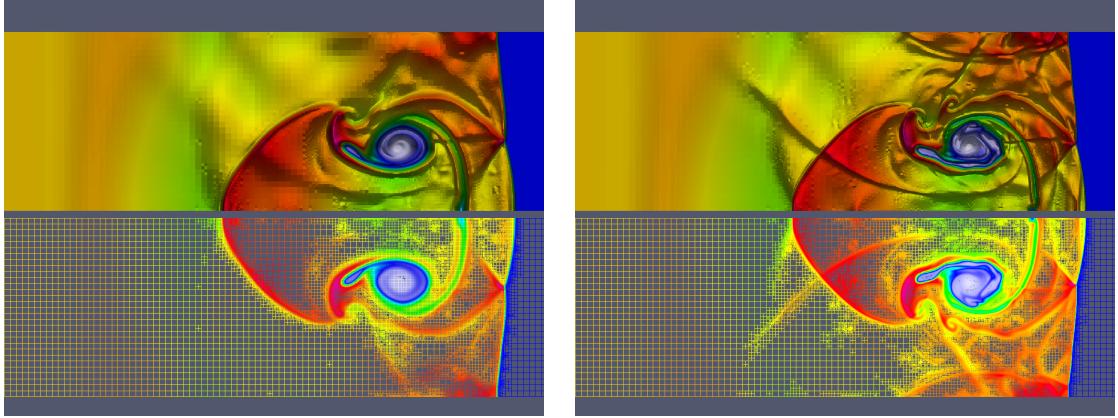


Figure 29: Shock-bubble interaction problem at final time — Comparison  $\ell = 4$  (left) and  $\ell = 5$  (right). 3D elevation colored density seen from above (top part of the panels) and the mesh colored by the density (bottom part).

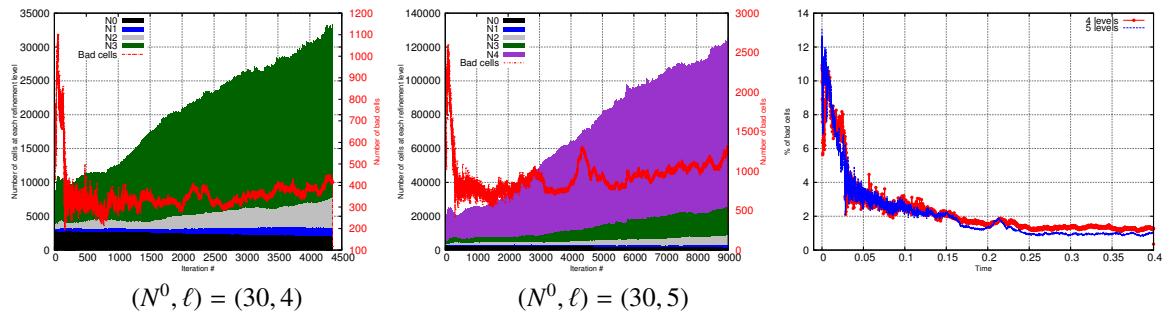


Figure 30: Shock-bubble interaction problem — Results for a mesh with 4 (left) or 5 (middle) levels of refinements — Left-middle: Colored histograms for the number of cells of each level ( $N_k$ ,  $k = 0, \dots, 2$  or 4) and number of bad cells (red symbols) as a function of time iteration — Right: percent of bad cells detected by MOOD loop.

## 6. Conclusion and Perspectives

This paper has presented a third-order finite volume (FV) numerical scheme dedicated to solve the Euler system of conservation laws in two dimensions under an Adaptive Mesh Refinement (AMR) framework for unstructured quadrangular grids. The numerical method is composed of a Runge-Kutta discretization in time coupled with a high resolution finite volume scheme using polynomial reconstructions of conservative variables. The AMR technology employs the numerical entropy production criterion to determine when to refine or coarsen the mesh. Previous to this work, this scheme was implemented within a 2D AMR framework which employed CWENO type of reconstruction procedure.

Contrarily, in this work the novel *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) approach is applied to damp the spurious numerical oscillations that may occur. MOOD is an *a posteriori* solution to the problem of limiting spurious numerical oscillations that inexorably occur with high resolution schemes. In our case high accuracy is obtained through Runge-Kutta discretization in time and high order accurate polynomial reconstructions in space. Each candidate solution at  $t^{n+1}$  is computed using third order accurate polynomial reconstructions without any limiting. Next, this candidate solution at  $t^{n+1}$  is checked against user-specified detection criteria which determine whether a numerical cell centered datum is admissible or not. If a cell (from the candidate solution) is detected as problematic, it is re-computed starting again at  $t^n$ , decrementing the polynomial degree of the reconstructions in this cell and, possibly in the neighbors. The reconstructions are either  $\mathbb{P}_2$  or  $\mathbb{P}_0$ . Note that other reconstructions in-between  $\mathbb{P}_2$  and  $\mathbb{P}_0$  could be tried, but in this work, we focused on the simplest possible choice.

This iterative procedure (MOOD loop) stops either if all cells have an admissible discrete numerical solution, or, if the polynomial degree has reached zero. Zero corresponds to updating the cell with a first-order accurate Godunov scheme, which is considered as the more robust scheme in our context, the solution of which is always considered as acceptable.

Appropriate detection criteria are derived from the underlying physics (positivity of density and pressure), and, from desired numerical properties (Discrete Maximum Principle relaxed by the so-called u2 criteria to allow the development of smooth extrema, check of invalid computer datum (NaN, Inf), etc.). The essentially non oscillatory behavior of the solution is observed when these numerical admissible detection criteria are employed for all the tests presented in this work.

A large list of test cases have been simulated. We have systematically observed that the AMR technology couples well with this *a posteriori* limiting. In fact the optimal accuracy is reached when the solution is smooth, and an essentially non-oscillatory behavior is observed in presence of shock waves and steep gradients. The refinement and coarsening occur where and when expected in each test, and few falsely refined regions are generated. Obviously the CPU time of AMR simulations is genuinely smaller than the cost of equivalent uniform mesh simulations, the acceleration is between 4 to 20 for an equivalent accuracy. This allows us to simulate refined mesh simulations even on serial computer in a rather acceptable duration. Accurate results are observed on these complex flows, and, as reported in this paper, only few percents of troubled cells demand to be recomputed. As a consequence the extra-cost brought by the MOOD loop is, if not negligible, truly minor.

In the future we plan to investigate the massively parallel version (under MPI) of this numerical method and its extension to 3D. In theory there is no limitation in the MOOD approach to deal with higher-order polynomial reconstructions, larger than 3rd order accurate. Therefore we plan to extend our scheme to higher resolution capabilities by improving the Runge-Kutta time discretization and allowing a larger stencil to reconstruct polynomials with higher degrees.

At last the AMR technology based on numerical entropy production threshold has been designed as an *a posteriori* technology in [PS11, SCR15], therefore, its association with an *a posteriori* limiting procedure seems fairly natural and proved effective.

## Acknowledgments

The material of this research has been partly built during the blue SHARK FV workshops which took place on May 2015, 2016 in Ofir, Portugal <http://www.math.univ-toulouse.fr/SHARK-FV/>. The authors would like to thank M. Dumbser, S. Clain for fruitful discussions, and W. Boscheri for providing the reference solution of the radial shock tube. This work was supported by “National Group for Scientific Computation (GNCS-INDAM)”. We also thank the partial support of the exchange program Partenariat Hubert Curien (PHC) Franco-Italian “Galileo” (project G14-19 #32272UL) under the supervision of Giacomo Dimarco (University of Ferrara, Italy), and Jacek Narksi (University of Toulouse, France).

## References

- [ADKN16] Martin Alkmper, Andreas Dedner, Robert Klöfkorn, and Martin Nolte, *The DUNE-ALUGrid module.*, Arch. Numer. Software **4** (2016), no. 1, 1–28.
- [BBD<sup>+</sup>08] P. Bastian, M. Blatt, A. Dedner, C. Engwer, R. Klöfkorn, M. Ohlberger, and O. Sander, *A Generic Grid Interface for Parallel and Adaptive Scientific Computing. Part I: Abstract Framework*, Computing **82** (2008), no. 2–3, 103–119.
- [BC89] M.J. Berger and P. Colella, *Local adaptive mesh refinement for shock hydrodynamics*, Journal of Computational Physics **82** (1989), no. 1, 64 – 84.
- [BF90] T.J. Barth and P.O. Frederickson, *Higher order solution of the euler equations on unstructured grids using quadratic reconstruction.*, 28th Aerospace Sciences Meeting (January 1990), AIAA paper no. 90-0013.
- [BLD15] Walter Boscheri, Raphaël Loubère, and Michael Dumbser, *Direct Arbitrary-Lagrangian-Eulerian ADER-MOOD finite volume schemes for multidimensional hyperbolic conservation laws*, Journal of Computational Physics **292** (2015), 56 – 87.
- [BO84] M. J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations.*, J. Comput. Phys. **53** (1984), 484–512.
- [CDL11a] S. Clain, S. Diot, and R. Loubère, *A high-order finite volume method for systems of conservation laws multi-dimensional optimal order detection (MOOD)*, Journal of Computational Physics **230** (2011), no. 10, 4028 – 4050.
- [CDL11b] 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*, FVCA 6, International Symposium, Prague, June 6–10 (Fort Fürst Halama Herbin Hubert (Eds.), ed.), Series: Springer Proceedings in Mathematics, vol. 4, 2011, 1st Edition. XVII, 1065 p. 106 illus. in color.
- [CM14] 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 **68** (2014), no. 10, 1292–1311.
- [CMNP13] 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** (2013), 43–64.
- [CS15] I. Cravero and M. Semplice, *On the accuracy of weno and cweno reconstructions of third order on nonuniform meshes*, Journal of Scientific Computing (2015), 1–28.
- [CT09] M. Cada and M. Torrilhon, *Compact third-order limiter functions for finite volume methods*, J. Comput. Phys. **228** (2009), no. 11, 4118–4145.
- [DCL12] 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** (2012), 43 – 63.
- [DK07] 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** (2007), 693–723.
- [DKTT07a] M. Dumbser, M. Käser, 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** (2007), 204–243.
- [DKTT07b] M. Dumbser, M. Käser, 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** (2007), 204 – 243.
- [DLC13] 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** (2013), 362–392.
- [GS98] S. Gottlieb and C.W. Shu, *Total variation diminishing Runge-Kutta schemes*, Mathematics of Computation **67** (1998), 73–85.
- [HAC74] C.W. Hirt, A.A. Amsden, and J.L. Cook, *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*, J. Comput. Phys. **14** (1974), no. 3, 227–253, doi:10.1016/0021-9991(74)90051-5.
- [HS99] C. Hu and C.W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes*, Journal of Computational Physics **150** (1999), 97–127.
- [HST10] Thomas J. R. Hughes, Guglielmo Scovazzi, and Tayfun E. Tezduyar, *Stabilized methods for compressible flows*, Journal of Scientific Computing **43** (2010), no. 3, 343–368.
- [JS96] G.-S. Jiang and C.W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics **126** (1996), 202–228.

- [KI05] M. Käser and A. Iske, *ADER schemes on adaptive triangular meshes for scalar conservation laws*, Journal of Computational Physics **205** (2005), 486–508.
- [Kol10] V. P. Kolgan, *Application of the principle of minimizing the derivative to the construction of finite-difference schemes for computing discontinuous solution of gas dynamics*, J. Comput. Phys. (2010), doi: 10.1016/j.jcp.2010.12.033.
- [LDD14] 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** (2014), 718–763.
- [LMS<sup>+</sup>10] Raphaël Loubère, Pierre-Henri Maire, Mikhail Yu. Shashkov, Jérôme Breil, and Stéphane Galéra, *Reale: A reconnection-based arbitrary-lagrangian-eulerian method*, J. Comput. Physics **229** (2010), no. 12, 4724–4761.
- [Lou13] R. Loubère, *Contribution to lagrangian and arbitrary-lagrangian-eulerian numerical schemes*, Ph.D. thesis, University of Toulouse, France, 2013, Habilitation à diriger des recherches.
- [LW60] P.D. Lax and B. Wendroff, *Systems of conservation laws*, Communications in Pure and Applied Mathematics **13** (1960), 217–237.
- [LW64] P. Lax and B. Wendroff, *Difference schemes for hyperbolic equations with high order of accuracy*, Comm. Pure Appl. Math. **XVII** (1964), 381–398.
- [Mai09] P.-H. Maire, *A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured mesh*, J. Comput. Phys. **228** (2009), no. 7, 2391–2425.
- [MB05] S. M. Murman M. Berger, M. J. Aftosmis, *Analysis of slope limiters on irregular grids*, Tech. Report NAS-05-007, NASA Technical Report, 2005.
- [NR50] J. Von Neumann and R. D. Richtmyer, *A method for the numerical calculation of hydrodynamic shocks*, Journal of Applied Physics **21** (1950), 232–237.
- [NRCL16] Xesús Nogueira, Luis Ramírez, Stéphane Clain, and Raphaël Loubère, *High accurate sph method with multidimensional optimal order detection*, Computer Methods in Applied Mechanics and Engineering **310** (2016), 134–155.
- [OGA02] 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** (2002), 729 – 752.
- [Pin11] Stéphane Del Pino, *Metric-based mesh adaptation for 2D Lagrangian compressible flows*, J. Comput. Phys. **230** (2011), no. 5, 1793–1821.
- [PS11] G. Puppo and M. Semplice, *Numerical entropy and adaptivity for finite volume schemes*, Commun. Comput. Phys. **10** (2011), no. 5, 1132–1160.
- [Pup04] G. Puppo, *Numerical entropy production for central schemes*, SIAM J. Sci. Comput. **25** (2003/04), no. 4, 1382–1415 (electronic).
- [SC14] M. Semplice and A. Coco, *dune-fv software*, 2014, <http://www.dipmatematica.unito.it/do/docenti.pl/Alias?matteo.semplice#ricerca>.
- [SCR15] M. Semplice, A. Coco, and G. Russo, *Adaptive mesh refinement for hyperbolic systems based on third-order compact weno reconstruction*, Journal of Scientific Computing **66** (2015), no. 2, 692–724.
- [Shu97] C.W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic Conservation Laws*, NASA/CR-97-206253 ICASE Report No.97-65 (1997).
- [S.K59] S.K.Godunov, *A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics*, Mat. Sbornik **47** (1959), 357–393.
- [Tor99] E.F. Toro, *Riemann solvers and numerical methods for fluid dynamics*, second ed., Springer, 1999.
- [TT03] H.-Z. Tang and T. Tang, *Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws*, SIAM J. Numer. Anal. **41** (2003), 487–515.
- [van74] B. van Leer, *Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme*, Journal of Computational Physics **14** (1974), 361–370.
- [van79] ———, *Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method*, Journal of Computational Physics **32** (1979), 101–136.
- [VC13] V.Desveaux and C.Berthon, *An entropic mood scheme for the euler equations*, International Journal of Finite Volumes (2013).
- [V.D13] V.Desveaux, *Contribution à l'approximation numérique des systèmes hyperboliques*, Ph.D. thesis, Université de Nantes, 2013.
- [Ven95] V. Venkatakrishnan, *Convergence to steady state solutions of the Euler equations on unstructured grids with limiters*, J. Comput. Phys. **118** (1995), no. 1, 120 – 130.
- [WC84] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics **54** (1984), 115–173.
- [ZDLS14] 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** (2014), 47–75.
- [ZFDH15a] 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*, Computer and Fluids **118** (2015), 204–224.
- [ZFDH15b] ———, *Space-time adaptive ADER discontinuous galerkin finite element schemes with a posteriori sub-cell finite volume limiting*, Computers and Fluids **118** (2015), 204 – 224.