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

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

Matteo Semplice[a], Raphaël Loubère*[b]

[a]*Dipartimento di Matematica G. Peano Università di Torino Via C. Alberto, 10 10123 Torino, Italy*
[b]*Institut de Mathématiques de Bordeaux (IMB), UMR 5219, Université de Bordeaux, France*

Abstract

In this paper we propose a third order accurate finite volume scheme based on *a posteriori* limiting of polynomial reconstructions 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 a more dissipative scheme but only locally, close to these cells. By locally decrementing the degree of the polynomial reconstructions from 2 to 0 we switch from a third-order to a first-order accurate but more stable scheme. 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 with the refined mesh, but but only locally, close to the new cells. We show on a large set of numerical tests that this *a posteriori* limiting procedure coupled with the entropy-based AMR technology can maintain not only 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 at least comparable in terms of accuracy and cost to a more classical CWENO approach within the same AMR context.

Keywords: 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 and most obvious one, is the development of robust and accurate conservative numerical methods based on the seminal works of Lax and Wendroff [16, 15], Godunov [31] and many others. 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 ensures a nominal third-order accuracy, while employing a small stencil of neighbors. What remains to design is a valid limiting strategy to ensure robustness and an essentially non-oscillatory behavior. In fact, any linear

*Corresponding author

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

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 [22], (slope) limiters [34, 35, 36, 19, 14] or stabilization techniques [12]. All have in common the desire to detect *a priori* within a solution where spurious numerical oscillations are or may appear and how much numerical dissipation is sufficient to avoid their growth. All of the above-mentioned techniques, more or less, answer these two questions, sometimes independently. In this work, instead we rely on the *a posteriori* MOOD paradigm which was developed initially in [5] and further extended to different contexts, for instance in [17, 38]. This *a posteriori* procedure checks at the end of the timestep for troubled cells, and further recomputes them with a more dissipative scheme. This procedure ensures that the physical admissibility of the solution is maintained, preserves an essentially-non-oscillatory behavior and even cures pathological behaviors like unrepresentable numbers such as NaN's and Inf's.

The efficient use of the degrees of freedom on a given mesh by the numerical method is not the only factor influencing the efficiency of the numerical simulation. Another important factor is the efficient use of the available computer resources, which can be achieved via parallel computing (CPU/GPU hybrid codes, OpenMP shared memory models, MPI massively parallel distributed memory) or by the 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) [11, 20]) or manually (Moving Mesh techniques [32]). Some techniques allow local changes of connectivity [18, 25], or, simply, emancipate from the strict notion of mesh, like in the case of particle methods such as the Smoothed Particle Hydrodynamics (SPH) method [23]. Contrarily, the AMR strategy adds new cells where appropriate and removes old ones which are no longer required. A drastic reduction of the costs without sacrificing the level of accuracy is the main reason why those techniques were developed [2, 3] and are still in use today [29].

In this work we rely on AMR technology using the so-called numerical entropy production refinement criterion [26, 27, 29], 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, the solution is sent back at t^n for local re-computation. Consequently a solution is always computed with an appropriately refined mesh 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;

Efficient: code architecture, parallelisation environment and numerical scheme should blend to allow the efficient use of available computer resources.

When dealing with non-linear systems of PDEs, guaranteeing the robustness of high order accurate numerical schemes is increasingly difficult. Our goal in this work is to show that, at least for 2D hydrodynamics equation, the coupling of *a posteriori* limiting strategy with *a posteriori* AMR technology leads to improvements in terms of resolution capability and effectiveness. Moreover we will show that this approach leads, on the one hand, to a fail-safe numerical method for extreme situations (when violation of positivity or unrepresentable numbers may occur) and, on the other hand, to a less dissipative scheme scheme for complex flows generating small-scale structures. 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 recomputing the same timestep if and where troubles have been detected.
~~Classical *a priori* techniques rely on data at discrete time t^n and on an appropriate behavior of the numerical scheme to prevent the occurence of spurious numerical oscillations. When dealing with non-linear systems of PDEs, guaranteeing the robustness of high order accurate numerical schemes is increasingly difficult. In this work we thus explore a different approach studying the coupling of *a posteriori* techniques with AMR. In this approach, once problematic situations are detected at t^{n+1}, the numerical method must adapt (locally), and, provide a better suited numerical solution starting again from the data at time t^n. In particular, both the AMR procedure based on entropy production diagnostics and the MOOD limiting procedure for the hydrodynamics solver rely on such an *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 driven by the numerical entropy production. 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. This overall numerical method with MOOD limiting is further validated and tested in section 5. There, we gather the numerical results for a large set of test cases involving complex smooth and non-smooth flows. 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 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 in two space dimensions. It reads

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

where ρ denotes the mass density, $\mathbf{v}=(u,v)$ the velocity vector, p the fluid pressure, E the total energy density and $\mathbf{I}$ the 2×2 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 γ:

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

where $\varepsilon=E-\frac{1}{2}\mathbf{v}^2$ is the specific internal energy. The sound speed is defined as $a=\sqrt{\gamma p/\rho}=\sqrt{\gamma(\gamma-1)\varepsilon}$. Physically admissible states are such that $\rho>0$ and $p>0$, or, equivalently $\rho>0$ and $\varepsilon>0$. In the state space defined by these inequalities the system is hyperbolic.
A general formulation of this nonlinear system of hyperbolic conservation laws is given by

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

with appropriate initial and boundary conditions. $\mathbf{x}=(x,y)$ is the coordinate vector, $\mathbf{U}=(\rho,\rho u,\rho v,\rho E)$ is the vector of 4 conserved variables, and $\mathbf{F}=(\mathbf{f},\mathbf{g})$ is the conservative nonlinear flux tensor depending on $\mathbf{U}$.

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. The computational domain $\Omega \subset \mathbb{R}^2$ is discretized by a set of N_E not necessarily conforming quadrangular elements, see figure 1. The union of all elements Ω_i 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 $\Theta^{n+1/2} = [t^n, t^{n+1}]$, the initial and final times being $t^0 = t^{\text{init}}$ and $t^N = t^{\text{final}}$. The (not necessarily constant) time step is $\Delta t = t^{n+1} - t^n$. We first introduce the cell averages of the solution as

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

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):

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

where $\mathbf{n}$ denotes the outgoing unit normal to the boundary of Ω_i. In the next sub-sections describe the main aspects of the numerical approximation of (5) with third order accuracy and our AMR technique.

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 Ω_i is

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

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

$$\Psi_{i,1} = x - x_i, \qquad \Psi_{i,2} = y - y_i, \tag{7a}$$

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

where $\mathbf{x}_i = (x_i, y_i)$ is the centroid of the cell Ω_i the size of which is $\Delta x_i \times \Delta y_i$, with $\Delta x_i = \Delta y_i$ in case of a square cell. Following [29], all basis functions have null cell average.

The reconstruction on element Ω_i is based on the cell averages in a so-called reconstruction stencil $\mathcal{S}_i$, that is an appropriate set of cells including Ω_i and a number $n_e(i)$ of its neighbors:

$$\mathcal{S}_i = \bigcup_{k=1}^{n_e(i)} \Omega_{j_i(k)}. \tag{8}$$

Here k is a local index listing the elements in the stencil, and $j_i(k)$ represents the mapping from the index k in the local stencil $\mathcal{S}_i$ to the global index in the 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 [24, 13]. In our case we include in $\mathcal{S}_i$ all the cells intersecting Ω_i at least in a point (see [29] 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 $\mathbf{w}_i$ should then satisfy

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

The above represents an over-determined system of linear equations in the coefficients of $\mathbf{w}_i$. This system is thus solved in a least square sense, but, for conservation, we insist that the equation for $k = 1$ be satisfied exactly. Thanks to our choice of basis functions with zero average, 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 [8, 9] for the same level of accuracy. $\mathcal{S}_i$ is the same as the stencil of the CWENO3 reconstruction of [29]. Here, however, by employing a single central unlimited polynomial, we save the extra cost of computing the lower degree polynomials 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

In order to compute the right hand side of (5) we decompose $\partial\Omega_i$ into the edges e_{ij} separating Ω_i from a neighboring cell Ω_j and choose a quadrature formula along each edge e_{ij} so that

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

where ω_g and ξ_g are respectively the weights and the quadrature nodes, while $|e_{ij}|$ is the length of edge e_{ij}. In this work, we employ the 2-point Gaussian formula: in a reference interval $[-\frac{1}{2}, \frac{1}{2}]$, we consider the formula with nodes at $\pm\sqrt{3}/6$ and both weights equal to $1/2$. On the two sides of any edge two reconstructions are defined ($\mathbf{w}_i(s)$ and $\mathbf{w}_j(s)$ for $s \in e_{ij}$), and they need not coincide. A numerical flux $\mathcal{F}$ is then employed to choose uniquely the flux at the point s, to finally get

$$\int_{\partial\Omega_i} \mathbf{F}(\mathbf{U}) \cdot \boldsymbol{n}(s)\, ds \approx \sum_{e_{ij}} |e_{ij}| \sum_{g=1}^{G} \omega_g \mathcal{F}(\mathbf{w}_i(\xi_g), \mathbf{w}_j(\xi_g), \boldsymbol{n}_{ij}(\xi_g)) \equiv |\Omega_i|\, \mathbf{L}_i(\mathbf{u}). \tag{11}$$

In this work we employ the simple Local Lax-Friedrichs (LLF) flux, i.e. $\mathcal{F}(\boldsymbol{a}, \boldsymbol{b}) = (\mathbf{F}(\boldsymbol{a}) + \mathbf{F}(\boldsymbol{b}) - \alpha(\boldsymbol{b} - \boldsymbol{a}))/2$, where α is the largest of the spectral radii of $\mathbf{F}'(\mathbf{a})$ and $\mathbf{F}'(\mathbf{b})$. Of course other exact or approximate Riemann solvers could be employed for this purpose [33], like the classical HLLC solver.

3.3. Time discretization

In this work we integrate the semidiscrete equation (5) with the classical SSP Runge-Kutta scheme or order 3 introduced in [10]. More in details our fully discrete scheme reads

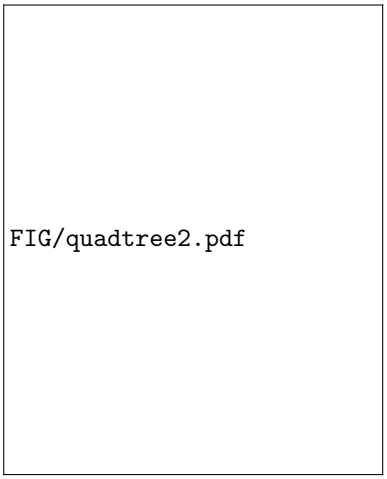
$$\mathbf{u}_i^{(1)} = \mathbf{u}_i^n - \Delta t\, \mathbf{L}_i(\mathbf{u}^n), \qquad \mathbf{u}_i^{(2)} = \frac{3}{4}\mathbf{u}_i^n + \frac{1}{4}\mathbf{u}_i^{(1)} - \frac{1}{4}\Delta t\, \mathbf{L}_i(\mathbf{u}^{(1)}), \tag{12a}$$

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

where $\Delta t = t^{n+1} - t^n$ and the spatial operator is defined in (11). The method (12) is by construction third order accurate in time for smooth solutions. Its stability is guaranteed under a classical CFL condition

$$\Delta t \leq 0.5 \min_i h_i/\lambda \tag{13}$$

where h_i is the characteristic length of the cell Ω_i and λ is the largest spectral radius of the flux Jacobians on the flow at time t^n . The value of λ is computed easily from the α's employed in the LLF flux, since they record the maximal eigenvalues of the flux function.



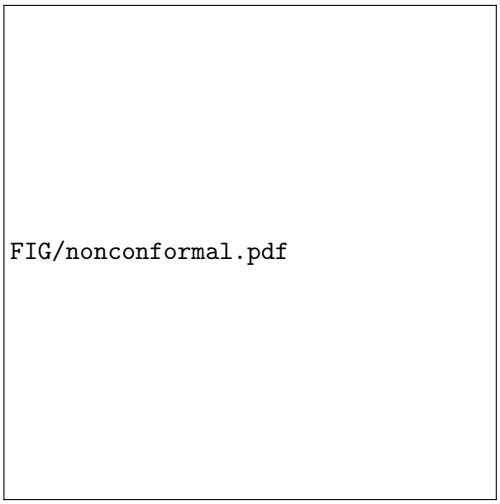


Figure 1: Left: Example of subdivision of an original cell with $\ell^{\max} = 3$ levels of refinement: quad tree (left) and associate grid (right) — Right: Example of subdivision of an original cell and associated non-conformal mesh. Quadrature points are represented by red bullets. The blue cell has six neighboring cells The intersection between the blue cell and one neighboring cell is called an edge or intersection. Each intersection demands at minima two quadrature points to reach third order accurate integration. 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.

3.4. Adaptive Mesh Refinement (AMR)

The adaptive grid is recursively generated starting from a coarse uniform, two-dimensional, Cartesian mesh of grid size H at level $\ell = 0$. Each cell may be recursively subdivided into four equal squares. In order to allow the coarsening of previously refined cells, the grid structure is stored in a quad-tree, where each grid cell correponds to a node in the tree.
One example of subdivision of an original cell with $\ell^{\max}$ 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 node of the quad-tree. Each cell of level $\ell \in \{1, \dots, \ell^{\max}\}$ has a father cell, which corresponds to a node in the quad-tree at level $\ell - 1$. In the tree, the four nodes connected to the father node are called the children of the node.

Such subdivisions may lead to non-conformal cells as can be observed in 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 neighboring cells. Each intersection demands 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 coarsen the mesh. In the present paper we employ the numerical entropy production, that was introduced in [26] and further extended in [27]. Its definition considers an admissible entropy pair (η, ψ) and forms the quantity

$$S_i^{n+1/2} = \frac{1}{\Delta t} \left| \overline{\eta}_i^{n+1} - \overline{\eta}_i^{n} + \frac{\Delta t}{|\Omega_i|} \sum_{e_{ij}} |e_{ij}| \sum_{g=1}^{G} \omega_g \Psi(\xi_g) \right|, \tag{14}$$

where $\Psi(\xi_g)$ is the numerical entropy flux function evaluated at integration point ξ_g using the reconstructions on both sides of intersection e_{ij} and $\overline{\eta}_i^n$, $\overline{\eta}_i^{n+1}$ denote the cell averages of the entropy on the numerical

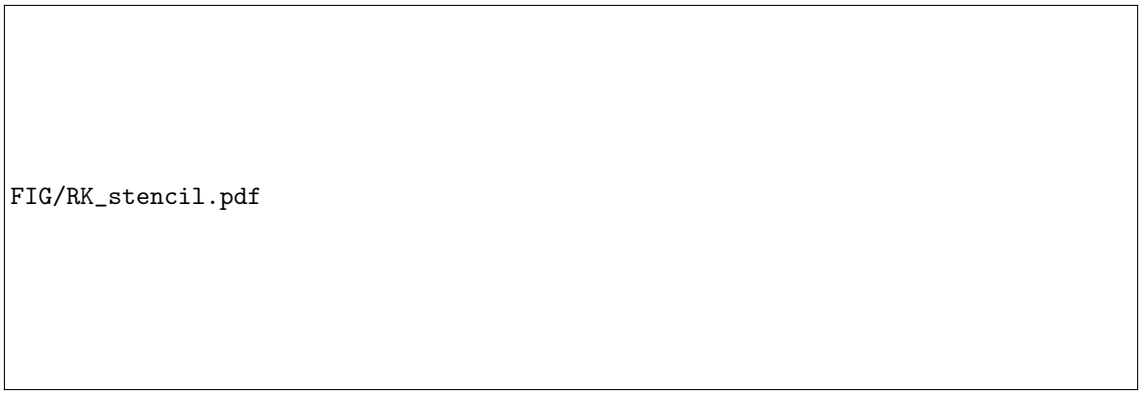


Figure 2: Re-computation after refinement — The top row depicts the tentative computation of the time advancement after which the red shaded cell is marked for refinement. Bullets (black or green) represent the cell averages and the arrows the numerical fluxes computed during the three Runge-Kutta stages — 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.

solution. Note that the computation of S_i^n is non-intrusive in a numerical scheme, since it amounts to the addition of an extra variable in the vector $\mathbf{u}$, whose numerical flux Ψ can be computed using the same intermediate results (notably the polynomial reconstructions) that are needed to be fed into $\mathcal{F}$.

Following [27], for any consistent numerical entropy flux, on a locally smooth solution $S_i^{n+1/2} = O(h^r)$ where r is the order of the scheme and h a characteristic length of the cell, thus scaling at the same rate of the truncation error. If a contact discontinuity (resp. shock) is located in Ω_i, then $S_i^{n+1/2} = O(1)$ (resp. $O(1/h)$). In particular, in this paper we employ the classical numerical entropy pair compatible with (1), defining entropy as $\eta(\mathbf{U}) = -\rho \log(p/(\gamma-1)\rho^\gamma)$ and entropy flux as $\psi(\mathbf{U}) = \eta(\mathbf{U})\mathbf{v}$. Furthermore, the numerical entropy flux is $\mathbf{\Psi}(\mathbf{a},\mathbf{b}) = [\psi(\mathbf{a}) + \psi(\mathbf{b}) - \alpha(\eta(\mathbf{b}) - \eta(\mathbf{a}))]/2$, which is reported in [27, 29] to yield the best results in association to the LLF numerical flux employed in the spatial operator (11).

At the end of each time-step, the quantity (14) is computed in every the cell i, and compared to a threshold S^{ref}. If $S_i^n > S^{\text{ref}}$ and the maximal refinement level $\ell^{\max}$ is not yet reached then the cell is refined. 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 Ω_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, in which one sets $\mathbf{U}^{n+1}$ to the average of the cell averages from the children cells. Following [27, 29], we employ $S^{\text{coa}} = S^{\text{ref}}/4$.

The AMR procedure consists in the following algorithm where C will denote the set of cells to be updated

0. 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 $\mathbf{u}_i^{n=0}$.
   Set C to the entire grid;
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
   Set $C = \emptyset$
   Do for all cell i

(a) If $S_i^{n+1/2} > S^{\texttt{ref}}$ and $\ell_i < \ell^{\max}$, re-computation is needed;
(b) Then split the cell i into children cells, associate new children data averaging the polynomial reconstruction $\mathbf{w}_i(\mathbf{u}^n)$;
(c) Insert the children cells into C;

3. Decision on re-computation
If $C \neq \emptyset$ then go back to 1; Otherwise accept $\mathbf{u}^{n+1}$ and continue;

4. Decision on coarsening
Do for all sets of sister cells (emanating from a common ancestor)
(a) If $S_i^{n+1/2} < S^{\texttt{coa}}$ 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 $\mathbf{u}^{n+1}$;
Set C to the entire grid and go back to 1 for the next timestep.

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

The scheme described so far has no embedded limiting procedure to dissipate the 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.

4. *a posteriori* limited Finite Volume scheme

In previous works, CWENO polynomial reconstructions have been used to limit and stabilize this scheme [27, 29, 6]. In this work we will use an *a posteriori* stabilization procedure based on a troubled cell detector and subsequent recomputation with a first order finite volume scheme. This procedure is based on the so called MOOD paradigm, see [5, 17] .

4.1. MOOD paradigm as an a posteriori stabilization technique

An *a posteriori* MOOD limiting strategy is 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.

(i) The *Detection criteria*. This is a list of properties based on physics or numerics which are checked to assess if a numerical solution in a cell is acceptable at the end of a time-step. If the detection criteria are not fulfilled, then, locally, some action must be taken to supplement the scheme with more dissipation.
(ii) 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 [5, 17], also in this work the first-order local Lax-Friedrichs finite volume scheme is used.
(iii) The *cascade*. The cascade is a list of ordered numerical schemes from the most accurate but prone to instability, down to the least accurate but most robust [17].

The MOOD loop embraces the main evolution routines of the high order numerical scheme. The solution is first updated, then the detection criteria are checked and bad cells are recomputed by lowering the polynomial degree of reconstructions from 2 to 0. Once a valid solution $\boldsymbol{U}^*$ at t^{n+1} is obtained, it is fed into the AMR procedure, which may decide to locally refine the mesh. If refinement is needed, then the current time-step is recomputed locally on the new mesh; otherwise coarsening and projection are performed to complete the time step. The gain in efficiency brought by using the *a posteriori* MOOD paradigm is due to the fact that usually few cells need a decrementing of the local polynomial degree. Moreover using an

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

unlimited polynomial reduces, in good cells, the numerical dissipation to a minimum. Moreover being the timestepping explicit, the recomputation needs only to be performed locally in the vicinity of the troubled cells. Therefore the extra-work implied by the MOOD loop to recompute a new candidate solution on the few problematic cells is genuinely low. Without any doubt the detection criteria is the most important entity of the *a posteriori* MOOD procedure.

4.2. Physical Admissibility Detection criteria (PAD)

The Physical Admissibility Detection criteria state that the candidate numerical solution $\boldsymbol{U}^*$ must belong to the physical state space defined in section 2 by the following positivity criteria:

$$\rho_i^* > 0, \qquad \text{and} \quad \varepsilon_i^* = E_i^* - \frac{1}{2}|\mathbf{v}_i^*|^2 > 0. \tag{15}$$

They ensure that pressure and sound-speed are unambiguously defined, and system (1) remains hyperbolic.

4.3. Numerical Admissibility Detection criteria (NAD)

The Numerical Admissibility Detection criteria are based on the Discrete Maximum Principle (DMP) with the so-called relaxed u2 criteria [7], a plateau detection and a NaN (Not-A-Number) detection. All these NAD criteria must detect spurious numerical oscillations or lethal situations ignoring smooth extrema or tiny oscillations.

Extrema and smooth extrema detector: DMP+u2 The DMP+u2 detection procedure acts on a generic scalar variable v. For a candidate solution v_i^* at time t^{n+1} in the cell i and a given set $\mathcal{S}_i$ of neighboring cells, we first check if v_i^* fulfills the DMP

$$m_i^n \equiv \min_{j\in\mathcal{S}_i, j\neq i}(v_j^n, v_i^n) \leq v_i^* \leq \max_{j\in\mathcal{S}_i, j\neq i}(v_j^n, v_i^n) \equiv M_i^n. \tag{16}$$

Figure 4: Sketch of the chain detector employed in the MOOD loop. A candidate solution 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.

If the DMP is fulfilled then the cell is admissible and the candidate solution v_i^* is accepted. Otherwise the u2 criterion described in [7] checks if this new extremum belongs to a smooth underlying function. Namely, a candidate solution violating the DMP is nonetheless eligible if

$$\left(\mathcal{X}_i^{max}\mathcal{X}_i^{min} > 0, \quad \text{and} \quad \left|\frac{\mathcal{X}_i^{min}}{\mathcal{X}_i^{max}}\right| \geq 1-\varepsilon\right) \quad \text{and} \quad \left(\mathcal{Y}_i^{max}\mathcal{Y}_i^{min} > 0, \quad \text{and} \quad \left|\frac{\mathcal{Y}_i^{min}}{\mathcal{Y}_i^{max}}\right| \geq 1-\varepsilon\right), \tag{17}$$

where ε is a smoothness parameter set to $1/2$, and

$$\mathcal{X}_i^{min} = \min_{j\in\mathcal{S}_i, j\neq i}\left(\mathcal{X}_i^*, \mathcal{X}_j^*\right), \mathcal{X}_i^{max} = \max_{j\in\mathcal{S}_i, j\neq i}\left(\mathcal{X}_i^*, \mathcal{X}_j^*\right), \text{ and } \mathcal{Y}_i^{min} = \min_{j\in\mathcal{S}_i, j\neq i}\left(\mathcal{Y}_i^*, \mathcal{Y}_j^*\right), \mathcal{Y}_i^{max} = \max_{j\in\mathcal{S}_i, j\neq i}\left(\mathcal{Y}_i^*, \mathcal{Y}_j^*\right). \tag{18}$$

$\mathcal{X}_i^*, \mathcal{Y}_i^*$ represent a measure of local discrete curvature in x, y direction respectively. We use the second derivative in x and y direction of the local third order polynomial reconstruction w_i^n, see section 3.1

$$\mathcal{X}_i^* = \partial_{xx}(w_i^n(\mathbf{x})), \quad \text{and} \quad \mathcal{Y}_i^* = \partial_{yy}(w_i^n(\mathbf{x})). \tag{19}$$

In this work the scalar variable w is chosen as either the density ρ or the entropy η [1].

Plateau detector A second numerical detection criteria consists in considering always valid the cells on a numerical plateau, that is cells such that $(M_i^n - m_i^n) < h_i^3$, where h_i is the (smallest) local characteristic length of the current cell i, if $h_i < 1$.

NaN detector Last a test for undefined or unrepresentable discrete values such as NaN (Not-a-Number) and Inf (Infinite) allows the code to restart at the previous time step at any occurrence of non-numeric floating point values.

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 possibilities, 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. 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 $\mathbb{P}_2 \to \mathbb{P}_0$[2]. Note that when two neighboring cells are assigned a different

[1] Several scalar variables could be tested successively, for instance ρ, u, v, p, or characteristic variables. This remains the choice of the user/developer. We only consider a single detection variable in this work to avoid unecessary complications.

[2] Note that more advance cascades have been employed elsewhere [5, 17], but, here we would like to demonstrate that even without fine tuning, clear improvements are attained.

Figure 5: Illustration of the decrementing procedure. When two neighboring 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.

cell polynomial degree, say $d_i = 2$ and $d_j = 0$ (black and blue cells in the figure), 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. 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.

4.5. Iterative MOOD loop

For each stage of the RK3 scheme an iterative MOOD 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 candidate solution $\mathbf{U}^{(j)}$ in the domain of dependence of $\mathcal{B}^{k-1}$
   (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. In our case if a cell is updated with the first order local Lax-Friedrichs FV scheme, ie with $d_i = 0$, then we set the cell as being valid, ensuring the convergence of the loop. As such, at the end of each RK3 stage, the computed 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 describes a list of representative test cases for the hydrodynamics system (1), with the perfect gas equation of state with $\gamma = 1.4$. The goals of this section can be summarized as follows.

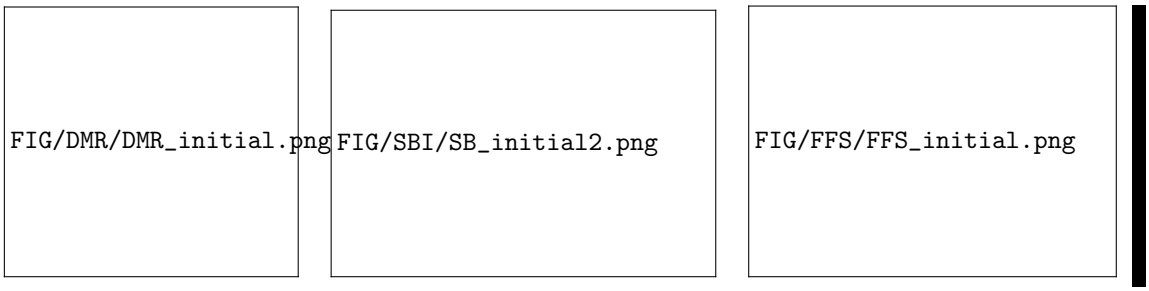


Figure 6: Examples of initial meshes (zooms) obtained with the AMR machinery with $\ell = 5$, showing the adaptation to gradients in the initial conditions (left-middle panels) or to the waves emanating during the very first time-step (right panel).

1. We will numerically validate that the coupling of MOOD and AMR techniques provides an accurate and robust numerical method in 2D:
   - for smooth solutions (isentropic vortex in motion), the effective optimal accuracy is attained, with and without AMR;
   - for non smooth flows (shock tubes, double Mach reflection, forward facing step, shock-bubble interaction) the local order reduction provided by MOOD and the local mesh refinement provided by AMR cooperate nicely, computing non oscillatory and accurate numerical solutions.
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 method of [29]. The blastwave of Colella-Woodward and the shock-bubble interaction problem will present two tests for which the present *a posteriori* MOOD approach outperforms the CWENO scheme.
3. We will show that the *a posteriori* limiting does not constitute an heavy computational overhead, by monitoring the number of troubled cells detected by MOOD.
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, forward facing step, shock-bubble interaction problems).

In order to perform our tests, the numerical scheme described in this paper was coded with the help of the DUNE numerics library [1], employing the ALUGrid quad-tree mesh [21]. The simulations with the AMR-CWENO method were performed with the `dune-fv` module [28]. For an AMR simulation with a coarse mesh of N^0 cells in each direction and employing ℓ refinement levels, we shall denote by $N^{eq} = N^0 \times 2^{\ell-1}$ the number of cells per direction in the equivalent fully refined mesh.

5.1. Initialization

Our preferred strategy to setup the initial mesh is to start with a uniform coarse mesh and recompute the first time step several times, 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. Figure 6 shows how the AMR procedure adapts the mesh to the presence of gradient (first two panels), or, to waves emanating during the very first timestep (third panel).

5.2. Isentropic vortex in motion

The isentropic vortex problem [30] 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

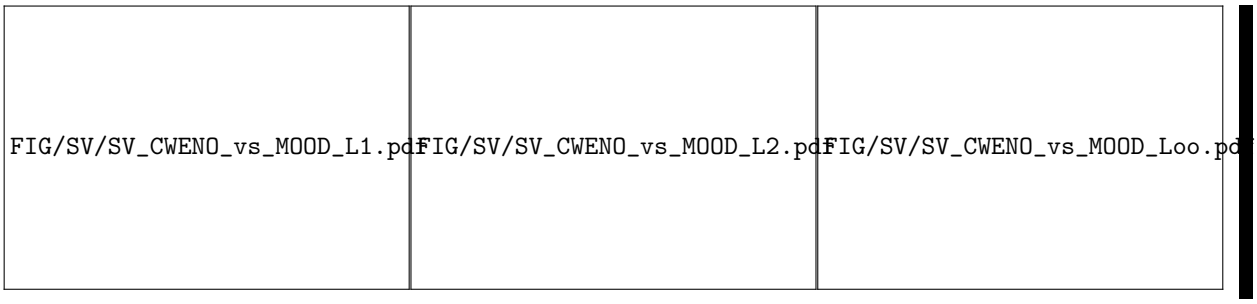


Figure 7: Isentropic vortex problem — L^1, L^2 and L^∞ error curves for CWENO and MOOD simulations as a function of CPU time — Results for uniform (line) and AMR (symbols) grids for all simulations for L^1 (left), L^2 (middle) and L^∞ (right) error norms. Each symbol corresponds to a specific threshold parameter, while each set of three symbols employs a grid made of $N^0 = 32$, 64 or 128 cells with $\ell = 3$ refinement levels.

$T^*_\infty = 1.0$. At the initial time $t = 0$, onto this ambient flow is superimposed a vortex centered at $(0, 0)$ with the following state: $u = u_\infty + \delta u$, $v = v_\infty + \delta v$, $T^* = T^*_\infty + \delta T^*$, where the increments are given by

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

with $r = \sqrt{x^2+y^2}$. The so-called strength of the vortex is set to $\beta = 5.0$ and the initial density is given by $\rho = \rho_\infty (T^*/T^*_\infty)^{\frac{1}{\gamma-1}}$. Periodic boundary conditions are prescribed. At final time $t = t_{\text{final}} = 10$ the vortex is back to its original position and the final solution matches the initial one. Because the solution is smooth, it should be simulated with optimal high accuracy, in other words the limiting/stabilization procedure employed in the scheme should not have any effect.

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

5.2.1. Fixed grids

In Table 1 we report the errors and the associated orders of convergence for three schemes when the AMR capability is shut off. First, the simulation is run with the most accurate, less expensive and physically valid scheme: the unlimited 3rd order accurate scheme (with PAD safety check). Second, we test the 3rd order accurate CWENO scheme and, third, the proposed MOOD schemes, reporting the data for the criteria DMP+u2 on density, since other detection criteria gave analogous results. The timings in the table are the average of the three runs that were made for each scheme configuration (in all cases the standard deviation of the data was below 1% of their average).

	Unlimited (PAD)					CWENO					MOOD ρ $\mathbb{P}_2 \longrightarrow \mathbb{P}_0$				
N_c	L^2 error		L^∞ error		CPU	L^2 error		L^∞ error		CPU	L^2 error		L^∞ error		CPU
32^2	2.47E-02	—	3.29E-01	—	7.00	2.44E-02	—	3.27E-01	—	7.25	2.47E-02	—	3.29E-01	—	6.87
64^2	1.16E-02	1.10	1.88E-01	0.81	54.2	1.02E-02	1.26	1.68E-01	0.96	56.6	1.16E-02	1.10	1.87E-01	0.81	53.9
128^2	2.36E-03	2.30	3.03E-02	2.63	445	2.12E-03	2.26	2.92E-02	2.53	475	2.36E-03	2.30	3.03E-02	2.63	443
256^2	4.14E-04	2.51	5.07E-03	2.58	3633	4.02E-04	2.40	6.33E-03	2.21	3776	4.14E-04	2.51	5.07E-03	2.58	3623
512^2	5.61E-05	2.88	6.78E-04	2.90	30006	5.44E-05	2.88	6.09E-04	3.38	30392	5.61E-05	2.88	6.78E-04	2.90	29425
Expected order		3		3			3		3			3		3	

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

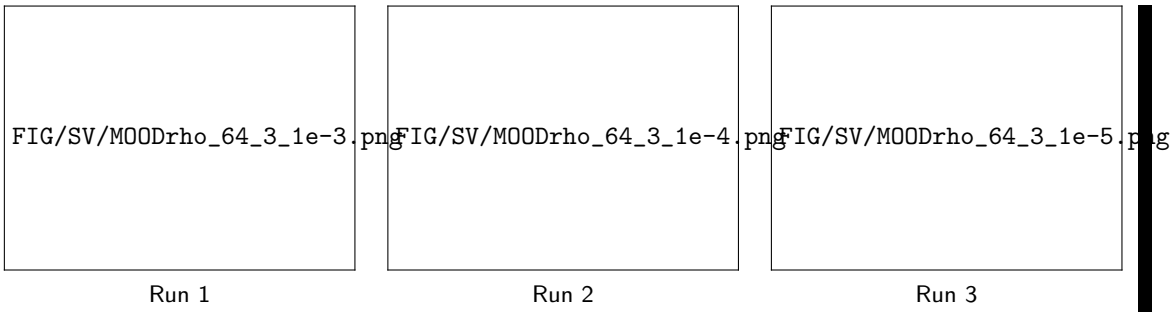


Figure 8: Isentropic vortex problem — Zoom on the final AMR grids for different threshold values (Run 1, 2, 3) starting with 64×64 uniform cells and $\ell = 3$ levels of refinement — MOOD results with DMP+u2 on density.

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^∞ 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 must always compute the non-linear weights. Due to the smoothness of this test, the non-linear weights will be close to the linear ones, but not exactly equal, therefore the errors are slightly different and this technique is slightly more expensive in terms CPU time. As already mentioned, all schemes tested are almost equivalent for uniform grids, as the lower errors obtained by CWENO are counterbalanced by the longer computational times.

5.2.2. AMR grids

In this section we consider the AMR. The coarse grid is made of $N^0 \times N^0$ square cells, with $N^0 = 32, 64$ and 128, and the AMR algorithm employs $\ell = 3$ levels, leading to $N^{eq} = 256$, 512 and 1024 respectively. The value $S^{\text{ref}} = \mathsf{e} = 10^{-3}$ was chosen for the run with $N^0 = 32$ and $\ell = 3$ so that the entire vortex was refined. 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 $\mathsf{e}/8$ and $\mathsf{e}/64$ (see [27, 29]). In order to assess the influence of the refinement parameter on the results, each of these reference runs was compared with simulations computed with a 10-fold increase and 10-fold decrease in S^{ref}, see Table 2.

As expected, the AMR grid initially covers, and later follows, the vortex during its drift for any N^0 and

	Threshold value. Base: $\mathsf{e} = 10^{-3}$		
N^0	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 2: Isentropic vortex problem — Threshold values used for different meshes.

all the schemes. In figure 8 we show the meshes and density (color) at final time for the three threshold values from table 2 and $N^0 = 64$, when the DMP+u2 criteria is employed on the density variable. We observe that the threshold value reduces or increases the layer of fine cells around the vortex structure for all schemes. Its choice hence controls the number of fine cells employed to capture small scale structures and, as a consequence, the CPU time for the simulation and the final error (see table 3). Ideally the error for (N^0, ℓ) in this table should be comparable with the error from table 1 with $N_c = N^{eq} \times N^{eq}$, but the computational time is expected to be lower. In this table we also report the final total number of cells $N_c(t)$ and the CPU time needed to complete the simulations.

To ease the analysis of these data, we have reported them in figure 7 as 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 $N^0 = 128$. Within each triplet, as the threshold is lowered, 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 curves obtained with uniform grids (straight lines) meaning that the AMR simulations are more accurate and less expensive. Moreover we have plotted 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 with N^0 and N^{eq} respectively, since an AMR grid should provide better errors than the uniformly coarse mesh and cannot do better than the uniformly fine one. In these AMR simulations we

$(N^0, \ell), N^{eq}$	Run	$N_c(t)$	L^2 error	L^∞ error	CPU	$N_c(t)$	L^2 error	L^∞ error	CPU	$N_c(t)$	L^2 error	L^∞ error	CPU
		Unlimited (with PAD check)				CWENO				MOOD DMP+u2 on density			
(32,3),128	1	1117	1.35E-02	2.13E-01	36.84	1150	1.06E-02	1.77E-01	38.24	1117	1.35E-02	2.13E-01	37.63
	2	1678	2.99E-03	3.84E-02	55.25	1687	2.71E-03	3.77E-02	54.96	1678	2.99E-03	3.84E-02	55.14
	3	2407	2.38E-03	3.15E-02	77.97	2380	2.15E-04	3.07E-02	77.22	2407	2.38E-03	3.15E-02	79.69
(64,3),256	1	5254	1.93E-03	2.49E-01	333.26	5305	1.74E-04	2.26E-02	332.62	5194	1.92E-03	2.51E-02	341.80
	2	6757	5.80E-04	5.83E-03	478.43	6832	5.64E-04	6.53E-03	477.02	6757	5.80E-04	5.83E-03	483.21
	3	9277	4.15E-04	5.12E-03	684.24	9505	4.03E-04	6.34E-03	662.18	9277	4.15E-04	5.12E-03	689.91
(128,3),512	1	21496	2.97E-04	2.91E-03	3099.37	21559	2.91E-04	3.13E-03	3036.33	21514	2.97E-04	2.99E-03	3173.55
	2	27757	8.35E-05	7.99E-04	4530.64	27901	8.13E-05	7.32E-04	4347.57	27757	8.35E-05	7.99E-04	4583.56
	3	36712	5.62E-05	6.71E-04	6364.56	36664	5.45E-05	6.09E-04	6145.22	36712	5.62E-05	6.71E-04	6415.37

Table 3: Final number of cells $N_c(t)$, L^2 and L^∞ 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 DMP+u2 detection on density schemes. N^0 is the number of cell in each direction for the initial grid, ℓ if the number of refinement level leading to N^{eq}, the number of cell for each direction in an equivalent fixed grid.

observe an almost constant number of cells, in total as well as in each refinement level.

5.3. Radial Sod shock tube

In this section we apply the schemes to a discontinuous solution, for which some amount of limiting is mandatory. In particular, we run the cylindrical Sod shock tube problem in order to assess the ability of the scheme to capture simple but non-aligned cylindrical waves without deforming them nor 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 computational domain is set to $\Omega = [0; 1] \times [0; 1]$ with symmetry boundary conditions on $x = 0$ and $y = 0$, and outflow boundary conditions otherwise. The final time is set to $t_{\text{final}} = 0.2$. This flow can be computed also as a one-dimensional Riemann problem in the radial coordinate (see [33]) and our reference solution is computed using a classical 1D MUSCL TVD scheme with 25000 cells.

5.3.1. Uniform grids

In figure 9 we present the MOOD troubled cell marker at final time on uniform meshes for MOOD with DMP+u2 on entropy (middle) or DMP+u2 on density (right) schemes. Each panel is decomposed into four quadrants and each quadrant is dedicated to one particular uniform mesh. The red cells are the problematic ones, that got updated by the first order scheme, while the blue ones were updated with the unlimited 3rd order scheme. Note that the 256×256 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) produces different patterns, the density criteria better capturing the cylindricity of the wave. In figure 10 we present the density as a function of the cell center radius for all cells in the final mesh, focusing on the contact and shock zones in the middle and bottom panels. As expected, when the mesh is refined, the waves (contact, shock) and the tail/head of the rarefaction are sharper and converge to the correct position according to the reference solution. On average, the same drawbacks can be attributed to all schemes: when an excessive diffusion is not observed ($N \geq 128$) then over- and under-shoots occur.

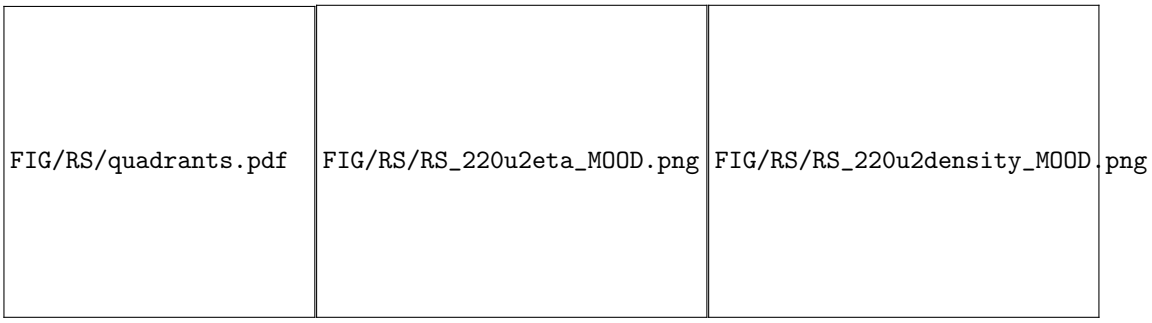


Figure 9: 2D radial Sod shock tube problem at $t_{\text{final}} = 0.2$ — Uniform $N \times N$ quadrangular mesh with $N = 32$ (NE), 64 (SE), 128 (SW), 256 (NW) — Meshes and valid/troubled cells (blue/red). Left: LEGEND, middle: MOOD with DMP+u2 on entropy, right: MOOD with DMP+u2 on density results.

While these are not observed at the same location for the three tested scheme, nonetheless they are a general characteristic. Note that applying the DMP+u2 on density or entropy does mark different features on the solution (see Fig. 9) and, accordingly, changes the pattern of spurious oscillations.

At last, we present the error in L^2 and L^∞ norm for three grids in table 4. Because of the presence of discontinuities we can not even expect a first order of convergence of all schemes on this test for L^2 norm. These data show that the three schemes provide the same accuracy even if MOOD schemes seem slightly more accurate.

5.3.2. AMR grids

For the AMR runs we employed a coarse grid with $N^0 = 32$ and varied the levels of refinement from $\ell = 3$ to 5. The threshold value $\mathsf{e} = 10^{-3}$ was chosen by examining few $(N^0, \ell) = (32, 3)$ runs and ensuring that the shock, the contact and the tail/head of the rarefaction were correctly refined. The thresholds for the other grids (see figure 11) 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 [27, 29]. Then we employ the thresholds $\mathsf{e} = 10^{-3}/2^q$ for $(N^0, \ell) = (32, 3 + q)$, and $q = 0, 1, 2$.

In figure 12 we present the results in four quadrants for each panel, as described in the legend. Note that the uniform mesh result with $N^0 = 256$ (top-left) should be compared to the AMR results ($N^0 = 32, \ell = 4$) (bottom-right), while the top-right and bottom-left panel should be less and more accurate respectively. In this figure one can see, from top to bottom, the troubled cells detected by MOOD and the meshes. The same three schemes are tested as before. The number of cells is drastically reduced with the AMR procedure and the refinement zones are located along the cylindrical waves as expected, apart from small 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 and thus the AMR results are equivalent in terms of accuracy to the uniform grid simulations of the previous section, showing that the AMR procedure couples effectively with the *a posteriori* MOOD limiting. Next, in figure 13, we plot the results as a function of the cell radius, as in figure 10, but only focusing on zooms. The same conclusion as in the uniform case holds.

In figure 14 we present, for the MOOD schemes (DMP+u2 on density), the histograms of the number of cells N^k in each of the refinement level as a function of the time when $\ell = 4$. Equivalent results are observed for $\ell = 3$, and $\ell = 5$, for MOOD scheme with DMP+u2 on entropy and for CWENO and therefore we omit them. 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

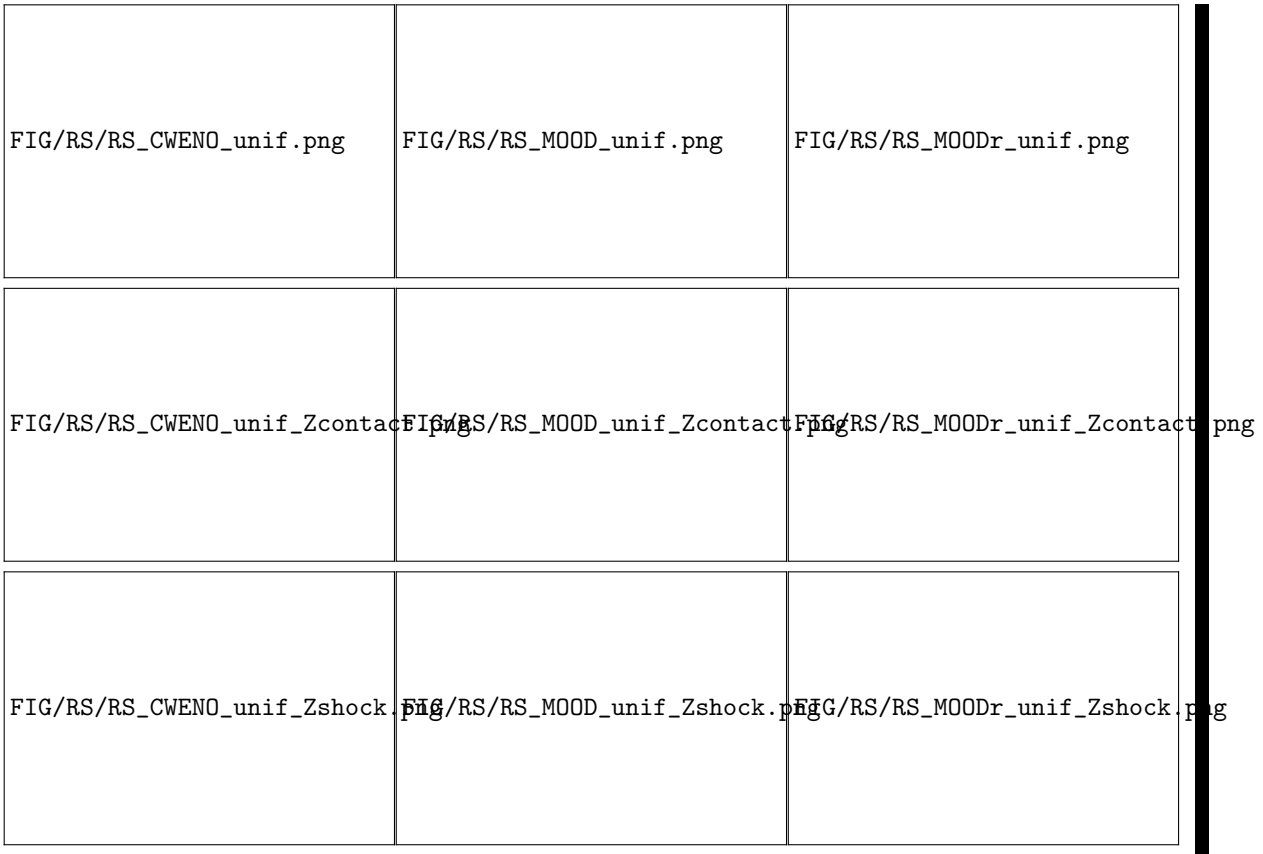


Figure 10: 2D radial Sod shock tube problem at $t_{\text{final}} = 0.2$ — Uniform $N \times N$ quadrangular mesh with $N = 32, 64, 128, 256$ — Density as a function of cell center radius for all cells — Left: CWENO, middle: MOOD with DMP+u2 on entropy, right: MOOD with DMP+u2 on density — Top: full view — Middle: Zoom on the contact — Bottom: Zoom on the shock.

	Threshold value S^{ref}		
(N^0, ℓ)	Run 1	Run 2	Run 3
(32,3)	**e**= 10^{-3}	0.1 e	0.01e
(32,4)	**e/2**	0.1 e/2	0.01e/2
(32,5)	**e/4**	0.1 e /4	0.01e/4

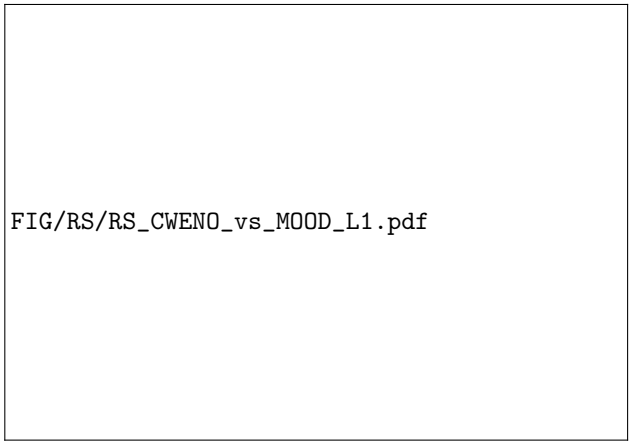


Figure 11: Radial Sod problem — Left: table of refinement thresholds for AMR runs. Right: L^1 error curves for CWENO and MOOD simulations as a function of CPU time. Results for uniform (line) and AMR (symbols) grids for all simulations. Each symbols corresponds to a specific threshold parameter, while each set of three symbols employs a grid made of $(32, 3)$, $(32, 4)$ or $(32, 5)$ cells.

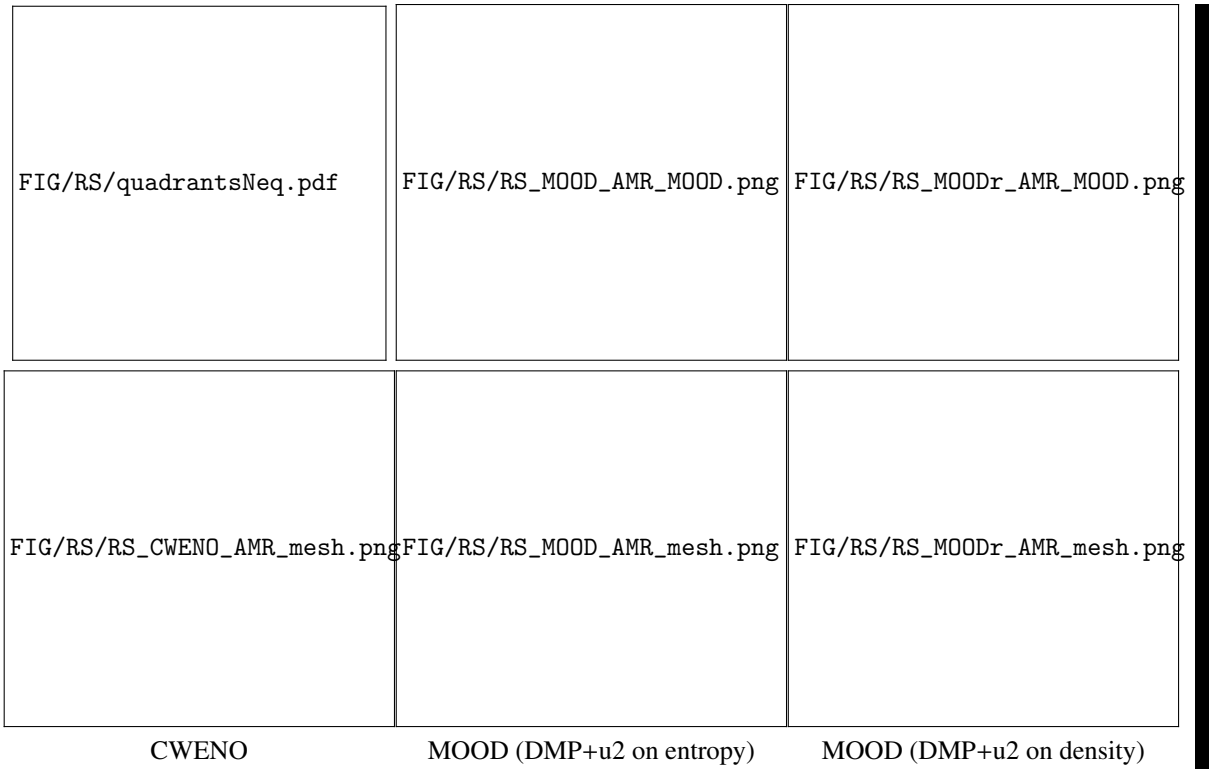


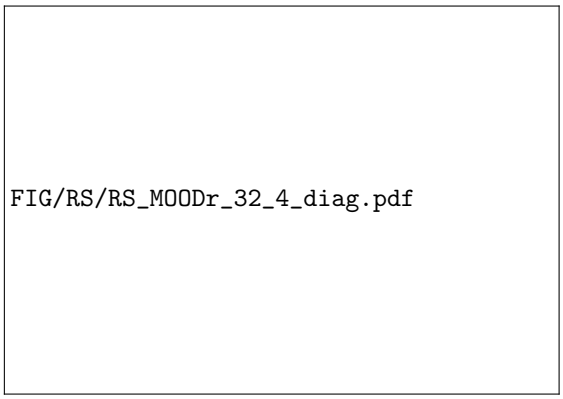
Figure 12: 2D radial Sod shock tube problem at $t_{\text{final}} = 0.2$ — AMR simulations with $N^0 = 32$ and $\ell = 3$ (NE), $\ell = 4$ (SE), $\ell = 5$ (SW), and uniform 256×256 mesh results (NW) — MOOD marker and meshes — Left: CWENO, middle: MOOD with DMP+u2 on entropy, right: MOOD with DMP+u2 on density.

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 256×256 mesh results (NW) — Density as a function of cell center radius for all cells — Left: CWENO, middle: MOOD with DMP+u2 on entropy, right: MOOD with DMP+u2 on density — Top: Zoom on the contact — Bottom: Zoom on the shock.

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. In the same figure (right panel) we plot the evolution of the total number of recomputed cells in the MOOD loop as a function of the time for three refinements $\ell = 3$, 4 and 5, showing that the MOOD limiting procedure needs to recompute only a small amount of cells. In table 4 we report the final number of cells $N_c(t)$, L^2 and L^∞ errors and, CPU time needed by the three schemes under AMR strategy. Moreover we have computed the acceleration (Acc.), that is the ratio between the CPU time of the N^{eq} uniform grid simulation, and the CPU time of the AMR simulation. We observe that in this test the number of cells may be truly different even if the threshold parameters are the same for given (N^0, ℓ). In this test, MOOD with DMP+u2 on entropy is the scheme which refines the most, CWENO the less. Note that it is to be expected that different spatial discretizations lead to different levels of numerical entropy production and thus to different refinement patterns. 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 schemes 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 are 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 ℓ. The L^1 error norm obtained from different threshold values as a function of the CPU time needed to complete the simulation are reported in figure 11 with symbols. This figure proves that the AMR simulations are more efficient than uniform ones and also shows that MOOD simulations are slightly more efficient than CWENO.

5.4. Colella-Woodward blastwave

Here we show a test for which has difficulty in completing the simulation: the 1D blastwave problem of Colella-Woodward [37] that simulates the collision of two strong shocks. The test is designed on $\Omega = \Omega_L \bigcup \Omega_M \bigcup \Omega_R = [0, 0.1] \bigcup [0.1, 0.9] \bigcup [0.9, 1]$ and the initial condition consists of three constant states at



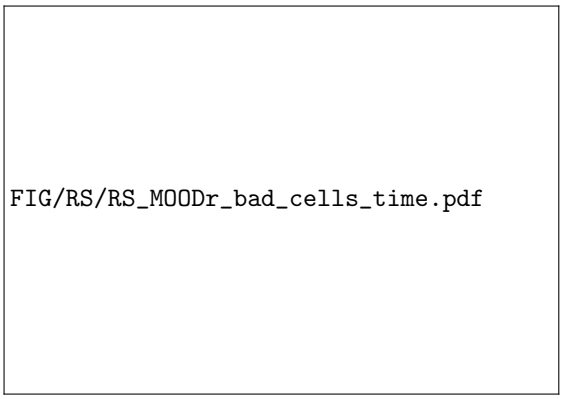


Figure 14: 2D radial Sod shock tube problem — MOOD scheme with DMP+u2 on density — Left: Results for an AMR mesh with 4 levels of refinements $(N^0, \ell) = (32, 4)$. Colored histograms for the number N_k $(k = 0, \cdots 4)$ of cells of each level (black vertical scale on the left) — Right: Evolution of the ratio between the number of recomputed cells and the total number of cells as a function of time for 3, 4 or 5 levels of refinements and $N^0 = 32$.

$(N^0, \ell), N^{eq}$	$N_c(t)$	L^2 error	L^∞ error	CPU	Acc.	$N_c(t)$	L^2 error	L^∞ error	CPU	Acc.	$N_c(t)$	L^2 error	L^∞ error	CPU	Acc.
	CWENO					MOOD DMP+u2 on density					MOOD DMP+u2 on entropy				
(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	2.82E-02	5.88E-02	91.4	2.8	5251	2.53E-02	6.13E-02	145.3	2.5	4852	2.49E-02	5.81E-02	145.1	2.5
(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	1.92E-02	5.41E-02	517.8	4.1	9709	2.42E-02	5.64E-02	830.4	3.9	18910	1.67E-02	5.39E-02	813.2	3.9
(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	1.27E-02	4.83E-02	2621.05	6.7	32011	1.14E-02	5.10E-02	5258.2	5.2	51796	1.10E-02	4.62E-02	4744.5	5.6

Table 4: Final number of cells $N_c(t)$, L^2 and L^∞ errors, CPU time, for the radial Sod problem simulated by several 3rd order schemes using AMR capability. Results for the CWENO and MOOD with DMP+u2 detection or density schemes. N^0 is the number of cell in each direction for the initial grid, ℓ 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, ℓ) simulation).

rest: $(\rho_L, p_L) = (1, 1000)$, $(\rho_M, p_M) = (1, 0.01)$, and $(\rho_R, p_R) = (1, 100)$. The final time is set to $t = 0.04$ and reflective boundary conditions are considered.

In this test, two very strong shocks as well as weaker waves originate from the initial discontinuities. Just before the collision of the larger waves (at $x \simeq 0.69$), when the strong shocks are separated by just one cell, the CWENO reconstruction produces a non-physical boundary extrapolated value with a negative pressure. This is due to the presence of very large gradients on both sides of the cell and thus to the absence of a smooth enough substencil among those considered by CWENO. This situation can not be overcome unless some special procedure, which would nevertheless locally lower the accuracy, is devised for these cells. On the other hand, a first-order finite volume scheme can run this simulation without the occurrence of unphysical states. The *a posteriori* MOOD approach exploits this fact: once it detects negativity or NaN, the incriminated cells are recomputed employing a stable, albeit first-order accurate, scheme.

In figure 15 we propose the results produced by MOOD using DMP+u2 criteria on density when $\ell = 8$ levels of refinements are employed with initially $N^0 = 32$ uniform cells ($N^{eq} = 4096$). A space-time diagram is proposed and the pressure, the recomputed cells and the refinement levels are plotted in colors on respectively the left, middle and right panels. The main and reflected/refracted waves are captured by the AMR procedure (see the colors on the right panel representing fine cells). MOOD detection procedure is also appropriately locating troubled cells along the main waves and, as such, can simulate the collision and interaction of shocks. Zooming around the main shock collision reveals that only 4 cells are updated at first order accuracy for 4 timesteps in order to compute the interaction of these very strong waves.

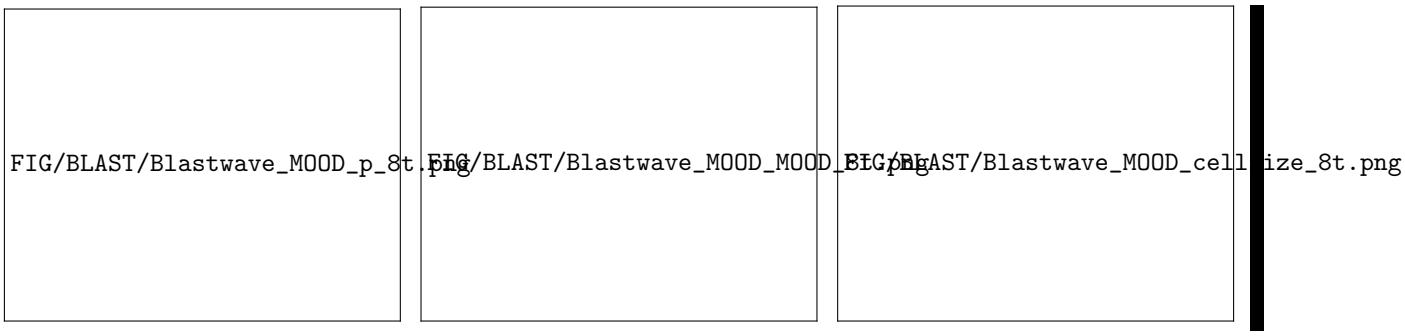


Figure 15: Colella-Woodward blastwave problem — Results of MOOD with DMP+u2 on density on a space-time diagram for $(N^0, \ell) = (30, 8)$ — Left: pressure variable (colors) — Middle: recomputed cells (red) vs unlimited cells (blue) — Right: cell refinement level from 0 to 7. (Note that CWENO produced negative pressure in regions where shocks are separated by one cell.)

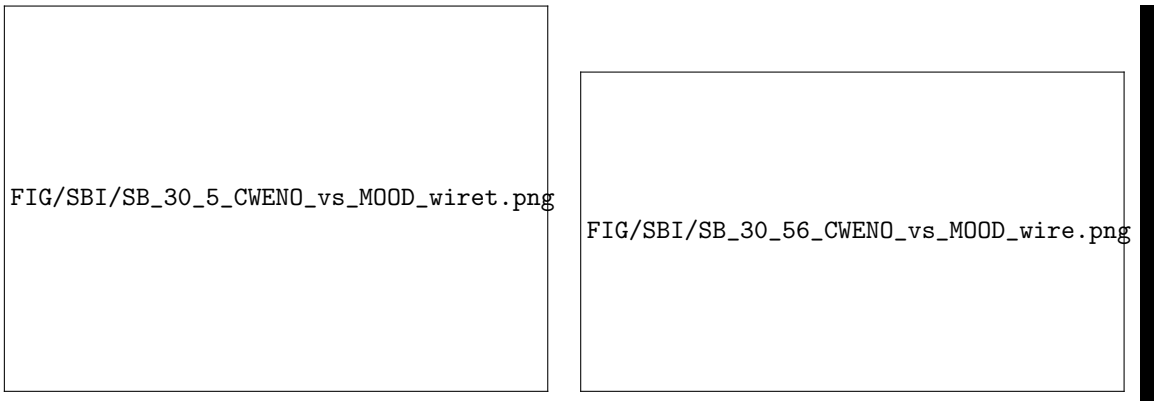


Figure 16: Shock-bubble interaction problem — Results at final time for CWENO (top panels) and MOOD (bottom panels) — AMR mesh colored by density — Left: comparaison at same resolution $(N^0, \ell) = (30, 5)$ — Right: comparaison when CWENO has one more level of refinement.

5.5. Shock bubble interaction

Here we report a test case which enlightens the lower dissipation of MOOD versus CWENO. We consider the classical shock-bubble interaction problem, in which a right-moving shock impinges on a standing bubble of gas at low pressure, see [4, 29]. 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 $x > 0$ and outside the bubble. The initial data are

$$\rho = \frac{11}{3},\ u = 2.7136021011998722,\ v = 0.0,\ p = 10.0 \qquad \text{if } (x, y) \in A,$$

$u = v = 0.0$, $p = 1.0$ otherwise and $\rho = 0.1$ in zone B and $\rho = 1.0$ in zone C. 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 $t_{\text{final}} = 0.4$. We start with a coarse grid made with squares of characteristic length $h = 1/N^0$ with $N^0 = 30$. The threshold parameter is 0.03 for $\ell = 5$ and 0.1 for $\ell = 4$.

5.5.1. CWENO vs MOOD

Our first objective with this test case is to compare on the same refinement level MOOD and CWENO approaches. On the left panel of figure 16 we plot the results of both schemes when $\ell = 5$ levels of

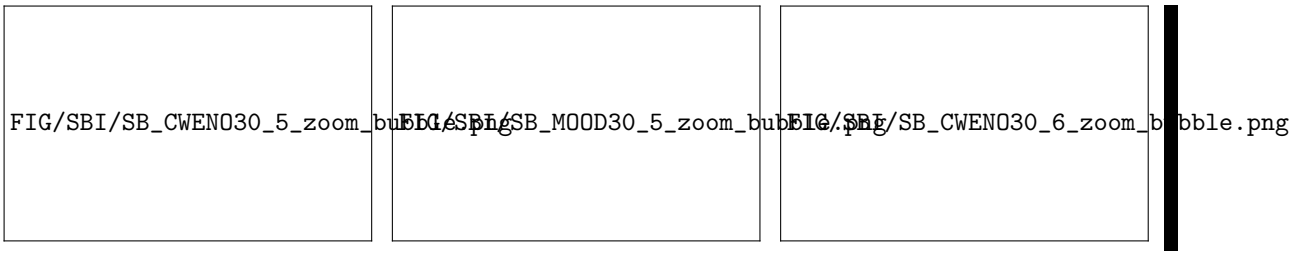


Figure 17: Shock-bubble interaction problem — Results at final time for CWENO $(N^0, \ell) = (30, 5)$ (left) $(30, 6)$ (right) and MOOD $(N^0, \ell) = (30, 5)$ (middle) — AMR mesh colored by density — Zoom on the bubble location. The same color scale as in figure 16 is used. Note that the faded colors for CWENO and MOOD with $(N^0, \ell) = (30, 5)$ are due to the one less level of refinement

refinements are employed. We plot the AMR mesh colored by the density variable for CWENO (top panel) and MOOD (bottom panel) to enhance the waves and the small-scale structures. Even if the very same resolution is employed, MOOD produces visibly less dissipative results than CWENO. On this complex and long simulation, MOOD captures smaller structures and the bubble surface is already presenting instabilities which is not the case for CWENO. In the right panel of figure 16 we show that the CWENO approach can also capture the details of the breaking bubble if allowed to employ one extra level of refinement.
This simulation shows that MOOD approach can outperform CWENO on simulations when small scale structures are generated. Last, in figure 17, to enhance the differences, we present zooms around the bubble location for CWENO results obtained with $(N^0, \ell) = (30, 5)$ (left panel), $(30, 6)$ (right panel) and MOOD with $(N^0, \ell) = (30, 5)$ (middle panel). In this test the MOOD approach with DMP+u2 criteria applied on entropy, in the $(30, 5)$ AMR run, misses a spurious oscillation that leads to the formation of a spurious, tiny, nearly vacuum, vortex close to the edge of the bubble. This numerical artefact is not diffused away in the next timesteps, eventually driving the sound speed to very high values and the timestep to very small ones, grinding the simulation to a halt. No such effect is apparent when the detection chain is applied to the density variable. For this reason and since the detection on density and on entropy performed on par in the previous tests, from now on we will show only results obtained with the detection applied on density.

5.5.2. Quantitative diagnostics for MOOD simulations

We display in figure 18 the results for the case $\ell = 5$ at time $t = 0.2$, 0.3 and at final time 0.4. In these figures we observe that the deformation of the bubble agrees well with already published results. Moreover, the numerical solution accuracy competes for instance with that obtained in [29]. The refinement follows the different wave structures (shocks, vortices, interacting waves, etc), as can be appreciated 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 and along some secondary waves, resulting in a very small amount of limiting. Consequently, most of the cells in the computational domain are effectively calculated with the unlimited third-order scheme, avoiding any extra-cost brought by the limiting apart from the MOOD detection. The refinement and detection strategies seem to cooperate effectively and the sharpness of small scale structures can be captured, maintained and later released when appropriate. The final number of cells reaches about 125000 cells (starting at 12000) for $\ell = 5$.
In figure 19 we plot the statistics on the number of cells for each refinement level (left for $\ell = 4$ and middle for $\ell = 5$) and the ratio between the number of recomputed cells and the total number of cells during the simulation (right panel). Consistently with the previous tests, the number of cells for the two finer levels increases in time. 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 amount of bad cells is of the order of 3%

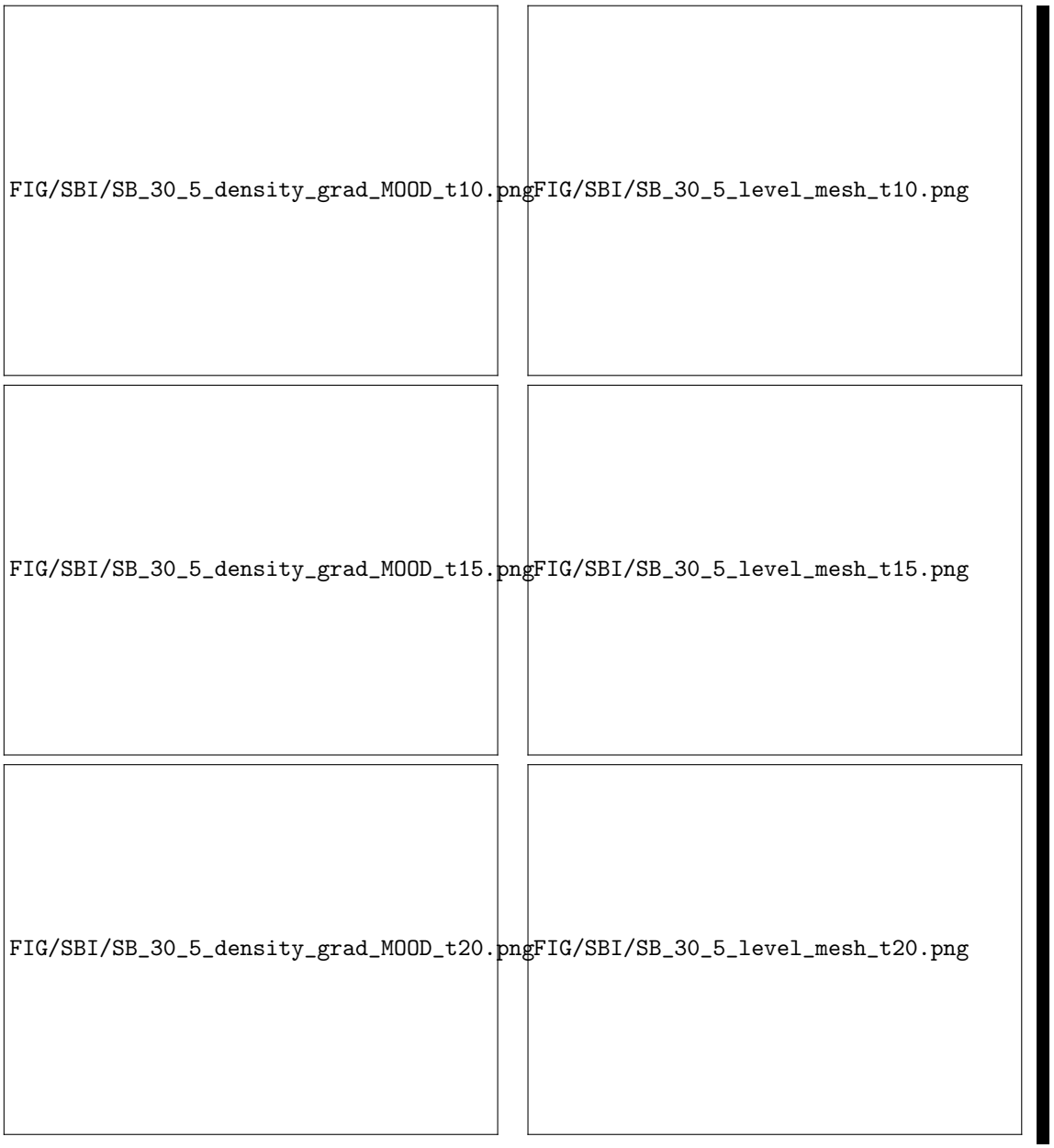


Figure 18: Shock-bubble interaction problem — Mesh with $\ell = 5$ refinement levels — Top-bottom: time $t = 0.2$, 0.3 and final 0.4 — Left panels: Density (top with colors), norm of velocity gradients (bottom with gray) and bad cells detected by MOOD (bottom in red) — Right panels: Cell levels (top with colors), adapted mesh (bottom)

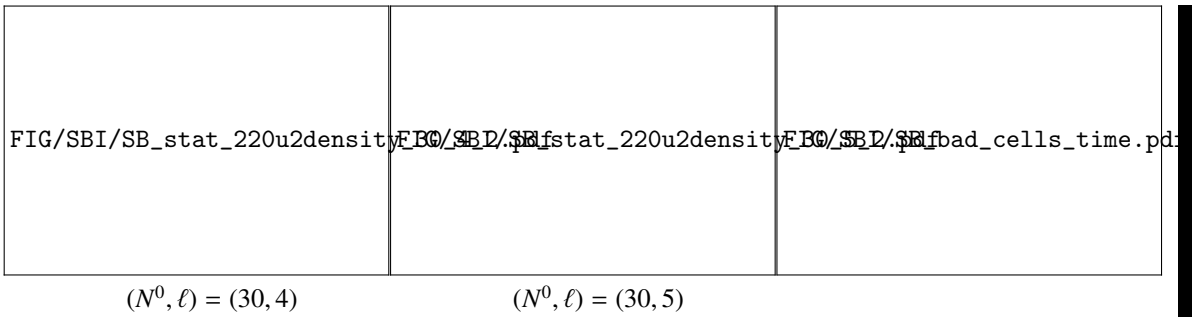


Figure 19: Shock-bubble interaction problem — Results for a mesh with 4 (left) or 5 (middle) levels of refinements — Left-middle: Colored histograms for the number of cells of each level (N_k, $k = 0, \cdots 2$ or 4) as a function of time — Right: ratio between the number of recompted cells and the total number of cells as a function of time.

after $t = 0.025$, and about 10% before. In other words, on average, only $4\% - 5\%$ of the cells demand to be recomputed, leading to an efficient way to limit and stabilize the numerical scheme.

In table 5 we gather the CPU time needed to perform the simulations. Note that, for these cases, the

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

Table 5: 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 $\ell = 5$. 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 indicates 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 $1/14$th (resp. $1/30$th) of cells at initial time, when the waves are very localized.

5.6. Double Mach reflection problem

The 2D double Mach reflection problem of a strong shock was proposed in [37] and is challenging due to the contemporaneous presence of strong waves, very weak ones and of complex smooth features, in the so-called recirculation zone. This problem involves a Mach-10 traveling shock in a perfect gas which hits a ramp at 30° with the x-axis. The initial conditions in front of and after the shock wave are given by

$$(\rho, u, v, p)(\mathbf{x}, t = 0) = \begin{cases} \frac{1}{\gamma}(8.0, 8.25, 0.0, 116.5), & \text{if} \quad x' < 0.1, \\ (1.0, 0.0, 0.0, \frac{1}{\gamma}), & \text{if} \quad x' \geq 0.1, \end{cases} \tag{20}$$

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 of square cells employing $N^0 = 30$ of them in the y direction. 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 840×240, 1680×480 and 3360×960 cells. 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 results are presented in color for the density variable at $t_{\text{final}} = 0.2$ in figure 20. There we also present the final AMR meshes along with the location of the bad cells detected by the MOOD procedure (red cells). Zooms for the y-momentum are proposed on the bottom panels of the same figure to enhance the shear waves, the meshes, the bad cells and the re-circulation zones. Left panels gather the results for $\ell = 5$ and right ones for $\ell = 6$ levels of refinement. One more extra level of refinement improves the sharpness of the small scale structures: 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).
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.
Next, we focus on diagnostics on the number of cells N^k at each level. In left panel of figure 21 we present the histograms of N^k as a function of time for the case $\ell = 4$. We observe that, despite the fact that cells on the finest level cover a small portion of the domain, they are the most numerous (which also occurs for lager values of ℓ) and their number increases in time as more structures to be refined appear and/or get bigger. The number of cells in intermediate levels are slightly increasing during the simulation. The right panel of figure 21 presents the ratio between recomputed and total cells as a function of time for three levels of refinement $\ell = 4$, 5 and 6: it never exceeds 3% except in the very first steps.
In table 6 we gather more diagnostics for these AMR runs: number of cells (final, average and initial), ratio

(N^0, ℓ)	N^{eq} unif.	$N_c(t), \overline{N}_c, N_c(t=0)$	Ratio	CPU	CPU unif.	Ratio
(30,4)	$840\times240 \simeq 2\times10^5$	30253, 17982, 4954	7, 11, 41	3.3h (3h)	≈16.4h (16h)	5
(30,5)	$1680\times480 \simeq 8\times10^5$	79549, 46310, 7411	10, 17, 109	20h (1d)	≈149h (6d)	7.5
(30,6)	$3360\times960 \simeq 32\times10^5$	274989, 136978, 12247	12, 24, 263	187h (8d)	≈1346h (56d)	7

Table 6: 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.

versus the equivalent uniform number of cells, CPU time, CPU time for a uniform equivalent number of cells, and their ratio. 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 indicates that AMR simulations are five to seven times faster than uniform ones. The efficiency of the AMR scheme is linked to its ability to perform the computation with 10 to 25 times less cells on average. The ratios of N^{eq} versus AMR cells keeps increasing when finer and finer levels are allowed because the main waves in this problem are well separated.

5.7. Forward facing step

The forward facing step (FFS) problem from [37] is considered next. The challenges of this problem are the initial boundary layer on the step, the flow around the corner and the emergence of shock waves bouncing on the walls and interacting among themselves. In this test, a Mach 3 flow enters in a 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 $\Omega = [0, 3] \times [0, 1] \backslash [0.6, 3] \times [0, 0.2]$. Reflective wall boundary conditions are applied on the upper/lower boundaries of the domain; at the inflow boundary (left) we prescribe the Mach 3 flow, while outflow boundary conditions are are employed on the right end of the domain. The final time is set to $t_{\text{final}} = 4$.

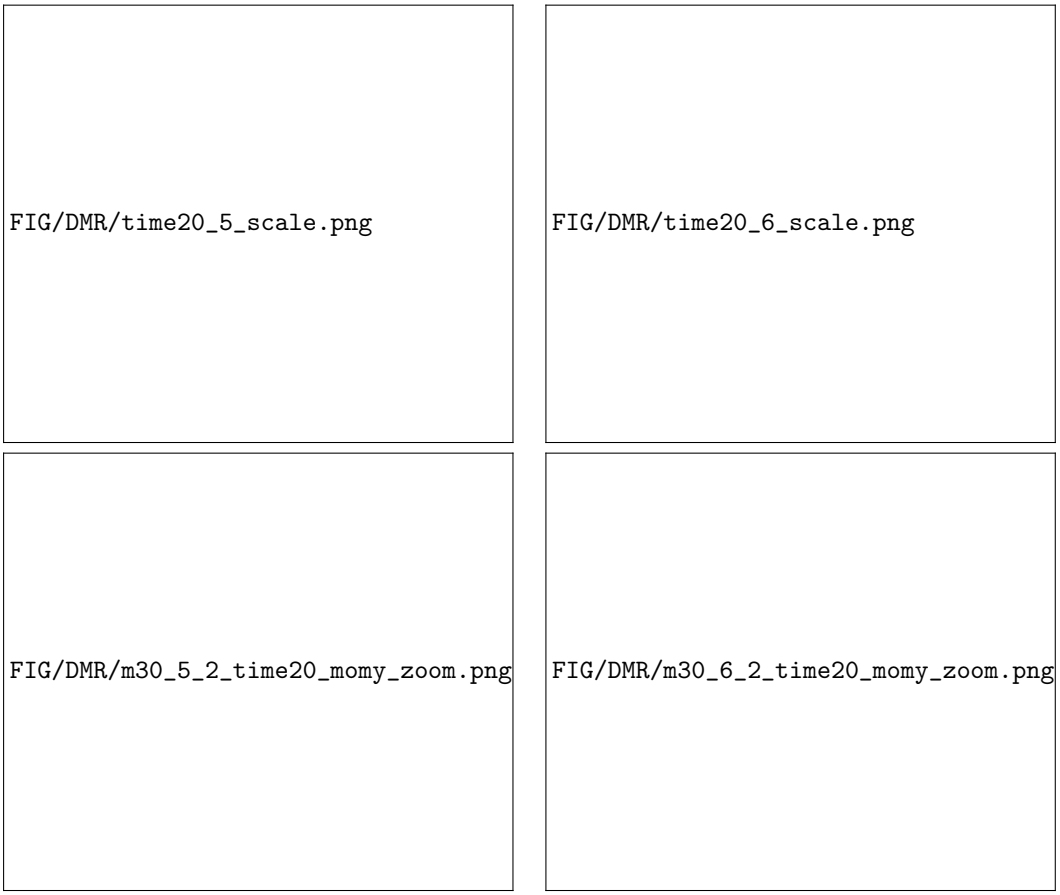


Figure 20: Double Mach reflection problem at $t_{\text{final}} = 0.2$ with AMR — Final colored density/y-momentum (top parts) and mesh and troubled cells (bottom parts) — Results for a mesh with $\ell = 5$ (left), and $\ell = 6$ (right) levels of refinements. Top: full view on density, bottom: zoom on y momentum to enhanced the shear waves.

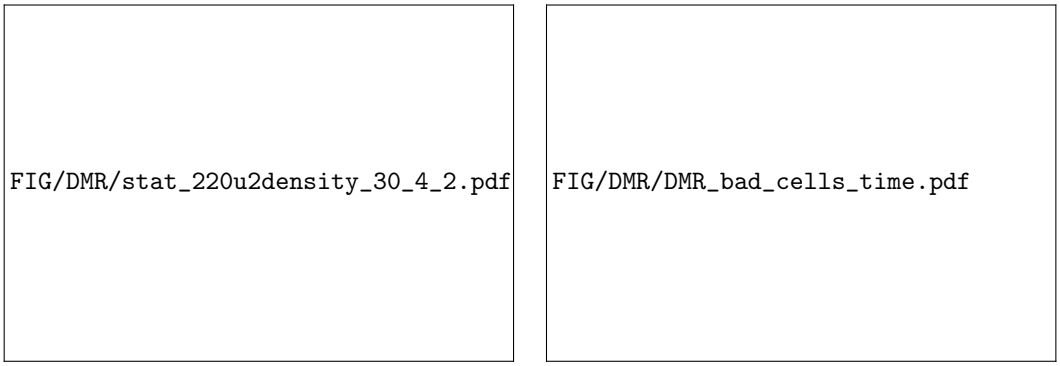


Figure 21: Double Mach reflection problem at $t_{\text{final}} = 0.2$ with AMR — Left: Results for a mesh with 4 levels of refinements. Colored histograms for the number of cells of each level (N_k, $k = 0, \cdots 4$) — Right: ratio between the number of recomputed cells and the total number of cells as a function of time for three levels of refinement $\ell = 4$, 5 and 6.

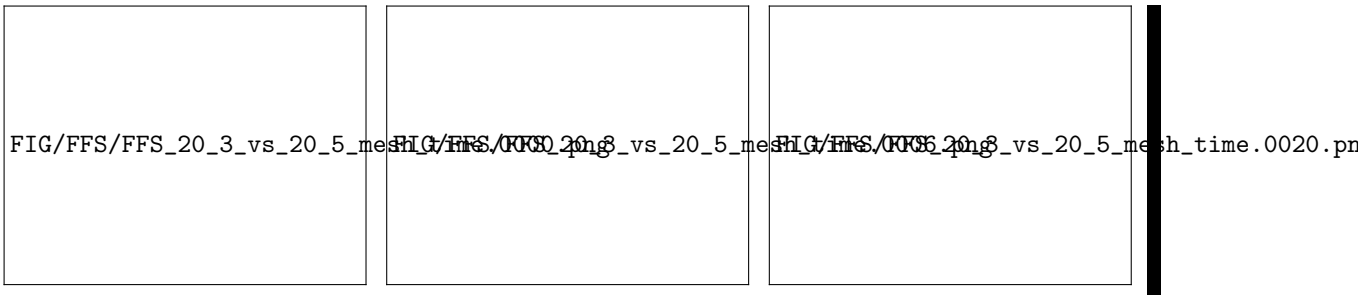


Figure 22: Forward Facing Step problem — Refined meshes with $\ell = 3$ (bottom part of each panel) and $\ell = 5$ (top parts) levels of refinement — Initial (first accepted timestep), intermediate ($t \simeq 1.2$) and final refined meshes.

The initial coarse mesh is uniform with characteristic length $h = 1/20$ in both directions. In this section, we show results for $\ell = 3$ and 5 levels of refinement, with the threshold parameter fixed to $S^{\text{ref}} = 0.05$ and 0.0125 respectively. For comparison purposes, we plot the case $\ell = 5$ on the top part of each figure, while the case $\ell = 3$ is reflected with respect to the x axis and plotted on the bottom part of each panel.

In figure 22 the initial, intermediate ($t \simeq 1.2$) and final refined meshes are plotted. Since the initial datum is constant, the initial gradient-based mesh adaption cannot be applied and all simulations start with a uniform coarse mesh of 1009 cells. The AMR procedure then adapts the mesh, during the first timestep, refining it up to the finest level close to the corner of the domain from where the first fluid structures will emanate (left panel). Later, the refinement zones follow the main waves and structures and it is clear that the $\ell = 5$ simulation is a lot sharper in capturing the dynamical elements of the flow.

In figure 23 we display, from top to bottom, the results at time $t = 0.4$, $t = 1.2$ and at final time. 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 shock waves. The refined regions are properly covering the waves and follow their motion. As expected, the great majority 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. Note that the $\ell = 5$ simulations can capture some waves at the right end of the domain, which are almost completely missed by the $\ell = 3$ run.

(N^0, ℓ)	N^{eq} unif.	$N_c(t), \overline{N}_c, N_c(t = \Delta t)$	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)	≈169h (1w)	9.9
(20,5)	258048	40588, 39523, 1909	6.4, 6.6, 135	130h(5d)	≈1350h (1.5m)	10.4

Table 7: 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).

For this test our simulation code demands about 129h36mn (about 5 days) for $\ell = 5$ and employs on average 40000 cells, and about 2h49mn for $\ell = 3$ and 6000 cells. The data in table 7 show that the ratio of mesh sizes is of the order 3 to 6.5, The simulation times with the uniform mesh were forecast by extrapolating the run time of the $(N^0 = 60, \ell = 1)$ run with an 8-fold increase for each mesh doubling. The

Figure 23: 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.

ratios of the simulation times range between 7 to 10.

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 the fail-safe MOOD approach, this parameter does not play a fundamental role in ensuring robustness.

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. Third order accuracy is obtained through SSP-Runge-Kutta discretization in time and third order accurate polynomial reconstructions in space. The novelty of this work is the coupling of AMR with the *a posteriori* Multi-dimensional Optimal Order Detection (MOOD) approach to damp the spurious numerical oscillations that may occur in an unlimited high order scheme: limiting is in fact achieved by local reduction of the polynomial degree of the reconstructions. In this work, as a proof of concept, only $\mathbb{P}_2$ or $\mathbb{P}_0$ reconstructions are considered. AMR based on numerical entropy production thresholding has been designed as an *a posteriori* technique in [27, 29] and therefore its association with an *a posteriori* limiting procedure seems fairly natural and proved effective. A large list of test cases have been simulated and we have systematically observed that this 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, and few falsely refined regions are generated. As expected the CPU time of AMR simulations is genuinely smaller than that of equivalent uniform mesh simulations, the acceleration is about 4 to 5 for an equivalent accuracy. This allows us to simulate refined mesh simulations even on serial computer within a rather acceptable duration. Accurate results are observed on complex flows, and, as reported in this paper, only few percents of troubled cells need to be recomputed. As a consequence the extra-cost brought by the MOOD loop is low.

In the future we plan to investigate the massively parallel version (under MPI) of this numerical method and its extension to 3D. We also plan to investigate even more accurate numerical methods. By increasing the reconstructions up to degree five or six, we plan to build an *a posteriori* limited MOOD scheme for which a CWENO (or a 2nd order TVD scheme) would be used as a second choice, secured by a first-order scheme as parachute. The following cascade would then be employed: $\mathbb{P}_5 \rightarrow$ CWENO (or TVD) $\rightarrow \mathbb{P}_0$.

Acknowledgments

We would like to thank the anonymous referees for their suggestions and remarks, that truly helped us in improving this paper. 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 Franco-Italian Partenariat Hubert Curien "Galileo" (project G14-19 #32272UL) under the supervision of G.Dimarco (University of Ferrara, Italy), and J.Narksi (University of Toulouse, France).

[1] 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(2–3):103–119, 2008.

[2] M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. J. Comput. Phys., 53:484–512, 1984.

[3] M.J. Berger and P. Colella. Local adaptive mesh refinement for shock hydrodynamics. J. Comput. Phys., 82(1):64 – 84, 1989.

[4] M. Cada and M. Torrilhon. Compact third-order limiter functions for finite volume methods. J. Comput. Phys., 228(11):4118–4145, 2009.

[5] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for systems of conservation laws – multi-dimensional optimal order detection (MOOD). J. Comput. Phys., 230(10):4028 – 4050, 2011.

[6] I. Cravero and M. Semplice. On the accuracy of WENO and CWENO reconstructions of third order on nonuniform meshes. Journal of Scientific Computing, pages 1–28, 2015.

[7] S. Diot, S. Clain, and R. Loubère. Improved detection criteria for the multi-dimensional optimal order detection (MOOD) on unstructured meshes with very high-order polynomials. Computers and Fluids, 64:43 – 63, 2012.

[8] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. J. Comput. Phys., 221:693–723, 2007.

[9] 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. J. Comput. Phys., 226:204–243, 2007.

[10] S. Gottlieb and C.W. Shu. Total variation diminishing Runge-Kutta schemes. Mathematics of Computation, 67:73–85, 1998.

[11] C.W. Hirt, A.A. Amsden, and J.L. Cook. An Arbitrary Lagrangian-Eulerian computing method for all flow speeds. J. Comput. Phys., 14(3):227–253, March 1974. doi:10.1016/0021-9991(74)90051-5.

[12] Thomas J. R. Hughes, Guglielmo Scovazzi, and Tayfun E. Tezduyar. Stabilized methods for compressible flows. Journal of Scientific Computing, 43(3):343–368, 2010.

[13] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. J. Comput. Phys., 205:486–508, 2005.

[14] 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.

[15] P. Lax and B. Wendroff. Difference schemes for hyperbolic equations with high order of accuracy. Comm. Pure Appl. Math., XVII:381–398, 1964.

[16] P.D. Lax and B. Wendroff. Systems of conservation laws. Communications in Pure and Applied Math., 13:217–237, 1960.

[17] R. Loubère, M. Dumbser, and S. Diot. A new family of high order unstructured MOOD and ADER finite volume schemes for multidimensional systems of hyperbolic conservation laws. Communication in Computational Physics, 16:718–763, 2014.

[18] R. Loubère, P.-H Maire, M. Y. Shashkov, J. Breil, and S. Galera. Reale: A reconnection-based Arbitrary-Lagrangian-Eulerian method. J. Comput. Physics, 229(12):4724–4761, 2010.

[19] S. M. Murman M. Berger, M. J. Aftosmis. Analysis of slope limiters on irregular grids. Technical Report NAS-05-007, 2005.

[20] P.-H. Maire. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured mesh. J. Comput. Phys., 228(7):2391–2425, 2009.

[21] M.Alkämper, A.Dedner, R.Klöfkorn, and M.Nolte. The DUNE-ALUGrid module. Arch. Numer. Software, 4(1):1–28, 2016.

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

[23] X. Nogueira, L. Ramírez, S. Clain, and R. Loubère. High accurate SPH method with multidimensional optimal order detection. Computer Methods in Applied Mechanics and Engineering, 310:134–155, 2016.

[24] C. Olliver-Gooch and M. Van Altena. A high-order–accurate unstructured mesh finite–volume scheme for the advection–diffusion equation. J. Comput. Phys., 181:729 – 752, 2002.

[25] S. Del Pino. Metric-based mesh adaptation for 2D Lagrangian compressible flows. J. Comput. Phys., 230(5):1793–1821, 2011.

[26] G. Puppo. Numerical entropy production for central schemes. SIAM J. Sci. Comput., 25(4):1382–1415 (electronic), 2003/04.

[27] G. Puppo and M. Semplice. Numerical entropy and adaptivity for finite volume schemes. Commun. Comput. Phys., 10(5):1132–1160, 2011.

[28] M. Semplice and A. Coco. `dune-fv` software, 2014. `http://www.dipmatematica.unito.it/do/docenti.pl/Alias?matteo.semplice#ricerca`.

[29] 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(2):692–724, 2015.

[30] 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, November 1997.

[31] S.K.Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sbornik, 47:357–393, 1959.

[32] H.-Z. Tang and T. Tang. Adaptive mesh methods for one- and two-dimensional hyperbolic conservation laws. SIAM J. Numer. Anal., 41:487–515, 2003.

[33] E.F. Toro. Riemann Solvers and Numerical Methods for Fluid Dynamics. Springer, second edition, 1999.

[34] B. van Leer. Towards the ultimate conservative difference scheme II: Monotonicity and conservation combined in a second order scheme. J. Comput. Phys., 14:361–370, 1974.

[35] B. van Leer. Towards the ultimate conservative difference scheme V: A second order sequel to Godunov's method. J. Comput. Phys., 32:101–136, 1979.

[36] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. J. Comput. Phys., 118(1):120 – 130, 1995.

[37] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys., 54:115–173, 1984.

[38] O. Zanotti, M. Dumbser, R. Loubère, and S.Diot. A posteriori subcell limiting for Discontinuous Galerkin finite element method for hyperbolic system of conservation laws. J. Comput. Phys., 278:47–75, 2014.