HAL
open science

High-Order Conservative Remapping with a posteriori MOOD stabilization on polygonal meshes

Ghislain Blanchard, Raphael Loubere

To cite this version:

Ghislain Blanchard, Raphael Loubere. High-Order Conservative Remapping with a posteriori MOOD stabilization on polygonal meshes. 2015. hal-01207156

HAL Id: hal-01207156

https://hal.science/hal-01207156v1

Preprint submitted on 1 Oct 2015

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

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

High-Order Conservative Remapping with *a posteriori* MOOD stabilization on polygonal meshes

Ghislain Blanchard[a], Raphaël Loubère[a]

[a] *CNRS and Institut de Mathématiques de Toulouse (IMT) Université Paul-Sabatier, Toulouse, France*

Abstract

In this article we present a 2D conservative remapping method which relies on exact polygonal mesh intersection, high accurate polynomial reconstruction (up to degree 5) and *a posteriori* stabilization based on MOOD paradigm [19, 29, 30, 70]. This paradigm does not compute any sort of *a priori* limiter for the polynomial reconstructions. Instead it rather observes if the candidate solution after remapping does not fulfill user-given criteria, and, in this case, locally to some so-called problematic cells, the method decrements the polynomial degree for the reconstructions. These problematic cells are then recomputed. Next they are again tested against the validity criteria and, possibly, some decrementation is applied. Such iterative procedure always ends either with a valid remapped solution obtained from polynomial degrees greater than 0, or, in the worst case scenario, with a cell remapped without polynomial reconstruction (formally with a first order accurate remap scheme). Numerical results assess the behavior of such remapping method on pure advection equations and on some emulation of system of equations.

Key words: Remapping, polynomial reconstruction, a posteriori stabilization, high accuracy, ALE, MOOD paradigm, mesh intersection, hyperbolic conservation laws

1. Introduction

In this paper we focus on the development of a new class of genuine high accurate remapping methods. In all generality a remapping method is devoted to to *accurately* transfer discrete data defined on one generic "Original/old" mesh $\mathcal{O}$ onto another possibly uncorrelated "Target/new" mesh $\mathcal{T}$.

A remap scheme may be used to initialize a Finite Volume (FV) simulation code. For instance if initial data are only known from a Computer Aided Design (CAD) technology, or from exact functional, some remapping method can provide the piece-wise constant data per cell needed to start a FV simulation. To be more specific, in a Volume-Of-Fluid (VOF) context one may need to initialize volume fractions in mixed cells by intersecting a given shape and a computational mesh. Usually a large sampling of random points is picked and each point is determined to be inside or outside the shape leading to approximate volume fractions. A well designed remapper could overcome this approach by exactly intersecting the shape and the mesh.

However an efficient (accurate and cheap) remapping method could be extremely appreciated in the context of indirect ALE scheme (Arbitrary-Lagrangian-Eulerian) solving a system of non linear PDEs [51, 12, 86]. In this introduction we will consider that the hydrodynamics system of PDEs is solved. For such an indirect ALE scheme a Lagrangian scheme evolves the physical variable on a moving mesh during the time step, a rezone strategy computes a new mesh with a better geometrical quality, and a conservative remap method

Email addresses: `ghislain.blanchard@math.univ-toulouse.fr` (Ghislain Blanchard), `raphael.loubere@math.univ-toulouse.fr` (Raphaël Loubère)

transfers the data from the Lagrangian mesh onto the rezoned one. The purpose of the remap scheme is to provide on the rezoned mesh data which represent accurately the state variables hold on the Lagrangian mesh. Under the term "accurately" are gathered several notions: accuracy (in the sense of a decreasing error in a Taylor expansion), monotonicity (no spurious numerical oscillation), physical admissibility (no nonphysical state), etc. In fact there is no clear agreement on what should be a "good remapped solution" when a system of non-linear PDEs is solved. Classically conservative (primitive or sometimes characteristics) variables are piece-wise linearly reconstructed onto the Lagrangian mesh. These reconstructions are supplemented with a slope limiter based on a Discrete Maximum Principle (DMP) of the reconstructed variable. Such piece-wise linear limited representations are further integrated during the remap stage to obtain the remapped data. Depending on the Lagrangian engine used, conservation of momentum and total energy are sometimes not ensured. For instance some staggered Lagrangian non compatible schemes are not conservative *per se* but, instead, they praise positivity preservation and entropy dissipation. Conversely for some cell-centered [27, 84, 85] or staggered compatible [16, 18, 17, 11] Lagrangian schemes, mass, momentum and total energy are reconstructed and limited, ensuring their conservation by construction. However the positivity of remapped specific internal energy is rarely ensured by construction and may require more advanced technologies. Already we can see that the conservation laws along with the admissibility of the remap state variables are important properties that a Lagrangian scheme may try to ensure. So must the remapper.

For any general system of PDEs, the overall accuracy of a remap scheme is limited by three types of cumulative errors.

Representation error. This corresponds to the difference between the exact profile of the current variable and its polynomial/functional reconstruction that the developer has chosen to employ: $\mathbb{P}_0$ or $\mathbb{P}_1$ reconstructions with or without slope limiter, etc. Limiters may or may not magnify such an error.

Geometrical error. This error is due to the inaccuracy of the method used to intersect the two meshes $\mathcal{O}$ and $\mathcal{T}$. If the exact geometrical intersection between a new cell of $\mathcal{T}$ and the old mesh $\mathcal{O}$ is computed then this error is zero by definition. (Remark that in 1D this error is by construction null.) If the intersection is approximated (swept region remapping [38, 37, 64, 62]) then, at minima, a second order geometrical error is produced. For such approximation methods, like the swept region method, the volume of the new cell is well computed, only some pieces are associated to an incorrect old cell data representation.

Integration error. If the representation of the underlying functional is complex then we may have to resort to numerical integration formula when evaluating their value on the intersection surfaces between the old and new meshes. This also produces an error which can be called integration error. Most of the time numerical integration formula are chosen accurate enough to virtually reduce this error to zero.

In this work we focus on a remapping method for which exact mesh intersection is adopted and virtually exact quadrature formula are employed. We also adopt conservative polynomial reconstructions. As a consequence the representation error only remains of the order of the polynomial reconstruction performed. We will reconstruct polynomials of degree up to 5 which leads to nominal 6th order accurate remapping schemes. Obviously the main concern is the preservation of physical properties (monotonicity, positivity, etc.) which depend on the system of PDEs solved and the essentially non-oscillatory behavior of the remapped solution. We do not adopt ENO or WENO type of polynomial reconstructions [49, 50, 69, 57, 92, 20, 45, 41] but rather rely on a recent *a posteriori* stabilization called MOOD for Multidimensional Optimal Order Detection method [19, 29, 30, 70]. The main novelty of this approach is to setup a list of admissible properties for a remapped solution, and to verify that the piece-wise constant remapped solution fulfills these properties. If not, locally to a bad cell, a more dissipative remap scheme is employed to recompute the remapped data for instance increasing the representation error by reducing the order of

accuracy with which the polynomial reconstruction is done. Then the problematic cell is further tested, and, possibly need to be recomputed after some more polynomial degree decrementing.

With such an approach the need for a limiter vanishes. Indeed the role of the limiter is twofold: (i) detect which cells need more dissipation, and (ii) compute how much dissipation is actually employed. Both are difficult tasks when only *a priori* data are available, that is when only data on the old mesh are usable. In an *a posteriori* approach, the detection (i) consists of a check against the validity criteria and the amount of dissipation (ii) is directly driven by the successive polynomial degree decrementing employed. In our case we mostly adopt the following decrementing cascade: $\mathbb{P}_5 \rightarrow \mathbb{P}_3 \rightarrow \mathbb{P}_1 \rightarrow \mathbb{P}_0$ involving only four orders of accuracy 6th, 4th, 2nd and 1st. The last scheme of the list must ensure that the detection criteria are valid. This so-called parachute scheme is important to assure the validity of the numerical solution in extreme situations.

The paper is organized as follows. A first section presents the general philosophy and properties that a high order remapping scheme must obey. Next we design a high accurate remap scheme based on polynomial reconstructions in 2D on general polygonal meshes. In the same section we show some numerical evidences that some limiting is needed and present classical second order accurate remapping schemes employing a slope limiter. An *a posteriori* MOOD like approach is designed and tested for pure remapping situations. Conclusion and perspectives are finally drawn.

2. Philosophy of high order remap scheme

In order to ease the description let us introduce generic notation. We call "old mesh" and denote by $\mathcal{M}$ the mesh onto which the finite volume vector of physical variables U is defined. In other words if a cell of index i of $\mathcal{M}$ is denoted by Ω_i then the finite volume piece-wise constant cell centered value U_i is defined by

$$U_i = \frac{1}{V_i} \int_{\Omega_i} U \, dx \, dy, \tag{1}$$

where V_i is the volume of cell Ω_i, *i.e* $V_i = \displaystyle\int_{\Omega_i} 1 dx \, dy$. Such entities are volume averaged conserved quantities. For system of PDEs conserved mass averaged quantities are defined by

$$U_i = \frac{1}{m_i} \int_{\Omega_i} \rho U \, dx \, dy, \tag{2}$$

where m_i is the mass of cell Ω_i, *i.e* $m_i = \displaystyle\int_{\Omega_i} \rho \, dx \, dy$ and ρ is the density.

The remapped solution is called $\widetilde{U}$ and is defined on the "new mesh" denoted $\widetilde{\mathcal{M}}$. Usually $\widetilde{\mathcal{M}}$ is a slight modification of $\mathcal{M}$ sharing the same connectivity. However in all generality we consider that the connectivity of $\widetilde{\mathcal{M}}$ may have no relation whatsoever with $\mathcal{M}$, that is the number of cell and neighborhood have possibly changed. However the old mesh must pave a larger computational domain than the new one to avoid undefined regions. In this work we only consider the same computational domain exactly paved by the old and new meshes.

2.1. Properties of the remapped solution

The unique goal of a remapping scheme is to properly remap variable U from mesh $\mathcal{M}$ onto mesh $\widetilde{\mathcal{M}}$. The formula "properly" is deliberately vague as there is no universally accepted definition of what kind of properties the obtained remapped solution should fulfill. Nevertheless we can provide a non-exhaustive list of properties for a remapped solution to be considered as "good" in the case of a system of conservation laws.

- *Physical admissibility.* If U is physically admissible, $\widetilde{U}$ must enjoy the same property. The definition of physical admissibility depends on the system of equations solved. It simply means that the discrete solution should be valid for the model of PDEs considered. As instance for Euler equations with simple equation of state one must require that the density and the specific internal energy are both positive. Other systems of PDEs may reasonably require that all computed velocities do not exceed the speed of light (relativistic magneto-hydrodynamic models) or that concentrations of species remain between zero and one (multimaterial fluid models).

- *Conservation.* If $Q = \sum_i U_i$ represents some conserved quantity such as mass, momentum or total energy then $\widetilde{Q} = \sum_i \widetilde{U}_i$ must be equal to Q.

- *Accuracy.* The remapping technology must be able to exactly retrieve a scalar field: constant, linear, parabolic, or polynomials of higher degree. In [63] the authors have proposed a linearity preserving scheme, meaning that the scheme is exact for piece-wise linear field. A counter-example is the so-called "donor cell" remap scheme considering piece-wise constant data which is exact only for constant fields.

- *Bound-preservation.* Because the remapped solution is obtained from a kind of conservative interpolation, it should not exceed local numerical bounds defined using the old discrete solution. This property also known as the Discrete Maximum Principle (DMP) states that, given a neighborhood $\mathcal{N}_i$ of cell i, the solution should fulfill

$$\min_{j\in\mathcal{N}_i}(U_j) \leq \widetilde{U}_i \leq \max_{j\in\mathcal{N}_i}(U_j). \tag{3}$$

  However it has been shown (see as instance in MOOD method [29, 19]) that the strict DMP property forces the scheme to remain at most second order accurate in L^∞ norm. An appropriate relaxation of the DMP must be designed to reach effective high-order of accuracy (such as the U2 detection process of [29, 19], or simpler relaxation [42]). This is due to the fact that the genuine "continuous" of Maximum Principle states that

$$\min_{x\in\mathcal{N}_i} U^{ex}(x) \leq \widetilde{U}_i \leq \max_{x\in\mathcal{N}_i} U^{ex}(x). \tag{4}$$

- *Efficiency.* In 2D, and even more pronounced in 3D, the remapping scheme may be expensive. Therefore some care must be taken to design an effective remap scheme with a clear gain in accuracy if extra computation time is required.

- *Reversibility.* In [74] the authors have proposed that a remap scheme be reversible. The two successive remaps from $\mathcal{M}$ to $\widetilde{\mathcal{M}}$ then back from $\widetilde{\mathcal{M}}$ to $\mathcal{M}$ should exactly lead back to U.

The previous list is not ordered by importance, only the physical admissibility property is essential to provide admissible data. We would like to emphasize that the bound preservation is only a numerical concern whereas the physical admissibility property checks the true validity of the discrete solution with respect to the system of equations; both should not be confused. Reversibility is often by-passed as this property is difficult to fulfill. Conservation property is most of the time presented in its global formulation as above, however conservation must be ensured in the vicinity of each cell. In other words the remapping scheme should not destroy the principle of causality; if a cell value has increased, this is only due to its interaction with close neighborhood only[1].

2.2. *Review of a generic remap scheme*

A generic remap scheme starting from piece-wise constant data is a three step process:

[1] We have to understand that the details of the previous list of property must be provided by the user (or developer) to determine what is a good enough solution. Consequently this list of properties may slightly change from user to user or may evolve if the system of equation is modified. Ideally the remapping scheme should be able to adapt to such modifications.

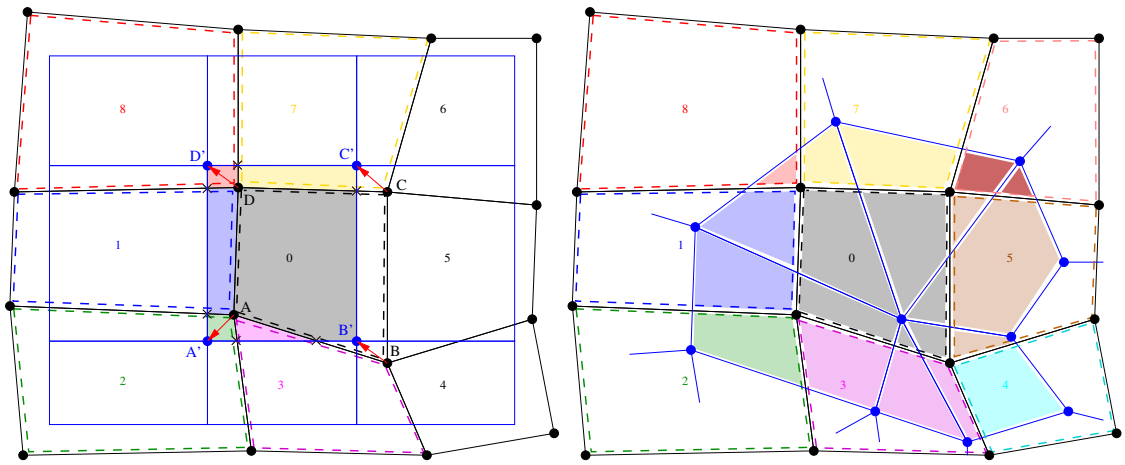


Figure 1: Exact mesh intersection (overlays) for subsequent remapping starting from an old black mesh onto a new blue mesh — Left: Exact intersection for meshes sharing the same connectivity. Cell 0 is surrounded by cells $1, \ldots, 8$. New blue cell 0 is paved with small pieces of old neighbor cells $0, 1, 2, 3, 7$ and 8. Red polygon corresponds to the intersection with corner black cell 8 (red number), blue polygon to the intersection with left black cell 1 (blue number), gray polygon to the surface of old cell 0 remaining in new cell 0. Right: Exact intersection for meshes having different connectivity. Only a portion of a generic blue mesh is shown. The intersection of one blue cell and the black mesh may lead to an arbitrary number of intersection polygons with an arbitrary number of blue cells.

- First, one defines a *representation* of the old solution U on the old mesh $\mathcal{M}$; The simplest choice is to maintain the piece-wise constant representation.
- Second, a *geometrical* intersection between a new cell and the old mesh is computed, it consists of a set of closed surfaces that pave the new cell without overlapping and gap, see Fig. 1 left panel for an example with quadrangles and shared connectivity, and, the right panel for the general case.
- Third, the remapped variable in the new cell is computed as the sum of *integrated* representations (from first step) onto the intersection surfaces (colored polygons in Fig. 1).

As already seen these three steps generate three errors, two of which can be made as small as desired if an exact mesh intersection and high accurate numerical integration are employed.
The 'representation error' remains present and at least two main flaws for such a straightforward remap method can be pointed:

1. the method is exact only for constant functions leading, in some sense, to a first order accurate method which may be considered as too much dissipative,
2. there is no check on the physical admissibility of the discrete remapped solution.

The first issue has led several authors to propose to reconstruct the underlying function as a piece-wise linear function [87, 63] like in a MUSCL finite volume scheme. If so, the remapping method becomes exact for linear function and, as such, is considered as a second-order accurate method. However the generation of non-physical remapped quantities enforces the utilization of slope limiter in the reconstruction phase. Such limiters may not always be trivial to define as instance for vector fields, tensors, etc. Classically the limiting is independently applied to each component leading to a frame dependent procedure[2]. Moreover most limiters are funded on a strict application of the discrete maximum principle. Consequently the remap scheme accuracy in L^∞ norm can never exceed 2. Using more advanced reconstructions following WENO

[2]This is one of the reasons why frame invariant limiters [73, 80] have been recently designed.

philosophy is also possible but it would require the reconstruction of several polynomials per cells and their mixing through non-linear weights, leading to a somewhat more expensive method.
The second flaw is often by-passed by stating that a lot of dissipation brought by the first order of accuracy will help. Rarely we can prove that a first order remap scheme can assure that the physical admissibility properties are fulfilled. This situation becomes even more complex when some variables are reconstructed, limited, and, further non-linearly mixed together in an equation of state for instance. For a scalar advection equation, a decent limiter often ensures that the remapped solution is admissible. Conversely for the hydrodynamic system of equations, the situation is mathematically more complex. Indeed ensuring that the remapped data (density, momentum and total energy) will generate a positive pressure through the equation of state is not trivial. In some sense remapping should not be though as a black box of pure geometrical or interpolation operators acting independently of the physical meaning of the remapped variables. Rather the remapping has to take into account the properties of the variables and their possible relationships. Note that slope limiters or WENO type of limiting have the same drawbacks when applied straightforwardly without any fine tuning.

2.3. Review of an a posteriori stabilization MOOD loop

To fulfill the list of properties from the previous section (physical admissibility, conservation, accuracy, bound-preservation, efficiency, reversibility) one must not only high accurately reconstruct the variables to remap, but, also, make sure that the conserved variables are conserved, that admissible variables remain so and that a non-oscillating behavior is observed.
Remark that the remapped variables may not be the conserved variables or the physical variables tested for admissibility. For instance, in an hydrodynamics context with a given equation of state (EOS), the density ρ, velocity $\boldsymbol{U}$ and specific internal energy ε are independent variables to be remapped. However the conserved quantities are the mass $m = \rho V$ (integral of density over a volume), the momentum $\boldsymbol{Q} = \rho \boldsymbol{U}$ (integral of $\rho \boldsymbol{U}$), and the total energy $E = m\varepsilon + \frac{1}{2} m \boldsymbol{U}^2$ (integral of total specific energy $\varepsilon + \frac{1}{2}\boldsymbol{U}^2$). Remapped and conserved variables do not coincide. Neither do they coincide with the constraints for physical admissibility for this system of PDEs: positivity of density, internal energy and possibly pressure. Note that one also could consider the second principle of thermodynamics, $\rho \frac{\partial}{\partial t} S \geq 0$, where the entropy S is related to the other thermodynamic variables by the Gibbs relation $TdS = d\varepsilon - pd(1/\rho)$ where T is the temperature.
In table 1 three sets of variables to remap are proposed. These lead to either the loss of momentum and/or total energy conservation (**Remap #0** and **Remap #1**) or the loss of specific internal energy positivity preservation (**Remap #2**).

In this work this multi-task remapping is based on high accurate reconstructions and remapping of conserved variables like for **Remap #2**. The admissibility properties are assured with a so-called *a posteriori* stabilization MOOD loop. In [19, 29, 30, 70, 42, 14] the authors have shown that in the context of finite volume, Discontinuous Galerkin and even direct Arbitrary-Lagrangian-Eulerian schemes for advection equation or hydrodynamics, an *a posteriori* limiting/stabilization approach denoted MOOD was revealed fruitful. In this work we will apply some of the tools from [19, 29, 30] to the remapping context.

2.3.1. Accuracy: unique unlimited high accurate polynomial reconstruction

High accuracy is obtained by reconstructing polynomials (up to 6th order of accuracy) by considering a unique central stencil with respect to the current cell. A least square procedure is employed to determine the best fitted conservative polynomial of degree $d > 0$. Conversely to WENO type of approach where several polynomials are reconstructed, in this work, a unique central stencil with enough neighbor cells is considered for the reconstruction. Moreover we choose to reconstruct conservative variables so that the question of conservation does not hold anymore (situation of **Remap #2** in table 1).

2.3.2. Monotonicity and stabilization: a posteriori decrementing

The remaining question is the construction of a dissipative mechanism to assure a non oscillatory behavior, to kill Gibbs phenomenon and preserve the positivity as well as other important physical properties

	Remapped variables	Conserved variables	Admissibility
Remap #0	mass $m = \rho V$ velocity $\boldsymbol{U}$ intern. energy ε	mass $m = \rho V$ ~~momentum $\boldsymbol{Q} = m\boldsymbol{U}$~~ ~~total energy E~~	density $\rho > 0$ spec. intern. energy $\varepsilon > 0$ pressure $p > 0$
Remap #1	mass $m = \rho V$ momentum $\boldsymbol{Q} = m\boldsymbol{U}$ total intern. energy $\mathcal{E} = m\varepsilon$	mass $m = \rho V$ momentum $\boldsymbol{Q} = m\boldsymbol{U}$ ~~total energy E~~	density $\rho > 0$ spec. intern. energy $\varepsilon > 0$ pressure $p > 0$
Remap #2	mass m momentum $\boldsymbol{Q} = m\boldsymbol{U}$ total energy E	mass m momentum $\boldsymbol{Q}$ total energy E	density $\rho > 0$ ~~spec. intern. energy $\varepsilon > 0$~~ ~~pressure $p > 0$~~

Table 1: Example of remap choices for the hydrodynamics system of PDEs. The hydrodynamics variables are deduced from each others: the mass is an integrated density: $m = \rho V$, momentum has the dimension of mass×velocity $\boldsymbol{Q} = m\boldsymbol{U} = \rho V\boldsymbol{U}$, total energy is $E = \mathcal{E} + \frac{1}{2}\boldsymbol{Q}^2/m$ where the total specific internal energy is $\mathcal{E} = m\varepsilon$ and the specific internal energy is denoted by ε which is usually employed along with density to define the gas pressure p through an equation of state (EOS) $p = EOS(\rho, \varepsilon)$. Physical admissibility demands that density, specific internal energy and pressure remain positive. Depending on the EOS the pressure positivity may be implied by density and energy positivity.

of the system of PDEs from which the variables are considered. Any classical types of limiting (slope limiting [107], hierarchical limiting [65], (W)ENO [56, 45, 39], etc.) are performed *a priori* by a clever analyze of the available data at t^n. This implies that the "worst case scenario" must always be considered as plausible, and, as a consequence, the "precautionary principle" applies. In other words because some investigation has found a plausible risk of instability development, *a priori* limiting strategy strikes more often and harder than necessary. We believe that such limiting can be relaxed only if further information emerges that provides evidence that no harm will result when using unlimited reconstructions. One way to gain more information is to check *a posteriori*, that is to say after the remapping has been done, if the candidate remapped solution fulfills some stability criteria[3]
These are the extremely simple and crude principles of the MOOD method: first compute a candidate unlimited solution, then detect if this solution locally fails to fulfill some stability criteria (in problematic cells) and further recompute only those problematic cells with a more dissipative scheme, for instance using lower accurate reconstructions to get a new candidate solution. The new candidate solution is then tested again for eligibility, and, if some problematic cells are still detected then an even more dissipative scheme (for instance by discarding any reconstruction) is triggered. This procedure iterates up to the acceptability of the numerical solution. Note that the detection procedure is always performed on a candidate remapped solution, hence the terminology *a posteriori*. Moreover a bad candidate solution in a given cell is never used, rather it is always discarded and one start again from valid t^n data. The *a posteriori* MOOD paradigm follows this fundamentally different way.

In the next section we detail our MOOD remapper by describing what are the detection criteria to test a candidate solution, which 'cascade' of schemes is adopted (from the most accurate scheme to the most dissipative one), and which scheme is the parachute/bullet-proof scheme, that is the last scheme of the cascade.

[3]Testing *a posteriori* the validity of a solution is not new and can be found in the context of remapping methods (decreasing of polynomial order in [52], repair methods [98, 88, 75, 26], or FCT [13, 66]) and presumably in many other areas.

3. High accurate MOOD remapping strategy with *a posteriori* stabilization

For our remapping scheme several developer choices must be made. A maximal polynomial degree $d^{\max}$ is first set. A $d^{\max}+1$th order accurate remap scheme then is expected in an ideal situation where the solution is smooth enough: this is the *Ultimate Scheme* one would like to use as much as possible, also denoted as the $\mathbb{P}_{\max}$ scheme. Next the developer must fulfill a list of properties that the remapped numerical solution in each cell must fulfill. If so, the solution in this cell is said to be *valid* otherwise the cell is called *problematic*. These properties are called the *Detection Criteria*. One must also design a 'low ranked' scheme, the solution of which is considered as always acceptable: it is called the *Parachute Scheme* (or bullet-proof scheme). In the remapping context we know that the first order remapping scheme (using constant data per cell, i.e donor cell remap scheme) is the most diffusive and stable scheme that can be built. If all contributions of the remapped solution are computed with such a low order scheme the obtained solution is considered as valid and stable, though not tremendously accurate. This parachute scheme is also referred to as the $\mathbb{P}_0$ scheme.
Finally one must determine a ranked list of schemes to be tested called the *Cascade*. This cascade is visited while decrementing occurs when problematic cells are detected starting from the ultimate scheme and dropping to the parachute scheme. As instance we can order the unlimited schemes according to the polynomial degree of the reconstructions: $\mathbb{P}_{\max} \rightarrow \mathbb{P}_{\max-1} \rightarrow \ldots \rightarrow \mathbb{P}_2 \rightarrow \mathbb{P}_1 \rightarrow \mathbb{P}_0$, possibly skipping some of them, see the last subsection for a discussion on this point.

In the following we present the main steps of our high accurate remapping scheme: polynomial reconstruction, geometrical mesh intersection, integration and *a posteriori* detection which is, *de facto*, the heart of the *a posteriori* stabilization technique inspired from the MOOD paradigm [19, 29, 30, 70]. This section is concluded by a summary and discussion on technical implementation and developer details.

3.1. Polynomial reconstruction

In this section we briefly recall the least-square polynomial reconstruction technique employed, which has already been described thoroughly in [19, 29, 30, 70] for instance. Let us describe a polynomial reconstruction for a generic variable $\mathcal{U}$ defined by cell centered mean values $\mathcal{U}_i$ in any cell Ω_i. Given a cell Ω_i, a degree $d > 0$, a large enough stencil $\mathcal{S}_i^d$ made of $N_i^d > 0$ cells in the vicinity of Ω_i, the problem of polynomial reconstruction of degree d in Ω_i consists in finding the coefficients $\mathcal{R}_i^{\alpha}$ of

$$\mathcal{U}_i(\boldsymbol{X}) = \mathcal{U}_i \; + \sum_{1\leq|\boldsymbol{\alpha}|\leq d} \mathcal{R}_i^{\alpha} \left((\boldsymbol{X}-\boldsymbol{X}_i)^{\alpha} - \frac{1}{|\Omega_i|}\int_{\Omega_i} (\boldsymbol{X}-\boldsymbol{X}_i)^{\alpha}\, d\boldsymbol{X} \right), \tag{5}$$

such that, in a least-squares sense, the following functional is minimal

$$E = \sum_{j\in\mathcal{S}_i^d} \omega_j \left(\frac{1}{|\Omega_j|}\int_{\Omega_j} \mathcal{U}_i(\boldsymbol{X})\, d\boldsymbol{X} \; - \; \mathcal{U}_j \right)^2. \tag{6}$$

In the previous equations $\boldsymbol{X}_i$ is the centroid of Ω_i, $\boldsymbol{X}$ a generic point in Ω_i and $\mathcal{R}_i^{\alpha}$ are the unknowns polynomial coefficients where $\boldsymbol{\alpha}=(\alpha_x,\alpha_y)\in\mathbb{N}^2$ is a multi-index with $|\boldsymbol{\alpha}| = \alpha_x+\alpha_y$. Last, the terms ω_j are positive weights used to provide better condition number of the associated matrix to be inverted as it dependents on a spatial characteristics length [43]. By construction, the mean value on Ω_i of the polynomial is equal to ρ_i since the integral over Ω_i of the term between parenthesis in (5) simply vanishes.
This technique is rather classical and is able to exactly reconstruct any polynomial of degree $d' \leq d$ if a large enough stencil $\mathcal{S}_i^d$ is provided. In 2D a minimum of $(d+1)(d+2)/2-1$ neighbor values are needed, and, for the sake of robustness we always pick more neighbors $D(d) = 1.5\lceil(d+1)(d+2)/2-1\rceil$ where $\lceil\rceil$ means the closest strictly superior integer. as a consequence stencil $\mathcal{S}_i^d$ is incrementally constructed, see figure 2, as

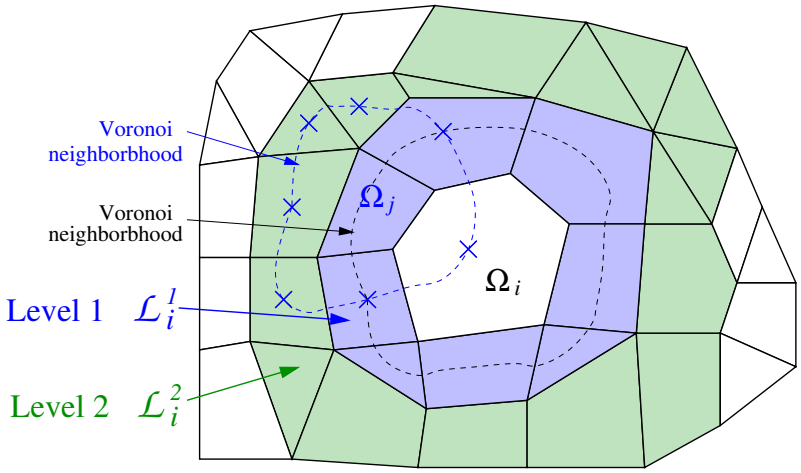


Figure 2: Sketch of the first level of neighborhood $\mathcal{L}_i^1$ (blue cells) of central white cell Ω_i and second level $\mathcal{L}_i^2$ (green cells). Each level is composed of Voronoi neighborhood of all cell belonging to the previous considered level - Level $\mathcal{L}_i^2$ is composed by the Voronoi neighborhood of all cells Ω_j belonging to $\mathcal{L}_i^1$. Stencil $\mathcal{S}_i^d$ is constructed by the union of one or more levels in such a way that $\text{card}(\mathcal{S}_i^d) > 1.5\lceil(d+1)(d+2)/2-1)\rceil$, where d is the degree of the polynomial degree to be reconstructed.

0. $\mathcal{S}_i^d = \mathcal{L}_i^1$ where $\mathcal{L}_i^1$ is the Voronoi neighborhood of Ω_i, that is to say $\mathcal{L}_i^1 = \{j,\ \text{s.t}\ \Omega_j \bigcap \Omega_i \neq \emptyset\}$. If $\text{card}(\mathcal{S}_i^d) \geq D(d)$ exit, else set $k = 1$
1. $\mathcal{S}_i^d = \mathcal{L}_i^{k+1}$ where $\mathcal{L}_i^{k+1}$ is the reunion of the Voronoi neighborhood of any $\Omega_j \in \mathcal{L}_i^k$.
2. If $\text{card}(\mathcal{S}_i^d) \geq D(d)$ exit, else $k = k+1$ and go back to 1.

Usually $\text{card}(\mathcal{S}_i^d) > D(d)$ and we could possibly reduce the number of neighbor cells from the last iterate, then reducing the size of the linear system to be inverted, however to the price of a non-symmetric final stencil.

In our remapping technique we only consider one stencil per degree d. Contrarily to WENO polynomial reconstructions we do not have to choose several stencils, neither do we have to non-linearly mix several reconstructions altogether. Moreover to fulfill the conservation property, the reconstructions are always performed for the conservative variables. Obviously if conservation is not to be maintained, primitive or characteristics variables could be reconstructed instead. Finally when $\mathcal{U}$ is a set of conservative variables as $\mathcal{U} = (\rho, \rho\boldsymbol{U}, \rho E)^t$ then the reconstruction step is performed for each conservative variable. Of course the extra cost is rather low as the variables all share the same matrix to be inverted.

3.2. Geometrical mesh intersection

The problem of mesh intersection can be mathematically stated in a relative simple fashion. Let us denote by $\mathcal{M}$ the set of cell index in the old mesh and $\widetilde{\mathcal{M}}$ in the new mesh and supposed that both meshes pave the same domain. Next we call intersection polygons the non empty figure ω_{kj} such that

$$\forall k \in \widetilde{\mathcal{M}}, \qquad \forall j \in \mathcal{M}, \qquad \omega_{kj} = \widetilde{\Omega}_k \bigcap \Omega_j \neq \emptyset. \tag{7}$$

Any cell $\widetilde{\Omega}_k$ in the new mesh is therefore paved by a set of P_k intersection polygons as, see the colored polygons in figure 1,

$$\widetilde{\Omega}_k = \bigcup_{j=1}^{P_k} \omega_{kj}, \tag{8}$$

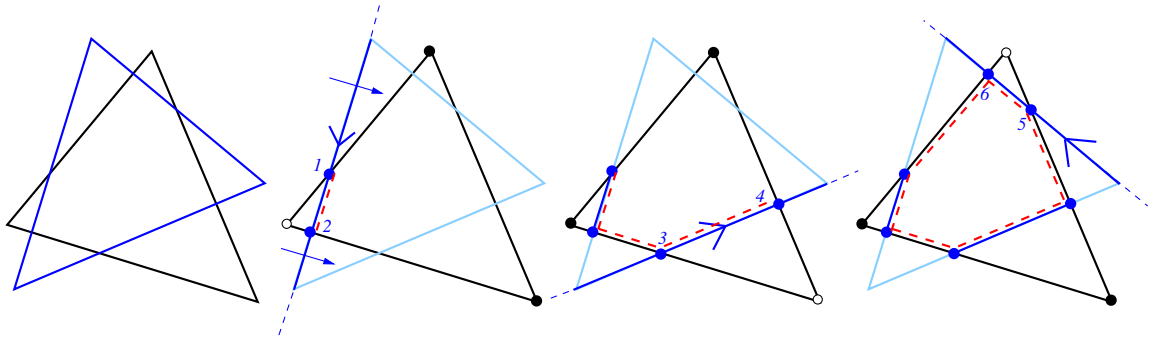


Figure 3: Sketch of a triangle/triangle intersection algorithm. The first edge of the blue triangle is visited and a decision is taken to state if all vertexes lay on the same half-plane defined by this edge (black vertexes are on one side whereas white ones on the other). If not then intersection between the blue edge and the black triangle occurs. Here two intersection points are computed (1 and 2). Each edge is visited successively to finally construct the red polygons which is nothing but the intersection polygon between the black and blue triangles.

From a computational point of view writing a robust polygonal intersection code is demanding. We have chosen to employ already validated algorithms which can be found in [102, 28]. Because those algorithms nicely extend to 3D polyhedral meshes, we have implemented a degenerated 2D version adapted to our context. Briefly the algorithms for intersecting two convex polygon rely on successively clipping one against the edges of the other. Following [28] we realize that this algorithm is an implementation in the remapping context of some ideas presented abstractly in [102] where the author suggests using the planar graph representations of convex polygons to ensure topological consistency of the resulting intersection polygons. In our implementation any star-shaped polygon is first split into simplexes (triangles). Then triangle-to-triangle intersection is performed by clipping the edges of one triangle against the other, see figure 3 for a brief illustration.
Fine tuned "2D oriented algorithms" may be presumably more efficient. However the algorithms proposed in [102, 28] have been designed and developed for general polyhedron in 3D and are shown to be robust to mesh degeneracy. These constitute a viable base to write an encapsulated 2D-3D intersection tool within a common environment.

3.3. Integration

When mesh intersection polygons 8 are available the remapped conservative variables are computed by integrating the reconstructed polynomials 5, as

$$\widetilde{\mathcal{U}}_k = \sum_{j=1}^{P_k} \int_{\omega_{kj}} \mathcal{U}_j(\boldsymbol{X})\, d\boldsymbol{X}. \tag{9}$$

More precisely each intersection polygon ω_{kj} is split into N_T triangles

$$\omega_{kj} = \bigcup_{i=1}^{N_T} T_i, \tag{10}$$

Then a N_g point Gauss-Legendre numerical integration is employed

$$\widetilde{\mathcal{U}}_k = \sum_{j=1}^{P_k} \int_{\omega_{kj}} \mathcal{U}_j(\boldsymbol{X})\, d\boldsymbol{X} = \sum_{j=1}^{P_k} \sum_{i=1}^{N_T} \left(\int_{T_i} \mathcal{U}_j(\boldsymbol{X})\, d\boldsymbol{X} \right) \simeq \sum_{j=1}^{P_k} \sum_{i=1}^{N_T} \left(\sum_{g=1}^{N_g} \mathbf{w}_g \mathcal{U}_j(\boldsymbol{X}_g) \right). \tag{11}$$

N_g is chosen according to the polynomial degree d of the reconstruction $\mathcal{U}_j$ in such a way that (11) is indeed exact. When d_j is relatively large, say about 5 in our case, then the number of integration point increases[4]. Moreover the triangular splitting (10) usually leads to several triangles (on average $\simeq 4$) implying a total number of polynomial evaluations of the order $N_E = P_k \times N_T \times N_g$, which for $d = 5$, $P_k = 4$, $N_g = 7$ yields $N_E = 140$. Therefore the implementation of the integration demands some care to avoid some waste of computational resources.

3.4. *a posteriori detection*

The main principles of the *a posteriori* detection step and subsequent decrementing have been described in the previous section 2.3.2 and we provide the development and implementation details in this section. Let us consider that the hydrodynamics system of equation is solved and, consequently the variables to remap are the conservative variables that is $\mathcal{U}_i = (\rho_i, \boldsymbol{Q}_i, E_i)^t$. The case of a single scalar variable is trivially deduced.

At the end of the remap scheme previously described (unlimited reconstruction of degree $d^{\max}$, exact mesh intersection and virtually exact integration), we have access to a candidate remapped solution $\widetilde{\mathcal{U}}_k$ in each cell $\widetilde{\Omega}_k$. The purpose of the detection is to filter 'good' cells which have been properly remapped with this high accurate remap scheme from 'bad' cells which present some issues (oscillations, non admissibility states, etc.) Following the MOOD paradigm the detection criteria are split into two different sets

Physical Admissibility Detection criteria (PAD). The physical admissibility of the remapped data $\widetilde{\mathcal{U}}_k$ for any cell k is tested here. In the hydrodynamics context, a 'good' cell is such that the positivity of density, specific internal energy and possibly the pressure, are observed

$$(PAD) \qquad C^{PAD}\left(\widetilde{\mathcal{U}}_k\right) = \begin{cases} 1 & \text{if} \quad \widetilde{\rho}_k > 0 \quad \text{and} \quad \widetilde{\varepsilon}_k > 0 \quad \text{and} \quad \widetilde{p}_k > 0 \\ 0 & \text{else} \end{cases} \tag{12}$$

These criteria must be adapted to the system of PDEs resolved.

Numerical Admissibility Detection criteria (NAD). With these criteria we test if newly created extrema on conservative variables are considered as smooth ones which do not need to be dissipated, or, non-smooth extrema which need stabilization. Hence, a new extrema is observed if, at least, one of the components of $\widetilde{\mathcal{U}}_k$ does not fulfill the following discrete maximum principle, that is if

$$\mathcal{D}_k = \begin{cases} 1 & \text{if} \quad \widetilde{\mathcal{U}}_k \leq \min_{j\in\mathcal{N}_k} \mathcal{U}_i, \quad \text{or} \quad \max_{j\in\mathcal{N}_k} \mathcal{U}_i \leq \widetilde{\mathcal{U}}_k \\ 0 & \text{else} \end{cases} . \tag{13}$$

In that case, we apply the so-called U_2 detection criteria which, according to "curvatures" in the vicinity of i, decides it k is to be kept as is or needs to be recomputed with w more dissipative scheme. The U_2 criteria is computed on the neighborhood $\mathcal{N}_k$ in the old mesh for each component of $\mathcal{U}$. First one considers the $\mathbb{P}_2$ reconstructions $\mathcal{U}_i(\boldsymbol{X})$ in cell i belonging to $\mathcal{N}_k$ and further computes the minimal/maximal directional "curvature" as

$$\mathcal{X}_k^{\min} = \min_{i\in\mathcal{N}_k} \left(\partial_{xx}\mathcal{U}_i(\boldsymbol{X})\right), \qquad \mathcal{X}_k^{\max} = \max_{i\in\mathcal{N}_k} \left(\partial_{xx}\mathcal{U}_i(\boldsymbol{X})\right). \tag{14}$$

and, for this direction the Boolean $\mathcal{X}_k$

$$\mathcal{X}_k = \begin{cases} 1 & \text{if} \quad \left[\mathcal{X}_k^{\min}\mathcal{X}_k^{\max} > -\delta \quad \text{and} \quad \left(\max\left(|\mathcal{X}_k^{\max}|, |\mathcal{X}_k^{\min}|\right) < \delta \quad \text{or} \quad \left|\frac{\mathcal{X}_k^{\min}}{\mathcal{X}_k^{\max}}\right| \geq \frac{1}{2}\right)\right] \\ 0 & \text{else} \end{cases} . \tag{15}$$

[4]For degrees $d = 1, 2, 3, 4, 5, 6, 7$ the number of Gauss-Legendre points scales in 2D as $N_g = 1, 3, 4, 6, 7, 12, 13$.

The same computation is also performed in the y direction to get $\mathcal{Y}_k$. Here δ is the local geometrical length, in our case we set it to the maximal local length edge in $\mathcal{N}_k$. The U_2 criteria finally states that an extreme remapped cell $\widetilde{\mathcal{U}}_k$ may be nonetheless valid if $\mathcal{X}_k\,\mathcal{Y}_k = 1$, and is problematic otherwise:

$$(NAD) \qquad C^{NAD}\left(\widetilde{\mathcal{U}}_k\right) = \begin{cases} 1 & \text{if} \quad \mathcal{D}_k = 0 \\ 1 & \text{if} \quad \mathcal{D}_k = 1 \quad \text{and} \quad \mathcal{X}_k\,\mathcal{Y}_k = 1 \\ 0 & \text{if} \quad \mathcal{D}_k = 1 \quad \text{and} \quad \mathcal{X}_k\,\mathcal{Y}_k \neq 1 \end{cases} . \tag{16}$$

A cell is flagged as problematic if either $C^{PAD}\left(\widetilde{\mathcal{U}}_k\right)$ or $C^{NAD}\left(\widetilde{\mathcal{U}}_k\right)$ are equal to 1. In the case of a system of PDEs, all components of $\mathcal{U}$ must be valid for the cell to be accepted.

3.5. Decrementing

When a cell k is problematic then it must be recomputed with a more dissipative scheme which employs lower degree polynomial reconstructions. Therefore for each old cell j being in interaction with new cell k ($j = 1, \dots P_k$), we reconstruct the polynomials $\mathcal{U}_j(\boldsymbol{X})$ choosing a lower degree d_j than previously used. In more details each old cell j carries along with data $\mathcal{U}_j$ the local cell polynomial degree $d_j \geq 0$. The polynomial reconstructions are of order $d_j + 1$. If j has participated in the occurrence of a problematic cell k then its polynomial degree d_j is reduced with the minimal possible value being $d_j = 0$. Then, after decrementing all polynomial degrees d_j of cells interacting with new cell k, we have *de facto* reduced the accuracy with which cell k is remapped. Note that only one cell polynomial degree is considered even in the case of several component of $\mathcal{U}$. This drastically simplifies the associated remap algorithm.

The decrementing step is designed in figure 4) as a decrease of the polynomial degree of 1 at each cycle. This is not mandatory and any other jump from $d_{\max}$ to 0 are possible. In [19, 29, 30] as instance the choice of the authors was $d_{\max} \to 2 \to 0$ and in [70] $d_{\max} \to 1^{\text{LIM}} \to 0$ where 1^{LIM} means $\mathbb{P}_1$ reconstruction with slope limiter (TVD scheme). In fact the only mandatory choices are the maximal degree $d_{\max}$ and the most dissipative donor cell remap scheme ($d = 0$). Ultimately we could choose $d_{\max} \to 0$ for efficiency purpose, at the cost, presumably, of de-gradating some of the results, or, $d_{\max} \to d_{\max} - 1 \to \cdots \to 0$ to get the best possible accuracy at the price of a waste of CPU time. In the numerical experiences gathered in the next section we pick $d_{\max} = 5$ and the cascade $5 \to 3 \to 1 \to 0$.

3.6. Summary

In this section we summarize our high accurate remap algorithm with *a posteriori* MOOD stabilization described in details in the previous sections. To ease the summary we split the remap scheme into main four components and sketch them in figure 4: geometrical Mesh intersection, polynomial Reconstruction of degree d, Integration, Detection of problematic cells. The 'MOOD loop' corresponds to some iterations of the chain (Mesh intersection, Reconstruction, Integration, Detection) only for cells detected as problematic for which lower degrees are used for the subsequent polynomial reconstruction.
The first three components are mandatory to provide a candidate high accurate solution fitting the nominal accuracy of $d_{\max} + 1$ (these are the black boxes in figure 4).
Contrarily the Detection and subsequent decrementing and re-update are parts of the *a posteriori* stabilization steps of MOOD. These are meant to maintain the high accuracy when smooth flows are encountered, or to decrease the accuracy (possibly to 1st order) on discontinuity.
Note that several iterations can be done but this loop always ends either when the validity of the cell values is confirmed by the detection step, or, when the threshold degree $d = 0$ is reached (which corresponds to a robust donor cell remap used for this cell)[5].

[5] A classical second order scheme with slope limiter would have only the first three components supplemented with a 'Limiting' component before Integration.

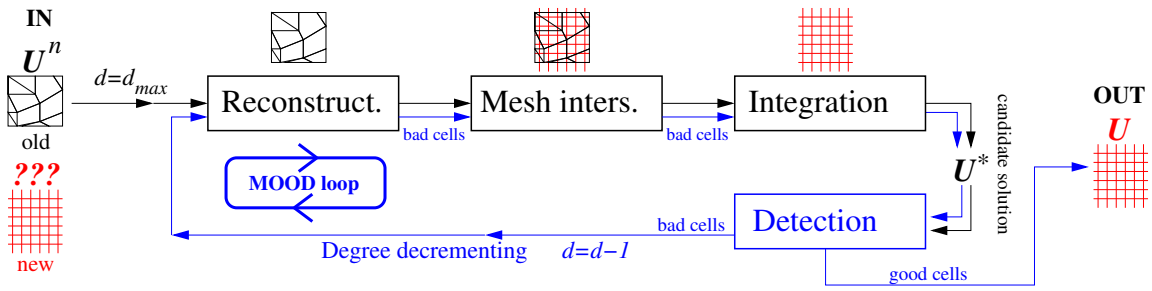


Figure 4: Sketch of the MOOD remapper split into four main components: Geometrical Mesh intersection, polynomial Reconstruction of degree d, Integration, Detection of problematic cells. The 'MOOD loop' corresponds to some iterations of the chain only for problematic cells for which lower degree (after decrementing) polynomial reconstructions have been done.

4. Numerical experiments

The goal of this section is to provide numerical evidences showing that a high accurate remap scheme with *a posteriori* stabilization based on a MOOD like paradigm provides solutions which are accurate, essentially non oscillatory and physically admissible. On purpose we have not coupled this remapper with any Lagrangian scheme building *de facto* an ALE code in order to avoid polluting the remapping effects from pure parasitical Lagrangian scheme behaviors. Consequently our goal is to properly present the behavior of our MOOD remapper, leaving for future work the validation and verification of a MOOD remapper within a 2D and 3D ALE code.

Our methodology of testing is based on a set of pure scalar/vectorial remapping test cases for which the number of remapping steps is either restricted to 5 to emphasize specific instantaneous behaviors, or, to tousandth to enhance their accumulation. To do so we have designed two situations: (i) A five quadrangle mesh situation which remaps from $\mathcal{M}_k$ to $\mathcal{M}_{k+1}$, with $k = 1, \ldots, 5$ and back to $\mathcal{M}_1$, where $\mathcal{M}_k$ and $\mathcal{M}_{k+1}$ are related by small deformations, and, (ii) a polygonal mesh rotation situation which rotates the polygonal mesh of the unit disk in such a way that two successive meshes can not be assumed to be related by small displacements.
Nonetheless initial and final meshes are the same yielding to a simple way to measure the accuracy of the numerical scheme by monitoring the error between initial and remapped solutions, as

$$\epsilon_p = \|\mathcal{U}^0 - \mathcal{U}^T\|_p, \qquad \epsilon_\infty = \max_i |\mathcal{U}_i^0 - \mathcal{U}_i^T|. \tag{17}$$

Moreover numerical rate of convergence can be deduced and general efficiency is also monitored by considering successively refined meshes.

The scalar shapes. Several "shapes" are considered in this work. We design smooth shapes based on sine, Gaussian and polynomials, see table 2. Discontinuous profiles are composed of the three static functions: a Hump, a Cylinder and a Cone

$$\boldsymbol{X}_0 = (0, -1/2), \qquad H(x,y) = \frac{1}{2}\left(1 + \cos\left(\pi \min(r_0(x,y), 1)\right)\right), \tag{18a}$$

$$\boldsymbol{X}_1 = (0, 1/2), \qquad C(x,y) = \begin{cases} 1 & \text{if} \quad r_1(x,y) \leq 0.75 \\ 0 & \text{else} \end{cases} \tag{18b}$$

$$\boldsymbol{X}_3 = (1/2, 0), \qquad K(x,y) = \max\left(1 - \frac{5}{2} R_3 r_3(x,y), 0\right) \tag{18c}$$

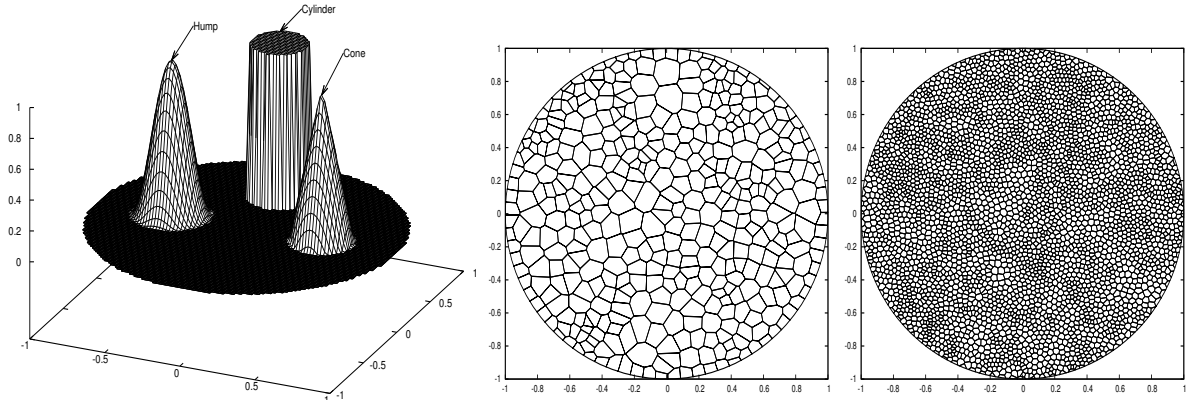


Figure 5: Three shapes used for the rotating remapping test case (left), 1024 polygonal cell mesh (middle) and 4096 polygonal cell mesh (right).

where $r_k(x,y) = \frac{2}{R_k}\sqrt{(x-x_k)^2+(y-y_k)^2}$ and R_k is the radius of shape centers $\boldsymbol{X}_k$, $R_k = \sqrt{x_k^2+y_k^2}$, for $k = 0,1,2,3$. These shapes are displayed in figure 5.

Name	Equation	Domain	Properties
Sine	$1+\sin(4\pi x)+\sin(4\pi y)$	$\boldsymbol{X}\in[0,1]^2$	C^∞
Gaussian	$1+e^{-25(x-1/2)^2}\,e^{-25(y-1/2)^2}$	$\boldsymbol{X}\in[0,1]^2$	C^∞
Polynomial 1	$1+xy$	$\boldsymbol{X}\in[-1,1]^2$	C^∞
Polynomial 2	$3+(2x^2-1)(3y^2-1)$	$\boldsymbol{X}\in[-1,1]^2$	C^∞
Polynomial 3	$3+(4x^3-3x)(4y^4-2y)$	$\boldsymbol{X}\in[-1,1]^2$	C^∞
Cylinder	$\begin{cases} 2 \text{ if } \sqrt{x^2+y^2}\le 0.25 \\ 1 \text{ else}\end{cases}$	$\boldsymbol{X}\in[0,1]^2$	Discontinuous
Three shapes	(18)	$r(\boldsymbol{X})\in[0,1]$	Discontinuous

Table 2: Initial shapes used to test the remapping schemes.

The vectorial shapes. We have emulated a non-smooth solution for the system of hydrodynamics in 2D. In this case four conserved components (mass, momentum and total energy) are simultaneously remapped and they are linked by means of a non-linear relation such as a perfect gas equation of state $p = (\gamma-1)\rho\varepsilon$, where $\varepsilon = E - \frac{1}{2}\rho(u^2+v^2)$, with $\gamma = 7/5$ for instance. The shapes given by the formula in table 3 loosely represent the origin point explosion solution, as instance a Sedov like problem, for which the shock wave is at position r_0 with parameter δ corresponding roughly to the initial temperature before the explosion. In our simulation we consider $r_0 = 0.75$ and $\delta = 10^{-6}$, see figure 6. The profiles are made of discontinuous parts (shocks at radius r_0) and smooth parts (rarefaction's zones). Note also that the density presents a peak before the shock wave leading to a smooth extrema for the pressure variable. A high frequency perturbation of specific internal energy as a function of radius $0.2 \le r \le 0.7$:

$$g(r) = 1.16656\, e^{-100(x-0.5)^2}\, \sin(25\pi r), \tag{19}$$

is super-imposed to this profile. The exact profiles are plotted in figure 6 where the smooth high frequency profile as well as the shock wave are ilustrated. Density and velocity remain the same, only the pressure is

Variable	Equation	Domain
ρ	$\begin{cases} \delta + 6(\frac{r}{r_0})^8 & \text{if } r = \sqrt{x^2+y^2} \leq r_0 \\ 1+\delta & \text{else} \end{cases}$	$\boldsymbol{X} \in [0;1]^2$
$\boldsymbol{U}$	$\begin{cases} 0.83\boldsymbol{X} & \text{if } r = \sqrt{x^2+y^2} \leq r_0 \\ 0 & \text{else} \end{cases}$	$\boldsymbol{X} \in [0;1]^2$
ε	$g(r)\begin{cases} 0.25 + 3(1-\frac{r}{r_0}) & \text{if } r = \sqrt{x^2+y^2} \leq r_0 \\ \delta & \text{else} \end{cases}$	$\boldsymbol{X} \in [0;1]^2$

Table 3: Initial shapes used to test the vectorial remapping schemes. We consider $r_0 = 0.75$ and $\delta = 10^{-6}$ and a high frequency perturbation of specific internal energy as a function of radius $0.2 \leq r \leq 0.7$ as: $g(r) = 1.16656\, e^{-100(x-0.5)^2} \sin(25\pi r)$.

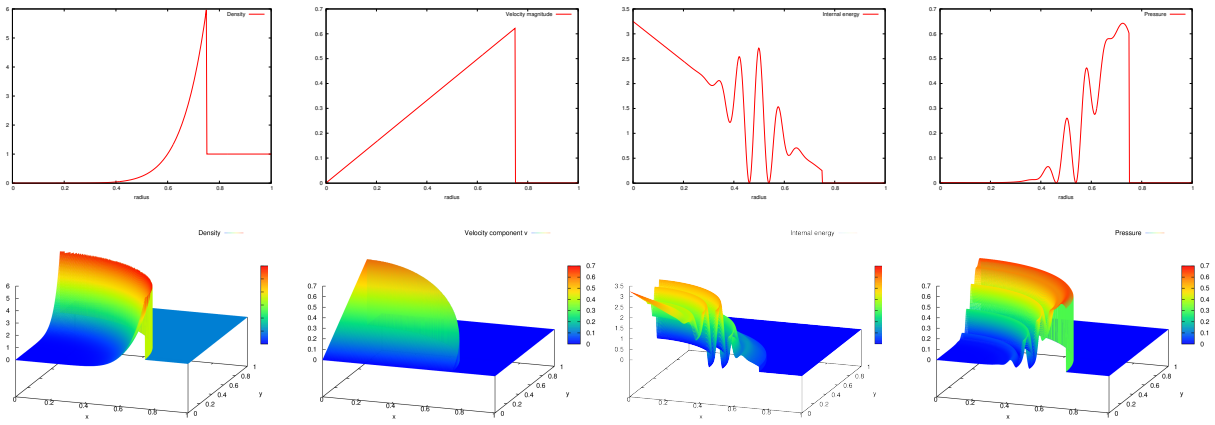


Figure 6: Exact profiles for the vectorial remapping test case. Top panels: profiles as a function of radius. Bottom panels: profiles as a function of $\boldsymbol{X} = (x, y)$. From left to right: density, velocity, internal energy and pressure. Note that the magnitude of the velocity is presented on the top panel and the v component on the bottom panel.

impacted by the perturbation on the energy initialization.

The meshes. Our methodology does not depend on the shape of the cells, as a consequence meshes made of quadrangular, triangular or general polygons can be considered without any restriction.
We have considered general polygonal meshes obtained through the Voronoi tessellation described in [99, 100], and summarized in [71]. Let us rapidly rephrase the main steps of the construction of a Voronoi tessellation. We consider a convex computational domain Ω in 2D. For a set of generators $\boldsymbol{G}_i = (x_i,\, y_i) \in \Omega$, the Voronoi cell Ω_i is defined by, see [90]

$$\Omega_i = \{\boldsymbol{X} = (x,\, y) \in \Omega : |\boldsymbol{X} - \boldsymbol{G}_i| \leq |\boldsymbol{X} - \boldsymbol{G}_j|, \text{for all } i \neq j\} \tag{20}$$

The Voronoi cell Ω_i is a convex polygon, and the set of Voronoi cells indeed covers Ω without holes or overlaps defining *de facto* the so-called tessellation of Ω. We use an algorithm for the construction of Voronoi tessellation which is described in [99, 100]. Depending on the position of generators, the Voronoi mesh can be genuinely non-uniform and one of the measure of non-uniformity is how far the centroid of a Voronoi cell is from the associated generator. In other words, the closer the centroids and the generators are, the more uniform the mesh is. Ultimately, the Voronoi tessellation for which the position of the cell centroid coincides, is called Centroidal Voronoi Tessellation (CVT) [55] and are asymptotically made of perfect hexagons. For our purpose we use a mesh smoother the so-called Lloyd's algorithm [34, 35, 36, 33].

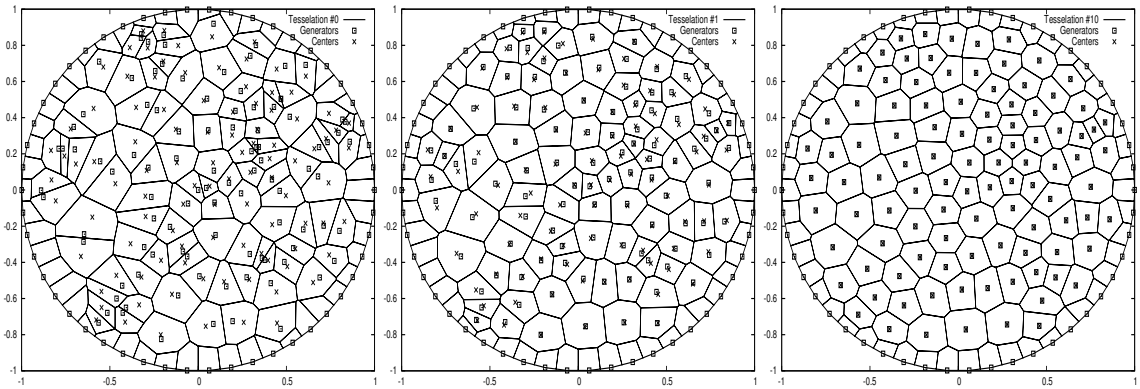


Figure 7: Illustration of the iterative Lloyd's algorithm to produce smooth Voronoi meshes on the unit disk (50 cells on the boundary, 200 inside the disk). Left panel: iteration 0, middle panel: iteration 1, right panel: iteration 10.

It simply consists on an incomplete iterative method to create a CVT. For instance this algorithm starts with an arbitrary distribution of generators (in our example on the unit disk, 50 fixed generators on the boundary and 200 randomly distributed in the disk), see the squares on the left panel of figure 7, constructs the corresponding Voronoi tessellation and deduces the centroids of the constructed Voronoi cells. On the next iterate, the centroids of the previous iterate are used as new generators, a new Voronoi mesh is created associated with its centroids, and so on. As an illustration the resulting meshes after one, two and ten iterations are presented in figure 7 and, visually, the regularity has improved during this process though the initial connectivity may have changed. Note that we can constrain some generators to remain fixed as we have done on the disk boundary. This may help to create some boundary zones made of rectangles should we desire so. All polygonal meshes in this paper are created following this procedure.

The schemes. The remap schemes which we are considering are the following ones

- UNLIM-$\mathbb{P}_k$: Unlimited of nominal accuracy $k+1$, that is with kth polynomial degree reconstructions,
- MUSCL: Second order accurate with Barth-Jespersen limiter [8, 9, 5, 6] using piece-wise linear reconstructions, that is $\mathbb{P}_1^{\text{LIM}}$,
- MOOD-$\mathbb{P}_k$: High order accurate with *a posteriori* stabilization as described in this paper with cascade $\mathbb{P}_5 \to \mathbb{P}_3 \to \mathbb{P}_1 \to \mathbb{P}_1^{\text{LIM}} \to \mathbb{P}_0$, and the following detection criteria
  - DMP: strict discrete maximum principle as NAD (Numerical Admissible Detection). (This scheme is proposed only as an illustration.)
  - DMPu2+PAD: relaxed discrete maximum principle as NAD along with Physical Admissible Detection (PAD) criteria as described in this paper.

Using these schemes one expects to observe some general behaviors:

On smooth solutions, the unlimited schemes provide the optimal order of convergence as well as MOOD like schemes whereas MUSCL scheme order of convergence is strictly less than 2. For MOOD-$\mathbb{P}_1$ which is of the same order of accuracy than MUSCL we expect that the accuracy of MOOD over-tops MUSCL one.

On discontinuous profiles, the unlimited schemes are oscillatory whereas MUSCL and MOOD schemes have an ENO behavior. The order of accuracy of MUSCL and MOOD can not exceed one due to the discontinuity in the solution, nevertheless MOOD results should present a better accuracy.

On difficult problems involving loss of positivity of pressure, one expects to observe the failure of the MUSCL scheme and, of course, of any unlimited scheme of order greater than one. MOOD and the finite volume remapper (unlimited of order one) are the only ones able to maintain the positivity not only of ρ but also ε and consequently p.

For most of the tests we expect that the number of problematic cells detected by MOOD remains of the order of few percents of the total amount of cells if the mesh is fine enough to capture the main features of the flow.

As a consequence the cost of a MOOD scheme must be close to the unlimited scheme of the same order on the same mesh. Indeed the extra cost brought by the problematic cells may be negligible if their number is small. For second order schemes, one expects MOOD scheme to be less expensive than MUSCL for the same reason.

4.1. Successive remappings of smooth profiles

On the unit square we define 5 Voronoi tessellations using random generators and smoothed by 10 iterations of Lloyd's algorithm. These meshes denoted $\mathcal{M}_k$, $k = 1, \cdots, 5$, differ by the fact that their internal generators are randomly located, yielding 5 uncorrelated polygonal meshes. The sequence of projection is $\mathcal{M}_1 \to \mathcal{M}_2 \to \mathcal{M}_3 \to \mathcal{M}_4 \to \mathcal{M}_5 \to \mathcal{M}_1$ so that the errors are easily computed at the end of any number of this sequence. We consider polygonal meshes of size $32^2, 64^2, 128^2, 256^2$ and 512^2. In the next sections we will remap the smooth profiles defined in table 2.

In table 4 we report the errors in L^1, L^2 and L^∞ errors as well as orders of convergence for MOOD remap scheme of nominal 2nd, 3rd and 6th order of convergence for the smooth profiles from table 2 and a polygonal mesh made of N^2 cells. For the first two polynomial profiles we observe that either the schemes are exact up to a machine precision error or converges with the expected orders of convergence. For the third polynomial which is of higher degree than the 6th order remap scheme, as well as for the sine and Gaussian profiles the expected order of convergence are reached. We have omitted the errors produced by the non limited high accurate schemes as they are extremely close to the ones obtained by MOOD remap schemes. To confirm this behavior we have gathered in table **??** the number of bad cells re-updated with one of the lower order scheme in the cascade for one smooth profile only. As can be seen almost no cell is ever detected, consequently any *a posteriori* MOOD schemes are run unlimited on these smooth profiles. This is expected for any scheme but not always attained with classical *a priori* stabilization techniques.

		MOOD-$\mathbb{P}_1 \to \mathbb{P}_0$ scheme						MOOD-$\mathbb{P}_3 \to \mathbb{P}_2 \to \mathbb{P}_1 \to \mathbb{P}_0$ scheme						MOOD-$\mathbb{P}_5 \to \cdots \to \mathbb{P}_0$ scheme					
Polynomial 1	N_x	L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error	
	32	1.01E-05	—	1.29E-05	—	4.28E-05	—	2.23E-16	—	2.93E-16	—	7.89E-16	—	2.18E-16	—	2.87E-16	—	6.77E-16	—
	64	3.12E-06	1.99	3.95E-06	2.02	1.62E-05	1.65	2.24E-16	∞	2.90E-16	∞	7.83E-16	∞	2.27E-16	∞	2.91E-16	∞	7.83E-16	∞
	128	9.07E-07	1.96	1.15E-06	1.96	4.07E-06	2.19	2.17E-16	∞	2.83E-16	∞	8.92E-16	∞	2.19E-16	∞	2.85E-16	∞	1.00E-15	∞
	256	2.45E-07	1.99	3.11E-07	1.99	1.51E-06	1.51	2.19E-16	∞	2.84E-16	∞	1.11E-15	∞	2.20E-16	∞	2.86E-16	∞	1.45E-15	∞
	512	6.36E-08	2.00	8.03E-08	2.01	4.10E-07	1.93	2.18E-16	∞	2.83E-16	∞	1.11E-15	∞	2.19E-16	∞	2.85E-16	∞	1.11E-15	∞
Polynomial 2	N_x	L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error	
	32	3.45E-05	—	4.95E-05	—	1.71E-04	—	1.33E-07	—	1.97E-07	—	4.95E-07	—	2.34E-16	—	2.99E-16	—	7.42E-16	—
	64	1.07E-05	2.00	1.50E-05	2.03	6.61E-05	1.62	8.52E-09	4.68	1.18E-08	4.79	4.14E-08	4.23	2.25E-16	∞	2.90E-16	∞	9.08E-16	∞
	128	3.09E-06	1.97	4.40E-06	1.95	3.00E-05	1.26	5.78E-10	4.27	7.66E-10	4.34	2.99E-09	4.17	2.82E-16	∞	2.82E-16	∞	8.98E-16	∞
	256	8.27E-07	2.00	1.18E-06	2.01	1.12E-05	1.49	3.90E-11	4.10	5.06E-11	4.13	2.71E-10	3.65	2.23E-16	∞	2.86E-16	∞	1.43E-15	∞
	512	2.14E-07	2.00	3.04E-07	2.00	2.73E-06	2.09	2.52E-12	4.05	3.25E-12	4.07	2.14E-11	3.76	2.23E-16	∞	2.85E-16	∞	1.07E-15	∞
Polynomial 3	N_x	L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error	
	32	7.01E-05	—	1.24E-04	—	6.76E-04	—	1.34E-06	—	3.51E-06	—	1.79E-05	—	6.30E-09	—	1.09E-08	—	6.25E-08	—
	64	2.22E-05	1.96	3.73E-05	2.05	2.74E-04	1.54	8.68E-08	4.66	1.95E-07	4.92	1.67E-06	4.04	1.20E-10	6.74	1.78E-10	7.00	9.32E-10	7.16
	128	6.19E-06	2.03	1.07E-05	1.98	1.47E-04	0.99	5.97E-09	4.25	1.19E-08	4.45	1.12E-07	4.30	2.60E-12	6.09	3.79E-12	6.11	3.82E-11	5.07
	256	1.67E-06	1.99	2.88E-06	2.00	4.24E-05	1.89	4.01E-10	4.10	7.48E-10	4.20	7.99E-09	4.01	4.74E-14	6.09	6.76E-14	6.12	5.21E-13	6.53
	512	4.36E-07	1.99	7.52E-07	1.99	1.39E-05	1.65	2.61E-11	4.05	4.77E-11	4.08	7.03E-10	3.60	8.56E-16	5.94	1.19E-15	5.98	1.59E-14	5.16
Sine	N_x	L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error	
	32	5.64E-04	—	6.87E-04	—	2.33E-03	—	2.39E-05	—	3.04E-05	—	9.80E-05	—	1.32E-06	—	1.70E-06	—	6.72E-06	—
	64	1.67E-04	2.07	2.10E-04	2.01	9.40E-04	1.55	2.27E-06	4.01	2.92E-06	3.99	1.54E-05	3.15	3.73E-08	6.08	4.84E-08	6.06	2.39E-07	5.68
	128	4.62E-05	2.04	5.88E-05	2.03	2.70E-04	1.98	1.87E-07	3.97	2.42E-07	3.96	1.22E-06	4.03	8.55E-10	6.00	1.13E-09	5.96	6.43E-09	5.74
	256	1.23E-05	2.01	1.57E-05	2.00	8.49E-05	1.76	1.34E-08	4.01	1.75E-08	4.00	1.51E-07	3.17	1.67E-11	5.99	2.23E-11	5.97	1.40E-10	5.82
	512	3.21E-06	1.99	4.09E-06	1.99	3.06E-05	1.51	9.03E-08	3.99	1.17E-09	4.00	7.58E-09	4.43	2.94E-13	5.98	3.93E-13	5.98	3.46E-12	5.47
Gaussian	N_x	L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error		L^1 error		L^2 error		L^∞ error	
	32	1.06E-04	—	2.00E-04	—	1.27E-03	—	5.67E-06	—	1.04E-05	—	4.25E-05	—	5.75E-07	—	9.42E-07	—	3.61E-06	—
	64	3.33E-05	1.97	6.31E-05	1.96	4.76E-04	1.66	5.26E-07	4.05	9.43E-07	4.09	5.02E-06	3.64	1.61E-08	6.08	2.73E-08	6.03	1.47E-07	5.45
	128	9.42E-06	2.00	1.81E-05	1.98	1.48E-04	1.86	4.44E-08	3.93	7.88E-08	3.94	4.95E-07	3.68	3.76E-10	5.97	6.47E-10	5.94	3.55E-09	5.91
	256	2.54E-06	1.99	4.94E-06	1.99	5.82E-05	1.42	3.26E-09	3.97	5.95E-09	3.93	5.00E-08	3.49	7.41E-12	5.97	1.32E-11	5.92	1.12E-10	5.26
	512	6.55E-07	2.00	1.27E-06	2.01	1.74E-05	1.79	2.19E-10	4.00	4.00E-10	4.00	4.40E-09	3.60	1.30E-13	5.99	2.34E-13	5.97	3.45E-12	5.15
Expected order			2		2		2		4		4		4		6		6		6

Table 4: L^1, L^2 and L^∞ errors and convergence rates for the smooth profiles for successive remaps on polygonal grids for MOOD-$\mathbb{P}_k$ schemes, $k = 1, 3, 5$.

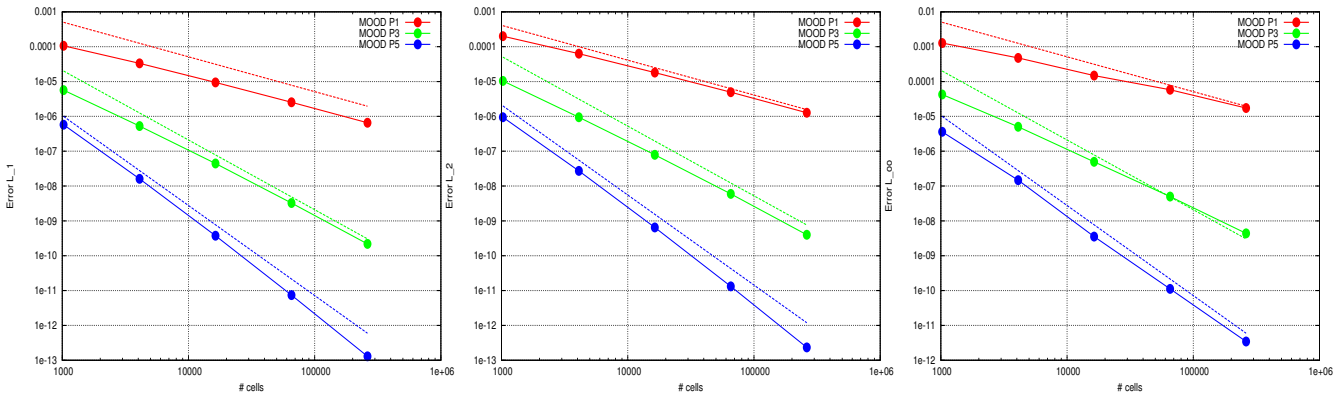


Figure 8: L_1, L_2, L_∞ errors for the smooth Gaussian profile remap in logscale as a function of mesh size for successive remaps on polygonal grids for MOOD-$\mathbb{P}_k$ schemes, $k = 1, 3, 5$ — Exact expected convergence rate are drawn with dashed lines.

Next in figure 8 we present the errors for the Gaussian profile in logscale as a function of mesh size for the three MOOD remap schemes. The data corresponds to the last row of table 4. As already seen on the figures of the table the expected orders of convergence are attained in all norms even if L_∞ norm results present some defects. Nonetheless the convergence of all MOOD schemes is achieved. It is important to remark that the finest mesh (512^2) error for the second order accurate MOOD-$\mathbb{P}_1$ scheme in red is larger than the coarsest mesh (32^2) error for the sixth order accurate MOOD-$\mathbb{P}_5$ scheme in blue. In other words to maintain approximately the same accuracy, a second-order accurate scheme needs about 256 times more cells. Later in section 4.4 we will observe that the MOOD-$\mathbb{P}_5$ scheme is 6 times more expensive than MOOD-$\mathbb{P}_1$. These results suggest that a (very) high accurate remap scheme is competitive in 2D on smooth profiles.

4.2. Rotating polygonal mesh remap of discontinuous profiles

The computational domain is the unit disk and we define five Voronoi tessellations made of 64^2 cells smoothed by 5 iterations of Lloyd's algorithm, see the description at the begining of section 4. Next we apply 10 rotations of angle θ where $\theta = 2\pi/N_r$ with $N_r = 10$. Consequently the last mesh coincides with the initial one and error computations are performed between the final remapped solutions and the initial data. The rotation angle is chosen relatively large so that two consecutive meshes have almost no connection close to the border of the disk, while coinciding in the vicinity of the center of rotation. As such 1000 remaps are performed.

In figure 9 we present the $x-z$ planar view of the three shapes in the top-left panel. This also coincides with the reference solution after 1000 remaps. Next we present the results provided by a classical second-order limited remap scheme (MUSCL) for which large diffusion can be observed both on the smooth peaks and also on the discontinuous cylinder. The second, third and fourth lines present the results for the second-, fourth- and fifth-order accurate remap schemes. The left panels are dedicated to the unlimited scheme (no stabilization at all), the middle panels present the results obtained by a MOOD scheme with a strict DMP detection, and, the right panels show the current MOOD scheme results. Red/blue color emphasizes the over/under-shoots produced by the schemes.

As expected the unlimited schemes accurately maintain the smooth peaks but oscillates close to the cylinder discontinuities and the C^1 discontinuity around the smooth profiles. Contrarily the MOOD scheme with strict DMP is able to maintain the cylinder in bounds but clip the maximal values on the smooth profiles. Not only the maximum value is diffused but the general shapes of these profiles are also diffused. Increasing the polynomial degree (the formal order of accuracy) does improve the results on the cylinder but not on the

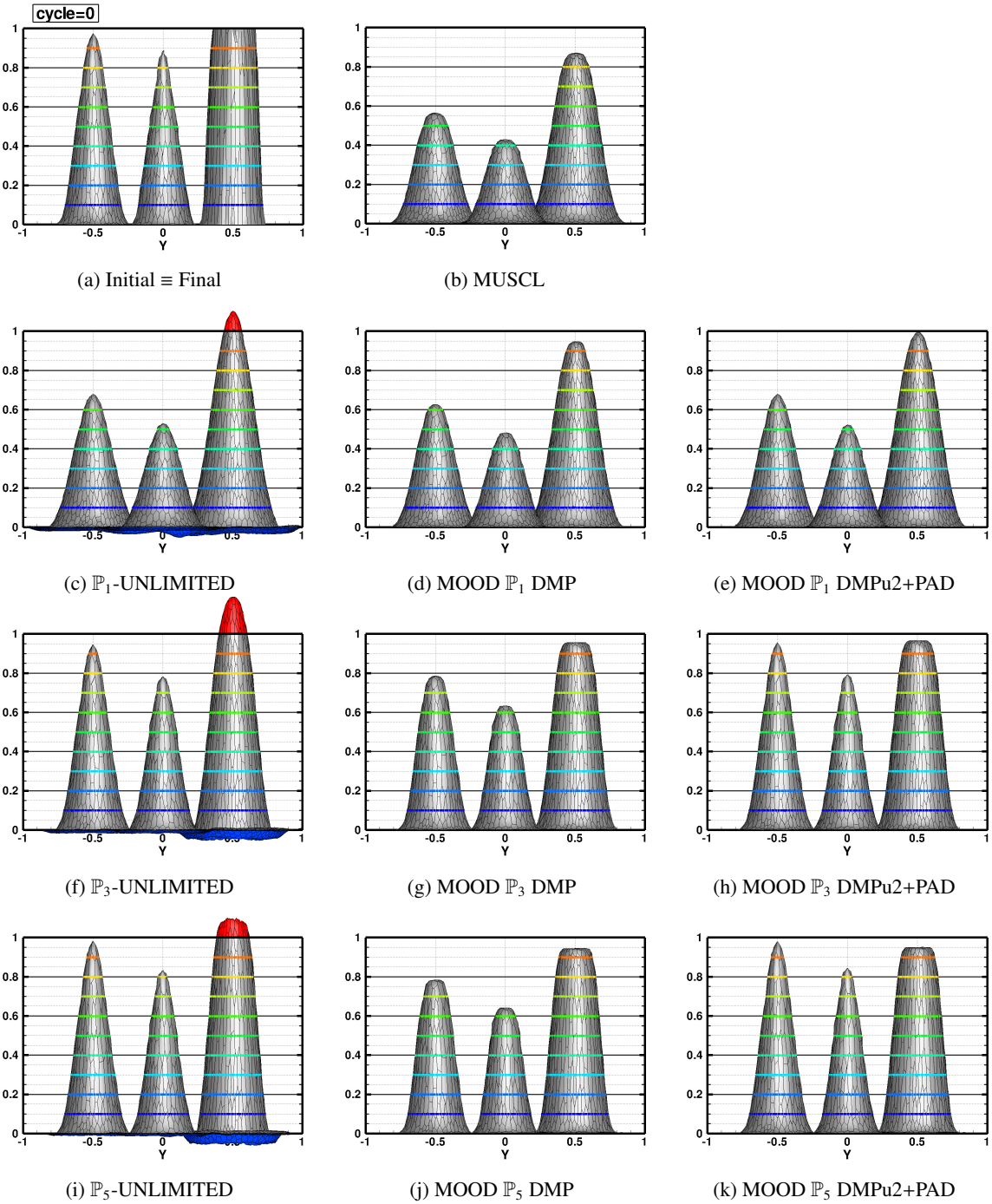
smooth profiles or only marginally. Last the genuine MOOD remap scheme is able to reproduce the accurat peaks that the unlimited scheme did produce, and, at the same time maintain the cylinder in bounds without spurious oscillations.
This test shows several phenomena. First that the strict application of the DMP even with the use of large polynomial degree is inadequate to improve the overall numerical results. Second, the application of a relaxed DMP with a PAD criteria results in a MOOD remap scheme which maintains the smooth profile and assure that the solution remains in bounds. (The accuracy of such MOOD scheme is almost comparable with the unlimited scheme on smooth profiles). Third, there is a net gain in employing more accurate reconstructions. Last, even if MOOD-$\mathbb{P}_1$ and MUSCL remap scheme are of the same nominal accuracy, the MOOD results are significantly more accurate and as stable.

In figure 10 are displayed on the top line the polynomial degrees employed by the MOOD $\mathbb{P}_5$ on the initial mesh (during the first remap stage) and on the but-last mesh (during the last remap). The right panel is a full view of the results from panel 9k of figure 9. The degrees can be $0, 1, 3, 5$ and -1 corresponding to a $\mathbb{P}_1$ reconstruction with slope limiting. From these figures we can observe that the decrementing was effective close to the discontinuities as expected. At final time, some diffusion has occured and the decrementing is not needed anymore apart from false detections on plateau[6], and few cells on the cylinder sharp transitions. The bottom panels of figure 10 present the location of the bad cells detected with the PAD (blue) or NAD (red) criteria for the final remap stage of the MOOD $\mathbb{P}_5$ scheme. The panel on the right presents the same results but on a log-scale to emphasize the modification of the curvatures detected by the u2 criteria (15) of the NAD for red cells close to the border of the three shapes. These cells trully need some decrementing to avoid spuriour oscillations. We can notice that even with a 6th order accurate remap scheme, enough diffusion has been deposit so that almost unlimited reconstructions can be used everywhere but on few cells. Consequently computing some sort of limiter apart from these few cells seems a waste of computational resources.

Last but not least, for the MOOD $\mathbb{P}_5$ remap scheme results, we provide in figure 11 the percentage of cells updated with one of the lower scheme in the cascade, i.e $\rightarrow \mathbb{P}_3 \rightarrow \mathbb{P}_1 \rightarrow \mathbb{P}_1^{\mathrm{LIM}} \rightarrow \mathbb{P}_0$. We observe that the percentage cells updated with any low order scheme does not exceed 0.2%. Overall the ratio of re-updated cells never exceeds 0.5%. Notice that in this scalar case the $\mathbb{P}_0$ scheme in the cascade is never triggered because the slope limiter is able to assure the preservation of the physical bounds 0 and 1. As such the slope limiter in the scalar case is sufficient to preserve physical bounds. This may not be the case anymore for a system of interleaved variables.

This test concludes our numerical study on the scalar case. We have observed that our high accurate remap scheme can reach the optimal order of accuracy on smooth profiles and maintain a stable solution on discontinuous profiles. This validates our *a posteriori* detection decrementing stabilization on the scalar case. Next sections will present the results obtained on a more demanding emulation of the hydrodynamics system of equations.

[6]These have no implication because on a plateau any reconstruction is accurate enough. Note that we could also add a test to avoid decrementing when a plateau is detected to avoid the extra cost of cell remap recomputation.



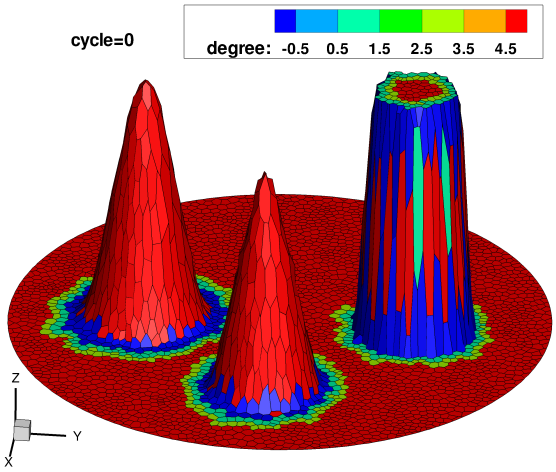
(a) Initial ≡ Final (b) MUSCL

(c) $\mathbb{P}_1$-UNLIMITED (d) MOOD $\mathbb{P}_1$ DMP (e) MOOD $\mathbb{P}_1$ DMPu2+PAD

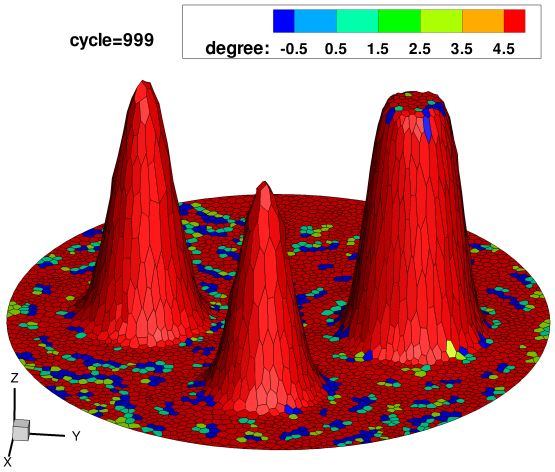
(f) $\mathbb{P}_3$-UNLIMITED (g) MOOD $\mathbb{P}_3$ DMP (h) MOOD $\mathbb{P}_3$ DMPu2+PAD

(i) $\mathbb{P}_5$-UNLIMITED (j) MOOD $\mathbb{P}_5$ DMP (k) MOOD $\mathbb{P}_5$ DMPu2+PAD

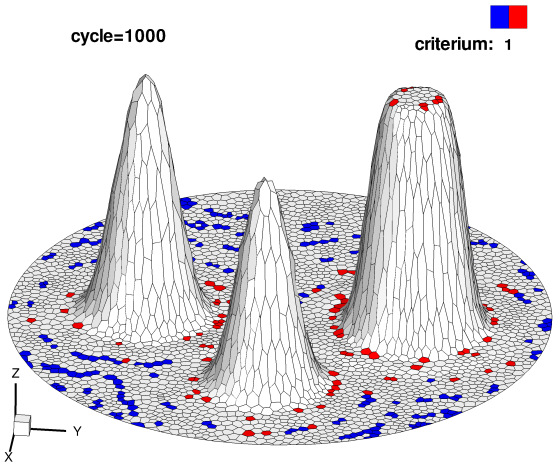
Figure 9: Results for the remapping of three shapes (Cylinder, Hump, Cone), polygonal mesh made of 64^2 cells, 10 rotations (1000 remaps). $x - z$ projected planar view, the red/blue cells present over/under-shoots — First-line: initial and final data, MUSCL remap scheme results — Second-, third-, fourth-lines: $\mathbb{P}_k$-UNLIMITED remap scheme, MOOD $\mathbb{P}_k$ with strict DMP, and current MOOD $\mathbb{P}_k$ (with relaxed DMP and PAD detection criteria) for $k = 1, 3, 5$.



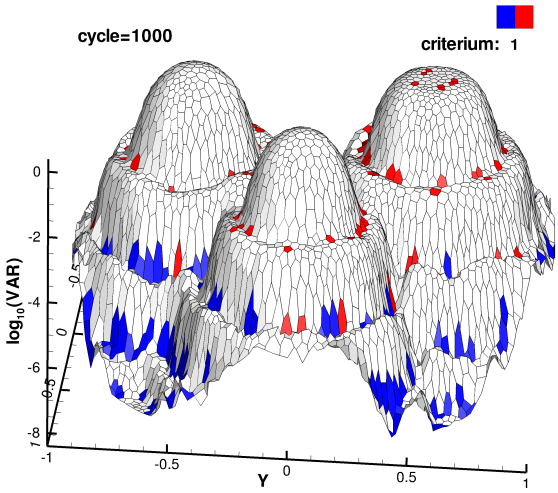
(a) First mesh



(b) Final mesh



(c) PAD-NAD on final mesh



(d) PAD-NAD on final mesh (elevation in log-scale)

Figure 10: MOOD $\mathbb{P}_5$ remap scheme results for the three shape problem — Top-line: cell polynomial degree on the first (left) and last (right) remap step — Bottom-line: map of the PAD (blue) and NAD (red) activated cells on the final remap step. Right panel is the log scale of the left one to emphasize the modifications of the curvature which is used to detect bad cells in the u2 criteria.

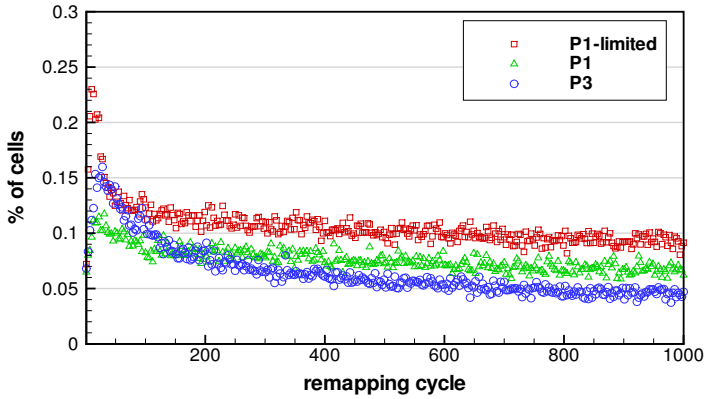


Figure 11: Evolution of the ratio of decremented cells for MOOD $\mathbb{P}_5$ scheme after 1000 remaps on the three shape problem. Blue: % of cells updates with the fourth order unlimited scheme. Green: % of cells updates with the second order unlimited scheme. Red: % of cells updates with the second order scheme with Barth-Jespersen limiter.

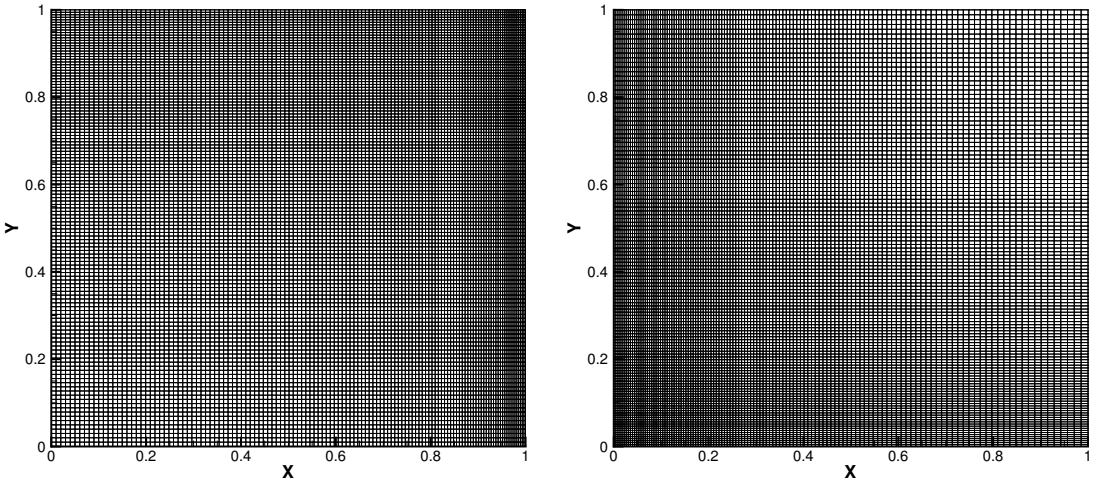


Figure 12: Examples of two meshes obtained during the cyclic remapping motion with $M = 128$ and $N = 1000$. Iterations 320 and 740.

4.3. Remapping on systems of variables

In this section we consider the remapping of four conserved components (mass, momentum and total energy) which are linked by means of a non-linear relation, refer to table 3.
The unit square is paved by a uniform $M \times M$ quadrangular mesh. The mesh is animated by a non uniform trully 2D motion following

$$x(\xi, t) = [1 - \alpha(t)]\,\xi + \alpha(t)\xi^3, \quad y(\eta, t) = [1 - \alpha(t)]\,\eta + \alpha(t)\eta^2, \quad \alpha(t) = \frac{1}{2}\sin(4\pi t), \tag{21}$$

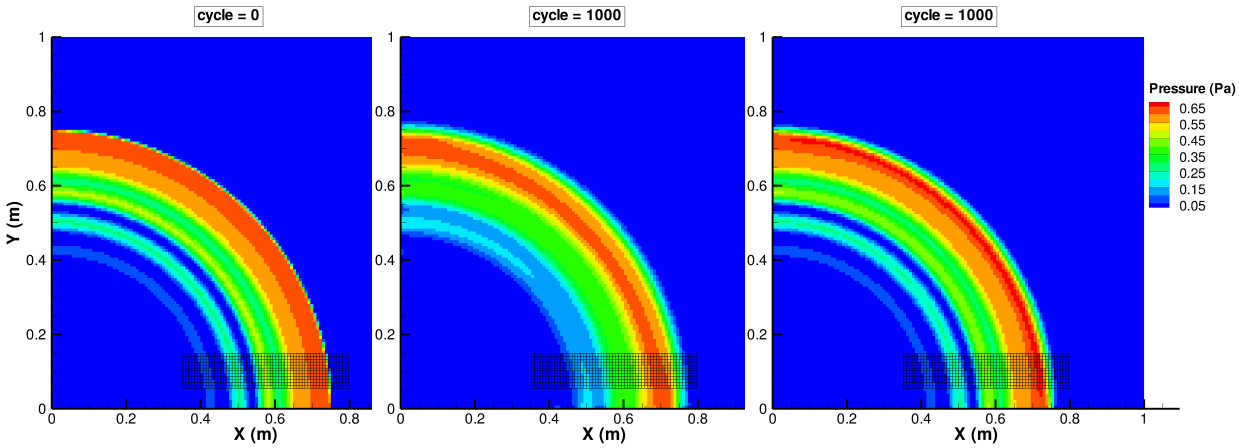
for $0 \le \xi, \eta \le 1$ and $0 \le t \le 1$. The vertex positions at pseudo time t^n $\boldsymbol{X}^n_{i,j}$ are defined by

$$\boldsymbol{X}^n_{i,j} = \left(x(\xi_i, t^n), y(\eta_j, t^n)\right) = \left(x\left(\frac{i-1}{M}, \frac{n}{N}\right), y\left(\frac{j-1}{M}, \frac{n}{N}\right)\right), \quad 0 \le n \le N, \quad 1 \le i, j \le M + 1. \tag{22}$$

usually refered to as a cyclic remapping motion [87, 88, 63, 64, 68, 106, 108]. Note that the first and last meshes do coincide. We choose $M = 128$ and $N = 1000$, see figure 12 for two examples at iterations 320 and 740.

We test the MUSCL remap scheme along with the sequence of MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ all using the DMP+u2 numerical detection criteria (NAD) and the positivity of density, specific internal energy and pressure as physical detection criteria (PAD). The MUSCL remap scheme applies a limiter on the density, momentum and total energy variables independently[7]. The goal of this test case is to show that the shock wave is maintained as accurate as possible, at least as accurate as the MUSCL scheme but with a clear improvement on the smooth regions of the flow, more specifically on the high frequency perturbation. In figure 17 we plot the initial pressure on the left. Ideally this initial pressure profile should be reproduced by an accurate remap scheme. In the middle/right panels are displayed the results given by a MUSCL like remap scheme after 1000 remaps and the MOOD-$\mathbb{P}_5$ results. A small portion of the final mesh is also

[7]Other choices of limiter function or variables to remap may improve slightly the MUSCL remap results. Nevertheless the general accuracy and main behaviors of a MUSCL remap scheme do remain comparable with the ones in this work.



(a) Initial/exact pressure (b) Final - MUSCL scheme (c) Final - MOOD-$\mathbb{P}_5$ scheme

Figure 13: Remapping on a system of variables — Exponential shock with high frequency perturbation problem — 128×128 mesh under 1000 cyclic remapping steps — Left panel: initial pressure which also corresponds to the target solution. Middle panel: pressure given by the MUSCL remap scheme. Right panel: pressure given by the MOOD-$\mathbb{P}_5$ remap scheme.

reported. From these figures we can observe that an excessive diffusion of the MUSCL remapper has destroyed the smooth flow for radius $r \leq 0.6$ whereas the 6th order accurate MOOD scheme can maintain these small scale features after 1000 remap stages. On the shock front, at $r \simeq 0.75$ both remap schemes need to diffuse the discontinuity with at most a first order accurate scheme to avoid spurious oscillations.

Because the solution is expected to remain cylindrical we plot the density, specific internal energy and pressure as a function of the cell radius for all cells after the remapping in red along with the initial conditions in blue, see figure 14. Each row presents the results produced by of one remap scheme; MUSCL on the first row, MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$ and MOOD-$\mathbb{P}_5$ on the last one. From these plots we oberve that the formally second order accurate schemes (MUSCL and MOOD-$\mathbb{P}_1$) have difficulties to maintain accurate results because of diffusion (loss of extrema) and dispersion (drift) errors. Nonetheless MOOD seems to produce slightly more accurate results. When the nominal accuracy of the MOOD scheme is increased then both errors are drastically reduced for the smooth parts of the flow. We can virtually remap many times these shapes with only little loss. The shock region, as expected, can be improved only marginally as any remap scheme must drop to a first-order accurate scheme to avoid parasitical oscillations.

In figure 15 we represent the elevation for the density, energy and pressure along with the the red/blue bad cells at the 200th remap step (top row) and for the final 1000th remap step (bottom row). Only MOOD-$\mathbb{P}_5$ remap scheme results are shown. The purpose is to show that each variable may generate bad cells. However if any bad cell flag for one specific variable is waved, the full solution is recomputed. In other word even if only the density variable is flagged as bad, all variables are recomputed with a next dissipative scheme from the cascade.

From these plots we can observe that the number of bad cells diminishes with time as the scheme adds some dissipation at each remap step. Blue cells are flagged because they do not fulfill the physical criteria, mainly the positivity in this situation. Red cells are flagged because the numerics is not valid according to the DMP+u2 detection criteria. In the next figure (figure 16) we present the same results for the internal energy variable and twelve remap stages between 40 and 980. The bad cells are mostly located around discontinuities or steep gradients and their number does not increase in time.

The previous figures have shown how, where and why some cells were flagged as bad. Now in figure 17 we plot the effective polynomial degree of the reconstructions used after the *a posteriori* MOOD loop has

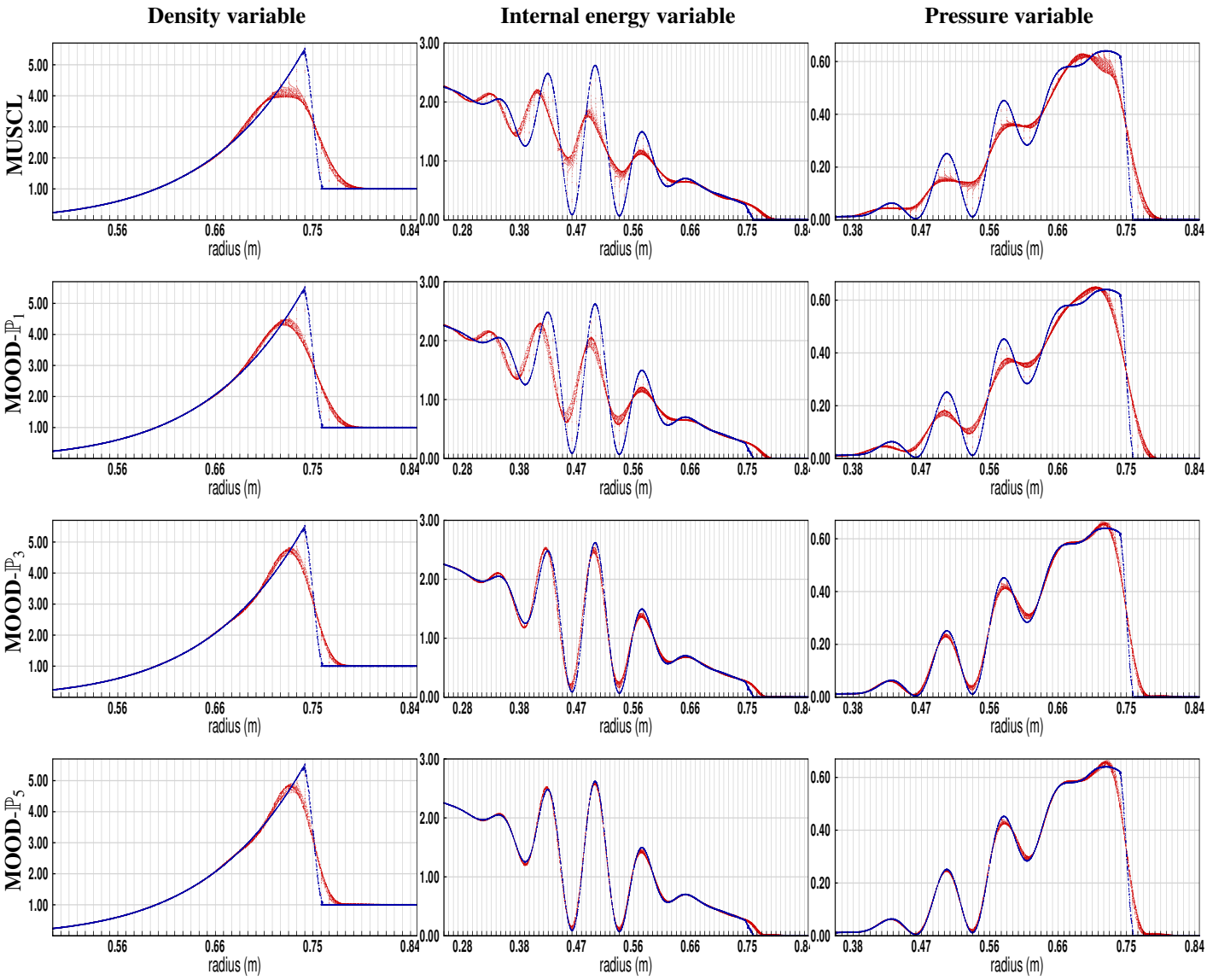


Figure 14: Remapping on a system of variables — Exponential shock with high frequency perturbation problem — 128×128 mesh under 1000 cyclic remapping steps — Left to right: density, specific internal energy and pressure as a function of the cell radius for all cells after the remapping in red along with the initial conditions in blue. Top to bottom: MUSCL, MOOD-$\mathbb{P}_1$, MOOD-$\mathbb{P}_3$, MOOD-$\mathbb{P}_5$ remap scheme results.

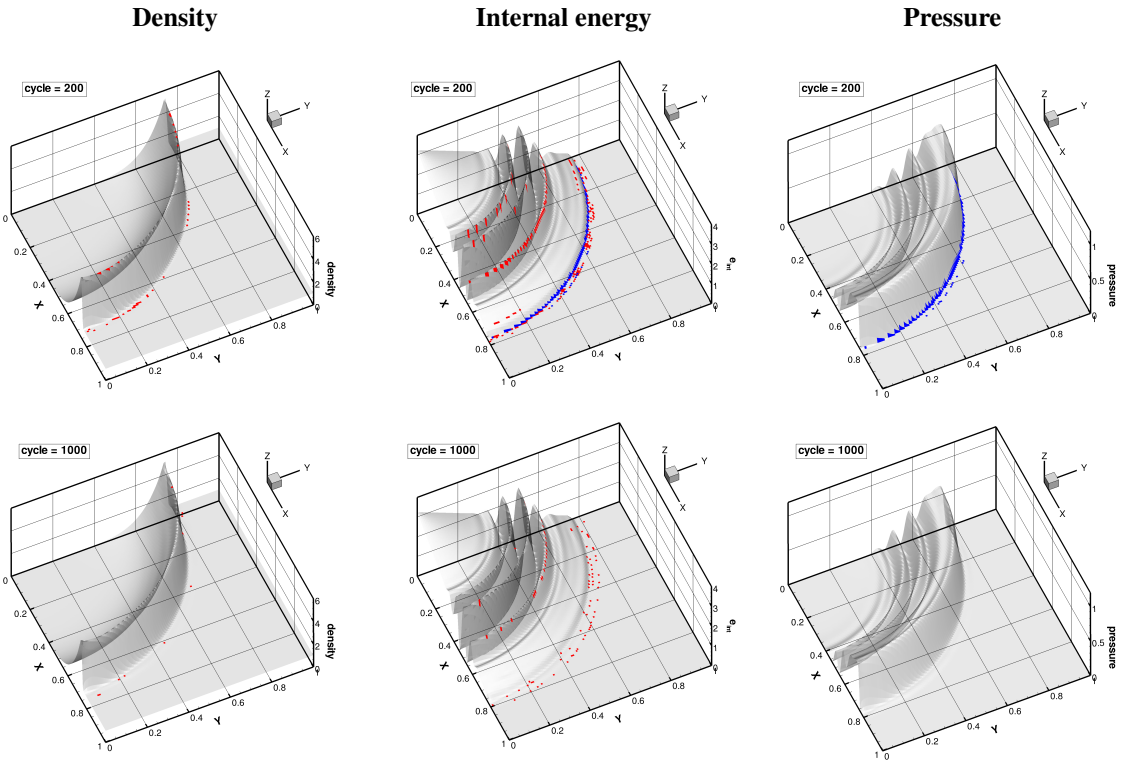


Figure 15: Remapping on a system of variables — Exponential shock with high frequency perturbation problem — 128×128 mesh under 1000 cyclic remapping steps — Detector maps for the density, internal energy and pressure variables (left to right columns). MOOD-$\mathbb{P}_5$ results at remap steps 200 (top row) and 1000 (bottom row). Blue cells: positivity PAD is activated, red cells: NAD is activated.

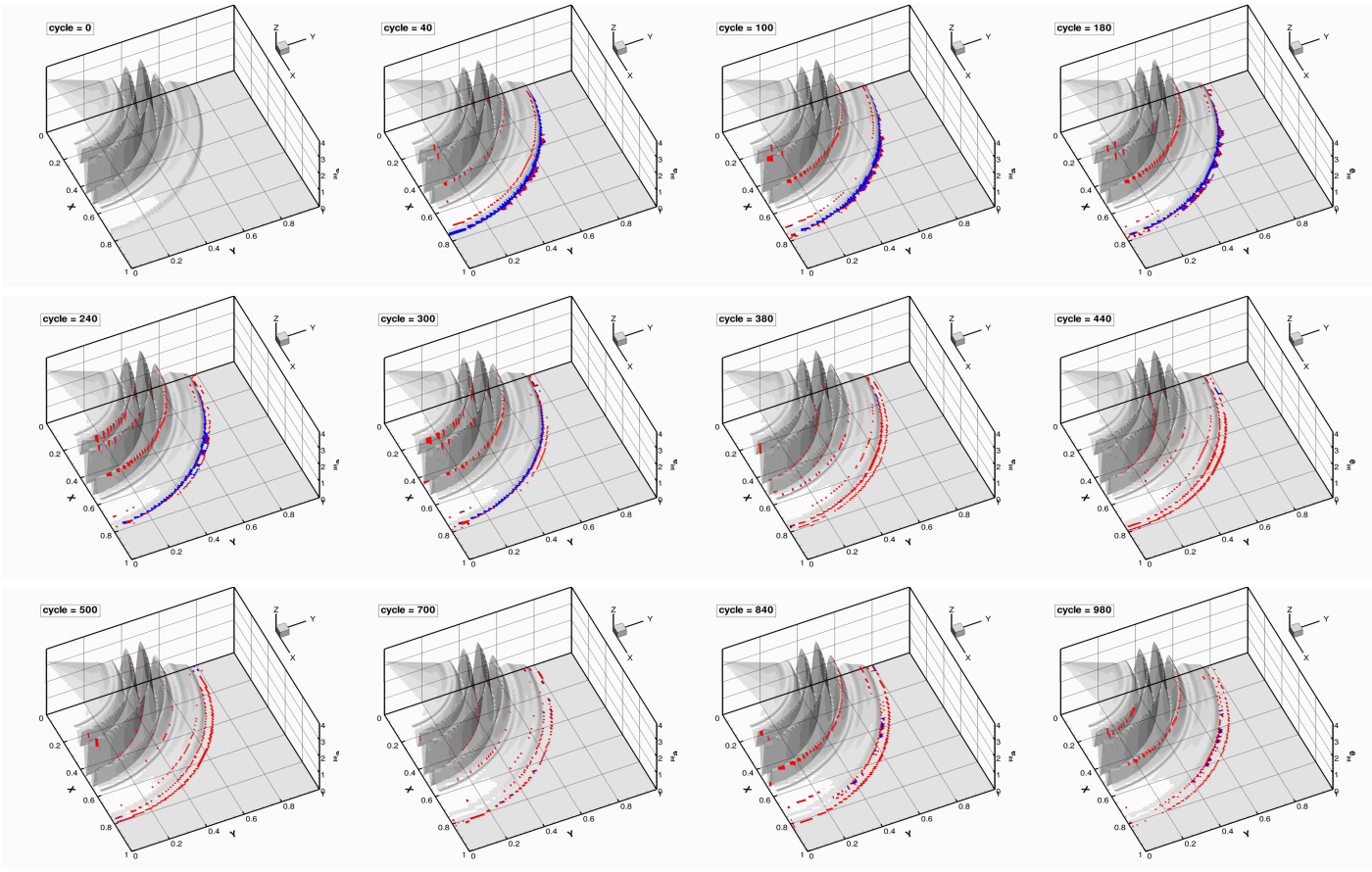
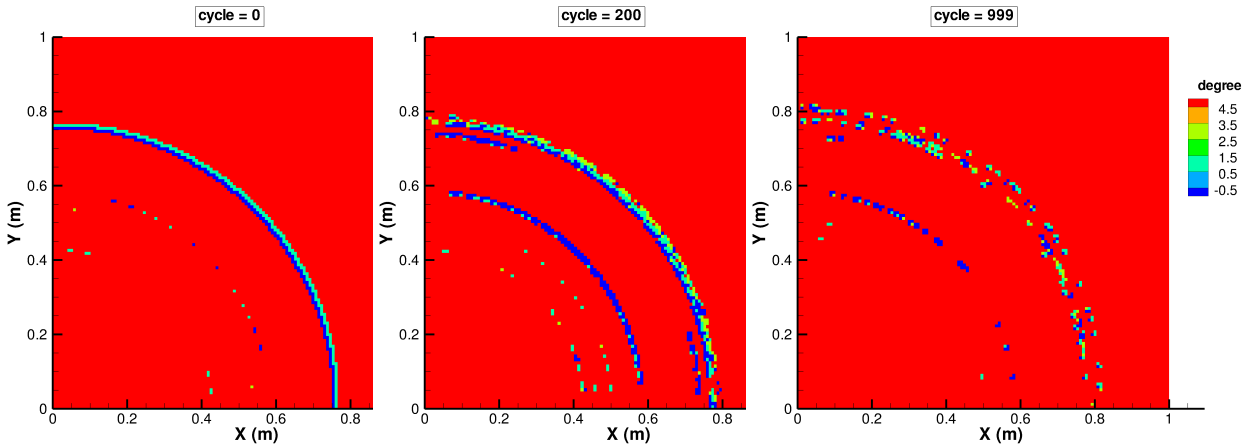


Figure 16: Remapping on a system of variables — Exponential shock with high frequency perturbation problem — 128×128 mesh under 1000 cyclic remapping steps — Detector maps for the internal energy variable MOOD-$\mathbb{P}_5$ results for twelve remap steps. Blue cells: positivity PAD is activated, red cells: NAD is activated.



(a) First remap step (b) 200th remap step (c) Last remap step

Figure 17: Remapping on a system of variables — Exponential shock with high frequency perturbation problem — 128×128 mesh under 1000 cyclic remapping steps — Polynomial degree effectively used by the MOOD-$\mathbb{P}_5$ results for the first, 200th and last remap steps. Polynomial degrees are 5 (red), 3 (green), 1 (light green), 0 (light blue) or -1 (dark blue) when a limiter is used along with a piecewise linear reconstruction.

operated. The first, 200th and last remap stages of the MOOD-$\mathbb{P}_5$ results are shown. Recall that, according to the cascade choosen, the polynomial degrees can take only five discrete values: 5, 3, 1, 0 or -1 when a limiter is used along with a piecewise linear reconstruction. The location of bad cells are of course consistent with the detection flags shown in previous figures. The five colors are used meaning that all schemes in the cascade do play a role in these results even if the majority of bad cells are dealt with the second-order limited scheme.

At last consistently with the scalar case, in figure 18 for MOOD $\mathbb{P}_5$ remap scheme results, we present the percentage of cells updated with one of the lower scheme in the cascade, i.e $\rightarrow \mathbb{P}_3 \rightarrow \mathbb{P}_1 \rightarrow \mathbb{P}_1^{\text{LIM}} \rightarrow \mathbb{P}_0$. We observe that the percentage cells updated with any low order scheme does not exceed 0.05%. The ratio of re-updated cells by any of the lower scheme never exceeds 0.06%. Rarely the cascade drops to the $\mathbb{P}_0$ scheme.

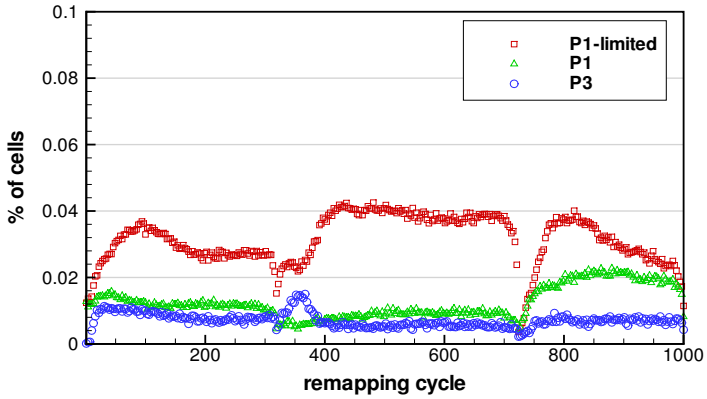


Figure 18: Evolution of the ratio of decremented cells for MOOD $\mathbb{P}_5$ scheme after 1000 remaps on the emulation of the hydrodynamics problem (system of variables). Blue: % of cells updates with the fourth order unlimited scheme. Green: % of cells updates with the second order unlimited scheme. Red: % of cells updates with the second order scheme with Barth-Jespersen limiter.

4.4. Relative cost of the high accurate MOOD-like remapping scheme

This study is made on the cyclic remapping test of the hydrodynamics like profiles. Remind that $N_{\text{cycle}} = 1000$ remappings are performed. As already seen this test involves discontinuous profiles and subsequent stabilization iterations. The code is split into several main subroutines: the mesh intersection, the stabilization procedure involving the detection/decrementing, the projection of the reconstructions that is the integration step and the polynomial reconstruction. All the simulations have been run on a non-dedicated sequential machine and a non-optimized implementation. As such they must be considered as an average tendency.

In table 5 we present such profiling for MOOD-$\mathbb{P}_k$ schemes with k varying from 1 to 5 in order to estimate the extra cost brought by the use of high accurate reconstructions. Notice that a MUSCL remap scheme cost is of the same order than MOOD-$\mathbb{P}_1$ therefore we did not report these figures. Independent of the number of cells in the mesh the same general behaviors are consistently observed: the stabilization cost for less than 10% of the overall cost, the cost of intersection and projection drops drastically when the polynomial degree increases (respectively 40%/60% for $\mathbb{P}_1$ and 8%/19% for $\mathbb{P}_5$), the cost of the reconstruction as expected increases with the polynomial degrees from 0% for $\mathbb{P}_1$ reconstruction up to 70% for $\mathbb{P}_5$.

Let us study the cost of increasing the nomila accuracy of a remap scheme. When the polynomial degree is 2 then the MOOD-$\mathbb{P}_2$ scheme costs about 15% more than a second order scheme (MOOD-$\mathbb{P}_1$). MOOD-$\mathbb{P}_3$ costs about 20% to 30% more, MOOD-$\mathbb{P}_4$ is about two times more expensive, and, MOOD-$\mathbb{P}_5$ six times. The reason for this increase in cost from MOOD-$\mathbb{P}_4$ to MOOD-$\mathbb{P}_5$ can be seen in the cost of the reconstruction stages which jumps from 30% to 70%. AS such this step should be monitored and profiled to improve the efficiency of the code.

Last in figure 19 we compare the L_2 errors produced by the MOOD schemes for the three refined meshes. The left row presents the L_2 error on the density variable on the top panel and the total internal on the bottom panel as a function of the mesh size (in logscale). The density error evolves as expected, the finer the mesh, the smaller the error and the higher the nominal accuracy, the smaller the error. On purpose we also present the total energy error which does not present the same nice behavior as the convergence is slower for high-accurate methods. The right row dislays the L_2 errors for density and total energy as a function of the polynomial degree for each MOOD-$\mathbb{P}_k$ scheme (i.e the nominal order of accuracy minus one). For the density variable the schemes behave as expected: the error decreases when the mesh is refined and the polynomial degree increases. For the energy we observe that even if the general behavior is correct, there are strange situation, for instance for MOOD-$\mathbb{P}_3$ and $\mathbb{P}_4$ the 32×32 mesh error is close to the 128×128 mesh error and smaller than the 64×64 mesh error without obvisou reason. Notice that the correct convergence behavior is also observed for the component of the momentum as we omit these figures here.

With this figure we can conclude that the MOOD-like remapping schemes behave in general as expected, that is, increasing the polynomial degree or the number of cells leads to a more accurate solution. Nonetheless we have also shown that, depending on the variable of interest, this general behavior may or may not be totally satisfactory. In other words, even if cost and error measures are good sensors to estimate the quality of a remapping scheme, only a large set of tests and associated diagnostics can convince the subjective view point of an ordinary end-user.

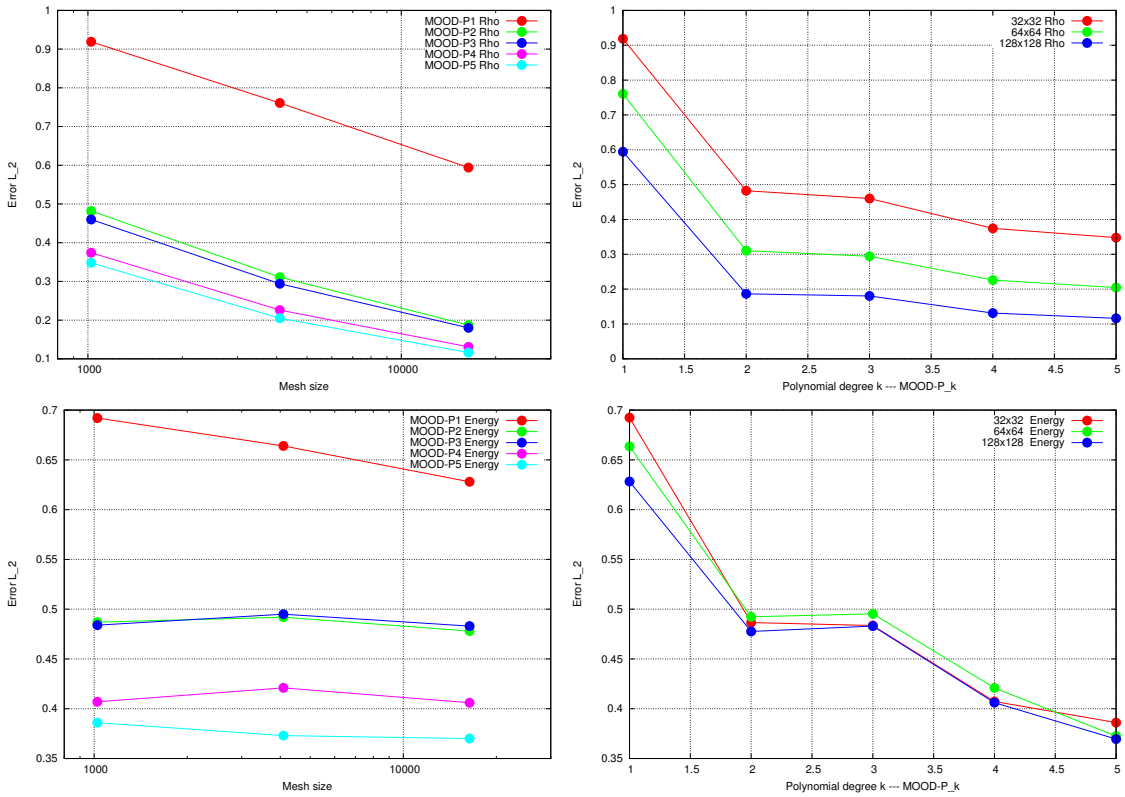


Figure 19: L_2 errors for the density (left) and total energy (right) as a function of mesh size (left panels) for meshes: 32×32, 64×64, 128×128, or, as a function of the polynomial degree (right panels), degree varying from 1 to 5. (The nominal order of accuracy being the degree plus one.)

	Scheme	Intersection	Stabilization	Projection	Reconstruction	Total	Ratio vs MOOD-$\mathbb{P}_1$
Mesh 32×32	MOOD-$\mathbb{P}_1$	104.90 41.73%	0.89 0.35%	145.11 57.72%	0.50 0.20%	251.40 100.00%	—
	MOOD-$\mathbb{P}_2$	106.10 37.06%	0.97 0.34%	155.78 54.41%	23.44 8.19%	286.29 100.00%	1.15
	MOOD-$\mathbb{P}_3$	110.32 33.93%	39.33 12.10%	154.14 47.41%	21.35 6.57%	325.14 100.00%	1.29
	MOOD-$\mathbb{P}_4$	116.89 22.48%	63.43 12.20%	192.47 37.01%	147.27 28.32%	520.07 100.00%	2.07
	MOOD-$\mathbb{P}_5$	125.11 8.23%	117.10 7.71%	293.30 19.30%	984.17 64.76%	1519.69 100.00%	6.04
Mesh 64×64	MOOD-$\mathbb{P}_1$	424.61 41.87%	3.61 0.36%	583.57 57.55%	2.25 0.22%	1014.03 100.00%	—
	MOOD-$\mathbb{P}_2$	429.17 37.34%	3.70 0.32%	620.09 53.96%	96.29 8.38%	1149.26 100.00%	1.13
	MOOD-$\mathbb{P}_3$	434.95 35.70%	76.45 6.27%	618.03 50.72%	88.99 7.30%	1218.42 100.00%	1.20
	MOOD-$\mathbb{P}_4$	451.55 23.43%	134.24 6.96%	749.02 38.86%	592.55 30.74%	1927.35 100.00%	1.90
	MOOD-$\mathbb{P}_5$	476.69 7.69%	282.17 4.55%	1086.65 17.53%	4353.45 70.23%	6198.97 100.00%	6.11
Mesh 128×128	MOOD-$\mathbb{P}_1$	1685.11 41.71%	14.84 0.37%	2329.49 57.66%	10.73 0.27%	4040.18 100.00%	—
	MOOD-$\mathbb{P}_2$	1732.20 37.41%	15.37 0.33%	2506.83 54.14%	375.50 8.11%	4629.91 100.00%	1.15
	MOOD-$\mathbb{P}_3$	1732.94 36.59%	187.87 3.97%	2470.87 52.18%	343.80 7.26%	4735.48 100.00%	1.17
	MOOD-$\mathbb{P}_4$	1779.98 22.87%	394.34 5.07%	3011.18 38.69%	2598.17 33.38%	7783.67 100.00%	1.93
	MOOD-$\mathbb{P}_5$	1849.96 7.55%	694.41 2.84%	4158.08 16.98%	17785.34 72.63%	24487.78 100.00%	6.06

Table 5: Cost of MOOD remapping schemes for the cyclic remapping of the hydrodynamics like profiles with $N \times N$ quad mesh for $N = 32, 64, 128$. Last colum corresponds to the ratio of the total cost between MOOD-$\mathbb{P}_k$ and MOOD-$\mathbb{P}_1$.

5. Conclusion

In this work we have developed a formally high accurate 2D remapping scheme based on polynomial reconstructions of the variables to remap. The stabilization strategy is based on an *a posteriori* detection procedure of bad cells and subsequent polynomial degree decrementing of the reconstructions. Those bad cells are *de facto* iteratively re-updated with a lower accurate scheme. Such a remap scheme is not exclusively but devoted to become the third stages of an indirect ALE scheme based on Lagrangian, Rezone and Remap stages.

More precisely this work has presented a scheme which conservatively remaps a set of cell-centered variables from one original/old mesh onto a target/new mesh, both being possibly uncorrelated. High accuracy is obtained by means of unlimited polynomial reconstructions. On smooth problems, the remapping scheme using such reconstructions maintains the optimal order of accuracy by construction. On the other hand, stability is obtained by means of a *a posteriori* MOOD loop: Given a set of validity criteria and a candidate remapped solution, the MOOD loop starts by detecting problematic cells which are flagged and subsequently re-updated using lower accurate polynomial reconstructions in the vicinity of the bad cell. Only problematic cells and few neighbors in direct interaction must be recomputed. Valid cells are accepted and are not re-updated during the MOOD loop. Within this loop a so-called 'cascade' of schemes are successively employed. In this work we have mainly considered a cascade of four schemes: the unlimited 6th order accurate scheme with $\mathbb{P}_5$ reconstructions, then the unlimited 4th order accurate scheme with $\mathbb{P}_3$ reconstructions, the 2nd order accurate scheme with $\mathbb{P}_1$ reconstructions possibly limited and finally a first order accurate scheme. Note that the choice of the cascade is left to the user, and, any other cascade would work equivalently provided that the 1st order accurate scheme is kept as the last resort scheme. The local solution provided by this last resort scheme must always be accepted, hence the name 'parachute scheme' in the MOOD phraseology.

The detection phase drives the quality of the remapped solution as it is responsible for detecting parasitical oscillations (Gibbs effect) and, also, small scale effects which can not be captured with the characteristics mesh size employed. In this work we have considered either scalar remap for which only a maximum principle applies, or vectorial remap from hydrodynamics-like data (density, momentum, total energy) for which physical properties apply such as positivity of density, specific internal energy (and pressure depending on the equation of state). These are Physically Admissible Detection (PAD) criteria which we test for any candidate solution, if a cell fails then it is marked as problematic and sent back for re-computation. To detect spurious numerical oscillations one relies on some sort of relaxed discrete maximum principle and this constitutes the Numerical Admissible Detection (NAD) criteria [19, 29, 30, 70, 42].
Reconstruction and numerical detection phases are performed on the variables which are remapped, in our work these are the conservative variables. On the other hand the PAD criteria are conducted on the physical variables of interest. This is not a limitation of the MOOD method, rather these are developer choices for the most general case[8]

We have shown on pure remapping test cases that the high accurate remap scheme with *a posteriori* stabilization

- attains the nominal order of accuracy on smooth profiles;
- presents an essentially non-oscillatory (ENO) behavior on discontinuous profiles;
- improves the numerical accuracy when the nominal order of accuracy increases still maintaining the ENO behavior;

[8]Note that this could be easily modified if numerical evidences show that other choices are producing better results. In this work we only focus on the proof of feasibility in the most general case. All practitioners, within their constrained development contexts, have their own preferred choices, reconstructions on primitive or characteristics variables, etc. MOOD paradigm will simply adapt to these choices.

- on a smooth Gaussian scalar remap test the ratio of the L_1 error between a 2nd and 6th order accurate scheme is at minima 184 and maxima 5 millions. The $32 \times 32 = 1024$ error for MOOD-$\mathbb{P}_5$ scheme is better than the $512 \times 512 \simeq 262000$ error for MOOD-$\mathbb{P}_1$ meaning that the second order accurate scheme demands in 2D about 256 times more cells than the 6th order accurate scheme.
- the number of bad cells that need re-computation is low and they are appropriately located close to the discontinuities or steep gradients;
- the cost of a MOOD-$\mathbb{P}_3$ scheme is about 1.3 times more important than the cost of a second order accurate scheme (6 times for a MOOD-$\mathbb{P}_5$ scheme);
- compares favorably with a classical 2nd order remapper using a slope limiter (MUSCL like).

For the last point the authors are aware that many 2nd order remap schemes exist employing different slope limiters, different machinery for interleaved variables (primitive, conservative, mixing, etc.), different ways of handling mesh intersection (swept based, directional splitting, etc.). Our 2nd order accurate scheme is just one representative of the well used family of remap schemes using *a priori* limiting. This family may present some defects such as un-attained nominal 2*nd* accuracy on smooth profiles, no physics based limiting leading to loss of positivity (or conservation), and, difficulty to extend the limiting procedure to reconstructions beyond 2nd order. This work has presented one solution to these defects.

In the next future we plan to develop the extension of this approach to 3D on general polyhedral meshes. We expect to show that, in this even more complex setting, the use of high accurate reconstructions along with exact mesh intersection may improve the general behaviors of a remap scheme. The polyhedral/polyhedral mesh intersection is described in [102, 28] and our first results of *a posteriori* MOOD like stabilization in 3D are promising.
In parallel we will focus on coupling this remapper with a 2D and 3D second order accurate cell-centered to obtain indirect ALE simulation codes for hydrodynamics equations. Last but not least is the parallelization of the while machinery which will become crucial in 3D.

Acknowledgments

The authors have been partly supported by the ANR under Projet ANR-JS01-012-01 “ALE INC(ubator) 3D”. This work was performed using HPC resources from CALMIP (Grant 2015-1542). R.L. would like to warmly thanks many collaborators/researchers with whom he has exchanged a lot on this subject: C. Aymard, J.P. Braeunig, J. Breil, A. Claisse, S. Clain, S. Del Pino, C. Fochesato, P. Hoch, R. Liska, R. Motte, J.P. Perlat, M. Peybernes, R. Poncet, M. Shashkov, P. Váchal.
Special thanks to P.H. Maire for extremely fruitful discussions and to S. Diot, M. Kucharík who were at the origin of the 1D tests.

References

[1] R. Abgrall. On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 144:45–58, 1994. (Cited page 41.)
[2] D. Balsara, C. Altmann, C.D. Munz, and M. Dumbser. A sub-cell based indicator for troubled zones in RKDG schemes and a novel class of hybrid RKDG+HWENO schemes. *Journal of Computational Physics*, 226:586–620, 2007. (Cited page 41.)
[3] D. Balsara and C.W. Shu. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000. (Cited page 41.)
[4] D.S. Balsara. Self–adjusting, positivity preserving high order schemes for hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231:7504–7517, 2012. (Cited page 41.)
[5] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In D. Kroner, M. Ohlberger, and C. Rohde, editors, *An introduction to Recent Developments in Theory and Numerics for Conservation Laws, Proceedings of the International School on Theory and Numerics for Conservation Laws*, pages 195–284, Berlin, 1997. Lecture Notes in Computational Science and Engineering, Springer. (Cited page 16 et 40.)

[6] T. J. Barth. Numerical methods for conservation laws on structured and unstructured meshes. Technical report, VKI Lecture Series, 2003. (Cited page 16 et 40.)
[7] T. J. Barth and D. C. Jespersen. The design and application of upwind schemes on unstructured meshes. In *AIAA paper 89-0366*, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989. (Cited page 41.)
[8] Timothy J Barth and Dennis C Jespersen. The design and application of upwind schemes on unstructured meshes. *AIAA Paper*, 89(89-0366):1–12, 1989. (Cited page 16 et 40.)
[9] T.J. Barth. Aspects of unstructured grids and finite-volume solvers for Euler and Navier-Stokes equations. *VKI/NASA/AGARD Special Course on Unstructured Grid Methods for Advection Dominated Flows AGARD Publication R-787*, 1995. (Cited page 16 et 40.)
[10] T.J. Barth and P.O. Frederickson. Higher order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA paper no. 90-0013*, 28th Aerospace Sciences Meeting January 1990. (Cited page 41.)
[11] A. L. Bauer, D. E. Burton, E. J. Caramana, R. Loubère, M. J. Shashkov, and P. P. Whalen. The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics. *J. Comput. Phys.*, 218(2):572–593, 2006. (Cited page 2.)
[12] D. J. Benson. An efficient, accurate, simple ALE method for nonlinear finite element programs. *Comp. Meth. Appl. Mech. Engrg.*, 72:305–350, 1989. (Cited page 1.)
[13] Jay P. Boris and David L. Book. Flux-corrected transport. *J. Comput. Phys.*, 135(2):172–186, August 1997. (Cited page 7.)
[14] Walter Boscheri, Raphaël Loubère, and Michael Dumbser. Direct arbitrary-lagrangianeulerian ader-mood finite volume schemes for multidimensional hyperbolic conservation laws. *Journal of Computational Physics*, 292(0):56 – 87, 2015. (Cited page 6.)
[15] A. Burbeau, P. Sagaut, and Ch. H. Bruneau. A problem-independent limiter for high-order runge—kutta discontinuous galerkin methods. *J. Comput. Phys.*, 169(1):111–150, May 2001. (Cited page 41.)
[16] E. J. Caramana, D. E. Burton, M. J. Shashkov, and P. P. Whalen. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J. Comput. Phys.*, 146(1):227–262, 1998. (Cited page 2.)
[17] E. J. Caramana, C. L. Rousculp, and D. E. Burton. A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry. *J. Comput. Phys.*, 157:89–119, 2000. (Cited page 2.)
[18] E. J. Caramana, M. J. Shashkov, and P. P. Whalen. Formulations of artificial viscosity for multidimensional shock wave computations. *J. Comput. Phys.*, 144:70–97, 1998. (Cited page 2.)
[19] S. Clain, S. Diot, and R. Loubère. A high-order finite volume method for hyperbolic systems: Multi-dimensional Optimal Order Detection (MOOD). *J. Comput. Phys.*, 230(10):4028–4050, 10 May 2011. (Cited page 1, 2, 4, 6, 8, 12 et 34.)
[20] B. Cockburn, C. Johnson, C.W. Shu, and E. Tadmor. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In A. Quarteroni, editor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, volume 1697 of *Lecture Notes in Mathematics*, pages 325–432. Springer, 1998. (Cited page 2.)
[21] B. Cockburn, G. E. Karniadakis, and C.W. Shu. *Discontinuous Galerkin Methods*. Lecture Notes in Computational Science and Engineering. Springer, 2000. (Cited page 41.)
[22] B. Cockburn and C. W. Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. *Journal of Scientific Computing*, 16:173–261, 2001. (Cited page 41.)
[23] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM J. Stat. Comput.*, 6(1):104–117, 1985. (Cited page 41.)
[24] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1):171 – 200, 1990. (Cited page 41.)
[25] P. Colella and M.D. Sekora. A limiter for PPM that preserves accuracy at smooth extrema. *Journal of Computational Physics*, 227:7069–7076, 2008. (Cited page 41.)
[26] B. Després and R. Loubère. Convergence and sensitivity analysis of repair algorithms in 1D. *Int. J. Finit. Vol.*, 3(1), 19 January 2006. (Cited page 7.)
[27] B. Després and C. Mazeran. Lagrangian gas dynamics in 2d and lagrangian systems. *Arch. Rat. Mech. Anal.*, 178:327–372, 2005. (Cited page 2.)
[28] Tom Abel Devon Powell. An exact general remeshing scheme applied to physically conservative voxelization. *J. Comput. Phys.*, pages 340–356, 2015. (Cited page 10 et 35.)
[29] 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. *Computer and Fluids*, 64:43–63, 2012. http://dx.doi.org/10.1016/j.compfluid.2012.05.004. (Cited page 1, 2, 4, 6, 8, 12 et 34.)
[30] S. Diot, R. Loubère, and S. Clain. The multidimensional optimal order detection method in the three-dimensional case: Very high-order finite volume method for hyperbolic systems. *International Journal for Numerical Methods in Fluids*, 73(4):362–392, 2013. cited By 5. (Cited page 1, 2, 6, 8, 12 et 34.)
[31] D.Kuzmin. Slope limiting for discontinuous galerkin approximations with a possibly non-orthogonal taylor basis. *International Journal for Numerical Methods in Fluids*, 71(9):1178–1190, 2013. (Cited page 41.)
[32] D.Kuzmin. Hierarchical slope limiting in explicit and implicit discontinuous galerkin methods. *Journal of Computational Physics*, 257, Part B(0):1140 – 1162, 2014. Physics-compatible numerical methods. (Cited page 41.)
[33] Q. Du, M. Emelianenko, and M. Gunzburger. Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations. *SAIM J. Numer. Anal.*, 44(1):102–119, 2006. (Cited page 15.)
[34] Q. Du, V. Faber, and M. Gunzburger. Centroidal Voronoi tesselations: applications and algorithms. *SIAM Review*, 41:637–676, 1999. (Cited page 15.)

[35] Q. Du and M. Gunzburger. Grid generation and optimization based on centroidal Voronoi tesselations. *Applied Mathematics and Computations*, 133:591–607, 2002. (Cited page 15.)
[36] Q. Du and D. Wang. Anisotropic centroidal Voronoi tesselations and their applications. *SIAM J. Sci. Comput.*, 26(3):737–761, 2005. (Cited page 15.)
[37] J.K. Dukowicz and J. Baumgardner. Incremental remapping as a transport/advection algorithm. *J. Comput. Phys.*, 160:318–335, 2000. (Cited page 2.)
[38] J.K. Dukowicz and J.W. Kodis. Accurate conservative remapping (rezoning) for Arbitrary-Lagrangian-Eulerian computations. *SIAM J. Stat. Comput.*, 8:305–321, 1987. (Cited page 2.)
[39] M. Dumbser. Arbitrary high order pnpm schemes on unstructured meshes for the compressible navier-stokes equations. *Computers and Fluids*, 39(1):60–76, 2010. cited By 72. (Cited page 7.)
[40] M. Dumbser, D. Balsara, E.F. Toro, and C.D. Munz. A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes. *Journal of Computational Physics*, 227:8209–8253, 2008. (Cited page 41.)
[41] M. Dumbser, D. Balsara, E.F. Toro, and C.D. Munz. A unified framework for the construction of one-step finite volume and discontinuous galerkin schemes on unstructured meshes. *Journal of Computational Physics*, 227:8209–8253, 2008. (Cited page 2.)
[42] M. Dumbser, O. Zanotti, R. Loubère, and S. Diot. A posteriori subcell limiting of the discontinuous galerkin finite element method for hyperbolic conservation laws. *Journal of Computational Physics*, 278:47–75, 2014. cited By 0. (Cited page 4, 6 et 34.)
[43] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *J. Comput. Phys.*, 144:194–212, 1998. (Cited page 8 et 41.)
[44] G. Gassner, M. Dumbser, F. Hindenlang, and C.D. Munz. Explicit one–step time discretizations for discontinuous Galerkin and finite volume schemes based on local predictors. *Journal of Computational Physics*, 230:4232–4247, 2011. (Cited page 41.)
[45] C.F. Ollivier Gooch. High-order ENO schemes for unstructured meshes based on least-squares reconstruction. Technical report, Argonne National Laboratory, 1996. Mathematics and Computer Science Division. (Cited page 2, 7 et 41.)
[46] S. Gottlieb and C.-W. Shu. Total variation diminishing rungekutta schemes. *Math. Comput.*, 67:73–85, 1998. (Cited page 41.)
[47] A. Harten. ENO scheme with subcell resolution. *J. Comput. Phys.*, 83:148, 1989. (Cited page 41.)
[48] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987. (Cited page 41.)
[49] A. Harten, B. Engquist, S. Osher, and S.R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes III. *Journal of Computational Physics*, 71:231–303, 1987. (Cited page 2.)
[50] A. Harten and S. Osher. Uniformly highorder accurate nonoscillatory schemes i. *SIAM J. Num. Anal.*, 24:279–309, 1987. (Cited page 2.)
[51] C.W. Hirt, A.A. Amsden, and J.L. Cook. An arbitrary Lagrangian-Eulerian computing method for all flow speeds. *J. Comput. Phys.*, 14:277–253, 1974. reprinted in Journal of Computational Physics, **135** (1997), pp. 203–216. (Cited page 1.)
[52] P. Hoch. An arbitrary lagrangian-eulerian strategy to solve compressible fluid flows. HAL, http://hal.archives-ouvertes.fr/hal-00366858, 2009. private communication. (Cited page 7.)
[53] H. Hoteit, Ph. Ackerer, R. Mos, J. Erhel, and B. Philippe. New two-dimensional slope limiters for discontinuous galerkin methods on arbitrary meshes. *International Journal for Numerical Methods in Engineering*, 61(14):2566–2593, 2004. (Cited page 41.)
[54] X.Y. Hu, N.A. Adams, and C.W. Shu. Positivity–preserving method for high–order conservative schemes solving compressible Euler equations. *Journal of Computational Physics*, 242:169–180, 2013. (Cited page 41.)
[55] Y. Huang, H. Quin, and D. Wang. Centroidal Voronoi tesselation-based finite element superconvergence. *Int. J. Numer. Meth. Engng.*, 76:1819–1839, 2008. (Cited page 15.)
[56] G. S. Jiang and C. W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996. (Cited page 7 et 41.)
[57] G.-S. Jiang and C.W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1996. (Cited page 2.)
[58] C.W. Shu J.Zhu, X.Zhong and J. Qiu. Rungekutta discontinuous galerkin method using a new type of weno limiters on unstructured meshes. *J. Comp. Phys.*, 248:200–220, 2013. (Cited page 41.)
[59] 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. (Cited page 41.)
[60] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, and J. E. Flaherty. Shock detection and limiting with discontinuous galerkin methods for hyperbolic conservation laws. *Appl. Numer. Math.*, 48(3-4):323–338, March 2004. (Cited page 41.)
[61] Lilia Krivodonova. Limiters for high-order discontinuous galerkin methods. *Journal of Computational Physics*, 226(1):879 – 896, 2007. (Cited page 41.)
[62] Milan Kucharík and Mikhail Shashkov. One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian-Eulerian methods. *Journal of Computational Physics*, 231(7):2851 – 2864, 2012. (Cited page 2.)
[63] Milan Kucharik, Mikhail Shashkov, and Burton Wendroff. An efficient linearity-and-bound-preserving remapping method. *J. Comput. Phys.*, 188(2):462–471, July 2003. (Cited page 4, 5 et 24.)
[64] M. Kuchavrík and M. Shashkov. Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity. *International Journal for Numerical Methods in Fluids*, 56(8):1359–1365, 2007. (Cited page 2 et 24.)

[65] D. Kuzmin. On the design of general-purpose flux limiters for finite element schemes. I. Scalar convection. *J. Comput. Phys.*, 219:513–531, December 2006. (Cited page 7 et 41.)

[66] D. Kuzmin, R. Löhner, and S. Turek. *Flux–Corrected Transport*. Scientific Computation. Springer, 2005. 3-540-23730-5. (Cited page 7.)

[67] Dmitri Kuzmin. A vertex-based hierarchical slope limiter for -adaptive discontinuous galerkin methods. *Journal of Computational and Applied Mathematics*, 233(12):3077 – 3085, 2010. Finite Element Methods in Engineering and Science (FEMTEC 2009). (Cited page 41.)

[68] Richard Liska, Mikhail Yu. Shashkov, Pavel Vchal, and Burton Wendroff. Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary lagrangian-eulerian methods. *J. Comput. Physics*, 229(5):1467–1497, 2010. (Cited page 24.)

[69] X.D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational physics*, 115:200–212, 1994. (Cited page 2.)

[70] 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. *Communications in Computational Physics*, 16(3):718–763, 2014. cited By 3. (Cited page 1, 2, 6, 8, 12 et 34.)

[71] R. Loubère, P.-H Maire, M. Shashkov, J. Breil, and S. Galera. ReALE: A Reconnection-based Arbitrary-Lagrangian-Eulerian method. *J. Comput. Phys.*, 229:4724–4761, 2010. (Cited page 15.)

[72] R. Loubère, P.-H. Maire, and P. Váchal. Formulation of a staggered two-dimensional Lagrangian scheme by means of cell-centered approximate riemann solver. In Maya Neytcheva Editors Gunilla Kreiss Per Lötstedt, Axel Malqvist, editor, *Proceedings of ENUMATH 2009, the 8th European Conference on Numerical Mathematics and Advanced Applications, Uppsala, Sweden*. Springer, 2009. (Cited page 40.)

[73] R. Loubère, P.-H. Maire, and P. Váchal. Staggered Lagrangian discretization based on cell-centered riemann solver and associated hydro-dynamics scheme. *Communication in Computational Physics*, 10(4):940–978, 2011. doi:10.4208/cicp.170310.251110a. (Cited page 5 et 40.)

[74] R. Loubère and M. Shashkov. A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian-Eulerian methods. *J. Comput. Phys.*, 209(1):105–138, 2006. (Cited page 4.)

[75] R. Loubère, M. Staley, and B.Wendroff. The repair paradigm: New algorithms and applications to compressible flow. *J. Comput. Phys.*, 1(2):385–404, 2006. (Cited page 7.)

[76] H. Luo, J.D. Baum, and R. Lhner. A discontinuous galerkin method based on a taylor basis for the compressible flows on arbitrary grids. *J. Comput. Phys.*, 227:8875–8893, 2008. (Cited page 41.)

[77] H. Luo, L. Luo, R. Nourgaliev, V.A. Mousseau, and N. Dinh. A reconstructed discontinuous Galerkin method for the compressible NavierStokes equations on arbitrary grids. *Journal of Computational Physics*, 229:6961–6978, 2010. (Cited page 41.)

[78] H. Luo, Y. Xia, S. Spiegel, R. Nourgaliev, and Z. Jiang. A reconstructed discontinuous Galerkin method based on a Hierarchical WENO reconstruction for compressible flows on tetrahedral grids . *Journal of Computational Physics*, 236:477–492, 2013. (Cited page 41.)

[79] G. Luttwak and J. Falcovitz. Slope limiting for vectors: a novel vector limiting algorithm. Conference on Numerical methods for multi-material fluid flows; Pavia University on September 21 - 25, 2009. Available at `http://www.eucentre.it/multimat09/media/presentazioni_congresso/luttwak.pdf`. (Cited page 40.)

[80] G. Luttwak and J. Falcovitz. Slope limiting for vectors: A novel vector limiting algorithm. *Int. J. Numer. Meth. Fluid*, 65:1365–1375, 2011. doi: 10.1002/fld.2367. (Cited page 5 et 40.)

[81] G. Luttwak and J. Falcovitz. Vip (vector image polygon) multi-dimensional slope limiters for scalar variables. *Computers and Fluids*, 83:90–97, 2013. (Cited page 40.)

[82] R. Lhner, K. Morgan, J. Peraire, and M. Vahdati. Finite element flux-corrected transport (femfct) for the euler and navierstokes equations. *Int. J. Numer. Methods Fluids*, pages 1093–1109, 1987. (Cited page 41.)

[83] S. M. Murman M. Berger, M.J. Aftosmis. Analysis of slope limiters on irregular grids. Technical Report NAS-05-007, NAS Technical Report, 2005. (Cited page 41.)

[84] P.-H. Maire, R. Abgrall, J. Breil, and J. Ovadia. A cell-centered Lagrangian scheme for two-dimensional compressible flow problems. *SIAM J. Sci. Comput.*, 29(4):1781–1824, 2007. (Cited page 2.)

[85] P.-H. Maire, J. Breil, and S. Galera. A cell-centred arbitrary Lagrangian-Eulerian (ALE) method. *Int. J. Numer. Meth. Fluid*, 56(8):1161–1166, 2008. (Cited page 2.)

[86] L.G. Margolin. Introduction to "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds". *J. Comput. Phys.*, 135:198–202, 1997. (Cited page 1.)

[87] L.G. Margolin and M. Shashkov. Second-order sign-preserving conservative interpolation (remapping) on general grids. *J. Comput. Phys.*, 184(1):266–298, 2003. (Cited page 5 et 24.)

[88] L.G Margolin and M. Shashkov. Remapping, recovery and repair on staggered grid. *Computer Methods in Applied Mechanics and Engineering*, 193:4139–4155, 2004. (Cited page 7 et 24.)

[89] K. Michalak and C. Ollivier-Gooch. Limiters for unstructured higher-order accurate solutions of the euler equations. In *Proceedings of the AIAA Forty-Sixth Aerospace Sciences Meeting*, 2008. (Cited page 41.)

[90] A. Okabe, B. Boots, K. Sugihara, and S.N. Chiu. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*. Chichester: John Wiley & Sons, 2nd edition, 2000. (Cited page 15.)

[91] J. Qiu, M. Dumbser, and C.W. Shu. The discontinuous Galerkin method with Lax-Wendroff type time discretizations. *Computer Methods in Applied Mechanics and Engineering*, 194:4528–4543, 2005. (Cited page 41.)

[92] J. Qiu and C.W. Shu. Finite difference WENO schemes with Lax-Wendroff type time discretization. *SIAM J. Sci. Comput.*, 24(6):2185–2198, 2003. (Cited page 2.)
[93] J. Qiu and C.W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case. *Journal of Computational Physics*, 193:115–135, 2003. (Cited page 41.)
[94] J. Qiu and C.W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case. *Computers and Fluids*, 34:642–663, 2005. (Cited page 41.)
[95] J. Qiu and C.W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing*, 26:907–929, 2005. (Cited page 41.)
[96] J. Reisner, J. Serencsa, and S. Shkoller. A space-time smooth artificial viscosity method for nonlinear conservation laws. *Journal of Computational Physics*, 235(0):912 – 933, 2013. (Cited page 41.)
[97] W.J. Rider and L. G. Margolin. Simple modifications of monotonicity-preserving limiter. *Journal of Computational Physics*, 174(1):473 – 488, 2001. (Cited page 41.)
[98] M. Shashkov and B. Wendroff. The repair paradigm and application to conservation laws. *J. Comput. Phys*, 198:265–277, 2004. (Cited page 7.)
[99] M.Yu. Shashkov and A.V. Solovjov. A generalization of the notion of dirichlet cell fo non- convex domain. *Keldysh Institute of Applied Mathematics USSR Academy of Sciences, Moscow, Russia*, Preprint No. 32, 1990. in Russian. (Cited page 15.)
[100] M.Yu. Shashkov and A.V. Solovjov. Numerical simulation of two-dimensional flows by the free-lagrangian flow simulations. *Mathematisches Institut, Technische Universität, Munchen*, Report TUM-M9105, 1991. available at http://cnls.lanl.gov/shashkov. (Cited page 15.)
[101] T. Sonar. On the construction of essentially non-oscillatory finite volume approximations to hyperbolic conservation laws on general triangulations: polynomial recovery, accuracy and stencil selection. *Computer methods in applied mechanics and engineering*, 140:157–181, 1997. (Cited page 41.)
[102] K. Sugihara. A robust and consistent algorithm for intersecting convex polyhedra. *Comput. Graph. Forum*, 3:45–54, 1994. (Cited page 10 et 35.)
[103] V.A. Titarev, P. Tsoutsanis, and D. Drikakis. WENO schemes for mixed–element unstructured meshes. *Communications in Computational Physics*, 8:585–609, 2010. (Cited page 41.)
[104] P. Tsoutsanis, V.A. Titarev, and D. Drikakis. WENO schemes on arbitrary mixed-element unstructured meshes in three space dimensions. *Journal of Computational Physics*, 230:1585–1601, 2011. (Cited page 41.)
[105] S. Tu and S. Aliabadi. A slope limiting procedure in discontinuous galerkin finite element method for gasdynamics applications. *Int. J. Numer. Anal. Model.*, 2:163–178, 2005. (Cited page 41.)
[106] P. Vachal, R. Liska, M. Shashkov, and B. Wendroff. Synchronized flux-corrected remapping for ale methods. *Computers & Fluids*, 46(1):312–317, 2011. (Cited page 24.)
[107] B. van Leer. Towards the ultimate conservative difference scheme. III - Upstream-centered finite-difference schemes for ideal compressible flow. IV - A new approach to numerical convection. *J. Comput. Phys.*, 23:263–299, March 1977. (Cited page 7.)
[108] J. Velechovský, M. Kuchavrík, R. Liska, M. Shashkov, and P. Váchal. Symmetry- and essentially-bound-preserving flux-corrected remapping of momentum in staggered ale hydrodynamics. *J. Comput. Phys.*, 255:590–611, December 2013. (Cited page 24.)
[109] V. Venkatakrishnan. Convergence to steady state solutions of the Euler equations on unstructured grids with limiters. *J. Comput. Phys.*, 118:120–130, 1995. (Cited page 41.)
[110] Y. Xing, X. Zhang, and C.W. Shu. Positivity preserving high order well balanced discontinuous Galerkin methods for the shallow water equations. *Advances in Water Resources*, 33:1476–1493, 2010. (Cited page 41.)
[111] M. Yang and Z.J. Wang. A parameter-free generalized moment limiter for high-order methods on unstrucured grids. *Adv. Appl. Math. Mech.*, 4:451–480, 2009. (Cited page 41.)
[112] X. Zhang and C.W. Shu. On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes. *Journal of Computational Physics*, 229:8918–8934, 2010. (Cited page 41.)
[113] X. Zhang and C.W. Shu. Maximum-principle-satisfying and positivity-preserving high order schemes for conservation laws: Survey and new developments. *Proceedings of the Royal Society A*, 467:2752–2776, 2011. (Cited page 41.)
[114] X. Zhang and C.W. Shu. Positivity–preserving high order finite difference WENO schemes for compressible Euler equations . *Journal of Computational Physics*, 231:2245–2258, 2012. (Cited page 41.)
[115] J. Zhu and J. Qiu. Runge-kutta discontinuous galerkin method using weno type limiters: Three dimensional unstructured meshes. *Commun.Comput. Phys.*, pages 985–1005, 2012. (Cited page 41.)

Appendix: *a priori vs a posteriori*

Let us illustrate some flaws that inexorably appear when classical *a priori* slope limiters based on the strict enforcement of the Discrete Maximum Principle are to be used with piecewise linear reconstructions.

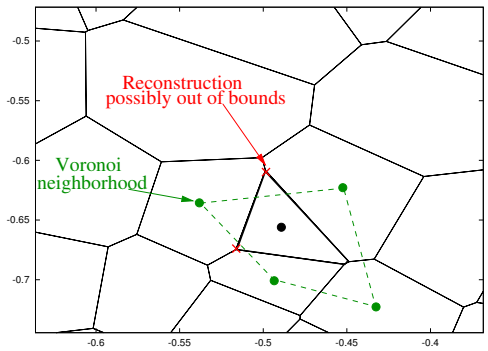


Figure 20: Zoom on a polygonal mesh generating inappropriate geometrical shaped cells (thick black cell) which may still have out-of-bounds vertex reconstructed values (at red points) even if a slope limiter considering the Voronoi neighborhood (green bullets) is employed.

Classical a priori slope limiter for piecewise linear reconstruction

In this work we have implemented a classical vertex-based Barth-Jespesen slope limiter[9] [8, 9, 5, 6]. The vertex evaluated values of the piecewise linear reconstruction must lay in-bounds defined by the local minimal/maximal mean values of the neighborhood. The limiter Φ_i is then applied to each reconstructed variable $\mathcal{U}$ as

$$\mathcal{U}_i(\boldsymbol{X}) = \mathcal{U}_i \; + \Phi_i \sum_{1\leq|\alpha|\leq d} \mathcal{R}_i^{\alpha} \left((\boldsymbol{X} - \boldsymbol{X}_i)^{\alpha} - \frac{1}{|\Omega_i|} \int_{\Omega_i} (\boldsymbol{X} - \boldsymbol{X}_i)^{\alpha} \, d\boldsymbol{X} \right) . \tag{23}$$

When a system of PDEs is considered there may be some in-adequation between the choice of the variables to reconstruct and the choice of variables that must remain in bounds. For instance we may want to reconstruct and project the momentum, but, only keep the velocity in-bounds, see other situations in table 1.

The quality of the underlying mesh may be important to ensure the in-bounds property even for scalar variable. For instance in figure 20 the strict application of a slope limiter does not ensure that the red vertices $\boldsymbol{X}_p$ have reconstructed values $\mathcal{U}_i(\boldsymbol{X}_p)$ which are in-bounds considering the Voronoi neighborhood (green dots). This is due to the local "bad" quality of the mesh. However when dealing with polygons, or worse with polyhedra, such situation is no more exceptional. Unless some preprocessing of the mesh is performed, the slope limiting may still fail in ensuring the in-bounds property.

When a vectorial or tensorial variable is considered, the choice of a component-by-component limiting may lead to non-frame-invariant results. This may be problematic for mesh moving techniques such as Lagrangian schemes [73, 72, 79, 80, 81]. More advanced limiters have then to be developed to cure this flaw which is usually not present for unlimited reconstructions.

[9]This choice is only meant to pick one of the classical *a priori* slope limiter. The same general conclusion would apply to most of the *a priori* limiters should they be applied to conservative, primitive or characteristics variables. Indeed the *a priori* character of these slope limiters is the characteristics we wish to emphasize. Of course there exist better choices than others depending on the system of PDEs solved.

High order slope limiter extensions

Many attempts of extending the 'slope limiter' approach to high order polynomials have been tried both in high order finite volume context under key word '(W)ENO' [48, 47, 1, 3, 45, 56, 43, 58, 115, 95, 93, 94, 91, 101, 2, 40, 103, 104] for which positivity preserving may demand some care [114, 4, 54] or 'hierachical slope limiting' or 'artificial viscosity' [59, 82, 24, 23, 25, 65, 10, 97, 67, 15, 61, 60, 53, 111, 83, 105, 7, 89, 109, 76, 46, 31, 32, 96] or even for 'discontinuous Galerkin' limiting strategies [22, 21, 77, 78, 44, 112, 110, 113]. Being exhautive for this matter is simply impossible, consequently the reader is guided to the references included in the cited works for a broaded view.

In summary

- the paradigms onto which are constructed slope limiter are not trivially extended to higher order accurate reconstructions;
- some slope limiters may demand that the mesh verifies geometrical constraints, otherwize it may activate inappropriately, see the example on figure 20;
- generally the bound preservation of the reconstructions within a cell does not imply the subsequent bound preservation of the remapped solution, for instance when system of interleaved variables are remapped.

While the second issue could be mitigated, as instance by a careful investigation of the mesh, the other two simply can not. Contrarily with our *a posteriori* approach we deal with those issues by construction, as the strategy does not assume any mesh quality, neither does it assume any form onto which the limiting must act, neither does it postpone the question of bound preservation of the remapped solution.

Delaunay triangular grids. Let us construct two Delaunay triangular grids made of about $N = XXX$ cells to project back and forth the sine profile from table 2, see figure **??** for the mesh. The Delaunay meshes have a good geometrical quality and the number of cells are far beyond the minimal number required to expect a good remap solution. MOOD, MUSCL and obvisouly UNLIM schemes are able to exactly reproduce linear field (not shown in the table) as expected. Nonetheless the results for the smooth function shows that, although MOOD performs as the unlimited scheme, that is not detecting a single bad cell hence leading to a almost perfect second order accurate scheme, MUSCL remapping has more difficulties to reach these errors and orders of convergence in L^∞ norm. This was expected because it is well known that the strict *a priori* application of the DMP clips extrema.

Polygonal grids. A Voronoi polygonal grid is constructed given a set of $N = XXX$ so-called Voronoi generators which are almost uniformly distributed. As such the mesh seems, at least visually, acceptable, see figure **??**. A second polygonal mesh is constructed by rotating the set of generators. The two meshes are therefore of same geometrical quality however they do not verify the fact that the current cell onto which the reconstructions are made, is not entirely covered by the convex hull constructed from the neighbor cell centers (green cells) as illustrated in figure 20. As such the value of the reconstruction at some vertexes may be out-of-bounds (see the red crosses in figure 20). If the underlying field is linear and if the piecewise constant data are exact mean values then a limiter should never be activated. However with such a 'bad quality' mesh, the limiter is acting on several cells leading to a loss of accuracy[10]. The supposedly second-order accurate MUSCL scheme is in this case not able to exactly remap a linear field. In table 6 are reported the errors and orders of convergence when the sine profile if remapped. MOOD and UNLIM are almost equivalent (as MOOD does detect almost no bad cell) whereas the *a priori* limiter in the MUSCL scheme is over-reacting for this smooth situation, leading to a net loss in accuracy and convergence.

	N_x	L^2 error		L^∞ error		N_b	L^2 error		L^∞ error		L^2 error		L^∞ error	
		MOOD $\mathbb{P}_1 \longrightarrow \mathbb{P}_0$					**MUSCL** $\mathbb{P}_1^{lim}$				**UNLIM** $\mathbb{P}_1$			
Delaunay mesh	32	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	64	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	128	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	256	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	512	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	Expected order		2		2			2		2		2		2
	N_x	L^2 error		L^∞ error		N_b	L^2 error		L^∞ error		L^2 error		L^∞ error	
		MOOD $\mathbb{P}_1 \longrightarrow \mathbb{P}_0$					**MUSCL** $\mathbb{P}_1^{lim}$				**UNLIM** $\mathbb{P}_1$			
Polygonal mesh	32	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	64	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	128	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	256	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	512	2.65E-02	—	3,33E-02	—	0	4.49E-02	—	3.24E-02	—	2.65E-02	—	3,33E-02	—
	Expected order		2		2			2		2		2		2

Table 6: L^2 and L^∞ errors and convergence rate for the sine profile on back and forth remap on triangular and polygonal grids for MOOD $\mathbb{P}_1$, MUSCL and UNLIMited schemes of the same nominal accuracy. N_b referes to the total number of bad cells detected during the entire simulation.

[10] We may argue that choosing a good quality mesh is mandatory. However in 3D this 'good quality' mesh may not be easily found, or, worse, may not even exist. Consequently numerical schemes which are marginally dependent on the mesh geometrical quality may be of great value.