The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5–9 June 2022, Oslo, Norway

VECTORIAL LIMITATION FOR MULTISLOPE MUSCL SCHEMES

ARTHUR TÉTELIN[1], CLÉMENT LE TOUZE[1] AND PHILIPPE VILLEDIEU[2]

[1] DMPE, ONERA
Université Paris Saclay
F-91123 Palaiseau - France
email: arthur.tetelin@onera.fr, clement.le_touze@onera.fr

[2] DMPE, ONERA
Université de Toulouse
F-31055 Toulouse - France
email: philippe.villedieu@onera.fr

Key words: Cell-Centered Finite-Volume Methods, Unstructured Meshes, Multislope MUSCL Schemes, Vectorial Slope Limitation, Frame Invariance

Abstract. In finite volume schemes with MUSCL interpolation of scalar variables at the faces of control volumes, a slope limiting function is used in order to prevent non-physical oscillations of the solution. More particularly, these functions are designed to ensure a certain monotonicity criterion at each face of the control volume, criterion which then ensures a stability property of the scheme. For vectorial variables, these slope limiting functions are generally applied componentwise, but this may result in a frame-dependance, as well as a loss of accuracy due to false detection of extrema. In this paper, a new vectorial interpolation method is introduced, which is frame-invariant, second-order accurate and stable in a sense that will be defined.

1 INTRODUCTION

CEDRE software is a numerical code developed by the french institute ONERA to solve multi-physics problems in energetics [1]. Like other CFD codes, it solves hyperbolic systems of conservation equations, which, by focusing only on the convective part, read:

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{f}(\mathbf{Q}, \boldsymbol{\lambda}) = \mathbf{0}, \tag{1}$$

where $\mathbf{Q}$ represents the vector of conserved variables, $\boldsymbol{\lambda}$ is the velocity vector field, and $\mathbf{f}$ the physical flux. In CEDRE, the spatial discretization relies on a finite volume cell-centered framework on general unstructured meshes [2], which allows us to write the semi-discretized formulation of equation (1) on a multi-dimensional domain:

$$\frac{\partial \mathbf{Q}_i}{\partial t} = -\frac{1}{|K_i|} \sum_{j \in \mathcal{V}(i)} |S_{ij}| \Phi \left(\mathbf{Q}_{ij}^n, \mathbf{Q}_{ji}^n, \mathbf{n}_{ij}, \boldsymbol{\lambda}_{ij} \right), \tag{2}$$

where K_i denotes a cell of the mesh, S_{ij} the face between cells K_i and K_j. Notation $|.|$ denote either the length, area or volume measure, depending on the dimension of the object. $\mathcal{V}(i)$ is the face neighborhood of the cell K_i, Φ is the numerical flux and $\mathbf{n}_{ij}$ is a face normal vector pointing from cell K_i to cell K_j. Lastly, $\mathbf{Q}_{ij}^n$ represents a set of interpolated values at the face center $\mathbf{M}_{ij}$ of face S_{ij}. With a first order scheme, these interpolated values are equal to the numerical variables $\mathbf{Q}_i$ which represents the mean value of $\mathbf{Q}$ over the cell K_i, located at the cell centroid $\mathbf{B}_i$. To reach a higher order of spatial accuracy, CEDRE code relies on a MUSCL approach, on which we will be focusing in this article.

As it is well known by Godunov's theorem [3], no linear reconstruction can be both high-order accurate and ensure the monotony of the scheme, that is why Van-Leer introduced the MUSCL approach [4–8]. This method consists in evaluating the gradient of the variable, which is then limited by a limitation function [9, 10] in order to ensure the scheme stability. For a one dimensional scheme, this stability is reached through a TVD property as introduced by Harten [11], but in higher dimension, it is known that the TVD property is incompatible with high order accuracy [12]. Since then, many other have studied this issue [13–15] in order to develop a MUSCL method on multi-dimensional meshes. These multi-dimensional MUSCL methods can be grouped in two categories: monoslope methods, in which a single limited gradient is computed for the entire cell [16], and multislope methods, in which a limited directional gradient is computed for each face [17]. The method that we will study here is of multislope type, and has been introduced by Le Touze [18].

As far as the authors know, while the limitation framework has already been well studied for scalar variables, such as temperature or pressure, limitation methods of vectorial variables such as velocity have received far less attention than their scalar counterpart. On a software such as CEDRE, vectorial limitation was obtained until now by limiting vectors componentwise, but this kind of procedure turns out to be frame-dependant, and a loss of accuracy can also be observed. For the vectorial limitation problem, one can divide solutions found in the literature in three major methods. The first one has been introduced by Luttwak and Falcovitz for a monoslope framework [19–21]. As the stability of a scalar variable is usually defined through a maximum principle, the authors defined vectorial variable stability by using the convex hull of vectors from cell neighborhood, and called it VIP (*Vector Image Polygon / Polyhedron*). A VIP method consists in the computation of a vector at the time step $n+1$, followed by its projection on the VIP set if it doesn't already lie into it. With this process, a vector will satisfy a maximum principle componentwise, *whatever basis we choose*, which is the natural extension of the scalar maximum principle to vector variables. The second method has been developped by Maire *et al.* [22, 23]. It consists in computing a local basis for vectors to be reconstructed. Then vectors are projected onto this basis, and the vector reconstruction is computed componentwise in this basis. The last method has been developped by Zeng and Scovazzi [24]. It consists in determining an axis on which the vector field will be projected in order to get scalar variables. Then, a simple scalar limiter is computed, in order to get the general vectorial limiter.

The VIP method has the advantage to present the most accurate extension of the scalar maximum principle to the vectorial case. However, its algorithmic cost seems to be important and even prohibitive on large-scale simulations, as we have to compute intersection of vectors with a convex hull for each reconstructed variables. On the other hand, projection methods seems to have a lower computation cost, but the choice of the axis or basis of projection remains arbitrary, and the vectorial aspect of the variable is dropped by these methods. For all these reasons, we propose a new vectorial limitation method in the context of multislope MUSCL methods. In section 2, we will quickly detail the multislope scalar MUSCL scheme, and we will develop a new way of limiting vectors variables in section 3. Nevertheless, we will only focus here on final results. Intermediate demonstrations and numerical tests will be presented in a forthcoming paper.

2 SCALAR MULTISLOPE MUSCL SCHEME

2.1 General reconstruction

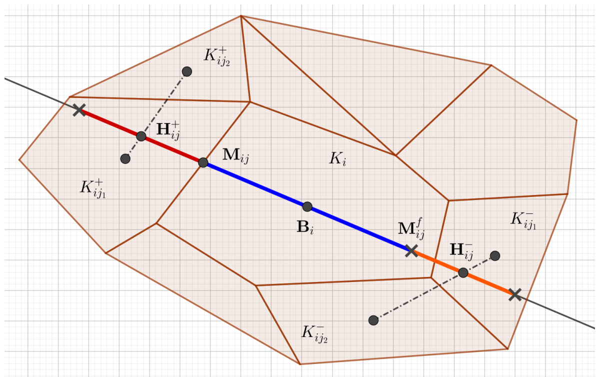

Figure 1: Multislope MUSCL method.

The numerical scheme introduced by Le Touze and al. [18] consists in the computation of various geometrical parameters for each face of each cell of the mesh. All these parameters are illustrated in Figure 1. Along the axis $\mathbf{B}_i\mathbf{M}_{ij}$, for a d-dimensional mesh, we determine d forward cells $(K^+_{ij,k})_{1\leq k\leq d}$ and d backward cells $(K^-_{ij,k})_{1\leq k\leq d}$. From the intersection of the line (or plane in dimension 3) drawn by the forward cells centroids $(\mathbf{B}^+_{ij,k})_{1\leq k\leq d}$ and the axis $\mathbf{B}_i\mathbf{M}_{ij}$, we define the forward point $\mathbf{H}^+_{ij}$. By a similar way, we define the backward point $\mathbf{H}^-_{ij}$. Then, by giving each scalar value $(U^+_{ij,k})_{1\leq k\leq d}$ of the forward cells a weight $(\beta^+_{ij,k})_{1\leq k\leq d}$, one can determine an interpolated value at the forward point: $U_{\mathbf{H}^+_{ij}} = \sum_{k=1}^d \beta^+_{ij_k} U^+_{ij_k}$. Similarly, one can write $U_{\mathbf{H}^-_{ij}} = \sum_{k=1}^d \beta^-_{ij_k} U^-_{ij_k}$. Hence, along the axis $\mathbf{B}_i\mathbf{M}_{ij}$, a scalar monodimensional framework has been created. Thus, we can define the forward and backward slopes, as well as their ratio:

$$p^+_{ij} = \frac{U_{\mathbf{H}^+_{ij}} - U_i}{\|\mathbf{B}_i\mathbf{H}^+_{ij}\|}, \quad p^-_{ij} = \frac{U_i - U_{\mathbf{H}^-_{ij}}}{\|\mathbf{B}_i\mathbf{H}^-_{ij}\|}, \quad r_{ij} = \frac{p^-_{ij}}{p^+_{ij}}. \tag{3}$$

This gives us the reconstructed scalar value at $\mathbf{M}_{ij}$:

$$U_{ij} = U_i + \|\mathbf{B}_i\mathbf{M}_{ij}\|\varphi_{ij}(r_{ij})p^+_{ij}, \tag{4}$$

where φ denotes the limiting function, ensuring a L^∞ stability property under a CFL condition with a second order accuracy where the solution is smooth. One of the drawback of this method is that the ratio r_{ij} can't be used directly in a vectorial framework, as slopes will also be vectors. One way to use it is to define complex slopes in the complex plane spanned by the slopes vectors. But actually, we will follow another approach that simplifies the stability proof, even in the scalar case, and thereafter makes the vectorial extension more natural. To do so, we recast the reconstruction (4) so that the slope ratio r_{ij} disappears.

2.2 Limitation function

We will focus here on the limitation function. We consider here a special case of equation (1), that is the scalar advection equation, as it is easier to study its behaviour and its stability. This equation reads:

$$\frac{\partial u}{\partial t} + \nabla \cdot (\boldsymbol{\lambda} u) = 0, \tag{5}$$

where u is the advected scalar. In the scalar case, the limitation function is shaped in order to achieve a second order accuracy on smooth area, and to limit the solution around discontinuities. Particularly, the limitation part of the limiter is designed so that the discretized scalar advection scheme with an explicit euler scheme as time-discretization, namely

$$U_i^{n+1} = U_i - \frac{\Delta t}{|K_i|} \sum_{j \in \mathcal{V}(i)} |S_{ij}| \Phi(U_{ij}, U_{ji}, \mathbf{n}_{ij}, \boldsymbol{\lambda}_{ij}), \tag{6}$$

achieve a maximum principle under a CFL condition, where U denotes here the discrete scalar solution of equation (5). From [18], the conditions on the shape of the limiting function read as follows:

$$0 \leq \varphi(r_{ij}) \leq \min(\eta_{ij}^+, r_{ij}\eta_{ij}^-), \tag{7}$$

where η_{ij}^+ and η_{ij}^- are geometrical parameters: $\eta_{ij}^+ = \frac{\|\mathbf{B}_i\mathbf{H}_{ij}^+\|}{\|\mathbf{B}_i\mathbf{M}_{ij}\|}$, and $\eta_{ij}^- = \frac{\|\mathbf{B}_i\mathbf{H}_{ij}^-\|}{\|\mathbf{B}_i\mathbf{M}_{ij}\|}$. The second-order accuracy is reached with another inequality:

$$\min(1, r) \leq \varphi(r) \leq \max(1, r). \tag{8}$$

This inequality means that the reconstructed scalar should be a convex combination of forward and backward slopes, which can be written as a κ-scheme [10, 25]

$$U_{ij} = U_i + \|\mathbf{B}_i\mathbf{M}_{ij}\| \left(\frac{1+\kappa}{2} p_{ij}^+ + \frac{1-\kappa}{2} p_{ij}^- \right), \tag{9}$$

where κ is a real parameter between -1 and 1. Taking now into account the monotonicity constraints, it means that κ can no longer be a constant but must depend on the specific slopes for the current face, which leads to write:

$$U_{ij} = U_i + \|\mathbf{B}_i\mathbf{M}_{ij}\| \left(\frac{1+\kappa_{ij}}{2} p_{ij}^+ + \frac{1-\kappa_{ij}}{2} p_{ij}^- \right), \tag{10}$$

where $\kappa_{ij} = \kappa(p_{ij}^+, p_{ij}^-) = \kappa(r_{ij})$ is the local value accounting for both second-order accuracy and monotonicity constraints (see Figure 2). This formulation of the reconstructed scalar is far more convenient as the former one (4) for a vectorial framework as it doesn't use directly the slope ratio r_{ij}. This is why we will use it to define our vectorial method in section 3. To get a limited reconstruction, κ_{ij} will therefore be defined as a function, whose maximum and minimum admissible values depend on the limitation inequalities (7). One can easily rewrite them as an inequality on the reconstructed variable:

$$\begin{array}{rcccl} \min(U_i, U_{\mathbf{H}_{ij}^+}) & \leq & U_{ij} & \leq & \max(U_i, U_{\mathbf{H}_{ij}^+}) \\ \min(U_i, U_{\mathbf{H}_{ij}^-}) & \leq & U_{ij}^f & \leq & \max(U_i, U_{\mathbf{H}_{ij}^-}) \end{array} \tag{11}$$

where we have defined $U_{ij}^f = 2U_i - U_{ij}$. If the scheme were monodimensional along the axis $\mathbf{B}_i\mathbf{M}_{ij}$, and if the variable U had been supposed linear on the cell $(\mathbf{M}_{ij}, \mathbf{M}_{ij}^f)$ (in blue in Figure 1), then U_{ij}^f would have been the real reconstructed value at $\mathbf{M}_{ij}^f$. For an unstructured mesh, the point $\mathbf{M}_{ij}^f$ doesn't play any role and is not explicitely defined but one can still define the hypothetical value of the discrete solution at this point, that's why we call it the fictitious reconstruction. One can see from (11) that even if this value doesn't need to be computed in practice, we still need to define it in order to establish the scheme stability property as inequalities on both U_{ij} and U_{ij}^f are required.

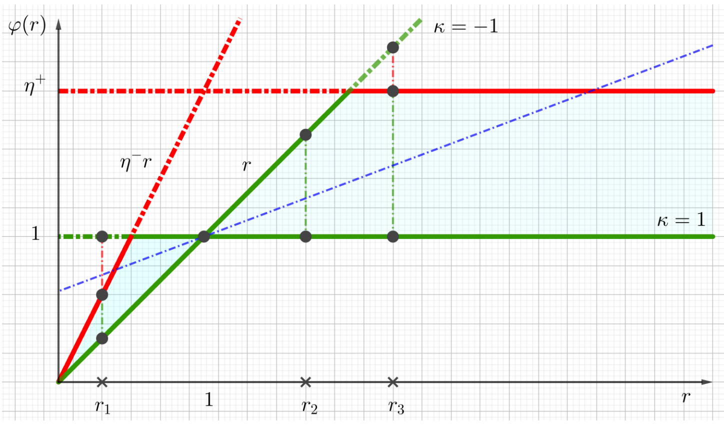

Figure 2: Combination of the second-order zone in green and the monotonicity constraints in red to get the stability zone for the limiter, and equivalence in terms of κ values.

3 VECTORIAL MULTISLOPE MUSCL SCHEME

3.1 Stability formula

The vectorial MUSCL scheme derives naturally from the stability formula of the vectorial version of the fully discretised scheme (6):

$$\mathbf{V}_i^{n+1} = \mathbf{V}_i^n - \frac{\Delta t}{|K_i|} \sum_{j \in \mathcal{V}(i)} |S_{ij}| \Phi(\mathbf{V}_{ij}, \mathbf{V}_{ij}), \tag{12}$$

which is the discrete formulation of the vectorial advection equation:

$$\frac{\partial \mathbf{v}}{\partial t} + (\boldsymbol{\lambda} \cdot \nabla)\mathbf{v} = \mathbf{0}, \tag{13}$$

where $\mathbf{v}$ is the exact vectorial solution of (13), and $\mathbf{V}$ represents its vectorial discrete solution. In order to get the stability formula, one has to make some assumption on the numerical flux function Φ. We will firstly assume that it is consistent:

$$\forall \mathbf{V} \in \mathbb{R}^d, \quad \Phi_{ij}(\mathbf{V}, \mathbf{V}) = (\lambda \cdot \mathbf{n}_{ij})\mathbf{V}_{ij}. \tag{14}$$

Secondly, we assume that there exists two scalar parameters A_{ij} and B_{ij} such that

$$\begin{aligned} A_{ij}(\mathbf{V}_{ij} - \mathbf{V}_i) &= \Phi_{ij}(\mathbf{V}_{ij}, \mathbf{V}_i) - \Phi_{ij}(\mathbf{V}_i, \mathbf{V}_i), \\ B_{ij}(\mathbf{V}_{ji} - \mathbf{V}_i) &= \Phi_{ij}(\mathbf{V}_{ij}, \mathbf{V}_{ji}) - \Phi_{ij}(\mathbf{V}_{ij}, \mathbf{V}_i). \end{aligned} \tag{15}$$

One can notice that if the numerical flux is the classical upwind numerical flux:

$$\Phi_{ij}(\mathbf{U}, \mathbf{V}) = \max(\boldsymbol{\lambda} \cdot \mathbf{n}_{ij}, 0)\mathbf{U} + \min(\boldsymbol{\lambda} \cdot \mathbf{n}_{ij}, 0)\mathbf{V}, \quad \forall \mathbf{U}, \mathbf{V} \in \mathbb{R}^d, \tag{16}$$

then these scalar parameters A_{ij} and B_{ij} exist and are equal to:

$$A_{ij} = \max(\boldsymbol{\lambda} \cdot \mathbf{n}_{ij}, 0), \quad B_{ij} = \min(\boldsymbol{\lambda} \cdot \mathbf{n}_{ij}, 0). \tag{17}$$

If we introduce the fictitious vectorial reconstruction $\mathbf{V}^f_{ij} = 2\mathbf{V}_i - \mathbf{V}_{ij}$, and two other parameters:

$$\nu^+_{ij} = \Delta t \frac{|S_{ij}|}{|K_i|} A_{ij}, \quad \nu^-_{ij} = -\Delta t \frac{|S_{ij}|}{|K_i|} B_{ij}, \tag{18}$$

we get:

$$\mathbf{V}^{n+1}_i = \mathbf{V}_i \left(1 - \sum_{j \in \mathcal{V}(i)} \left(\nu^-_{ij} + \nu^+_{ij}\right)\right) + \sum_{j \in \mathcal{V}(i)} \nu^-_{ij} \mathbf{V}_{ji} + \sum_{j \in \mathcal{V}(i)} \nu^+_{ij} \mathbf{V}^f_{ij}. \tag{19}$$

In order to get a convex combination, we have to assume that the numerical flux is monotonous:

$$A_{ij} \geq 0, \quad B_{ij} \leq 0, \tag{20}$$

which means that the coefficients ν^+_{ij} and ν^-_{ij} are both positive. From this point, one can define the following CFL condtition

$$1 - \sum_{j \in \mathcal{V}(i)} \left(\nu^-_{ij} + \nu^+_{ij}\right) \geq 0, \tag{21}$$

which ensures that the formula (19) is a convex combination of $\mathbf{V}_i$, $\mathbf{V}_{ji}$ and $\mathbf{V}^f_{ij}$. In order to get the vectorial scheme stability, we still have to choose the way we will limit the reconstruction $\mathbf{V}_{ij}$ and its fictitious counterpart $\mathbf{V}^f_{ij}$ in the same time.

3.2 Monotonicity criterion

From the formula (19), one can observe that with the right choice of limitation for the reconstruction $\mathbf{V}_{ij}$, we can get almost any kind of control over $\mathbf{V}_i^{n+1}$. In the scalar case, a monotonicity criterion is applied on the reconstructed variable in order that U_i^{n+1} achieve a maximum principle. On the vectorial case, the control we want over $\mathbf{V}_i^{n+1}$ should ideally be the vectorial extension of this maximum principle. This vectorial extension corresponds in fact to the VIP area from Luttwak and Falcovitz [19–21], that is the convex hull of vectors from a given cell neighborhood. Hence, if we write our reconstruction under a κ-scheme form:

$$\mathbf{V}_{ij} = \mathbf{V}_i + \|\mathbf{B}_i\mathbf{M}_{ij}\| \left(\frac{1+\kappa_{ij}}{2}\mathbf{p}_{ij}^{+} + \frac{1-\kappa_{ij}}{2}\mathbf{p}_{ij}^{-} \right), \tag{22}$$

we only have to find an interval $I = [\kappa^-, \kappa^+]$ such that for any value of κ_{ij} within I, we have that both $\mathbf{V}_{ij}$ and its associated fictitious $\mathbf{V}_{ij}^f = 2\mathbf{V}_i - \mathbf{V}_{ij}$ lie into the convex hull of the neighborhood vectors. Nevertheless, this is the best method only from a theoretical point of view. Indeed, as already explained in the introduction, the algorithmic cost of both the convex hull construction and the "lying in the Convex Hull" test may be prohibitive in practice. For this reason, we introduce here an alternative method to the convex hull. This alternative limitation takes the form of a truncated circular sector area around each reconstructed vector $\mathbf{V}_{ij}$ and its associated fictitious reconstruction $\mathbf{V}_{ij}^f$ (see Figure 3). This method ensures some kind of control over the norm and the direction of $\mathbf{V}_i^{n+1}$ as illustrated in Figure 4. To summarize, this area is constructed as follows:

1. $\mathbf{V}_{ij}$ and its fictitious counterpart have both to satisfy an upper bound on their norm:

$$\|\mathbf{V}_{ij}\| \le \max(\|\mathbf{V}_{\mathbf{H}_{ij}^+}\|, \|\bar{\mathbf{V}}_i\|), \quad \|\mathbf{V}_{ij}^f\| \le \max(\|\mathbf{V}_{\mathbf{H}_{ij}^-}\|, \|\bar{\mathbf{V}}_i\|). \tag{23}$$

   This condition ensures that the vector norm $\|\mathbf{V}_i^{n+1}\|$ is bounded. It is theoretically possible to build a stable numerical scheme only with this condition in the sense that the norm won't tend toward infinity. But this stability area is much bigger than the well designed convex hull previously mentionned. Especially, if all vectors point roughly to the same direction, we can assume that $\mathbf{V}_i^{n+1}$ should also point to this direction. With only the norm condition, such a control on the vector direction can not be guaranteed. That's why we introduce the next condition.

2. $\mathbf{V}_{ij}$ has to lie in an angular sector defined by $\mathbf{V}_i$ and $\mathbf{V}_{\mathbf{H}_{ij}^+}$ on the one hand, and its fictitious counterpart $\mathbf{V}_{ij}^f$ has to lie on the angular sector defined by $\mathbf{V}_i$ and $\mathbf{V}_{\mathbf{H}_{ij}^-}$ on the other hand. This condition ensures that $\mathbf{V}_{ij}$ and its associated fictitious $\mathbf{V}_{ij}^f$ won't oscillate, which implies an angular stability for $\mathbf{V}_i^{n+1}$. We have to notice that this condition can occur only for two dimensional vectors. Indeed, in a three dimensional framework, $\mathbf{V}_{ij}$ won't automatically lie in the same plane defined $\mathbf{V}_i$ and $\mathbf{V}_{\mathbf{H}_{ij}^+}$. A similar problematic occurs for the fictitious reconstruction. It means that the angular sector has to be redefined for vector of dimension 3, which won't be done here.

3. Both previous conditions are enough to ensure a vectorial stability. But if all vectors are colinear, the current limitation won't tend toward the scalar case limitation, as in the scalar case, a lower bound is also required on the reconstruction U_{ij} as shown in formula (11). If we impose a lower bound on the vectors norm:

$$\|\mathbf{V}_{ij}\| \geq \min(\|\mathbf{V}_{\mathbf{H}^+_{ij}}\|, \|\bar{\mathbf{V}}_i\|), \quad \|\mathbf{V}^f_{ij}\| \geq \min(\|\mathbf{V}_{\mathbf{H}^-_{ij}}\|, \|\bar{\mathbf{V}}_i\|), \tag{24}$$

the scheme won't be satisfying because it will prevent to use an admissible reconstruction for smooth vector fields. This is why we decided not to impose a minimum constraint on the norm, but to truncate the angular sector defined by the two previous bounds. For the angular sector defined for $\mathbf{V}_{ij}$, we draw the line going from $\min(\|\mathbf{V}_i\|, \|\mathbf{V}_{\mathbf{H}^+_{ij}}\|)\frac{\mathbf{V}_i}{\|\mathbf{V}_i\|}$ to $\min(\|\mathbf{V}_i\|, \|\mathbf{V}_{\mathbf{H}^+_{ij}}\|)\frac{\mathbf{V}_{\mathbf{H}^+_{ij}}}{\|\mathbf{V}_{\mathbf{H}^+_{ij}}\|}$. A similar line is drawn for the angular sector associated with the fictitious reconstruction $\mathbf{V}^f_{ij}$.

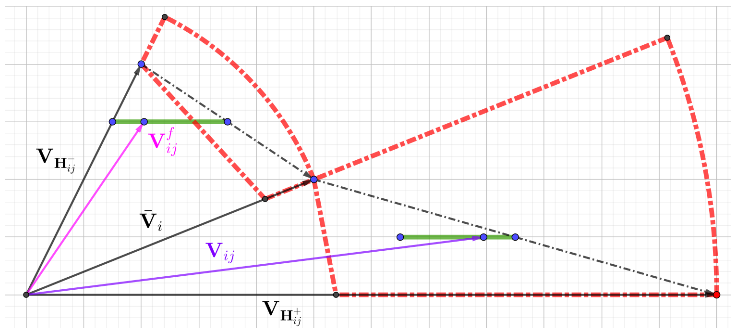

Figure 3: Shape of the vectorial stability area. Red dashed line represents the truncated circular sector. Green lines represent the set of all second-order accurate reconstructions. If $\mathbf{V}_{ij}$ and $\mathbf{V}^f_{ij}$ lie both on the green line and within the red area, then the reconstruction is admissible. Here, $\mathbf{V}^f_{ij}$ lies outside the truncated circular sector. The reconstruction $\mathbf{V}_{ij}$ is thus not accepted.

Hence, the reconstruction and limitation process is as follows:

1. We compute the two limit values κ_- and κ_+ such that $\mathbf{V}_{ij}$ lies on the boundary of its truncated circular sector. It means that for any value κ lying inside the interval $I = [\kappa_-, \kappa_+]$, the reconstructed vector $\mathbf{V}_{ij}$ assiociated with κ will lie inside its stability area.

2. We compute the two limit values κ^f_- and κ^f_+ such that $\mathbf{V}^f_{ij}$ lies on the boundary of its truncated circular sector. It means that for any value κ lying inside the interval $I^f = [\kappa^f_-, \kappa^f_+]$, the fictitious reconstructed vector $\mathbf{V}^f_{ij}$ associated with κ will lie inside its stability area.

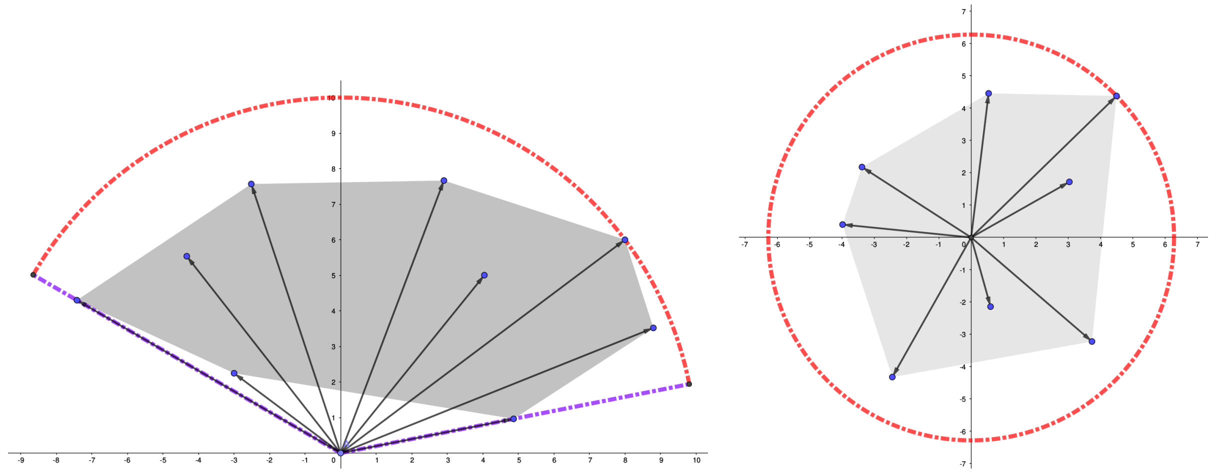

Figure 4: Bounds on $\mathbf{V}_i^{n+1}$. In grey is represented the convex hull of vectors from a given neighborhood. Red dashed line represents the norm maximum principle. Purple dashed line represents the angular bound. Left figure represents a case where vectors from the neighborhood point roughly in the same direction. Here, an angular bound seems to be required. Right figure represents a case where cell K_i is a convergence or divergence point, and therefore no angular bound holds for $\mathbf{V}_i^{n+1}$.

3. We compute the intersection of the two previous intervals, $I \cap I^f$. If this intersection is empty, then it means that we are on a kind of vectorial extremum. As in the scalar case, a vectorial extremum means that the method degenerates at first order, and $\mathbf{V}_{ij} = \mathbf{V}_i$.

4. If $I \cap I^f$ is not empty, we choose by any kind of way a value κ included in $I \cap I^f$. One can for instance choose the closest value to a unique κ_0 previously chosen, or we can choose the value which maximizes or minimizes the norm $\|\mathbf{V}_i - \mathbf{V}_{ij}\|$ on the interval $I \cap I^f$. One can also choose for example the mean value of $I \cap I^f$. Different choices of $\kappa = f(I \cap I^f)$ imply different kinds of limitation functions, and eventually different interpolation schemes.

4 CONCLUSION

In this paper, we have presented a new way of limiting vectors in the framework of multislope MUSCL schemes for finite volume methods. This limiting method consists in requiring that the reconstruction vector and its fictitious counterpart lie in a truncated circular sector, which ensures a control over the norm and the direction of $\mathbf{V}_i^{n+1}$. Nevertheless, we didn't presented detailed calculus, and extension of the angular bound to three-dimensional vectors has not been presented here. Moreover, results for this method are only theoretical so far, as no numerical tests have been presented in this paper. All these aspects will be addressed in a forthcoming paper.

REFERENCES

[1] A. Refloch, B. Courbet, A. Murrone, P. Villedieu, C. Laurent, P. Gilbank, J. Troyes, L. Tessé, G. Chaineray, J. Dargaud, E. Quémerais, and F. Vuillot. CEDRE Software. *Aerospace Lab*, (2):p. 1–10, March 2011. Publisher: Alain Appriou.

[2] B. Courbet, C. Benoit, V. Couaillier, F. Haider, M. C. L. Pape, and S. Péron. Space Discretization Methods. (2):15, 2011.

[3] S. K. Godunov and I. Bohachevsky. Finite difference method for numerical computation of discontinuous solutions of the equations of fluid dynamics. *Matematičeskij sbornik*, 47(89)(3):271–306, 1959. Publisher: Steklov Mathematical Institute of Russian Academy of Sciences.

[4] B. Van Leer. Towards the ultimate conservative difference scheme. I. The quest of monotonicity. In H. Cabannes and R. Temam, editors, *Proceedings of the Third International Conference on Numerical Methods in Fluid Mechanics*, Lecture Notes in Physics, pages 163–168, Berlin, Heidelberg, 1973. Springer.

[5] B. Van Leer. Towards the ultimate conservative difference scheme. II. Monotonicity and conservation combined in a second-order scheme. *Journal of Computational Physics*, 14(4):361–370, March 1974.

[6] B. Van Leer. Towards the ultimate conservative difference scheme. IV. A new approach to numerical convection. *Journal of Computational Physics*, 23(3):276–299, March 1977.

[7] B. Van Leer. Towards the ultimate conservative difference scheme. III. Upstream-centered finite-difference schemes for ideal compressible flow. *Journal of Computational Physics*, 23(3):263–275, March 1977.

[8] B. Van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101–136, July 1979.

[9] F. Kemm. A comparative study of TVD-limiters-well-known limiters and an introduction of new ones. *Int. J. Numer. Meth. Fluids*, 67(4):404–440, October 2011.

[10] N. Waterson and H. Deconinck. Design principles for bounded higher-order convection schemes – a unified approach. *Journal of Computational Physics*, 224(1):182–207, May 2007.

[11] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49(3):357–393, March 1983.

[12] J. B. Goodman and R. J. LeVeque. On the Accuracy of Stable Schemes for 2D Scalar Conservation Laws. *Mathematics of Computation*, 45(171):15–21, 1985. Publisher: American Mathematical Society.

[13] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87(1):171–200, March 1990.

[14] S. Spekreijse. Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws. *Math. Comp.*, 49(179):135–155, 1987.

[15] A. Jameson. Analysis and Design of Numerical Schemes for Gas Dynamics, 1: Artificial Diffusion, Upwind Biaising, Limiters and their Effect on Accuracy and Multigrid Convergence. *International Journal of Computational Fluid Dynamics*, 4(3-4):171–218, January 1995.

[16] T. Barth and M. Ohlberger. Finite Volume Methods: Foundation and Analysis. In *Encyclopedia of Computational Mechanics*. American Cancer Society, 2004. Section: 15 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/0470091355.ecm010.

[17] V. Clauzon. *Analyse de schémas d'ordre élevé pour les écoulements compressibles. Application à la simulation numérique d'une torche à plasma.* phdthesis, Université Blaise Pascal - Clermont-Ferrand II, January 2008.

[18] C. Le Touze, A. Murrone, and H. Guillard. Multislope MUSCL method for general unstructured meshes. *Journal of Computational Physics*, 284:389–418, March 2015.

[19] G. Luttwak and J. Falcovitz. Slope limiting for vectors: A novel vector limiting algorithm. *International Journal for Numerical Methods in Fluids*, 65(11-12):1365–1375, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/fld.2367.

[20] G. Luttwak and J. Falcovitz. VIP (Vector Image Polygon) multi-dimensional slope limiters for scalar variables. *Computers & Fluids*, 83:90–97, August 2013.

[21] G. Luttwak. On the Extension of Monotonicity to Multi-Dimensional Flows. page 12, 2016.

[22] P.-H. Maire, R. Loubère, and P. Váchal. Staggered Lagrangian Discretization Based on Cell-Centered Riemann Solver and Associated Hydrodynamics Scheme. *Communications in Computational Physics*, 10(4):940–978, October 2011. Publisher: Cambridge University Press.

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

[24] X. Zeng and G. Scovazzi. A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations. *Journal of Computational Physics*, 270:753–783, August 2014.

[25] B. van Leer. Upwind-difference methods for aerodynamic problems governed by the Euler equations. pages 327–336, January 1985. Conference Name: Large-Scale Computations in Fluid Mechanics.