NORMA RESEARCH REPORT

T2-D2 : ENO and WENO edge-based reconstructions

Tatiana Kozubskaya(*), Alain Dervieux(**)

July 20th 2021

(*) Keldysh Institute of Applied Mathematics (KIAM),
Russian Academy of Sciences, Moscow, Russia
(**)Projet ECUADOR, INRIA (UCA)
2004 route des lucioles, BP 93
06902 Sophia-Antipolis France

1 Introduction

The Norma project [**?**] is a Russia-France cooperation for improving high-fidelity numerical models in order to better control the noise produced by new and less new aeronefs like drones and helicopters which should move around towns with the smallest sound pollution.

Among Norma's goals is the improvement of numerical approximations based on unstructured meshes, for solving the Navier-Stokes equations (and linearisations). Dissipation and dispersion are the abomination of desolation of second order approximations. The Russian and French teams use extensions of second-order schemes called superconvergent approximation which indeed reduce dissipation and dispersion. Research will tend to further reduce them and examine how the research in high order schemes can help in further improvement of the approximations.

The present paper recalls the features of the superconvergent and ENO/WENO EBR schemes used by KIAM and INRIA and proposes several extensions.

2 Godunov method on vertex-centered tetrahedrizations

2.1 Spatial representation

Mathematical model We write the unsteady Euler equations as follows in the computational domain $\Omega \subset \mathbb{R}^3$:

$$\Psi(W) = \frac{\partial W}{\partial t} + \nabla \cdot \mathcal{F}(W) = 0 \quad \text{in} \quad \Omega, \tag{1}$$

where $W = {}^t(\rho, \rho u, \rho v, \rho w, \rho E)$ is the vector of conservative variables. $\mathcal{F}(W) = (\mathcal{F}_1(W), \mathcal{F}_2(W), \mathcal{F}_3(W))$ is the convective flux:

$$\mathcal{F}_1(W) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (\rho E + p)u \end{pmatrix}, \mathcal{F}_2(W) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (\rho E + p)v \end{pmatrix}, \mathcal{F}_3(W) = \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (\rho E + p)w \end{pmatrix}$$

so that the state equation becomes:

$$\frac{\partial W}{\partial t} + \frac{\partial \mathcal{F}_1(W)}{\partial x} + \frac{\partial \mathcal{F}_2(W)}{\partial y} + \frac{\partial \mathcal{F}_3(W)}{\partial z} = 0.$$

ρ, p and E hold respectively for the density, the thermodynamical pressure and the total energy per mass unit. Symbols u, v and w stand for the Cartesian components of velocity vector $\mathbf{u} = (u, v, w)$. For a calorically perfect gas, we have

$$p = (\gamma - 1)\big(\rho E - \frac{1}{2}\rho|\mathbf{u}|^2\big),$$

where γ is constant. A weak formulation including boundary conditions of this system writes for $W \in V = [H^1(\Omega)]^5$ as follows:

$$\forall \phi \in V, \quad (\Psi(W)\,,\,\phi) = \int_\Omega \left(\phi\,\frac{\partial W}{\partial t} + \phi \nabla \cdot \mathcal{F}(W)\right)\,\mathrm{d}\Omega - \int_\Gamma \phi\,\hat{\mathcal{F}}(W)\cdot\mathbf{n}\,\mathrm{d}\Gamma \;=\; 0, \quad (2)$$

where Γ is the boundary of the computational domain Ω (Fig.**??**), $\mathbf{n}$ the outward normal to Γ and the boundary flux $\hat{\mathcal{F}}$ contains the boundary conditions. We are interested by this unsteady formulation together with the steady one, in which the time derivative is not introduced.

Discrete variational representation We consider here the *steady* case, written:

$$\nabla \cdot \mathcal{F}(W) = 0$$

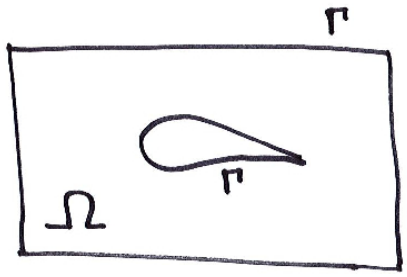

Figure 1: A typical computational domain. $\Gamma = \partial\Omega$.

or in variational formulation:

$$\forall \phi \in V, \quad (\Psi(W)\,,\,\phi) = \int_\Omega (\phi \nabla \cdot \mathcal{F}(W))\ \mathrm{d}\Omega - \int_\Gamma \phi\, \hat{\mathcal{F}}(W) \cdot \mathbf{n}\ \mathrm{d}\Gamma \ = \ 0. \quad (3)$$

The discretization chosen relies on two main choices. First, we consider a *tetrahedrisation* as the discretization of the computational domain. This choice is made in connection with the progresses made for automatically generating and adapting meshes of this kind. Second, once the mesh is chosen, we have to put on it a set of nodes, that are the geometrical supports of the degrees of freedom. The option chosen is the *set of vertices*. It is the option of the usual continuous P^1 FEM approximation. It corresponds to the smallest number of nodes for a given mesh. Let $\mathcal{T}_h$ be a tetrahedrization of Ω which is admissible for Finite-Elements *i.e.*, Ω is partitioned in tetrahedra, and the intersection of two different tetrahedra is either empty, or a vertex, or an edge, or a face. The test functions are taken into the approximation space V_h made of continuous piecewise linear functions included in $V = [H^1(\Omega)]^5$:

$$V_h \ = \ \left\{ \phi_h \ \middle|\ \phi_h \text{ is continuous and } \phi_{h|T} \text{ is linear } \forall T \in \mathcal{T}_h \right\} .$$

In order to avoid the management of projectors applicable in the whole H^1 space, we shall work inside the following spaces:

$$\bar{V} = ([H^2(\Omega)]^5) \ \text{ and } \ \bar{V}_h = \bar{V} \cup V_h \,.$$

It is useful to introduce Π_h, the corresponding P^1 interpolation operator:

$$\begin{aligned} &\Pi_h : \bar{V}_h \longrightarrow V_h \\ &\qquad \phi \longmapsto \Pi_h \phi \qquad \text{with} \quad \Pi_h \phi(i) = \phi(i) \ \ \forall i \text{ vertex of } \mathcal{T}_h \,. \end{aligned}$$

Then the discrete steady formulation of Problem (**??**) writes:

$$\forall \phi_h \in V_h, \quad \int_\Omega \phi_h \nabla \cdot \mathcal{F}_h(W_h)\ d\Omega - \int_\Gamma \phi_h \hat{\mathcal{F}}_h(W_h) \cdot \mathbf{n}\ d\Gamma \quad = \quad 0\,, \qquad (4)$$

where $\mathcal{F}_h$ is by definition the P^1 interpolate of $\mathcal{F}$, in the sense that:

$$\mathcal{F}_h(W) = \Pi_h \mathcal{F}(W) \ \text{ and } \ \mathcal{F}_h(W_h) = \Pi_h \mathcal{F}(W_h)\,, \qquad (5)$$

and, as the operator $\mathcal{F}_h$ applies to the values of W at the mesh vertices, we have:

$$\mathcal{F}_h(W) = \mathcal{F}_h(\Pi_h W) = \Pi_h \mathcal{F}(\Pi_h W)\,. \qquad (6)$$

We get the same relations for $\hat{\mathcal{F}}_h(W)$:

$$\hat{\mathcal{F}}_h(W) = \Pi_h \hat{\mathcal{F}}(\Pi_h W) \;\text{ and }\; \hat{\mathcal{F}}_h(W_h) = \Pi_h \hat{\mathcal{F}}(W_h)\,. \tag{7}$$

Practically, this definition means that nodal fluxes values $\mathcal{F}_h(\mathbf{x}_i) = \mathcal{F}_h(W(\mathbf{x}_i))$ of fluxes $\mathcal{F}_h$ are evaluated at the mesh vertices i. Discrete fluxes functions $\mathbf{x} \mapsto \mathcal{F}_h(\mathbf{x})$ are derived from the nodal values by P^1 intrapolation inside every element. In contrast to the standard Galerkin approach, this definition emphasizes that the discrete fluxes are in V_h.

2.1.1 Mixed-Element-Volume (MEV) basic equivalence

The discrete Formulation (**??**) can be transformed into a *vertex-centered finite-volume scheme* applied to tetrahedral unstructured meshes. This assumes a particular partition in control cells C_i of the discretized domain Ω_h:

$$\Omega_h = \bigcup_{i=1}^{n_c} C_i \;, \tag{8}$$

each control cell being associated with a vertex i of the mesh. The corresponding test functions are the piecewise constant characteristic functions of

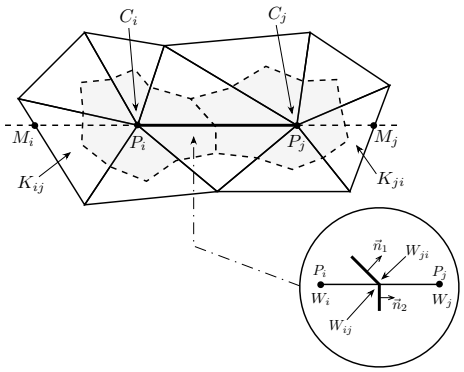

Figure 2: Illustration of finite-volume cells construction in two dimensions with two neighboring cells, C_i and C_j around i (P_i on the figure) and j (P_j on the figure) respectively, and of the upwind triangles K_{ij} and K_{ji} associated with the edge ij. Representation of the common boundary ∂C_{ij} with the solution extrapolated values for the MUSCL type approach. Dash lines are segments of medians of the triangles.

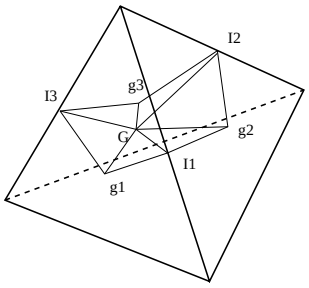

Figure 3: The planes which delimit the finite-volume cell (related to upper vertex) inside a tetrahedron (3D case). G is the tetrahedron centroid, g_k's are face centroids, I_k's are edge centers.

cells:

$$\chi^i(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in C_i, \\ 0 & \text{otherwise.} \end{cases}$$

Then, using the Stokes formula, the finite-volume weak formulation (steady case) becomes for each vertex i, *i.e.* for each cell C^i:

$$\int_{\partial C_i} \mathbf{n}_i \mathcal{F}(W)\ d\sigma = 0 \tag{9}$$

where $\mathbf{n}_i$ holds for the unit normal to ∂C_i outpointing from C_i.

The *dual finite-volume cell* is built by the rule of medians. In 2D, the median cell is limited by segments of medians between centroids and mid-

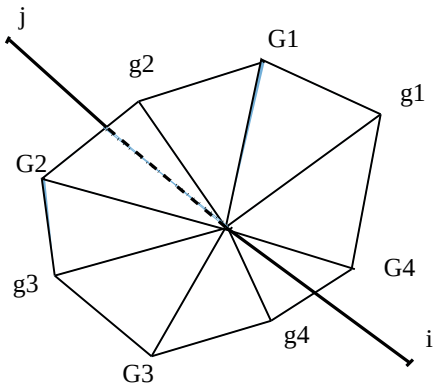

Figure 4: Illustration of finite-volume cell interface ∂C_{ij} between two neighboring cells C_i and C_j (3D case).

edge (Figure **??**). In 3D, each tetrahedron T of the mesh is split into four enneahedra (polyhedra with nine faces) constructed around each of its four vertices. For a vertex i, the enneahedron $C_i \cap T$ is defined by the following points (Figure **??**):

(i) the three middle points of the edges issued from i,

(ii) the three gravity centers of the faces containing i,

(iii) the center of gravity of the tetrahedron and

(iv) the vertex i.

The cell C_i of vertex i is the collection of all enneahedra linked to i. The common boundary $\partial C_{ij} = \partial C_i \cap \partial C_j$ between two neighboring cell C_i and C_j is decomposed in several triangular interface facets. An illustration of this construction is shown in Figure **??** for the 3D case.

The finite-volume fluxes between cells around vertices i and j are integrated through the common boundary ∂C_{ij} with a value of $\mathcal{F}_h$ equal to the half-sum of $\mathcal{F}_h(W_i)$ and $\mathcal{F}_h(W_j)$:

$$\Phi_{ij}^{MEV} = \frac{\mathcal{F}_h(W_i) + \mathcal{F}_h(W_j)}{2} \cdot \nu_{ij} \,, \tag{10}$$

where ν_{ij} denotes the integral of the normal $\mathbf{n}_i$ to common boundary between cells C_i and C_j,

$$\nu_{ij} = \int_{\partial C_{ij}} \mathbf{n}_i \; d\sigma$$

and $W_i = W(i)$. The finite-volume formulation for an internal vertex i writes as the sum of all the fluxes evaluated from the vertices j belonging to $V(i)$ where $V(i)$ is the set of all neighboring vertices of i. Taking into account the boundary fluxes, the discrete Scheme (**??**) then writes:

$$\sum_{j \in V(i)} \Phi_{ij}^{MEV} \; - \; \int_{\Gamma \cap \partial C_i} \bar{\mathcal{F}}_h(W_h) \cdot \mathbf{n} \; d\Gamma \; = \; 0 \,. \tag{11}$$

We obtain a vertex-centered finite-volume approximation which is P^1-exact with respect to the flux function $\mathcal{F}_h$. This scheme enjoys most of the accuracy properties of the Galerkin method [**?**], such as the second-order accuracy on any mesh for diffusion-convection models. However, it lacks stability and cannot be applied to purely hyperbolic models such as the Euler equations.

2.1.2 Flux integration

Once the cells are defined, the spatial divergence $div\mathcal{F}$ is transformed via the Stokes formula into integrals of normal fluxes $\mathcal{F}.\mathbf{n}$ at cell boundaries. In the proposed family of schemes, the accuracy of the integration quadrature on cell boundaries is not as crucial: we choose a very simple option, the *edge-based integration*. On the contrary, flux integration sets the important problem of scheme stabilisation. The variables are assumed to be constant by cell, and therefore, they are discontinuous from a cell to its neighbor. Upwind integration will rely on the Godunov method based on the two different values at each side of the discontinuity.

Central differencing. Let us write a vertex-centered central differenced finite-volume scheme for the steady Euler equations applied to an unstructured mesh as follows:

$$\begin{aligned} \mathbf{\Psi}_h(\Gamma, W)_i &= 0, \quad \text{with} \\ \mathbf{\Psi}_h(\Gamma, W)_i &= \sum_{j \in V(i)} \Phi^{central}(W_i, W_j, \nu_{ij}) \; + \; \mathbf{B}_h(\Gamma, W)_i \end{aligned} \tag{12}$$

where $V(j)$ is the set of vertices that are neighbors of j, ν_{jk} is the integral on interface between j and k of the normal vector. Symbol $\mathbf{B}_h(\Gamma, W)_j$ holds for boundary fluxes in which Euler fluxes take into account the available boundary information. The centered integration for elementary flux Φ is written as follows:

$$\Phi^{central}(W_i, W_j, \nu_{ij}) \; = \; 0.5(\mathcal{F}_i + \mathcal{F}_j) \cdot \nu_{ij} \tag{13}$$

where $\mathcal{F}_i = \mathcal{F}(W_i)$ are the Euler fluxes computed at W_i. This is equivalent to introduce the following discrete space operator ∇_h^*:

$$\nabla_h^*(\mathcal{F})_i \; = \; \sum_{j \in V(i)} 0.5(\mathcal{F}_i + \mathcal{F}_j) \cdot \nu_{ij} \; / \; a(i) \tag{14}$$

where $a(i)$ is the measure of cell C_i.

Godunov differencing. Godunov-type methods rely on the discontinous representation of the unknowns at cell interfaces and on the computation

of the fluxes at these discontinuities in function of both "left" and "right" values through the application of an approximate or an exact Riemann solver. This process introduces numerical viscosity terms that are very useful for stabilizing transonic flows. First we consider that W is constant by cell equal to W_i in C_i [1]. We write a vertex-centered first-order Godunov scheme for the Euler equations applied to an unstructured mesh as follows:

$$\mathbf{\Psi}_h(\Gamma, W)_i \;=\; \sum_{j \in V(i)} \Phi^{ARS}(W_i, W_j, \nu_{ij}) \;+\; \mathbf{B}_h(\Gamma, W)_i. \tag{15}$$

Here $\Phi^{ARS}(W_i, W_j, \nu_{ij})$ is evaluated by an approximate Riemann solver.

Roe approximate Riemann solver. A standard option is the Roe flux difference splitting [**?**]:

$$\Phi^{Roe}(W_i, W_j, \nu_{ij}) \;=\; 0.5(\mathcal{F}_i + \mathcal{F}_j)\cdot\nu_{ij} \;+\; 0.5|\mathcal{A}|(W_i - W_j) \tag{16}$$

where $|\mathcal{A}|$ is the absolute value of the Jacobian flux along ν_{ij}:

$$\begin{gathered}\mathcal{A} = \left(\frac{\partial \mathcal{F}}{\partial W}\right)_1 (\nu_{ij})_1 \;+\; \left(\frac{\partial \mathcal{F}}{\partial W}\right)_2 (\nu_{ij})_2 \;+\; \left(\frac{\partial \mathcal{F}}{\partial W}\right)_3 (\nu_{ij})_3 \quad \text{(3D case)} \\ \mathcal{A} = T\Lambda T^{-1}\;, \quad \Lambda \;=\; \text{diagonal eigenvalues matrix,} \\ |\mathcal{A}| = T|\Lambda|T^{-1}\;.\end{gathered} \tag{17}$$

These matrices are computed at an intermediate value $\overline{W}_{ij}$ of W_i and W_j, in short:

$$\overline{W}_{ij} \;=\; (\rho_i^{\frac{1}{2}} W_i + \rho_j^{\frac{1}{2}} W_j)/(\rho_i^{\frac{1}{2}} + \rho_j^{\frac{1}{2}})$$

which enjoys the following property:

$$\mathcal{F}(W_i) - \mathcal{F}(W_j) \;=\; \mathcal{A}(\overline{W}_{ij})(W_i - W_j)\;.$$

In the fully supersonic cases where $\mathcal{A}(\overline{W}_{ij}) = |\mathcal{A}(\overline{W}_{ij})|$ or $\mathcal{A}(\overline{W}_{ij}) = -|\mathcal{A}(\overline{W}_{ij})|$, Roe's splitting is fully upwind. By the hyperbolicity assumption, matrix $\mathcal{A}(\overline{W}_{ij})$ can be diagonalised. The absolute value $|\mathcal{A}(\overline{W}_{ij})|$ writes:

$$|\mathcal{A}(\overline{W}_{ij})| \;=\; TDiag(|\lambda_1|, |\lambda_2|, |\lambda_3|, |\lambda_4|, |\lambda_5|)T^{-1} \;=\; sign(\mathcal{A}(\overline{W}_{ij}))\mathcal{A}(\overline{W}_{ij})\;,$$

[1] In the MUSCL method, we consider that the mean value in cell C_i is identical to the value at vertex i, an approximation bringing simplification but limiting the extension to very high order

where $sign(\mathcal{A}) = TDiag(sign(\lambda_1), sign(\lambda_2), sign(\lambda_3), sign(\lambda_4), sign(\lambda_5))T^{-1}$.
Thus, this averaging also permits the following equivalent formulation:

$$\Phi^{Roe}(W_i, W_j, \nu_{ij}) = 0.5(\mathcal{F}_i + \mathcal{F}_j).\nu_{ij} + 0.5 sign(\mathcal{A}(\overline{W}_{ij}))(\mathcal{F}(W_j) - \mathcal{F}(W_i)) \quad .(18)$$

3 MUSCL and V6 scheme

3.1 Spatial second-order accuracy: MUSCL

The above schemes with Roe or HLLC are spatially first-order accurate. First-order upwind schemes of Godunov type enjoy a lot of interesting qualities and in particular HLLC enjoys formally monotonicity or, in the case of the Euler model ρ-, T- and p-positivity. They can be extended to second order by applying the MUSCL method. Indeed, the fact that the Godunov method builds fluxes between cells with unknown variables constant by cells implies first-order accuracy. Van Leer proposed [**?**, **?**] to reconstruct a linear interpolation of the variables inside each cell and then to introduce in the Riemann solver the boundary values of these interpolations. Further, the slopes used for linear reconstruction can be limited in order to represent the variable without introducing new extrema. The resulting MUSCL method produces positive second-order schemes. We describe now an extension of MUSCL to unstructured triangulations with dual cells. The MUSCL ideas also apply to reconstructions which are different on each interface between cells, or equivalently on each edge. Several slopes of a dependant variable F are defined on the two vertices i and j of an edge ij as follows :

1. *Gradients.* First, the *centered gradient* $(\nabla F)^c_{ij}$ is defined as $(\nabla F)^c_{ij} \cdot \vec{ij} = F_j - F_i$.

We consider a couple of two triangles, one having i as a vertex, and the second having j as a vertex. With reference to Figure **??**, we define ϵ_{ni} and ϵ_{mi} (resp. ϵ_{jr}, ϵ_{js}) as the components of vector $\vec{ji}$ (resp. $\vec{ij}$) in the oblique system of axes $(\vec{in}, \vec{im})$ (resp. $(\vec{jr}, \vec{js})$):

$$\vec{ji} = \epsilon_{ni}\,\vec{in} + \epsilon_{mi}\,\vec{im}\,,$$
$$\vec{ij} = \epsilon_{jr}\,\vec{jr} + \epsilon_{js}\,\vec{js}\,.$$

We shall say that T_{ij} and T_{ji} are upwind and downwind elements with respect to edge ij if the components $\epsilon_{ni}, \epsilon_{mi}, \epsilon_{jr}, \epsilon_{js}$ are all nonnegative:

$$T_{ij} \text{ upstream and } T_{ji} \text{ downstream } \Leftrightarrow Min(\epsilon_{ni}, \epsilon_{mi}, \epsilon_{jr}, \epsilon_{js}) \geq 0.$$

The *upwind gradient* $(\nabla W)^u_{ij}$ is computed as the usual finite-element gradient on T_{ij} and the *downwind gradient* $(\nabla W)^d_{ij}$ on T_{ji}. This writes :
$(\nabla W)^u_{ij} = \nabla W\,|_{T_{ij}}$ and $(\nabla W)^d_{ij} = \nabla W\,|_{T_{ji}}$ where $\nabla W\,|_T = \sum_{k\in T} W_k \nabla \Phi_k|_T$ are the P1-Galerkin gradients on triangle T.

2. *Interpolation at cell interface.* We now specify our method for computing the *interpolation slopes* $(\nabla W)_{ij}$ and $(\nabla W)_{ji}$:

$$(\nabla W)_{ij} \cdot \vec{ij} = \ (1-\beta)(\nabla W)^c_{ij} \cdot \vec{ij} + \beta (\nabla W)^u_{ij} \cdot \vec{ij} \ . \tag{19}$$

The computation of W_{ji} is analogous:

$$(\nabla W)_{ji} \cdot \vec{ij} = \ (1-\beta)(\nabla W)^c_{ij} \cdot \vec{ij} + \beta (\nabla W)^d_{ij} \cdot \vec{ij} \ . \tag{20}$$

The coefficient β is an upwinding parameter that controls the combination of fully upwind and centered slopes and that is generally taken equal to $1/3$, according to the error analysis below.

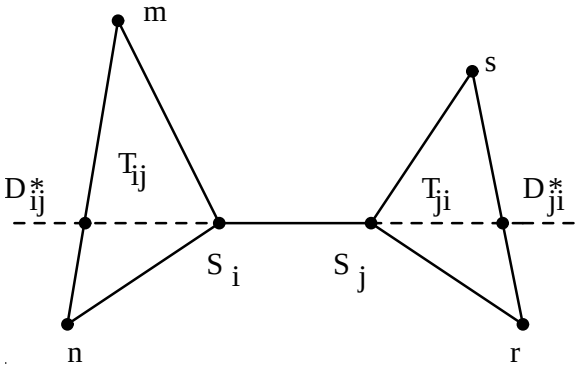

Figure 5: Butterfly molecule in 2D: Localisation of the extra interpolation points D^*_{ij} and D^*_{ji} of nodal gradients. This allows to evaluate three derivatives along direction S_iS_j, namely with D^*_{ij} and S_i, or S_i and S_j, or S_j and D^*_{ji}.

3. *Flux balance.* The scheme description is completed by replacing the first-order formulation (**??**) by the following flux balance:

$$\mathbf{\Psi}_h(\Gamma, W)_i \ = \ \sum_{j\in V(i)} \Phi^{ARS}(W_{ij}, W_{ji}, \nu_{ij}) \ + \ \mathbf{B}_h(\Gamma, W)_i. \tag{21}$$

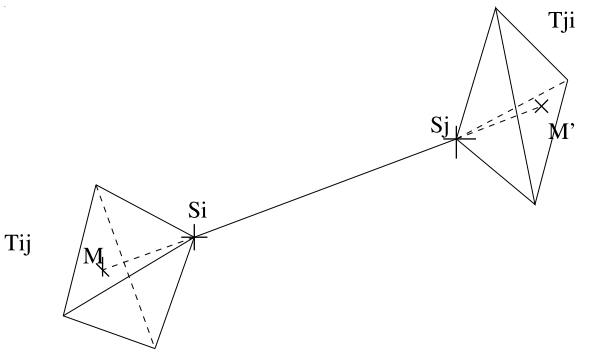

Figure 6: Butterfly molecule in 3D: Downwind and Upwind tetrahedra are tetrahedra having respectively S_i and S_j as a vertex and such that line S_iS_j intersects the opposite face.

with

$$W_{ij} = W_i + \frac{1}{2}(\nabla W)_{ij}.\vec{ij} \quad , \quad W_{ji} = W_j + \frac{1}{2}(\nabla W)_{ji}.\vec{ji} \ .$$

Remark 3.1 *The numerical viscosity introduced by the MUSCL method is equivalent to a fourth-order derivative, weighted by a $O(h^3)$ coefficient, where h is local mesh size. When using certain very stretched meshes, the combination between MUSCL and median cells can produce a severe loss in accuracy. A cure for it is discussed in Annex 1.*

For this second-order version, the amount of dissipation which is introduced is the dominant term of the numerical error and may seems larger than needed in many applications.

3.2 Low dissipation advection schemes

The approximation described in the previous section is spatially second-order accurate. Note that the method combines finite differences in the local reconstruction and finite volume for fluxes. In contrast to reconstruction based on mean values (e.g. ENO schemes), trying a higher order accurate interpolation does not bring a higher accuracy for the scheme. More precisely, for a nonlinear flux function, the accuracy of MUSCL schemes is limited to second order, as remarked by Wu and Wang [**?**].

We now examine how to moderately change the reconstruction in order to improve the scheme. We get inspired by Direct Simulation techniques in which non-dissipative high-order approximations are stabilised in good accuracy conditions thanks to filters which rely on very-high even order derivatives. In order to do this, we have to further extend the discretization stencil. Then it can be also interesting to choose a stencil extension which also improves dispersion properties, since a less dispersive scheme needs less dissipation for avoiding Gibbs-like oscillations. This leads to the idea of *superconvergent advection schemes* in the sense that they are of higher order (than two) on Cartesian mesh regions and for simpler models, like linear hyperbolic ones. More sophisticated versions giving high-order accuracy for nonlinear advection are described in [**?**].

3.2.1 Spatial scheme

This section defines a low dissipation/low dispersion scheme introduced in [**?**, **?**]. This scheme can be called superconvergent because it is designed in such a way that it is a higher-order scheme when applied to a Cartesian mesh.

The scheme is built as follows:

0. A background flow $W = (\rho, \rho u, \rho v, E)$ on each vertex of the mesh is given.

1. Compute the primitive variable $\tilde{U} = (\rho, u, v, p)$ on each vertex (vertexwise loop).

2. Compute the nodal gradients $\nabla \tilde{U}$.

$$(\nabla \tilde{U})_i = \frac{1}{meas(C_i)} \sum_{T \in C_i} \frac{meas(T)}{3} \sum_{k \in T} (\tilde{U})_k \nabla \Phi_k|_T \quad \text{(2D case)}. \tag{22}$$

3. Start *edgewise assembly loop*: compute the extrapolated slopes :

$$\begin{aligned} (\nabla \tilde{U})_{ij} \cdot \vec{ij} = \; & (1-\beta)(\nabla \tilde{U})^c_{ij} \cdot \vec{ij} + \beta (\nabla \tilde{U})^d_{ij} \cdot \vec{ij} \\ & + \xi_c \left[(\nabla \tilde{U})^u_{ij} \cdot \vec{ij} - 2(\nabla \tilde{U})^c_{ij} \cdot \vec{ij} + (\nabla \tilde{U})^d_{ij} \cdot \vec{ij} \right] \\ & + \xi_d \left[(\nabla \tilde{U})_M \cdot \vec{ij} - 2(\nabla \tilde{U})_i \cdot \vec{ij} + (\nabla \tilde{U})_j \cdot \vec{ij} \right] \quad , \end{aligned} \tag{23}$$

where $(\nabla\tilde{U})_M$ is the gradient at the point M, intersection of line ij with the face of T_{ij} which does not contain i as a vertex, as shown in Figure **??**. The expression is analog for $\nabla(\tilde{U})_{ji}$. Let us define left and right variable interpolations:

$$\begin{array}{rcl} \tilde{U}_{ij} & = & \tilde{U}_i + \nabla\tilde{U}_{ij} \\ \tilde{U}_{ji} & = & \tilde{U}_j - \nabla\tilde{U}_{ji} \end{array} \tag{24}$$

and recover the left and right values of conservative variables $W_{ij} = W(\tilde{U}_{ij})$, $W_{ji} = W(\tilde{U}_{ji})$. The upwind differenced flux then writes (the index V6 holds for the numerical viscosity by 6-th order derivatives):

$$\mathbf{\Psi}_h^{V6}(\Gamma, W)_i \;=\; \sum_{j\in V(i)} \Phi^{ARS}(W_{ij}, W_{ji}, \nu_{ij}) \;+\; \mathbf{B}_h(\Gamma, W)_j. \tag{25}$$

3.3 Time advancing

After semi-discretization of the unsteady Euler equation using the above scheme, we obtain by applying mass lumping:

$$meas(C_i)\frac{dW_i}{dt} + \mathbf{\Psi}_h(\Gamma, W)_i = 0 \tag{26}$$

The standard Runge-Kutta scheme can be used for time advancing the solution:

$$\begin{array}{l} V_1 = \Delta t\ \Psi(W^n)_i \\ V_2 = \Delta t\ \Psi(W^n + V_1/2)_i \\ V_3 = \Delta t\ \Psi(W^n + V_2/2)_i \\ V_4 = \Delta t\ \Psi(W^n + V_3)_i \\ W_i^{n+1} \;=\; W_i^n + V_1/6 + V_2/3 + V_3/3 + V_4/6 \end{array} \tag{27}$$

where $\Psi(W)_i = -\mathbf{\Psi}_h(\Gamma, W)_i / meas(C_i)$.
In many case, a linearized version of the time stepping can be used. Let us recall the Jameson variant [**?**] which writes as follows (N-stage version):

$$\begin{array}{l} W^{(0)} = W^n \\ W^{(k)} = W^{(0)} + \dfrac{\Delta t}{N-k+1}\Psi\left(W^{(k-1)}\right), \quad k = 1\ldots N \\ W^{n+1} = W^{(N)}. \end{array} \tag{28}$$

Scheme	β	ξ^c	ξ^d	CFL_{max}
RK4(0.11, 0.2766, 0.5, 1.)	1/3	0	0	1.9
RK6[**?**]	1/3	- 1/30	- 2/15	1.867
RK3-SSP[**?**]	1/3	0	0	2.

Table 1: Maximal Courant numbers for MUSCL-third-order ($\xi^c = \xi^d = 0$) and V6 spatial schemes (1D analysis). The RK4 first and second coefficients are optimized for higher CFL with MUSCL.

An A-stability analysis as in [**?**] can be applied. We give in Table **??** some typical maximal CFL numbers for the six-stage RK scheme, which ensure a global accuracy order of five for the two best schemes of the proposed family. This table illustrates that the above schemes can be used with CFL number of the order of unity.

In the sequel, we consider spatial improvements for monotony and positivity. When using these improvements, the explicit time advancing to be used is the TVD Runge Kutta of Shu.

All of these schemes can be advanced in time with implicit schemes such as BDF1 and BDF2 . A spatially first-order accurate simplified Jacobian is systematically used. In the unsteady case, that option is used inside a two-step Newton-like process refered as unsteady Defect Correction [**?**] is also possible. Linear stability is unconditional in all cases. In the case where we seek a steady solution or a slowly evoluting solution during a long time, the efficiency of an explicit scheme applying on unstructured mesh is severely limited by the Courant condition on the time step. The implicit time advancing is mandatory and needs be combined with an efficient linear solver.

Remark 3.2 *It can be efficient to apply a multigrid iteration in combination with pseudo-time advancing (steady case) or (for both cases) an efficient implicit time advancing. Designing a multigrid scheme for unstructured meshes rises the problem of defining a series of coarser grids. In other words, we have to define several new meshes or to find an alternative strategy. In [**?**, **?**], this is done in a transparent manner from the fine mesh by using the so-called cell agglomeration. Parallel multigrid extensions are proposed in [**?**, **?**].*□

Another option well adapted to message passing parallelism is a Krylov-Newton-Schwarz (KNS) algorithm, as in [**?**]. A first version of KNS, under

the form of the Restrictive Additive Schwarz (**RAS**) was developed in [**?**, **?**]. A deflation-based coarse grid extension of RAS is studied in [**?**].

4 EBR interpolation

The present section considers another family of vertex-centered schemes, namely EBR (Edge-Based Reconstruction) schemes, based on a quasi-1D edge-oriented reconstruction of variables1. This idea was firstly proposed in the 1990s, [**?**, **?**], and then has been developed (see, for instance, [**?**, **?**, **?**]) as an efficient algorithm providing higher accuracy on unstructured meshes for point-wise values. Here we present an interpretation of the EBR schemes in terms of finite differences for irregular meshes. We introduce a concept of translationally symmetric meshes which are uniform grid-like meshes and prove a high accuracy of the EBR schemes for this family of meshes. We show that the underlying high-accuracy scheme for Cartesian meshes keeps its high accuracy for linearly deformed meshes under the EBR approach. We propose a new efficient implementation of quasi-1D reconstruction techniques and thereby present the new SEBR (Simplified EBR) scheme of EBR family optimal in terms of computational costs. We estimate the computational costs of different EBR schemes in comparison with the quadratic-polynomial-based FV method and the FC scheme. Finally, we give new verification results for the 2D Rankine vortex and 3D Gaussian pulse and show the examples of problems solved by the EBR schemes.

4.1 Basic 1D high accuracy scheme

4.1.1 Scheme for the linear transport equation on uniform meshes

Let us consider the linear transport equation

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial u} = 0, \quad a > 0. \tag{29}$$

According to the method of lines, the semi-discrete approximation of equation (1) can be written as the ODE

$$\left(\frac{du}{dt}\right)_j = -\Psi_j(u)$$

where $\Psi_j(u) = a\left(\frac{\partial u}{\partial x}\right)_j$ is an approximation of the space derivative. Superscript L denotes the upwind approximation of the gradient using the stencil skewed to the left. Let us now introduce the uniform mesh with nodes x_j and the constant mesh step $\Delta x = x_{j+1} - x_j$ for all j. The computational cell for

the node j is defined as the segment with the boundaries $x_{j\pm1/2} = x_j \pm \Delta x/2$. The unknown function is defined in the mesh nodes: $u_j = u(x_j)$. We approximate the space derivative as

$$\Psi_j(u) = a\frac{-2u_{j-3} + 15u_{j-2} + 60u_{j-1} + 20u_j + 30u_{j+1} - 3u_{j+2}}{60\Delta x}. \tag{30}$$

Using the Taylor expansion, one can see that formula (**??**) provides the 5th order of accuracy. Approximation (**??**) can be also presented in the divergent form

$$\Psi_j(u) = a\frac{u^L_{j+1/2} - u^L_{j-1/2}}{\Delta x} \tag{31}$$

with the reconstructed values of unknown function u at the interface points written in terms of first finite differences:

$$\begin{aligned} u^L_{j+1/2} &= u_j + \tfrac{1}{2}\Big(-\tfrac{1}{15}\Delta u_{j-3/2} + \tfrac{11}{30}\Delta u_{j-1/2} + \tfrac{4}{5}\Delta u_{j+1/2} - \tfrac{1}{10}\Delta u_{j+3/2}\Big) \\ u^L_{j-1/2} &= u_{j-1} + \tfrac{1}{2}\Big(-\tfrac{1}{15}\Delta u_{j-5/2} + \tfrac{11}{30}\Delta u_{j-3/2} + \tfrac{4}{5}\Delta u_{j-1/2} - \tfrac{1}{10}\Delta u_{j+1/2}\Big) \end{aligned} \tag{32}$$

where $\Delta u_{k+1/2} = u_{k+1} - u_k$. If the advection velocity is negative, i.e. $a < 0$, the corresponding 5th order approximation takes the form

$$\Psi_j(u) = a\Big[\frac{\partial u}{\partial x}\Big]^R_j = a\frac{u^R_{j+1/2} - u^R_{j-1/2}}{\Delta x} \tag{33}$$

where the reconstructed values$u^R_{j\pm1/2}$ are defined as

$$\begin{aligned} u^R_{j+1/2} &= u_{j+1} - \tfrac{1}{2}\Big(-\tfrac{1}{10}\Delta u_{j-1/2} + \tfrac{4}{5}\Delta u_{j+1/2} + \tfrac{11}{30}\Delta u_{j+3/2} - \tfrac{1}{15}\Delta u_{j+5/2}\Big) \\ u^R_{j-1/2} &= u_j - \tfrac{1}{2}\Big(-\tfrac{1}{10}\Delta u_{j-3/2} + \tfrac{4}{5}\Delta u_{j-1/2} + \tfrac{11}{30}\Delta u_{j+1/2} - \tfrac{1}{15}\Delta u_{j+3/2}\Big) \end{aligned} \tag{34}$$

Here superscript R denotes the upwind approximation of the gradient using the stencil skewed to the right.

4.1.2 Scheme for the linear transport equation on non-uniform meshes

In the case of a non-uniform mesh with nodes x_j and mesh steps $\Delta x_{j+1/2} = x_{j+1} - x_j$ the computational cell for the node j is defined as the segment with the boundaries $x_{j-1/2} = (x_{j-1} + x_j)/2$ and $x_{j+1/2} = (x_j + x_{j+_1})/2$ and the

length $\bar{h}_j = x_{j+1/2} - x_{j-1/2}$ We build the higher-accuracy scheme of (**??**)-(**??**) or (**??**)-(**??**) as follows:

$$\Psi_j(u) = a\Big[\frac{\partial u}{\partial x}\Big]_j^{L/R} = a\frac{u_{j+1/2}^{L/R} - u_{j-1/2}^{L/R}}{\bar{h}_j} \tag{35}$$

where the reconstructed values $u_{j+1/2}^{L/R}$ are defined as

$$\Psi_j(u) = a\Big[\frac{\partial u}{\partial x}\Big]_j^{R} = a\frac{u_{j+1/2}^{R} - u_{j-1/2}^{R}}{\Delta x} \tag{36}$$

where the reconstructed values$u_{j\pm1/2}^{R}$ are defined as

$$\begin{aligned} u_{j+1/2}^{L} &= u_j + \tfrac{\Delta x_{j+1/2}}{2}\Big(- \tfrac{1}{15}\tfrac{\Delta u_{j-3/2}}{\Delta x_{j-3/2}} + \tfrac{11}{30}\tfrac{\Delta u_{j-1/2}}{\Delta x_{j-1/2}} + \tfrac{4}{5}\tfrac{\Delta u_{j+1/2}}{\Delta x_{j+1/2}} - \tfrac{1}{10}\tfrac{\Delta u_{j+3/2}}{\Delta x_{j+3/2}}\Big) \\ u_{j+1/2}^{R} &= u_{j+1} - \tfrac{\Delta x_{j+1/2}}{2}\Big(- \tfrac{1}{10}\tfrac{\Delta u_{j-1/2}}{\Delta x_{j-1/2}} + \tfrac{4}{5}\tfrac{\Delta u_{j+1/2}}{\Delta x_{j+1/2}} + \tfrac{11}{30}\tfrac{\Delta u_{j+3/2}}{\Delta x_{j+3/2}} - \tfrac{1}{15}\tfrac{\Delta u_{j+5/2}}{\Delta x_{j+5/2}}\Big) \end{aligned} \tag{37}$$

We can also consider the scheme on the reduced 3-points stencil as

$$\begin{aligned} u_{j+1/2}^{L} &= u_j + \tfrac{\Delta x_{j+1/2}}{2}\Big(\tfrac{1}{3}\tfrac{\Delta u_{j-1/2}}{\Delta x_{j-1/2}} + \tfrac{2}{3}\tfrac{\Delta u_{j+1/2}}{\Delta x_{j+1/2}}\Big) \\ u_{j-1/2}^{R} &= u_{j+1} - \tfrac{\Delta x_{j+1/2}}{2}\Big(\tfrac{2}{3}\tfrac{\Delta u_{j+1/2}}{\Delta x_{j+1/2}} + \tfrac{1}{3}\tfrac{\Delta u_{j+3/2}}{\Delta x_{j+3/2}}\Big) \end{aligned} \tag{38}$$

To define the values $u_{j-1/2}^{L/R}$, we apply the same procedure of reconstruction. Thus, for non-uniform meshes we use the same coefficients in front of divided differences as in the uniform-mesh reconstruction. Such definition of the reconstructed variables on the non-uniform mesh guarantees the following two important properties: 1) formulas (**??**) coincide with the high-accuracy reconstruction (3b), (4b) on uniform meshes; 2) formulas (**??**) provide the exactness on linear functions of values and in interface points . We remark here that (**??**) is not the only possible way of reconstruction on non-uniform meshes which provides these two properties.

4.1.3 Scheme for nonlinear equations

Consider the 1D hyperbolic system of conservation laws

$$\frac{\partial \mathbf{Q}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{Q})}{\partial x} = 0. \tag{39}$$

The general formulation of an upwind scheme for equation (**??**) can be written as

$$\begin{aligned}&\Big(\frac{d\mathbf{Q}}{dt}\Big)_j + \frac{H_j}{\Delta x_{j+1/2} + \Delta x_{j-1/2}} = 0\\ &H_j = \Big(h_{j+1/2}(\mathbf{F}^R_{j+1/2}, \mathbf{F}^L_{j+1/2}, \mathbf{Q}^R_{j+1/2}, \mathbf{Q}^L_{j+1/2})\\ &-h_{j-1/2}(\mathbf{F}^R_{j-1/2}, \mathbf{F}^L_{j-1/2}, \mathbf{Q}^R_{j-1/2}, \mathbf{Q}^L_{j-1/2})\Big)\end{aligned} \tag{40}$$

where $\mathbf{Q}^L_{j\pm1/2}$ and $\mathbf{Q}^R_{j\pm1/2}$ are the values of conservative variables reconstructed from the left and right sides (notations L and R, respectively, see (**??**)) with respect to the computational cell boundaries , and are reconstructed flux variables. is the numerical flux defined by some Riemann solver. For instance, the above scheme V6 on unstructured meshes is based on the Roe solver disregarding the values of reconstructed fluxes:

$$\begin{aligned}&h^{Roe}(\mathbf{Q}^R_{j\pm1/2}, \mathbf{Q}^L_{j\pm1/2}) = \frac{\mathbf{F}(\mathbf{Q}^R_{j\pm1/2}) + \mathbf{F}(\mathbf{Q}^L_{j\pm1/2})}{2}\\ &-\frac{\delta}{2}|\frac{d\mathbf{F}}{d\mathbf{Q}}(\mathbf{Q}^{Roe}_{j\pm1/2})|(\mathbf{Q}^R_{j\pm1/2} - \mathbf{Q}^L_{j\pm1/2})\end{aligned} \tag{41}$$

Here the parameter δ controls the scheme dissipation: $0 < \delta < 1$. Within the family of schemes (**??**) for nonlinear hyperbolic equations there exist no schemes of accuracy higher than of the second-order in the point-wise sense even on uniform meshes [**?**]. To overcome this barrier, NLV6 was presented based on the 1D solver of Huang[**?**] :

$$\begin{aligned}&h^{Huang}(\mathbf{F}^R_{j\pm1/2}, \mathbf{F}^L_{j\pm1/2}) = \frac{\mathbf{F}^R_{j\pm1/2} + \mathbf{F}^L_{j\pm1/2}}{2}\\ &-\frac{\delta}{2}sign\Big(\frac{d\mathbf{F}}{d\mathbf{Q}}(\mathbf{Q}_{j\pm1/2})\Big)(\mathbf{F}^R_{j\pm1/2} - \mathbf{F}^L_{j\pm1/2})\end{aligned} \tag{42}$$

In contrast to (**??**), in this scheme we reconstruct the fluxes $\mathbf{F}^R_{j\pm1/2}$ and $\mathbf{F}^R_{j\pm1/2}$ using the same formulae (**??**) as for the reconstruction of variables. The finite-difference scheme (**??**), (**??**) with the flux reconstruction (**??**) possesses the 5th order of accuracy on uniform meshes and transforms to the 6th order central-difference scheme if $\delta = 0$. However this scheme can cause instability near physical boundaries. So there we switch to the reconstruction of conservative variables (**??**), (**??**).

A more robust algorithm which also provides the 5th-6th order of accuracy on uniform meshes is obtained if we use the reconstructed values both for

variables and fluxes:

$$h^{Roe}(\mathbf{F}^R_{j\pm1/2}, \mathbf{F}^L_{j\pm1/2}, \mathbf{Q}^R_{j\pm1/2}, \mathbf{Q}^L_{j\pm1/2}) = \frac{\mathbf{F}(\mathbf{Q}^R_{j\pm1/2}) + \mathbf{F}(\mathbf{Q}^L_{j\pm1/2})}{2} \\ -\frac{\delta}{2}|\frac{d\mathbf{F}}{d\mathbf{Q}}(\mathbf{Q}^{Roe}_{j\pm1/2})|(\mathbf{Q}^R_{j\pm1/2} - \mathbf{Q}^L_{j\pm1/2}) \tag{43}$$

This way is slightly more expensive than schemes (**??**), (**??**) and (**??**), (**??**) because of a larger number of reconstructions. However in some cases it provides a more stable computation. Unless otherwise specified later on we imply the Riemann solver of Huang type (**??**) and, correspondingly, the reconstruction of fluxes, as it is in NLV6.

4.2 Edge-based reconstruction schemes for unstructured meshes

4.2.1 Basic conservative vertex-centered formulation

To solve system (**??**) numerically on an arbitrary mesh, we construct a scheme with variables determined at mesh nodes. In what follows, such schemes are referred to as vertex-centered. Around each node, we construct median cells, for which, according to the finite-volume approach, difference analogues of con- servation laws are formulated. The grid function W_i is defined as the integral mean of the function W over the cell constructed around the node i . By using the Ostrogradsky–Gauss formula, system (1) is rewritten in vector-matrix form as

$$meas(C_i)\frac{dW_i}{dt} + \sum_{j\in\mathcal{V}(i)} \mathbf{F}_{ij}s_{ij} = F_{i,V} \ ,$$

where $meas(C_i)$ is the volume of the cell corresponding to node i, $\mathbf{F}_{ij}$ is the integral mean of the function $\mathbf{F}\cdot\mathbf{n}$ over the cell face separating nodes i and j, s_{ij} is the area of this cell face, n is the unit normal vector, $\mathcal{V}(i)$ is the set of first-order neighbor for node i , and $F_{i,V}$, is an integral function of the viscous flux F_v on the cell corresponding to node i . The convective fluxes $\mathbf{F}_{ij}$ are computed using the Roe method for the approxi- mate solution of the Riemann problem:

$$\mathbf{F}_{ij} = \frac{1}{2}(\mathbf{F}^R_{ij} + \mathbf{F}^L_{ij}) - \frac{1}{2}|\mathbf{A}_{ij}|(W^R_{ij} - (W^L_{ij}).$$

The value ($W_{ij}^{L/R}$ to the left and right of the interface are determined using quasi-one-dimensional reconstructions $R_{ij}^{L/R}(W)$ defined on stencils whose points belong to the straight line containing the edge ij . The values of the fluxes $\mathbf{F}_{ij}^{L/R}$ are set equal to $F(R_ij^{L/R}(W))\mathbf{n}_{ij}$ or $R_{ij}^{L/R}(F \cdot \mathbf{n}_{ij})$ depending on the chosen type of reconstructions [**?**]. Here, $\mathbf{n}_{ij}$ is the integral mean of the vector $\mathbf{n}$ over the common face between the cells corresponding to nodes i and j , $|\mathbf{A}_{ij}| = |dFn/dW(\bar{W}_{ij})|$, $\bar{W}_{ij}$ is the Roe average computed over the values $W_ij^{L/R}$. The scheme thus designed involves edge-based reconstructions of variables and, in a broad sense, belongs to the class of EBR schemes. In a narrower sense, according to [**?**], EBR schemes make use of reconstructions that, on translation-invariant meshes (i.e., ones mapped into itself under the translation through the vector of any mesh edge, Figure **??**),

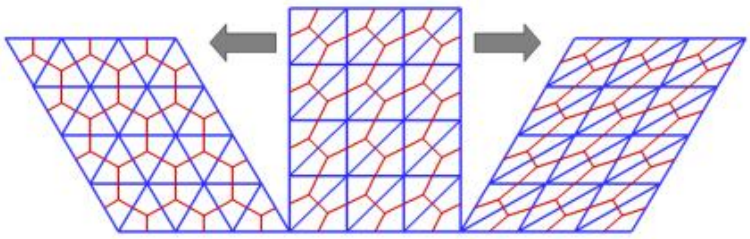

Figure 7: *Translationally symmetric meshes.*

transform the given method in a high order finite-differ- ence scheme.

Moreover, a scheme of this family is called an EBR n scheme if, in the linear case, its order on translation-invariant meshed is equal to n. Below, the method for constructing quasi-one-dimensional reconstructions used in the original for- mulation of EBR schemes is described for the EBR5 scheme in the two-dimensional formulation (Fig. **??**).

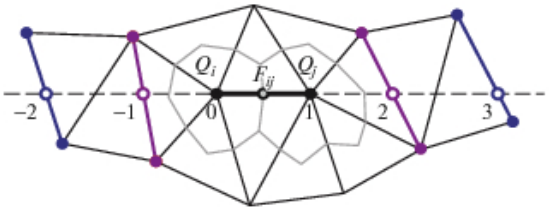

Figure 8: EBR5 stencil for edge ij on a triangular unstructured mesh.

Suppose that we need to reconstruct the value of a function W at the midpoint of an edge ij . For each node of edge ij , we construct the sets of its first- and second-order topological neighbors. The inter- section point of the ray ji with the set of faces all of whose nodes are second-order neighbors of node i is denoted by index –2, while the intersection point of this ray with the set of faces all of whose nodes are first-order neighbors of node i is denoted by index –1. If the former point is nonunique, the index –2 denotes the most distant point from node i . Similarly, for the ray ij and node j , we obtain points with indi- ces 3 and 2, respectively. The values of W at the points –2, –1, 2, 3 are determined using linear interpo- lation over the corresponding faces crossed by the ray. If nodes i and j are assigned indices 0 and 1, respec- tively, then the reconstruction operators for the function W in terms of divided differences, namely,

$$\Delta_m^L(W) = \frac{W_{m+1} - W_m}{r_{m+1} - r_m}, \quad \Delta_m^R(W) = -\Delta_{-m}^L(W)$$

can be written as

$$\begin{aligned} R_{ij}^L(W) &= W_j + \frac{1}{2}|r_i - r_j| \sum_m \beta_m \delta_m^L(W) \\ R_{ij}^R(W) &= W_j - \frac{1}{2}|r_i - r_j| \sum_m \beta_m \delta_m^R(W) \end{aligned} \tag{44}$$

where $\beta_{-2} = -1/15, \beta_{-1} = 11/30, \beta_0 = 4/5$, and $\beta_1 = -1/10$. In the EBR3 scheme, which has a shorter stencil, these coefficients are $\beta_{-2} = 0, \beta_{-1} = 1/3, \beta_0 = 2/3$, and $\beta_1 = 0$. As was noted above, for linearized equations on translation-invariant meshes, the EBR5 and EBR3 schemes are theoretically fifth- and third-order accurate, respectively. In the arbitrary case, the numerical order of accuracy of the EBR3 and EBR5 schemes varies from 5/4 to 3 depending on the quality of the used unstructured mesh. In the numerical method used to solve system (**??**) convective fluxes are approximated by applying the EBR5 or EBR3 scheme. Viscous fluxes are approximated using the Galerkin method with piecewise linear basis functions (with a diagonalized mass matrix). Time stepping relies on a first-order implicit scheme with Newton linearization of the system of discrete equations. Within one Newton itera- tion, the system of linear algebraic equations is solved using the biconjugate gradient method with ILU0 preconditioner.

5 ENOV6: for better behavior on non-uniform mesh

5.1 Motivation for ENOV6

The V6 scheme is very accurate in Cartesian mesh, and more precisely in mesh *orthogonal and uniform* (the steps in x and in y may be different).
Like any MUSCL scheme, the V6 scheme is a *fake finite volume scheme*:
- the formulation is divergent, conservative for a certain integration constant per cell, but:
- the unknowns are values at the vertices (= nodes) of the tetraedrization,
- so that *nodal reconstructions* are carried out: knowing the values taken at the nodes, we interpolate.
- Under these conditions, increasingly precise interpolations / extrapolations of flows *does not give* ([**?**]) increasingly accurate schemas as for the schemas with average value based reconstructions. Rather, we need corrected formulas based on a truncation analysis of the schema, which is relatively easy with Cartesian meshes.

The behavior of the V6 scheme thus obtained has been widely evaluated in any mesh. According to the regularity of the mesh, we keep some of the advantages of Cartesian insofar as the precision is better than order two, with little dissipation. This with an inexpensive scheme.

When using a it strongly adaptive mesh the V6 scheme reaches its limits and its solutions present in particular too large phase errors. A evident source of this fault is the presence of very strong variations in mesh size (between neighboring edges),

The purpose of this last section is to study the possibilities of finding a variant of V6 which will behave better in it variable orthogonal mesh. The working criterion will therefore be an increase in precision in orthogonal mesh (not in x variable, not in y variable), and this at least in Barth cells.

This section analyzes two different attacks:

(a) Study of the CENO 1D formulation:

- This one-dimensional formulation in variable mesh is made more explicit.

- We suggest correctors.

- We define a multidimensional extension.

(b) In appendix a 1D analysis of a MUSCL formulation. We make finite volumes by calculating flows from values at vertices. We propose a 1D analysis of the main term and its error. We deduce that *with variable mesh, the third order correction of MUSCL cannot be done conservatively.*

5.2 Analysis of CENO2 in 1D for a variable mesh

5.2.1 Notations

We adopt the model:

$$u_t + cu_x = 0$$

that we discretize on an interval with pseudo-nodes x_i and interfaces $x_{i+\frac{1}{2}} = \frac{1}{2}(x_i + x_{i+1})$.
For a continuous solution of the equation we will denote by $u_i$the values at points x_i and $\bar{u}_i$means on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$:

$$\bar{u}_i = \frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u(x)dx.$$

Given average values $\bar{u} = (\bar{u}_i)_i$, we consider for all i the *quadratic reconstruction* $P_i(x)$ of $\bar{u}$in each cell $]x_{i-1/2}, x_{i+1/2}[$:

$$u_i^r(x) \equiv P_i(x - x_i) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i$$

the averga of which on cells $j = i - 1, i, i + 1$ defined by:

$$\bar{P}_i^j := \frac{2}{x_{j+1} - x_{j-1}} \int_{x_{j-1/2}}^{x_{j+1/2}} P_i(x)dx$$

satisfies:

$$\bar{P}_i^j = \bar{u}_j \quad , \quad i - 1, i, i + 1.$$

5.2.2 CENO2 1D Reconstruction

We explicit now the variable mesh formulation for an implementation in an existing V6 code. We have :

$$\bar{P}_i^j = \frac{2}{x_{j+1} - x_{j-1}} \left[\frac{1}{3} a_i (x - x_i)^3 + \frac{1}{2} b_i (x - x_i)^2 + c_i x \right]_{x_{j-1/2}}^{x_{j+1/2}}$$

For $j = i - 1$:

$$\bar{P}_i^{i-1} = 2 \frac{\frac{a_i}{3}[(x_{i-1/2} - x_i)^3 - (x_{i-3/2} - x_i)^3]}{x_i - x_{i-2}}$$
$$+2 \frac{\frac{b_i}{2}[(x_{i-1/2} - x_i)^2 - (x_{i-3/2} - x_i)^2]}{x_i - x_{i-2}} + 2 \frac{c_i [x_{i-1/2} - x_{i-3/2}]}{x_i - x_{i-2}}$$

Pour $j = i$:

$$\bar{P}_i^{i} = 2 \frac{\frac{a_i}{3}[(x_{i+1/2} - x_i)^3 - (x_{i-1/2} - x_i)^3]}{x_{i+1} - x_{i-1}}$$
$$+2 \frac{\frac{b_i}{2}[(x_{i+1/2} - x_i)^2 - (x_{i-1/2} - x_i)^2]}{x_{i+1} - x_{i-1}} + 2 \frac{c_i [x_{i+1/2} - x_{i-1/2}]}{x_{i+1} - x_{i-1}}$$

Pour $j = i + 1$:

$$\bar{P}_i^{i+1} = 2 \frac{\frac{a_i}{3}[(x_{i+3/2} - x_i)^3 - (x_{i+1/2} - x_i)^3]}{x_{i+2} - x_i}$$
$$+2 \frac{\frac{b_i}{2}[(x_{i+3/2} - x_i)^2 - (x_{i+1/2} - x_i)^2]}{x_{i+2} - x_i} + 2 \frac{c_i [x_{i+3/2} - x_{i+1/2}]}{x_{i+2} - x_i}$$

We note that:

$$\frac{1}{2}(x_i - x_{i-2}) = x_{i-1/2} - x_{i-3/2} \quad = \quad (x_{i-1/2} - x_i) - (x_{i-3/2} - x_i)$$

$$(x_{i-1/2}-x_i)^3-(x_{i-3/2}-x_i)^3 \quad =$$

$$[(x_{i-1/2}-x_i)-(x_{i-3/2}-x_i)][(x_{i-1/2}-x_i)^2+(x_{i-3/2}-x_i)^2+(x_{i-1/2}-x_i)(x_{i-3/2}-x_i)]$$

Then:

$$2\frac{\frac{a_i}{3}[(x_{i-1/2}-x_i)^3-(x_{i-3/2}-x_i)^3]}{x_i-x_{i-2}} \quad =$$

$$\frac{a_i}{3}\ [(x_{i-1/2}-x_i)^2+(x_{i-3/2}-x_i)^2+(x_{i-1/2}-x_i)(x_{i-3/2}-x_i)].$$

De même:

$$2\frac{\frac{a_i}{3}[(x_{i+1/2}-x_i)^3-(x_{i-1/2}-x_i)^3]}{x_{i+1}-x_{i-1}} \quad =$$

$$\frac{a_i}{3}\ [(x_{i+1/2}-x_i)^2+(x_{i-1/2}-x_i)^2+(x_{i+1/2}-x_i)(x_{i-1/2}-x_i)].$$

et:

$$2\frac{\frac{a_i}{3}[(x_{i+3/2}-x_i)^3-(x_{i+1/2}-x_i)^3]}{x_{i+1}-x_i} \quad =$$

$$\frac{a_i}{3}\ [(x_{i+3/2}-x_i)^2+(x_{i+1/2}-x_i)^2+(x_{i+3/2}-x_i)(x_{i+1/2}-x_i)].$$

Similarly, we simplify the coefficients of the b's:

$$\frac{[(x_{i-1/2}-x_i)^2-(x_{i-3/2}-x_i)^2]}{x_i-x_{i-2}} =$$

$$\frac{1}{2}\frac{[(x_{i-1/2}-x_i)^2-(x_{i-3/2}-x_i)^2]}{(x_{i-1/2}-x_i)-(x_{i-3/2}-x_i)} = \frac{1}{2}[(x_{i-1/2}-x_i)+(x_{i-3/2}-x_i)]$$

$$\frac{[(x_{i+1/2}-x_i)^2-(x_{i-1/2}-x_i)^2]}{x_{i+1}-x_{i-1}} = \frac{1}{2}[(x_{i+1/2}-x_i)+(x_{i-1/2}-x_i)]$$

$$\frac{[(x_{i+3/2}-x_i)^2-(x_{i+1/2}-x_i)^2]}{x_{i+2}-x_i} = \frac{1}{2}[(x_{i+3/2}-x_i)+(x_{i+1/2}-x_i)]$$

The polynomial coefficients (a_i, b_i, c_i) are solution of the matrix system:

$$\alpha a_i + \beta b_i + \gamma c_i = \bar{u}_{i-1}$$
$$\delta a_i + \varepsilon b_i + \eta c_i = \bar{u}_i$$
$$\theta a_i + \kappa b_i + \lambda c_i = \bar{u}_{i+1}$$

where the $\alpha, \beta, ...$ also depend on i (index i is omitted for simpler notation):

$$\alpha = \frac{1}{3}[(x_{i-1/2} - x_i)^2 + (x_{i-3/2} - x_i)^2 + (x_{i-1/2} - x_i)(x_{i-3/2} - x_i)]$$
$$\delta = \frac{1}{3}[(x_{i+1/2} - x_i)^2 + (x_{i-1/2} - x_i)^2 + (x_{i+1/2} - x_i)(x_{i-1/2} - x_i)]$$
$$\theta = \frac{1}{3}[(x_{i+3/2} - x_i)^2 + (x_{i+1/2} - x_i)^2 + (x_{i+3/2} - x_i)(x_{i+1/2} - x_i)]$$

et:

$$\beta = \frac{1}{8}(x_{i-1/2} - x_{i-3/2}) \quad ; \quad \gamma = 1$$
$$\varepsilon = \frac{1}{8}(x_{i+1/2} - x_{i-1/2}) \quad ; \quad \eta = 1$$
$$\kappa = \frac{1}{8}(x_{i+3/2} - x_{i+1/2}) \quad ; \quad \lambda = 1.$$

$$\begin{array}{l} ddx1 = x_{i-1/2} - x_i = -(x_i - x_{i-1})/2 \\ ddx2 = x_{i-3/2} - x_i = -(x_{i-1} - x_{i-2})/2 - (x_i - x_{i-1}) \\ ddx3 = x_{i+1/2} - x_i = (x_{i+1} - x_i)/2 \\ ddx4 = x_{i+3/2} - x_i = (x_{i+2} - x_{i+1})/2 + (x_{i+1} - x_i) \end{array} \tag{45}$$

$$\begin{array}{l} \alpha = (ddx1 * ddx1 + ddx2 * ddx2 + ddx1 * ddx2)/3. \\ \delta = (ddx3 * ddx3 + ddx1 * ddx1 + ddx3 * ddx1)/3. \\ \theta = (ddx4 * ddx4 + ddx3 * ddx3 + ddx4 * ddx3)/3. \\ \beta = ddx1 - ddx2 \ ; \ \varepsilon = ddx3 - ddx1 \ ; \ \kappa = ddx4 - ddx3 \end{array} \tag{46}$$

Remark (uniform case) : in the uniform case (cf. PHD of Carabias) we have:

$$\alpha = \theta = 13\Delta x^2/12 \;\; ; \;\; \delta = \Delta x^2/12 \;\; ; \;\; \beta = -\Delta x = -\kappa \;\; ; \;\; \varepsilon = \Delta x$$

which shows the system is well-posed.□

Let us eliminate the c_i's:

$$(\alpha - \gamma\delta/\eta)a_i + (\beta - \gamma\varepsilon/\eta)b_i = \bar{u}_{i-1} - \gamma\bar{u}_i/\eta$$
$$(\theta - \lambda\delta/\eta)a_i + (\kappa - \lambda\varepsilon/\eta)b_i = \bar{u}_{i+1} - \lambda\bar{u}_i/\eta$$

then:

$$P_i^{quadr}(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i$$

$$\begin{aligned} a_i &= \frac{(\kappa - \lambda\varepsilon/\eta)(\bar{u}_{i-1} - \gamma\bar{u}_i/\eta) - (\beta - \gamma\varepsilon/\eta)(\bar{u}_{i+1} - \lambda\bar{u}_i/\eta)}{(\kappa - \lambda\varepsilon/\eta)(\alpha - \gamma\delta/\eta) - (\beta - \gamma\varepsilon/\eta)(\theta - \lambda\delta/\eta)} \\ b_i &= \frac{(\bar{u}_{i-1} - \gamma\bar{u}_i/\eta) - (\alpha - \gamma\delta/\eta)a_i}{(\beta - \gamma\varepsilon/\eta)} \\ c_i &= \tfrac{1}{\eta}\Big[\bar{u}_i - \varepsilon b_i - \delta a_i\Big]. \end{aligned} \tag{47}$$

We use the fact that γ, η, λ are equal to

$$P_i^{quadr}(x) = a_i(x - x_i)^2 + b_i(x - x_i) + c_i$$

$$\begin{aligned} a_i &= \frac{(\kappa - \varepsilon)(\bar{u}_{i-1} - \bar{u}_i) - (\beta - \varepsilon)(\bar{u}_{i+1} - \bar{u}_i)}{(\kappa - \varepsilon)(\alpha - \delta) - (\beta - \varepsilon)(\theta - \delta)} \\ b_i &= \frac{(\bar{u}_{i-1} - \bar{u}_i) - (\alpha - \delta)a_i}{(\beta - \varepsilon)} \\ c_i &= \bar{u}_i - \varepsilon b_i - \delta a_i. \end{aligned} \tag{48}$$

then:

$$\boxed{\begin{aligned} &kameps = \kappa - \varepsilon \;;\; bemeps = \beta - \varepsilon \;;\; almdel = \alpha - \delta \\ &a_i = \frac{kameps * (\bar{u}_{i-1} - \bar{u}_i) - bemeps * (\bar{u}_{i+1} - \bar{u}_i)}{kameps * almdel - bemeps * (\theta - \delta)} \\ &b_i = \frac{\bar{u}_{i-1} - \bar{u}_i - almdel * a_i}{bemeps} \\ &c_i = \bar{u}_i - \varepsilon * b_i - \delta * a_i \\ &P_i^{quadr}(x_{i+1/2}) = a_i * ddx3 * ddx3 + b_i * ddx3 + c_i \end{aligned}} \tag{49}$$

Verification :
Quadratic function: $x^2 + x + 1$; Δx is uniform.

For $j = i - 1$:

$$\bar{u}_{i-1} = 2\frac{\frac{1}{3}[(x_{i-1/2} - x_i)^3 - (x_{i-3/2} - x_i)^3]}{x_i - x_{i-2}}$$

$$+2\frac{\frac{1}{2}[(x_{i-1/2} - x_i)^2 - (x_{i-3/2} - x_i)^2]}{x_i - x_{i-2}} + 2\frac{[x_{i-1/2} - x_{i-3/2}]}{x_i - x_{i-2}}$$

$$\bar{u}_{i-1} = \frac{2}{3}\frac{(\Delta x/2)^3 - (3\Delta x/2)^3}{2\Delta x} + \frac{(\Delta x/2)^2 - (3\Delta x/2)^2}{2\Delta x} + 2\frac{\Delta x}{2\Delta x}$$

$$\bar{u}_{i-1} = -\frac{13}{12}\Delta x^3 - \Delta x^2 + 1$$

For $j = i$:

$$\bar{u}_i = 2\frac{\frac{1}{3}[(x_{i+1/2} - x_i)^3 - (x_{i-1/2} - x_i)^3]}{x_{i+1} - x_{i-1}}$$

$$+2\frac{\frac{1}{2}[(x_{i+1/2} - x_i)^2 - (x_{i-1/2} - x_i)^2]}{x_{i+1} - x_{i-1}} + 2\frac{[x_{i+1/2} - x_{i-1/2}]}{x_{i+1} - x_{i-1}}$$

$$\bar{u}_i =$$

For $j = i + 1$:

$$\bar{u}_{i+1} = 2\frac{\frac{1}{3}[(x_{i+3/2} - x_i)^3 - (x_{i+1/2} - x_i)^3]}{x_{i+2} - x_i}$$

$$+2\frac{\frac{1}{2}[(x_{i+3/2} - x_i)^2 - (x_{i+1/2} - x_i)^2]}{x_{i+2} - x_i} + 2\frac{[x_{i+3/2} - x_{i+1/2}]}{x_{i+2} - x_i}$$

$$\bar{u}_{i+1} =$$

5.2.3 CENO scheme

Definition of CENO2 1D in variable mesh Using the quadratic reconstruction for obtaining the values at interfaces at left and right, we get a third-order accurate CENO:

$$(x_{i+1/2} - x_{i-1/2})\frac{\partial \bar{u}_i}{\partial t} = \big[\Phi_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) - \Phi_{i-1/2}(\bar{P}_i, \bar{P}_{i-1})\big].$$

This scheme would dissipate with a fourth-order derivative if the reconstructed values were put in a Riemann solver like Roe's:

$$\Phi^{Roe}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) = \Phi^{central}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) + \Phi^{visq}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i)$$

$$\Phi^{central}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) = \tfrac{c}{2}\ [\bar{P}_{i+1}(x_{i+1/2}) + \bar{P}_i(x_{i+1/2})]$$

$$\Phi^{visq}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) = \tfrac{|c|}{2}[\bar{P}_{i+1}(x_{i+1/2}) - \bar{P}_i(x_{i+1/2})],$$

but we do not choose this option since it produces too much dissipation (comparable to MUSCL). We prefer a central-CENO option with arithmetic mean:

$$\Phi^{central}_{i+1/2}(\bar{P}_{i+1}, \bar{P}_i) = \frac{c}{2}\ [\bar{P}_{i+1}(x_{i+1/2}) + \bar{P}_i(x_{i+1/2})].$$

CENO2 1D in a uniform mesh It satisfies:

$$-\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial (cu)}{\partial x}dx + \frac{1}{\Delta x}\big[\Phi_{i+1/2} - \Phi_{i-1/2}\big] = -\frac{c}{30}u_i^{(5)}\Delta x^4 + o(\Delta x^5).$$

It is fourth-order accurate, with h^4-fifth-derivative dispersive first troncation term. When c varies its enters in the derivative.

The two previous terms are analysed separately:

$$\frac{1}{\Delta x}\int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial (cu)}{\partial x}dx = cu_i' + cu_i^{(3)}\frac{c\Delta x^2}{24} + cu_i^{(5)}\frac{c\Delta x^4}{1920} + o(\Delta x^5)$$

$$\frac{1}{\Delta x}\big[\Phi_{i+1/2} - \Phi_{i-1/2}\big] = cu_i' + cu_i^{(3)}\frac{c\Delta x^2}{24} + cu_i^{(5)}\frac{21c\Delta x^4}{640} + o(\Delta x^5)$$

as a consequence, inside a FD formulation, accuracy reduces to second order.
CENO2 1D in non-uniform mesh Since we are 2-exact, we have third order,

$$-\frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial (cu)}{\partial x}dx + \frac{1}{x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}}\big[\Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}}\big] = O(\Delta x^3).$$

The non-uniform mesh analysis is certainly very complex and will do, as in the case of FVM, a first term appears:
- non-conservative,
- of order three,
- canceling out if the mesh is uniform,
- and probably containing a fourth derivative which will make it dissipative or anti-dissipative (more oscillating in variable mesh).

The term to correct *in non-uniform* appearing as non-conservative, whereas we want to stay in conservative discretization, we don't correcting it.

On the other hand, we can find the superconvergence in uniform mesh like the original V6. We will consider a correction with respect to its *uniform mesh dispersion.*

And in addition, we will improve its stability in general via a *dissipation* of order six.

5.3 Antidispersive extrapolation

In the uniform 1D case, the scheme remains 2-exact, precise to order three and even, in its CENO-centered version, precise to order four since present as the first error term (cf. thesis of Carabias):

$$err = -\frac{1}{30} u_i^{(5)} \Delta x^4. \tag{50}$$

In chapter 3 (5.3.3 page 72) of Carabias, it is shown that this error *will not be not compensated* by passing to an interpolation of a higher degree, that is, by introducing a term of order three dictated by Taylor's formula. Indeed :
- on the one hand, a global cubic reconstruction would have been necessary to guarantee order four,
- on the other hand we are already precise to order four.

The transition to order five in a uniform mesh will be defined via a truncation analysis (Carabias page 129), which leads to the following corrector

adhoc defined for two neighboring points i and j:

$$\boxed{\begin{array}{l} P_i^{antidisp}(\frac{x_i+x_j}{2}) = P_i^{quadr}(\frac{x_i+x_j}{2}) + \frac{16}{9}\frac{1}{3!}\frac{||B_1B_2||^3}{2^3}u^{(3)}_{B_1B_2}(i). \\ P_j^{antidisp}(\frac{x_i+x_j}{2}) = P_j^{quadr}(\frac{x_i+x_j}{2}) - \frac{16}{9}\frac{1}{3!}\frac{||B_2B_1||^3}{2^3}u^{(3)}_{B_1B_2}(j). \end{array}} \quad (51)$$

where B_1B_2 denotes the edge ij, and $u^{(3)}_{B_1B_2}(i)$ (resp. $u^{(3)}_{B_1B_2}(j)$), are third derivatives in the direction of ij and in the direction ji. In the first line we recognize the addition of an additional term of the Taylor formula. this is also the case in the second line except that we must take into account that B_1B_2 change its sign. The difference of these additional terms results in a fourth derivative in the interface flux. Their summation (Green) around each cell gives a fifth derivative intended to compensate for the error (ref D5Dx4).

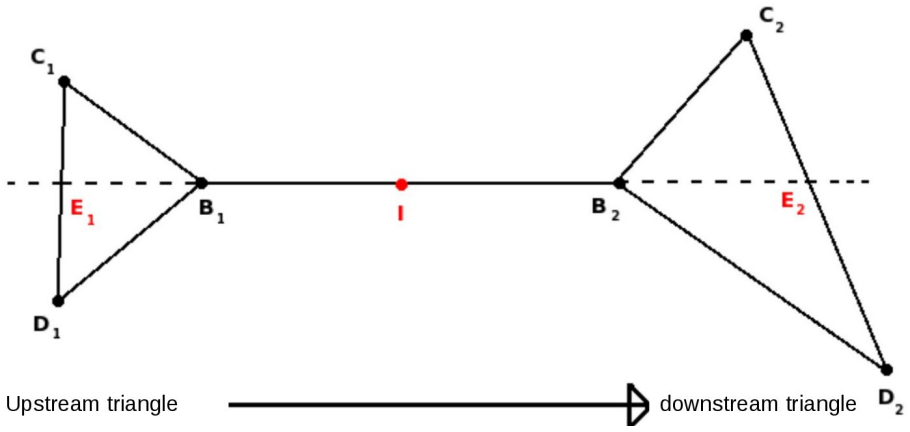

Figure 9: *Butterfly molecule related to flux between cell around vertex B_1 and cell around vertex B_2. Vertices C_1, D_1, and C_2, D_2 are the other vertices of "upwind" and "downwind" triangles.*

Approximation of third derivatives: in order to minimize surprises, we apply exactly the strategy proposed in Carabias' thesis in 1D (paragraph 5.3.3) then in multi-D.

This strategy consists of using the second derivatives from the *reconstruction* to get the third derivatives by discrete derivations. In uniform mesh however, these derivatives are expressed as follows:

$$\alpha_i = \frac{1}{3}\frac{\bar{u}_{i+2} - 2\bar{u}_{i+1} + 2\bar{u}_{i-1} - \bar{u}_{i-2}}{4\Delta x^3}.$$

The 1D polynomial correction to obtain order five is written:

$$\delta P_i = \dot{\alpha}\alpha_i (x - x_i)^3, \quad \text{avec} \quad \dot{\alpha} = -\frac{16}{5}.$$

The resulting 1D scheme is accurate to order five. In digital experiments, it turns out to be the most accurate of the schemes considered in Carabias' thesis with an L2 error of 0.2 % on the propagation of a sine wave discretized on 6 nodes, figure 5.13 page 75).

5.4 Dérivées supérieures P2 exactes

We try to introduce correction terms from the previous scheme which will improve its performances in Cartesian mesh without risking degrade its precision to a non-Cartesian mesh.

To this end we will first define approximations of the directional derivatives along the edges.

Let $f = cu$. For the evaluation of the third derivatives $D^3 f_{j+\ frac12}, ...,$ we propose to choose P_2-exact formulas, i.e. giving third zero derivatives for quadratic functions. The 1D scheme will remain P2-exact and therefore of order three.

The basic assumption will be that we have six mean values over a set of six nodes centered on $j, j+1$ 1:

$$\bar{f}_{j-2}, \bar{f}_{j-1}, \bar{f}_j, \bar{f}_{j+1}, \bar{f}_{j+2}, \bar{f}_{j+3}.$$

Let's do each derivation in turn. Five discrete prime derivatives can be defined

$$D^1 f_{k+\frac{1}{2}} = \frac{\bar{f}_{k+1} - \bar{f}_k}{x_{k+1} - x_k} + O(\Delta x^2), \quad k = j-2, ..., j+2. \tag{52}$$

Lemma: *This formulation is second-order accurate*

$$D^1 f_{k+\frac{1}{2}} = f'(x_{k+\frac{1}{2}}) + O(x_{k+1} - x_k)^2$$

for nodal values f_i as well for mean values $\bar{f}_i$.

Proof in case of mean values: Thesis de Carabias page 57: we consider:

$$\bar{f}_{i+1} = f_i + u'_i \Delta x + f_i^{(2)} \frac{13}{24} \Delta x^2 + ...$$

$$\bar{f}_{i-1} = f_i - f'_i \Delta x + f_i^{(2)} \frac{13}{24} \Delta x^2 + ...$$

with $i-1 = k, i+1 = k+1$ et $x_i = x_{k+\frac{1}{2}} = (x_{k+1} - x_k)/2$, on obtient (**??**).□

For a parabola $x \mapsto P_2(x)$, this derivative is exact only at point $x_{k+\frac{1}{2}}$:

$$D^1 P_{2,k+\frac{1}{2}} = \frac{dP_2}{dx}(x_{k+\frac{1}{2}}).$$

WE omit the mean notation using f instead of $\bar{f}$. We deduce from D^1 four discrete derivatives':

$$D^2 f_k = \frac{D^1 f_{k+\frac{1}{2}} - D^1 f_{k-\frac{1}{2}}}{x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}}}, \quad k = j-1, ..., j+2.$$

which are exact for P_2 and else of maximal accuracy at point:

$$D^2 f_k \approx \frac{d^2 f}{dx^2}(d_k), \quad avec \quad d_k = \frac{x_{k+\frac{1}{2}} + x_{k-\frac{1}{2}}}{2}.$$

We can have three third derivatives:

$$D^3 f_{k+\frac{1}{2}} = \frac{D^2 f_{k+1} - D^2 f_k}{d_{k+1} - d_k}, \quad k = j-1, j, j+1$$

which are exact (vanishing) for P_2 and else of maximal accuracy at point:

$$D^3 f_{k+\frac{1}{2}} \approx \frac{d^3 f}{dx^3}(e_{k+\frac{1}{2}}), \quad avec \quad e_{k+\frac{1}{2}} = \frac{d_{k+1} + d_k}{2}.$$

Deux dérivées quatrièmes peuvent être évaluées comme suit:

$$D^4 f_k = \frac{D^3 f_{k+\frac{1}{2}} - D^3 f_{k-\frac{1}{2}}}{e_{k+1} - e_k}, \quad k = j, j+1.$$

Gathering the above, the new variable-mesh formulation is written:

$$\boxed{\begin{array}{lll} D^1 f_{k+\frac{1}{2}} = & \dfrac{f_{k+1}-f_k}{x_{k+1}-x_k}, \quad x_{k+\frac{1}{2}} = \dfrac{x_{k+1}+x_k}{2} & k = j-1, ..., j+2. \\ D^2 f_k = & \dfrac{D^1 f_{k+\frac{1}{2}} - D^1 f_{k-\frac{1}{2}}}{x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}}}, \quad d_k = \dfrac{x_{k+\frac{1}{2}} + x_{k-\frac{1}{2}}}{2} & k = j-1, ..., j+2. \\ D^3 f_{k+\frac{1}{2}} = & \dfrac{D^2 f_{k+1} - D^2 f_k}{d_{k+1} - d_k}, & k = j-1, j, j+1. \\ D^4 f_k = & \dfrac{D^3 f_{k+\frac{1}{2}} - D^3 f_{k-\frac{1}{2}}}{e_{k+1} - e_k}, & k = j, j+1. \end{array}}$$

In order to benefit from the analysis of Carabias (page 69), instead of $D^3 f_{k+\frac{1}{2}}$ we prefer a third-order derivative centered on an node:

$$\boxed{\begin{array}{l} D^3 f_k \approx \dfrac{d^3 f}{dx^3}(g_k), \quad with \quad g_k = \dfrac{d_{k+1}+d_{k-1}}{2} \\ \\ D^3 f_k = \dfrac{D^2 f_{k+1} - D^2 f_{k-1}}{d_{k+1} - d_{k-1}}, \quad k = j-1, j, j+1. \end{array}} \tag{53}$$

Remark: *The uniform case become:*

$$\begin{array}{rll} \dfrac{df}{dx}(\dfrac{x_{k+1}+x_k}{2}) \approx D^1 f_{k+\frac{1}{2}} = & \dfrac{f_{k+1}-f_k}{\Delta x} & \\ \dfrac{d^2 f}{dx^2}(x_k) \approx D^2 f_k = & \dfrac{D^1 f_{k+\frac{1}{2}} - D^1 f_{k-\frac{1}{2}}}{\Delta x} = \dfrac{f_{i+1} - 2f_i + f_{i-1}}{\Delta x^2} & \\ \dfrac{d^3 f}{dx^3}(x_k) \approx D^3 f_k = & \dfrac{D^2 f_{k+1} - D^2 f_{k-1}}{\Delta x} = \dfrac{f_{i+2} - 2f_{i+1} + 2f_{i-1} - f_{i-2}}{\Delta x^3} & \end{array}$$

which is in accordance with Carabias (page 69).

5.5 Extrapolation fot sixth-order diffusion

We continue to add corrections as in Carabias pages 83-87 and 128-133. We currently have a central antidispersive scheme that is not very useful because it is not dissipative. We will add just enough dissipation by simply adding to our polynomial interpolation an approximation of the fourth term of Taylor's formula. We won't gain any precision other than that resulting from an additional stability:

$$\boxed{\begin{array}{l} P_i^{antidisp+dissip}(\frac{x_i+x_j}{2}) = P_i^{antidisp}(\frac{x_i+x_j}{2}) + T_{ij}^4 \\ \\ T_{ij}^4 = \frac{1}{4!}\frac{||B_1B_2||^4}{2^4} u^{(4)}_{B_1B_2}(i) \\ \\ P_j^{antidisp+dissip}(\frac{x_i+x_j}{2}) = P_j^{antidisp}(\frac{x_i+x_j}{2}) + T_{ji}^4 \\ \\ T_{ji}^4 = \frac{1}{4!}\frac{||B_2B_1||^4}{2^4} u^{(4)}_{B_2B_1}(j) \end{array}} \tag{54}$$

This term is the only term to inject into the viscous term $\Phi^{visq}_{i+1/2}$ of the Riemann solver:

$$\Phi^{CENOV6}_{i+1/2} = \Phi^{central}_{i+1/2}(P^{antidisp}_{i+1}, P^{antidisp}_{i}) + \Phi^{visq}_{i+1/2}(T^4_{ij}, T^4_{ji})$$

It will result in a difference (Right-Left) giving a fifth derivative. Then the divergence will result in a sixth derivative, our stabilizer term.
Let us now choose an approximation of these fourth derivatives. They must satisfy the following properties:
- to be more or less centered on the node,
- to be consistent in a variable mesh,
- have their restrictions to a uniform mesh in accordance with the discrete derivative chosen in Carabias (first formula at the top of page 86). We recall this formula:

$$u_i^{(4)} = \frac{a_{i+1} - 2a_i + a_{i-1}}{\Delta x^2} \quad \text{avec} \quad a_i = \frac{\bar{u}_{i+1} - 2\bar{u}_i + \bar{u}_{i-1}}{\Delta x^2}.$$

La formule introduite plus haut:

$$u^{(4)}_{B_1B_2}(k) = D^4u_k = \frac{D^3u_{k+\frac{1}{2}} - D^3u_{k-\frac{1}{2}}}{e_{k+1} - e_k}, \quad k = i, k = j$$

is conveninient since compatible with the uniform case.

5.6 Multi-dimensional extension: reflexions about splitting

We consider:

$$u_t + u_x + u_y = 0.$$

The MUSCL V6 method uses directional finite volume splitting, allowing to implement one-dimensional schemes in the directions of the edges.
In a regular molecule (2D equilateral mesh), this is justified by Stokes-based decomposition of the divergence in three directions for which we have approximate directional derivatives at the center i.

It is interesting to ask whether it is possible to combine multidimensionally one-dimensional CENO schemas.

We can first examine this question on a *cartésien* context, by applying CENO on *square* cells. We therefore consider:

- the CENO2-2D diagram reconstructed on the square and its eight neighbors in 2D, with integrated flows on two Gauss points of each interface. Classically with this exact integration for quadratic functions, the scheme is 2-exact and therefore third order accurate.

- a CENO2-1D * 1D diagram with a horizontal reconstruction, a vertical reconstruction, and integration on only the midpoints of the interfaces:

This diagram starts from values $\bar{u}_{ij}$ which are like in CENO2-2D mean values on square cells ij.

Reconstruction following x:

We consider the three cells
- cell i, j,
- cell $i-1, j$,
- cell $i+1, j$.

The starting unknowns are the means over these three squares:

$$\bar{u}_{ij} = \int_{C_{ij}} u(x,y)dx = \int_{y_{i-1/2}}^{y_{i+1/2}} \big[\int_{x_{i-1/2}}^{x_{i+1/2}} u(x,y)dx \big] dy$$

We introduce the y-mean :

$$u_j^{(x)}(x) = \frac{1}{y_{j+1/2} - y_{j-1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x,y)dy.$$

Let $\bar{u}_{ij}^{(x)}$ be the 1D function reconstructued withthe three means: de $u_j^{(x)}$:

$$\bar{u}_j^{(x,i)} = \frac{1}{x_{i+1/2} - x_{i-1/2}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_j^{(x)}dx = \bar{u}_{ij}$$

$$\bar{u}_j^{(x,i+1)} = \frac{1}{x_{i+3/2} - x_{i+1/2}} \int_{x_{i+\frac{1}{2}}}^{x_{i+\frac{3}{2}}} u_j^{(x)}dx = \bar{u}_{i+1j}$$

$$\bar{u}_j^{(x,i-1)} = \frac{1}{x_{i-1/2} - x_{i-3/2}} \int_{x_{i-\frac{3}{2}}}^{x_{i-\frac{1}{2}}} u_j^{(x)}dx = \bar{u}_{i-1j}.$$

If u is quadratic in (x,y), then $u_j^{(x)}$ is quadratic in x and its reconstruction 1D on three aligned cells is exact, so that its reconstructed value in $x_{i-\frac{1}{2}}$ is *exactly the mean of* u on the corresponding edge of the square:

$$u_j^{(x)}(x_{i-\frac{1}{2}}) = \frac{1}{y_{i+1/2} - y_{i-1/2}} \int_{y_{i-1/2}}^{y_{i+1/2}} u(x_{i-\frac{1}{2}}, y)dy.$$

Reconstruction in y direction: is similar.

Integration of fluxes:

We use only one integration point, the middle of edge:

$$\int_{(x_i,y_{j-\frac{1}{2}})}^{(x_i,y_{j+\frac{1}{2}})} u(x)dy \equiv (y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}})u_j^{(x)}(x_{i-\frac{1}{2}})$$

$$= \int_{y_{i-1/2}}^{y_{i+1/2}} u(x_{i-\frac{1}{2}}, y)dy$$

and the scheme is 2-exact.

Lemme: *Schemes CENO2-1D * 1D in 2D, CENO2-1D * 1D * 1D in 3D are 2-exact and therefore precisely in order of three. The property is true for any orthogonal structured mesh.*□

Remarque: *This property is true to any order. In practice on V6, taking into account the accessible nodes, we can switch to* **cubic**.□

This property is extended to include Cartesian edition of the Gourwitch/Barth type.

It should be noted that Barth cells in MUSCL formulation were at a disadvantage because deprived of the element-volume equivalence true only for median. Going to a formulation in averages, Barth is better justified.

For Cartesian meshes of the Friedrichs-Keller median type, the superconvergence is not proven. We probably have here a loss compared to the old version of V6.

5.7 Identification of means and mesh sizes, 2D/3D

In order not to regress compared to the MUSCL-V6, the first design criterion in our adaptation to the non-uniform case is that the purely Cartesian case must be processed keeping the order of maximum precision (five in dissipative). The progress towards MUSCL-V6 will consist of also have this order in orthogonal.

We consider the butterfly node (figure 1) between the vertices X_i and X_j. The butterfly knot is used to build fictitious knots upstream and downstream of the edge X_iX_j oriented in this direction, namely:

$$X_{i-2},\ X_{i-1},\quad \text{upwind to } i \ \text{ et } \quad X_{j+1},\ X_{j+2} \quad \text{downwind to } j.$$

We need to identify five consecutive interval lengths following the direction X_iX_j:

$$\ell_{i-2,i-1} = ||X_{i-2} - X_{i-1}||, \quad \ell_{i-1,i} = ||X_{i-1} - X_i||, \quad \ell_{i,j} = ||X_j - X_i||,$$
$$\ell_{j,j+1} = ||X_j - X_{j+1}||, \quad \ell_{j+1,j+2} = ||X_{j+1} - X_{j+2}||$$

and six values of the dependent variable from $i-2$ to $j+2$.
bf Interval lengths:

- we directly calculate $\ell_{i,j} = ||X_j - X_i||$.

noindent - we have calculated in INTERPOL3D the intersections of the prolongation of ij with the upstream and downstream triangles and we deduce from this $\ell_{i-1,i}$ and $\ell_{j,j+1}$. These measurements will be exact in the Cartesian case.

noindent - for the two end segments, we do not have the data, we will take the following approximation, which is still exact in the Cartesian case:

$$\ell_{j+1,j+2} \triangleq \ell_{j,j+1} \quad , \quad \ell_{i-2,i-1} \triangleq \ell_{i-1,i}.$$

Remark: Another option assumes that the mesh progression is geometric (this option, useful for turbulent boundary layer meshes also contains the uniform case):

$$\ell_{j+1,j+2} \triangleq \ell^2_{j,j+1}/\ell_{j-1,j} \quad , \quad \ell_{i-2,i-1} \triangleq \ell^2_{i-1,i}/\ell_{i,i+1,j}.\square$$

Values of unknowns:

- we know the values in i, j.

- we will be able to calculate in INTERPOL3D the values in $i-1$ and $j+1$.

- we will extrapolate the available gradients (one nodal and one between two nodes) to correct the values in $i-1$ and $j+1$ (paying attention to the signs)

$$f_{i-2} = f_{i-1} - \Big(2\ \nabla f|_{i-1}(x_i - x_{i-1}) - (f_i - f_{i-1})\Big)$$
$$f_{j+2} = f_{j+1} - \Big(2\ \nabla f|_{j+1}(x_i - x_{i-1}) - (f_{j+1} - f_j)\Big)$$

where the *grad* $f|_k$ denote nodal gradients calculated as usual.

In the cas of a linear V6, we shall take:

$$f \quad = \quad (\rho, \rho u, \rho v, \rho w, \rho E).$$

We admit (like most ENO users and in particular Carabias) that the k-exact reconstruction of the variables, although not making the approximation Euler k-exacts fluxes, still allows to have an accurate schema to the order $k+1$.

In the case of a non-linear V6, we will take the Euler fluxes:

$$f \quad = \quad (\rho u, \rho^2 + p, \rho uv, \rho uw, \rho(E+p), ...).$$

5.8 Summary of the definition of the new schema

Loop on the edges: we remove the incomplete edges which do not have both an upstream element and a downstream element (we suppose that we have defined the gradients on the nodes of the edge).

On the complete edges:

a. *Identification of steps* l_i*, 1D volumes* v_i*, and values* u_i *linked to* v_i, knowing that from now on these are average values in 1D:

- the values are calculated from the nodal values for u_4, u_4, and reconstituted from the various nodal gradients for the others.
- l_2, l_3, l_4 are evaluated from the mesh. l_1, l_5 can either be evaluated or extrapolated.

- we deduce the 1D volumes v_2, v_3, v_4, v_5 by half-sums. We can extrapolate for v_1, v_6 but we will not use them.

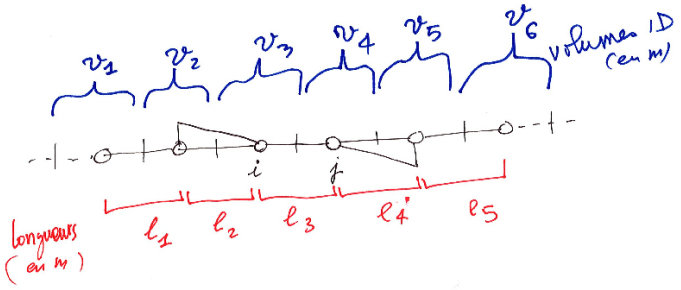

Figure 10: *Six values, six corresponding volumes, and five interval lengths between the six points.*

$l_3 = ||x_i - x_j||$

l_2 eand l_4 via intersection with opposite face

$l_1 = l_2^2/l_3 \quad ; \quad l_5 = l_4^2/l_3$

$i = 2, 5, \quad v_i = (l_i + l_{i+1})/2.$

b. *Values of unknows :*

$u_3 = u_i, \quad u_4 = u_j$

u_2 and u_5 via gradients on upwind/downwind elements

c. *Two ENO reconstructions:*

$-\mathcal{P}_i$ at i, based on v_2, v_3, v_4 and u_2, u_3, u_4

$-\mathcal{P}_j$ at j, based on v_3, v_4, v_5 and u_3, u_4, u_5.

d. *Reconstructed values in central flows (mid-edges):*

$-\mathcal{P}_i(\frac{x_3+x_4}{2})$,

$-\mathcal{P}_j(\frac{x_3+x_4}{2})$.

e. *Two anti-dispersive corrective terms (side i, side j in the it centered fluxes on the mid-edge).*

$$
\begin{aligned}
D^1 u_{k+\frac{1}{2}} &= \frac{u_{k+1}-u_k}{x_{k+1}-x_k}, \quad x_{k+\frac{1}{2}} = \frac{x_{k+1}+x_k}{2} \quad k = j-1, ..., j+2. \\
D^2 u_k &= \frac{D^1 u_{k+\frac{1}{2}} - D^1 u_{k-\frac{1}{2}}}{x_{k+\frac{1}{2}} - x_{k-\frac{1}{2}}}, \quad d_k = \frac{x_{k+\frac{1}{2}} + x_{k-\frac{1}{2}}}{2} \quad k = j-1, ..., j+2. \\
D^3 u_{k+\frac{1}{2}} &= \frac{D^2 u_{k+1} - D^2 u_k}{d_{k+1} - d_k}, \quad k = j-1, j, j+1. \\
\bar{D}^3 u_k &= \frac{D^2 u_{k+1} - D^2 u_{k-1}}{d_{k+1} - d_{k-1}}, \quad k = j-1, j, j+1, \\
D^4 u_k &= \frac{D^3 u_{k+\frac{1}{2}} - D^3 u_{k-\frac{1}{2}}}{e_{k+1} - e_k}, \quad k = j, j+1.
\end{aligned}
$$

$$\boxed{\begin{array}{l} P_i^{antidisp}(\dfrac{x_i+x_j}{2}) = P_i^{quadr}(\dfrac{x_i+x_j}{2}) + \dfrac{16}{9}\dfrac{1}{3!}\dfrac{||x_j-x_i||^3}{2^3}\bar{D}^3 u_3. \\ P_j^{antidisp}(\dfrac{x_i+x_j}{2}) = P_j^{quadr}(\dfrac{x_i+x_j}{2}) - \dfrac{16}{9}\dfrac{1}{3!}\dfrac{||x_j-x_i||^3}{2^3}\bar{D}^3 u_4. \end{array}} \tag{55}$$

f. *Two anti-dissipative corrective terms (side i, side j in the it centered fluxes on the mid-edge).*

$$\boxed{\begin{array}{l} P_i^{antidisp+dissip}(\dfrac{x_i+x_j}{2}) = P_i^{antidisp}(\dfrac{x_i+x_j}{2}) + \dfrac{1}{4!}\dfrac{||x_j-x_i||^4}{2^4}\bar{D}^4 u_3 \\ \\ P_j^{antidisp+dissip}(\dfrac{x_i+x_j}{2}) = P_j^{antidisp}(\dfrac{x_i+x_j}{2}) + \dfrac{1}{4!}\dfrac{||x_j-x_i||^4}{2^4}\bar{D}^4 u_4 \end{array}} \tag{56}$$

g. *Final assembly of flows by substitution of reconstructed and corrected values.*

5.9 Final remarks

The V6 method is based on directional processing (“ edgewise ”) advection. We show in the appendix that the 1D extension of V6 to variable mesh is not possible without loss of conservativeness. This document therefore proposes to replace the 1D MEV scheme by a 1D ENO scheme. The new scheme shouldn't be much more expensive than the V6. It should be more precise on orthogonal meshes with variable pitch. On such meshes, the reconstructions used are P2-exact on horizontal and vertical. The scheme will be of order three in Gourvitch-Barth.

It will be less CPU consuming than the CENO version multidimensional proposed by Carabias (remember that these diagrams have explicit stability similar to MUSCL). From Carabias experiences, the CENO 1D scheme is about as good as the V6 while the CENO 2D versions seemed less good. In the case of adapted meshes, the new scheme should work better on metric-orthogonal mesh and in Barth. A priori, ENO in Barth cells being purely finite volumes, it has a better chance of being precise than MEV in Barth. It remains, however, to manage the combination with the viscous Galerkin terms.

The current document does not deal with boundary conditions (domain boundaries, parallel partitions boundaries). The reconstructions 'being consistent with of multidimensional CENO, it may be useful to switch near edges to multidimensional CENO (more expensive, bur more accurate) where the reconstructions will look for enough neighbors to stay accurate.

Finally, it should be noted that we could question the initial directional option. A possible multi-D track opens via the Katz scheme.

6 Concluding remarks

This document has presented the superconvergent methods used by the Norma partners. Numerical comparisons will be reported in deliverable T2-D3. Improvement of these methods are currently studied and will be described in the Norma reports in preparation.

7 ANNEXE: Analyse FVM 1D en maillage non-uniforme

Our study will be based on a truncation analysis of the FVM scheme. It is known that the truncations in FVM, in particular in variable mesh are often very low order but that the approximation errors, images of the truncation error by the reverse of the linearised operator, are often of a larger order (in other words the order of the schema is better than the order of the truncation). We assume here that the order of the diagram will be at least as good

The FVM scheme writes ($f = cu$):

$$(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})\frac{\partial}{\partial t}\bar{u}_i = f_{i+\frac{1}{2}} - f_{i-\frac{1}{2}}.$$

with:

$$f_{i+\frac{1}{2}} = \frac{1}{2}(f_i + f_{i+1}) \quad ; \quad f_{i-\frac{1}{2}} = \frac{1}{2}(f_i + f_{i-1})$$

then:

$$(x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}})\frac{\partial}{\partial t}\bar{u}_i = \frac{1}{2}f_{i+1} - f_{i-1} \Leftrightarrow \frac{1}{2}(x_{i+1} - x_{i-1})\frac{\partial}{\partial t}\bar{u}_i = \frac{1}{2}(f_{i+1} - f_{i-1}).$$

The order of truncation will be the asymptotic order of the following residue:

$$R \;=\; \frac{2}{x_{i+1} - x_{i-1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \frac{\partial u}{\partial t}(x)dx - \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}}$$

For continuous solutions u and f.

Analysis of the mean of the derivative in time:

$$v = \frac{\partial u}{\partial t}$$

$$v(x) = u_i + (x - x_i)v_i' + \sum_{k=2}^{\infty} \frac{1}{k!}(x - x_i)^k v_i^{(k)}$$

$$\int^{x} v(\xi)d\xi = v_i\ x + v_i'\ \frac{1}{2}(x-x_i)^2 + \sum_{k=2}^{\infty}\frac{1}{k!}v_i^{(k)}\frac{1}{k+1}(x-x_i)^{k+1} + C$$

$$\int_{x_{i-1/2}}^{x_{i+1/2}} v(\xi)d\xi = v_i(x_{i+1/2}-x_{i-1/2}) + \sum_{k=1}^{\infty}\frac{1}{k!}v_i^{(k)}\frac{1}{k+1}[(x_{i+1/2}-x_i)^{k+1}-(x_{i-1/2}-x_i)^{k+1}]$$

that is:

$$d_i := \frac{1}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}}\int_{x_{i-1/2}}^{x_{i+1/2}} v(\xi)d\xi$$

$$d_i = v_i + \sum_{k=1}^{\infty}\frac{1}{k!}v_i^{(k)}\frac{1}{k+1}\frac{(x_{i+\frac{1}{2}}-x_i)^{k+1}-(x_{i-\frac{1}{2}}-x_i)^{k+1}}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}}$$

and writting the first terms:

$$d_i = v_i + v_i^{(1)}\frac{(x_{i+\frac{1}{2}}-x_i)^2-(x_{i-\frac{1}{2}}-x_i)^2}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}} + \frac{1}{2}v_i^{(2)}\frac{1}{3}\frac{(x_{i+\frac{1}{2}}-x_i)^3-(x_{i-\frac{1}{2}}-x_i)^3}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}} + r_i$$

with:

$$r_i = \sum_{k=4}^{\infty}\frac{1}{k!}v_i^{(k)}\frac{1}{k+1}\frac{(x_{i+\frac{1}{2}}-x_i)^{k+1}-(x_{i-\frac{1}{2}}-x_i)^{k+1}}{x_{i+\frac{1}{2}}-x_{i-\frac{1}{2}}}$$

We simplify with:

$$x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = \frac{1}{2}(x_{i+1}-x_{i-1})$$

$$x_{i+\frac{1}{2}} - x_i = \frac{1}{2}(x_{i+1}-x_i)$$

$$x_{i-\frac{1}{2}} - x_i = \frac{1}{2}(x_{i-1}-x_i)$$

$$d_i = v_i + v_i^{(1)}\frac{\frac{1}{4}(x_{i+1}-x_i)^2-\frac{1}{4}(x_{i-1}-x_i)^2}{\frac{1}{2}(x_{i+1}-x_{i-1})} + \frac{1}{2}v_i^{(2)}\frac{1}{3}\frac{\frac{1}{8}(x_{i+1}-x_i)^3-\frac{1}{8}(x_{i-1}-x_i)^3}{\frac{1}{2}(x_{i+1}-x_{i-1})} + r_i$$

$$d_i = v_i + \frac{1}{2} v_i^{(1)} \frac{(x_{i+1} - x_i)^2 - (x_{i-1} - x_i)^2}{x_{i+1} - x_{i-1}} + \frac{1}{8} v_i^{(2)} \frac{1}{3} \frac{(x_{i+1} - x_i)^3 - (x_{i-1} - x_i)^3}{x_{i+1} - x_{i-1}} + r_i$$

and remarking that:

$$x_{i+1} - x_{i-1} = (x_{i+1} - x_i) - (x_{i-1} - x_i)$$

we get:

$$d_i = v_i + \frac{v_i^{(1)}}{2}[(x_{i+1} - x_i) + (x_{i-1} - x_i)] + \frac{v_i^{(2)}}{24}[(x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2 + (x_{i+1} - x_i)(x_{i-1} - x_i)] + r_i$$

Remark : Uniform case: the first coefficient also writes:

$$(x_{i+1} - x_i) + (x_{i-1} - x_i) = (x_{i+1} - x_i) - (x_i - x_{i-1})$$

which vanishes if $x_{i+1} - x_i = x_i - x_{i-1}$. The central mean gives second-order accuracy on uniform mesh.□

Analysis of the spatial derivative:

Taylors series of f :

$$f_{i+1} = \quad f_i + (x_{i+1} - x_i) f_i' + \frac{(x_{i+1} - x_i)^2}{2!} f_i^{(2)} + \frac{(x_{i+1} - x_i)^3}{3!} f_i^{(3)} + ...$$

$$-f_{i-1} = -f_i - (x_{i-1} - x_i) f_i' - \frac{(x_{i-1} - x_i)^2}{2!} f_i^{(2)} - \frac{(x_{i-1} - x_i)^3}{3!} f_i^{(3)} + ...$$

$$f_{i+1} - f_{i-1} = (x_{i+1} - x_{i-1}) f_i' + \frac{f_i^{(2)}}{2!}[(x_{i+1} - x_i)^2 - (x_{i-1} - x_i)^2] + \frac{f_i^{(3)}}{3!}[(x_{i+1} - x_i)^3 - (x_{i-1} - x_i)^3] + ...$$

divising by $x_{i+1} - x_{i-1}$:

$$s_i := \frac{f_{i+1} - f_{i-1}}{x_{i+1} - x_{i-1}} = f_i' + \frac{f_i^{(2)}}{2!} \frac{(x_{i+1} - x_i)^2 - (x_{i-1} - x_i)^2}{x_{i+1} - x_{i-1}} + \frac{f_i^{(3)}}{3!} \frac{(x_{i+1} - x_i)^3 - (x_{i-1} - x_i)^3}{x_{i+1} - x_{i-1}} + ...$$

That is, with the simpification:

$$s_i = f'_i + \frac{f_i^{(2)}}{2!}[(x_{i+1}-x_i)+(x_{i-1}-x_i)] + \frac{f_i^{(3)}}{3!}[(x_{i+1}-x_i)^2+(x_{i-1}-x_i)^2+(x_{i+1}-x_i)(x_{i-1}-x_i)]+...$$

Truncation analysis:

We consider:

$$e_i = d_i + s_i - (\frac{\partial u}{\partial t})_i + (\frac{\partial cu}{\partial x})_i$$

using the fact that:

$$v_i = \frac{\partial u}{\partial t} = -cu'_i \quad ; \quad f_i = cu_i.$$

$$e_i/c = -\frac{u_i^{(2)}}{2}[(x_{i+1}-x_i)+(x_{i-1}-x_i)] - \frac{u_i^{(3)}}{24}[(x_{i+1}-x_i)^2+(x_{i-1}-x_i)^2+(x_{i+1}-x_i)(x_{i-1}-x_i)]$$
$$+\frac{u_i^{(2)}}{2!}[(x_{i+1}-x_i)+(x_{i-1}-x_i)] + \frac{u_i^{(3)}}{3!}[(x_{i+1}-x_i)^2+(x_{i-1}-x_i)^2+(x_{i+1}-x_i)(x_{i-1}-x_i)]+...$$

We observe that the first-order terms destroy mutually: the unsteady finite volume scheme centered remains of order two on variable mesh (provided that the derivation in time is also precise to order two).

In addition we have identified the first error term (of order two):

$$e_i/c = \frac{1}{2}u_i^{(3)}[(x_{i+1}-x_i)^2 + (x_{i-1}-x_i)^2 + (x_{i+1}-x_i)(x_{i-1}-x_i)].$$

Remark : uniform case:

$$e_i/c = \frac{1}{2}u_i^{(3)}(\Delta x)^2$$

to be compared with the *beta*-term de MUSCL. □

Orientation: To improve order conservatively the term e_i must be compensated by fluxes in $x_{i+1/2}$ and $x_{i-1/2}$ defined symmetrically with respect to neighboring points. These fluxes will be different on the left and on the right of i and will only be able to compensate approximately for the term e_i.

We can define:

$$q_i = 2e_i/(cu_i^{(3)}) = (x_{i+1} - x_i)^2 + (x_{i-1} - x_i)^2 + (x_{i+1} - x_i)(x_{i-1} - x_i)$$

In the case of uniform mesh, q_i is uniform, the term to correct is a third derivative of the flux (here from the unknown) which is easily evaluated by Stokes on an evaluation of the derivative second.

In the non-uniform case the term to be compensated e_i is not a derivative, we can make a decomposition between (1) a derivative to be processed (taking into account or not the term q_i) and (2) a remainder which will not be *conservative. We abandon this way.*