

Extension of ALE methodology to unstructured conical meshes

Benjamin Boutin, Erwan Deriaz, Philippe Hoch, Pierre Navaro

▶ **To cite this version:**

Benjamin Boutin, Erwan Deriaz, Philippe Hoch, Pierre Navaro. Extension of ALE methodology to unstructured conical meshes. ESAIM: Proceedings, 2011, 32, pp.32-55. 10.1051/proc/2011011 . hal-00777271

HAL Id: hal-00777271
https://hal.science/hal-00777271

Submitted on 5 Apr 2022

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

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



Distributed under a Creative Commons Attribution 4.0 International License

ESAIM: PROCEEDINGS, October 2011, Vol. 32, p. 31-55
E. Cancès, N. Crouseilles, H. Guillard, B. Nkonga, and E. Sonnendrücker, Editors

EXTENSION OF ALE METHODOLOGY TO UNSTRUCTURED CONICAL MESHES

Benjamin Boutin[1], Erwan Deriaz[2], Philippe Hoch*,[3] and Pierre Navaro[4]

Abstract. We propose a bi-dimensional finite volume extension of a continuous ALE method on unstructured cells whose edges are parameterized by rational quadratic Bezier curves. For each edge, the control point possess a weight that permits to represent any conic (see for example [LIGACH]) and thanks to [WAGUSEDE, WAGU], we are able to compute the *exact area* of our cells. We then give an extension of scheme for remapping step based on volume fluxing [MARSHA] and self-intersection flux [ALE2DHAL]. For the rezoning phase, we propose a three step process based on moving nodes, followed by control point and weight re-adjustment. Finally, for the hydrodynamic step, we present the GLACE scheme [GLACE] extension (at first-order) on conic cell using the same formalism. We only propose some preliminary first-order simulations for each steps: Remap, Pure Lagrangian and finally ALE (rezoning and remapping).

Résumé. Nous proposons une extension volumes finis bi-dimensionnelle d'une méthode ALE continue sur des cellules non structurées dont les bords sont paramétrés par des courbes de Bézier quadratiques rationnelles. Pour chaque arête, le point de contrôle possède un poids qui permet de représenter n'importe quelle conique [LIGACH] et grâce à [WAGUSEDE, WAGU], nous pouvons calculer l'*aire exacte* de nos cellules. Pour la phase de remapping, on donne l'extension de deux schéma, l'un basé sur le calcul de flux de volumes [MARSHA] et l'autre par flux avec auto-intersection [ALE2DHAL]. Pour la phase de lissage de maillage, nous proposons un processus en trois étapes basées sur le déplacement des noeuds, suivi de celui des points de contrôle puis finalement du réajustement du poids. Enfin, pour la phase hydrodynamique, on présente l'extension du schéma GLACE [GLACE] (à l'ordre un) sur les cellules coniques en utilisant le même formalisme. Nous montrons seulement des simulations préliminaires à l'ordre 1 sur chaque étape : Remap, Lagrange pur et ALE (rezoning et remapping).

Introduction

The prototype system of interest is the Euler system written in flux forms. Written over a frame moving with an arbitrary speed $\mathbf{V}$, the evolution of volume, mass, momentum and total energy on a generic cell $C(t)$ writes (see [DUVAUT] [ALED] [EQU] [GLACE2D] [EUCCLHYD] [HdrPhMaire]):

$$\left\{\begin{array}{l} D_t \int_{C(t)} 1\, dv - \int_{\partial C(t)} \mathbf{V}.\hat{\mathbf{N}}\, ds = 0, \\ D_t \int_{C(t)} \rho\, dv + \int_{\partial C(t)} \rho(\mathbf{U}-\mathbf{V}).\hat{\mathbf{N}}\, ds = 0, \\ D_t \int_{C(t)} \rho\mathbf{U}\, dv + \int_{\partial C(t)} (\rho\mathbf{U}\otimes(\mathbf{U}-\mathbf{V}) + P)\hat{\mathbf{N}}\, ds = 0, \\ D_t \int_{C(t)} \rho E\, dv + \int_{\partial C(t)} (\rho E(\mathbf{U}-\mathbf{V}) + P\mathbf{U}).\hat{\mathbf{N}}\, ds = 0, \qquad P(\rho,\epsilon),\ \epsilon \text{ internal energy}, \epsilon = E - \frac{1}{2}|\mathbf{U}|^2. \end{array}\right. \tag{1}$$

[1] IRMAR, Université de Rennes 1, Campus de Beaulieu, 35042 Rennes Cedex, France
[2] M2P2 - UMR-6181 CNRS IMT La Jetée Technopôle de Château-Gombert, 38 Rue Frédéric Joliot-Curie, 13451 MARSEILLE Cedex 20, France
[3] , * Corresponding author: CEA, DAM, DIF, Bruyères-le-Chatel, F-91297 Arpajon Cedex, France
[4] IRMA, 7 rue René Descartes, 67084 Strasbourg Cedex, France

© EDP Sciences, SMAI 2011

Article published online by EDP Sciences and available at http://www.esaim-proc.org or http://dx.doi.org/10.1051/proc/2011011

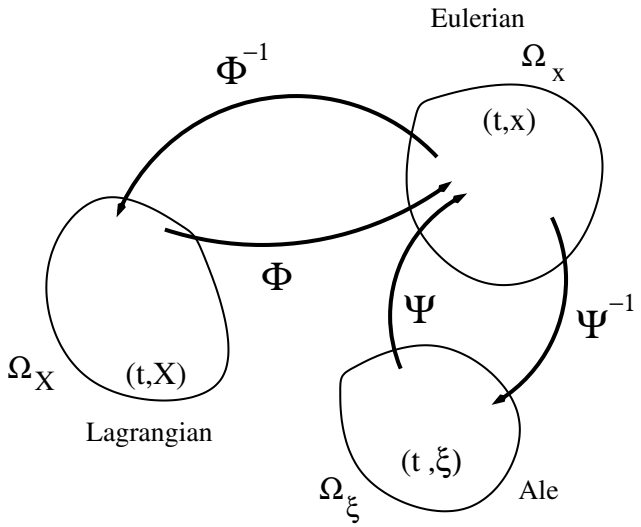


FIGURE 1. Different coordinate system at same time t, $x = \Phi(t, X)$ and $x = \Psi(t, \xi)$. The trajectory of a particule in Ω_x is defined by $\frac{d}{dt}x = V(t, x)$ (with $x(0) = \xi$).

Where D_t is defined by $D_t \int_{C(t)} \phi \, dv = \int_{C(t)} \partial_t \phi \, dv + \int_{\partial C(t)} \phi \mathbf{V}.\hat{\mathbf{N}} \, ds$ (where $\mathbf{V}$ and ϕ are written with Eulerian coordinates). The speed $\mathbf{V}$ is decomposed into a physical and grid velocities: $\mathbf{V} = \mathbf{U} - \mathbf{U}_g$, so the resulting system is solved by a splitting separating fluid and grid motions.
In a first purely Lagrangian step $\mathbf{V} = \mathbf{U}$, only pressure appears in physical thermodynamical flux:

$$\begin{cases} D_t^{\text{MAT}} \int_{C(t)} 1 \, dv - \int_{\partial C(t)} \mathbf{U}.\hat{\mathbf{N}} \, ds = 0, \\ D_t^{\text{MAT}} \int_{C(t)} \rho \, dv = 0, \\ D_t^{\text{MAT}} \int_{C(t)} \rho \mathbf{U} \, dv + \int_{\partial C(t)} P\hat{\mathbf{N}} \, ds = 0, \\ D_t^{\text{MAT}} \int_{C(t)} \rho E \, dv + \int_{\partial C(t)} P\mathbf{U}.\hat{\mathbf{N}} \, ds = 0. \end{cases} \tag{2}$$

We recall that (2) is the semi-Lagrangian (or updated Lagrangian) form of Euler system, where all the spatial quantities are written with Eulerian coordinates like (1), it corresponds to a local differential operator at point (t,x) (x Eulerian coordinate):

$$\begin{array}{l} d_t^{\text{MAT}} \phi = \frac{\partial \phi}{\partial t} + \mathbf{U}.\nabla \phi, \\ d_t^{\text{MAT}} (\phi J) = J \left(\frac{\partial \phi}{\partial t} + \nabla.(\phi \mathbf{U}) \right), \\ \text{J determinant of the Jacobian of Euler/Lagrange transformation.} \\ D_t^{\text{MAT}} \int_{C(t)} \phi \text{dx} = \int_{C(0)} \left(J(d_t^{\text{MAT}} \phi) + \phi \partial_t J \right) \text{dX}, \qquad \text{X} \quad \text{Lagrangian} \quad \text{coordinate.} \end{array} \tag{3}$$

The second part of the splitting $\mathbf{V} = -\mathbf{U}_g$ gives the system of conservative advection equations:

$$\begin{cases} D_t^{\text{ADV}} \int_{C(t)} 1 \, dv + \int_{\partial C(t)} \mathbf{U}_g.\hat{\mathbf{N}} \, ds = 0, \\ D_t^{\text{ADV}} \int_{C(t)} \rho \, dv + \int_{\partial C(t)} \rho \mathbf{U}_g.\hat{\mathbf{N}} \, ds = 0, \\ D_t^{\text{ADV}} \int_{C(t)} \rho \mathbf{U} \, dv + \int_{\partial C(t)} (\rho \mathbf{U}) \mathbf{U}_g.\hat{\mathbf{N}} \, ds = 0, \\ D_t^{\text{ADV}} \int_{C(t)} \rho E \, dv + \int_{\partial C(t)} \rho E \mathbf{U}_g.\hat{\mathbf{N}} \, ds = 0. \end{cases} \tag{4}$$

D_t^{ADV} is associated to grid velocity. Generally such systems are discretized on polygonal type cells. However, some problems involve accurate geometric description such as curved boundary of the domain itself, discretization of a circular initial condition. Of course, it exists also non-stationary problem that will certainly be better approximated if geometry of control volumes does possess non null curvature (circles for symmetries on polar radial flows, interface curve between two materials, multi-material interfaces in mixed

cells, etc ...).
We want to extend this finite volume formalism (and then obtain higher genericity in geometric modeling) to cells with weaker hypothesis on edge representation. Generally, for cells with three of four edges, the curvilinear edges parameterization can be obtained by high order finite element such as P_2, Q_2 for Lagrangian higher order (at least third order) hydrodynamics ; see also [FEV, SHUSHEN, RIB, NEWEX]. In our case, we deal with arbitrary number of edges per cells and we apply a full ALE methodology not only to the Lagrangian hydrodynamic part (2) but also for solving (4) with $\mathbf{U}_g \neq 0$.
The paper is organized as follows, in the first section we make some essential description and geometric properties of the quadratic rational Bezier curves that are used in other sections. Specially, we recall the exact flux formula in [WAGUSEDE, WAGU] used to compute the area of a cell with an arbitrary number of conic edges, we then reinterpret the area in terms of nodal/control point contributions like in GLACE formalism [GLACE] giving a first step toward the design of hydrodynamic conic GLACE scheme. The second section is a direct application of such area formula to extend the remapping step of ALE method by computing area swept by edges displacement [MARSHA]. For the self intersection based flux, the formula of [WAGUSEDE, WAGU] are not sufficient because we need an evaluation of the flux over an arbitrary value of the curve parameter, we then explain our extension. Some numerical examples of first-order scheme are presented. The third section is the rezoning step of ALE on conic cells for which we propose to move the control points after moving the nodes with standard (polygonal) algorithm. The control points are adjusted in such a way that a conic cell is valid if the underlying logical polygonal is also. Moreover making the weight of control points a (geometric) variable, the type of conic may change in this process. In the fourth section, we construct the first-order GLACE hydrodynamic scheme, and show some numerical examples for pure Lagrangian and ALE tests in the last section.

1. Computing exact area for arbitrary conic cells

In this part, we want to obtain an exact formula for the computation of area for arbitrary polygonal curved cells with any edge number. This is of course the first step to build any finite volume method. Let $\mathbf{X}$ in $\mathbb{R}^2$ be the position of a point inside $\bar{C}(t)$, then

$$|C(t)| = \int_{C(t)} 1\, dv = \int_{C(t)} \frac{1}{2} \operatorname{div} \mathbf{X}\, dv = \frac{1}{2} \int_{\partial C(t)} \mathbf{X}.\hat{\mathbf{N}}\, ds, \quad \hat{\mathbf{N}} \text{ unit normal, } s \text{ curvilinear abscissa,} \tag{5}$$

where by definition s verifies $|\frac{d}{ds}\mathbf{X}| = 1$ in (5). In the case where $\partial C(t)$ can be continuously decomposed into ne edges, each one being parameterized by a curve $q \in [0,1] \mapsto \mathbf{X}^e(q,t) \in \mathbb{R}^2$. We impose the continuity $\mathbf{X}^e(1,t) = \mathbf{X}^{e+1}(0,t)$ (cell's boundary is closed $\mathbf{X}^{ne}(1,t) = \mathbf{X}^1(0,t)$) and the orientation to be trigonometric. In the general case, q does not satisfy $|\frac{d}{dq}\mathbf{X}| = 1$ but to get an admissible parameterization, we need to impose $|\frac{d}{dq}\mathbf{X}| \neq 0$ everywhere on the curve. For this re-parameterization, formula (5) then writes:

$$|C(t)| = \frac{1}{2} \sum_{e=1}^{ne} \left(\int_0^1 \mathbf{X}^e . \left(\tfrac{d}{dq}\mathbf{X}^e\right)^{\perp} dq \right). \tag{6}$$

We emphasize that at this level, our constraints are the following:

1) Compute exactly this integral for a parameterization including straight polygonal, circular and parabolic cells as special cases.
2) Obtain a formula involving the control points in an explicit analytical form to get a formalism of [GLACE].

We now describe the parameterization that fulfills these two constraints.

1.1. Rational quadratic-Bezier parameterization

For computer graphic and applied geometry communities, it is a well known result that any conic segment can be expressed by a rational quadratic Bezier curve, which is a special kind of NURBS (Non Uniform

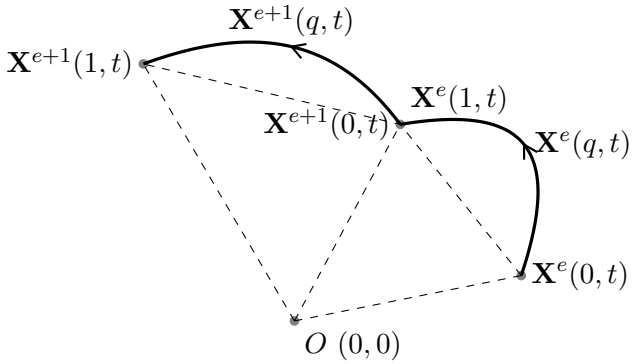


FIGURE 2. Decomposition of boundary integral for area computation.

Rational B-Spline). Straight segment to any conic section can be represented by this kind of parameterization (see Figure 3 left):

$$M^{\omega}(q) = \frac{M_0(1-q)^2 + M_1 2\omega q(1-q) + M_2 q^2}{(1-q)^2 + 2\omega q(1-q) + q^2}, \qquad q \in [0,1]. \tag{7}$$

The two parameters (M_1, ω) are the control point and the associated weight (here $\omega \geq 0$) for a given logical edge (M_0, M_2).

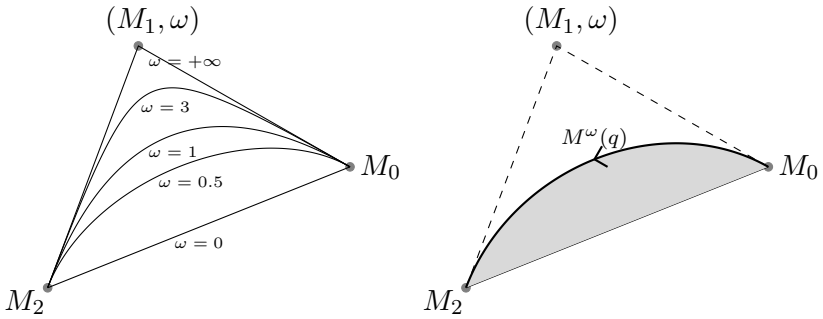


FIGURE 3. Left: Evolution of parameterization with respect to positive values of the parameter ω (*all conic curves can be continuously represented*), see property (4) below. Right: Conic section representation with rational quadratic-Bezier parameterization for edges, filled region needs to be exactly computed.

We recall some properties presented in [LIGACH], that we will use in the following sections:

(1) The control simplex (M_0, M_1, M_2) is none degenerate if and only if M_1 is not aligned to logical edge (M_0, M_2).
(2) Convex hull: the curved segment $\{M^{\omega}(q), q \in [0,1]\}$ lies inside the control simplex for $\omega \geq 0$.
(3) Endpoints interpolation:

$$\begin{array}{rclrcl} M^{\omega}(0) & = & M_0, & M^{\omega}(1) & = & M_2, \\ \frac{d}{dq}M^{\omega}(0) & = & 2\omega \mathbf{M_0 M_1}, & \frac{d}{dq}M^{\omega}(1) & = & 2\omega \mathbf{M_1 M_2}. \end{array}$$

Hence $M^{\omega}(q)$ passes through M_0 and M_2 and for $\omega > 0$ the two straight lines passing at these points respectively with directions $\frac{d}{dq}M^{\omega}(0)$ and $\frac{d}{dq}M^{\omega}(1)$ intersect at the control point M_1.

(4) Conic type parameter, see Figure 3 left:

(a) $\omega = 0$: degenerate straight segment.
(b) $0 < \omega < 1$: ellipse segment, and in the case where M_1 is on the perpendicular bisector of (M_0, M_2) and when $\omega := \cos(\frac{\theta}{2})$, circle segment is recovered, see Figure 4

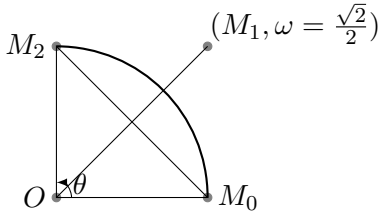


FIGURE 4. Circle recovered by special values of the couple (M_1, ω), if O is the local "origin" attached to edge : $OM_1 = \frac{R}{\omega}(\cos(\frac{\theta}{2}), \sin(\frac{\theta}{2}))$, $R = |OM_0| = |OM_2|$.

(c) $\omega = 1$: parabolic segment.
(d) $\omega > 1$: hyperbolic segment, and when $\omega \to \infty$ the conic segment tends to go through M_1.

(5) Shoulder point, see Figure 5: $S{:=}M^\omega(q = 1/2)$ is the middle point of the curved segment, it can also be computed by:

$$S = \frac{1}{2}(Q_0 + Q_2), \text{ where } \quad Q_0 = \frac{1}{1+\omega}(M_0 + \omega M_1), \quad Q_2 = \frac{1}{1+\omega}(\omega M_1 + M_2). \tag{8}$$

The tangent line of $M^\omega(q)$ at S passes through Q_0 and Q_2 and is parallel to (M_0, M_2), moreover S is the point of the curve that maximizes the distance between the curve and (M_0, M_2).

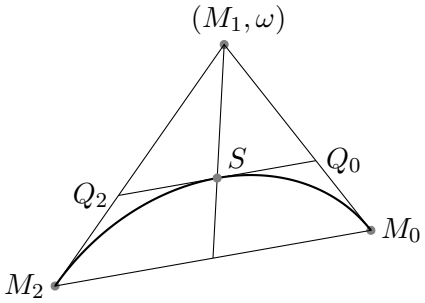


FIGURE 5. Shoulder point of rational quadratic Bezier curve (7).

(6) Implicit form: the implicit equation for $M^\omega(q) = (x^\omega(q), y^\omega(q))$ writes

$$\lambda_1(x,y)^2 - 4\omega^2\lambda_0(x,y)\lambda_2(x,y) = 0, \tag{9}$$

where $\{\lambda_i(x,y)\}_{i=0,1,2}$ correspond to the barycentric coordinates of the point (x,y) with respect to the control simplex (M_0, M_1, M_2).

1.2. **Edge based formula**

Computing (6) with rational quadratic Bezier parameterization (7) writes:

$$I = \frac{1}{2}\int_0^1 (x^\omega(q)\frac{d}{dq}y^\omega(q) - \frac{d}{dq}x^\omega(q)y^\omega(q))\,dq, \tag{10}$$

where $(x^\omega(q), y^\omega(q))$ are the cartesian coordinates of the point $M^\omega(q)$. This integral can be decomposed in two parts (see Figure 2):

(1) the area between the curve $M^\omega(q)$ and the segment (M_0, M_2) (see Figure 3 right). We denote $A(M^\omega(q), M_0, M_1, M_2)$ this (exact) area and $A^T(M_0, M_1, M_2)$ the (exact) area of the straight edges control simplex.
(2) the area of the simplex (O,M_0,M_2) with straight edges: $A^T(O, M_0, M_2)$.

In [WAGU], the author gives a way to compute it when $M_0 = (0, 0)$ and $M_2 = (x_2, 0)$ so that we need to add this term to compute the integral (10), hence for arbitrary M_0 and M_2:

$$I = A(M^\omega(q), M_0, M_1, M_2) + A^T(O, M_0, M_2) = A(M^\omega(q), M_0, M_1, M_2) + \frac{1}{2}\det(\mathbf{OM_0}, \mathbf{OM_2}). \tag{11}$$

We now recall the results in [WAGU] to compute $A(M^\omega(q), M_0, M_1, M_2)$. For each kind of conic ($\omega = 0$ to $\omega \to \infty$), there exists a function $f(\omega)$ such that:

$$A(M^\omega(q), M_0, M_1, M_2) = f(\omega)\ A^T(M_0, M_1, M_2). \tag{12}$$

We give the details of $f(\omega)$ (see Figure 6):

$$\begin{cases} (1)\ \text{if } \omega = 0,\ \text{then } f(\omega) = 0, \\ (2)\ \text{if } 0 < \omega < 1,\ \text{then } f(\omega) = \frac{2\omega}{1-\omega^2}\left(\frac{1}{\sqrt{1-\omega^2}}\text{atan}(\sqrt{\frac{1-\omega}{1+\omega}}) - \frac{\omega}{2}\right), \\ (3)\ \text{if } \omega = 1,\ \text{then } f(\omega) = \frac{2}{3}, \\ (4)\ \text{if } \omega > 1,\ \text{then } f(\omega) = \frac{\omega}{\omega^2-1}\left(\omega + \frac{1}{\sqrt{\omega^2-1}}\log(\omega - \sqrt{\omega^2-1})\right). \end{cases} \tag{13}$$

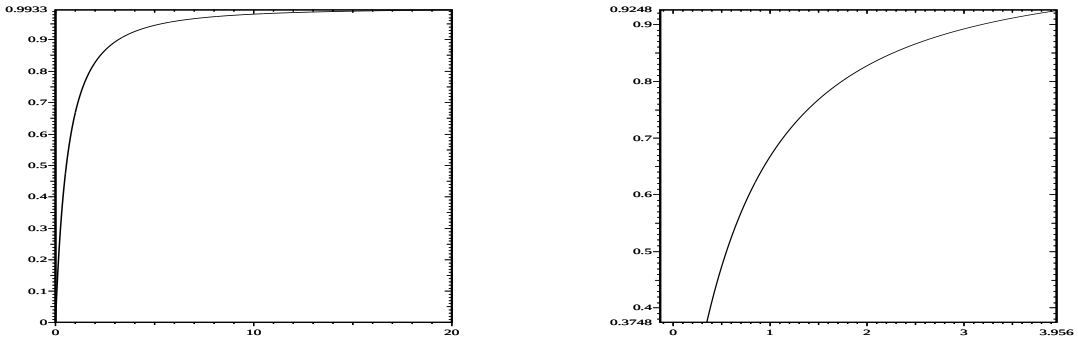


FIGURE 6. Left: graph of the nonlinear function $f(\omega)$ weighting the area of simplex (M_0, M_1, M_2) in formula (12)(13), in abscissa weight ω. Right: zoom near $(\omega = 1, f(1) = \frac{2}{3})$.

Remark 1. (1) *f(w) is a continuous strictly increasing function on* $[0, \infty[$ *with value on* $[0, 1[$.
(2) *Using relation (12), we give the cell area with arbitrary number of edges, where* r *and* $r+1$ *are the logical endpoint and* $r + 1/2$ *is the control point.*

$$|C_j| = \sum_{(r,r+1)} sA(M^\omega_{(r,r+1)}(q), M_r, M_{r+1/2}, M_{r+1}) + \frac{1}{2}(x_r y_{r+1} - x_{r+1} y_r), \tag{14}$$

where s is equal to -1 or +1, depending on the orientation of the control point on edge.
If M_r *in Figure 7 corresponds to* $M_0(e)$ *in Figure 3 then s=1, else (if it corresponds to* $M_2(e)$ *in Figure 3) then s=-1;*

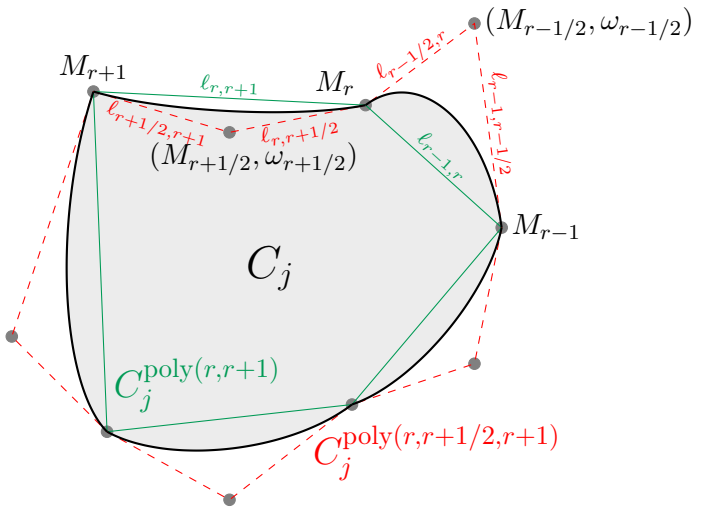


FIGURE 7. Generic cell C_j and its inclusion in the convex hull of logical polygonal with (in red dashed line) and without (in green polyline) control point (here $\ell_{a,b}$ is the euclidean distance between point a and b).

1.3. **Nodal based formula**

In this part, we want to express the formula (14) in terms of degrees of freedom (M_r, M_{r+1} and $(M_{r+1/2},\omega)$) defining the boundary of cell. We recall that in the case of polygonal (straight) cells, the

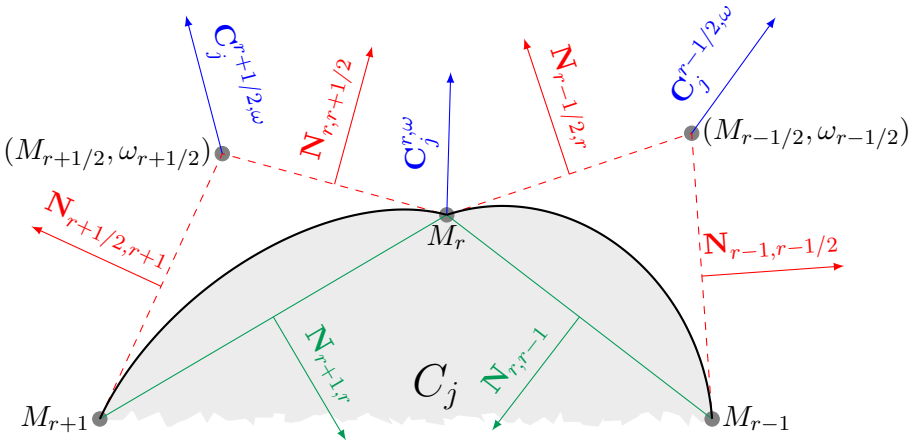


FIGURE 8. Decomposition of vector $\mathbf{C}_j^{\text{dof},\omega}$ at degree of freedom (for short: dof = nodes and control points) for conic cells, $\mathbf{C}_j^{\text{dof},\omega} = \nabla_{M_{\text{dof}}}|C_j|$.

volume of cell can be expressed by (see e.g. [GLACE]):

$$|C_j| = \frac{1}{2}\sum_r \mathbf{C}_j^r.\mathbf{OM}_r \tag{15}$$

where $\mathbf{C}_j^r$ are geometric vectors defined at node r by: $\frac{1}{2}(\ell_{r-1,r}\hat{\mathbf{N}}_{r-1,r} + \ell_{r,r+1}\hat{\mathbf{N}}_{r,r+1})$, where vectors $\hat{\mathbf{N}}_{r-1,r}$, $\hat{\mathbf{N}}_{r,r+1}$ are normal unit vectors, $\mathbf{N}_{r-1,r}$, $\mathbf{N}_{r,r+1}$ are non-normalized normal vectors (of length $\ell_{r-1,r}$, $\ell_{r,r+1}$), they correspond to the opposite $\mathbf{N}_{r,r-1}$, $\mathbf{N}_{r+1,r}$ in Figure 8.

Another property is that $\mathbf{C}_j^r = \nabla_{M_r}|C_j|$ using homogeneity (with respect to nodes) of discrete formula giving area $|C_j| = Area(M_1, ..., M_{ne})$, $Area$ is such that $Area(\lambda M_1, ..., \lambda M_{ne}) = \lambda^d Area(M_1, ..., M_{ne})$,(d is the space dimension).
Now, for cells with conic edges, we get the following.

Proposition 2. (1) *The formula of area expressed by edges contribution in (14) can be rewritten using contribution of nodes, control points and weights (M_r and $M_{r+1/2}$ and $\omega_{r+1/2}$) see Figure 8, by extension of (15) into:*

$$|C_j| = \frac{1}{2}\left(\sum_r \mathbf{C}_j^{r,\omega}.\mathbf{OM}_r + \sum_{r+1/2} \mathbf{C}_j^{r+1/2,\omega}.\mathbf{OM}_{r+1/2}\right) := \frac{1}{2}\sum_{\text{dof}} \mathbf{C}_j^{\text{dof},\omega}.\mathbf{OM}_{\text{dof}} \tag{16}$$

(2) *The expression of $\mathbf{C}_j^{r,\omega}$ and $\mathbf{C}_j^{r+1/2,\omega}$ are described below using notation and results of (13):*

$$\mathbf{C}_j^{r,\omega} = \frac{1}{2}\left((1-f(\omega_{r-1/2}))\mathbf{N}_{r-1,r} + (1-f(\omega_{r+1/2}))\mathbf{N}_{r,r+1} + f(\omega_{r-1/2})\mathbf{N}_{r-1/2,r} + f(\omega_{r+1/2})\mathbf{N}_{r,r+1/2}\right), \tag{17}$$

$$\mathbf{C}_j^{r+1/2,\omega} = \frac{f(\omega_{r+1/2})}{2}(\mathbf{N}_{r,r+1/2} + \mathbf{N}_{r+1/2,r+1}). \tag{18}$$

Moreover, we have the following identities:

$$\begin{cases} \text{For each cell } C_j: & \sum_r \mathbf{C}_j^{r,\omega} + \sum_e \mathbf{C}_j^{r+1/2,\omega} = 0, \\ \text{for each internal node } M_r: & \sum_j \mathbf{C}_j^{r,\omega} = 0, \text{ for two adjacent cells } j \text{ and } k: \ \mathbf{C}_j^{r+1/2,\omega} + \mathbf{C}_k^{r+1/2,\omega} = 0. \end{cases} \tag{19}$$

Remark 3. *This is a continuous ω-GLACE formalism going from $\mathbf{C}_j^{\text{poly}(r,r+1)}$ to $\mathbf{C}_j^{\text{poly}(r,r+1/2,r+1)}$ cell, see Figure 7:*

1. *For $\omega = 0$, $\mathbf{C}_j^{r+1/2,\omega} = 0$, $\mathbf{C}_j^{r,\omega} = \frac{1}{2}(\mathbf{N}_{r-1,r} + \mathbf{N}_{r,r+1})$, so that for the last, we recover the classical $\mathbf{C}_j^r$ of [GLACE] on polygonal cells $\mathbf{C}_j^{\text{poly}(r,r+1)}$.*
2. *For $\omega \to \infty$, $\mathbf{C}_j^{r+1/2,\omega} \to \frac{1}{2}(\mathbf{N}_{r,r+1/2} + \mathbf{N}_{r+1/2,r+1})$ and $\mathbf{C}_j^{r,\omega} \to \frac{1}{2}(\mathbf{N}_{r-1/2,r} + \mathbf{N}_{r,r+1/2})$, so that for both, we also recovered classical $\mathbf{C}_j^r$ of [GLACE] on polygonal cells $\mathbf{C}_j^{\text{poly}(r,r+1/2,r+1)}$.*

Hence the continuity of $f(\omega)$ over $\mathbb{R}^+$ and properties (2) and (3) in section 1.1 imply that for $\omega \geq 0$, $\mathbf{C}_j^{r+1/2,\omega}$ and $\mathbf{C}_j^{r,\omega}$ are varying continuously inside polygonal bounds defined by the values (1)(2) above.

2. Conservative flux based remapping on conic cells

We now focus on the numerical approximation of conservative advection equation see (4):

$$D_t \int_{C(t)} Q\, dv + \int_{\partial C(t)} Q\mathbf{U}_g.\hat{\mathbf{N}}\, ds = 0. \tag{20}$$

where $\mathbf{U}_g(t,x)$ is a given continuous velocity and $Q = (1, \rho, ...)$ the conservative unknowns.
Integration of (20) over a time interval (t^n, t^{n+1}) gives

$$|C(t^{n+1})|\bar{Q}^{n+1} = |C(t^n)|\bar{Q}^n - \int_{t^n}^{t^{n+1}} \left(\int_{\partial C(t)} Q\mathbf{U}_g.\hat{\mathbf{N}}\, ds\right) dt, \tag{21}$$

where $\bar{Q}(t) = \frac{1}{|C(t)|}\int_{C(t)} Q dv$, and for any t, all volumes $|C(t)|$ are supposed to be strictly positive.
We recall that continuous conservative advection in ALE may be done by rezoning (which corresponds to possess a "better" mesh at time t^{n+1} and therefore to know $|C(t^{n+1})|$ in (21)) plus remapping phase.
The later needs to define new finite volume value $\bar{Q}^{n+1}$ on a grid for which we have done the following operation.

Definition 4. *The extended continuous rezoning is defined by a three steps process (see Figure 10):*

(1) *Nodes of mesh can move.*
(2) *Control point of edges can move.*
(3) *Control point weight can change $\omega(t)$ (conic segment type may change).*

Remark 5. *To take into account these three aspects, the speed in (20) must be decomposed into*

$$\mathbf{U}_g = \mathbf{U}_{\text{displacement}} + \mathbf{U}_{\text{representation}}. \tag{22}$$

The exact geometric problem see Figure 9 involves an exact conic/conic mesh intersection at each time step, even if feasible, it is nearly hopeless in 3D and too expensive in any case. Moreover, this complex exact geometric problem is not necessary to obtain a conservative scheme that satisfies a local maximum principle (bound preserving).

We propose to extend two first-order local volume fluxing methods presented respectively in [MARSHA]

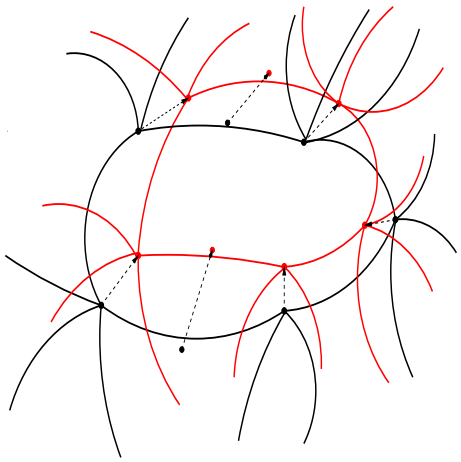


FIGURE 9. Exact geometric problem: computation of conic/conic mesh intersection between old mesh (black) and new mesh (red) after moving nodes, control points and weight re-adjustment.

and [ALE2DHAL] that approximate (21).

2.1. Single zone edge based flux

The first-order extension of [MARSHA] is:

$$Q_j^N |C_j|^N = Q_j^O |C_j|^O + \sum_e Q_e^* \delta V_e, \tag{23}$$

where index O (resp. N) stands for Old (resp. New), δV_e is the algebraic area swept by edge displacement and Q_e^* the flux built in an upwind way. Note that to compute this area we just apply (14) or (16) over a four edges cell (see Figure 10).

We can establish a maximum principle (local bound preserving scheme) for first-order scheme if we have the CFL-like condition:

$$|C_j|^O + \sum_{e, \delta V_e < 0} \delta V_e \geq 0. \tag{24}$$

If (24) is not fulfilled for a given cell j, we apply sub-cycling, until this constraint is reached (it does always converge), this constraint ensures that Q_j^N in (23) is a convex combination of neighboring old values Q_k^O.

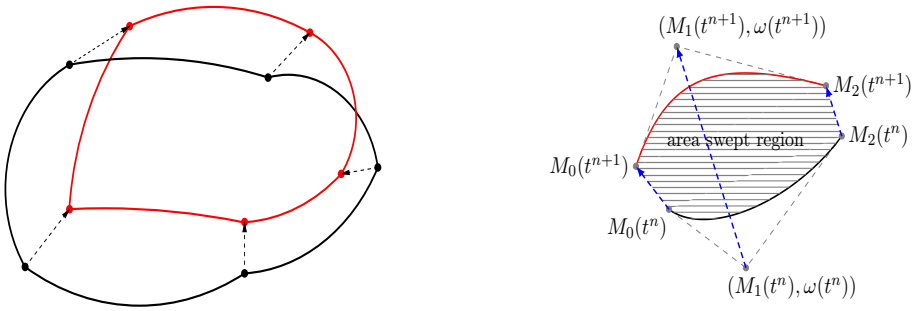


FIGURE 10. Left: Old cell (black) moves to new cell (red); Right: The area swept by edges forms conic cells which can be computed exactly by (14) or (16).

2.1.1. *Numerical basic test cases*

Now, we show some numerical examples. In Figures 11 and 12, we do only one remapping step of characteristic function, each one has the same displacement of vertices, only the weight is different ($\mathbf{U}_{\text{displacement}} = \mathbf{0}$ and $\mathbf{U}_{\text{representation}} \neq \mathbf{0}$ see (22)). In Figure 13, we apply a large displacements in such a way that (24) is not satisfied. In this case, we need 4 sub-cycling to guarantee positivity, note also that ($\mathbf{U}_{\text{displacement}} \neq \mathbf{0}$ and $\mathbf{U}_{\text{representation}} \neq \mathbf{0}$ see (22)).

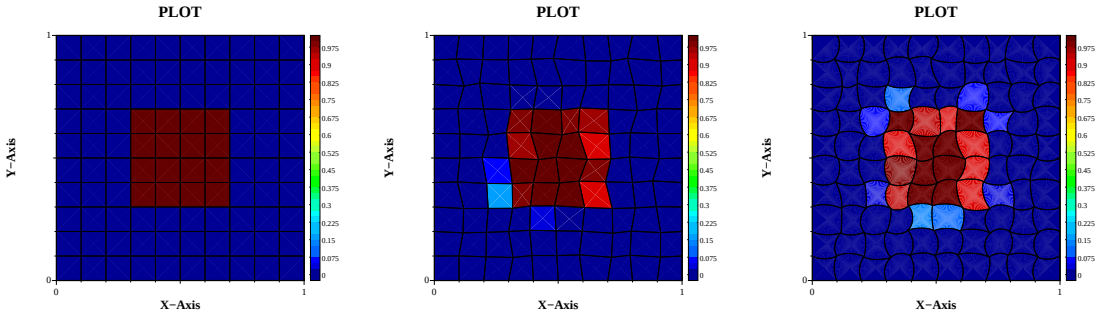


FIGURE 11. Left: Initial data and mesh, Final meshes and solutions of first-order remapping scheme (23) for polygons ($\omega = 0$) on center and ellipse ($\omega = 0.5$) on the right.

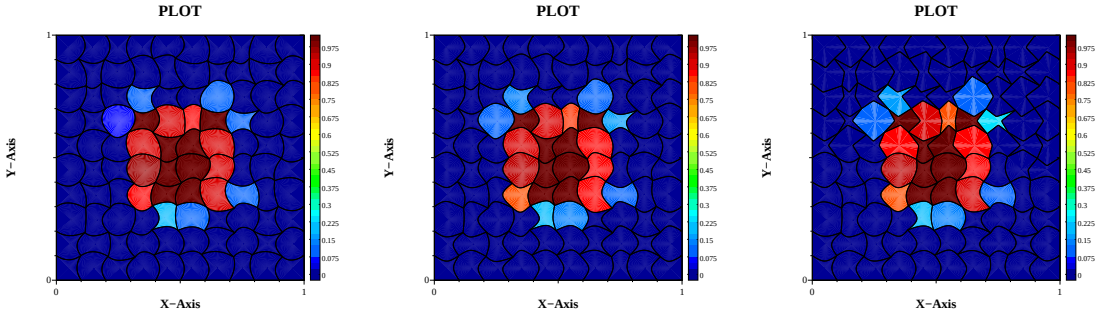


FIGURE 12. Final meshes and solutions of first-order remapping scheme (23) with initial data and mesh Figure 11. From left to right, parabola ($\omega = 1$), hyperbola ($\omega = 2$), mixed.

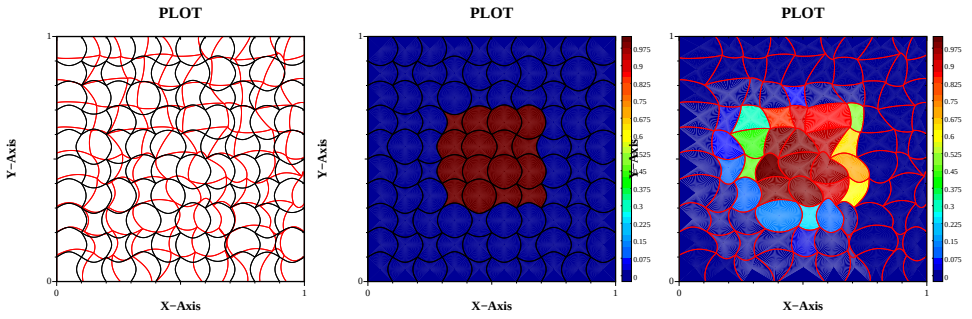


FIGURE 13. Large displacement field test for first-order fluxing scheme (23). Left: superposition of meshes before/after (black/red) rezoning. Center: Initial mesh and data. Right: Final mesh (with modification of weights ω) and solution. Here sub-cycling is needed (4 cycles) because of large displacement, nevertheless maximum principle is still maintained.

2.1.2. *Preservation of circular symmetry for non equi-angular sector*

In this test case, the radial geometry is such that $(r,\theta) \in [0,1]\times[0,\pi/2]$. The initial circular mesh is made by 4 sectors and 10 equidistant layers. The i'th angular sector is computed by:

$$\theta(i) = \frac{i\pi(1-\cos(\frac{i\pi}{2ns}))}{2ns}, \qquad \text{where } ns = 4. \tag{25}$$

To obtain circles, we first set the weights $w = 0$ for those edges aligned to radial direction and $w = \cos(\frac{(\theta(i+1)-\theta(i))}{2})$ with control point $M_1 = \frac{R}{w}(\cos(\frac{(\theta(i+1)+\theta(i))}{2}), \sin(\frac{(\theta(i+1)+\theta(i))}{2}))$ for the others (see Figure (4)). In this advection test case, we impose that the nodal displacement is done only in the radial direction, and we re-adjust the control points in such a way that circles are recovered. The initial condition is constant in each sector. For i'th sector $u^0 = \frac{(\theta(i+1)+\theta(i))}{2}$, see Figure (14).

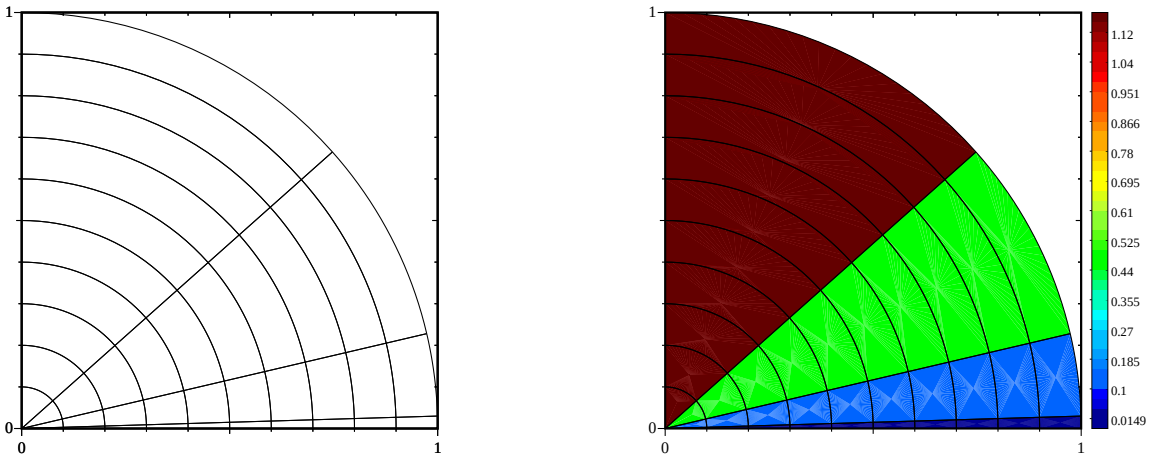


FIGURE 14. Initial circular mesh with non equi-angular sector (left) and initial condition (right).

We recall that circular heaviside function $1_{r<R_0}$ or $1_{ax^2+by^2+cxy+dx+ey+f<R_0}$ can be exactly represented on our conical meshes.

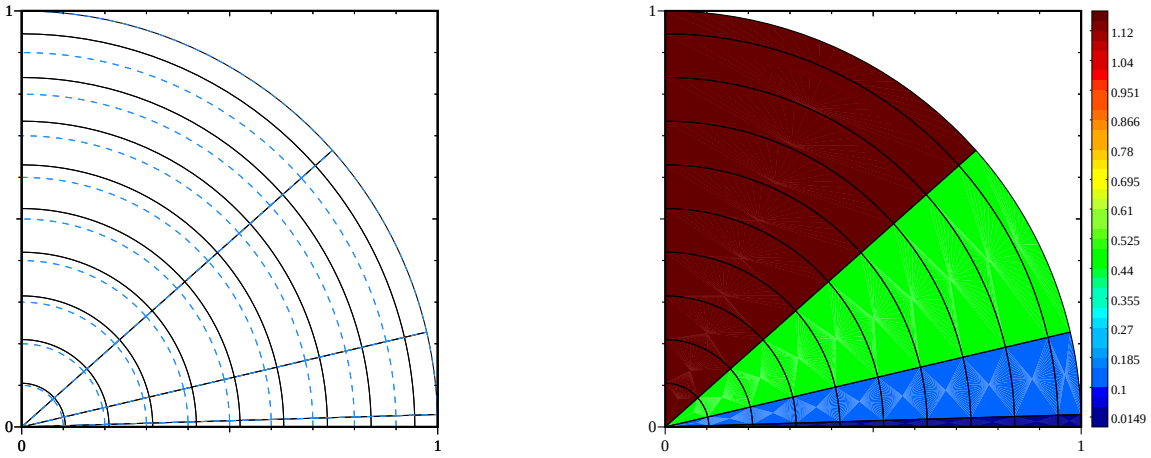


FIGURE 15. Left: Final mesh obtained after 5 radial displacement (solid) and initial mesh (dashed), Right: Final solution, we recover exactly the initial data for each layer/sector.

We emphasize that in our case, in such radial circular test case, we have no geometric error that appear on straight polygonal mesh.

2.2. **Self-intersection edge based flux**

In this section, we want to extend the self-intersection flux based on polygonal cells [ALE2DHAL]. This is a refinement of the previous flux (23) based on sub-zone area computation. In case of an edge e rotating on itself during rezoning, see Figure 16, two sub-areas $\delta V_{e,1}, \delta V_{e,2}$ are created, with $\delta V_{e,1}\delta V_{e,2} < 0$.
The first-order extension of [ALE2DHAL] is:

$$Q_j^N |C_j|^N = Q_j^O |C_j|^O + \sum_e \sum_k^{\#\text{subareas}} Q_{e,k}^* \delta V_{e,k}, \tag{26}$$

where where index O (resp. N) stands for Old (resp. New) and $Q_{e,k}^*$ is the flux built in an upwind way. This is more accurate with respect to the geometric description than (23). We emphasize that this edge-based formulation is less accurate but more robust than exact meshes intersection which is cell-based by construction. We can establish a maximum principle (local bound preserving scheme) for first-order scheme

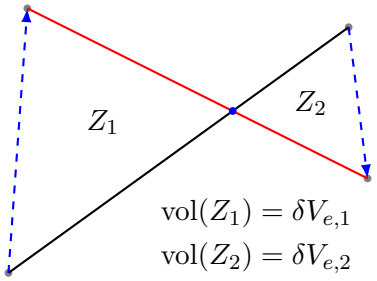


FIGURE 16. Self-intersection for straight polygonal: always at most one intersection point and two sub-areas to compute.

(see [ALE2DHAL] for polygons) if we have an other CFL-like condition (this constraint ensures that Q_j^N in (26) is a convex combination of neighboring old values Q_k^O):

$$|C_j|^O + \sum_e \sum_{k;\delta V_{e,k}<0}^{\#subareas} \delta V_{e,k} \geq 0. \tag{27}$$

As in the case of the scheme (23)(24), if (27) is not fulfilled for some cell, we apply sub-cycling. We then need

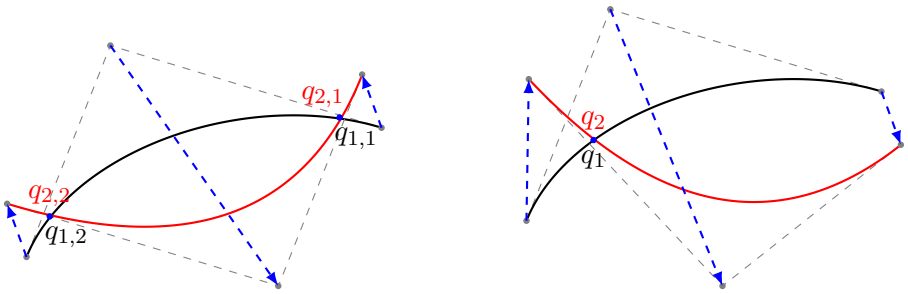


FIGURE 17. Self-intersection for arbitrary conic: at most two intersection points to compute, depending on nodal/control point displacement and weight $\omega(t)$ evolution.

to extend the formula (12) of [WAGUSEDE, WAGU] to the case where we only integrate over a sub-interval of (0,1), say (q_1, q_2) with $0 \leq q_1 < q_2 \leq 1$, see Figure 19:

$$I_{q_1,q_2} = \frac{1}{2}\int_{q_1}^{q_2} \left(x(q)\frac{d}{dq}y(q) - \frac{d}{dq}x(q)y(q)\right) dq. \tag{28}$$

We denote $A(M^\omega(q), M_0, M_1, M_2, q_1, q_2)$ the area between parameters q_1 and q_2, hence:

$$\begin{aligned} I_{q_1,q_2} &= A(M^\omega(q), M_0, M_1, M_2, q_1, q_2) + A^T(O, M(q_1), M(q_2)), \\ &= A(M^\omega(q), M_0, M_1, M_2, q_1, q_2) + \frac{1}{2}\det(\mathbf{OM}(q_1), \mathbf{OM}(q_2)). \end{aligned} \tag{29}$$

The natural extension of relation (12) is the following, by linearity and additivity, we suppose $q_1 = 0$ and q_2 is the arbitrary parameter q:

$$A(M^\omega(q), M_0, M_1, M_2, 0, q) = (g(\omega, q) - g(\omega, 0))\ A^T(M_0, M_1, M_2), \tag{30}$$

where the function $g(\omega, q)$ is described below. Its computation is equivalent to find a primitive (not necessary vanishing at $q = 0$ because of difference form) of the function $\frac{1}{2}(x(q)\frac{d}{dq}y(q) - \frac{d}{dq}x(q)y(q))$ in (28).

Proposition 6. *The primitive for arbitrary values of the parameter q in $[0, 1]$ is given by:*

$$\begin{cases} (1)\ \textit{if } \omega = 0, \ \textit{then } g(\omega, q) = 0, \\ (2)\ \textit{if } 0 < \omega < 1, \ \textit{then } g(\omega, q) = \frac{\omega}{1-\omega^2}\left(\frac{1}{\sqrt{1-\omega^2}}\text{atan}(\sqrt{\frac{1-\omega}{1+\omega}}(2q-1)) - \frac{2\omega q+1}{2+4(\omega-1)q(1-q)}\right), \\ (3)\ \textit{if } \omega = 1, \ \textit{then } g(\omega, q) = \frac{2}{3}q^3, \\ (4)\ \textit{if } \omega > 1, \ \textit{then } g(\omega, q) = \frac{\omega}{2(\omega^2-1)}\left(\frac{1}{\sqrt{\omega^2-1}}\log(\frac{\sqrt{\omega+1}-(2q-1)\sqrt{\omega-1}}{\sqrt{\omega+1}+(2q-1)\sqrt{\omega-1}}) + \frac{2\omega q+1}{1+2(\omega-1)q(1-q)}\right). \end{cases} \tag{31}$$

Remark 7. (1) *The reader can check that $g(\omega, 1) - g(\omega, 0)$ is equal to $f(\omega)$ in (13).*
(2) *For fixed q, the function $g(\omega, q) - g(\omega, 0)$ is continuous in ω.*

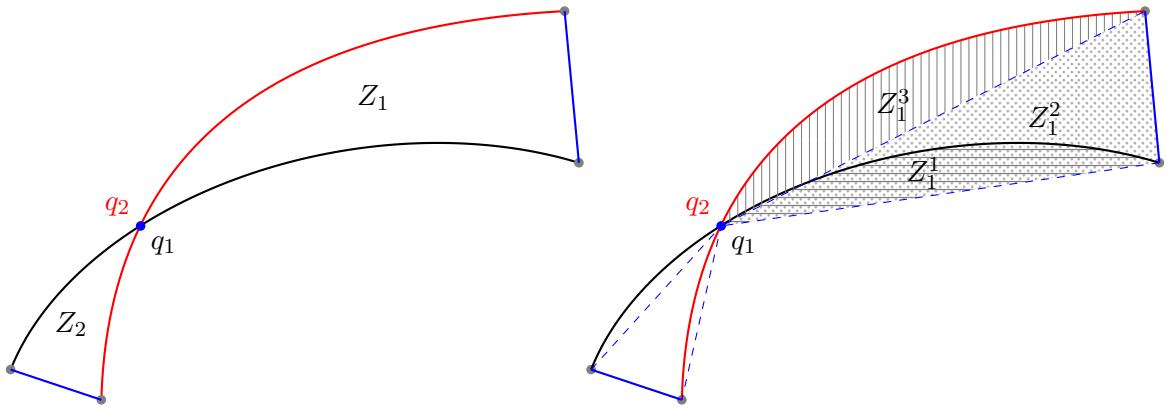


FIGURE 18. Sub-area decomposition for conic self-intersection in case of only one intersection point.

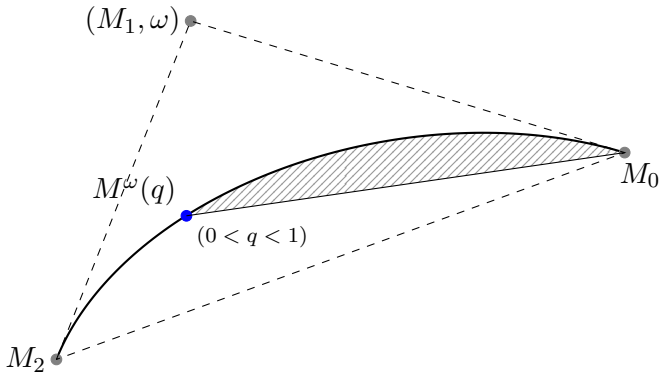


FIGURE 19. Self-intersection computation: Primitive of conic at arbitrary value parameter q gives the area of shaded region.

We emphasize that solving the problem of finding the intersections of two conic segments is a fourth order polynomial equation that can be solved by algebraic formula. Here, we do not use this approach and use an iterative method that must be done at each time step (or at each sub-cycling phase).

(1) Localization: compute the flux by creation of bounding boxes for each conic segment (see Figure 20), b_i is the straight segment joining the two endpoints of conic Cn_i and h_i the translated segment passing by the shoulder point S_i. For each conic Cn_i, we compute the intersection with the two straight lines b_j, h_j $(j \neq i)$. We give an exact solution of intersection straight/conic segment:

(a) We take the segment (M_a, M_b) equation: $(y_b - y_a)(x - x_a) - (x_b - x_a)(y - y_a) = 0$, into which we replace (x, y) by equation (7), we obtain a degree two equation in q:

$$\begin{aligned}&(y_b - y_a)\left((x_0 - x_a)(1-q)^2 + 2\omega q(1-q)(x_1 - x_a) + q^2(x_2 - x_a)\right) - \\ &(x_b - x_a)\left((y_0 - y_a)(1-q)^2 + 2\omega q(1-q)(y_1 - y_a) + q^2(y_2 - y_a)\right) = 0.\end{aligned} \tag{32}$$

(b) For each q_i real solution of (32), we check if $0 < q_i < 1$ to keep it.

(2) Iterative step: there are at most four intersection points. If there are no intersection point at all for both conics, then these two conics do not intersect and we can use the flux (23) in Figure 10.

If there exists an intersection point for problem (32), we use that initial guess to compute iteratively a q^n sequence of parameter (for each conic, we have at most four such sequences q_k^n, k=1,..,4). We use the implicit form $f(x, y) = 0$ of both conics Cn_i, Cn_j (see property (6) in section I), and then

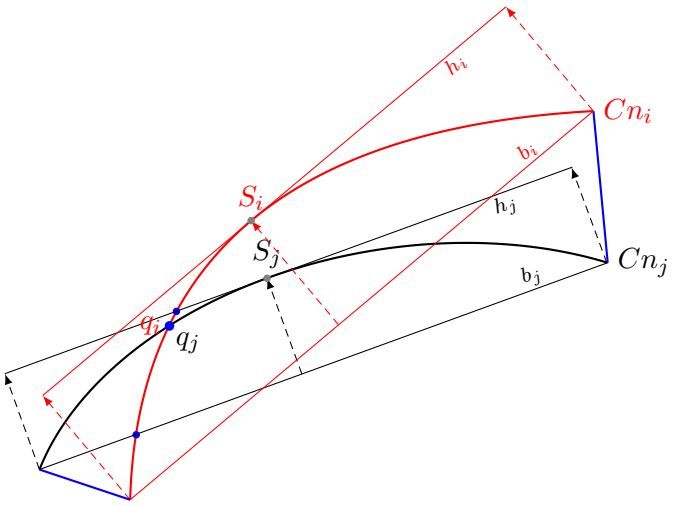


FIGURE 20. Construction of mutual bounding boxes.

iterate to solve:

$$f_j(x^i(q_k^{n+1}), y^i(q_k^{n+1})) = 0, \qquad j \neq i, \tag{33}$$

and stop if $|f_j(x^i(q_k^{n+1}), y^i(q_k^{n+1}))| < \epsilon$ ($\epsilon = 10^{-11}$).

(3) Filtering: We filter redundant q_i (for $j \neq k$, if $|q_j - q_k| < \epsilon$ we keep only one). Having unique q_i, we check if $0 < q_i < 1$ to keep that point. See example of Figure 21.

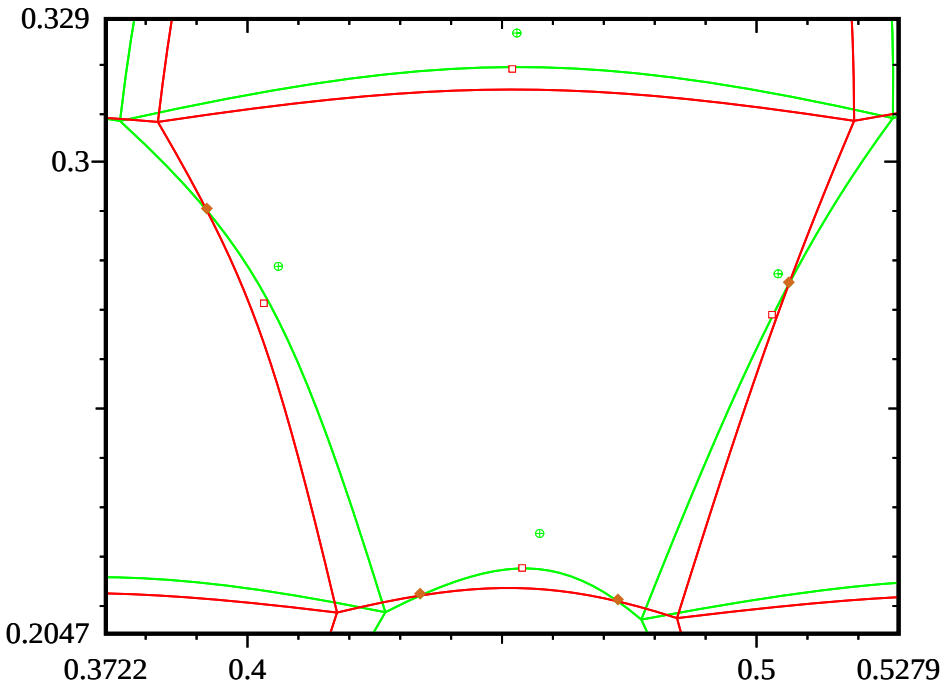


FIGURE 21. Zoom of intersection points (filled diamond ⋄) obtained with self-intersection flux calculation (26), circles ∘ (resp. square □) are control points on OLD (green) mesh (resp. NEW (red) mesh).

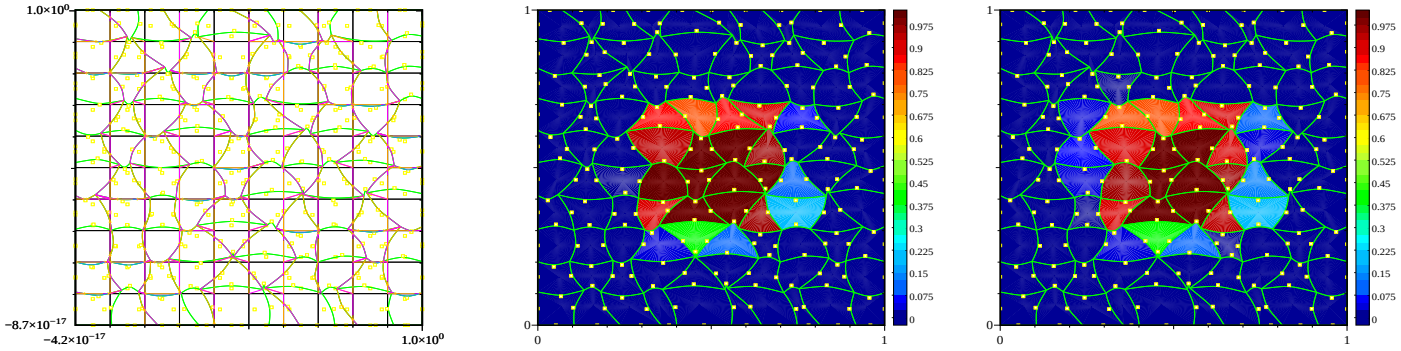


FIGURE 22. Left Superposition of OLD (cartesian) and NEW mesh (truly conic), initial data is given in Figure 11. Comparison of two schemes: center single zone flux scheme (23) and self-intersection scheme (26) on right, slightly difference here, maximum principle obtained in each case (here little white squares represent control points of conic edges).

3. REZONING ON CONIC CELLS

In this section, we want to reuse the rezoning technologies based on the logical mesh to control the quality of underlying polygonal cells. In [ALE2DHAL], we define short and medium ranges of a node and define a nodal quality that can be translated to this logical mesh. The control points of edges are moved in a second step. Now, the rezoning step can be described by the following three steps process:

(1) Move the nodes using logical polygonal connectivity (without control points). Here we use existing algorithms [JUN, RJM, ESCO] and the nodal mesh quality of [ALE2DHAL] using short/medium range of a node (without control points).

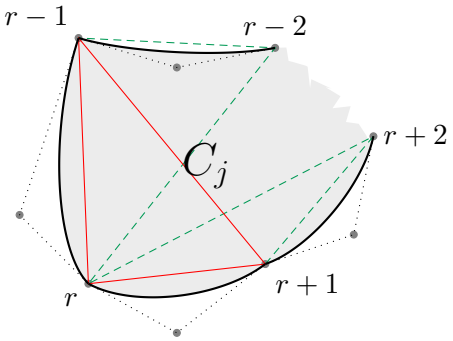


FIGURE 23. Definition of short-medium range same as logical underlying polygonal mesh: Short $r = \{r+1, r-1\}$, Med $r = \{r+2, r-2\}$.

(2) Move the control points in a predictor-corrector procedure:
   (a) *Predictor*: The first displacement are taken as the half sum of extremities displacements (see Figure 24 left). All these displacements are done independently (Jacobi update).
   (b) *Corrector*: If necessary, control point M^{pred} is moved to a position such that the corresponding edge does not intersect two adjacent control simplexes and is truly inside the conic cell. We emphasize that thanks to properties (2)(3) in section I (Convex hull and Endpoints interpolation),

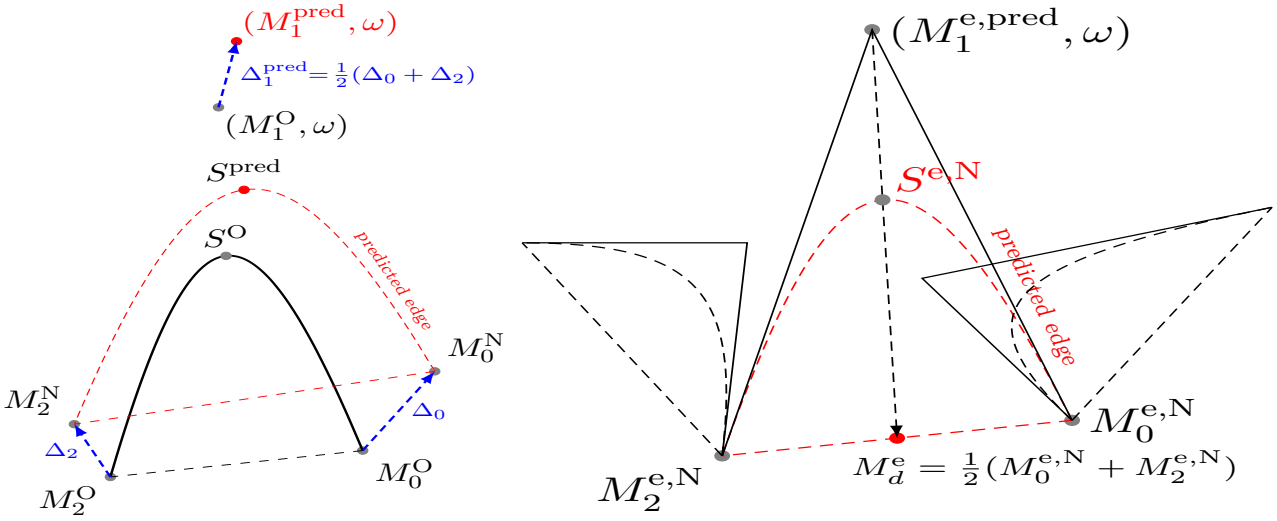


FIGURE 24. Rezoning of control points. Predictor displacement $\mathbf{\Delta}_1^{\text{pred}}=\frac{1}{2}(\mathbf{\Delta}_0+\mathbf{\Delta}_2)$, and Corrector: Predicted control point projected on relaxed position of middle logical edge and Shoulder point of predicted Edge.

it is sufficient to ensure that the control point does not belong to any neighboring simplex. In the case of such an intersection, the control point position is (iteratively) relaxed with predicted Shoulder point position (and middle of logical edge e $M_d^e=\frac{1}{2}(M_0+M_2)$). Such a treatment has the property to (iteratively) diminish the maximum curvature of the edge and thus tends toward a valid cell (with the hypothesis that the underlying logical polygonal cell is valid): see (24) right and (25). See a numerical example Figure 26 where predictor is always done.

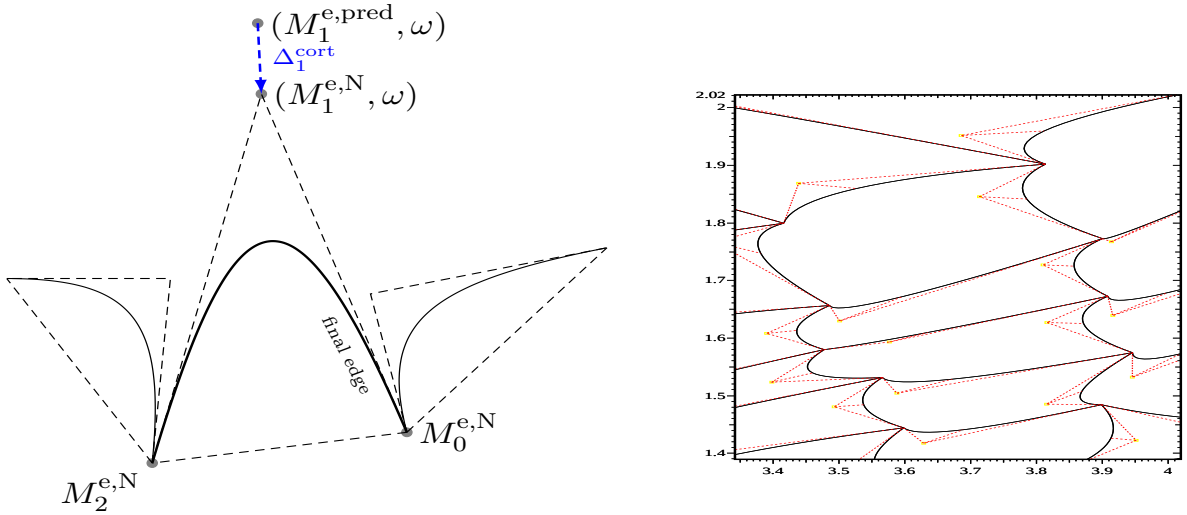


FIGURE 25. Rezoning: Moving node and control point with distance diminution in case of non valid cell (total displacement $\mathbf{\Delta}_1=\mathbf{\Delta}_1^{\text{pred}}+\mathbf{\Delta}_1^{\text{cort}}$ (Figure 24 and 25)). Right: dashed line draw the link between control point to both end points and Shoulder point: final conic cells are valid if underlying logical polygon are valid.

(3) For each control points re-compute the optimal weight ω. Hence we may obtain naturally a *continuous geometric cell boundary re-adaptation*, we postpone this question for future work and keep the same (initial) value here.

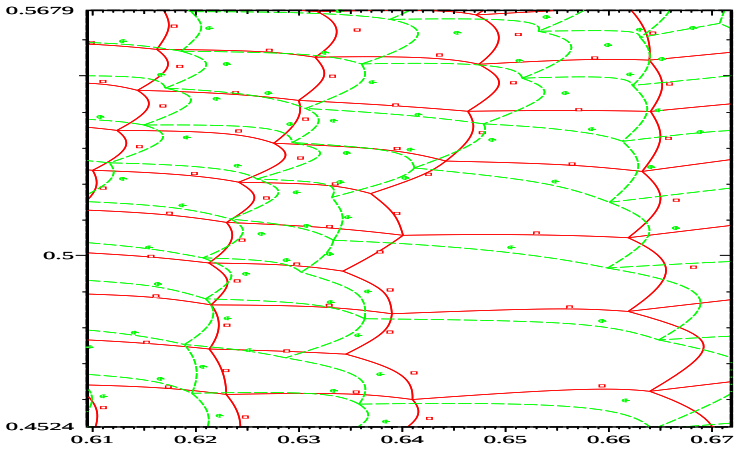


FIGURE 26. Smoothing of conic unstructured cells (arbitrary number of edges) using [ESCO] for node displacement (on polygonal mesh without control points) and a posteriori control point displacement using check of cell validity, Old cells: dashed line (green) with control point represented by (green) dashed circle and New cells: solid line (red) with control point represented by (red) solid square.

4. Extension of GLACE Hydrodynamic scheme on conic cells : ω-GLACE

In this section, we give an extension of polygonal GLACE hydrodynamic scheme [GLACE] to arbitrary conical meshes (2):

$$\begin{cases} D_t \int_{C(t)} 1\, dv - \int_{\partial C(t)} \mathbf{U}.\hat{\mathbf{N}}\, ds = 0, \\ D_t \int_{C(t)} \rho\, dv = 0, \\ D_t \int_{C(t)} \rho \mathbf{U}\, dv + \int_{\partial C(t)} P.\hat{\mathbf{N}}\, ds = 0, \\ D_t \int_{C(t)} \rho E\, dv + \int_{\partial C(t)} P\mathbf{U}.\hat{\mathbf{N}}\, ds = 0. \end{cases} \tag{34}$$

If we denote $M(t)$ the mass of cell $C(t)$ ($M(t) = \int_{C(t)} \rho\, dv$) which is constant , the semi-discrete ω-GLACE

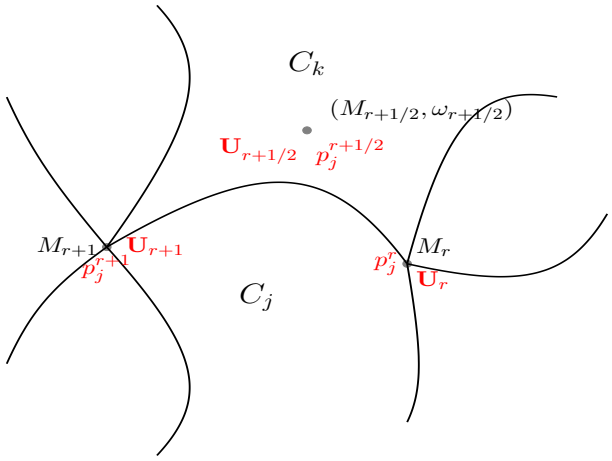


FIGURE 27. Degree of freedom for Lagrangian ω-GLACE scheme on conic cell.

scheme on conic cells writes:

$$\left\{\begin{array}{l} M_j\tau_j'(t) = \sum_r(\mathbf{C}_j^{r,\omega},\mathbf{U}_r) + \sum_{r+1/2}(\mathbf{C}_j^{r+1/2,\omega},\mathbf{U}_{r+1/2}), \qquad \tau_j(t) = \frac{1}{\rho_j(t)}, \\ M_j\mathbf{U}_j'(t) = -\sum_r \mathbf{C}_j^{r,\omega}p_j^r - \sum_{r+1/2}\mathbf{C}_j^{r+1/2,\omega}p_j^{r+1/2}, \\ M_jE_j'(t) = -\sum_r(\mathbf{C}_j^{r,\omega},\mathbf{U}_r)p_j^r - \sum_{r+1/2}(\mathbf{C}_j^{r+1/2,\omega},\mathbf{U}_{r+1/2})p_j^{r+1/2}. \end{array}\right. \tag{35}$$

where the $\mathbf{C}_j^{r,\omega}$ and $\mathbf{C}_j^{r+1/2,\omega}$ are computed in (17) and (18). In order to determine $\mathbf{U}_r$ and $\mathbf{U}_{r+1/2}$ and the corresponding pressure p_j^r and $p_j^{r+1/2}$, we apply the same techniques as in [GLACE] for nodes and control points. The Riemann invariant is discretized

(1) for nodes r, onto $\mathbf{C}_j^{r,\omega}$ direction giving:

$$p_j^r - p_j + \alpha_j^r(\mathbf{U}_r - \mathbf{U}_j, \mathbf{n}_j^{r,\omega}) = 0, \tag{36}$$

where $\mathbf{n}_j^{r,\omega} = \frac{\mathbf{C}_j^{r,\omega}}{|\mathbf{C}_j^{r,\omega}|}$ see (17).

(2) for control points $r+1/2$ of edges, onto $\mathbf{C}_j^{r+1/2,\omega}$ direction giving:

$$p_j^{r+1/2} - p_j + \alpha_j^{r+1/2}(\mathbf{U}_{r+1/2} - \mathbf{U}_j, \mathbf{n}_j^{r+1/2,\omega}) = 0, \tag{37}$$

where $\mathbf{n}_j^{r+1/2,\omega} = \frac{\mathbf{C}_j^{r+1/2,\omega}}{|\mathbf{C}_j^{r+1/2,\omega}|}$ see (18).

Here α_j^r and $\alpha_j^{r+1/2}$ in (36), (37) are the acoustic impedance $\rho_j c_j$ (c is the sound speed $c^2 = \frac{\partial P}{\partial \rho}|_S$, and S the entropy) of the cell C_j. To construct the solver we also enforce the following conservation condition:

- Around internal nodes:

$$\sum_j p_j^r \mathbf{C}_j^{r,\omega} = \mathbf{0} \tag{38}$$

- Around control point of internal edges:

$$\begin{array}{l} \sum_j p_j^{r+1/2}\mathbf{C}_j^{r+1/2,\omega} = \mathbf{0}, \qquad \text{which is equivalent to} \\ p_j^{r+1/2}\mathbf{C}_j^{r+1/2,\omega} + p_k^{r+1/2}\mathbf{C}_k^{r+1/2,\omega} = \mathbf{0}. \end{array} \tag{39}$$

Hence in this case, thanks to relation $\mathbf{C}_k^{r+1/2,\omega} + \mathbf{C}_j^{r+1/2,\omega} = 0$ see (19), we obtain the continuity of pressure at control point:

$$p_j^{r+1/2} = p_k^{r+1/2} \tag{40}$$

To solve (36), we consider the following two by two linear systems and obtain the velocities at each node:

$$\left\{\begin{array}{l} A_r\mathbf{U}_r = \mathbf{b}_r, \qquad \text{with} \\ A_r = \sum_j \rho_j c_j \frac{\mathbf{C}_j^{r,\omega}\otimes\mathbf{C}_j^{r,\omega}}{|\mathbf{C}_j^{r,\omega}|}, \\ b_r = \sum_j \mathbf{C}_j^{r,\omega}p_j + \rho_j c_j \frac{\mathbf{C}_j^{r,\omega}\otimes\mathbf{C}_j^{r,\omega}}{|\mathbf{C}_j^{r,\omega}|}\mathbf{U}_j. \end{array}\right. \tag{41}$$

This system is always invertible whatever the weight value for edges impinging at node r.
Once the velocity computed by (41) the pressure p_j^r in (35) is computed using (36).
For control point (37), we first apply the one dimensional Riemann solver that gives both normal velocity and continuous pressure (see (40)):

$$\left\{\begin{array}{l} \mathbf{U_{r+1/2}}.\mathbf{n_j^{r+1/2,\omega}} = \frac{\rho_j c_j \mathbf{U_j}.\mathbf{n_j^{r+1/2,\omega}} + \rho_k c_k \mathbf{U_k}.\mathbf{n_j^{r+1/2,\omega}}}{\rho_j c_j + \rho_k c_k} + \frac{1}{\rho_j c_j + \rho_k c_k}(p_j - p_k), \\ p^{r+1/2} = \frac{\rho_j c_j p_j + \rho_k c_k p_k}{\rho_j c_j + \rho_k c_k} + \frac{\rho_j c_j \rho_k c_k}{\rho_j c_j + \rho_k c_k}(\mathbf{U_j} - \mathbf{U_k}).\mathbf{n_j^{r+1/2,\omega}}. \end{array}\right. \tag{42}$$

To obtain a truly 2D velocity for the control point, we approximate the tangential part $\mathbf{U_{r+1/2}}.\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega}$ (with $\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega}.\mathbf{n_j^{r+1/2},\omega} = \mathbf{0}$), see e.g. [Godunov]:

$$\mathbf{U_{r+1/2}}.\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega} = \begin{cases} \mathbf{U_j}.\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega}, & if \quad \mathbf{U_{r+1/2}}.\mathbf{n_j^{r+1/2},\omega} > 0, \\ \mathbf{U_k}.\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega}, & if \quad \mathbf{U_{r+1/2}}.\mathbf{n_j^{r+1/2},\omega} < 0, \\ \frac{\rho_j c_j \mathbf{U_j} + \rho_k c_k \mathbf{U_k}}{\rho_j c_j + \rho_k c_k}.\tau_{\mathbf{j}}^{\mathbf{r+1/2},\omega}, & else. \end{cases} \tag{43}$$

In the last approximation (43) where no flux passes through the edge, we have just made a simple convex interpolation. In this case, we could use a more sophisticated approach to treat sliding condition.

5. Numerical Section

5.1. Pure Lagrangian test: Sedov

In this case, we solve the Sedov problem. The initial condition is given by $(\rho^0, P^0, \mathbf{U^0}) = (1, 10^{-6}, \mathbf{0})$ and $\gamma = 1.4$. The pressure in cell C containing the origin is such that, see [HdrPhMaire],

$$P_c = (\gamma - 1)\rho_C \frac{\varepsilon^0}{|C|} \tag{44}$$

where $\varepsilon^0 = 0.244816$ so that the solution consists of a diverging shock whose front is located at radius R=1 at time t=1. Moreover, the density peak reaches the value 6.
For conical ω-GLACE simulations, the initial control points are set to the middle of logical edges and all the weights are set to $\omega = 1$.

Remark 8.
- *In Figure 28, we notice that our new conical ω-GLACE scheme do not exhibit spurious mesh instability in the diagonal direction (for $\omega = 1$). This surely could be explained by an analysis of the associate corner matrix (see for example [HdrPhMaire]) at nodes and control points.*
- *Link to the previous remark, the result of ω-GLACE and polygonal EUCCLHYD (see [HdrPhMaire]) are qualitatively quite similar. We recall that despite original GLACE, the conical ω-GLACE scheme possess intermediate pressure located at edge and EUCCLHYD gets two.*
- *The ω-GLACE with $\omega = 0$ gives the same results as original polygonal GLACE scheme.*

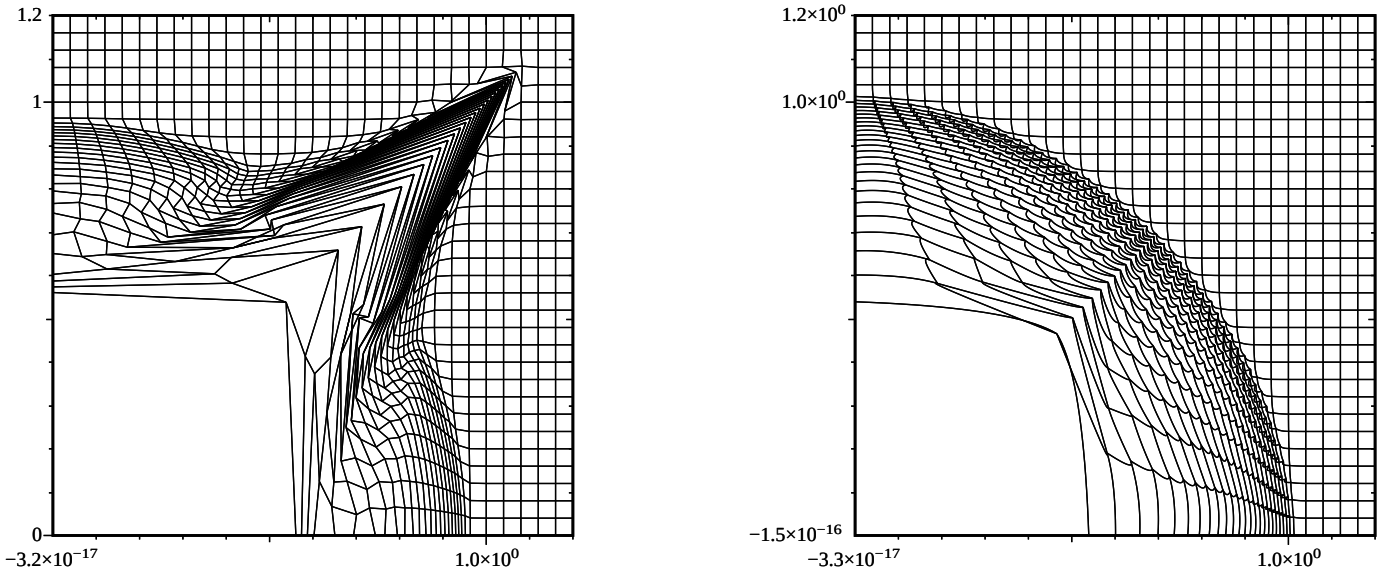


FIGURE 28. Initial logical mesh made by 30×30 cells. Meshes at final time for first-order original polygonal GLACE scheme (left) and new conical ω-GLACE scheme (right).

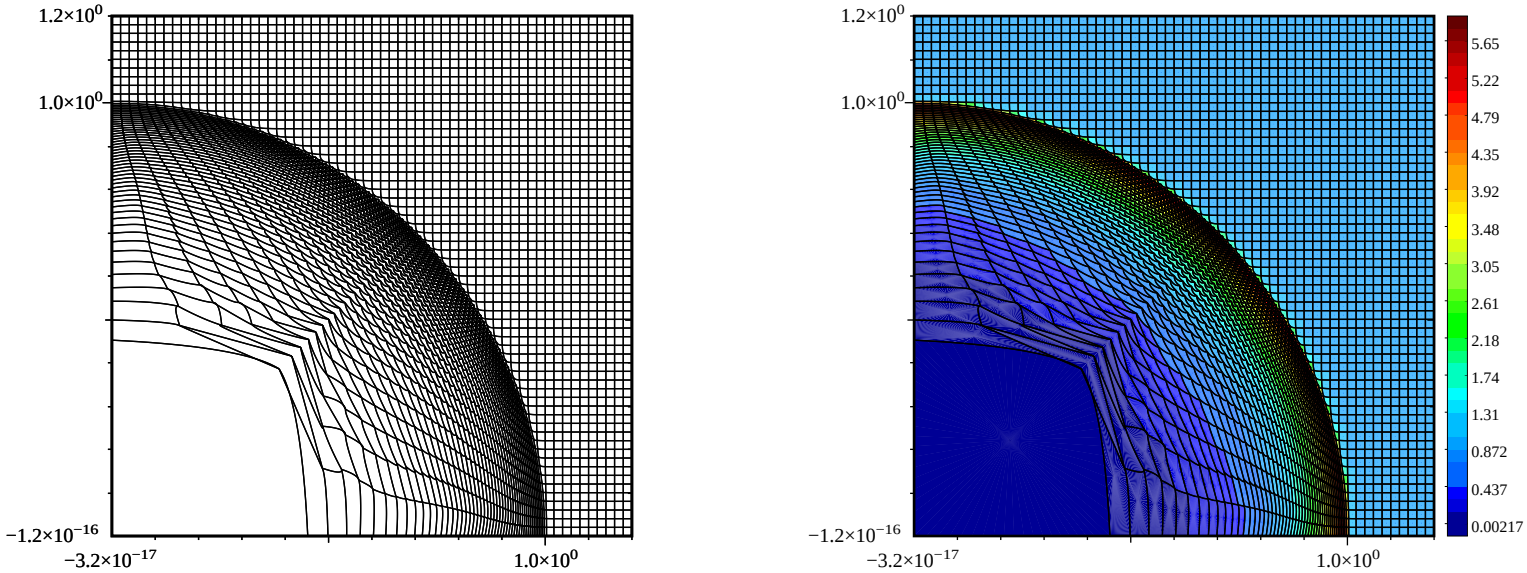


FIGURE 29. Initial Logical mesh made by 60×60 cells. At final time, we have depicted the mesh (left) and the numerical solution (right) of the first-order conical ω-GLACE scheme.

5.2. **Comparison Lagrangian/ALE for conical cells**

For the rezoning strategy, we use the mixed Jun/Escobar [JUN, ESCO] smoothing algorithm with nodal mesh quality control [ALE2DHAL]. The remapping is the single zone edge based flux (23).

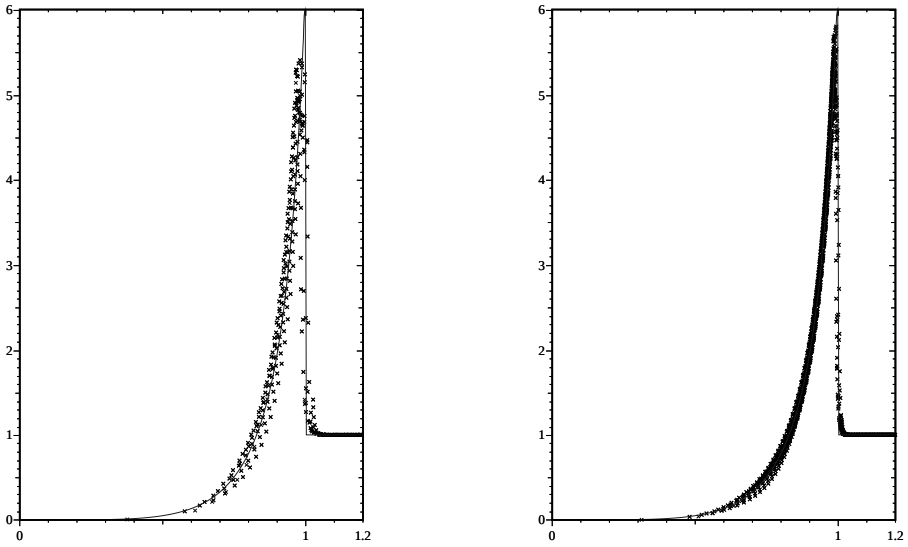


FIGURE 30. Density in all cells as functions of the cell-center at final time for first-order ω-GLACE scheme with 30×30 logical cells (left) and 60×60 cells (right), reference solution in solid line.

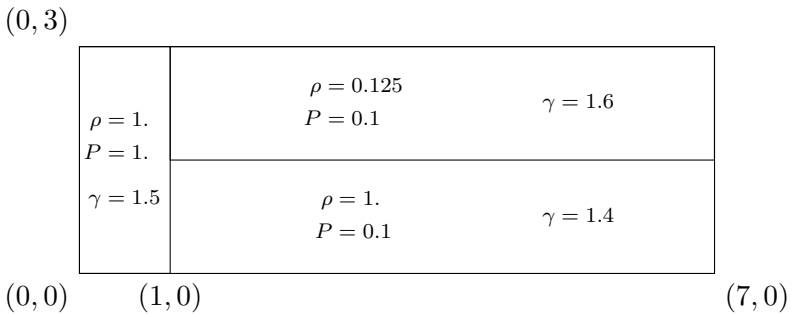


FIGURE 31. Initial regular cartesian grid with $n_x = 36$, $n_y = 16$. We take three perfect gases (with different γ) at rest in three different zones. Control points are initially set in the middle of logical edge.

Remark 9. *The pure Lagrangian scheme ω-GLACE does not give accurate compuation in this case because of too much tangled cells. However, we do not observe hourglass modes as for polygonal GLACE scheme. For ALE simuation, we do not observe qualitative difference between polygonal and conical approach.*

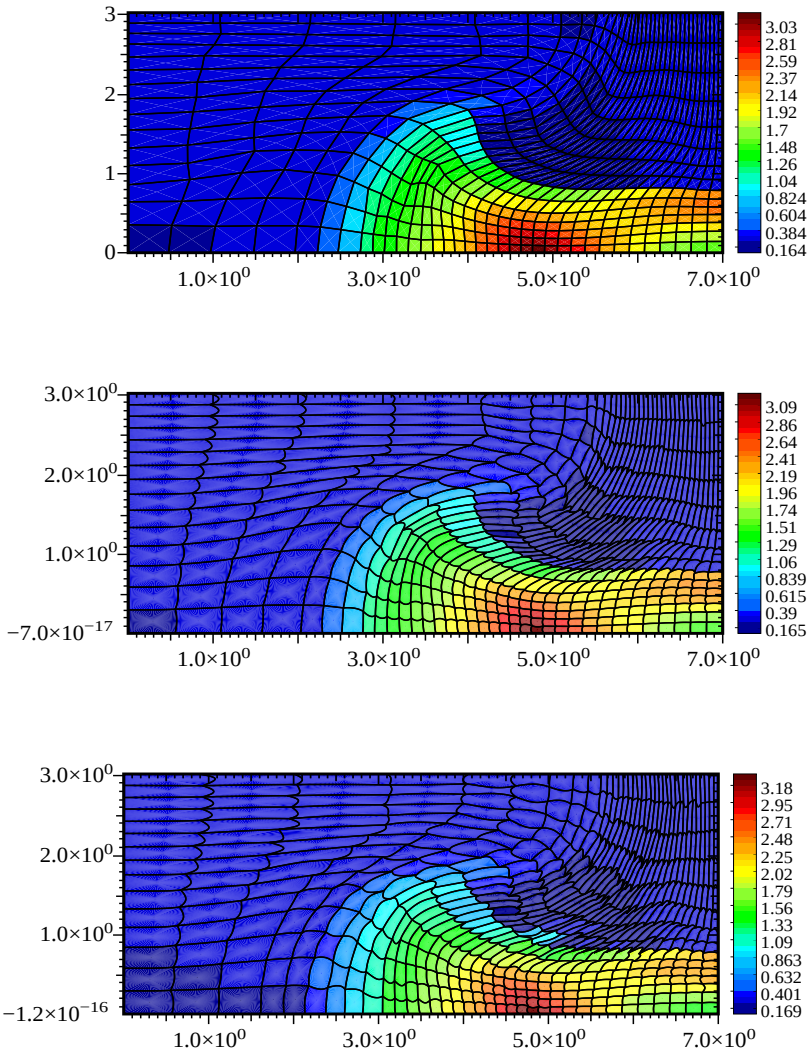


FIGURE 32. Final time t=5. Comparison between ALE with straight polygons (Up), and two versions of conical meshes ($\omega \equiv 1$) (Center) and ω randomly choosen in [0,2] for each edges (Down). Caution color levels are not exactly the same.

6. CONCLUSION

We have proposed an extension of continuous ALE formalism in finite volume discretization for arbitrary conic cells. Each edge of the cell is parameterized by a rational quadratic Bezier curve defined by a control point and a positive weight ω giving the conic type.
Thanks to [WAGUSEDE, WAGU], we obtain an analytic formula for the area of arbitrary cells (with any number of edges). We propose two extensions of first-order remapping finite volume scheme [MARSHA] and [ALE2DHAL]. A maximum principle local bound preserving is deduced if a CFL-like condition is imposed. The rezoning step is done by a three steps process. The first one is the nodal movement using connectivity of logical polygonal mesh without their control points, here we use existing algorithms [JUN,ESCO] and also nodal quality of [ALE2DHAL]. It is followed by the movement of the control points through a predictor/corrector procedure leading to valid cells. The third step corresponds to a possible re-adjustment of the conic parameter ω, it has not been addressed in this paper.
Finally, we also give a representation of GLACE scheme [GLACE] for conic cells in terms of $\mathbf{C}_j^r$ formalism.

This ω-GLACE is contained in the convex hull of two limit polygonal GLACE schemes $\omega = 0$ and $\omega \to \infty$. Morevover, the new GLACE scheme possess a degree of freedom located at each edge, in particular better stability is observed in practice.
To summarize, we give here an overall feasibility presentation, many questions are still to be clarified, nevertheless preliminary results seem to be very promising (we use [GO++] for developments and simulations).

7. Future works

We will adress a second order version of these conical Lagrangian and ALE schemes. We would like also to change the ω in the Lagrangian step, to take into account deformation due to the flows. Finally, we are interested in deriving an ω-EUCCLHYD version, see [EUCCLHYD].

8. Acknowledgments

We would like to thank G. Samba for fruitful discussions and S. Del Pino for valuable remarks. The authors are grateful to organizers of cemracs 2010.
We use the free software 'sage' (see `http://sagenb.org/`) to compute analytic formulas in this paper.

References

[DUVAUT] Duvaut G., *Mécanique des milieux continus*, Coll. Math. appl. pour la maîtrise, MASSON 1990.
[WAGUSEDE] Wang Guojin, Sederberg T. W., *Computing areas bounded by rational Bézier curves*, CADDM, Vol 4, No. 2, pp. 18-27, September 1994.
[WAGU] Wang Guojin, *Computing integral values involving nurbs curves*, Jour. of Software, Vol 7, No. 9, pp. 542-546, September 1996.
[LIGACH] Ming Li, Xiao-Shan Gao, Shang-Ching Chou, *Quadratic approximation to plane parametric curves and its application in approximate implicitization*, Visual Comput., Vol 22, pp. 906-917, 2006.
[GLACE2D] B. Després, C. Mazeran., *Lagrangian gas dynamics in two dimensions and lagrangian systems. Arch. Rational Mech. Anal.*, Vol 178, pp 327-372, 2005.
[GLACE] G. Carre, S. Del Pino, B. Despres, E. Labourasse, *A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension* , Jour. Comp. Physic., Vol 228,pp 5160-5183, 2009.
[EUCCLHYD] Maire, P.H. and Abgrall, R. and Breil, J. and Ovadia, J., *A cell-centered Lagrangian scheme for two-dimensional compressible flow problems*, SIAM J.Sci.Comput., Vol 29,pp 1781-1824, 2007.
[EQU] A. Chorin and J. Marsden., *A Mathematical Introduction to Fluid Mechanics.*, Springer Verlag, 1992.
[ALE2DHAL] Hoch P., *An Arbitrary Lagrangian-Eulerian strategy to solve compressible fluid flows*, HAL 2009 : http://hal.archives-ouvertes.fr/hal-00366858/fr/.
[MARSHA] L.G. Margolin, M. Shashkov, *Second Order sign preserving remapping on general grids*, J.Comp.Phys, Vol 184, pp. 266-298, 2003.
[FEV] S. Del Pino, *A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates*, CRAS, Vol 348, num 17-18, pp. 1027-1032, 2010.
[RIB] V. Dobrev, T. Ellis, T. Kolev, R. Rieben, *Energy conserving finite element discretizations of Lagrangian hydrodynamics. Part 1: Theoretical framework*, Downloadable presentation of Multimat'09 conference, or http://www.math.univ-toulouse.fr/HYDRO/conferences.html.
[ALED] Hirt, C. W. and Amsden, A. A. and Cook, J. L., *An arbitrary Lagrangian-Eulerian computing method for all flow speeds*,J. Comput. Phys., vol 135, pp 203-216, 1997.
[SHUSHEN] Cheng J., Shu C.W., *A third order Conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations*, comm. on comput. phys., vol 4, no5, pp 1008-1024, 2008.
[NEWEX] Claisse,A. and Després,B. and Labourasse,E. and Ledoux,F. *A new exceptional points method with application to cell-centered Lagrangian schemes and curved meshes*, preprint LJLL 2011, http://www.ljll.math.upmc.fr/publications/2011/R11017.php.
[HdrPhMaire] Maire P.H. , *Contribution to the numerical modeling of inertial confinement fusion*, HDR 2001.
[GO++] Benamou J.D., Hoch P., *GO++ : A modular lagrangian/eulerian software for hamilton jacobi equations of geometric optics type*, M2AN, 36(5):883-905, 2002.
[MARSHA2] Margolin L.G., Shashkov M., *Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics*, J. Comput. Phys., vol 149, number 2, pp 389-417, 1999.
[JUN] Jun B.I., *A modified equipotential method for grid relaxation*, technical report LLNL 2000.
[RJM] Knupp, P. and Margolin, L.G. and Shashkov, M., *Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian-Eulerian methods*, J. Comput. Phys., vol 176, pp 93-112, 2002.

[ESCO] Escobar, J.M. and Rodriguez, E. and Montenegro, R. and Montero, G. and Gonzalez-Yuste, J.M., *Simultaneous untangling and smoothing of tetrahedral meshes*, Comp. Meth. in Applied Mechanics and Engineering, vol 192(25), pp 2775-2787, 2003.

[Godunov] S. K. Godunov, A. V. Zabrodin, and M. Ya. Ivanov, et al., Résolution numérique des problèmes multidimensionnels de la dynamiques des gaz (Editions Mir, Moscou, 1979).