

## NON CONFORMAL ADAPTATION AND MESH SMOOTHING FOR COMPRESSIBLE LAGRANGIAN FLUID DYNAMICS\*

HOCH P.<sup>1</sup>, MARCHAL S.<sup>2</sup>, VASILENKO Y.<sup>3</sup> AND FEIZ A. A.<sup>4</sup>

**Abstract.** The context of the present work is the numerical approximation of two-dimensional compressible Lagrangian fluid dynamics. The equation on  $\rho$ ,  $U$  and  $E$  respectively the density, the velocity and the total energy describes the conservation of mass, momentum, and total energy of the fluid. For a numerical approximation, the starting point is a first order non direct ALE (Arbitrary Lagrangian Eulerian method) that consists of two stages. The first step is a pure Lagrangian step with a centered scheme [1] or [19] (for staggered schemes see [15], [7], [4], but we did not use them in this work). The second step is the union of a rezoning step (mesh is smoothed to cure pathological cells and/or adapted to concentrate nodes near singularities, but the connectivity is fixed here) [18], [2], [10], [16], and a remapping step, defining the old values on this new mesh in a conservative way (see [12], [21], [16]). In some cases, we need to refine/derefine locally the moving mesh, so that this adaptation step takes place between existing Lagrangian/ALE formulations, making the entire process a three steps scheme. We propose an adaptation strategy based on a variant of local non conformal cells. More precisely, as we will see the notion of “non conformal” is only partial, meaning that the new nodes created by adaptation can be a degree of freedom exactly as the master nodes, we call them semi-slaves. With a different approach for local non conformal quadrangular based refinement (see [8]), in our case unlike their Lagrangian step, we consider *all nodes as degree of freedom* meaning that we do not consider hanging nodes.

The refinement step is generic with respect to the polygonal mesh, the number of nodes in a cell is arbitrary and these are not restricted to be a constant (only triangles and/or quadrilaterals), the conservation is straightforward. The derefinement step is done by local non conformity and the conservation can be obtain either by self-intersection volumes fluxing (line clipping) or by polygonal clipping (more precise than coarse volume averaging).

Our approach naturally leads to define this notion of non conformity as a special object inside the description of a generic unstructured polygonal mesh. The development and the simulations have been done with the GO++ package.

**keywords:** Polygonal Unstructured ALE, Hybrid Mesh Representation, Local non Conforming Cell, Dynamic Adaptation, Centered Lagrangian Hydrodynamic, C++.

---

\* The authors thank the organizers of the CEMRACS 2007 P.Frey and C.Dobrzynski for their availability

<sup>1</sup> CEA-DIF, Bruyères-Le-Châtel 91297, Arpajon Cedex, France, email : philippe.hoch@cea.fr

<sup>2</sup> Université Henri Poincaré Nancy 1., Institut Elie Cartan Nancy (Mathématiques), B.P. 239, F-54506 Vandoeuvre-lès-Nancy Cedex, France, email : Sandrine.Marchal@iecn.u-nancy.fr

<sup>3</sup> University of West Bohemia, Faculty of Applied Sciences, Department of Mathematics, Univerzitni 22, Plzen 306 14, Czech Republic, email : yana@kma.zcu.cz

<sup>4</sup> Université Paris-Est Marne-la-Vallée, LETEM, Champs sur Marne, 77454 Marne-la-Vallée, Cedex 2, France, email: feiz@univ-mlv.fr

## 1. INTRODUCTION AND DESCRIPTION OF THE PROJECT

The project is about a numerical approximation of two-dimensional compressible fluid dynamics in (pseudo) Lagrangian frame, the mesh is moving at the flow speed and the equations are:

$$\begin{cases} D_t \rho + \rho \nabla \cdot U = 0, \\ \rho D_t U + \nabla P = 0, \\ \rho D_t E + \nabla \cdot (PU) = 0, \end{cases} \quad (1)$$

Where  $\rho$ ,  $U$ ,  $E$  and  $P$  are respectively the density, the velocity, the total energy and the pressure of the fluid,  $D_t$  is the Lagrangian derivative  $\forall q \in \mathbb{R}^2$ ,  $D_t q = \partial_t q + U \cdot \nabla q$ . To approximate (1), we use a non direct ALE (Arbitrary Lagrangian Eulerian) method consisting of a pure Lagrangian step with a centered scheme [1], [19] followed by a rezoning phase (mesh is smoothed to cure pathological cells and/or adapted to concentrate nodes near singularities) (see [18], [2], [10], [16]), and finally a remapping step, defining the old values on this new mesh in a conservative way (see [12], [16]).

Unfortunately, in some situations, this strategy is not sufficient, and we need sometimes to delete/and/or/add some cells locally (see Fig. 1).

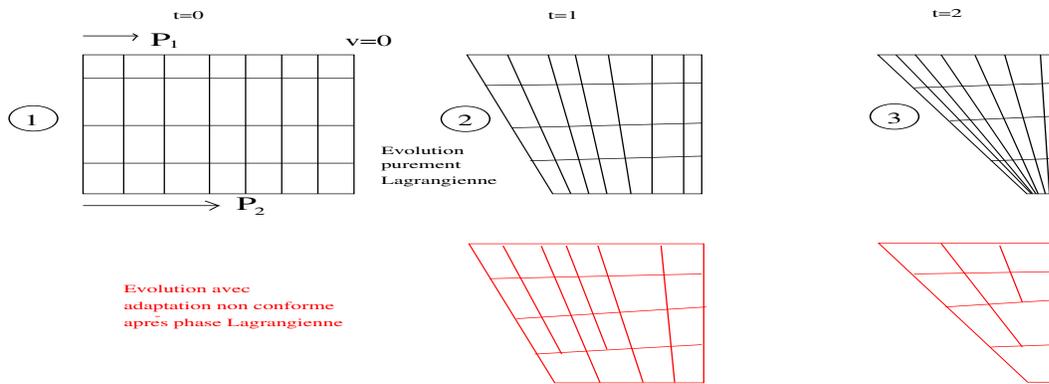


FIGURE 1. Up : Purely Lagrangian evolution, if the two bottom corners nearly collapse, whatever the ALE strategy it will be too much cells in between see time  $t = 2$ , Down (red) : Lagrangian evolution coupled with adaptation

The aim of the project is to obtain such a general scheme,

- (A) **Lagrangian** step (with arbitrary number of nodes per cell) and *all nodes are degree of freedom* (speed is given by the nodal solver).
- (B) **Adaptation** using refinement/unrefinement study,
- (C) **Rezoning** (caution when non conformity) and **Remapping** (no change because polygonal mesh notion is used).

which includes the previous non-direct ALE (A) and (C) when no such non conformity.

With the new versions of full centered Lagrangian schemes [1], [19] for step (A), the combination of the steps (B) and (C) is very promising. The reasons are the following:

- (1) There is no dual mesh to define for momentum remapping (C) as in staggered version [15].
- (2) Momentum conservation for adaptation step (B) is not so easy (see [8]).

Moreover, the cell refinement with sub cell creation is a straightforward operation in terms of conservative sub quantities computation. The cell derefinement is more complicated for the general case, but is feasible and more convenient in this non-conformal framework (even to obtain conservation by volumes fluxing or by polygonal clipping, see ANNEX last section).

The paper is organized as follows. In section two, we briefly recall the standard non-direct ALE formalism consisting of pure Lagrangian step followed by two steps rezoning and remapping. In the third section we define our central object for the adaptation by local non conformity, then we propose a strategy, which incorporates this adaptation step just after the Lagrangian step. Some details of refinement/derefinement process are described as well as some numerical examples for refinement step are shown. And finally, we make conclusion and some outlook to this initial work.

## 2. STANDARD ALE FORMULATION

In this chapter, we recall the standard approach of non-direct ALE formulation, a Lagrangian step followed by a rezoning plus a remapping process.

### 2.1. Lagrangian schemes

We recall the centered scheme of Despres/Mazeran [1] and Maire&al. [19]. This scheme can be described as:

- (1) All variables are centered (finite volumes type),  $(\rho_j, U_j, E_j)$  are centered in cell  $j$ <sup>1</sup>. Semi-discrete equations are:

$$\begin{aligned} m_j \frac{d}{dt} \tau_j - \sum_{r=1}^{nvertex(j)} l_{r,r+1} U_{r,r+1} \cdot n_{r,r+1} &= 0, \\ m_j \frac{d}{dt} U_j + \sum_{r=1}^{nvertex(j)} l_{r,r+1} P_{r,r+1} \cdot n_{r,r+1} &= 0, \\ m_j \frac{d}{dt} E_j + \sum_{r=1}^{nvertex(j)} l_{r,r+1} (PU)_{r,r+1} \cdot n_{r,r+1} &= 0, \end{aligned} \tag{2}$$

$m_j$  stands for the mass of cell  $j$ ,  $\tau_j = \frac{1}{\rho_j}$ ,  $nvertex(j)$  is the number of nodes in cell  $j$ , and  $n_{r,r+1}$  is the unit outer normal of  $(r, r + 1)$  in  $j$ , and  $l_{r,r+1}$  is the length of the edge joining the node  $r$  and  $r + 1$ , see Fig. 2.

- (2) Momentum and Total energy conservation (planar geometry),
- (3) Approximate linear Riemann solver,
- (4) Hourglass control if necessary,
- (5) Nodal fluxes (compatibility of mesh movement and continuity equation). The evolution of nodes is given by:

$$\frac{d}{dt} M_r(t) = U_r \tag{3}$$

where  $M_r$  (resp.  $U_r$ ) is the position (resp. velocity) of node  $r$ , the velocity is an unknown at this level. The mass conservation for cell  $j$  gives the following relation between the speed of edge  $(r, r + 1)$  and the speed of his end points:

$$U_{r,r+1} = \frac{1}{2}(U_r + U_{r+1}). \tag{4}$$

The nodal solver between the two scheme change now for the pressure computation of Fig. 2, each one use a linearisation of the Riemann invariant  $dP = \rho c dU$ , where  $c$  is the sound speed (acoustic Riemann solver), to obtain a well posed discrete system for all the unknowns  $U_r$  and the degree of freedoms for the pressure term.

- (1) Despres-Mazeran : one pressure for each node  $r$  in each cell (see Fig. 2 on left).

$$P_{r,r+1}^j = \frac{1}{2}(P_r^j + P_{r+1}^j) \tag{5}$$

---

<sup>1</sup>for clarity we put sometimes  $j$  as superscript, specially for  $P^j$

The face flux for total energy is then  $(PU)_{r,r+1}^j = \frac{1}{2}(P_r^j U_r^j + P_{r+1}^j U_{r+1}^j)$ . The unknowns are then  $P_r^j$  and  $U_r$ . To close the system, they use an approximation of Riemann invariant, and finally the problem is to solve :

$$P^j - P_r^j + \rho_j c_j (U_j - U_r, n_{jr}) = 0, \quad \forall r \text{ inside cell } j. \quad (6)$$

The nodal normal  $n_{jr}$  is defined by  $l_{r-1,r} n_{r-1,r} + l_{r,r+1} n_{r,r+1} := l_{jr} n_{jr}$ ,  $l_{r-1,r}$  and  $l_{r,r+1}$  being the length of edge  $(r-1,r)$  and  $(r,r+1)$ .

This defines an invertible linear system of  $U_r$  unknown. The solution  $U_r$  is then put in (6) to obtain  $P_r^j$ .

(2) Maire&al: one pressure for each half edge in each cell (see Fig. 2 on right).

$$P_{r,r+1}^j = \frac{1}{2}(P_{r,r+\frac{1}{2}}^j + P_{r+\frac{1}{2},r+1}^j) \quad (7)$$

The face flux for total energy is then  $(PU)_{r,r+1}^j = \frac{1}{2}(P_{r,r+\frac{1}{2}}^j U_r^j + P_{r+\frac{1}{2},r+1}^j U_{r+1}^j)$ . The unknowns are then  $P_{r,r+\frac{1}{2}}^j$ ,  $P_{r+\frac{1}{2},r+1}^j$  and  $U_r$ ,

$$\begin{cases} P^j - P_{r-\frac{1}{2},r}^j + \rho_j c_j (U_j - U_r, n_{r-1,r}) = 0, & \forall r \text{ inside cell } j. \\ P^j - P_{r,r+\frac{1}{2}}^j + \rho_j c_j (U_j - U_r, n_{r,r+1}) = 0, & \forall r \text{ inside cell } j \end{cases} \quad (8)$$

This defines an invertible linear system of  $U_r$  unknown. The solution  $U_r$  is then put in (8) to obtain  $P_{r-\frac{1}{2},r}^j$  and  $P_{r,r+\frac{1}{2}}^j$ .

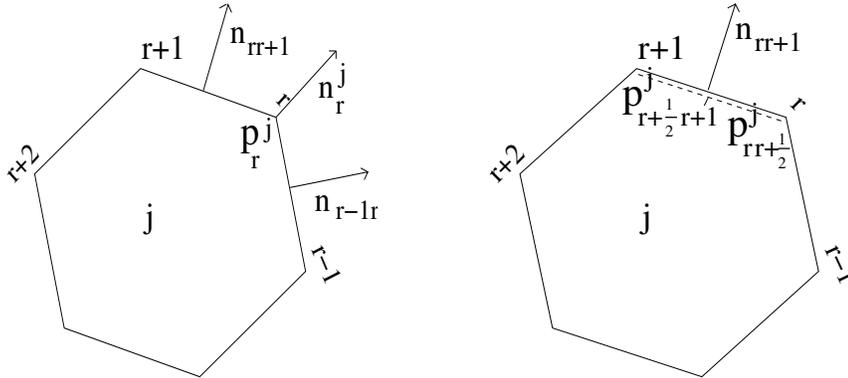


FIGURE 2. Localization of Pressure unknowns for nodal solver : Despres/Mazeran(Left) one pressure per node in each cell and Maire&al (right) one pressure per half edge in each cell.

The full first order scheme is obtained by the discretization of (2) with the explicit Euler scheme for time derivative. Generally speaking, the general behavior of pure Lagrangian scheme is that sometimes :

- (1) **Interacting shocks** can destroy mesh quality and increase the error,
- (2) **High curvature of interfaces** and boundaries (with fast variations), specially for rarefaction waves from a solid into a light material (vacuum) cause the mesh to be very distorted.

These facts lead naturally to the use of mesh smoothing and remapping (non-direct ALE).

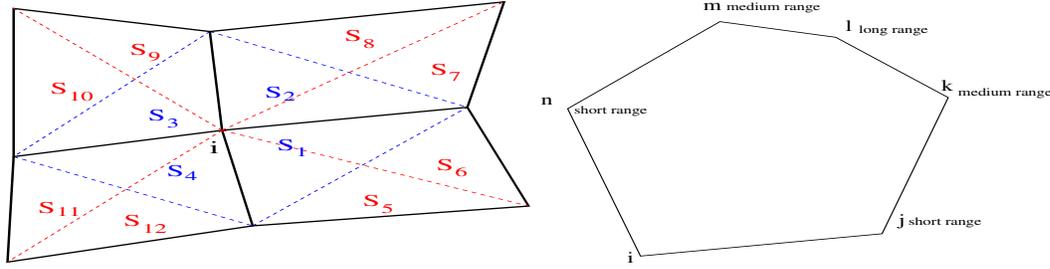


FIGURE 3. Short (blue) and medium (red) range of a node, (node/cell) and (node/node) neighborhood

## 2.2. Rezoning and remapping

We emphasize that in this section, the connectivity of the mesh is **fixed**. For this section and for notation, we refer the reader to [16].

### 2.2.1. Rezoning

The unknowns are only the *displacement* of nodes, there exist more or less two different approach to compute a smoothed mesh:

- (1) Elliptic PDE based: Barycentric, Tipton [22], Jun [2].
- (2) Optimization based: Reference Jacobian Matrix [17], condition number [20], Escobar [10], etc.

We use here an evolution of Escobar’s algorithm [10]. This is an optimization based algorithm that simultaneously untangle and smooth a (tetrahedral) mesh. The objective function at each node can be written as  $F_i(x) = \sum_{m=1}^{NbSimplexe} \tilde{f}_m$ . Where  $\tilde{f}_m = \frac{|S_m||\Sigma_m|}{\sigma_m + \sqrt{\sigma_m^2 + \epsilon}}$ , which is a modification of the inverse of the algebraic quality of  $m^{th}$  simplex (which is  $f_m = \frac{|S_m||\Sigma_m|}{2\sigma_m}$ ) attached to node  $i$  ( $\sigma_m$  is proportional to the area of the simplex). This modification is such that this functional is defined everywhere (this ia not true for the original one  $f_m$ ). For arbitrary connectivity mesh case, we do not know what is an ideal triangle, and even more the different simplexes do not take the same amount of information (Short Range / Medium Range) see Fig. 3. For this reason, the application that maps the reference simplex to ideal (equilateral) simplex (for all simplex linked to  $i$ ) is only an approximation. The Jacobian is given by

$$W = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix} \quad S = AW^{-1} \quad A = \begin{pmatrix} x_1^1 - x_0^1 & x_2^1 - x_0^1 \\ x_1^2 - x_0^2 & x_2^2 - x_0^2 \end{pmatrix}$$

$A$  is the Jacobian matrix of the map referenced to node  $x_0$  ( $i$  for all simplexes) between the real simplex and the reference simplex. In fact, for each simplex, we will introduce a parameter that we will adapt to each class Short / Medium. It will introduce a different matrix ( $W_\alpha$ ), and we will adapt this to the non homogeneous local connectivity around each node of Fig. 3. From a generical point of view, we propose to introduce a parameter that take into account non homogeneous connectivities around a node, the adapted Jacobian matrix writes:

$$W_\alpha = \begin{pmatrix} 1 & \cos(\alpha) \\ 0 & \sin(\alpha) \end{pmatrix} \quad S^\alpha = AW_\alpha^{-1}$$

Extension of Escobar functional to non homogeneous ideal simplexes <sup>2</sup>:

<sup>2</sup>moreover ideal simplex is not equilateral for our purpose

**Proposition 2.1.** For internal simplexes (Short Range), the optimal parameter is given by

$$\alpha = \frac{2\pi}{\mathbf{degree}(\mathbf{i})} \tag{9}$$

For external (diagonal) simplexes (Medium Range) belonging to the  $m$ 'th cell,

$$\alpha = \frac{2\pi}{\mathbf{degree}(\mathbf{i})(\mathbf{nvertex}(\mathbf{m}) - 2)} \tag{10}$$

Finally, to control the entire smoothing process (convex cell, mesh relaxation), we use the approach of nodal mesh quality, see [16].

2.2.2. Remapping

We need to compute the values of  $(\rho, U, E)^A$  from the old mesh  $M^A$  to the new smoothed mesh  $M^N$ . First we recall the following relation (see [13]) between each cell  $K_i$  of the old and new mesh, let  $Q$  an extensive conservative quantity (let say  $\rho$ ) :

$$\int_{K_i^N} QdV = \int_{K_i^A} QdV + \sum_{j \in V(K_i^A)} F_{ij}^Q \tag{11}$$

The quantity  $F_{ij}^Q \stackrel{def}{=} \left[ \int_{K_i^N \cap K_j^A} QdV - \int_{K_i^A \cap K_j^N} QdV \right]$  is the exact multi-dimensional flux (but **we have to compute intersection between 2 meshes** and must be done by using a polygonal clipping algorithm, see ANNEX). A first order scheme is then:

$$|K_i^N| \bar{Q}_i^N = |K_i^A| \bar{Q}_i^A + \sum_{j \in V(K_i^A)} (|K_i^N \cap K_j^A| \bar{Q}_j^A - |K_i^A \cap K_j^N| \bar{Q}_i^A) \tag{12}$$

Margolin-Shashkov [13] proposes a first order approximation of (11) without any clipping:

$$F_{ik}^{sda,1} = \max(0, V(\delta F_{ik})) \bar{Q}_{mv(i,k)}^A + \min(0, V(\delta F_{ik})) \bar{Q}_i^A \tag{13}$$

$\delta F_{ik}$  being the region swept by the displacement of edge (i,k) and  $V(\delta F_{ik})$  the algebraic value. It is easy to check that this scheme is positive if  $|K_i^A| > |\sum_{V(\delta F_{ik}) < 0} V(\delta F_{ik})|$ , which implies a local maximum principle (wrt Edge neighborhood).

We use a modification of this swept regions flux (see [16]), which is based on line clipping:

**Definition 2.2. Self-intersection flux**, is a flux that eventually computes a “self-tangled” patch created by edge displacement :  $e^A \cap e^N = \{S^*\}$  and  $S^* \neq \{\emptyset\}$

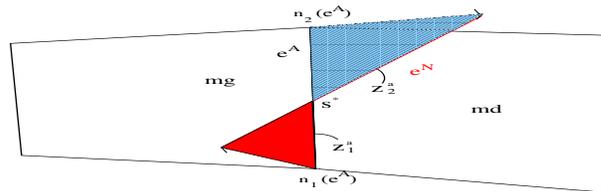


FIGURE 4. Self-intersection flux : two sub-volumes of fluxing for an edge for both adjacent cell mg and md.

The flux is the same as (13) excepting the case, when there is an intersection point  $S^*$ . Now the approximate flux  $F_{ik}$  has really two contributions, which is not the case for (13). This means that a cell may be simultaneously donor and the reverse receive. The scheme also has a local maximum principle if it is positive ([16]).

### 3. ADAPTATION STEP

In this chapter, we first introduce the minimal description of our central object: the welding line for general cells. Afterward, we propose a strategy based on this notion.

#### 3.1. Description of local non conformity

**Definition 3.1.** The topology of a welding line  $S$  is an edge, and it is one-sided non conforming. In the first side of  $S$  there is only one cell (the master cell), and on the other side, more than one cell, see Fig. 5.

**Remark 3.2.** The master cell can be viewed as cell with an arbitrary number of nodes. From the point of view of the mesh connectivity, it means that this is a conformal mesh.

We can pass easily from a polygonal unstructured point of view to a non-conformal description, this is a key point of our approach. Such non conforming objects are localized inside the description of the polygonal unstructured mesh class. Here is some description:

- (1) Two end points, the master nodes,  $n1(S)$  and  $n2(S)$ , are not considered to belong to  $S$  (they exist before the creation of  $S$ ).
- (2) The semi-slaves nodes  $S_1, \dots, S_{nz}$  distributed on it,  $nz$  is the number of such nodes, and represents the size of  $S$ .
- (3) The master cell and the local index of  $n1(S)$  inside the master cell.
- (4) It's level, meaning that the overall structure of our mesh class is hybrid between unstructured and treeview. It is defined from the level of master nodes (see (15) for node level definition):

$$l(S) = \min(l(n1(S), l(n2(S)))) \tag{14}$$

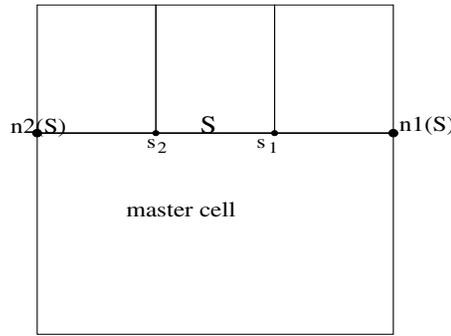


FIGURE 5. A welding line of size two, master cell, master nodes ( $n1(S), n2(S)$ ), semi-slaves nodes ( $S_1, S_2$ ). For polygonal unstructured point of view, the master cell here is an hexagon.

**Remark 3.3.** A semi-slave node belongs to *only one* welding line.

A consequence is the definition of *level* to the nodes class (see Fig. 6). It carries the history of all the nodes, semi-slaves or master node. We need to define this because a master node can be a semi-slaves nodes for other welding line (see Fig. 6). The depth of the mesh is then defined by the minimum of node's level.

**Definition 3.4.** The level of a node  $i$ ,  $l(i)$  is given by the following rule:

- (1)  $l(i) = 0$ , if  $i$  does not belong to any welding line.
- (2) For all semi-slaves node  $i$ ,  $l(i)$  is defined by:

$$l(i) = \min(l(n1(S)), l(n2(S))) - 1, \quad \text{it is computed by Neighborhood.} \tag{15}$$

**Definition 3.5.** The depth of the mesh  $M$  is then defined by:

$$depth(M) = \min_{i \in M} l(i), \quad \in \mathbb{N}^- . \tag{16}$$

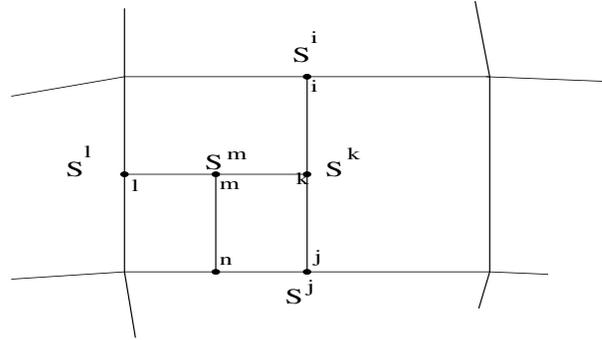


FIGURE 6. The level of nodes and welding line  $l(S)$ ,

Notice that we can deal with an arbitrary nested levels (not always two levels: coarse 0 and fine -1), and we are not dealing with an octree data structure. Indeed, our hybridation data structure permits to deal with non homogeneous refinement at each level.

### 3.2. A strategy

We propose a three steps strategy. It gives naturally an alternative to Lagrangian (or even non-direct ALE) scheme, and it is valid for arbitrary unstructured mesh. In this approach we have decided that the weaker treatment of the global scheme is to track the non-convex cells after the Lagrangian step.

- (1) **Lagrangian Step:** A centered Lagrangian scheme is used to march in time with **ALL nodes considered as degree of freedom**. In converse to [8], we do not impose slipping condition for hanging nodes. We consider that semi-slaves nodes of a welding line can move without any constrained to slip along the two master end points <sup>3</sup>, this a key point of our strategy. We apply one of the centered schemes valid for cells with arbitrary vertices (we use only the polygonal unstructured representation of the mesh, see the remark above).

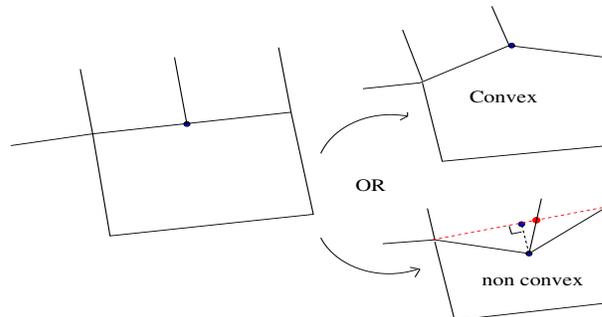


FIGURE 7. Two different evolutions : master cell stay convex or concave (computation of orthogonal projection (blue) or intersection point (red) (more precise for self intersection flux))

<sup>3</sup>this it is the motivation for the “semi”-slave designation

- (2) **Adaptation Step:** After one time step, the new Lagrangian mesh may contain non convex master cells. In this case, we propose to straighten the welding line, see Fig. 7 (we have also an option to do it for any semi-slaves for any welding line). A strategy is to perform a projection for all semi-slave nodes and to recompute the physical quantities on all the modified cells either by self intersection volumes fluxing (2.2.2) either by polygonal clipping. The adaptation step below does not take into account the neighboring cells. This is called buffering (see [14], [8]), some non-checked refined cells are forced to be refined. This operation is done a posteriori. For refinement/derefinement, our approach is based on the construction of nodal information (nodes are marked either “ref”/“deref”). After that, we do a pass on every cell and collect all this nodal information to check if a cell need to be refined. The buffering process is somehow nearly contained in this approach.

We will explain now our refinement/derefinement steps.

- (a) Refinement : At every node  $i$  (or sub set), we compute some criteria (geometrical and physic)

- (i)  $C_i^{GEOM}$  based on a geometric quantities:

$$\begin{cases} C_i^{GEOM} = \frac{\min_j Area(j)}{\max(\max_j Area(j), |\min_j Area(j)|)}, & \text{Area of cell } j \text{ around } i \\ C_i^{GEOM} = \max_j Area(j), \\ C_i^{GEOM} = \max_k l_{i,k}, & \text{length of edge (i,k)}. \end{cases} \quad (17)$$

- (ii)  $C_i^{PHYS}$  based on a physical quantities:

$$\begin{cases} C_i^{PHYS} = |\nabla \rho_i|, & \text{nodal gradient reconstruction,} \\ C_i^{PHYS} = \max_j \rho_j, & \text{density of cell } j \text{ around } i. \end{cases} \quad (18)$$

We mark a node as “ref” (to be refine) if we have both  $C_i^{GEOM} > C^{GEOM,raf}$  and  $C_i^{PHYS} > C^{PHYS,raf}$ , a better way is to use relative value (reference value to be the quantities before the last adaptation step) and tensor quantities. In this work, we have not tuned this analysis and we concentrate much more on the overall approach and code development.

For each cell  $j$ , we collect two numbers: all the vertices marked as “ref”, let  $Nb^{ref}(j)$  this number, and all of the *consecutive* marked vertices  $Nb^{cons}(j)$ . The hypothesis for the three following refinement case is that the cell do not posses any welding line:

- (i) *Isotropic refinement* : we compute a center point (the barycenter or better the centroid), and we join this node to each of edge center of the cell. We apply isotropic refinement when  $Nb^{ref}(j) > \frac{nvertex(j)}{2}$  or  $(Nb^{ref}(j) > 2 \text{ and } nvertex(j) > 4)$  or  $(Nb^{ref}(j) = \frac{nvertex(j)}{2} \text{ and } Nb^{cons}(j) < Nb^{ref}(j))$ <sup>4</sup>. It remains that the only case is for two consecutive “ref” nodes see (ii) and (iii).
- (ii) *Anisotropic refinement (1D)* : For the two consecutive node, if the values of criteria are nearly equal, we join the two mid-edge of the edge before the first node and the edge after the second node, see second draw in Fig. 8.
- (iii) *Quasi one dimensional refinement (Q1D)* : For the two consecutive node, if the values of criteria are different, we do the same construction as for anisotropic case and we had an edge joining each mid edge, see third draw in Fig. 8.

Some times we need to refine a master cell, this is a more complicated task. But, it can be checked that if the welding lines inside this cell are all of length one, then we are able to use one of the previous three adaptations Fig. 8. The consequence is that sometimes, for example in the isotropic case, we need to delete an existing welding line, see Fig. 9. Here, the computation of new

<sup>4</sup>for this last, for sake of simplicity we perform an isotropic refinement, even if it not the best

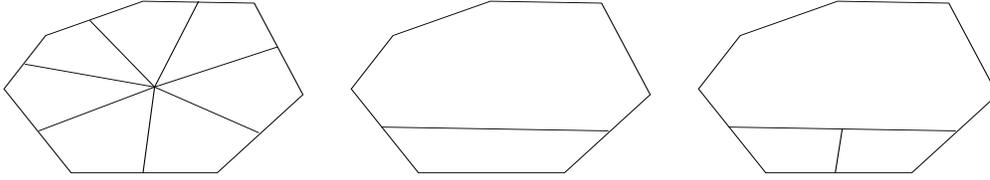


FIGURE 8. Isotropic refinement    anisotropic refinement (1D)    Quasi anisotropic refinement (Q1D)

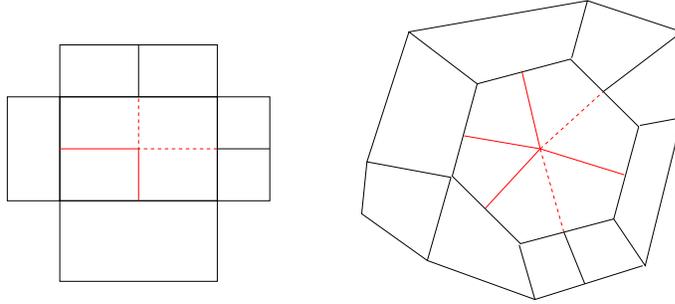


FIGURE 9. Isotropic refinement with welding line cancelation: the dashed new edges are linked to a welding line lead to disappear.

conservative quantities is straightforward for a first order scheme, we just put the coarse value on each of the sub cells.

- (b) Unrefinement: At every node  $i$  (or sub set), we compute some criteria (geometrical and physic)
- (i)  $C_i^{GEOM}$  is based on the following geometric quantities:

$$\begin{cases} C_i^{GEOM} = \frac{\min_j Area(j)}{\max(\max_j Area(j), |\min_j Area(j)|)}, & \text{Area of cell } j \text{ around } i \\ C_i^{GEOM} = \min_j Area(j), \\ C_i^{GEOM} = \min_k l_{i,k}, & \text{length of edge } (i,k). \end{cases} \quad (19)$$

- (ii)  $C_i^{PHYS}$  is based on the following physical quantities:

$$\begin{cases} C_i^{PHYS} = |\nabla \rho_i|, & \text{nodal gradient reconstruction,} \\ C_i^{PHYS} = \min_j \rho_j, & \text{density of cell } j \text{ around } i. \end{cases} \quad (20)$$

We mark a node as “deref” (to be derefined) if we have both  $C_i^{GEOM} < C^{GEOM,deraf}$  and  $C_i^{PHYS} < C^{PHYS,deraf}$ , the coefficient  $C^{GEOM,deraf}$  and  $C^{PHYS,deraf}$  are not the same as  $C^{GEOM,raf}$  and  $C^{PHYS,raf}$  exactly as in [8] to prevent for blinking.

Here, we give our first (and for the moment), the only attempt. We check if there exist two neighboring consecutive nodes  $(i, j)$  marked as “deref”. In this case, we glue the two adjacent cells on both sides of edge  $(i, j)$  see Fig. 10:

- (i) A precise way: compute the new conservative quantities by self intersection scheme or clipping.
- (ii) A crude way: compute the total mass of the patch, and we redistribute the density proportionally to the volume of each new cell.
- (3) **ALE Step:** In case of no welding line see the previous section Rezoning and Remapping, otherwise, we may need to make some few modifications.
- (a) Slaves nodes are moved after the movement of masters endpoints. This process depends on the level of each node (15) (14) (it may be important specially in the case of all the semi-slaves nodes

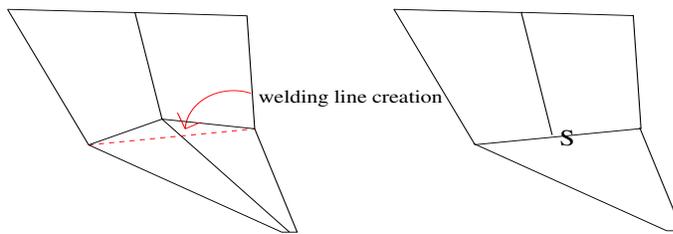
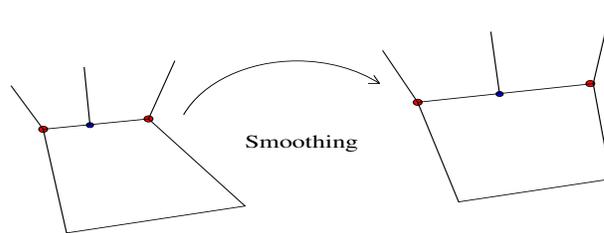
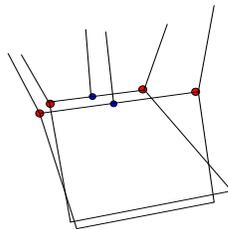


FIGURE 10. Derefinement situation : two adjacent cells are coarsen in only one cell, a welding line is created.

are projected on every welding lines after Lagrangian step). This means that we need to sort the nodes with respect to their level (vertices are taken into account and moved from higher to lower level).



(b) We use the polygonal mesh mode to use Self intersection (or clipping) without changing anything due to the presence of non conformal cells.



## 4. NUMERICAL RESULTS: ALE TESTS

The domain is  $[0, 7] \times [0, 3]$ . We take the following mesh :  $nx = 71, ny = 31$ . The initial condition is at rest and

$$\begin{cases} \rho = 1, P = 0.25, & x < 1, \text{ else} \\ \rho = 1, P = 0.1, & y < 1.5, \text{ else} \\ \rho = 0.1, P = 0.1. \end{cases} \quad (21)$$

and we take  $\gamma = 1.4$ . The rezoning step is done with Escobar adapted (cf. formula (9)) with 5 iterations, and self intersection is performed (nodal mesh quality of [16] is used to control rezoning). We have compared the two centered Lagrangian schemes of Despres/Mazeran and Maire&alwhen used in the same ALE strategy.

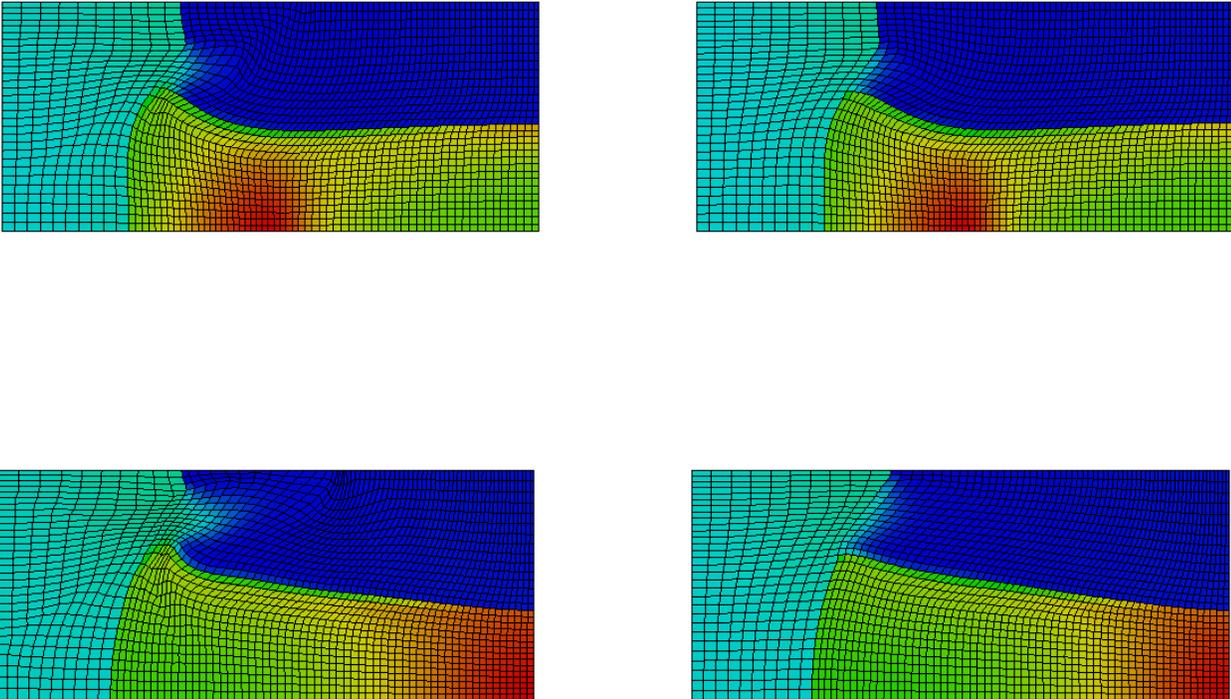


FIGURE 11. Mesh and density at times  $t=5.75$  (up) and  $t=12.25$  (down) (medit visualisation) for Despres/Mazeran (Left) and Maire&al(Right)

In the next, we propose two test cases for refinement step. We have coupled Lagrangian to adaptation step.

- (1) In the first test case, we impose an isotropic pressure on the external radial boundary, the initial condition for physical quantities is at rest,  $\rho = 1, P = 1$  (here the initial mesh contains pentagons). We decide here to consider the three refinements individually of Fig. 8.
- (2) The second test case is done on (21), and we want to illustrate the overall refinement strategy, meaning that all the three cases of Fig. 8 may appear simultaneously as well as the refinement of Fig. 9.

5. NUMERICAL RESULTS: LAGRANGIAN + ADAPTATION TESTS (REFINEMENT)

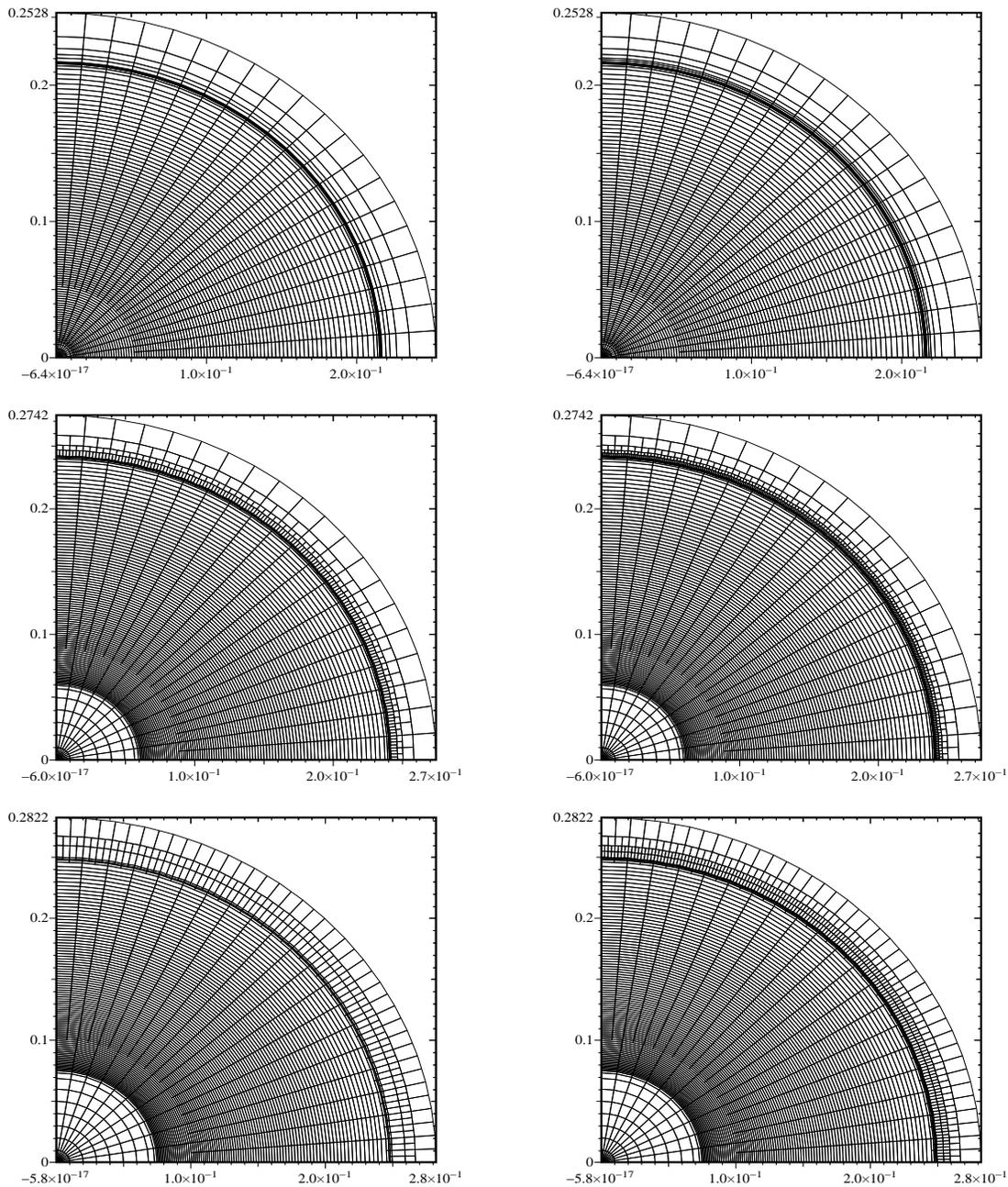


FIGURE 12. Radial test case: The 3 different refinement situations defined by Fig. 8 . The left/right correspond respectively to before/after refinement; Up to down : only 1D refinement, only Quasi 1D refinement, only isotropic refinement.

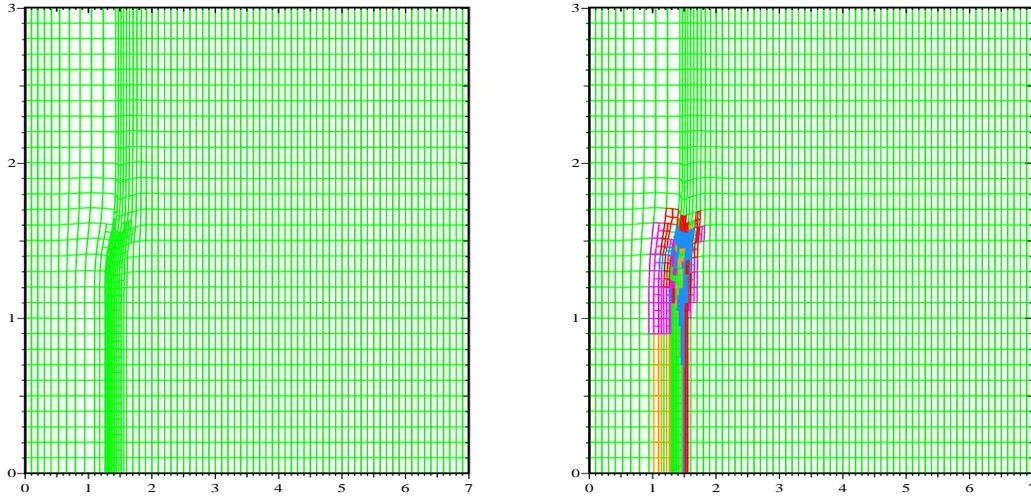


FIGURE 13. Physical data of case (21) with adaptation (without rezoning and remapping): Before / after refinement

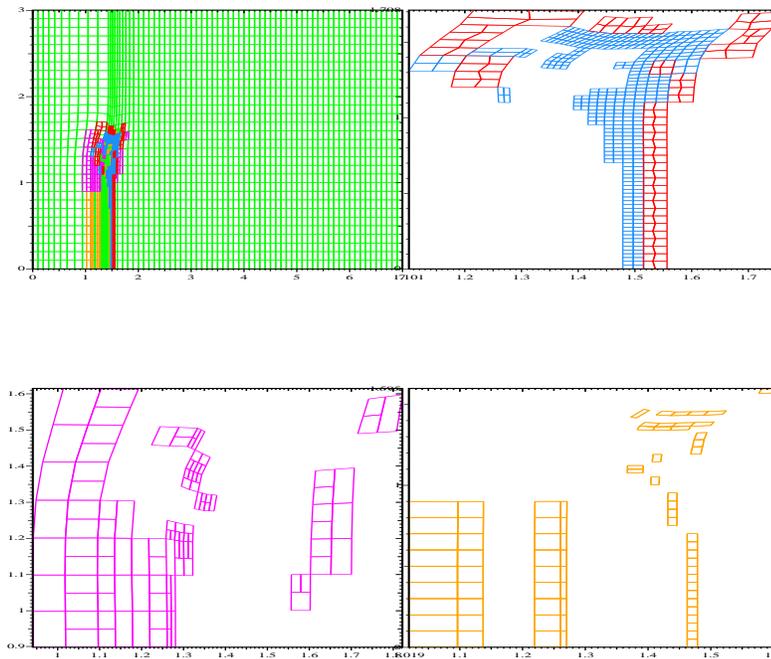


FIGURE 14. Mixed refinement defined for case (21) : Zoom of each individual refinement. Up/Right : isotropic refinement in case of Fig. 8 (blue) or of Fig. 9 (red)), Down/Left : Quasi anisotropic (Q1D), Down/Right : anisotropic (1D).

## 6. CONCLUSION AND FUTURE WORK

In this work, we give a general strategy for an alternative to Lagrangian schemes (with or without ALE) on arbitrary polygonal cells. This is based on three consecutive steps, Lagrangian centered scheme eg. [1] or [19] *with all nodes as degree of freedom simultaneously*, an adaptation step based on refinement/derefinement study. The marked adapted cell process is nodal criteria based, and for refinement cell we propose an early genericity based on sub-cell creation. The conservative sub-quantities are straightforward to compute. The derefinement step is just a first version and need to be enforced toward better resolution. It remains also to coupled ADAP-TATION with ALE, afterwards, we can use a second order version for each step. The code is not multi-material at the moment and although it is still many upgrade to do, we have proposed a way to insert an adaptation step in ALE method.

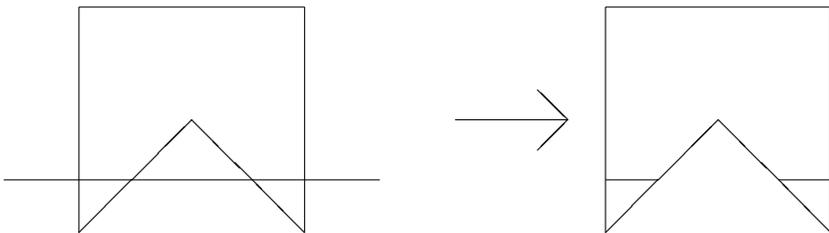
We have developed mainly inside the GO++ numerical package [9]. It was the basis containing all the mesh description (polygonal mesh and hybrid non conformity) and other numerical tools (ALE). For the visualization, we use plotmtv and also medit developed by P.Frey ([www.ann.jussieu.fr/~frey/logiciels/medit.html](http://www.ann.jussieu.fr/~frey/logiciels/medit.html)).

## 7. ANNEX: POLYGON CLIPPING

We want to add an adaptation step to the code that numerically solves the Euler equations. This adaptation step should realize an unrefinement of the mesh. We draw welding lines in order to remove some edges. The only thing we need to do this is to compute the coordinates of the vertices of the new polygons drawn by the welding lines. Roughly speaking, we need to compute the intersection of two meshes. That is why we need to implement a Polygon Clipping Code.

Clipping refers to the removal of part of a scene. Internal clipping removes part of a picture outside a given region. External clipping removes parts inside a region. During CEMRACS 2007, we focused on internal line/polygon clipping.

### Line clipping



The idea of polygon clipping is the following: We have got two polygons  $P_1$  and  $P_2$ , not necessarily convex.  $P_1$  is the window and  $P_2$  is the polygon to be clipped. The result of the clipping of polygons is just the part  $P_3$  of  $P_2$  which is inside  $P_1$ .  $P_3$  can consist of one or several polygons and can in some special cases be empty or equal to  $P_2$ . The output is therefore a list of ordered lists of vertices. There are several algorithms that give that output. The most famous ones are the Sutherland-Hodgman's algorithm and the Weiler-Atherton's algorithm, which are explained below.

### 7.1. Sutherland-Hodgman's algorithm

The Sutherland-Hodgman's polygon clipping algorithm was used in this work. This algorithm is based on a divide-and-conquer strategy that solves a series of simple and identical problems that, when combined, solve the overall problem. The simple problem is to clip a polygon against a single infinite clipping edge. This process outputs the series of vertices that define the clipped polygon.

Within this algorithm, the term *inside* means that the clipping plane's normal vector points towards the point, as defined by:

$$(\vec{p} - \vec{p}_0) \cdot \vec{n} \geq 0,$$

where  $\vec{p}$  is the point in question,  $\vec{p}_0$  is a point on the plane, and  $\vec{n}$  is normal to the plane. Details about inside-outside test and line segment intersection can be found e.g. in [6]. The Sutherland-Hodgman's algorithm is actually simple:

1. For each edge, check both endpoints, ( $s$  and  $p$ ).
  - If both endpoints ( $s$  and  $p$ ) are inside the clipping edge, vertex  $p$  is added to the output list.
  - If vertex  $s$  is inside the clipping edge and vertex  $p$  is outside of it, the intersection point,  $i$ , is output.
  - If both endpoints ( $s$  and  $p$ ) of the edge are outside of the clipping edge, there is no output.
  - If vertex  $p$  is inside of the clipping edge and vertex  $s$  is outside of it, the intersection point,  $i$ , and vertex  $p$  are both added to the output list.
2. The resultant ordered list of vertices forms the clipped polygon.

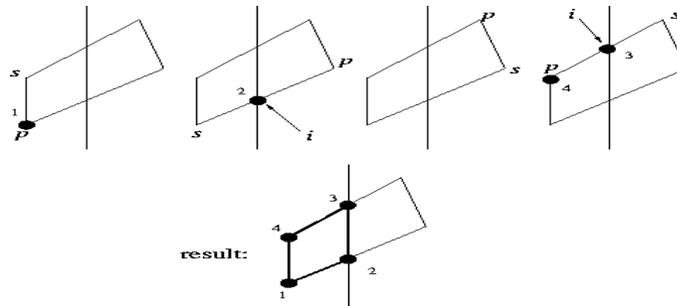


FIGURE 15. Sutherland-Hodgman's polygon clipper example

Within CEMRACS'07 project this algorithm was implemented in C++. Some modifications to this algorithm were carried out. For instance before using the above algorithm we should check three cases: 1. If the polygon is outside the window, then the output is the empty set of vertices; 2. If the polygon is inside the window, the output is all polygon vertices; 3. If polygon intersects the window, the Sutherland-Hodgman's algorithm realizes.

#### Drawbacks:

- Sutherland-Hodgman's algorithm can be used for any arbitrary polygons but the window must be convex.
- In case when there are multiple polygons in the result there will be extra line segments defined between adjacent clipping points in different polygons.

These problems can be solved by using a more general clipping algorithm. For example, the Weiler-Atherton's polygon clipper.

## 7.2. Weiler-Atherton's algorithm

The polygon clipping algorithm described in the next part calculates the analytical sets of the intersection and the difference of two input 2-D polygons that may be concave. The Weiler-Atherton's clipping algorithm is a modification to the Suherland-Hodgman's algorithm that handles concave polygons correctly.

Weiler and Atherton [11] describe an approach which doubles the input polygons in an inside and outside contour. Intersection points between the double contoured polygons are calculated. Using a given set of rules the output polygons are created. The approach labels the edges of the input polygons. This labeling is used in classifying the resulting polygons in the sets  $A \cap B$ ,  $A \setminus B$  and  $B \setminus A$ .

### The basic algorithm:

The algorithm accepts polygons that either do or do not wrap the first vertex to the last vertex. Polygons cannot be self intersecting or co-linear but they can be concave. Duplicate vertices are not allowed either, since this is a degenerate form of self-intersection. If the  $A$  or the  $B$  polygons are self intersecting or contain holes then they must be clipped into disjoint polygons before being sent in to be clipped by the code.

The clipping process consists of the following steps:

1. Calculate the intersections between the two input polygons  $A$  and  $B$ . This results in the same polygons, with the difference that the intersection points with the other polygon are added as vertices to the polygons.
2. Label the edges from both polygons to Inside, Shared, or Outside. Inside means that an edge is inside the other polygon. Shared means that both polygons share this edge. Outside means that an edge is outside the other polygon.
3. Find the minimal polygons which are created by the intersection.
4. Classify all minimal polygons into the output sets  $A \cap B$ ,  $A \setminus B$  and  $B \setminus A$ .

### Draft description:

Two lists are created from the coordinates of each polygons  $A$  and  $B$ . The list entries are labeled as either inside or outside the other polygon. All the polygon intersections are then found and are inserted into both lists, linking the lists at the intersections. Care will be needed where the polygons share an edge. If there are no intersections then one of three situations exist:

1.  $A$  is inside  $B$  - return  $A$  for clipping,  $B$  for merging.
2.  $B$  is inside  $A$  - return  $B$  for clipping,  $A$  for merging.
3.  $A$  and  $B$  do not overlap - return None for clipping or  $A \& B$  for merging.

A list of inbound intersections is then generated. Each intersection in the list is then followed clockwise around the linked lists until the start position is found. One or more concave polygons may produce more than one intersecting polygon. Convex polygons will only have one intersecting polygon.

Proceeding clockwise around the original polygon vertex list, perform the following :

- If a line segment enters the clip region (outside-to-inside vertex pair, add the intersection to the output list, follow the polygon boundary.
- If a line segment leaves the clip region (an inside-to-outside vertex pair), add the intersection to the output list, follow the boundary of the clip region, clockwise.

Fig. 16 shows an example clipping problem generated by the algorithm.

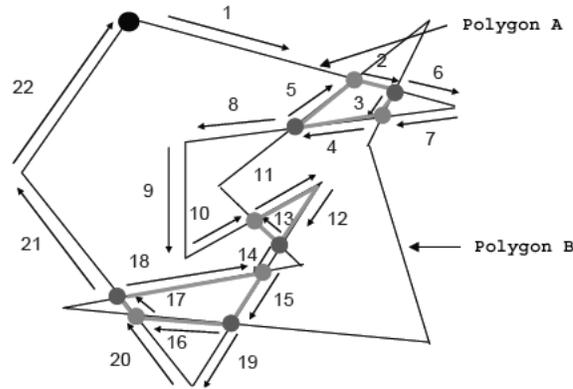


FIGURE 16. An example of the Weiler-Atherton's clipping process with two input polygons. The area shared by the polygons is bordered with grey.

The same algorithm can be used for merging two polygons by starting at the outbound intersections rather than the inbound ones. However this can produce counter-clockwise holes. Some polygon combinations may be difficult to resolve, specially when holes are allowed. Points on the edge of the polygon may be considered as both in and out until their status can be confirmed after the intersections have been found and verified, however this increases the complexity (A detailed analysis of the complexity of the polygon clipping problem can be found e.g. in [5] and [3]).

During CEMRACS 2007 our project led us to study of the polygon clipping. We have studied two main algorithms that realize polygon clipping: Sutherland-Hodgman's and Weiler-Atherton's algorithms, which we have implemented in C++. Polygon clipping seems to be a useful and convenient tool to make some operations on meshes (intersection, union, removal of a part, extraction of a polygonal part, etc.).

The authors would like to thank S.Delpino, B.Despres, P.H.Maire, B.Nkongha and B.Rebourcet for fruitful discussions around Lagrangian/ALE schemes during the CEMRACS 2007.

## REFERENCES

- [1] Despres B. and Mazeran C. Lagrangian gas dynamics in two dimensions and lagrangian systems. *Arch. Ration. Mech. Anal.*, 178:327–372, 2005.
- [2] Jun B.I. A modified equipotential method for grid relaxation. *J.Comput. Physic*, 2003.
- [3] Kim D.H. and Kim J. An extension of polygon clipping to resolve degenerate cases. *Computer-Aided Design Applications*, 3:447–456, 2006.
- [4] Caramana E.J., Burton D.E., Shashkov M.J., and Whalen P.P. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. *J.Comput.Physics*, 146:227–262, 1998.
- [5] Greiner G. and Horman K. Efficient clipping of arbitrary polygons. *ACM Transactions on Graphics*, 17(2):71–83, 1998.
- [6] Thomas S. Hall and Kenneth B. Kent. Design, verification and implementation of a polygon clipping application using co-design techniques. Technical report, University of New Brunswick, June 2007.
- [7] Campbell J. and Shashkov M. A compatible lagrangian hydrodynamics algorithm for unstructured grids. *Selcuk J.Appl.Math.*, 4:53–70, 2003.
- [8] Morrell J., Sweby P., and Barlow A. A cell by cell anisotropic adaptive mesh ale scheme for the numerical solution of the euler equations. *J. Comp. Phys.*, 226(1):1152–1180, 2007.
- [9] Benamou J.D. and Hoch Ph. Go++ : A modular lagrangian/eulerian software for hamilton jacobi equations of geometric optics type. *M2AN*, 36(5):883–905, 2002.

- [10] Escobar J.M., Rodriguez E., Montenegro R., Montero G., and Gonzalez-Yuste J.M. Simultaneous untangling and smoothing of tetrahedral meshes. *Computer Methods in Applied Mechanics and Engineering*, 192(25):2775–2787, June 2003.
- [11] Weiler K. and Atherton P. Hidden surface removal using polygon area sorting. In *Proc. of the 4th ann. Conf. on Computer Graphics and Interactive Techniques*, pages 214–222, 1977.
- [12] Margolin L.G. and Shashkov M. Second order sign preserving remapping on general grids. Technical report, LA-UR02-525, 2002.
- [13] Margolin L.G. and Shashkov M. Second order sign preserving remapping on general grids. *J. Comput. Phys.*, 184:266–298, 2003.
- [14] Berger M.J. and Colella P. Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.*, 82:64, 1989.
- [15] Wilkins M.L. Calculation of elastic-plastic flow. *Methods in Computational Physics*, 3:211–263, 1964.
- [16] Hoch P. An Arbitrary Lagrangian-Eulerian strategy to solve compressible flows. 2008. submitted.
- [17] Knupp P., Margolin L.G., and Shashkov M. Reference jacobian optimization-based rezone strategies for arbitrary lagrangian-eulerian methods. *J. Comput. Phys.*, 176(1):93–112, 2002.
- [18] Knupp P. and Shashkov M. Optimization-based reference-matrix rezone strategies for arbitrary lagrangian-eulerian methods on unstructured grids. *Selcuk J. Appl. Math.*, 3(81-99), 2000.
- [19] Maire P.H., Abgrall R., Breil J., and Ovardia J. A cell-centered lagrangian scheme for two-dimensional compressible flow problems. *SIAM J.Sci. Comput.*, 29(4):1781–1824, 2007.
- [20] Knupp P.M. Algebraic mesh quality metrics. *SIAM J.Sci. Comput.*, 23:193–218, 2001.
- [21] Loubère R. and Shashkov M.J. A subcell remapping method on staggered polygonal grids for arbitrary-lagrangian eulerian methods. *J. Comput. Phys.*, 209:105–138, 2005.
- [22] Tipton R. Grid optimization by equipotential relaxation. Technical report, LLNL, 1992.